User's Manual

32

RA8M2 Group

User's Manual: Hardware

32-Bit MCU

Renesas Advanced (RA) Family

Renesas RA8 Series

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

Rev.1.30 Feb 2026

Corporate Headquarters

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

Contact Information

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

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

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

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

Specification Differences Between Products

There are the following specification differences in the RA8M2 group products, depending on chip versions.

Table 1.1 Specification Differences Depending on Chip Versions

Chapter		Specification Differences	
		Chip version A	Chip version B
section 1, Overview	section 1.3. Part Numbering	Production identification code is 0	Production identification code is 1
section 7, Option-Setting Memory	section 7.2.11. FSBLCTRL0 : FSBL Control Register 0	Prohibit to set the FSBLEN [2:0] bits to 000b	The FSBLEN[2:0] bits can be set to 000b
section 9, Clock Generation Circuit	section 9.2.36. MOSCSCR : Main Clock Oscillator Standby Control Register	Prohibit to set the MOSCSOKP bit to 1	The MOSCSOKP bit can be set to 1
	section 9.2.37. HOCOSCR : High-Speed On-Chip Oscillator Standby Control Register	Prohibit to set the HOCOSOKP bit to 1	The HOCOSOKP bit can be set to 1
section 11, Low Power Mode	section 11.7. Voltage Scaling Control	During a DVFS transition (VSCR.VSCMTSF = 1), Don’t access to the CM85's TCM and CM85's cache. Before voltage scaling operation, disable the I-cache and D-cache by setting CCR.DC and CCR.IC to 0, and then deactivate them by setting MSCR.DCACTIVE and MSCR.ICACTIVE to 0	During a DVFS transition (VSCR.VSCMTSF = 1), there is no restriction on access to the CM85's TCM and CM85's cache
section 59, MRAM	section 59.5.43. MCUVER : MCU Version Register	Value after reset is 0x1	Value after reset is 0x2
section 62, Electrical Characteristics	section 62.2.2. I/O V_{IH}, V_{IL}	TCK/SWCLK 1.62V or above V_{IH} : Min = VCC × 0.7 V_{IL} : Max = VCC × 0.3	TCK/SWCLK 1.62V or above V_{IH} : Min = VCC × 0.8 V_{IL} : Max = VCC × 0.2 ΔV_T : Min = VCC × 0.05

Preface

1. About this document

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

2. Audience

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

3. Renesas Publications

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

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

4. Numbering Notation

The following numbering notation is used throughout this manual:

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

5. Typographic Notation

The following typographic notation is used throughout this manual:

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

6. Unit and Unit Prefix

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

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

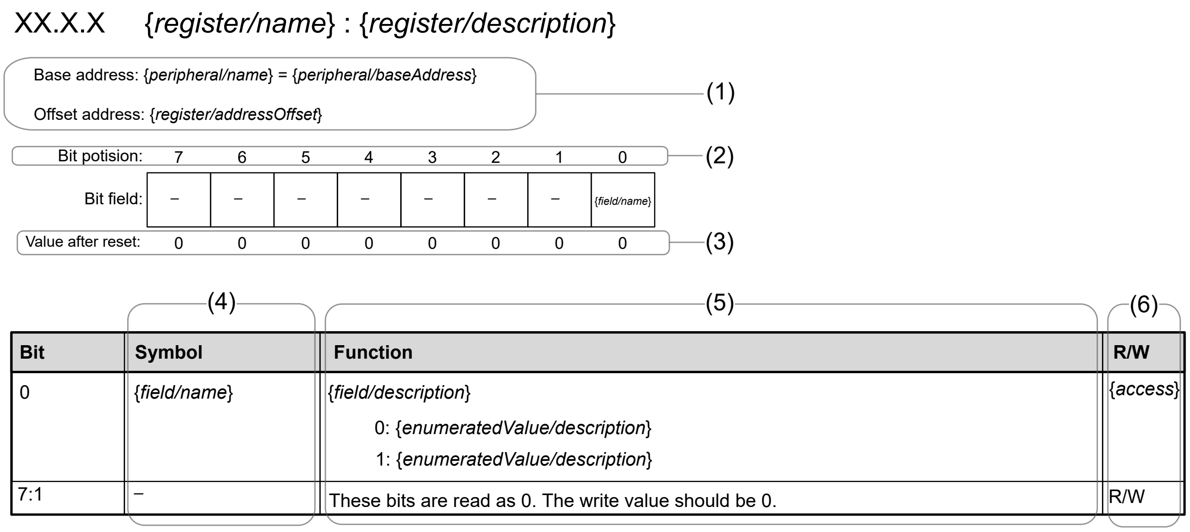
7. Special Terms

The following terms have special meanings.

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

8. Register Description

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



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

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

(2) Bit number

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

(3) Value after reset

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

0: Indicates that the value is 0 after a reset.

1: Indicates that the value is 1 after a reset.

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

(4) Symbol

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

(5) Function

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

(6) R/W

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

R/W: The bit field is readable and writable.

R: The bit field is readable only. Writing to this bit field has no effect.

W: The bit field is writable only. The read value is the same as after a reset unless specified otherwise.

9. Abbreviations

Abbreviations used in this document are shown in the following table.

Abbreviation	Description
AES	Advanced Encryption Standard
AHB	Advanced High-performance Bus
AHB-AP	AHB Access Port
APB	Advanced Peripheral Bus
ARC	Alleged RC
ATB	Advanced Trace Bus
BCD	Binary Coded Decimal
BSDL	Boundary Scan Description Language
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
ETB	Embedded Trace Buffer
ETM	Embedded Trace Macrocell
FLL	Frequency Locked Loop
FPU	Floating Point Unit
HMI	Human Machine Interface
IrDA	Infrared Data Association
LSB	Least Significant Bit
MSB	Most Significant Bit
NVIC	Nested Vector Interrupt Controller
OTP	One Time Programmable memory
PC	Program Counter
PFS	Port Function Select
PLL	Phase Locked Loop
POR	Power-on reset
PWM	Pulse Width Modulation
RSA	Rivest Shamir Adleman
SHA	Secure Hash Algorithm
S/H	Sample and Hold
SP	Stack Pointer
SWD	Serial Wire Debug
SW-DP	Serial Wire-Debug Port
TRNG	True Random Number Generator
UART	Universal Asynchronous Receiver/Transmitter
VCO	Voltage Controlled Oscillator

10. Proprietary Notice

11. Feedback on the product

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

Contents

Renesas RA8M2 Group

User's Manual

High-performance 1 GHz Arm® Cortex®-M85 core, 250 MHz Arm® Cortex®-M33 core, up to 1 MB code MRAM, and 2 MB SRAM with ECC. High-integration with Layer 3 Ethernet Switch Module, USB 2.0 High-Speed, CANFD, SDHI, I3C, Octal SPI, Decryption on-the-fly, and advanced analog. Integrated Renesas Security IP with cryptography accelerators, key management support, tamper detection and power analysis resistance in concert with Arm® TrustZone for integrated Secure element functionality.

Features

■ Arm® Cortex®-M85 Core

- Armv8.1-M architecture profile
- Armv8-M Security Extension
- Maximum operating frequency: 1 GHz
- Memory Protection Unit (Arm MPU)
  - Protected Memory System Architecture (PMSAv8)
  - Secure MPU (MPU_S): 8 regions
  - Non-secure MPU (MPU_NS): 8 regions
- SysTick timer
  - Embeds two Systick timers: Secure and Non-secure instance
  - Driven by CPUCLK0 or MOCO divided by 8
- CoreSight™ ETM-M85

■ Arm® Cortex®-M33 core

- Armv8-M architecture profile
- Armv8-M Security Extension
- Maximum operating frequency: 250 MHz
- Memory Protection Unit (Arm MPU)
  - Protected Memory System Architecture (PMSAv8)
  - Secure MPU (MPU_S): 8 regions
  - Non-secure MPU (MPU_NS): 8 regions
- SysTick timer
  - Embeds two Systick timers: Secure and Non-secure instance
  - Driven by CPUCLK1 or MOCO divided by 8
- CoreSight™ ETM-M33

■ Memory

- Up to 1-MB MRAM
- 2 MB SRAM including 256 KB of CM85 TCM and 128 KB of CM33 TCM
- Up to 8-MB Flash for SiP product

■ Connectivity

- Serial Communications Interface (SCI) × 10, up to 60 Mbps
- I^2C bus interface (IIC) × 3
- I^3C bus interface (I3C)
- Serial Peripheral Interface (SPI) × 2, up to 166 Mbps
- Octal Serial Peripheral Interface (OSPI) × 2, up to 333 MB/s
- USB 2.0 Full-Speed Module (USBFS)
- USB 2.0 High-Speed Module (USBHS)
- CAN with Flexible Data-rate (CANFD) × 2
- Layer 3 Ethernet Switch Module (ESWM)
  - 2 × Gigabit Ethernet
- SD/MMC Host Interface (SDHI) × 2
- Serial Sound Interface Enhanced (SSIE) × 2
- Pulse Density Modulation Interface (PDMIF)

■ Analog

- 16-bit A/D Converter (ADC16H) × 2, up to 23 channels
- 12-bit D/A Converter (DAC12) × 2
- High-Speed Analog Comparator (ACMPHS) × 4
- Temperature Sensor (TSN)

■ Timers

- General PWM Timer 32-bit (GPT32) with High Resolution × 4
  - 52 ps resolution in 300 MHz
- General PWM Timer 32-bit (GPT32) × 10
- Low Power Asynchronous General Purpose Timer (AGT) × 2
- Ultra-Low-Power Timer (ULPT) × 2

■ Security and Encryption

- Renesas Security IP (RSIP-E50D)
- Arm® TrustZone®
- Privileged control
- Device lifecyle management
- Secure boot
  - Immutable first stage boot loader in OTP
- Decryption on-the-fly (DOTF)
- Pin function
  - Up to three tamper-resistant pins
  - Secure pin multiplexing
- HUK zeroization

■ System and Power Management

- Low power modes
- Battery backup function (VBATT)
- Realtime Clock (RTC) with calendar and VBATT support
- Event Link Controller (ELC)
- Data Transfer Controller (DTC) × 2
- DMA Controller (DMAC) × 16
- Power-on reset
- Programable Voltage Detection (PVD) with voltage settings
- Watchdog Timer (WDT) × 2
- Independent Watchdog Timer (IWDT)

■ Human Machine Interface (HMI)

- Capture Engine Unit (CEU)

■ Multiple Clock Sources

- Main clock oscillator (MOSC) (8 to 48 MHz)
- Sub-clock oscillator (SOSC) (32.768 kHz)
- High-speed on-chip oscillator (HOCO) (16/18/20/32/48 MHz)
- Middle-speed on-chip oscillator (MOCO) (8 MHz)
- Low-speed on-chip oscillator (LOCO) (32.768 kHz)
- Clock trim function for HOCO/MOCO/LOCO
- PLL1/PLL2
- Clock out support

■ General-Purpose I/O Ports

- 5-V tolerance, open drain, input pull-up, switchable driving ability

■ Operating Voltage

- Standard product
  - VCC/VCC2: 1.62 to 3.63 V
- SiP product
  - VCC/VCC2: 1.62 to 3.63 V/1.70 to 2.00 V

■ Operating Junction Temperature and Packages

- Tj = 0 °C to +95 °C
  - 289-pin BGA (12 mm × 12 mm, 0.65 mm pitch)
  - 224-pin BGA (11 mm × 11 mm, 0.65 mm pitch)
  - 303-pin BGA (15 mm × 15 mm, 0.8 mm pitch)
  - 169-pin BGA (7 mm × 7 mm, 0.5 mm pitch)
- Tj = -40 °C to +105 °C
  - 289-pin BGA (12 mm × 12 mm, 0.65 mm pitch)
  - 224-pin BGA (11 mm × 11 mm, 0.65 mm pitch)
  - 303-pin BGA (15 mm × 15 mm, 0.8 mm pitch)
  - 169-pin BGA (7 mm × 7 mm, 0.5 mm pitch)
- Tj = -40 °C to +125 °C
  - 289-pin BGA (12 mm × 12 mm, 0.65 mm pitch)
  - 224-pin BGA (11 mm × 11 mm, 0.65 mm pitch)
  - 176-pin HLQFP (24 mm × 24 mm, 0.5 mm pitch)

1. Overview

The MCU integrates multiple series of software-compatible Arm®-based 32-bit cores that share a common set of Renesas peripherals to facilitate design scalability and efficient platform-based product development.

The MCU in this series incorporates a high-performance Arm® Cortex®-M85 core running up to 1 GHz and Arm® Cortex®-M33 core running up to 250 MHz with the following features:

- Up to 1 MB MRAM
- 2 MB SRAM (256 KB of CM85 TCM RAM, 128 KB CM33 TCM RAM, 1664 KB of user SRAM)
- Octal Serial Peripheral Interface (OSPI)
- Layer 3 Ethernet Switch Module (ESWM), USBFS, USBHS, SD/MMC Host Interface
- Analog peripherals
- Security and safety features

1.1 Function Outline

Table 1.1 Arm core

Feature	Functional description
Arm® Cortex®-M85 core	● Maximum operating frequency: up to 1 GHz ● Arm® Cortex®-M85 core – Revision: (r1p1-00rel0) – ARMv8.1-M architecture profile – Armv8-M Security Extension – Floating Point Unit (FPU) compliant with the ANSI/IEEE Std 754-2008 Scalar half, single, and double-precision floating-point operation – M-profile Vector Extension (MVE) Integer, half-precision, and single-precision floating-point MVE (MVE-F) ● Arm® Memory Protection Unit (Arm MPU) – Protected Memory System Architecture (PMSAv8) – Secure MPU (MPU_S): 8 regions – Non-secure MPU (MPU_NS): 8 regions ● SysTick timer – Embeds two Systick timers: Secure instance (SysTick_S) and Non-secure instance (SysTick_NS) – Driven by CPUCLK0 or MOCO divided by 8 ● CoreSight™ ETM-M85
Arm® Cortex®-M33 core	● Maximum operating frequency: up to 250 MHz ● Arm® Cortex®-M33 core – Revision: (r0p4-00rel2) – ARMv8-M architecture profile – Armv8-M Security Extension – Armv8-DSP Extension – Floating Point Unit (FPU) compliant with the ANSI/IEEE Std 754-2008 single-precision floating-point operation ● Arm® Memory Protection Unit (Arm MPU) – Protected Memory System Architecture (PMSAv8) – Secure MPU (MPU_S): 8 regions – Non-secure MPU (MPU_NS): 8 regions ● SysTick timer – Embeds two Systick timers: Secure instance (SysTick_S) and Non-secure instance (SysTick_NS) – Driven by CPUCLK1 or MOCO divided by 8 ● CoreSight™ ETM-M33

Table 1.2 Memory (1 of 2)

Feature	Functional description
Code MRAM	Maximum 1 MB of code MRAM. See section 59, MRAM.
Flash memory	System in package (SiP) maximum 8 MB serial flash memory. See section 44.4. SiP Product Configuration.

Table 1.2 Memory (2 of 2)

Feature	Functional description
Option-setting memory	The option-setting memory determines the state of the MCU after a reset. See section 7, Option-Setting Memory.
SRAM	On-chip high-speed SRAM with Error Correction Code (ECC). See section 58, SRAM.
OTP	On-chip OTP contains First Stage Bootloader (FSBL) General purpose 96-byte OTP

Table 1.3 System

Feature	Functional description
Operating modes	Three operating modes: ● Single-chip mode ● JTAG boot mode ● SCI/USB boot mode See section 4, Operating Modes.
Resets	This MCU provides the following 21 types of reset. See section 6, Resets.
Programmable Voltage Detection (PVD)	The Programmable Voltage Detection (PVD) module monitors the voltage level input to the VCC pin. The detection level can be selected by register settings. The PVD module consists of five separate voltage level detectors (PVD0, PVD1, PVD2, PVD4, PVD5). These PVDs measure the voltage level input to the VCC pin. PVD registers allow your application to configure detection of VCC changes at various voltage thresholds. See section 8, Programmable Voltage Detection (PVD).
Clocks	● Main clock oscillator (MOSC) ● Sub-clock oscillator (SOSC) ● High-speed on-chip oscillator (HOCO) ● Middle-speed on-chip oscillator (MOCO) ● Low-speed on-chip oscillator (LOCO) ● PLL1/PLL2 ● Clock out support See section 9, Clock Generation Circuit.
Clock Frequency Accuracy Measurement Circuit (CAC)	The Clock Frequency Accuracy Measurement Circuit (CAC) counts pulses of the clock to be measured (measurement target clock) within the time generated by the clock selected as the measurement reference (measurement reference clock), and determines the accuracy depending on whether the number of pulses is within the allowable range. When measurement is complete or the number of pulses within the time generated by the measurement reference clock is not within the allowable range, an interrupt request is generated. See section 10, Clock Frequency Accuracy Measurement Circuit (CAC).
Interrupt Controller Unit (ICU)	The Interrupt Controller Unit (ICU) controls which event signals are linked to the Nested Vector Interrupt Controller (NVIC), DMA Controller (DMAC) module and the Data Transfer Controller (DTC) modules. The ICU also controls non-maskable interrupts. See section 14, Interrupt Controller Unit (ICU).
Low power modes	Power consumption can be reduced in multiple ways, including setting clock dividers, controlling EBCLK output, controlling SDCLK output, stopping modules, power gating control, selecting operating power control modes in normal operation, and transitioning to low power modes and processor low power modes. See section 11, Low Power Mode.
Battery backup function	A battery backup function is provided for partial powering by a battery. The battery-powered area includes the RTC, SOSC, backup register, tamper detection and VBATT_R voltage drop detection and switch between VCC and VBATT. See section 12, Battery Backup Function.
Register write protection	The register write protection function protects important registers from being overwritten due to software errors. The registers to be protected are set with the Protect Register (PRCR_S and PRCR_NS). See section 13, Register Write Protection Function.
Memory Protection Unit (MPU)	All bus masters have Memory Protection Units (MPUs). See section 16, Memory Protection Unit (MPU).

Table 1.4 Event link

Feature	Functional description
Event Link Controller (ELC)	The Event Link Controller (ELC) uses the event requests generated by various peripheral modules as source signals to connect them to different modules, allowing direct link between the modules without CPU intervention. See section 19, Event Link Controller (ELC).

Table 1.5 Direct memory access

Feature	Functional description
Data Transfer Controller (DTC)	A Data Transfer Controller (DTC) module is provided for transferring data when activated by an interrupt request. See section 18, Data Transfer Controller (DTC).
DMA Controller (DMAC)	The 8-channel direct memory access controller (DMAC) that can transfer data without intervention from the CPU. When a DMA transfer request is generated, the DMAC transfers data stored at the transfer source address to the transfer destination address. See section 17, DMA Controller (DMAC).

Table 1.6 External bus interface

Feature	Functional description
External buses	● CS area (ECBI): Connected to the external devices (external memory interface) ● SDRAM area (ECBI): Connected to the SDRAM (external memory interface) ● OSPI0 area (OSPI0BI): Connected to the OSPI0 (external device interface) ● OSPI1 area (OSPI1BI): Connected to the OSPI1 (external device interface)

Table 1.7 Timers (1 of 2)

Feature	Functional description
General PWM Timer (GPT)	The General PWM Timer (GPT) is a 32-bit timer with GPT32 × 14 channels. PWM waveforms can be generated by controlling the up-counter, down-counter, or the up- and down-counter. In addition, PWM waveforms can be generated for controlling brushless DC motors. The GPT can also be used as a general-purpose timer. See section 22, General PWM Timer (GPT).
PWM Delay Generation Circuit (PDG)	The PWM Delay Generation circuit (PDG) has 4 channels delay circuits that can connect to the GPT. The PDG can control the rise and fall edge timing with which the PWM output for the GPT320 through the GPT323. See section 23, PWM Delay Generation Circuit (PDG).
Port Output Enable for GPT (POEG)	The Port Output Enable (POEG) function can place the General PWM Timer (GPT) output pins in the output disable state See section 21, Port Output Enable for GPT (POEG).
Low Power Asynchronous General Purpose Timer (AGT)	The Low Power Asynchronous General Purpose Timer (AGT) is a 16-bit timer that can be used for pulse output, external pulse width or period measurement, and counting external events. This timer consists of a reload register and a down counter. The reload register and the down counter are allocated to the same address, and can be accessed with the AGT register. See section 24, Low Power Asynchronous General Purpose Timer (AGT).
Ultra-Low-Power Timer (ULPT)	The Ultra-Low-Power Timer (ULPT) is a 32-bit timer which can be used for outputting pulses or counting external events. This 32-bit timer consists of reload registers and a down-counter. The reload registers and the down-counter are allocated to the same address and can be accessed through the ULPTCNT register. See section 25, Ultra-Low-Power Timer (ULPT)
Realtime Clock (RTC)	The realtime clock (RTC) has two counting modes, calendar count mode and binary count mode, that are used by switching register settings. For calendar count mode, the RTC has a 100-year calendar from 2000 to 2099 and automatically adjusts dates for leap years. For binary count mode, the RTC counts seconds and retains the information as a serial value. Binary count mode can be used for calendars other than the Gregorian (Western) calendar. See section 26, Realtime Clock (RTC).
Watchdog Timer (WDT)	The Watchdog Timer (WDT) is a 14-bit down counter that can be used to reset the MCU when the counter underflows because the system has run out of control and is unable to refresh the WDT. In addition, the WDT can be used to generate a non-maskable interrupt or an underflow interrupt. See section 27, Watchdog Timer (WDT).

Table 1.7 Timers (2 of 2)

Feature	Functional description
Independent Watchdog Timer (IWDT)	The Independent Watchdog Timer (IWDT) has a 14-bit down-counter, which resets the MCU by a reset output when the down-counter underflows. Alternatively, generation of an interrupt request when the counter underflows can be selected. This enables detection of a program runaway taking the refresh interval into account. The IWDT has two start modes: auto start mode, in which counting automatically starts after release from the reset state, and register start mode, in which counting is started by refreshing (writing to a specific register). See section 28, Independent Watchdog Timer (IWDT).

Table 1.8 Communication interfaces (1 of 2)

Feature	Functional description
Serial Communications Interface (SCI)	The Serial Communications Interface (SCI) × 10 channels have asynchronous and synchronous serial interfaces: ● Asynchronous interfaces (UART and Asynchronous Communications Interface Adapter (ACIA)) ● 8-bit clock synchronous interface ● Simple IIC (master-only) ● Simple SPI ● Smart card interface ● Manchester interface ● Simple LIN interface The smart card interface complies with the ISO/IEC 7816-3 standard for electronic signals and transmission protocol. All channels have FIFO buffers to enable continuous and full-duplex communication, and the data transfer speed can be configured independently using an on-chip baud rate generator. The maximum rate supported on this MCU. Refer to the electrical characteristics for the actual rate. See section 38, Serial Communications Interface (SCI).
I²C Bus interface (IIC)	The I²C Bus interface (IIC) has 3 channels. The IIC module conforms with and provides a subset of the NXP I²C (Inter-Integrated Circuit) bus interface functions. See section 39, I²C Bus Interface (IIC).
I3C Bus Interface (I3C)	The I3C Bus Interface (I3C) has 1 channel. The I3C module conform with and provide a subset of the NXP I²C (Inter-Integrated Circuit) bus interface functions and a subset of the MIPI I3C. See section 40, I3C Bus Interface (I3C).
Serial Peripheral Interface (SPI)	The Serial Peripheral Interface (SPI) provides high-speed full-duplex synchronous serial communications with multiple processors and peripheral devices. The maximum rate supported on this MCU. Refer to the electrical characteristics for the actual rate. See section 43, Serial Peripheral Interface (SPI).
Control Area Network with Flexible Data-Rate Module (CANFD)	The CAN with Flexible Data-Rate (CANFD) module can handle classical CAN frames and CANFD frames complied with ISO 11898-1 standard. The module supports 4 transmit buffers per channel and 16 receive buffers per channel. See section 41, CAN with Flexible Data-rate (CANFD).
USB 2.0 Full-Speed module (USBFS)	The USB 2.0 Full-Speed module (USBFS) can operate as a host controller or device controller. The module supports full-speed and low-speed (host controller only) transfer as defined in Universal Serial Bus Specification 2.0. The module has an internal USB transceiver and supports all of the transfer types defined in Universal Serial Bus Specification 2.0. The USB has buffer memory for data transfer, providing a maximum of 10 pipes. Pipes 1 to 9 can be assigned any endpoint number based on the peripheral devices used for communication or based on your system. See section 36, USB 2.0 Full-Speed Module (USBFS).
USB 2.0 High-speed Module (USBHS)	The USB 2.0 High-Speed Module (USBHS) that operates as a host or a device controller compliant with the Universal Serial Bus (USB) Specification revision 2.0. The host controller supports USB 2.0 high-speed, full speed, and low-speed transfers, and the device controller supports USB 2.0 high-speed and full-speed transfers. The USBHS has an internal USB transceiver and supports all of the transfer types defined in the USB 2.0 specification. The USBHS has FIFO buffer for data transfers, providing a maximum of 10 pipes. See section 37, USB 2.0 High-Speed Module (USBHS).

Table 1.8 Communication interfaces (2 of 2)

Feature	Functional description
Octal Serial Peripheral Interface (OSPI)	The Octal Serial Peripheral Interface (OSPI) is a memory controller that supports Expanded Serial Peripheral Interface (xSPI) (JEDEC Standard JESD251, JESD251-1 and JESD252). The OSPI supports 1-bit, 2-bit, 4-bit and 8-bit protocols. JESD251 specifies two interface profiles where profile 1.0 is Octal SPI and profile 2.0 is HyperBus^{TM} (HyperRAM^{TM} and HyperFlash^{TM}). OSPI supports QSPI protocol. See section 44, Octal Serial Peripheral Interface (OSPI).
Serial Sound Interface Enhanced (SSIE)	The Serial Sound Interface Enhanced (SSIE) peripheral provides functionality to interface with digital audio devices for transmitting I^2S/Monaural/TDM audio data over a serial bus. The SSIE supports an audio clock frequency of up to 50 MHz, and can be operated as a slave or master receiver, transmitter, or transceiver to suit various applications. The SSIE includes 32-stage FIFO buffers in the receiver and transmitter, and supports interrupts and DMA-driven data reception and transmission. See section 46, Serial Sound Interface Enhanced (SSIE).
SD/MMC Host Interface (SDHI)	The Secure Digital (SD) Card and Multi Media Card (MMC) Host Interface provides the functionality required to connect a variety of external memory cards to the MCU. The SDHI supports both 1- and 4-bit buses for connecting memory cards that support SD, SDHC, and SDXC formats. When developing host devices that are compliant with the SD Specifications, you must comply with the SD Host/Ancillary Product License Agreement (SD HALA). The MMC interface supports 1-bit, 4-bit, and 8-bit MMC buses that provide eMMC 4.51 (JEDEC Standard JESD 84-B451) device access. This interface also provides backward compatibility and supports high-speed SDR transfer modes. See section 47, SD/MMC Host Interface (SDHI).
Layer 3 Ethernet Switch Module (ESWM)	The Layer 3 Ethernet Switch Module (ESWM) consists of two channels of Gigabit Ethernet controller, an Ethernet switch with high level routing capability, and multi-protocol interface support. The Gigabit Ethernet controller conforms to the definition of the Ethernet MAC (Media Access Control) layer in the IEEE 802.3 standard. This can transmit and receive Ethernet (IEEE 802.3) frames by connecting with an external physical-layer LSI chip (PHY-LSI) which complies with the standard. The Ethernet switch allows autonomous frame routing within a same network interface protocol, or between different network interfaces protocols or optimized gateway applications. See section 29, Layer 3 Ethernet Switch Module (ESWM).
Pulse Density Modulation Interface (PDMIF)	PDM-IF has maximum three channels that are connectable with external microphone which outputs the pulse density modulated (PDM) signal. PDM-IF is connectable with up to three external microphones. PDM-IF can filter and convert 1-bit digital data streams that were pulse density modulated at a high sampling rate into 20-bit or 16-bit digital data at a lower sampling rate. See section 49, Pulse Density Modulation Interface (PDM-IF).

Table 1.9 Analog

Feature	Functional description
16-bit A/D Converter (ADC16H)	A 16-bit A/D Converter is provided. Up to 23 analog input channels are selectable. Temperature sensor output, and internal reference voltage and VBATT 1/6 voltage monitor are selectable for conversion. See section 53, 16-bit A/D Converter (ADC16H).
12-bit D/A Converter (DAC12)	A 12-bit D/A Converter (DAC12) is provided. See section 54, 12-Bit D/A Converter (DAC12).
Temperature Sensor (TSN)	The on-chip Temperature Sensor (TSN) determines and monitors the die temperature for reliable operation of the device. The sensor outputs a voltage directly proportional to the die temperature, and the relationship between the die temperature and the output voltage is fairly linear. The output voltage is provided to the ADC16H for conversion and can be further used by the end application. The sensor outputs an abnormal temperature detection signal to the reset control circuit and can be used to prevent the malfunction due to abnormal temperature. See section 55, Temperature Sensor (TSN).
High-Speed Analog Comparator (ACMPHS)	The High-Speed Analog Comparator (ACMPHS) can be used to compare an analog input voltage with a reference voltage and to provide a digital output based on the result of conversion. Both the analog input voltage and the reference voltage can be provided to the ACMPHS from internal sources (D/A converter output or internal reference voltage) and an external source. Such flexibility is useful in applications that require go/no-go comparisons to be performed between analog signals without necessarily requiring A/D conversion. See section 56, High-Speed Analog Comparator (ACMPHS).

Table 1.10 Human machine interfaces

Feature	Functional description
Capture Engine Unit (CEU)	The Capture Engine Unit (CEU) is a capture module that fetches image data externally input and transfers it to the memory. See section 60, Capture Engine Unit (CEU).

Table 1.11 Data processing

Feature	Functional description
Cyclic Redundancy Check (CRC) calculator	The Cyclic Redundancy Check (CRC) calculator generates CRC codes to detect errors in the data. The bit order of CRC calculation results can be switched for LSB-first or MSB-first communication. Additionally, various CRC-generation polynomials are available. The snoop function allows monitoring reads from and writes to specific addresses. This function is useful in applications that require CRC code to be generated automatically in certain events, such as monitoring writes to the serial transmit buffer and reads from the serial receive buffer. See section 48, Cyclic Redundancy Check (CRC).
Data Operation Circuit (DOC)	The Data Operation Circuit (DOC) compares, adds, and subtracts 32-bits data. When a selected condition applies, 32-bit data is compared and an interrupt can be generated. See section 57, Data Operation Circuit (DOC).

Table 1.12 Security

Feature	Functional description
Security function	• ARMv8-M TrustZone security • Privileged control • Device lifecycle management • Authentication Level (AL) • Key injection • Secure pin multiplexing • HUK zeroization • VBATT backup registers zeroization • Secure boot • Secure factory programming See section 51, Security Features.
Renesas Secure IP (RSIP-E50D)	• Symmetric cryptography: AES and ChaCha20-Poly1305 • Asymmetric cryptography: RSA and ECC • Message digest computation: HASH, HMAC • 128-bit true random number generation circuit • 256-bit Hardware Unique Key (HUK) • 128-bit unique ID • OEM boot loader version • Key data for the decryption on-the-fly (DOTF) • SPA/DPA Protections See section 52, Renesas Secure IP (RSIP-E50D).
Decryption on-the-fly (DOTF)	Decryption on-the-fly (DOTF) decrypts the encrypted content stored in the external memory in real-time. See section 45, Decryption On The Fly (DOTF).

1.2 Block Diagram

Figure 1.1 shows a block diagram of the MCU superset. Some individual devices within the group have a subset of the features.

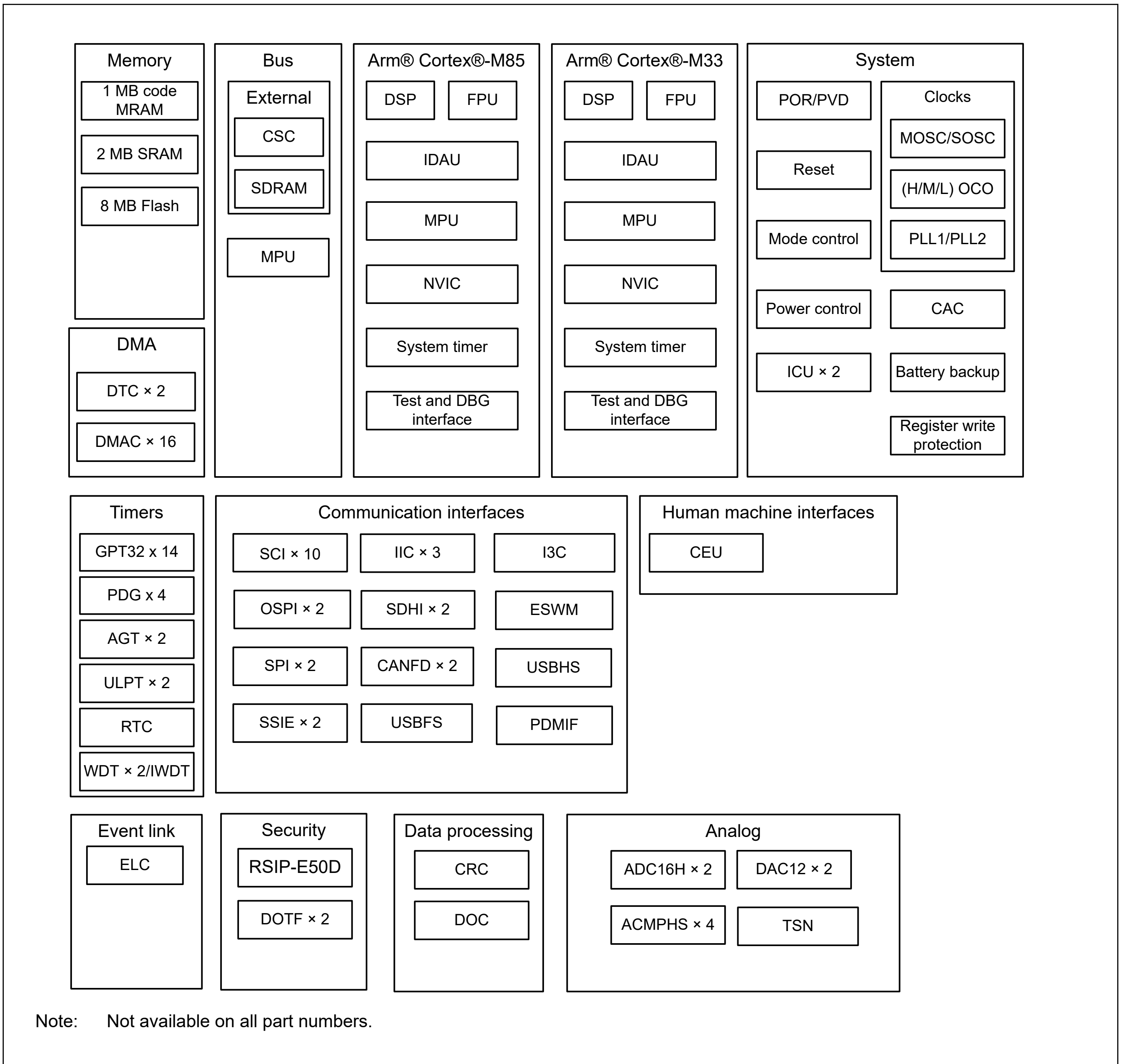


Figure 1.1 Block diagram

1.3 Part Numbering

Figure 1.2 shows the product part number information, including memory capacity and package type. Table 1.13 shows a list of products.

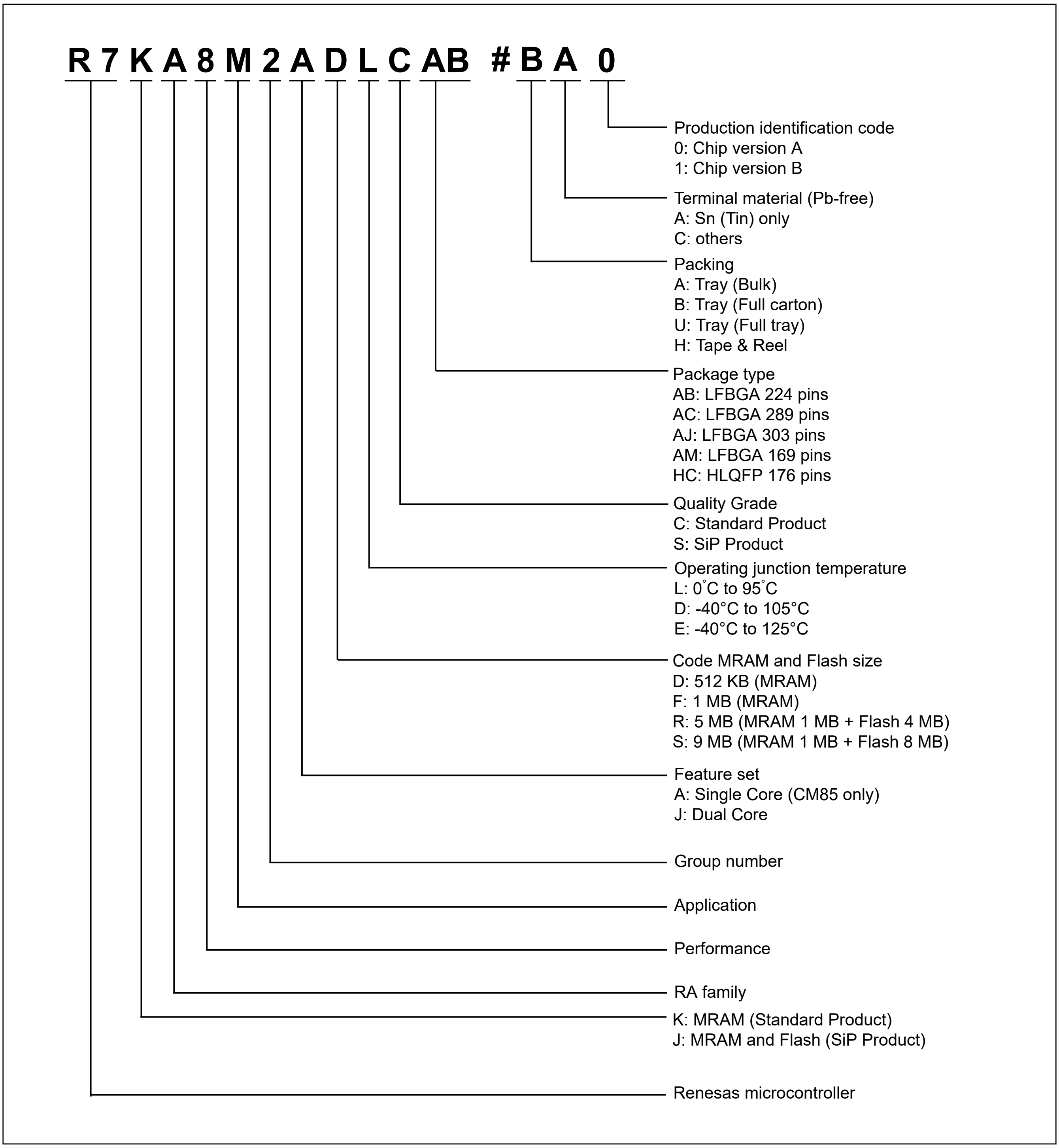


Figure 1.2 Part numbering scheme

Table 1.13 Product list

Product part number	Product group	CPU	Package code	Code MRAM	SRAM	Flash	Operating junction temperature
R7KA8M2ADLCAB	A	single	PLBG0224JA-A	512 KB	2 MB	—	0 to 95 °C
R7KA8M2ADLCAC			PLBG0289JA-A				
R7KA8M2ADDCAB	B		PLBG0224JA-A				-40 to 105 °C
R7KA8M2ADDCAC			PLBG0289JA-A				
R7KA8M2ADECAB	C		PLBG0224JA-A				-40 to 125 °C
R7KA8M2ADECAC			PLBG0289JA-A				
R7KA8M2ADECHC			PLQP0176KK-A				
R7KA8M2AFLCAM	A		PLBG0169xx-x	1 MB			0 to 95 °C
R7KA8M2AFLCAB			PLBG0224JA-A				
R7KA8M2AFLCAC			PLBG0289JA-A				
R7KA8M2AFDCAM	B		PLBG0169xx-x				-40 to 105 °C
R7KA8M2AFDCAB			PLBG0224JA-A				
R7KA8M2AFDCAC			PLBG0289JA-A				
R7KA8M2AFECAB	C		PLBG0224JA-A				-40 to 125 °C
R7KA8M2AFECAC			PLBG0289JA-A				
R7KA8M2AFECHC			PLQP0176KK-A				
R7KA8M2JFLCAM	A	dual	PLBG0169xx-x	1 MB			0 to 95 °C
R7KA8M2JFLCAB			PLBG0224JA-A				
R7KA8M2JFLCAC			PLBG0289JA-A				
R7KA8M2JFDCAM	B		PLBG0169xx-x				-40 to 105 °C
R7KA8M2JFDCAB			PLBG0224JA-A				
R7KA8M2JFDCAC			PLBG0289JA-A				
R7KA8M2JFECAB	C		PLBG0224JA-A				-40 to 125 °C
R7KA8M2JFECAC			PLBG0289JA-A				
R7KA8M2JFECHC			PLQP0176KK-A				
R7JA8M2JRLSAJ	A	dual	PLBG0303GA-A	1 MB		4 MB	0 to 95 °C
R7JA8M2JSLSAJ						8 MB	
R7JA8M2JRDSAJ	B					4 MB	-40 to 105 °C
R7JA8M2JSDSAJ						8 MB	

1.4 Function Comparison

Table 1.14 Function Comparison (1 of 2)

Parts number		R7KA8M2 AxxCAC	R7KA8M2 JxxCAC	R7KA8M2 AxxCAB	R7KA8M2 JxxCAB	R7KA8M2 AxxCAM	R7KA8M2 JxxCAM	R7KA8M2 AxxCHC	R7KA8M2 JxxCHC	R7JA8M2 JxxSAJ
Pin count		289		224		169		176		303
Package		BGA						HLQFP		BGA
I/O Port		208		149		109		130		195
Code MRAM		1 MB, 512 KB								1 MB
CPU0 TCM		256 KB								
CPU1 TCM		No	128 KB	No	128 KB	No	128 KB	No	128 KB	
CPU0 I/D Caches		32 KB								
CPU1 C/S Caches		No	32 KB	No	32 KB	No	32 KB	No	32 KB	
SRAM		1792 KB	1664 KB	1792 KB	1664 KB	1792 KB	1664 KB	1792 KB	1664 KB	
Flash		No								8 MB, 4 MB
DMA	DTC	1	2	1	2	1	2	1	2	
	DMAC	8	16	8	16	8	16	8	16	
BUS	External bus	32-bit bus		16-bit bus		No		16-bit bus		
	SDRAM	32-bit bus		16-bit bus		No		16-bit bus		
System	CPU0 clock	1 GHz (max.)						600 MHz (max.)		1 GHz (max.)
	CPU1 clock	No	250 MHz (max.)	No	250 MHz (max.)	No	250 MHz (max.)	No	200 MHz (max.)	250 MHz (max.)
	CPUs clock sources	MOSC, SOSC, HOCO, MOCO, PLL1P								
	CAC	Yes								
	WDT	1	2	1	2	1	2	1	2	
	IWDT	Yes								
	Backup register	128 B								
Communication	SCI	10		9		10		9		10
	IIC	3								
	I3C	Yes								
	SPI	2								
	CANFD	2								
	USBFS	Yes								
	USBHS	Yes				No		Yes		
	OSPI	2		1		1*3		1		2*2
	SSIE	2								
	SDHI/MMC	2				1*4		2		
	ESWM	MII, RMII, GMII, RGMII		MII, RMII, RGMII		RMII*5		MII, RMII		MII, RMII, GMII, RGMII
	PDMIF	Yes								

Table 1.14 Function Comparison (2 of 2)

Parts number		R7KA8M2 AxxCAC	R7KA8M2 JxxCAC	R7KA8M2 AxxCAB	R7KA8M2 JxxCAB	R7KA8M2 AxxCAM	R7KA8M2 JxxCAM	R7KA8M2 AxxCHC	R7KA8M2 JxxCHC	R7JA8M2 JxxSAJ
Timers	GPT32*1	14								
	PDG	4								
	AGT*1	2								
	ULPT*1	2								
	RTC	Yes								
Analog	ADC16H	Unit 0: 15, Unit 1: 15		Unit 0: 7, Unit 1: 5		Unit 0: 13, Unit 1: 11		Unit 0: 7, Unit 1: 5		Unit 0: 15, Unit 1: 15
	DAC12	2								
	ACMPHS	4								
	TSN	Yes								
HMI	CEU	Yes				Yes*6		Yes		
Data processing	CRC	Yes								
	DOC	Yes								
Event control	ELC	Yes								
Security		RSIP-E50D, Decryption on-the-fly, Secure Debug, OTP, TrustZone, and Lifecycle management								

Note: The product name differs depending on the supported memory size. See section 1.3. Part Numbering.
Note: The maximum frequency varies by product group. See section 62.3.1. Frequency for details.
Note 1. Available pins depend on the Pin count, about details see section 1.7. Pin Lists.
Note 2. OSPI1 is connected to the serial Flash in the SiP product.
Note 3. OM_0_ECSINT1, OM_0_RSTO1 and OM_0_WP1 pins are not supported.
Note 4. Only SD1CLK_B, SD1CMD_B, SD1DAT0_B to SD1DAT3_B and SD1WP_B pins are supported.
Note 5. Only port 1 is supported.
Note 6. 16-bit interface is not supported.

1.5 Pin Functions

Table 1.15 Pin functions (1 of 7)

Function	Signal	I/O	Description
Power supply	VCC_01 to VCC_10, VCC2_11 to VCC2_15	Input	Power supply pin. Connect it to the system power supply. Connect this pin to the same numbered VSS_01 to VSS_15 by a 0.1-μF capacitor. The capacitor should be placed close to the pin. In the SiP product, connect VCC2_11 to VCC2_15 to the 1.8V system power supply.
	VCC2_16 to VCC2_19	Input	Dedicated power supply pin for the SiP product. Connect it to the 1.8V system power supply. Connect this pin to the same numbered VSS_16 to VSS_19 by a 0.1-μF capacitor. The capacitor should be placed close to the pin.
	VCC_DCDC	Input	Switching regulator power supply pin.
	VLO	I/O	Switching regulator pin.
	VCL0 to VCL11	Input	Connect this pin to the same numbered VSS0 to VSS11 pin by the smoothing capacitor used to stabilize the internal power supply. Place the capacitor close to the pin.
	VBATT	Input	Battery Backup power pin
	VSS_01 to VSS_15, VSS0 to VSS11, VSS_DCDC	Input	Ground pin. Connect it to the system power supply (0 V).
	VSS_16 to VSS_19, VSS	Input	Dedicated ground pin for the SiP product. Connect it to the system power supply (0 V).
	Vpp	Input	Power supply pin for serial Flash programming operation. See the ISSI serial Flash IS25WX064 datasheet for the details. If not used, the Vpp pin can be floated. Note that the deep power down current varies depending on the power supply status of the Vpp pin.
Clock	XTAL	Output	Pins for a crystal resonator. An external clock signal can be input through the EXTAL pin.
	EXTAL	Input	
	XCIN	Input	Input/output pins for the sub-clock oscillator. Connect a crystal resonator between XCOUT and XCIN.
	XCOUT	Output	
	EXCIN	Input	External sub-clock input
	CLKOUT	Output	Clock output pin
Operating mode control	MD	Input	Pin for setting the operating mode. The signal level on this pin must not be changed during operation mode transition on release from the reset state.
System control	RES	Input	Reset signal input pin. The MCU enters the reset state when this signal goes low.
	PUP	Input	Connect to VCC2 through a resistor.
CAC	CACREF	Input	Measurement reference clock input pin
On-chip emulator	TMS	Input	On-chip emulator or boundary scan pins
	TDI	Input	
	TCK	Input	
	TDO	Output	
	TCLK	Output	Output clock for synchronization with the trace data
	TDATA0 to TDATA3	Output	Trace data output
	SWO	Output	Serial wire trace output pin
	SWDIO	I/O	Serial wire debug data input/output pin
	SWCLK	Input	Serial wire clock pin

Table 1.15 Pin functions (2 of 7)

Function	Signal	I/O	Description
Interrupt	NMI	Input	Non-maskable interrupt request pin
	IRQn	Input	Maskable interrupt request pins
	IRQn-DS	Input	Maskable interrupt request pins that can also be used in Deep Software Standby mode
External bus interface	EBCLK	Output	Outputs the external bus clock for external devices
	RD	Output	Strobe signal indicating that reading from the external bus interface space is in progress, active-low
	WR	Output	Strobe signal indicating that writing to the external bus interface space is in progress, in 1-write strobe mode, active-low
	WRn	Output	Strobe signals indicating that either group of data bus pins (D07 to D00, D15 to D08, D23 to D16 or D31 to D24) is valid in writing to the external bus interface space, in byte strobe mode, active-low
	BCn	Output	Strobe signals indicating that either group of data bus pins (D07 to D00, D15 to D08, D23 to D16 or D31 to D24) is valid in access to the external bus interface space, in 1-write strobe mode, active-low
	ALE	Output	Address latch signal when address/data multiplexed bus is selected
	WAIT	Input	Input pin for wait request signals in access to the external space, active-low
	CSn	Output	Select signals for CS areas, active-low
	A00 to A23	Output	Address bus
	D00 to D31	I/O	Data bus
	A00/D00 to A15/D15	I/O	Address/data multiplexed bus
SDRAM interface	SDCLK	Output	Outputs the SDRAM-dedicated clock
	CKE	Output	SDRAM clock enable signal
	SDCS	Output	SDRAM chip select signal, active low
	RAS	Output	SDRAM low address strobe signal, active low
	CAS	Output	SDRAM column address strobe signal, active low
	WE	Output	SDRAM write enable signal, active low
	DQMn	Output	SDRAM I/O data mask enable signal for DQ07 to DQ00, DQ15 to DQ08, DQ23 to DQ16 or DQ31 to DQ24
	A00 to A16	Output	Address bus
	DQ00 to DQ31	I/O	Data bus

Table 1.15 Pin functions (3 of 7)

Function	Signal	I/O	Description
GPT	GTETRGA, GTETRGB, GTETRGC, GTETRGD	Input	External trigger input pins
	GTIOCnA, GTIOCnB	I/O	Input capture, output compare, or PWM output pins
	GTADSM0, GTADSM1	Output	A/D conversion start request monitoring output pins
	GTCPPOn	Output	Toggle output synchronized with PWM period
	GTIU	Input	Hall sensor input pin U
	GTIV	Input	Hall sensor input pin V
	GTIW	Input	Hall sensor input pin W
	GTOUUP	Output	3-phase PWM output for BLDC motor control (positive U phase)
	GTOULO	Output	3-phase PWM output for BLDC motor control (negative U phase)
	GTOVUP	Output	3-phase PWM output for BLDC motor control (positive V phase)
	GTOVLO	Output	3-phase PWM output for BLDC motor control (negative V phase)
	GTOWUP	Output	3-phase PWM output for BLDC motor control (positive W phase)
	GTOWLO	Output	3-phase PWM output for BLDC motor control (negative W phase)
AGT	AGTEEn	Input	External event input enable signals
	AGTIOn	I/O	External event input and pulse output pins
	AGTOn	Output	Pulse output pins
	AGTOAn	Output	Output compare match A output pins
	AGTOBn	Output	Output compare match B output pins
ULPT	ULPTEEn	Input	External count control input
	ULPTEVIn	Input	External event input
	ULPTEEn-DS	Input	External count control input that can also be used in Deep Software Standby mode 1
	ULPTEVIn-DS	Input	External event input that can also be used in Deep Software Standby mode 1
	ULPTOn	Output	Pulse output
	ULPTOAn	Output	Output compare match A output
	ULPTOBn	Output	Output compare match B output
	ULPTOn-DS	Output	Pulse output that can also be used in Deep Software Standby mode 1
	ULPTOAn-DS	Output	Output compare match A output that can also be used in Deep Software Standby mode 1
	ULPTOBn-DS	Output	Output compare match B output that can also be used in Deep Software Standby mode 1
RTC	RTCOUT	Output	Output pin for 1-Hz or 64-Hz clock
	RTCICn	Input	Time capture event input pins

Table 1.15 Pin functions (4 of 7)

Function	Signal	I/O	Description
SCI	SCKn	I/O	Input/output pins for the clock (clock synchronous mode)
	RXDn	Input	Input pins for received data (asynchronous mode/clock synchronous mode)
	TXDn	Output	Output pins for transmitted data (asynchronous mode/clock synchronous mode)
	CTSn_RTSn	I/O	Input/output pins for controlling the start of transmission and reception (asynchronous mode/clock synchronous mode), active-low.
	CTSn	Input	Input for the start of transmission.
	DEn	Output	Driver enable signal for RS-485
	SCLn	I/O	Input/output pins for the IIC clock (simple IIC mode)
	SDAn	I/O	Input/output pins for the IIC data (simple IIC mode)
	SCKn	I/O	Input/output pins for the clock (simple SPI mode)
	MISOn	I/O	Input/output pins for slave transmission of data (simple SPI mode)
	MOSIn	I/O	Input/output pins for master transmission of data (simple SPI mode)
	SSn	Input	Chip-select input pins (simple SPI mode), active-low
IIC	SCLn	I/O	Input/output pins for the clock
	SDAn	I/O	Input/output pins for data
I3C	I3C_SCL0	I/O	Input/output pins for the clock
	I3C_SDA0	I/O	Input/output pins for data
SPI	RSPCKA, RSPCKB	I/O	Clock input/output pin
	MOSIA, MOSIB	I/O	Input or output pins for data output from the master
	MISOA, MISOB	I/O	Input or output pins for data output from the slave
	SSLA0, SSLB0	I/O	Input or output pin for slave selection
	SSLA1 to SSLA3, SSLB1 to SSLB3	Output	Output pins for slave selection
CANFD	CRXn	Input	Receive data
	CTXn	Output	Transmit data
USBFS	VCC_USB	Input	Power supply pin
	VSS_USB	Input	Ground pin
	USB_DP	I/O	D+ pin of the USB on-chip transceiver. Connect this pin to the D+ pin of the USB bus.
	USB_DM	I/O	D- pin of the USB on-chip transceiver. Connect this pin to the D- pin of the USB bus.
	USB_VBUS	Input	USB cable connection monitor pin. Connect this pin to VBUS of the USB bus. The VBUS pin status (connected or disconnected) can be detected when the USB module is operating as a function controller.
	USB_EXICEN	Output	Low-power control signal for external power supply (OTG) chip
	USB_VBUSEN	Output	VBUS (5 V) supply enable signal for external power supply chip
	USB_OVRCURA, USB_OVRCURB	Input	Connect the external overcurrent detection signals to these pins. Connect the VBUS comparator signals to these pins when the OTG power supply chip is connected.
	USB_OVRCURA-DS, USB_OVRCURB-DS	Input	Overcurrent pins for USBFS that can also be used in Deep Software Standby mode 1. Connect the external overcurrent detection signals to these pins. Connect the VBUS comparator signals to these pins when the OTG power supply chip is connected.
	USB_ID	Input	Connect the MicroAB connector ID input signal to this pin during operation in OTG mode

Table 1.15 Pin functions (5 of 7)

Function	Signal	I/O	Description
USBHS	VCC_USBHS	Input	Power supply pin
	VSS1_USBHS, VSS2_USBHS	Input	Ground pin
	AVCC_USBHS	Input	Analog power supply
	USBHS_RREF	I/O	Reference current source pin for the USBHS Must be connected to the VSS2_USBHS pin through a 2.2-kΩ (±1%) resistor.
	USBHS_DP	I/O	Input/output pin for the D+ data line of the USB bus
	USBHS_DM	I/O	Input/output pin for the D- data line of the USB bus
	USBHS_EXICEN	Output	Must be connected to the OTG power supply IC
	USBHS_ID	input	Must be connected to the OTG power supply IC
	USBHS_VBUSEN	Output	VBUS power supply enable pin for the USBHS
	USBHS_OVRCURA, USBHS_OVRCURB	Input	Overcurrent pin for the USBHS
	USBHS_OVRCURA-DS, USBHS_OVRCURB-DS	Input	Overcurrent pin for the USBHS that can also be used in Deep Software Standby mode 1.
	USBHS_VBUS	Input	USB cable connection monitor input pin
OSPI	OM_n_SCLK	Output	Clock output (OCTACLK divided by 2)
	OM_n_SCLKN	Output	Inverted clock output (OCTACLK divided by 2)
	OM_n_CSn	Output	Chip select signal for an OctaFlash device, active-low
	OM_n_DQS	I/O	Read data strobe/write data mask signal
	OM_n_SIOn	I/O	Data input/output
	OM_n_RESET	Output	Reset signal for both slave devices, active-low
	OM_n_ECSINT1	Input	Error Correction Status and Interrupt for slave1
	OM_n_RSTO1	Input	Slave reset status for slave1
	OM_n_WP1	Output	Write Protect for slave1, active-low
SSIE	SSIBCK0, SSIBCK1	I/O	SSIE serial bit clock pins
	SSILRCK0/SSIFS0, SSILRCK1/SSIFS1	I/O	LR clock/frame synchronization pins
	SSITXD0	Output	Serial data output pin
	SSIRXD0	Input	Serial data input pin
	SSIDATA1	I/O	Serial data input/output pins
	AUDIO_CLK	Input	External clock pin for audio (input oversampling clock)
SDHI/MMC	SDnCLK	Output	SD clock output pins
	SDnCMD	I/O	Command output pin and response input signal pins
	SDnDAT0 to SDnDAT7	I/O	SD and MMC data bus pins
	SDnCD	Input	SD card detection pins
	SDnWP	Input	SD write-protect signals

Table 1.15 Pin functions (6 of 7)

Function	Signal	I/O	Description
ESWM	ETn_GTX_CLK	Output	1000 Mb/s transmit clock
	ETn_TX_CLK	Input	100 Mb/s,10 Mb/s transmit clock
	ETn_RX_CLK	Input	Receive clock
	ETn_TX_EN	Output	Transmit enable
	ETn_TXD0 to ETn_TXD7	Output	Transmit data
	ETn_TX_ER	Output	Transmit coding error
	ETn_RX_DV	Input	Receive data valid
	ETn_RXD0 to ETn_RXD7	Input	Receive data
	ETn_RX_ER	Input	Receive error
	ETn_MDC	Output	Management data clock
	ETn_MDIO	I/O	Management data input/output
	RGMIIn_TXC	Output	Transmit clock
	RGMIIn_RXC	Input	Receive clock
	RGMIIn_TX_CTL	Output	Transmit control
	RGMIIn_TXD0 to RGMIIn_TXD3	Output	Transmit data
	RGMIIn_RX_CTL	Input	Receive control
	RGMIIn_RXD0 to RGMIIn_RXD3	Input	Receive data
	RMIIn_REF50CK	Input	Synchronous clock reference
	RMIIn_TX_EN	Output	Transmit enable
	RMIIn_TXD0 to RMIIn_TXD1	Output	Transmit data
	RMIIn_CRS_DV	Input	Carrier sense/Receive data valid
	RMIIn_RXD0 to RMIIn_RXD1	Input	Receive data
	RMIIn_RX_ER	Input	Receive error
	ETn_LINKSTA	Input	PHY Link Status
	ETn_INT	Input	PHY interrupt
	ETn_WOL	Output	Wake-on-LAN. This signal indicates that a Magic Packet was received.
	GPTP_CAPTUREn	Input	Media clock capture input
	GPTP_MATCHn	Output	Media clock recovery output
	GPTP_PPSn	Output	PPS signal
	GPTP_PTPOUT0 to GPTP_PTPOUT3	Output	PTP Pulse generator signal
	ET_TAS_STA0 to ET_TAS_STA3	Output	TAS status monitor
	ETHPHYCLK	Output	Clock output for PHY
PDMIF	PDMCLK0 to PDMCLK2	Output	Clock output pin
	PDMDAT0 to PDMDAT2	Input	Data input pin

Table 1.15 Pin functions (7 of 7)

Function	Signal	I/O	Description
Analog power supply	AVCC0	Input	Analog voltage supply pin. This is used as the analog power supply for the respective modules.
	AVSS0	Input	Analog ground pin. This is used as the analog ground for the respective modules. Supply this pin with the same voltage as the VSS pin.
	VREFH	Input	Analog reference voltage supply pin for the ADC16H (unit 1) and D/A Converter. Connect this pin to AVCC0 when not using the ADC16H (unit 1) and D/A Converter.
	VREFL	Input	Analog reference ground pin for the ADC16H and D/A Converter. Connect this pin to AVSS0 when not using the ADC16H (unit 1) and D/A Converter.
	VREFH0	Input	Analog reference voltage supply pin for the ADC16H (unit 0). Connect this pin to AVCC0 when not using the ADC16H (unit 0).
	VREFL0	Input	Analog reference ground pin for the ADC16H. Connect this pin to AVSS0 when not using the ADC16H (unit 0).
ADC16H	ANxxx	Input	Input pins for the analog signals to be processed by the A/D converter.
	ADTRGm	Input	Input pins for the external trigger signals that start the A/D conversion, active-low.
	ADSTm	Output	AD conversion start
	ADmFLAG1	Output	AD conversion end
	ADSYNC	Output	Synchronization signal between units
DAC12	DAn	Output	Output pins for the analog signals processed by the D/A converter.
ACMPHS	VCOUT	Output	Comparator output pin
	IVREFn	Input	Reference voltage input pins for comparator
	IVCMPn	Input	Analog voltage input pins for comparator
I/O ports	Pmn	I/O	General-purpose input/output pins (m: port number, n: pin number)
	P200	Input	General-purpose input pin
CEU	VIO_D15 to VIO_D0	Input	CEU data bus pins
	VIO_CLK	Input	CEU clock pins
	VIO_VD	Input	CEU vertical sync pins
	VIO_HD	Input	CEU horizontal sync pins
	VIO_FLD	Input	Field signal pins

1.6 Pin Assignments

The following figures show the pin assignments from the top view.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A	P609	P113	P115	P112	P302	P915	VLO	VLO	VSS_DCDC	VCC_DCDC	VCC_DCDC	P309	P906	P905	P907	P904	P207	A
B	P813	PA12	P114	PA11	P300	P303	VLO	VLO	VSS_DCDC	VCC_DCDC	VCC_DCDC	P311	P908	P909	P206	PD01	PD02	B
C	PA06	P613	PA13	P301	P200	P210/TMS/SWDIO	P208/TDI	P110	P308	P305	P307	P911	P312	PD04	PD03	PD05	PD06	C
D	PA04	P611	P610	PA14	RES	P211/TCK/SWCLK	P109	P108	P903	P304	P306	P912	PB04	PB07	PB05	PB03	PB01	D
E	PA15	P615	P614	P612	P914	P201/MD	P209/TDO	P111	P902	P310	P910	P913	PB02	PB06	PD07	PB00	P706	E
F	PA02	PA10	PA08	PA09	PC14	VCC_08	VSS_08	VSS3	VCL3	VSS_07	VCC_07	P700	P702	P406	P701	P707	P705	F
G	PA00	PA03	PA05	PA07	PC12	VCC_09	VSS_09	VSS4	VCL4	VSS_06	VCC_06	P405	P704	P703	VSS_03	VCC_05	VSS_05	G
H	P504	P503	P505	PA01	PC11	VCC_10	VSS_10	VSS7	VCL5	VSS5	VCC_04	VSS_04	P403	VCC_03	VCC_USBHS	USBHS_DP	USBHS_DM	H
J	P506	P507	P508	P509	PC13	VCC2_11	VSS_11	VCL7	VCL6	VSS6	VCL2	VSS2	P404	VSS_02	USBHS_RREF	VSS2_USBHS	VSS1_USBHS	J
K	PC15	P608	P510	PD00	PC07	VSS_12	VSS9	VCL9	VCL8	VSS8	VCL1	VSS1	P410	VCC_02	AVCC_USBHS	P213/XTAL	P212/EXTAL	K
L	PC03	PC02	PC04	PC09	PC05	VCC2_12	VSS_14	VSS_15	VSS10	VCL10	VCL0	VSS0	P414	P402	VCC_01	P214/XCOUT	P215/XCIN/EXCIN	L
M	PC00	P607	PC01	PC08	PC10	P104	VCC2_14	VCC2_15	P810	VSS11	VCL11	P412	P710	P411	P408	VBATT	VSS_01	M
N	P605	P604	P606	PC06	P107	P106	P105	P811	P013	P011	P807	P708	P712	P714	P711	P713	P401	N
P	P603	P602	P600	P601	P102	P801	P803	P812	P012	P010	P009	P805	P512	P413	P515	P709	P400	P
R	VCC2_13	P315	P900	P103	P101	P802	P804	P501	AVCC0	AVSS0	P005	P003	P513	P514	P415	P409	P407	R
T	P205	P203	P313	P901	P809	P800	P502	P014	VREFL	VREFL0	P004	P007	P001	P806	P715	P815/USB_DM	VSS_USB	T
U	P204	P202	P314	VSS_13	P808	P100	P500	P015	VREFH	VREFH0	P008	P006	P000	P002	P511	P814/USB_DP	VCC_USB	U
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	

Figure 1.3 Pin assignment for BGA 289-pin

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
A	NC	PA11	P114	P112	P300	VLO	VLO	VSS_DCDC	VCC_DCDC	VCC_DCDC	P309	P312	P908	P905	P206	A
B	P610	PA12	P115	P113	P302	VLO	VLO	VSS_DCDC	VCC_DCDC	VCC_DCDC	P311	P310	P906	P907	P909	B
C	P612	P611	PA13	P609	P301	RES	P210/TMS/SWDIO	P211/TCK/SWCLK	P304	P306	P305	P307	PB03	PB00	PB01	C
D	P615	P613	P614	PA14	P200	P208/TDI	P201/MD	P209/TDO	P902	P308	PB02	PB04	P705	P707	P706	D
E	PA15	PA08	P813	PA09	VCC_08	VSS_08	VSS5	VCL5	VSS_07	VCC_07	P405	P702	P704	P406	P701	E
F	PA06	PA10	PA05	PA07	VCC_09	VSS_09	VSS6	VCL6	VCL4	VSS4	P700	P703	VSS_03	VCC_05	VSS_05	F
G	PA04	PA02	PA01	PA03	VCC_10	VSS_10	VSS7	VCL7	VCL3	VSS3	P404	VCC_03	VCC_USBHS	USBHS_DP	USBHS_DM	G
H	PA00	P504	P503	P505	PC14	VSS_15	VSS8	VCL8	VCL2	VSS2	P403	VSS_02	USBHS_RREF	VSS2_USBHS	VSS1_USBHS	H
J	P506	P510	P507	P508	PC12	VCC2_15	VSS9	VCL9	VCL1	VSS1	P402	VCC_02	AVCC_USBHS	P213/XTAL	P212/EXTAL	J
K	PC15	P608	PD00	P509	VCC2_14	VSS_14	VSS10	VCL10	VCL0	VSS0	P410	P407	VCC_01	P214/XCOUT	P215/XCIN/EXCIN	K
L	PC13	P604	P603	P107	P106	P104	P105	VSS11	VCL11	P409	P414	P408	P415	VBATT	VSS_01	L
M	PC11	P602	P600	P601	P102	P801	P803	P009	P007	P708	P411	P710	P709	P711	P401	M
N	VCC2_12	P315	VSS_13	P103	P101	P802	P804	AVCC0	AVSS0	P005	P001	P712	P714	P713	P400	N
P	P205	P203	P313	VCC2_13	P809	P800	P015	VREFL	VREFL0	P006	P002	P003	P512	P815/USB_DM	VSS_USB	P
R	P204	P202	P314	VSS_12	P808	P100	P014	VREFH	VREFH0	P008	P004	P000	P511	P814/USB_DP	VCC_USB	R
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	

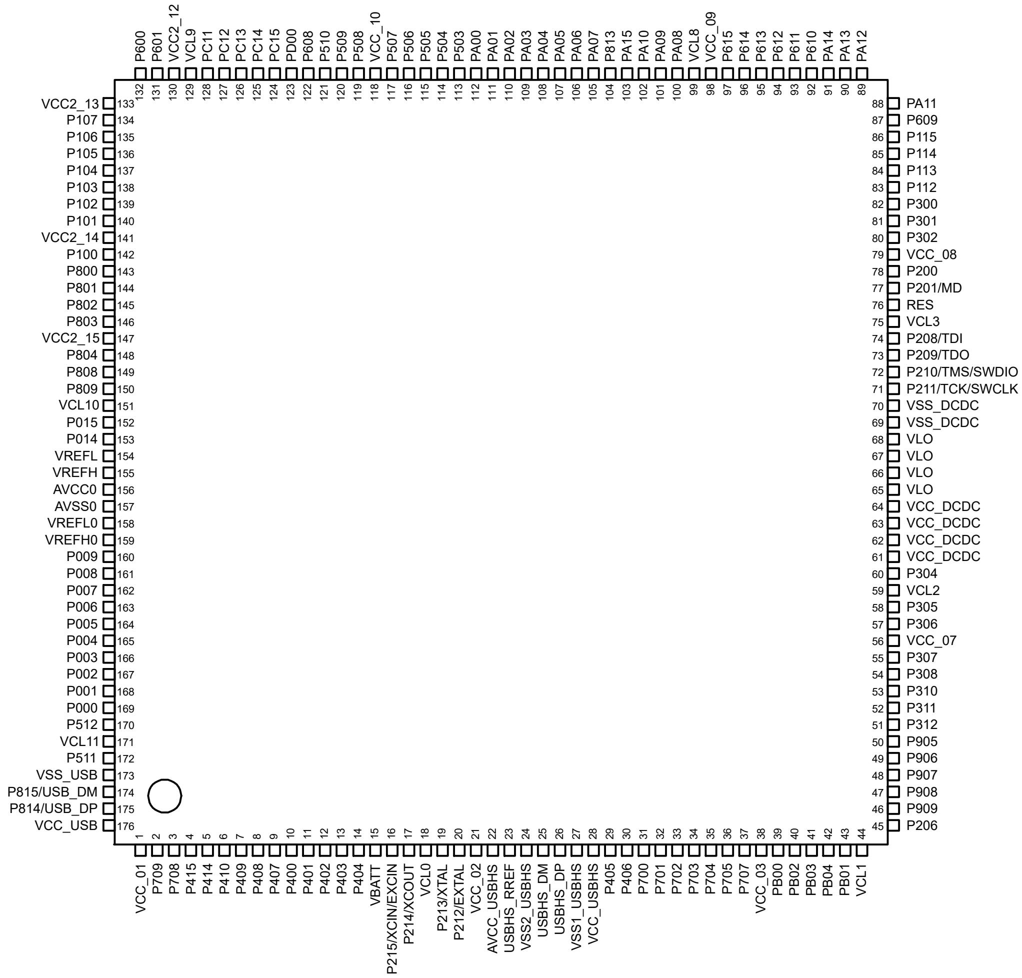
Figure 1.4 Pin assignment for BGA 224-pin

	1	2	3	4	5	6	7	8	9	10	11	12	13	
A	PA11	P114	P300	P302	VLO	VLO	VSS_DCDC	VCC_DCDC	VCC_DCDC	P306	P312	P905	P206	A
B	P115	P112	P113	P200	VLO	VLO	VSS_DCDC	VCC_DCDC	VCC_DCDC	P307	P906	P908	P909	B
C	P610	P609	PA12	P301	RES	P201/MD	P208/TDI	P209/TDO	P210/TMS/SWDIO	P211/TCK/SWCLK	P310	P907	P703	C
D	P613	P612	PA13	VCC_08	VSS_08	VSS5	VCL5	VSS_07	VCC_07	P311	P700	P702	P701	D
E	P614	PA08	PA14	VCC_09	VSS_09	VSS6	VCL6	VCL4	VSS4	P405	P404	VCC_02	VSS_02	E
F	PA10	PA09	P615	VCC_10	VSS_10	VSS7	VCL7	VCL3	VSS3	P402	P403	P213/XTAL	P212/EXTAL	F
G	PA02	PA04	PA05	VCC2_14	VSS_14	VSS8	VCL8	VCL2	VSS2	P408	VCC_01	P214/XCOUT	P215/XCIN/EXCIN	G
H	PA00	P509	PA03	VCC2_15	VSS_15	VSS9	VCL9	VCL1	VSS1	P409	P407	VBATT	VSS_01	H
J	P510	P603	PC13	PC14	VSS11	VCL11	VCL10	VCL0	VSS0	P410	P414	P415	P401	J
K	P602	P106	P104	PC11	P501	P015	VSS10	P009	P007	P709	P708	P514	P400	K
L	P107	P103	P101	P802	P500	AVCC0	AVSS0	P005	P003	P805	P512	P511	P515	L
M	P102	P800	P808	P803	P502	VREFL	VREFL0	P006	P004	P001	P513	VSS_USB	P815/USB_DM	M
N	P100	P801	P809	P804	P014	VREFH	VREFH0	P008	P002	P806	P000	VCC_USB	P814/USB_DP	N
	1	2	3	4	5	6	7	8	9	10	11	12	13	

Figure 1.5 Pin assignment for BGA 169-pin

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
A	VSS	P114	P609	P113	P301	P208/TDI	P210/TMS/SWDIO	VLO	VLO	VSS_DCDC	VCC_DCDC	VCC_DCDC	P309	P906	P905	P907	P207	VSS	A
B	P813	PA12	P115	PA11	P112	P209/TDO	P211/TCK/SWCLK	VLO	VLO	VSS_DCDC	VCC_DCDC	VCC_DCDC	P311	P908	P909	P904	PD01	PD02	B
C	PA06	P613	PA13	P300	P302	P200	RES	P110	P903	P308	P305	P307	P911	P206	PD04	PD03	PD05	PD06	C
D	PA04	P611	P610	PA14	P303	P915	P108	P111	P109	P310	P304	P306	P912	PB04	PB07	PB05	PB03	PB01	D
E	PA15	P615	P614	P612	P914	P201/MD				P902	P312	P910	P913	PB02	PB06	PD07	PB00	P706	E
F	PA02	PA10	PA08	PA09	PC14		VCC_08	VSS_08	VSS3	VCL3	VSS_07	VCC_07	P700	P702	P406	P701	P707	P705	F
G	PA00	PA03	PA05	PA07	PC12		VCC_09	VSS_09	VSS4	VCL4	VSS_06	VCC_06	P405	P704	P703	VSS_03	VCC_05	VSS_05	G
H	P504	P503	P505	PA01	PC11		VCC_10	VSS_10	VSS7	VCL5	VSS5	VCC_04	VSS_04		VCC_03	VCC_USBHS	USBHS_DP	USBHS_DM	H
J	P506	P507	P508	P509	PC13		VCC2_11	VSS_11	VCL7	VCL6	VSS6	VCL2	VSS2		VSS_02	USBHS_RREF	VSS2_USBHS	VSS1_USBHS	J
K	PC15	P608	P510	PD00	VSS	VSS	VSS_12	VSS9	VCL9	VCL8	VSS8	VCL1	VSS1		VCC_02	AVCC_USBHS	P213/XTAL	P212/EXTAL	K
L	PC10	VSS	PUP	VCC2_16	VSS_16		VCC2_12	VSS_14	VSS_15	VSS10	VCL10	VCL0	VSS0	P403	P404	VCC_01	P214/XCOUT	P215/XCIN/EXCIN	L
M	PC09	VSS	VSS	VCC2_17	VSS_17			VCC2_14	VCC2_15		VSS11	VCL11		P414	P402	P410	VBATT	VSS_01	M
N	PC08	VSS	VSS	VCC2_18	VSS_18		P105			P810				P710	P411	P408	P412	P401	N
P	VSS	VSS	VSS	VCC2_19	VSS_19	P104	P107	P106	P811	P013	P011	P807	P708	P712	P714	P711	P713	P400	P
R	P602	VSS	VSS	P600	P601	P102	P801	P803	P812	P012	P010	P009	P805	P512	P413	P515	P709	P407	R
T	Vpp	VSS	P315	P900	P103	P101	P802	P804	P501	AVCC0	AVSS0	P005	P003	P513	P514	P415	P409	VCC_USB	T
U	VCC2_13	P205	P203	P313	P901	P809	P800	P502	P014	VREFL	VREFL0	P004	P007	P001	P806	P715	P815/USB_DM	VSS_USB	U
V	VSS	P204	P202	P314	VSS_13	P808	P100	P500	P015	VREFH	VREFH0	P008	P006	P000	P002	P511	P814/USB_DP	VSS	V
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	

Figure 1.6 Pin assignment for BGA 303-pin



Note: VSS_01 to VSS_15 and VSS0 to VSS11 are assigned to Exposed PAD on the QFP package bottom. The PAD must be connected to the ground (0V).

Figure 1.7 Pin assignment for 176-pin

1.7 Pin Lists

Table 1.16 Pin list for the Standard product (1 of 9)

BGA289	BGA224	BGA169	HLQFP176	Power, System, Clock, Debug, CAC	I/O ports	ExBus/ SDRAM	Ex.Interrupt	SCI/IIC/I3C/SPI/CANFD/USBFS/ USBHS/OSPI/SSIE/SDHI/MMC/ ESWM(GMII, RGMII, MII, RMII)/ PDMIF	GPT/AGT/ULPT/RTC	ADC16H/ DAC12/ ACMPHS	CEU
A1	C4	C2	87	—	P609	D7/DQ7	IRQ29	TXD0_C/SDA0_C/MOSI0_C/MISOA_B/CTX1	GTIU/GTIOC5B/ULPTOA1-DS	AD1FLAG1	—
A2	B4	B3	84	—	P113	D4/DQ4	IRQ28	RXD0_A/SCL0_A/MISO0_A/SSLA1_B/SSILRCK0_B/SSIFS0_B/SD0DAT5_B	GTETRGB/GTIOC2A/ULPTOA0-DS	ADST1	—
A3	B3	B1	86	—	P115	D6/DQ6	IRQ31-DS	CTS0_A/MOSIA_B/SSITXD0_B/SD0DAT7_B	GTETRGD/GTIOC5A	AD0FLAG1	—
A4	A4	B2	83	—	P112	D3/DQ3	IRQ27	TXD0_A/SDA0_A/MOSI0_A/SSLA2_B/SSIBCK0_B/SD0DAT4_B	GTETRGA/GTIOC3B/ULPTOB0-DS	ADST0	—
A5	B5	A4	80	—	P302	D0/DQ0	IRQ5	RXD6_B/SCL6_B/MISO6_B/SD0DAT1_B	GTOUUP/GTIOC4A/ULPTO0-DS	—	—
A6	—	—	—	—	P915	—	IRQ8	CTS6_B	GTIOC5A	—	—
A7	A6	A5	68	VLO	—	—	—	—	—	—	—
A8	A7	A6	66	VLO	—	—	—	—	—	—	—
A9	A8	A7	70	VSS_DCDC	—	—	—	—	—	—	—
A10	A9	A8	64	VCC_DCDC	—	—	—	—	—	—	—
A11	A10	A9	62	VCC_DCDC	—	—	—	—	—	—	—
A12	A11	—	—	—	P309	—	IRQ25-DS	CTS9_B/ET1_GTX_CLK/RGMII1_TXC	GTCPPO8	VCOUT	VIO_D10
A13	B13	B11	49	—	P906	—	IRQ9	CTS6_A/USB_ID/SSILRCK1_A/SSIFS1_A/ET1_RXD0/RGMII1_RXD0/RMII1_RXD0/PDMDAT0	GTIOC13B/ULPTO1	AD0FLAG1	VIO_D5
A14	A14	A12	50	—	P905	—	IRQ8	RXD3_B/SCL3_B/MISO3_B/ET1_RX_CLK/RGMII1_RXC/RMII1_REF50CK/PDMDAT1	GTCPPO13	AD1FLAG1	VIO_D6
A15	B14	C12	48	—	P907	—	IRQ10	SCK6_A/DE6/USB_EXICEN/SSIBCK1_A/ET1_RXD1/RGMII1_RXD1/RMII1_RXD1/PDMCLK2	GTIOC13A/ULPTEE1	ADSYNC	VIO_D4
A16	—	—	—	—	P904	—	IRQ2	ET1_RXD4	GTIOC11B	—	—
A17	—	—	—	—	P207	—	IRQ25	ET1_RXD5	GTCPPO3	—	—
B1	E3	—	104	—	P813	SDCS	IRQ15	SCK7_A/DE7/PDMCLK2	GTETRGA/GTIOC7B	—	VIO_D13
B2	B2	C3	89	—	PA12	D9/DQ9	IRQ11	RXD9_C/SCL9_C/MISO9_C	GTIW/GTIOC6B	—	—
B3	A3	A2	85	—	P114	D5/DQ5	IRQ30-DS	CTS_RTS0_A/SS0_A/DE0/SSLA0_B/SSIRXD0_B/SD0DAT6_B	GTETRGC/GTIOC2B	ADSYNC	—
B4	A2	A1	88	—	PA11	D8/DQ8	IRQ10	SCK9_C/DE9	GTIV/GTIOC6A	—	—
B5	A5	A3	82	—	P300	D2/DQ2	IRQ4	SCK0_A/DE0/SSLA3_B/SD0DAT3_B	GTIOC3A/ULPTEVI0-DS	—	—
B6	—	—	—	—	P303	—	IRQ29-DS	SCK6_B/DE6	GTIOC7B	—	—
B7	B6	B5	67	VLO	—	—	—	—	—	—	—
B8	B7	B6	65	VLO	—	—	—	—	—	—	—
B9	B8	B7	69	VSS_DCDC	—	—	—	—	—	—	—
B10	B9	B8	63	VCC_DCDC	—	—	—	—	—	—	—
B11	B10	B9	61	VCC_DCDC	—	—	—	—	—	—	—
B12	B11	D10	52	—	P311	—	IRQ23-DS	SCK3_B/DE3/CRX0/ET1_TX_CLK	GTADSM1/GTCPPO6/AGTOB1	—	VIO_D8
B13	A13	B12	47	—	P908	—	IRQ11	TXD6_A/SDA6_A/MOSI6_A/CRX1/USB_OVRCURB/USBHS_ID/ET1_RXD2/RGMII1_RXD2/PDMCLK1	GTIOC12B/ULPTEVI1	ADST1	VIO_D3
B14	B15	B13	46	—	P909	—	IRQ21-DS	RXD6_A/SCL6_A/MISO6_A/CTX1/USB_OVRCURA/USBHS_EXICEN/ET1_RXD3/RGMII1_RXD3/PDMCLK0	GTIOC12A/ULPTOA1	ADST0	VIO_D2
B15	A15	A13	45	CLKOUT	P206	CS7	IRQ0-DS	USB_VBUSEN/SSIDATA1_A/SD0DAT7_C/ET1_RX_DV/RGMII1_RX_CTL/RMII1_CRS_DV	GTIU/GTCPPO0/ULPTOB1	—	VIO_D0
B16	—	—	—	—	PD01	—	IRQ22	SCK8_C/DE8/SD0DAT2_C/ET1_RXD6	GTCPPO2	—	—

Table 1.16 Pin list for the Standard product (2 of 9)

BGA289	BGA224	BGA169	HLQFP176	Power, System, Clock, Debug, CAC	I/O ports	ExBus/ SDRAM	Ex.Interrupt	SCI/IIC/I3C/SPI/CANFD/USBFS/ USBHS/OSPI/SSIE/SDHI/MMC/ ESWM(GMII, RGMII, MII, RMII)/ PDMIF	GPT/AGT/ULPT/RTC	ADC16H/ DAC12/ ACMPHS	CEU
B17	—	—	—	—	PD02	—	IRQ21	TXD8_C/SDA8_C/MOSI8_C/ SD0DAT1_C/ET1_RXD7	GTCPPO1	—	—
C1	F1	—	106	—	PA06	CS1/CKE	IRQ17	CTS2_C/SD0DAT1_A/PDMDAT1	GTETRGC/GTIOC7B	—	VIO_D11
C2	D2	D1	95	—	P613	D15/ DQ15	IRQ19	CTS0_C/USBHS_OVRCURB	GTETRGA/GTIOC9B/ AGTO1	—	—
C3	C3	D3	90	—	PA13	D10/ DQ10	IRQ12	CTS_RTS9_C/SS9_C/DE9	GTOVUP/GTIOC10A	—	—
C4	C5	C4	81	—	P301	D1/DQ1	IRQ6	TXD6_B/SDA6_B/MOSI6_B/ SD0DAT2_B	GTOULO/GTIOC4B/ AGTIO0/ULPTEE0-DS	—	—
C5	D5	B4	78	—	P200	—	NMI	—	—	—	—
C6	C7	C9	72	TMS/SWDIO	P210	—	IRQ24	CTS_RTS9_B/SS9_B/DE9	GTOULO/GTIOC0B	—	—
C7	D6	C7	74	TDI	P208	—	IRQ3	RXD9_B/SCL9_B/MISO9_B/CRX1	GTOVLO/GTIOC1B	VCOUT	—
C8	—	—	—	—	P110	—	IRQ20	SD0DAT4_C	GTIOC9B	—	—
C9	D10	—	54	TCLK	P308	—	IRQ26-DS	CTS3_B/SD0CLK_B/ET1_TX_ER/ ETHPHYCLK	GTIU/GTCPPO9/ ULPTOB1	—	VIO_D11
C10	C11	—	58	TDATA2	P305	—	IRQ8	SD0WP/ET1_TXD2/ RGMII1_TXD2	GTOVUP/GTCPPO12/ ULPTEE1	—	VIO_D14
C11	C12	B10	55	TDATA0	P307	—	IRQ27-DS	CTS_RTS6_A/SS6_A/DE6/ SD0CMD_B/ET1_TXD0/ RGMII1_TXD0/RMII1_TXD0	GTIV/GTCPPO10/ ULPTOA1	—	VIO_D12
C12	—	—	—	—	P911	—	IRQ6	ET1_TXD5	GTIOC3B	—	—
C13	A12	A11	51	—	P312	—	IRQ22-DS	CTS_RTS3_B/SS3_B/DE3/CTX0/ ET1_RX_ER/RMII1_RX_ER/ PDMDAT2	GTADSM0/GTCPPO5/ AGTOA1	—	VIO_D7
C14	—	—	—	—	PD04	—	IRQ20	CTS_RTS8_C/SS8_C/DE8/ USBHS_ID/SD0CMD_C/ ET0_RXD5	GTIOC3A	—	—
C15	—	—	—	—	PD03	—	IRQ21	RXD8_C/SCL8_C/MISO8_C/ USBHS_EXICEN/SD0DAT0_C/ ET0_RXD4	GTIOC3B	—	—
C16	—	—	—	—	PD05	—	IRQ19	CTS8_C/USBHS_OVRCURB/ SD0CLK_C/ET0_RXD6	GTIOC2B	—	—
C17	—	—	—	—	PD06	—	IRQ18	USBHS_OVRCURA/SD0WP/ ET0_RXD7	GTIOC2A	—	—
D1	G1	G2	108	—	PA04	A1/ DQM3	IRQ19	SCK2_C/DE2/SD0DAT3_A	GTIU/GTIOC4B	ADST0	VIO_D9
D2	C2	—	93	CACREF/CLKOUT	P611	D13/ DQ13	IRQ17	SCK0_C/DE0/MOSIA_B/ USBHS_VBUSEN	GTOULO/GTIOC4B	—	—
D3	B1	C1	92	—	P610	D12/ DQ12	IRQ16	RXD0_C/SCL0_C/MISO0_C/ RSPCKA_B/CRX1	GTOUUP/GTIOC4A/ ULPTOB1-DS	—	—
D4	D4	E3	91	—	PA14	D11/ DQ11	IRQ13	TXD9_C/SDA9_C/MOSI9_C	GTOVLO/GTIOC10B	—	—
D5	C6	C5	76	RES	—	—	—	—	—	—	—
D6	C8	C10	71	TCK/SWCLK	P211	—	IRQ23	SCK9_B/DE9	GTOUUP/GTIOC0A	—	—
D7	—	—	—	—	P109	—	IRQ23	SD0DAT5_C	GTIOC10A	—	—
D8	—	—	—	—	P108	—	IRQ24	SD0DAT6_C	GTIOC10B	—	—
D9	—	—	—	—	P903	—	IRQ1	—	GTIOC11A	—	—
D10	C9	—	60	TDATA3	P304	—	IRQ9	SD0DAT0_B/ET1_TXD3/ RGMII1_TXD3	GTOVLO/GTIOC7A/ ULPTO1	—	VIO_D15
D11	C10	A10	57	TDATA1	P306	—	IRQ28-DS	SD0CD/ET1_TXD1/ RGMII1_TXD1/RMII1_TXD1	GTIW/GTCPPO11/ ULPTEVI1	—	VIO_D13
D12	—	—	—	—	P912	—	IRQ5	ET1_TXD6	GTIOC3A	—	—
D13	D12	—	42	—	PB04	—	IRQ9	SCK5_C/DE5/ET0_TXD3/ RGMII0_TXD3	GTCPPO3	AD0FLAG1	VIO_CLK
D14	—	—	—	—	PB07	—	IRQ1	ET0_TXD5	GTIOC9B	—	—
D15	—	—	—	—	PB05	—	IRQ15	CTS5_C/ET0_TXD7	GTCPPO4	—	—
D16	C13	—	41	—	PB03	—	IRQ13	TXD5_C/SDA5_C/MOSI5_C/ ET0_TXD2/RGMII0_TXD2	GTCPPO1	ADSYNC	VIO_HD
D17	C15	—	43	—	PB01	ALE	IRQ12	CTS_RTS1_B/SS1_B/DE1/ ET0_TX_CLK	GTCPPO2	AD1FLAG1	VIO_FLD
E1	E1	—	103	—	PA15	EBCLK/ SDCLK	IRQ14	CTS9_C/PDMCLK1	GTIOC7A	—	VIO_D14

Table 1.16 Pin list for the Standard product (3 of 9)

BGA289	BGA224	BGA169	HLQFP176	Power, System, Clock, Debug, CAC	I/O ports	ExBus/ SDRAM	Ex.Interrupt	SCI/IIC/I3C/SPI/CANFD/USBFS/ USBHS/OSPI/SSIE/SDHI/MMC/ ESWM(GMII, RGMII, MII, RMII)/ PDMIF	GPT/AGT/ULPT/RTC	ADC16H/ DAC12/ ACMPHS	CEU
E2	D1	F3	97	—	P615	WR2/BC2/DQM2	IRQ7	TXD7_A/SDA7_A/MOSI7_A/ USBHS_EXICEN	GTETRGC/GTCPPO10	—	—
E3	D3	E1	96	—	P614	WR/WR0/DQM0	IRQ20	RXD7_A/SCL7_A/MISO7_A/ USBHS_ID	GTETRGB/GTCPPO9/ AGTO0	—	—
E4	C1	D2	94	—	P612	D14/ DQ14	IRQ18	CTS_RTS0_C/SS0_C/DE0/ SSLA0_B/USBHS_OVRCURA	GTIOC9A	—	—
E5	—	—	—	—	P914	—	IRQ9	CTS_RTS6_B/SS6_B/DE6	GTIOC5B	—	—
E6	D7	C6	77	MD	P201	—	IRQ4	—	—	—	—
E7	D8	C8	73	TDO/SWO/CLKOUT	P209	—	IRQ25	TXD9_B/SDA9_B/MOSI9_B/CTX1	GTOVUP/GTIOC1A	—	—
E8	—	—	—	—	P111	—	IRQ19	SD0DAT3_C	GTIOC9A	—	—
E9	D9	—	—	—	P902	ALE	IRQ0	AUDIO_CLK/ETHPHYCLK	GTCPPO13	—	VIO_D1
E10	B12	C11	53	—	P310	—	IRQ24-DS	TXD3_B/SDA3_B/MOSI3_B/ ET1_TX_EN/RGMII1_TX_CTL/ RMII1_TX_EN	GTCPPO7/AGTEE1	—	VIO_D9
E11	—	—	—	—	P910	—	IRQ7	ET1_TXD4	GTCPPO12	—	—
E12	—	—	—	CLKOUT	P913	—	IRQ3	ET1_TXD7	GTCPPO11	—	—
E13	D11	—	40	—	PB02	—	IRQ11	RXD5_C/SCL5_C/MISO5_C/ ET0_TXD1/RGMII0_TXD1/ RMII0_TXD1	GTCPPO0	ADST1	VIO_VD
E14	—	—	—	—	PB06	—	IRQ0	CTS_RTS5_C/SS5_C/DE5/ ET0_TXD6	GTIOC9A	—	—
E15	—	—	—	—	PD07	—	IRQ17	USBHS_VBUSEN/SD0CD/ ET0_TXD4	GTCPPO0	—	—
E16	C14	—	39	—	PB00	—	IRQ10	SCK1_B/DE1/USBHS_VBUSEN/ ET0_TXD0/RGMII0_TXD0/ RMII0_TXD0/PDMDAT2	GTCPPO4	ADST0	—
E17	D15	—	—	—	P706	—	IRQ7	RXD1_B/SCL1_B/MISO1_B/ USBHS_OVRCURB-DS/ ET0_GTX_CLK/RGMII0_TXC/ ETHPHYCLK/PDMDAT0	GTCPPO2/AGTIO0	—	VIO_D10
F1	G2	G1	110	—	PA02	A3	IRQ31	RXD2_C/SCL2_C/MISO2_C/ SD0DAT5_A	GTIW/GTCPPO9	ADSYNC	VIO_D7
F2	F2	F1	102	—	PA10	CS2/RAS	IRQ4	SCK5_B/DE5/PDMCLK0	GTCPPO13	—	VIO_D15
F3	E2	E2	100	—	PA08	CS0/WE	IRQ6	RXD5_B/SCL5_B/MISO5_B	GTETRGD/GTCPPO11	—	—
F4	E4	F2	101	—	PA09	CS3/CAS	IRQ5	TXD5_B/SDA5_B/MOSI5_B	GTCPPO12	—	—
F5	H5	J4	125	—	PC14	D16/ DQ16	IRQ0	TXD6_C/SDA6_C/MOSI6_C/ ET0_WOL	GTADSM1/GTCPPO9	—	—
F6	E5	D4	79	VCC_08	—	—	—	—	—	—	—
F7	E6	D5	—	VSS_08	—	—	—	—	—	—	—
F8	G10	F9	—	VSS3	—	—	—	—	—	—	—
F9	G9	F8	75	VCL3	—	—	—	—	—	—	—
F10	E9	D8	—	VSS_07	—	—	—	—	—	—	—
F11	E10	D9	56	VCC_07	—	—	—	—	—	—	—
F12	F11	D11	31	—	P700	—	IRQ16-DS	RXD2_B/SCL2_B/MISO2_B/ MISOA_C/SSIDATA1_B/SD1WP/ ET0_RXD2/RGMII0_RXD2	GTIOC5A	—	VIO_D4
F13	E12	D12	33	—	P702	—	IRQ18-DS	CTS2_B/RSPCKA_C/SSIBCK1_B/ SD1DAT5_B/ET0_RXD0/ RGMII0_RXD0/RMII0_RXD0	GTIOC6A/ULPTO0	—	VIO_D6
F14	E14	—	30	—	P406	—	IRQ31	TXD2_B/SDA2_B/MOSI2_B/ SSLA3_C/SSIRXD0_A/SD1CD/ ET0_RXD3/RGMII0_RXD3	GTIOC1B	—	VIO_D3
F15	E15	D13	32	—	P701	—	IRQ17-DS	CTS_RTS2_B/SS2_B/DE2/ MOSIA_C/SSILRCK1_B/ SSIFS1_B/SD1DAT4_B/ ET0_RXD1/RGMII0_RXD1/ RMII0_RXD1	GTIOC5B/ULPTO1	—	VIO_D5
F16	D14	—	37	—	P707	—	IRQ8	TXD1_B/SDA1_B/MOSI1_B/ USBHS_OVRCURA-DS/ ET0_TX_ER/ETHPHYCLK/ PDMDAT1	GTCPPO3	—	VIO_D11
F17	D13	—	36	—	P705	—	IRQ19	CTS1_B/SSLA2_C/CRX0/ ET0_TX_EN/RGMII0_TX_CTL/ RMII0_TX_EN/PDMCLK2	GTADSM1/GTCPPO1/ AGTIO0	—	VIO_D9

Table 1.16 Pin list for the Standard product (4 of 9)

BGA289	BGA224	BGA169	HLQFP176	Power, System, Clock, Debug, CAC	I/O ports	ExBus/ SDRAM	Ex.Interrupt	SCI/IIC/I3C/SPI/CANFD/USBFS/ USBHS/OSPI/SSIE/SDHI/MMC/ ESWM(GMII, RGMII, MII, RMII)/ PDMIF	GPT/AGT/ULPT/RTC	ADC16H/ DAC12/ ACMPHS	CEU
G1	H1	H1	112	—	PA00	A5	IRQ22	CTS_RTS5_B/SS5_B/DE5/ SD0DAT7_A	GTOVLO/GTCPPO7	AD1FLA G1	VIO_D5
G2	G4	H3	109	—	PA03	A2	IRQ20	TXD2_C/SDA2_C/MOSI2_C/ SD0DAT4_A	GTIV/GTCPPO10	ADST1	VIO_D8
G3	F3	G3	107	—	PA05	A0/BC0/ DQM1	IRQ18	CTS_RTS2_C/SS2_C/DE2/ SD0DAT2_A/PDMDAT2	GTETRGD/GTIOC4A	—	VIO_D10
G4	F4	—	105	—	PA07	RD	IRQ16	CTS7_A/SD0DAT0_A/PDMDAT0	GTETRGB/GTIOC7A	VCOUT	VIO_D12
G5	J5	—	127	—	PC12	D18/ DQ18	IRQ2	SCK6_C/DE6/ET0_MDIO	GTCPPO11	—	—
G6	F5	E4	98	VCC_09	—	—	—	—	—	—	—
G7	F6	E5	—	VSS_09	—	—	—	—	—	—	—
G8	F10	E9	—	VSS4	—	—	—	—	—	—	—
G9	F9	E8	—	VCL4	—	—	—	—	—	—	—
G10	—	—	—	VSS_06	—	—	—	—	—	—	—
G11	—	—	—	VCC_06	—	—	—	—	—	—	—
G12	E11	E10	29	—	P405	—	IRQ30	SCK2_B/DE2/SSITXD0_A/ SD1DAT3_B/ET0_RX_DV/ RGMII0_RX_CTL/RMII0_CRS_DV	GTIOC1A/AGTIO1	—	VIO_D2
G13	E13	—	35	—	P704	—	IRQ26	SSLA1_C/CTX0/SD1DAT7_B/ ET0_RX_ER/RMII0_RX_ER/ PDMCLK1	GTADSM0/GTCPPO0/ AGTO0	—	VIO_D8
G14	F12	C13	34	—	P703	—	IRQ19-DS	SSLA0_C/SD1DAT6_B/ ET0_RX_CLK/RGMII0_RXC/ RMII0_REF50CK/PDMCLK0	GTIOC6B/AGTO1	VCOUT	VIO_D7
G15	F13	—	—	VSS_03	—	—	—	—	—	—	—
G16	F14	—	—	VCC_05	—	—	—	—	—	—	—
G17	F15	—	—	VSS_05	—	—	—	—	—	—	—
H1	H2	—	114	—	P504	A7	IRQ7	SD0WP	GTOULO/GTCPPO1	—	VIO_D3
H2	H3	—	113	—	P503	A6	IRQ6	SD0CD	GTOUUP/GTCPPO6	—	VIO_D4
H3	H4	—	115	—	P505	A8	IRQ8	SD0CLK_A	GTOWUP/GTCPPO2	—	VIO_D2
H4	G3	—	111	—	PA01	A4	IRQ21	CTS5_B/SD0DAT6_A	GTOVUP/GTCPPO8	AD0FLA G1	VIO_D6
H5	M1	K4	128	—	PC11	D19/ DQ19	IRQ3	CTS_RTS6_C/SS6_C/DE6/ ET0_MDC	GTCPPO12	—	—
H6	G5	F4	118	VCC_10	—	—	—	—	—	—	—
H7	G6	F5	—	VSS_10	—	—	—	—	—	—	—
H8	G7	F6	—	VSS7	—	—	—	—	—	—	—
H9	E8	D7	—	VCL5	—	—	—	—	—	—	—
H10	E7	D6	—	VSS5	—	—	—	—	—	—	—
H11	—	—	—	VCC_04	—	—	—	—	—	—	—
H12	—	—	—	VSS_04	—	—	—	—	—	—	—
H13	H11	F11	13	—	P403	—	IRQ14-DS	CTS_RTS1_A/SS1_A/DE1/ SSIBCK0_A/SD1DAT1_B/ ET1_WOL	GTIOC3A/RTCIC1	AD0FLA G1	—
H14	G12	—	38	VCC_03	—	—	—	—	—	—	—
H15	G13	—	28	VCC_USBHS	—	—	—	—	—	—	—
H16	G14	—	26	USBHS_DP	—	—	—	—	—	—	—
H17	G15	—	25	USBHS_DM	—	—	—	—	—	—	—
J1	J1	—	116	—	P506	A9	IRQ9	SD0CMD_A	GTOWLO/GTCPPO3	—	VIO_D1
J2	J3	—	117	—	P507	A10	IRQ10	CTS_RTS7_A/SS7_A/DE7/ ET_TAS_STA0	GTADSM0/GTIOC0A	—	VIO_D0
J3	J4	—	119	—	P508	A11	IRQ1	CTS5_A/ET_TAS_STA1	GTADSM1/GTIOC0B	—	VIO_VD
J4	K4	H2	120	—	P509	A12	IRQ2	CTS_RTS5_A/SS5_A/DE5/ ET_TAS_STA2	GTIOC1A/ULPTEVI1	—	VIO_HD
J5	L1	J3	126	—	PC13	D17/ DQ17	IRQ1	RXD6_C/SCL6_C/MISO6_C/ ET0_INT	GTCPPO10	—	—
J6	—	—	—	VCC2_11	—	—	—	—	—	—	—
J7	—	—	—	VSS_11	—	—	—	—	—	—	—

Table 1.16 Pin list for the Standard product (5 of 9)

BGA289	BGA224	BGA169	HLQFP176	Power, System, Clock, Debug, CAC	I/O ports	ExBus/ SDRAM	Ex.Interrupt	SCI/IIC/I3C/SPI/CANFD/USBFS/ USBHS/OSPI/SSIE/SDHI/MMC/ ESWM(GMII, RGMII, MII, RMII)/ PDMIF	GPT/AGT/ULPT/RTC	ADC16H/ DAC12/ ACMPHS	CEU
J8	G8	F7	—	VCL7	—	—	—	—	—	—	—
J9	F8	E7	—	VCL6	—	—	—	—	—	—	—
J10	F7	E6	—	VSS6	—	—	—	—	—	—	—
J11	H9	G8	59	VCL2	—	—	—	—	—	—	—
J12	H10	G9	—	VSS2	—	—	—	—	—	—	—
J13	G11	E11	14	—	P404	—	IRQ15-DS	CTS1_A/SSILRCK0_A/SSIFS0_A/ SD1DAT2_B/ET0_WOL	GTIOC3B/RTCIC2	AD1FLAG1	—
J14	H12	E13	—	VSS_02	—	—	—	—	—	—	—
J15	H13	—	23	USBHS_RREF	—	—	—	—	—	—	—
J16	H14	—	24	VSS2_USBHS	—	—	—	—	—	—	—
J17	H15	—	27	VSS1_USBHS	—	—	—	—	—	—	—
K1	K1	—	124	—	PC15	A16	IRQ30	CTS6_C/CRX1	GTADSM0	—	—
K2	K2	—	122	CACREF	P608	A14	IRQ22	TXD5_A/SDA5_A/MOSI5_A	GTOWUP/GTCPPO4	—	VIO_FLD
K3	J2	J1	121	—	P510	A13	IRQ3	RXD5_A/SCL5_A/MISO5_A/ ET_TAS_STA3	GTIOC1B/ULPTEVI0	—	VIO_CLK
K4	K3	—	123	—	PD00	A15	IRQ23	SCK5_A/DE5/CTX1	GTOWLO/GTCPPO5	—	—
K5	—	—	—	—	PC07	D23/ DQ23	IRQ21	OM_1_RESET	GTCPPO0	—	—
K6	R4	—	—	VSS_12	—	—	—	—	—	—	—
K7	J7	H6	—	VSS9	—	—	—	—	—	—	—
K8	J8	H7	129	VCL9	—	—	—	—	—	—	—
K9	H8	G7	99	VCL8	—	—	—	—	—	—	—
K10	H7	G6	—	VSS8	—	—	—	—	—	—	—
K11	J9	H8	44	VCL1	—	—	—	—	—	—	—
K12	J10	H9	—	VSS1	—	—	—	—	—	—	—
K13	K11	J10	6	—	P410	A19	IRQ5	SCK3_A/DE3/SCL0_A*1/ USB_OVRCURB-DS/ USBHS_OVRCURB/ GPTP_MATCH0	GTOVLO/GTIOC9B/ AGTOB1	ADST0	—
K14	J12	E12	21	VCC_02	—	—	—	—	—	—	—
K15	J13	—	22	AVCC_USBHS	—	—	—	—	—	—	—
K16	J14	F12	19	XTAL	P213	—	IRQ2	TXD1_C/SDA1_C/MOSI1_C	GTETRGC/GTIOC0A/ ULPTEE0	ADTRG1	—
K17	J15	F13	20	EXTAL	P212	—	IRQ3	RXD1_C/SCL1_C/MISO1_C	GTETRGD/GTIOC0B/ AGTEE1	—	—
L1	—	—	—	—	PC03	D27/ DQ27	IRQ25	TXD7_C/SDA7_C/MOSI7_C/ OM_1_SIO4	GTCPPO4	—	—
L2	—	—	—	—	PC02	D28/ DQ28	IRQ26	SCK7_C/DE7/OM_1_SIO3	GTCPPO5	—	—
L3	—	—	—	—	PC04	D26/ DQ26	IRQ24	RXD7_C/SCL7_C/MISO7_C/ OM_1_SIO2	GTCPPO3	—	—
L4	—	—	—	—	PC09	D21/ DQ21	IRQ5	OM_1_RSTO1	—	—	—
L5	—	—	—	—	PC05	D25/ DQ25	IRQ23	OM_1_CS1	GTCPPO2	—	—
L6	N1	—	130	VCC2_12	—	—	—	—	—	—	—
L7	K6	G5	—	VSS_14	—	—	—	—	—	—	—
L8	H6	H5	—	VSS_15	—	—	—	—	—	—	—
L9	K7	K7	—	VSS10	—	—	—	—	—	—	—
L10	K8	J7	151	VCL10	—	—	—	—	—	—	—
L11	K9	J8	18	VCL0	—	—	—	—	—	—	—
L12	K10	J9	—	VSS0	—	—	—	—	—	—	—
L13	L11	J11	5	—	P414	A23	IRQ9	RXD4_B/SCL4_B/MISO4_B/ SSLB0_B/CRX1/ET1_MDIO	GTIOC0B	—	VIO_CLK
L14	J11	F10	12	CACREF	P402	—	IRQ4-DS	SCK1_A/DE1/CRX0/AUDIO_CLK/ SD1DAT0_B/ET0_LINKSTA	RTCIC0	—	—
L15	K13	G11	1	VCC_01	—	—	—	—	—	—	—

Table 1.16 Pin list for the Standard product (6 of 9)

BGA289	BGA224	BGA169	HLQFP176	Power, System, Clock, Debug, CAC	I/O ports	ExBus/ SDRAM	Ex.Interrupt	SCI/IIC/I3C/SPI/CANFD/USBFS/ USBHS/OSPI/SSIE/SDHI/MMC/ ESWM(GMII, RGMII, MII, RMII)/ PDMIF	GPT/AGT/ULPT/RTC	ADC16H/ DAC12/ ACMPHS	CEU
L16	K14	G12	17	XCOUT	P214	—	IRQ21	—	—	—	—
L17	K15	G13	16	XCIN/EXCIN	P215	—	IRQ20	—	—	—	—
M1	—	—	—	—	PC00	D30/ DQ30	IRQ28	CTS_RTS7_C/SS7_C/DE7/ OM_1_SIO5	GTCPPO7	—	—
M2	—	—	—	—	P607	D31/ DQ31	IRQ23	OM_1_DQS	—	—	—
M3	—	—	—	—	PC01	D29/ DQ29	IRQ27	CTS7_C/OM_1_SIO0	GTCPPO6	—	—
M4	—	—	—	—	PC08	D22/ DQ22	IRQ29	OM_1_CS0	GTCPPO8	—	—
M5	—	—	—	—	PC10	D20/ DQ20	IRQ4	OM_1_WP1	GTCPPO13	—	—
M6	L6	K3	137	—	P104	—	IRQ1	CTS9_A/SSLB1_A/OM_0_CS1/ GPTP_MATCH0	GTETRGB/GTIOC1B	AD0FLA G1	—
M7	K5	G4	141	VCC2_14	—	—	—	—	—	—	—
M8	J6	H4	147	VCC2_15	—	—	—	—	—	—	—
M9	—	—	—	—	P810	—	IRQ21	SCK7_B/DE7/SD1DAT2_A/ PDMCLK0	GTIOC10A/ULPTOA0	—	—
M10	L8	J5	—	VSS11	—	—	—	—	—	—	—
M11	L9	J6	171	VCL11	—	—	—	—	—	—	—
M12	—	—	—	—	P412	A21	IRQ20-DS	CTS3_A/USB_EXICEN/ USBHS_EXICEN/ GPTP_PTPOUT0	GTOULO/GTCPPO8/ AGTEE1	—	—
M13	M12	—	—	—	P710	CS5	IRQ17	CTS4_B/SSLB3_B/ET0_LINKSTA	GTIOC11B	—	VIO_D12
M14	M11	—	—	CACREF	P411	A20	IRQ4	CTS_RTS3_A/SS3_A/DE3/ USB_ID/USBHS_ID/ GPTP_PTPOUT1	GTOVUP/GTIOC9A/ AGTOA1	—	—
M15	L12	G10	8	—	P408	A17	IRQ7	RXD3_A/SCL3_A/MISO3_A/ SCL0_B*1/USB_VBUSEN/ USBHS_VBUS/GPTP_PTPOUT2	GTOWLO/GTIOC10A/ ULPTOB0	ADSYNC	—
M16	L14	H12	15	VBATT	—	—	—	—	—	—	—
M17	L15	H13	—	VSS_01	—	—	—	—	—	—	—
N1	—	—	—	—	P605	—	IRQ25	CTS0_B/OM_1_SIO1	GTIOC8A	—	—
N2	L2	—	—	—	P604	—	IRQ26	CTS_RTS0_B/SS0_B/DE0/ OM_1_SIO7	GTIOC8B	—	—
N3	—	—	—	—	P606	WR3/BC 3	IRQ24	OM_1_SIO6	—	—	—
N4	—	—	—	—	PC06	D24/ DQ24	IRQ22	OM_1_ECSINT1	GTCPPO1	—	—
N5	L4	L1	134	—	P107	—	IRQ31	CTS4_A/OM_0_CS0/ET1_INT	GTOWUP/GTIOC8A/ AGTOA0	ADST0	—
N6	L5	K2	135	—	P106	—	IRQ16	CTS8_B/SSLB3_A/ OM_0_RESET/ET1_LINKSTA	GTOWLO/GTIOC8B/ AGTOB0/ULPTEE1-DS	ADST1	—
N7	L7	—	136	—	P105	—	IRQ0	CTS_RTS8_B/SS8_B/DE8/ SSLB2_A/OM_0_ECSINT1/ GPTP_CAPTURE0	GTIOC1A/ULPTO1-DS	ADSYNC	—
N8	—	—	—	—	P811	—	IRQ22	CTS7_B/USB_ID/SD1DAT3_A/ PDMCLK1	GTIOC10B/ULPTOB0	—	—
N9	—	—	—	—	P013	—	IRQ14	—	—	AN013	—
N10	—	—	—	—	P011	—	IRQ16	—	—	AN011	—
N11	—	—	—	—	P807	—	IRQ11	—	GTIOC13A	—	—
N12	M10	K11	3	CACREF	P708	WR1/BC 1	IRQ11	SCK4_B/DE4/SDA2_A*1/ MOSIB_B/AUDIO_CLK/ET0_MDC	GTCPPO6	—	VIO_VD
N13	N12	—	—	—	P712	—	IRQ2	CTS1_C/SSLB1_B/ GPTP_CAPTURE1	GTIOC2B/AGTOB0	—	—
N14	N13	—	—	—	P714	—	IRQ13	TXD4_C/SDA4_C/MOSI4_C/ GPTP_PPS1	GTIOC12B	—	—
N15	M14	—	—	—	P711	—	IRQ3	CTS_RTS1_C/SS1_C/DE1/ SSLB2_B/GPTP_PPS0	GTIOC11A/AGTEE0	—	—
N16	N14	—	—	—	P713	—	IRQ14	CTS4_C/GPTP_MATCH1	GTIOC2A/AGTOA0	—	—
N17	M15	J13	11	—	P401	—	IRQ5-DS	RXD1_A/SCL1_A/MISO1_A/ I3C_SDA0/CTX0/SD1CMD_B	GTETRGA/GTIOC6B	—	VIO_D1

Table 1.16 Pin list for the Standard product (7 of 9)

BGA289	BGA224	BGA169	HLQFP176	Power, System, Clock, Debug, CAC	I/O ports	ExBus/ SDRAM	Ex.Interrupt	SCI/IIC/I3C/SPI/CANFD/USBFS/ USBHS/OSPI/SSIE/SDHI/MMC/ ESWM(GMII, RGMII, MII, RMII)/ PDMIF	GPT/AGT/ULPT/RTC	ADC16H/ DAC12/ ACMPHS	CEU
P1	L3	J2	—	—	P603	—	IRQ27	TXD0_B/SDA0_B/MOSI0_B/ OM_1_SCLK	GTIOC7A/ULPTO0	—	—
P2	M2	K1	—	—	P602	—	IRQ28	RXD0_B/SCL0_B/MISO0_B/ OM_1_SCLKN	GTIOC7B/ULPTEE0	—	—
P3	M3	—	132	CACREF	P600	—	IRQ30	OM_0_RSTO1/ET1_WOL	GTIOC6B/ULPTEVI1-DS	—	—
P4	M4	—	131	—	P601	—	IRQ29	SCK0_B/DE0/OM_0_WP1	GTIOC6A/ULPTEVI0/ RTCOUT	—	—
P5	M5	M1	139	—	P102	—	IRQ17	TXD9_A/SDA9_A/MOSI9_A/ RSPCKB_A/CRX0/OM_0_SIO4	GTOWLO/GTIOC2B/ AGTO0	ADTRG0	—
P6	M6	N2	144	—	P801	—	IRQ12	TXD2_A/SDA2_A/MOSI2_A/ OM_0_DQS/GPTP_PPS1	GTIV/GTIOC11B/ AGTOB0	—	—
P7	M7	M4	146	—	P803	—	IRQ19	SCK2_A/DE2/OM_0_SIO1	GTETRGC/GTIOC12B	—	—
P8	—	—	—	—	P812	—	IRQ23	CTS_RTS7_B/SS7_B/DE7/ USB_EXICEN/SD1DAT4_A/ PDMCLK2	GTIOC11A	AN022	—
P9	—	—	—	—	P012	—	IRQ15	—	—	AN012	—
P10	—	—	—	—	P010	—	IRQ14	—	—	AN010	—
P11	M8	K8	160	—	P009	—	IRQ13-DS	—	—	AN009/ IVREF1	—
P12	—	L10	—	—	P805	—	IRQ30	TXD8_A/SDA8_A/MOSI8_A/ ET1_MDIO	—	AN017/ IVCMP0	VIO_D15
P13	P13	L11	170	—	P512	—	IRQ14	CTS8_A/SCL1_A*1/CTX1/ ET1_INT	GTIOC0A	—	—
P14	—	—	—	—	P413	A22	IRQ18	ET_TAS_STA3	GTOUUP/GTCPPO7/ ULPTEE1	—	—
P15	—	L13	—	—	P515	—	IRQ12	CTS_RTS4_C/SS4_C/DE4/ SCL2_B*1/ET_TAS_STA0	GTIOC13A	—	—
P16	M13	K10	2	—	P709	CS4	IRQ10	CTS_RTS4_B/SS4_B/DE4/ SCL2_A*1/MISOB_B/ET0_MDIO	GTCPPO5	—	VIO_D13
P17	N15	K13	10	—	P400	—	IRQ0	TXD1_A/SDA1_A/MOSI1_A/ I3C_SCL0/AUDIO_CLK/ SD1CLK_B	GTIOC6A/AGTIO1	ADTRG1	VIO_D0
R1	P4	—	133	VCC2_13	—	—	—	—	—	—	—
R2	N2	—	—	—	P315	—	IRQ29	SCK3_C/DE3/SSLA3_A	—	—	—
R3	—	—	—	—	P900	—	IRQ30	CTS3_C	GTADSM0	—	—
R4	N4	L2	138	—	P103	—	IRQ16	CTS_RTS9_A/SS9_A/DE9/ SSLB0_A/CTX0/OM_0_SIO2/ GPTP_PPS0	GTOWUP/GTIOC2A	AD1FLAG1	—
R5	N5	L3	140	—	P101	—	IRQ1	RXD9_A/SCL9_A/MISO9_A/ MOSIB_A/OM_0_SIO3/ GPTP_CAPTURE1	GTETRGB/GTIOC8A/ AGTEE0	—	—
R6	N6	L4	145	—	P802	—	IRQ18	RXD2_A/SCL2_A/MISO2_A/ OM_0_SIO6	GTIW/GTIOC12A	—	—
R7	N7	N4	148	—	P804	—	IRQ14	CTS_RTS2_A/SS2_A/DE2/ OM_0_SIO7	GTETRGD/GTIOC13A	—	—
R8	—	K5	—	—	P501	—	IRQ25	TXD8_B/SDA8_B/MOSI8_B/ USB_OVRCURA/SD1DAT6_A/ PDMDAT1	GTIOC12A	AN020	—
R9	N8	L6	156	AVCC0	—	—	—	—	—	—	—
R10	N9	L7	157	AVSS0	—	—	—	—	—	—	—
R11	N10	L8	164	—	P005	—	IRQ10-DS	—	—	AN005/ IVCMP3	—
R12	P12	L9	166	—	P003	—	IRQ29	—	—	AN003/ IVCMP3	—
R13	—	M11	—	—	P513	—	IRQ31	SCK8_A/DE8/ET0_INT	GTIOC13B	AN016/ IVCMP0	VIO_FLD
R14	—	K12	—	—	P514	—	IRQ13	SCK4_C/DE4/SDA2_B*1/ ET_TAS_STA1	GTIOC13B	—	—
R15	L13	J12	4	—	P415	WAIT	IRQ8	TXD4_B/SDA4_B/MOSI4_B/ RSPCKB_B/CTX1/ET1_MDC	GTIOC0A	—	VIO_HD
R16	L10	H10	7	—	P409	A18	IRQ6	TXD3_A/SDA3_A/MOSI3_A/ SDA0_A*1/USB_OVRCURA-DS/USBHS_OVRCURA/ GPTP_CAPTURE0	GTOWUP/ULPTOA0	ADST1	—

Table 1.16 Pin list for the Standard product (8 of 9)

BGA289	BGA224	BGA169	HLQFP176	Power, System, Clock, Debug, CAC	I/O ports	ExBus/ SDRAM	Ex.Interrupt	SCI/IIC/I3C/SPI/CANFD/USBFS/ USBHS/OSPI/SSIE/SDHI/MMC/ ESWM(GMII, RGMII, MII, RMII)/ PDMIF	GPT/AGT/ULPT/RTC	ADC16H/ DAC12/ ACMPHS	CEU
R17	K12	H11	9	—	P407	CS6	IRQ22	SCK1_C/DE1/SDA0_B*1/ USB_VBUS/USBHS_VBUSEN/ GPTP_PTPOUT3	GTIOC10B/AGTIO0/ RTCOUT	ADTRG0	—
T1	P1	—	—	CLKOUT	P205	—	IRQ1-DS	TXD4_A/SDA4_A/MOSI4_A/ SCL1_B*1/SSLA1_A/ USB_OVRCURA/SD1CD	GTIV/GTIOC4A/ AGTO1	—	—
T2	P2	—	—	—	P203	—	IRQ2-DS	RXD4_A/SCL4_A/MISO4_A/ RSPCKA_A/CTX0/USB_VBUSEN/ SD1CLK_A	GTIOC5A/ULPTOA1	—	—
T3	P3	—	—	—	P313	—	IRQ27	TXD3_C/SDA3_C/MOSI3_C/ MISOA_A/USB_ID/SD1DAT0_A	—	—	—
T4	—	—	—	—	P901	—	IRQ31	CTS_RTS3_C/SS3_C/DE3	GTADSM1/AGTIO1	—	—
T5	P5	N3	150	—	P809	—	IRQ20	TXD7_B/SDA7_B/MOSI7_B/ OM_0_SCLKN	—	—	—
T6	P6	M2	143	—	P800	—	IRQ11	CTS2_A/OM_0_SIO5	GTIU/GTIOC11A/ AGTOA0	—	—
T7	—	M5	—	—	P502	—	IRQ26	SCK8_B/DE8/USB_OVRCURB/ SD1DAT7_A/PDMDAT2	GTIOC12B	AN019	—
T8	R7	N5	153	—	P014	—	IRQ27	—	—	AN014/D A0/ IVCMP0	—
T9	P8	M6	154	VREFL	—	—	—	—	—	—	—
T10	P9	M7	158	VREFL0	—	—	—	—	—	—	—
T11	R11	M9	165	—	P004	—	IRQ9-DS	—	—	AN004/ IVCMP2	—
T12	M9	K9	162	—	P007	—	IRQ28	—	—	AN007/ IVCMP3	—
T13	N11	M10	168	—	P001	—	IRQ7-DS	—	—	AN001/ IVCMP3	—
T14	—	N10	—	—	P806	—	IRQ0	RXD8_A/SCL8_A/MISO8_A/ ET1_MDC	—	AN018	VIO_D14
T15	—	—	—	—	P715	—	IRQ12	RXD4_C/SCL4_C/MISO4_C/ ET_TAS_STA2	GTIOC12A	—	—
T16	P14	M13	174	—	P815	—	IRQ15	CTX0/USB_DM	GTIOC8A	—	—
T17	P15	M12	173	VSS_USB	—	—	—	—	—	—	—
U1	R1	—	—	CACREF	P204	—	IRQ26	SCK4_A/DE4/SDA1_B*1/ SSLA0_A/USB_OVRCURB/ SD1WP	GTIW/GTIOC4B/ AGTIO1	—	—
U2	R2	—	—	—	P202	—	IRQ3-DS	CTS_RTS4_A/SS4_A/DE4/ MOSIA_A/CRX0/USB_EXICEN/ SD1CMD_A	GTIOC5B/ULPTOB1	—	—
U3	R3	—	—	—	P314	—	IRQ28	RXD3_C/SCL3_C/MISO3_C/ SSLA2_A/SD1DAT1_A	—	ADTRG0	—
U4	N3	—	—	VSS_13	—	—	—	—	—	—	—
U5	R5	M3	149	—	P808	—	IRQ15	RXD7_B/SCL7_B/MISO7_B/ OM_0_SCLK	GTIOC13B	—	—
U6	R6	N1	142	—	P100	—	IRQ2	SCK9_A/DE9/MISOB_A/ OM_0_SIO0/GPTP_MATCH1	GTETRGA/GTIOC8B/ AGTIO0	—	—
U7	—	L5	—	CACREF	P500	—	IRQ24	RXD8_B/SCL8_B/MISO8_B/ USB_VBUSEN/SD1DAT5_A/ PDMDAT0	GTIOC11B	AN021	—
U8	P7	K6	152	—	P015	—	IRQ13	—	—	AN015/D A1/ IVCMP0	—
U9	R8	N6	155	VREFH	—	—	—	—	—	—	—
U10	R9	N7	159	VREFH0	—	—	—	—	—	—	—
U11	R10	N8	161	—	P008	—	IRQ12-DS	—	—	AN008/ IVREF0	—
U12	P10	M8	163	—	P006	—	IRQ11-DS	—	—	AN006/ IVCMP2	—
U13	R12	N11	169	—	P000	—	IRQ6-DS	—	—	AN000/ IVCMP2	—
U14	P11	N9	167	—	P002	—	IRQ8-DS	—	—	AN002/ IVCMP2	—
U15	R13	L12	172	—	P511	—	IRQ15	CTS_RTS8_A/SS8_A/DE8/ SDA1_A*1/CRX1/ET1_LINKSTA	GTIOC0B	—	—

Table 1.16 Pin list for the Standard product (9 of 9)

BGA289	BGA224	BGA169	HLQFP176	Power, System, Clock, Debug, CAC	I/O ports	ExBus/ SDRAM	Ex.Interrupt	SCI/IIC/I3C/SPI/CANFD/USBFS/ USBHS/OSPI/SSIE/SDHI/MMC/ ESWM(GMII, RGMII, MII, RMII)/ PDMIF	GPT/AGT/ULPT/RTC	ADC16H/ DAC12/ ACMPHS	CEU
U16	R14	N13	175	—	P814	—	IRQ16	CRX0/USB_DP	GTIOC8B	—	—
U17	R15	N12	176	VCC_USB	—	—	—	—	—	—	—

Note: Several pin names have the added suffix of _A, _B, and _C. These suffixes have special conditions for electrical characteristics. See section 62, Electrical Characteristics for detail.

Note 1. There are two types for IIC function, one is a simple IIC by SCI and another is a dedicated IIC. This terminal is for a dedicated IIC.

Table 1.17 Pin list for the SiP product (1 of 8)

BGA303	Power, System, Clock, Debug, CAC	I/O ports	ExBus/ SDRAM	Ex.Interrupt	SCI/IIC/I3C/SPI/CANFD/USBFS/USBHS/ OSPI/SSIE/SDHI/MMC/ESWM(GMII, RGMII, MII, RMII)/PDMIF	GPT/AGT/ULPT/RTC	ADC16H/ DAC12/ ACMPHS	CEU
A1	VSS	—	—	—	—	—	—	—
A2	—	P114	D5/DQ5	IRQ30-DS	CTS_RTS0_A/SS0_A/DE0/SSLA0_B/ SSIRXD0_B/SD0DAT6_B	GTETRGC/GTIOC2B	ADSYNC	—
A3	—	P609	D7/DQ7	IRQ29	TXD0_C/SDA0_C/MOSI0_C/MISOA_B/ CTX1	GTIU/GTIOC5B/ULPTOA1-DS	AD1FLAG1	—
A4	—	P113	D4/DQ4	IRQ28	RXD0_A/SCL0_A/MISO0_A/SSLA1_B/ SSILRCK0_B/SSIFS0_B/SD0DAT5_B	GTETRGB/GTIOC2A/ULPTOA0-DS	ADST1	—
A5	—	P301	D1/DQ1	IRQ6	TXD6_B/SDA6_B/MOSI6_B/SD0DAT2_B	GTOULO/GTIOC4B/AGTIO0/ ULPTEE0-DS	—	—
A6	TDI	P208	—	IRQ3	RXD9_B/SCL9_B/MISO9_B/CRX1	GTOVLO/GTIOC1B	VCOUT	—
A7	TMS/SWDIO	P210	—	IRQ24	CTS_RTS9_B/SS9_B/DE9	GTOULO/GTIOC0B	—	—
A8	VLO	—	—	—	—	—	—	—
A9	VLO	—	—	—	—	—	—	—
A10	VSS_DCDC	—	—	—	—	—	—	—
A11	VCC_DCDC	—	—	—	—	—	—	—
A12	VCC_DCDC	—	—	—	—	—	—	—
A13	—	P309	—	IRQ25-DS	CTS9_B/ET1_GTX_CLK/RGMII1_TXC	GTCPPO8	VCOUT	VIO_D10
A14	—	P906	—	IRQ9	CTS6_A/USB_ID/SSILRCK1_A/ SSIFS1_A/ET1_RXD0/RGMII1_RXD0/ RMII1_RXD0/PDMDAT0	GTIOC13B/ULPTO1	AD0FLAG1	VIO_D5
A15	—	P905	—	IRQ8	RXD3_B/SCL3_B/MISO3_B/ ET1_RX_CLK/RGMII1_RXC/ RMII1_REF50CK/PDMDAT1	GTCPPO13	AD1FLAG1	VIO_D6
A16	—	P907	—	IRQ10	SCK6_A/DE6/USB_EXICEN/SSIBCK1_A/ ET1_RXD1/RGMII1_RXD1/RMII1_RXD1/ PDMCLK2	GTIOC13A/ULPTEE1	ADSYNC	VIO_D4
A17	—	P207	—	IRQ25	ET1_RXD5	GTCPPO3	—	—
A18	VSS	—	—	—	—	—	—	—
B1	—	P813	SDCS	IRQ15	SCK7_A/DE7/PDMCLK2	GTETRGA/GTIOC7B	—	VIO_D13
B2	—	PA12	D9/DQ9	IRQ11	RXD9_C/SCL9_C/MISO9_C	GTIW/GTIOC6B	—	—
B3	—	P115	D6/DQ6	IRQ31-DS	CTS0_A/MOSIA_B/SSITXD0_B/ SD0DAT7_B	GTETRGD/GTIOC5A	AD0FLAG1	—
B4	—	PA11	D8/DQ8	IRQ10	SCK9_C/DE9	GTIV/GTIOC6A	—	—
B5	—	P112	D3/DQ3	IRQ27	TXD0_A/SDA0_A/MOSI0_A/SSLA2_B/ SSIBCK0_B/SD0DAT4_B	GTETRGA/GTIOC3B/ULPTOB0-DS	ADST0	—
B6	TDO/SWO/CLKOUT	P209	—	IRQ25	TXD9_B/SDA9_B/MOSI9_B/CTX1	GTOVUP/GTIOC1A	—	—
B7	TCK/SWCLK	P211	—	IRQ23	SCK9_B/DE9	GTOUUP/GTIOC0A	—	—
B8	VLO	—	—	—	—	—	—	—
B9	VLO	—	—	—	—	—	—	—
B10	VSS_DCDC	—	—	—	—	—	—	—
B11	VCC_DCDC	—	—	—	—	—	—	—
B12	VCC_DCDC	—	—	—	—	—	—	—
B13	—	P311	—	IRQ23-DS	SCK3_B/DE3/CRX0/ET1_TX_CLK	GTADSM1/GTCPPO6/AGTOB1	—	VIO_D8
B14	—	P908	—	IRQ11	TXD6_A/SDA6_A/MOSI6_A/CRX1/ USB_OVRCURB/USBHS_ID/ET1_RXD2/ RGMII1_RXD2/PDMCLK1	GTIOC12B/ULPTEVI1	ADST1	VIO_D3
B15	—	P909	—	IRQ21-DS	RXD6_A/SCL6_A/MISO6_A/CTX1/ USB_OVRCURA/USBHS_EXICEN/ ET1_RXD3/RGMII1_RXD3/PDMCLK0	GTIOC12A/ULPTOA1	ADST0	VIO_D2
B16	—	P904	—	IRQ2	ET1_RXD4	GTIOC11B	—	—

Table 1.17 Pin list for the SiP product (2 of 8)

BGA303	Power, System, Clock, Debug, CAC	I/O ports	ExBus/ SDRAM	Ex.Interrupt	SCI/IIC/I3C/SPI/CANFD/USBFS/USBHS/ OSPI/SSIE/SDHI/MMC/ESWM(GMII, RGMII, MII, RMII)/PDMIF	GPT/AGT/ULPT/RTC	ADC16H/ DAC12/ ACMPHS	CEU
B17	—	PD01	—	IRQ22	SCK8_C/DE8/SD0DAT2_C/ET1_RXD6	GTCPPO2	—	—
B18	—	PD02	—	IRQ21	TXD8_C/SDA8_C/MOSI8_C/SD0DAT1_C/ ET1_RXD7	GTCPPO1	—	—
C1	—	PA06	CS1/CKE	IRQ17	CTS2_C/SD0DAT1_A/PDMDAT1	GTETRGC/GTIOC7B	—	VIO_D11
C2	—	P613	D15/DQ15	IRQ19	CTS0_C/USBHS_OVRCURB	GTETRGA/GTIOC9B/AGTO1	—	—
C3	—	PA13	D10/DQ10	IRQ12	CTS_RTS9_C/SS9_C/DE9	GTOVUP/GTIOC10A	—	—
C4	—	P300	D2/DQ2	IRQ4	SCK0_A/DE0/SSLA3_B/SD0DAT3_B	GTIOC3A/ULPTEVI0-DS	—	—
C5	—	P302	D0/DQ0	IRQ5	RXD6_B/SCL6_B/MISO6_B/SD0DAT1_B	GTOUUP/GTIOC4A/ULPTO0-DS	—	—
C6	—	P200	—	NMI	—	—	—	—
C7	RES	—	—	—	—	—	—	—
C8	—	P110	—	IRQ20	SD0DAT4_C	GTIOC9B	—	—
C9	—	P903	—	IRQ1	—	GTIOC11A	—	—
C10	TCLK	P308	—	IRQ26-DS	CTS3_B/SD0CLK_B/ET1_TX_ER/ ETHPHYCLK	GTIU/GTCPPO9/ULPTOB1	—	VIO_D11
C11	TDATA2	P305	—	IRQ8	SD0WP/ET1_TXD2/RGMII1_TXD2	GTOVUP/GTCPPO12/ULPTEE1	—	VIO_D14
C12	TDATA0	P307	—	IRQ27-DS	CTS_RTS6_A/SS6_A/DE6/SD0CMD_B/ ET1_TXD0/RGMII1_TXD0/RMII1_TXD0	GTIV/GTCPPO10/ULPTOA1	—	VIO_D12
C13	—	P911	—	IRQ6	ET1_TXD5	GTIOC3B	—	—
C14	CLKOUT	P206	CS7	IRQ0-DS	USB_VBUSEN/SSIDATA1_A/ SD0DAT7_C/ET1_RX_DV/ RGMII1_RX_CTL/RMII1_CRS_DV	GTIU/GTCPPO0/ULPTOB1	—	VIO_D0
C15	—	PD04	—	IRQ20	CTS_RTS8_C/SS8_C/DE8/USBHS_ID/ SD0CMD_C/ET0_RXD5	GTIOC3A	—	—
C16	—	PD03	—	IRQ21	RXD8_C/SCL8_C/MISO8_C/ USBHS_EXICEN/SD0DAT0_C/ ET0_RXD4	GTIOC3B	—	—
C17	—	PD05	—	IRQ19	CTS8_C/USBHS_OVRCURB/SD0CLK_C/ ET0_RXD6	GTIOC2B	—	—
C18	—	PD06	—	IRQ18	USBHS_OVRCURA/SD0WP/ET0_RXD7	GTIOC2A	—	—
D1	—	PA04	A1	IRQ19	SCK2_C/DE2/SD0DAT3_A	GTIU/GTIOC4B	ADST0	VIO_D9
D2	CACREF/CLKOUT	P611	D13/DQ13	IRQ17	SCK0_C/DE0/MOSIA_B/ USBHS_VBUSEN	GTOULO/GTIOC4B	—	—
D3	—	P610	D12/DQ12	IRQ16	RXD0_C/SCL0_C/MISO0_C/RSPCKA_B/ CRX1	GTOUUP/GTIOC4A/ULPTOB1- DS	—	—
D4	—	PA14	D11/DQ11	IRQ13	TXD9_C/SDA9_C/MOSI9_C	GTOVLO/GTIOC10B	—	—
D5	—	P303	—	IRQ29-DS	SCK6_B/DE6	GTIOC7B	—	—
D6	—	P915	—	IRQ8	CTS6_B	GTIOC5A	—	—
D7	—	P108	—	IRQ24	SD0DAT6_C	GTIOC10B	—	—
D8	—	P111	—	IRQ19	SD0DAT3_C	GTIOC9A	—	—
D9	—	P109	—	IRQ23	SD0DAT5_C	GTIOC10A	—	—
D10	—	P310	—	IRQ24-DS	TXD3_B/SDA3_B/MOSI3_B/ET1_TX_EN/ RGMII1_TX_CTL/RMII1_TX_EN	GTCPPO7/AGTEE1	—	VIO_D9
D11	TDATA3	P304	—	IRQ9	SD0DAT0_B/ET1_TXD3/RGMII1_TXD3	GTOVLO/GTIOC7A/ULPTO1	—	VIO_D15
D12	TDATA1	P306	—	IRQ28-DS	SD0CD/ET1_TXD1/RGMII1_TXD1/ RMII1_TXD1	GTIW/GTCPPO11/ULPTEVI1	—	VIO_D13
D13	—	P912	—	IRQ5	ET1_TXD6	GTIOC3A	—	—
D14	—	PB04	—	IRQ9	SCK5_C/DE5/ET0_TXD3/RGMII0_TXD3	GTCPPO3	AD0FLAG1	VIO_CLK
D15	—	PB07	—	IRQ1	ET0_TXD5	GTIOC9B	—	—
D16	—	PB05	—	IRQ15	CTS5_C/ET0_TXD7	GTCPPO4	—	—
D17	—	PB03	—	IRQ13	TXD5_C/SDA5_C/MOSI5_C/ET0_TXD2/ RGMII0_TXD2	GTCPPO1	ADSYNC	VIO_HD
D18	—	PB01	ALE	IRQ12	CTS_RTS1_B/SS1_B/DE1/ET0_TX_CLK	GTCPPO2	AD1FLAG1	VIO_FLD
E1	—	PA15	EBCLK/ SDCLK	IRQ14	CTS9_C/PDMCLK1	GTIOC7A	—	VIO_D14
E2	—	P615	—	IRQ7	TXD7_A/SDA7_A/MOSI7_A/ USBHS_EXICEN	GTETRGC/GTCPPO10	—	—
E3	—	P614	WR/WR0/ DQM0	IRQ20	RXD7_A/SCL7_A/MISO7_A/USBHS_ID	GTETRGB/GTCPPO9/AGTO0	—	—

Table 1.17 Pin list for the SiP product (3 of 8)

BGA303	Power, System, Clock, Debug, CAC	I/O ports	ExBus/ SDRAM	Ex.Interrupt	SCI/IIC/I3C/SPI/CANFD/USBFS/USBHS/ OSPI/SSIE/SDHI/MMC/ESWM(GMII, RGMII, MII, RMII)/PDMIF	GPT/AGT/ULPT/RTC	ADC16H/ DAC12/ ACMPHS	CEU
E4	—	P612	D14/DQ14	IRQ18	CTS_RTS0_C/SS0_C/DE0/SSLA0_B/ USBHS_OVRCURA	GTIOC9A	—	—
E5	—	P914	—	IRQ9	CTS_RTS6_B/SS6_B/DE6	GTIOC5B	—	—
E6	MD	P201	—	IRQ4	—	—	—	—
E10	—	P902	ALE	IRQ0	AUDIO_CLK/ETHPHYCLK	GTCPPO13	—	VIO_D1
E11	—	P312	—	IRQ22-DS	CTS_RTS3_B/SS3_B/DE3/CTX0/ ET1_RX_ER/RMII1_RX_ER/PDMDAT2	GTADSM0/GTCPPO5/AGTOA1	—	VIO_D7
E12	—	P910	—	IRQ7	ET1_TXD4	GTCPPO12	—	—
E13	CLKOUT	P913	—	IRQ3	ET1_TXD7	GTCPPO11	—	—
E14	—	PB02	—	IRQ11	RXD5_C/SCL5_C/MISO5_C/ET0_TXD1/ RGMII0_TXD1/RMII0_TXD1	GTCPPO0	ADST1	VIO_VD
E15	—	PB06	—	IRQ0	CTS_RTS5_C/SS5_C/DE5/ET0_TXD6	GTIOC9A	—	—
E16	—	PD07	—	IRQ17	USBHS_VBUSEN/SD0CD/ET0_TXD4	GTCPPO0	—	—
E17	—	PB00	—	IRQ10	SCK1_B/DE1/USBHS_VBUSEN/ ET0_TXD0/RGMII0_TXD0/RMII0_TXD0/ PDMDAT2	GTCPPO4	ADST0	—
E18	—	P706	—	IRQ7	RXD1_B/SCL1_B/MISO1_B/ USBHS_OVRCURB-DS/ET0_GTX_CLK/ RGMII0_TXC/ETHPHYCLK/PDMDAT0	GTCPPO2/AGTIO0	—	VIO_D10
F1	—	PA02	A3	IRQ31	RXD2_C/SCL2_C/MISO2_C/SD0DAT5_A	GTIW/GTCPPO9	ADSYNC	VIO_D7
F2	—	PA10	CS2/RAS	IRQ4	SCK5_B/DE5/PDMCLK0	GTCPPO13	—	VIO_D15
F3	—	PA08	CS0/WE	IRQ6	RXD5_B/SCL5_B/MISO5_B	GTETRGD/GTCPPO11	—	—
F4	—	PA09	CS3/CAS	IRQ5	TXD5_B/SDA5_B/MOSI5_B	GTCPPO12	—	—
F5	—	PC14	—	IRQ0	TXD6_C/SDA6_C/MOSI6_C/ET0_WOL	GTADSM1/GTCPPO9	—	—
F7	VCC_08	—	—	—	—	—	—	—
F8	VSS_08	—	—	—	—	—	—	—
F9	VSS3	—	—	—	—	—	—	—
F10	VCL3	—	—	—	—	—	—	—
F11	VSS_07	—	—	—	—	—	—	—
F12	VCC_07	—	—	—	—	—	—	—
F13	—	P700	—	IRQ16-DS	RXD2_B/SCL2_B/MISO2_B/MISOA_C/ SSIDATA1_B/SD1WP/ET0_RXD2/ RGMII0_RXD2	GTIOC5A	—	VIO_D4
F14	—	P702	—	IRQ18-DS	CTS2_B/RSPCKA_C/SSIBCK1_B/ SD1DAT5_B/ET0_RXD0/RGMII0_RXD0/ RMII0_RXD0	GTIOC6A/ULPTO0	—	VIO_D6
F15	—	P406	—	IRQ31	TXD2_B/SDA2_B/MOSI2_B/SSLA3_C/ SSIRXD0_A/SD1CD/ET0_RXD3/ RGMII0_RXD3	GTIOC1B	—	VIO_D3
F16	—	P701	—	IRQ17-DS	CTS_RTS2_B/SS2_B/DE2/MOSIA_C/ SSILRCK1_B/SSIFS1_B/SD1DAT4_B/ ET0_RXD1/RGMII0_RXD1/RMII0_RXD1	GTIOC5B/ULPTO1	—	VIO_D5
F17	—	P707	—	IRQ8	TXD1_B/SDA1_B/MOSI1_B/ USBHS_OVRCURA-DS/ET0_TX_ER/ ETHPHYCLK/PDMDAT1	GTCPPO3	—	VIO_D11
F18	—	P705	—	IRQ19	CTS1_B/SSLA2_C/CRX0/ET0_TX_EN/ RGMII0_TX_CTL/RMII0_TX_EN/ PDMCLK2	GTADSM1/GTCPPO1/AGTIO0	—	VIO_D9
G1	—	PA00	A5	IRQ22	CTS_RTS5_B/SS5_B/DE5/SD0DAT7_A	GTOVLO/GTCPPO7	AD1FLAG1	VIO_D5
G2	—	PA03	A2	IRQ20	TXD2_C/SDA2_C/MOSI2_C/SD0DAT4_A	GTIV/GTCPPO10	ADST1	VIO_D8
G3	—	PA05	A0/BC0/ DQM1	IRQ18	CTS_RTS2_C/SS2_C/DE2/SD0DAT2_A/ PDMDAT2	GTETRGD/GTIOC4A	—	VIO_D10
G4	—	PA07	RD	IRQ16	CTS7_A/SD0DAT0_A/PDMDAT0	GTETRGB/GTIOC7A	VCOUT	VIO_D12
G5	—	PC12	—	IRQ2	SCK6_C/DE6/ET0_MDIO	GTCPPO11	—	—
G7	VCC_09	—	—	—	—	—	—	—
G8	VSS_09	—	—	—	—	—	—	—
G9	VSS4	—	—	—	—	—	—	—
G10	VCL4	—	—	—	—	—	—	—
G11	VSS_06	—	—	—	—	—	—	—
G12	VCC_06	—	—	—	—	—	—	—

Table 1.17 Pin list for the SiP product (4 of 8)

BGA303	Power, System, Clock, Debug, CAC	I/O ports	ExBus/ SDRAM	Ex.Interrupt	SCI/IIC/I3C/SPI/CANFD/USBFS/USBHS/ OSPI/SSIE/SDHI/MMC/ESWM(GMII, RGMII, MII, RMII)/PDMIF	GPT/AGT/ULPT/RTC	ADC16H/ DAC12/ ACMPHS	CEU
G13	—	P405	—	IRQ30	SCK2_B/DE2/SSITXD0_A/SD1DAT3_B/ ET0_RX_DV/RGMII0_RX_CTL/ RMII0_CRS_DV	GTIOC1A/AGTIO1	—	VIO_D2
G14	—	P704	—	IRQ26	SSLA1_C/CTX0/SD1DAT7_B/ ET0_RX_ER/RMII0_RX_ER/PDMCLK1	GTADSM0/GTCPPO0/AGTO0	—	VIO_D8
G15	—	P703	—	IRQ19-DS	SSLA0_C/SD1DAT6_B/ET0_RX_CLK/ RGMII0_RXC/RMII0_REF50CK/ PDMCLK0	GTIOC6B/AGTO1	VCOUT	VIO_D7
G16	VSS_03	—	—	—	—	—	—	—
G17	VCC_05	—	—	—	—	—	—	—
G18	VSS_05	—	—	—	—	—	—	—
H1	—	P504	A7	IRQ7	SD0WP	GTOULO/GTCPPO1	—	VIO_D3
H2	—	P503	A6	IRQ6	SD0CD	GTOUUP/GTCPPO6	—	VIO_D4
H3	—	P505	A8	IRQ8	SD0CLK_A	GTOWUP/GTCPPO2	—	VIO_D2
H4	—	PA01	A4	IRQ21	CTS5_B/SD0DAT6_A	GTOVUP/GTCPPO8	AD0FLAG1	VIO_D6
H5	—	PC11	—	IRQ3	CTS_RTS6_C/SS6_C/DE6/ET0_MDC	GTCPPO12	—	—
H7	VCC_10	—	—	—	—	—	—	—
H8	VSS_10	—	—	—	—	—	—	—
H9	VSS7	—	—	—	—	—	—	—
H10	VCL5	—	—	—	—	—	—	—
H11	VSS5	—	—	—	—	—	—	—
H12	VCC_04	—	—	—	—	—	—	—
H13	VSS_04	—	—	—	—	—	—	—
H15	VCC_03	—	—	—	—	—	—	—
H16	VCC_USBHS	—	—	—	—	—	—	—
H17	USBHS_DP	—	—	—	—	—	—	—
H18	USBHS_DM	—	—	—	—	—	—	—
J1	—	P506	A9	IRQ9	SD0CMD_A	GTOWLO/GTCPPO3	—	VIO_D1
J2	—	P507	A10	IRQ10	CTS_RTS7_A/SS7_A/DE7/ET_TAS_STA0	GTADSM0/GTIOC0A	—	VIO_D0
J3	—	P508	A11	IRQ1	CTS5_A/ET_TAS_STA1	GTADSM1/GTIOC0B	—	VIO_VD
J4	—	P509	A12	IRQ2	CTS_RTS5_A/SS5_A/DE5/ET_TAS_STA2	GTIOC1A/ULPTEVI1	—	VIO_HD
J5	—	PC13	—	IRQ1	RXD6_C/SCL6_C/MISO6_C/ET0_INT	GTCPPO10	—	—
J7	VCC2_11	—	—	—	—	—	—	—
J8	VSS_11	—	—	—	—	—	—	—
J9	VCL7	—	—	—	—	—	—	—
J10	VCL6	—	—	—	—	—	—	—
J11	VSS6	—	—	—	—	—	—	—
J12	VCL2	—	—	—	—	—	—	—
J13	VSS2	—	—	—	—	—	—	—
J15	VSS_02	—	—	—	—	—	—	—
J16	USBHS_RREF	—	—	—	—	—	—	—
J17	VSS2_USBHS	—	—	—	—	—	—	—
J18	VSS1_USBHS	—	—	—	—	—	—	—
K1	—	PC15	A16	IRQ30	CTS6_C/CRX1	GTADSM0	—	—
K2	CACREF	P608	A14	IRQ22	TXD5_A/SDA5_A/MOSI5_A	GTOWUP/GTCPPO4	—	VIO_FLD
K3	—	P510	A13	IRQ3	RXD5_A/SCL5_A/MISO5_A/ ET_TAS_STA3	GTIOC1B/ULPTEVI0	—	VIO_CLK
K4	—	PD00	A15	IRQ23	SCK5_A/DE5/CTX1	GTOWLO/GTCPPO5	—	—
K5	VSS	—	—	—	—	—	—	—
K6	VSS	—	—	—	—	—	—	—
K7	VSS_12	—	—	—	—	—	—	—
K8	VSS9	—	—	—	—	—	—	—
K9	VCL9	—	—	—	—	—	—	—
K10	VCL8	—	—	—	—	—	—	—

Table 1.17 Pin list for the SiP product (5 of 8)

BGA303	Power, System, Clock, Debug, CAC	I/O ports	ExBus/ SDRAM	Ex.Interrupt	SCI/IIC/I3C/SPI/CANFD/USBFS/USBHS/ OSPI/SSIE/SDHI/MMC/ESWM(GMII, RGMII, MII, RMII)/PDMIF	GPT/AGT/ULPT/RTC	ADC16H/ DAC12/ ACMPHS	CEU
K11	VSS8	—	—	—	—	—	—	—
K12	VCL1	—	—	—	—	—	—	—
K13	VSS1	—	—	—	—	—	—	—
K15	VCC_02	—	—	—	—	—	—	—
K16	AVCC_USBHS	—	—	—	—	—	—	—
K17	XTAL	P213	—	IRQ2	TXD1_C/SDA1_C/MOSI1_C	GTETRGC/GTIOC0A/ULPTEE0	ADTRG1	—
K18	EXTAL	P212	—	IRQ3	RXD1_C/SCL1_C/MISO1_C	GTETRGD/GTIOC0B/AGTEE1	—	—
L1	—	PC10	—	IRQ4	—	GTCPPO13	—	—
L2	VSS	—	—	—	—	—	—	—
L3	PUP	—	—	—	—	—	—	—
L4	VCC2_16	—	—	—	—	—	—	—
L5	VSS_16	—	—	—	—	—	—	—
L7	VCC2_12	—	—	—	—	—	—	—
L8	VSS_14	—	—	—	—	—	—	—
L9	VSS_15	—	—	—	—	—	—	—
L10	VSS10	—	—	—	—	—	—	—
L11	VCL10	—	—	—	—	—	—	—
L12	VCL0	—	—	—	—	—	—	—
L13	VSS0	—	—	—	—	—	—	—
L14	—	P403	—	IRQ14-DS	CTS_RTS1_A/SS1_A/DE1/SSIBCK0_A/ SD1DAT1_B/ET1_WOL	GTIOC3A/RTCIC1	AD0FLAG1	—
L15	—	P404	—	IRQ15-DS	CTS1_A/SSILRCK0_A/SSIFS0_A/ SD1DAT2_B/ET0_WOL	GTIOC3B/RTCIC2	AD1FLAG1	—
L16	VCC_01	—	—	—	—	—	—	—
L17	XCOUT	P214	—	IRQ21	—	—	—	—
L18	XCIN/EXCIN	P215	—	IRQ20	—	—	—	—
M1	—	PC09	—	IRQ5	—	—	—	—
M2	VSS	—	—	—	—	—	—	—
M3	VSS	—	—	—	—	—	—	—
M4	VCC2_17	—	—	—	—	—	—	—
M5	VSS_17	—	—	—	—	—	—	—
M8	VCC2_14	—	—	—	—	—	—	—
M9	VCC2_15	—	—	—	—	—	—	—
M11	VSS11	—	—	—	—	—	—	—
M12	VCL11	—	—	—	—	—	—	—
M14	—	P414	A23	IRQ9	RXD4_B/SCL4_B/MISO4_B/SSLB0_B/ CRX1/ET1_MDIO	GTIOC0B	—	VIO_CLK
M15	CACREF	P402	—	IRQ4-DS	SCK1_A/DE1/CRX0/AUDIO_CLK/ SD1DAT0_B/ET0_LINKSTA	RTCIC0	—	—
M16	—	P410	A19	IRQ5	SCK3_A/DE3/SCL0_A*1/ USB_OVRCURB-DS/USBHS_OVRCURB/ GPTP_MATCH0	GTOVLO/GTIOC9B/AGTOB1	ADST0	—
M17	VBATT	—	—	—	—	—	—	—
M18	VSS_01	—	—	—	—	—	—	—
N1	—	PC08	—	IRQ29	—	GTCPPO8	—	—
N2	VSS	—	—	—	—	—	—	—
N3	VSS	—	—	—	—	—	—	—
N4	VCC2_18	—	—	—	—	—	—	—
N5	VSS_18	—	—	—	—	—	—	—
N7	—	P105	—	IRQ0	CTS_RTS8_B/SS8_B/DE8/SSLB2_A/ OM_0_ECSINT1	GTIOC1A/ULPTO1-DS	ADSYNC	—
N10	—	P810	—	IRQ21	SCK7_B/DE7/SD1DAT2_A/PDMCLK0	GTIOC10A/ULPTOA0	—	—
N14	—	P710	CS5	IRQ17	CTS4_B/SSLB3_B/ET0_LINKSTA	GTIOC11B	—	VIO_D12

Table 1.17 Pin list for the SiP product (6 of 8)

BGA303	Power, System, Clock, Debug, CAC	I/O ports	ExBus/ SDRAM	Ex.Interrupt	SCI/IIC/I3C/SPI/CANFD/USBFS/USBHS/ OSPI/SSIE/SDHI/MMC/ESWM(GMII, RGMII, MII, RMII)/PDMIF	GPT/AGT/ULPT/RTC	ADC16H/ DAC12/ ACMPHS	CEU
N15	CACREF	P411	A20	IRQ4	CTS_RTS3_A/SS3_A/DE3/USB_ID/ USBHS_ID/GPTP_PTPOUT1	GTOVUP/GTIOC9A/AGTOA1	—	—
N16	—	P408	A17	IRQ7	RXD3_A/SCL3_A/MISO3_A/SCL0_B*1/ USB_VBUSEN/USBHS_VBUS/ GPTP_PTPOUT2	GTOWLO/GTIOC10A/ULPTOB0	ADSYNC	—
N17	—	P412	A21	IRQ20-DS	CTS3_A/USB_EXICEN/USBHS_EXICEN/ GPTP_PTPOUT0	GTOULO/GTCPPO8/AGTEE1	—	—
N18	—	P401	—	IRQ5-DS	RXD1_A/SCL1_A/MISO1_A/I3C_SDA0/ CTX0/SD1CMD_B	GTETRGA/GTIOC6B	—	VIO_D1
P1	VSS	—	—	—	—	—	—	—
P2	VSS	—	—	—	—	—	—	—
P3	VSS	—	—	—	—	—	—	—
P4	VCC2_19	—	—	—	—	—	—	—
P5	VSS_19	—	—	—	—	—	—	—
P6	—	P104	—	IRQ1	CTS9_A/SSLB1_A/OM_0_CS1	GTETRGB/GTIOC1B	AD0FLAG1	—
P7	—	P107	—	IRQ31	CTS4_A/OM_0_CS0	GTOWUP/GTIOC8A/AGTOA0	ADST0	—
P8	—	P106	—	IRQ16	CTS8_B/SSLB3_A/OM_0_RESET	GTOWLO/GTIOC8B/AGTOB0/ ULPTEE1-DS	ADST1	—
P9	—	P811	—	IRQ22	CTS7_B/USB_ID/SD1DAT3_A/PDMCLK1	GTIOC10B/ULPTOB0	—	—
P10	—	P013	—	IRQ14	—	—	AN013	—
P11	—	P011	—	IRQ16	—	—	AN011	—
P12	—	P807	—	IRQ11	—	GTIOC13A	—	—
P13	CACREF	P708	WR1/BC1	IRQ11	SCK4_B/DE4/SDA2_A*1/MOSIB_B/ AUDIO_CLK/ET0_MDC	GTCPPO6	—	VIO_VD
P14	—	P712	—	IRQ2	CTS1_C/SSLB1_B/GPTP_CAPTURE1	GTIOC2B/AGTOB0	—	—
P15	—	P714	—	IRQ13	TXD4_C/SDA4_C/MOSI4_C/GPTP_PPS1	GTIOC12B	—	—
P16	—	P711	—	IRQ3	CTS_RTS1_C/SS1_C/DE1/SSLB2_B/ GPTP_PPS0	GTIOC11A/AGTEE0	—	—
P17	—	P713	—	IRQ14	CTS4_C/GPTP_MATCH1	GTIOC2A/AGTOA0	—	—
P18	—	P400	—	IRQ0	TXD1_A/SDA1_A/MOSI1_A/I3C_SCL0/ AUDIO_CLK/SD1CLK_B	GTIOC6A/AGTIO1	ADTRG1	VIO_D0
R1	—	P602	—	IRQ28	RXD0_B/SCL0_B/MISO0_B	GTIOC7B/ULPTEE0	—	—
R2	VSS	—	—	—	—	—	—	—
R3	VSS	—	—	—	—	—	—	—
R4	CACREF	P600	—	IRQ30	OM_0_RSTO1	GTIOC6B/ULPTEVI1-DS	—	—
R5	—	P601	—	IRQ29	SCK0_B/DE0/OM_0_WP1	GTIOC6A/ULPTEVI0/RTCOUT	—	—
R6	—	P102	—	IRQ17	TXD9_A/SDA9_A/MOSI9_A/RSPCKB_A/ CRX0/OM_0_SIO4	GTOWLO/GTIOC2B/AGTO0	ADTRG0	—
R7	—	P801	—	IRQ12	TXD2_A/SDA2_A/MOSI2_A/OM_0_DQS	GTIV/GTIOC11B/AGTOB0	—	—
R8	—	P803	—	IRQ19	SCK2_A/DE2/OM_0_SIO1	GTETRGC/GTIOC12B	—	—
R9	—	P812	—	IRQ23	CTS_RTS7_B/SS7_B/DE7/USB_EXICEN/ SD1DAT4_A/PDMCLK2	GTIOC11A	AN022	—
R10	—	P012	—	IRQ15	—	—	AN012	—
R11	—	P010	—	IRQ14	—	—	AN010	—
R12	—	P009	—	IRQ13-DS	—	—	AN009/ IVREF1	—
R13	—	P805	—	IRQ30	TXD8_A/SDA8_A/MOSI8_A/ET1_MDIO	—	AN017/ IVCMP0	VIO_D15
R14	—	P512	—	IRQ14	CTS8_A/SCL1_A*1/CTX1/ET1_INT	GTIOC0A	—	—
R15	—	P413	A22	IRQ18	ET_TAS_STA3	GTOUUP/GTCPPO7/ULPTEE1	—	—
R16	—	P515	—	IRQ12	CTS_RTS4_C/SS4_C/DE4/SCL2_B*1/ ET_TAS_STA0	GTIOC13A	—	—
R17	—	P709	CS4	IRQ10	CTS_RTS4_B/SS4_B/DE4/SCL2_A*1/ MISOB_B/ET0_MDIO	GTCPPO5	—	VIO_D13
R18	—	P407	CS6	IRQ22	SCK1_C/DE1/SDA0_B*1/USB_VBUS/ USBHS_VBUSEN/GPTP_PTPOUT3	GTIOC10B/AGTIO0/RTCOUT	ADTRG0	—
T1	Vpp	—	—	—	—	—	—	—
T2	VSS	—	—	—	—	—	—	—

Table 1.17 Pin list for the SiP product (7 of 8)

BGA303	Power, System, Clock, Debug, CAC	I/O ports	ExBus/ SDRAM	Ex.Interrupt	SCI/IIC/I3C/SPI/CANFD/USBFS/USBHS/ OSPI/SSIE/SDHI/MMC/ESWM(GMII, RGMII, MII, RMII)/PDMIF	GPT/AGT/ULPT/RTC	ADC16H/ DAC12/ ACMPHS	CEU
T3	—	P315	—	IRQ29	SCK3_C/DE3/SSLA3_A	—	—	—
T4	—	P900	—	IRQ30	CTS3_C	GTADSM0	—	—
T5	—	P103	—	IRQ16	CTS_RTS9_A/SS9_A/DE9/SSLB0_A/ CTX0/OM_0_SIO2	GTOWUP/GTIOC2A	AD1FLAG1	—
T6	—	P101	—	IRQ1	RXD9_A/SCL9_A/MISO9_A/MOSIB_A/ OM_0_SIO3	GTETRGB/GTIOC8A/AGTEE0	—	—
T7	—	P802	—	IRQ18	RXD2_A/SCL2_A/MISO2_A/OM_0_SIO6	GTIW/GTIOC12A	—	—
T8	—	P804	—	IRQ14	CTS_RTS2_A/SS2_A/DE2/OM_0_SIO7	GTETRGD/GTIOC13A	—	—
T9	—	P501	—	IRQ25	TXD8_B/SDA8_B/MOSI8_B/ USB_OVRCURA/SD1DAT6_A/PDMDAT1	GTIOC12A	AN020	—
T10	AVCC0	—	—	—	—	—	—	—
T11	AVSS0	—	—	—	—	—	—	—
T12	—	P005	—	IRQ10-DS	—	—	AN005/ IVCMP3	—
T13	—	P003	—	IRQ29	—	—	AN003/ IVCMP3	—
T14	—	P513	—	IRQ31	SCK8_A/DE8/ET0_INT	GTIOC13B	AN016/ IVCMP0	VIO_FLD
T15	—	P514	—	IRQ13	SCK4_C/DE4/SDA2_B*1/ET_TAS_STA1	GTIOC13B	—	—
T16	—	P415	WAIT	IRQ8	TXD4_B/SDA4_B/MOSI4_B/RSPCKB_B/ CTX1/ET1_MDC	GTIOC0A	—	VIO_HD
T17	—	P409	A18	IRQ6	TXD3_A/SDA3_A/MOSI3_A/SDA0_A*1/ USB_OVRCURA-DS/USBHS_OVRCURA/ GPTP_CAPTURE0	GTOWUP/ULPTOA0	ADST1	—
T18	VCC_USB	—	—	—	—	—	—	—
U1	VCC2_13	—	—	—	—	—	—	—
U2	CLKOUT	P205	—	IRQ1-DS	TXD4_A/SDA4_A/MOSI4_A/SCL1_B*1/ SSLA1_A/USB_OVRCURA/SD1CD	GTIV/GTIOC4A/AGTO1	—	—
U3	—	P203	—	IRQ2-DS	RXD4_A/SCL4_A/MISO4_A/RSPCKA_A/ CTX0/USB_VBUSEN/SD1CLK_A	GTIOC5A/ULPTOA1	—	—
U4	—	P313	—	IRQ27	TXD3_C/SDA3_C/MOSI3_C/MISOA_A/ USB_ID/SD1DAT0_A	—	—	—
U5	—	P901	—	IRQ31	CTS_RTS3_C/SS3_C/DE3	GTADSM1/AGTIO1	—	—
U6	—	P809	—	IRQ20	TXD7_B/SDA7_B/MOSI7_B/ OM_0_SCLKN	—	—	—
U7	—	P800	—	IRQ11	CTS2_A/OM_0_SIO5	GTIU/GTIOC11A/AGTOA0	—	—
U8	—	P502	—	IRQ26	SCK8_B/DE8/USB_OVRCURB/ SD1DAT7_A/PDMDAT2	GTIOC12B	AN019	—
U9	—	P014	—	IRQ27	—	—	AN014/DA0/ IVCMP0	—
U10	VREFL	—	—	—	—	—	—	—
U11	VREFL0	—	—	—	—	—	—	—
U12	—	P004	—	IRQ9-DS	—	—	AN004/ IVCMP2	—
U13	—	P007	—	IRQ28	—	—	AN007/ IVCMP3	—
U14	—	P001	—	IRQ7-DS	—	—	AN001/ IVCMP3	—
U15	—	P806	—	IRQ0	RXD8_A/SCL8_A/MISO8_A/ET1_MDC	—	AN018	VIO_D14
U16	—	P715	—	IRQ12	RXD4_C/SCL4_C/MISO4_C/ ET_TAS_STA2	GTIOC12A	—	—
U17	—	P815	—	IRQ15	CTX0/USB_DM	GTIOC8A	—	—
U18	VSS_USB	—	—	—	—	—	—	—
V1	VSS	—	—	—	—	—	—	—
V2	CACREF	P204	—	IRQ26	SCK4_A/DE4/SDA1_B*1/SSLA0_A/ USB_OVRCURB/SD1WP	GTIW/GTIOC4B/AGTIO1	—	—
V3	—	P202	—	IRQ3-DS	CTS_RTS4_A/SS4_A/DE4/MOSIA_A/ CRX0/USB_EXICEN/SD1CMD_A	GTIOC5B/ULPTOB1	—	—
V4	—	P314	—	IRQ28	RXD3_C/SCL3_C/MISO3_C/SSLA2_A/ SD1DAT1_A	—	ADTRG0	—
V5	VSS_13	—	—	—	—	—	—	—

Table 1.17 Pin list for the SiP product (8 of 8)

BGA303	Power, System, Clock, Debug, CAC	I/O ports	ExBus/ SDRAM	Ex.Interrupt	SCI/IIC/I3C/SPI/CANFD/USBFS/USBHS/ OSPI/SSIE/SDHI/MMC/ESWM(GMII, RGMII, MII, RMII)/PDMIF	GPT/AGT/ULPT/RTC	ADC16H/ DAC12/ ACMPHS	CEU
V6	—	P808	—	IRQ15	RXD7_B/SCL7_B/MISO7_B/OM_0_SCLK	GTIOC13B	—	—
V7	—	P100	—	IRQ2	SCK9_A/DE9/MISOB_A/OM_0_SIO0	GTETRGA/GTIOC8B/AGTIO0	—	—
V8	CACREF	P500	—	IRQ24	RXD8_B/SCL8_B/MISO8_B/ USB_VBUSEN/SD1DAT5_A/PDMDAT0	GTIOC11B	AN021	—
V9	—	P015	—	IRQ13	—	—	AN015/DA1/ IVCMP0	—
V10	VREFH	—	—	—	—	—	—	—
V11	VREFH0	—	—	—	—	—	—	—
V12	—	P008	—	IRQ12-DS	—	—	AN008/ IVREF0	—
V13	—	P006	—	IRQ11-DS	—	—	AN006/ IVCMP2	—
V14	—	P000	—	IRQ6-DS	—	—	AN000/ IVCMP2	—
V15	—	P002	—	IRQ8-DS	—	—	AN002/ IVCMP2	—
V16	—	P511	—	IRQ15	CTS_RTS8_A/SS8_A/DE8/SDA1_A*1/ CRX1/ET1_LINKSTA	GTIOC0B	—	—
V17	—	P814	—	IRQ16	CRX0/USB_DP	GTIOC8B	—	—
V18	VSS	—	—	—	—	—	—	—

Note: Several pin names have the added suffix of _A, _B, and _C. These suffixes have special conditions for electrical characteristics. See section 62, Electrical Characteristics for detail.

Note 1. There are two types for IIC function, one is a simple IIC by SCI and another is a dedicated IIC. This terminal is for a dedicated IIC.

2. CPU

The MCU is based on the Arm® Cortex-M85 Processor and Arm® Cortex®-M33 Processor.

2.1 Overview

2.1.1 CPU

(1) CPU0

- Arm® Cortex-M85
  - Revision: r1p1-00rel0
  - ARMv8.1-M architecture profile
  - Floating Point Unit (FPU) compliant with the ANSI/IEEE Std 754-2008
    Scalar half, single, and double-precision floating-point operation.
    Interrupt notification when FPU exception occurs.
  - M-profile Vector Extension (MVE)
    Integer, half-precision, and single-precision floating-point MVE (MVE-F)
  - Armv8-M Security Extension
- Security Attribution Unit (SAU): 8 regions
- Implementation Defined Attribution Unit (IDAU)
- Memory Protection Unit (MPU)
  - Protected Memory System Architecture (PMSAv8)
  - Secure MPU (MPU_S): 8 regions
  - Non-secure MPU (MPU_NS): 8 regions
- System Timer (SysTick)
  Reference clock:
  - CPUCLK0
  - 1 MHz generated from on-chip oscillator (MOCO/8)

  Two system timers are implemented:
  - Secure instance (SysTick_S)
  - Non-secure instance (SysTick_NS)
- Low power modes
  - Sleep mode
  - Deep Sleep mode
- CPU individual reset
- Cache
  - Instruction cache: 16 KB with ECC
  - Data cache: 16 KB with ECC
- Tightly Coupled Memory (TCM)
  - ITCM: 128 KB (16 blocks × 8 KB) with ECC
  - DTCM: 128 KB (16 blocks × 8 KB) with ECC
- Nested Vectored Interrupt Controller (NVIC): 96 IRQs
- Programable MBIST Controller: Excluded
- Global exclusive monitor is not supported

See reference [1] ARM Limited., ARMv8-M Architecture Reference Manual (ARM DDI 0553B), [2] ARM Limited., Arm(R) Cortex-M85 Processor Technical Reference Manual (ARM 101924) for the detail.

(2) CPU1

- Arm® Cortex®-M33 Processor
  - Revision: r0p4-00rel2
  - ARMv8-M architecture profile
  - Floating Point Unit (FPU) compliant with the ANSI/IEEE Std 754-2008
    Single-precision floating-point operation.
    Interrupt notification when FPU exception occurs.
  - Armv8-M Security Extension (Can be disabled by OTP. CPU1 can only be selected as primary CPU if its Security Extension is enabled.)
  - Armv8-DSP Extension.
- Security Attribution Unit (SAU): 8 regions
- Implementation Defined Attribution Unit (IDAU)
- Memory Protection Unit (MPU)
  - Protected Memory System Architecture (PMSAv8)
  - Secure MPU (MPU_S): 8 regions
  - Non-secure MPU (MPU_NS): 8 regions
- System Timer (SysTick)
  Reference clock:
  - CPUCLK1
  - 1 MHz generated from on-chip oscillator (MOCO/8)

  Two system timers are implemented:
  - Secure instance (SysTick_S)
  - Non-secure instance (SysTick_NS)
- Low power modes
  - Sleep mode
  - Deep Sleep mode
- Cache
  - Code-bus Cache (C-Cache): 16 KB with ECC
  - System-bus Cache (S-Cache): 16 KB with ECC
- Tightly Coupled Memory (TCM)
  - CTCM: 64 KB with ECC
  - STCM: 64 KB with ECC
- Nested Vectored Interrupt Controller (NVIC): 96 IRQs
- Global exclusive monitor is not supported

See the following references for the detail:

- [1] ARM Limited., ARMv8-M Architecture Reference Manual (ARM DDI 0553B)
- [9] ARM Limited., Arm(R) Cortex-M33 Processor Technical Reference Manual (ARM 100230).

2.1.2 Debug

(1) CPU0 Debug Components

- Arm® CoreSight™ ETM-M85
  - Revision: r1p1-00rel0
  - ARM ETM Architecture version 4.5
- Instrumentation Trace Macrocell (ITM)
- Data Watchpoint and Trace unit (DWT)
  - 8 Data Watchpoint and Trace (DWT) comparators, up to 2 data value comparison
- Breakpoint Unit (BPU)
  - 8 instruction comparators
- Cross Trigger Interface (CTI0)
- Arm® CoreSight™ Trace Port Interface Unit for Cortex-M85 (TPIU)
  - TPIU Formatter output: Excluded
  - Serial Wire Output (SWO)

(2) CPU1 Debug Components

- Arm® CoreSight™ ETM-M33
  - Revision: r0p2-00rel0
  - ARM ETM Architecture version 4.2
- Instrumentation Trace Macrocell (ITM)
- Data Watchpoint and Trace unit (DWT)
  - 4 Data Watchpoint and Trace (DWT) comparators
- Breakpoint Unit (BPU)
  - 8 instruction comparators
- Cross Trigger Interface (CTI1)
- Arm® CoreSight™ Trace Port Interface Unit for Cortex-M33 (TPIU)
  - TPIU Formatter output: Excluded
  - Serial Wire Output (SWO)

(3) Common Debug Components

- Debug Access Port (DAP)
  - CoreSight SoC-400 Access Port :
    - #0: AHB-AP0
    - #1: APB-AP
    - #2: AHB-AP1
  - JTAG Debug Port and Serial Wire Debug Port (SWJ-DP)
    Serial Wire protocol version 2 (Target instance is fixed value: 0000)
- Timestamp Generator (TSG)
  - CoreSight SoC-400 Timestamp Generator
  - Global Timestamp for ETM and ITM
  - Driven by ICLK
- Embedded Cross Trigger

- CoreSight SoC-400 Cross Trigger Interface (CTI)
- CoreSight SoC-400 Cross Trigger Matrix (CTM)

- Embedded Trace FIFO (ETF)
  - CoreSight™ Trace Memory Controller with ETF configuration
  - FIFO size: 8 KB
- Debug Register Module (DBGREG)
- On-Chip Debug Register Module (OCDREG)
- Authentication Method using JTAG/SWD: Challenge response
- Trace Output Selector: Select between CPU0 and CPU1 SWO trace output.
- CoreSight SoC-400 Trace Port Interface Unit (TPIU) up to 4 bits TPIU Formatter output.

See the following references for the detail:

- [1] ARM Limited., ARMv8-M Architecture Reference Manual (ARM DDI 0553B)
- [2] ARM Limited., Arm(R) Cortex-M85 Processor Technical Reference Manual (ARM 101924)
- [4] ARM Limited., Arm(R) CoreSight SoC-400 Technical Reference Manual (ARM 0302-01 2nd release for r3p2)
- [9] ARM Limited., Arm(R) Cortex-M33 Processor Technical Reference Manual (ARM 100230).

2.2 Block Diagram

Figure 2.1 shows the block diagram of CPUs and CoreSight Debug subsystem (DBGSS).

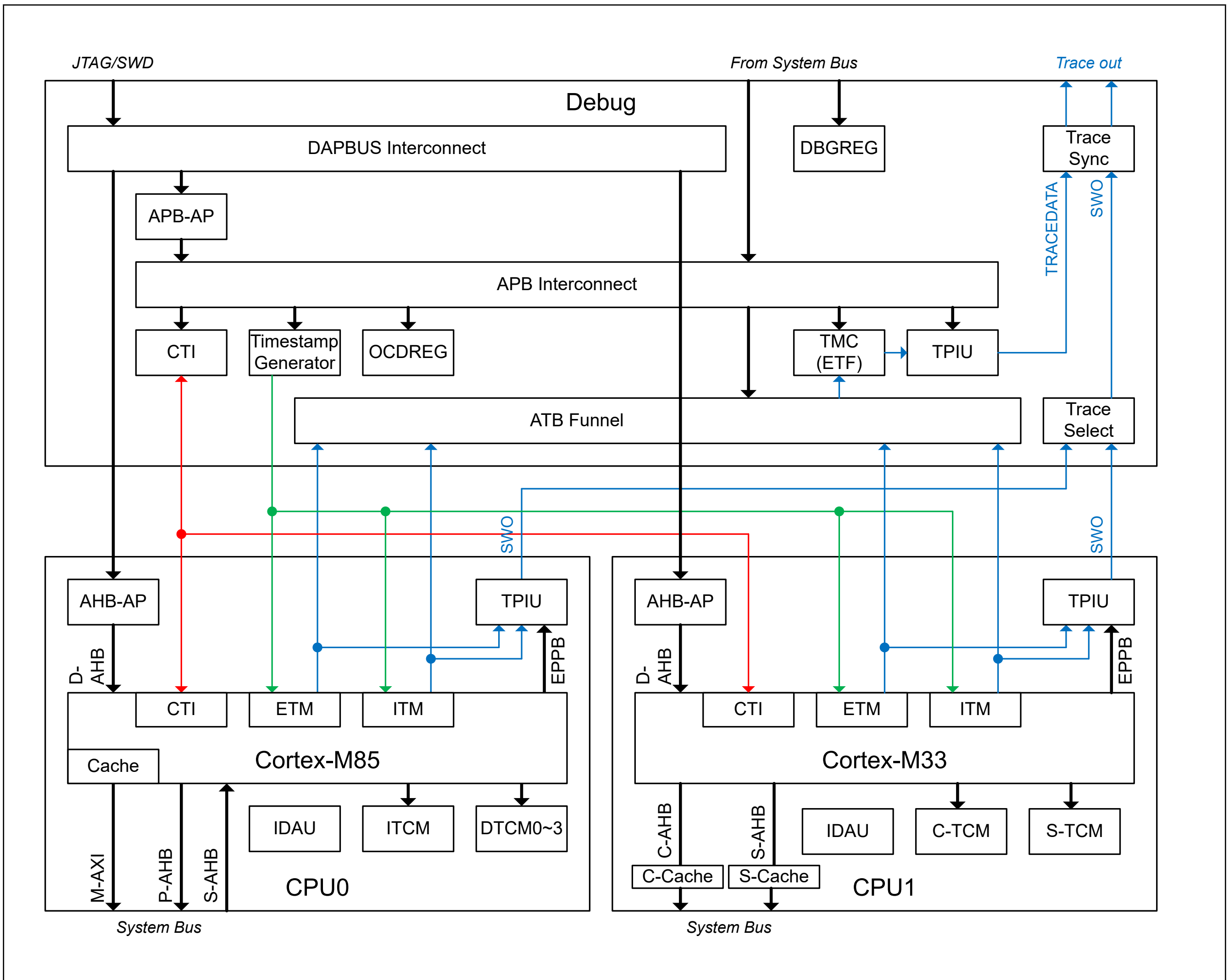


Figure 2.1 CPU and Debug block diagram

2.3 Implementation Options

Table 2.1 shows the implementation options related to CPU0 (Cortex-M85) and Table 2.2 shows the implementation options related to CPU1 (Cortex-M33).

Table 2.1 Implementation options related to CPU0 (1 of 2)

Option	Implementation
SAU	8 regions
IDAU	Included
MPU	MPU_S: 8 regions MPU_NS: 8 regions
MVE	Integer and half and single-precision floating-point MVE
FPU	Scalar half, single, and double-precision floating-point
Interrupts	96
Priority bits	4 bits (16 levels)
Sleep mode power-saving	Sleep mode, Deep Sleep mode
TCM	• ITCM : 128 KB (16 blocks × 8 KB) • DTCM : 128 KB (16 blocks × 8 KB) • ECC enable (determined by OFS1.INITECCEN) • ITCMCR.EN = 1 and DTCMCR.EN = 1, cannot be rewritten

Table 2.1 Implementation options related to CPU0 (2 of 2)

Option	Implementation
CACHE	● I-Cache: 16 KB ● D-Cache: 16 KB ● ECC enable (determined by OFS1.INITECCEN) ● Automatic invalidation is enabled During debug, the cache is controlled by CACHEDBGCR.L1RSTDIS.
P-AHB	512 MB, always enabled (PAHBCR.EN is always 1)
SysTick	Included two system timers: SysTick_S and SysTick_NS External reference clock with frequency = 1 MHz (MOCO/8) SYST_CALIB = 0x0000270F NOREF, bit[31] = 0: Reference clock is implemented SKEW, bit[30] = 0: TENMS calibration value is exact Bits[29:24] = 0x00: Reserved TENMS, bits[23:0] = 0x00270F: (0x270F + 1) x 1 [µs] = 10 [ms]
System reset request output	AIRCR.SYSRESETREQ bit causes the system reset
DWT	8 comparators
BPU	8 comparators
CTI	Included, CTI0 (for CPU0)
ITM	Included
ETM	Included
TPIU	Included ● TPIU Formatter output: Excluded ● Serial Wire Output (SWO)
Programmable MBIST Controller (PMC-100)	Excluded

Table 2.2 Implementation options related to CPU1 (1 of 2)

Option	Implementation
SAU	8 regions
IDAU	Included
MPU	MPU_S: 8 regions MPU_NS: 8 regions
SECEXT (Security Extension)	Supported. Can be enabled/disabled by setting OTP.
FPU	Single-precision floating-point
Interrupts	96
Priority bits	4 bits (16 levels)
Sleep mode power-saving	Sleep mode, Deep Sleep mode
SysTick	Included two system timers: SysTick_S and SysTick_NS External reference clock with frequency = 1 MHz (MOCO/8) SYST_CALIB = 0x0000270F NOREF, bit[31] = 0: Reference clock is implemented SKEW, bit[30] = 0: TENMS calibration value is exact Bit[29:24] = 0x00: Reserved TENMS, bits[23:0] = 0x00270F: (0x270F + 1) x 1 [µs] = 10 [ms]
System reset request output	AIRCR.SYSRESETREQ bit causes the system reset request output
DWT	4 comparators
BPU	8 comparators
CTI	Included (CTI1 for CPU1)
ITM	Included
ETM	Included

Table 2.2 Implementation options related to CPU1 (2 of 2)

Option	Implementation
TPIU	Included ● TPIU formatter output: Excluded ● Serial Wire Output (SWO)

Table 2.3 DBGSS implementation options

Option	Implementation
TSG	Included Driven by clock with frequency = ICLK
TMC (ETF configuration)	8 KB
DAP	SWJ-DP AHB-AP0 (for CPU0): AP0 APB-AP (for CoreSight SoC-400): AP1 AHB-AP1 (for CPU1): AP2
TPIU	Included ● Up to 32-bits TPIU formatter output

2.4 Trace Interface

A Trace Port Interface Unit (TPIU) and Serial Wire Output (SWO) provide trace output. Table 2.4 shows the MCU pins for the function. These pins are multiplexed with other functions.

Table 2.4 Trace function pins

Name	I/O	Function	When not in use
TCLK	Out	Trace clock	Open
TDATA0	Out	Trace data output	Open
TDATA1	Out	Trace data output	Open
TDATA2	Out	Trace data output	Open
TDATA3	Out	Trace data output	Open
SWO	Out	Serial Wire Output Mutiplexed with JTAG TDO pin	Open

2.5 JTAG/SWD Interface

Table 2.5 shows JTAG/SWD chip pins.

Table 2.5 JTAG/SWD pins

Name	I/O	Function	When not in use
TCK/SWCLK	In	JTAG Clock pin Serial Wire Data Clock pin	Pull-up
TMS/SWDIO	In-Out	JTAG TMS pin Serial Wire Data In-Out pin	Pull-up
TDI	In	JTAG TDI pin	Pull-up
TDO	Out	JTAG TDO pin Mutiplexed with Serial Wire Output	Open

2.6 Boot

The MCU supports Single-chip mode, Boot modes (JTAG, SCI/USB). For details of each mode, see section 4, Operating Modes.

2.6.1 Single-chip Mode

In this mode, CPUs execute user application code. However, when the First Stage Boot Loader (FSBL) execution is enabled, the primary CPU executes the FSBL code provided by Renesas before executing the user application code.

2.6.1.1 Boot CPU Selection

In the dual-core system, the primary CPU can be selected by the boot firmware command and booted first after a system reset release. By default, CPU0 is the primary CPU.

During the system reset release, the secondary CPU is in power gating state until it is activated by the primary CPU by writing to CPUnACTCSR register. Then depends on CPUnWAITCR.CPUWAIT, the secondary CPU may be in the wait state (CPUWAIT) or starts program execution. While in the wait state, code and data can be transferred to TCMs of the secondary CPU (when CPU0 is the secondary CPU, its activation status must be confirmed before code/data transferring or release of CPUWAIT). The primary CPU is not affected by CPUnWAITCR.CPUWAIT.

2.6.2 Initial Vector Table Base Address

In the MCU, the secure vector table base address of each CPU is determined according to the operating mode and whether it is the primary or secondary CPU as shown in Table 2.6, and the non-secure vector table base address is fixed at 0x0000_0000 for both CPUs except the case when CPU1 (CM33) with Security Extension is disabled, non-secure vector table base address is determined by CPUnINITVTOR value.

Table 2.6 Initial secure vector table base address in Single-chip mode

Operating mode	Primary CPU INITSVTOR	Secondary CPU INITSVTOR
Single-chip mode	0x0200_0000	CPUnINITVTOR

2.7 Sleep Modes for Low Power States

Each CPU supports two Sleep modes: Sleep mode and Deep Sleep mode.

2.7.1 Sleep Mode

2.7.1.1 Transition to Sleep Mode

When a WFI/WFE instruction or Sleep-on-exit function is executed while SCR.SLEEPDEEP bit is 0, the corresponding CPU enters Sleep mode. In this mode, the CPU stops operating but the contents of its internal registers are retained. Other peripheral functions do not stop. Available resets or interrupts in Sleep mode cause the CPU to cancel Sleep mode. All interrupt sources are available. If using an interrupt to cancel Sleep mode, you must set the associated IELSRn register before executing a WFI instruction. For details, see section 14, Interrupt Controller Unit (ICU).

Table 2.7 shows the operation of IWDT when primary CPU enters Sleep mode.

Table 2.7 IWDT operation when primary CPU enters Sleep mode

IWDT operation	OFS0.IWDTSTRT	OFS0.IWDTSTPCTL*1	IWDTCSTPR.SLCSTP*2
Stop counting	0 (auto start mode)	1	—
Continue counting	0 (auto start mode)	0	—
Stop counting	1 (register start mode)	—	1
Continue counting	1 (register start mode)	—	0

Note 1. OFS0.IWDTSTPCTL controls IWDT stops/continues counting while IWDT is in auto start mode and the primary CPU is in Sleep mode.

Note 2. IWDTCSTPR.SLCSTP controls IWDT stops/continues counting while IWDT is in register start mode and the primary CPU is in Sleep mode.

Counting by IWDT stops when the primary CPU enters Sleep mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 1 (IWDT stops when the primary CPU is in Sleep mode).

Counting by IWDT continues when the primary CPU enters Sleep mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 0 (IWDT does not stop when the primary CPU is in Sleep mode).

Counting by the IWDT stops when the primary CPU enters Sleep mode while the IWDT is in register start mode and the SLCSTP bit in IWDTCSTPR is 1 (IWDT stops when the primary CPU is in Sleep mode).

Counting by the IWDT continues when the primary CPU enters Sleep mode while the IWDT is in register start mode and the SLCSTP bit in IWDTCSTPR is 0 (IWDT does not stop when the primary CPU is in Sleep mode).

The operation of WDT0/1 when CPU0/1 enters Sleep mode is shown in Table 2.8 and Table 2.9.

Table 2.8 WDT0 operation when CPU0 enters Sleep mode

WDT0 operation	OFS0.WDTSTRT0	OFS0.WDTSTPCTL0*1	WDT0.WDTCSTPR.SLCSTP*2
Stop counting	0 (auto start mode)	1	—
Continue counting	0 (auto start mode)	0	—
Stop counting	1 (register start mode)	—	1
Continue counting	1 (register start mode)	—	0

Note 1. OFS0.WDTSTPCTL0 controls WDT0 stops/continues counting while WDT0 is in auto start mode and the CPU0 is in Sleep mode.
Note 2. WDT0.WDTCSTPR.SLCSTP controls WDT0 stops/continues counting while WDT0 is in register start mode and the CPU0 is in Sleep mode.

Counting by WDT0 stops when CPU0 enters Sleep mode while the WDT0 is in auto start mode and the OFS0.WDTSTPCTL0 bit is 1 (WDT0 stops when CPU0 is in Sleep mode). Similarly, counting by WDT0 stops when CPU0 enters Sleep mode while the WDT0 is in register start mode and the SLCSTP bit in WDTCSTPR is 1 (WDT0 stops when CPU0 is in Sleep mode).

Counting by WDT0 continues when the CPU0 enters Sleep mode while the WDT0 is in auto start mode and the OFS0.WDTSTPCTL0 bit is 0 (WDT0 does not stop when CPU0 is in Sleep mode). Similarly, counting by WDT0 continues when CPU0 enters Sleep mode while the WDT0 is in register start mode and the SLCSTP bit in WDTCSTPR is 0 (WDT0 does not stop when CPU0 is in Sleep mode).

Table 2.9 WDT1 operation when CPU1 enters Sleep mode

WDT1 operation	OFS3.WDTSTRT1	OFS3.WDTSTPCTL1*1	WDT1.WDTCSTPR.SLCSTP*2
Stop counting	0 (auto start mode)	1	—
Continue counting	0 (auto start mode)	0	—
Stop counting	1 (register start mode)	—	1
Continue counting	1 (register start mode)	—	0

Note 1. OFS3.WDTSTPCTL1 controls WDT1 stops/continues counting while WDT1 is in auto start mode and the CPU1 is in Sleep mode.
Note 2. WDT1.WDTCSTPR.SLCSTP controls WDT1 stops/continues counting while WDT1 is in register start mode and the CPU1 is in Sleep mode.

Counting by WDT1 stops when CPU1 enters Sleep mode while the WDT1 is in auto start mode and the OFS3.WDTSTPCTL1 bit is 1 (WDT1 stops when CPU1 is in Sleep mode). Similarly, counting by WDT1 stops when CPU1 enters Sleep mode while the WDT1 is in register start mode and the SLCSTP bit in WDTCSTPR is 1 (WDT1 stops when CPU1 is in Sleep mode).

Counting by WDT1 continues when the CPU1 enters Sleep mode while the WDT1 is in auto start mode and the OFS3.WDTSTPCTL1 bit is 0 (WDT1 does not stop when CPU1 is in Sleep mode). Similarly, counting by WDT1 continues when CPU1 enters Sleep mode while the WDT1 is in register start mode and the SLCSTP bit in WDTCSTPR is 0 (WDT1 does not stop when CPU1 is in Sleep mode).

For details on WFI/WFE instruction and sleep-on-exit function to enter Sleep mode, see [1] ARM Limited., ARMv8-M Architecture Reference Manual (ARM DDI 0553B).

2.7.1.2 Cancelling Sleep Mode

Sleep mode of each CPU is canceled by:

- An interrupt/event
- A reset:
  - RES pin reset
  - Power-on reset

- Voltage monitor 0 reset
- Overcurrent protection reset
- Core voltage monitor reset
- Independent watchdog timer reset
- CPUn (n = 0, 1) watchdog timer reset
- Voltage Monitor m reset (m = 1, 2 and 4, 5)
- Software reset
- CPUn (n = 0, 1) Lockup reset
- Common memory error reset
- Local memory n (n = 0, 1) error reset
- Bus error reset

The operations are as follows:

Canceling by an interrupt/event

- When an available interrupt request is generated, Sleep mode is canceled, and the CPUn (n = 0, 1) starts the interrupt handling.
- When the debugger writes 1 to MCUCTRL.EDBGRQn (n = 0, 1 for CPU0 and CPU1), Sleep mode is canceled, and the CPUn (n = 0, 1) enters Debug state[*1].
- When the CPUn (n = 0, 1) receives an event issued by an SEV instruction executed by another CPU and the transition to Sleep mode is caused by WFE instruction, Sleep mode is canceled, and the CPUn (n = 0, 1) executes the next instruction after WFE.

Canceling by RES pin reset

When the RES pin is driven low, the MCU enters the reset state. Be sure to keep the RES pin low for a period specified in section 62, Electrical Characteristics. When the RES pin is driven high after the specified period, the primary CPU starts the reset exception handling.

Canceling by IWDT reset

Sleep mode is canceled by a system reset generated by an IWDT underflow and the primary CPU starts the reset exception handling. However, IWDT stops in Sleep mode and a system reset for canceling Sleep mode is not generated in the following conditions:

- OFS0.IWDTSTRT = 0 (auto start mode) and OFS0.IWDTSTPCTL = 1
- OFS0.IWDTSTRT = 1 (register start mode) and IWDTCSTPR.SLCSTP = 1

Canceling by WDT0 reset

Sleep mode is canceled by a system reset generated by a WDT0 underflow and the primary CPU starts the reset exception handling. However, WDT0 stops in Sleep mode even when counting in Normal mode and a system reset for canceling Sleep mode is not generated in the following conditions:

- OFS0.WDTSTRT0 = 0 (auto start mode) and OFS0.WDTSTPCTL0 = 1
- OFS0.WDTSTRT0 = 1 (register start mode) and WDT0.WDTCSTPR.SLCSTP = 1

Canceling by WDT1 reset

Sleep mode is canceled by a system reset generated by a WDT1 underflow and the primary CPU starts the reset exception handling. However, WDT1 stops in Sleep mode even when counting in Normal mode and a system reset for canceling Sleep mode is not generated in the following conditions:

- OFS3.WDTSTRT1 = 0 (auto start mode) and OFS3.WDTSTPCTL1 = 1
- OFS3.WDTSTRT1 = 1 (register start mode) and WDT1.WDTCSTPR.SLCSTP = 1

Canceling by other resets available in Sleep mode

Sleep mode is canceled by other resets and the Primary CPU starts the reset exception handling.

Note: For details on proper setting of the interrupts, see section 14, Interrupt Controller Unit (ICU).

Note 1. CPU cannot enter Debug state if it is in Secure state while debugger is in Non-Secure state.

2.7.2 Deep Sleep Mode

2.7.2.1 Transition to Deep Sleep Mode

When a WFI instruction or sleep-on-exit function is executed while SCR.SLEEPDEEP is set to 1, the corresponding CPU enters Deep Sleep mode. In this mode, the CPU stops operating but the contents of its internal registers are retained. Other peripheral functions do not stop. However, accessing the TCM of CPU0 (when CPU0 is in Deep Sleep mode) is not available and SysTick also stops in this mode. Available resets or all interrupts in Deep Sleep mode cause the MCU to cancel Deep Sleep mode.

In Deep Sleep mode, the CPU is in power gating state (for CPU0, it depends on CPDLPSTATE setting).

IWDT and WDT0/1 operations in Deep Sleep mode is the same in Sleep mode.

2.7.2.2 Cancelling Deep Sleep Mode

Deep Sleep mode is canceled by:

- Interrupts shown in Table 14.5
- A reset:
  - RES pin reset
  - Power-on reset
  - Voltage monitor 0 reset
  - Overcurrent protection reset
  - Core voltage monitor reset
  - Independent watchdog timer reset
  - CPUn (n = 0, 1) watchdog timer reset
  - Voltage monitor m reset (m = 1, 2 and 4, 5)
  - Software reset
  - CPUn (n = 0, 1) lockup reset
  - Common memory error reset
  - Local memory n (n = 0, 1) error reset
  - Bus error reset

The operations are as follows:

Canceling by an interrupt

When an interrupt request is generated, Deep Sleep mode is canceled and the CPUn (n = 0, 1) starts the interrupt handling.

Canceling by RES pin reset

When the RES pin is driven low, the MCU enters the reset state. Be sure to keep the RES pin low for a period specified in section 62, Electrical Characteristics. When the RES pin is driven high after the specified period, the CPUs starts the reset exception handling.

Canceling by IWDT reset

Deep Sleep mode is canceled by a system reset generated by an IWDT underflow and the primary CPU starts the reset exception handling. However, IWDT stops in Deep Sleep mode and a system reset for canceling Deep Sleep mode is not generated in the following conditions:

- OFS0.IWDTSTRT = 0 (auto start mode) and OFS0.IWDTSTPCTL = 1
- OFS0.IWDTSTRT = 1 (register start mode) and IWDTCSTPR.SLCSTP = 1

Canceling by WDT0 reset

Deep Sleep mode is canceled by a system reset generated by a WDT0 underflow and the primary CPU starts the reset exception handling. However, WDT0 stops in Deep Sleep mode even when counting in Normal mode and a system reset for canceling Deep Sleep mode is not generated in the following conditions:

- OFS0.WDTSTRT0 = 0 (auto start mode) and OFS0.WDTSTPCTL0 = 1
- OFS0.WDTSTRT0 = 1 (register start mode) and WDT0.WDTCSTPR.SLCSTP = 1

Canceling by WDT1 reset

Deep Sleep mode is canceled by a system reset generated by a WDT1 underflow and the primary CPU starts the reset exception handling. However, WDT1 stops in Deep Sleep mode even when counting in Normal mode and a system reset for canceling Deep Sleep mode is not generated in the following conditions:

- OFS3.WDTSTRT1 = 0 (auto start mode) and OFS3.WDTSTPCTL1 = 1
- OFS3.WDTSTRT1 = 1 (register start mode) and WDT1.WDTCSTPR.SLCSTP = 1

Canceling by other resets available in Deep Sleep mode

Deep Sleep mode is canceled by other resets and the primary CPU starts the reset exception handling.

Note: For details on proper setting of the interrupts, see section 14, Interrupt Controller Unit (ICU).

2.8 Debug Function

The MCU provides ability to debug software by either external debug or self-hosted debug.

The external debug function is enabled by either JTAG/SWD or software programming. See section 2.8.1. External Debugger Connectivity, Authentication by JTAG/SWD and section 2.8.2. Debug Protection Mechanism by Software for details.

Self-hosted debug function can be enabled by software programming. See section 2.8.2. Debug Protection Mechanism by Software for details.

When debug function is enabled, operation of system can be affected. See section 2.8.3. Effect of Debug Function for details.

2.8.1 External Debugger Connectivity, Authentication by JTAG/SWD

In this MCU, three Protection Levels (PL) can be set:

- PL0 — Debug is not permitted
- PL1 — Non-secure debug is permitted
- PL2 — Secure debug is permitted.

When debug function is enabled by performing authentication using JTAG/SWD, three debug levels corresponding to three Authentication Level (AL) are available:

- AL0 — No debug functions are available
- AL1 — Only non-secure debug functions (both invasive and non-invasive debug) are enabled, and debugger can access only defined non-secure debug accessible regions
- AL2 — Non-secure and secure debug functions (both invasive and non-invasive debug) are enabled and accessible from the debugger.

Note: See reference [1] ARM Limited., ARMv8-M Architecture Reference Manual (ARM DDI 0553B) for details about invasive, non-invasive, secure, and non-secure debug.

Authentication Level is determined as follows:

- When Device Life Cycle state (DLM state) is not OEM (LCK_BOOT, RMA_RET, RMA_REQ, RMA_ACK), AL is determined by DLM state. See Table 2.10.
- When Device Life Cycle state (DLM state) is equal to OEM, AL can be determined by PL set by user and authentication performed using JTAG/SWD. Authentication method of the MCU is the challenge response method with the authentication key installed by user.

The initial value of Authentication Level (AL) is determined by the value of the Protection Level (PL). See Table 2.10.

For the authentication sequence, see section 2.14.2.2. Connecting Sequence and JTAG/SWD Authentication.

In addition, using authentication key AL1 and AL2 can be prohibited by setting Protection LCKS and LCKNS, respectively. Setting LCKS and LCKNS can be performed by a boot firmware command.

Table 2.10 shows condition for authentication level that can be used for serial programmer.

Table 2.10 Condition for authentication level

Condition				Available authentication level
DLM state	PL	LCKS	LCKNS	
OEM	PL2	Don't care	Don't care	AL2 (default)
OEM	PL1	111	Don't care	AL1 (default) AL2 (authentication using AL2_KEY)
OEM	PL1	000 (using AL2_KEY prohibited)	Don't care	AL1 (default)
OEM	PL0	111	111	AL0 (default) AL1 (authentication using AL1_KEY) AL2 (authentication using AL2_KEY)
OEM	PL0	111	000 (using AL1_KEY prohibited)	AL0 (default) AL2 (authentication using AL2_KEY)
OEM	PL0	000 (using AL2_KEY prohibited)	111	AL0 (Default) AL1 (authentication using AL1_KEY)
OEM	PL0	000 (using AL2_KEY prohibited)	000 (using AL1_KEY prohibited)	AL0 (default)
LCK_BOOT	PL0	Don't care	Don't care	AL0 (default)
RMA_RET	PL0	Don't care	Don't care	AL0 (default)
RMA_REQ	PL0	Don't care	Don't care	AL0 (default)
RMA_ACK	PL2	Don't care	Don't care	AL2 (default)

Note: When debugging using SWJ-DP, set CDBGPWRUPREQ = 1.

Note: Debug connection from external emulator to CoreSight components using Debug APB is enabled as default setting, and as don't care for authentication mechanism result. However, this connection can be disabled by software programming. See DBGAUTH0.DEVICEEN for details.

Debug capability is not available while MCU is operating with the Boot mode or during the FSBL execution.

2.8.2 Debug Protection Mechanism by Software

In this mechanism, the challenge and response authentication procedure is not performed, instead software can be programmed to enable debug function of the MCU directly. In other words, debug function is controlled by software completely.

The debug function can be enabled by setting OFS1.SWDBG = 0 and one of the following bits:

- DBGAUTH0.NIDENn = 1: Non-invasive debug enable (n = 0,1 for CPU0 and CPU1)
- DBGAUTH0.DBGENn = 1: Invasive debug enable (n = 0,1 for CPU0 and CPU1)
- DAUTHCTRL.SPNIDENSEL = 1 and DAUTHCTRL.INTSPNIDEN = 1: Secure non-invasive debug enable. See [1] ARM Limited., ARMv8-M Architecture Reference Manual (ARM DDI 0553B) for details.
- DAUTHCTRL.SPIDENSEL = 1 and DAUTHCTRL.INTSPIDEN = 1: Secure invasive debug enable. See [1] ARM Limited., ARMv8-M Architecture Reference Manual (ARM DDI 0553B) for details.

Note: Authentication using software programming is not restricted by DLM state, PL, LCKS and LCKNS.

Note: Except DAUTHCTRL.INTSPNIDEN and DAUTHCTRL.INTSPIDEN overrides processor external secure invasive/ non-invasive debug authentication interface, when debug function enabled by authentication using JTAG/SWD, it cannot be disabled by software.

2.8.3 Effect of Debug Function

The debug function affects inside and outside of CPU. This section describes the effects excluding debug authentication.

2.8.3.1 Low Power Mode

All CoreSight debug components can store the register settings even when the CPU enters Deep Sleep mode, or when the MCU enters Software Standby or Deep Software Standby mode if SYOCDCR.DBGEN is 1 (see section 11, Low Power Mode) and any of the following condition is true:

- CDBGPWRUPREQ = 1 and AL is AL2 or AL1
- OFS1.SWDBG is 0 and DBGAUTH0.DBGEN0 or DBGAUTH0.NIDEN0 is 1
- OFS1.SWDBG is 0 and DBGAUTH0.DBGEN1 or DBGAUTH0.NIDEN1 is 1

However, AHB-APn (n = 0, 1 for AHB-AP0, AHB-AP1 of each CPU0, 1) cannot respond to On-Chip Debug (OCD) access in these low power modes. The OCD must wait for cancellation of the low power mode to access the CoreSight debug components.

The MCU does not respond to On-Chip Debug (OCD), when OCD starts connecting to the MCU while it is in Software Standby mode or Deep Software Standby mode.

2.8.3.2 Reset/Interrupt

During debug when CPU enters debug state, the IWDT and corresponding WDTn (n = 0, 1) is stopped*1. During debug when CPU does not enter debug state and debug function is not AL0 or self-hosted debug is enabled, then some resets are restrained depend on DBGSTOPCR setting as WDT0, WDT1, IWDT*2. When PVDn (n! = 0), interrupt and reset depends on DBGSTOPCR whether CPU is in halting mode or not. Some access error is not generated by external debugger (DAP) access, see Table 2.11.

Table 2.11 Controlling of resets and interrupts generation in OCD mode

Reset/Interrupt name	Control in On-Chip Debug (OCD) mode	
	OCD break mode	OCD run mode
RES pin reset	Same as user mode	
Power-on reset	Same as user mode	
Independent watchdog timer reset/interrupt*2	Does not occur*1	Depends on DBGSTOPCR setting*2
Watchdog timer 0/1 reset/interrupt	Does not occur*1	Depends on DBGSTOPCR setting*2
CPU Lockup reset/interrupt	Does not occur	Same as user mode
Voltage monitor 0 reset	Same as user mode	
Voltage monitor 1 reset/interrupt	Depends on DBGSTOPCR setting*3	
Voltage monitor 2 reset/interrupt	Depends on DBGSTOPCR setting*3	
CPU0 Cache/TCM ECC error reset	Depends on DBGSTOPCR setting	
CPU1 Cache/TCM ECC error reset/interrupt	Cache: Depends on CAPOAD.OAD TCM: Depends on TCMCRC.OAD and TCMCRS.OAD	
MRAM ECC error reset/interrupt	Depends on DBGSTOPCR setting	
Cache/TCM ECC error BusFaults	Same as user mode*4	
SRAM ECC error reset/interrupt	Same as user mode*4	
Deep Software Standby reset	Same as user mode	
Software reset	Same as user mode	
Bus error reset/interrupt	Same as user mode*4	

Note 1. In OCD Break mode
IWDT always stops if any of the following condition is true:

- CDBGPWRUPREQ is 1, and AL is AL2 or AL1
- OSF1.SWDBG is 0, DBGAUTH0.DBGEN0 is 1, and CPU0 is the primary CPU
- OSF1.SWDBG is 0, DBGAUTH0.DBGEN1 is 1, and CPU1 is the primary CPU.

WDT0 always stops if any of the following condition is true:

- CDBGPWRUPREQ is 1, and AL is AL2 or AL1
- OSF1.SWDBG is 0 and DBGAUTH0.DBGEN0.

WDT1 always stops if any of the following condition is true:

- CDBGPWRUPREQ is 1, and AL is AL2 or AL1
- OSF1.SWDBG is 0 and DBGAUTH0.DBGEN1.

Note 2. In OCD run mode, IWDT and WDTn (n = 0, 1) operation depends on DBGSTOPCR setting if any of the following condition is true:
IWDT:

- CDBGPWRUPREQ is 1, and AL is AL2 or AL1
- OSF1.SWDBG is 0, DBGAUTH0.DBGEN0 is 1, and CPU0 is the primary CPU
- OSF1.SWDBG is 0, DBGAUTH0.DBGEN1 is 1, and CPU1 is the primary CPU.

WDT0:

- CDBGPWRUPREQ is 1, and AL is AL2 or AL1
- OSF1.SWDBG is 0 and DBGAUTH0.DBGEN0.

WDT1:

- CDBGPWRUPREQ is 1, and AL is AL2 or AL1
- OSF1.SWDBG is 0 and DBGAUTH0.DBGEN1.

Note 3. Reset or interrupt masking depends on DBGSTOPCR setting.
Note 4. The system reset request or interrupt request is not generated when an error is detected for DAP access (applies to the Cache and TCM of CPU1).

2.8.3.3 First Stage Bootloader (FSBL)

During FSBL execution, debug is prohibited regardless of Authentication Level (AL).

2.8.4 Hot Plug-in Debug

Hot plug-in debugging is supported if challenge-response authentication is not required. Also, it is not supported in Software Standby and Deep Software Standby modes.

2.8.5 Unprivileged Debug

Unprivileged debug is supported only in CPU0 (Armv8.1-M).

2.9 Register Descriptions

2.9.1 CPU Control Registers

In addition to the internal registers of the CPU, the MCU also provides additional specific control and status registers that are considered as CPU control register.

Table 2.12 shows a list of CPU control registers.

Table 2.12 List of CPU control registers (1 of 2)

Name	Abbreviation	Address (Secure)	Address (Non-secure)	Access size	R/W
CPU Security Attribution Register	CPUSAR	0x4000_8170	0x5000_8170	32-bit	Read/Write
CPU0 Lockup Control Register	CPU0LCKUPCR	0x4000_F030	0x5000_F030	8-bit	Read/Write
CPU1 Lockup Control Register	CPU1LCKUPCR	0x4000_F034	0x5000_F034	8-bit	Read/Write
CPU0 Control Register Protection Register	CPU0CRPT	0x4000_F840	0x5000_F840	16-bit	Read/Write

Table 2.12 List of CPU control registers (2 of 2)

Name	Abbreviation	Address (Secure)	Address (Non-secure)	Access size	R/W
CPU1 Control Register Protection Register	CPU1CRPT	0x4000_F844	0x5000_F844	16-bit	Read/Write
CPU0 Function Lock Control Register	CPU0LOCKCR	0x4000_F400	—	8-bit	Read/Write
CPU1 Function Lock Control Register	CPU1LOCKCR	0x4000_F404	—	8-bit	Read/Write
CPU0 Non-secure Function Lock Control Register	CPU0LOCKCRNS	—	0x5000_F500	8-bit	Read/Write
CPU1 Non-secure Function Lock Control Register	CPU1LOCKCRNS	—	0x5000_F504	8-bit	Read/Write
CPU0 Initial Vector Base Address Register	CPU0INITVTOR	0x4000_F040	—	32-bit	Read/Write
CPU1 Initial Vector Base Address Register	CPU1INITVTOR	0x4000_F044	0x5000_F044	32-bit	Read/Write
CPU0 CPUWAIT Control Register	CPU0WAITCR	0x4000_F050	—	8-bit	Read/Write
CPU1 CPUWAIT Control Register	CPU1WAITCR	0x4000_F054	0x5000_F054	8-bit	Read/Write
CPU0 Activation Control and Status Register	CPU0ACTCSR	0x4000_F060	—	16-bit	Read/Write
CPU1 Activation Control and Status Register	CPU1ACTCSR	0x4000_F064	0x5000_F064	16-bit	Read/Write
CPU0 Local Memory Error Control Register	CPU0LMECR	0x4000_F070	0x5000_F070	8-bit	Read/Write
CPU Identification Register	CPUIDR	0x4000_F078	0x5000_F078	8-bit	Read
CPU0 Status Monitor Register	CPU0STATM	0x4000_F080	0x5000_F080	8-bit	Read
CPU1 Status Monitor Register	CPU1STATM	0x4000_F084	0x5000_F084	8-bit	Read
CPU SECEXT Monitor Register	SECEXTMON	0x4000_F090	—	8-bit	Read
Non-secure CPU Control Register	NSCPUCR	0x4000_F094	—	32-bit	Read/Write

2.9.1.1 CPUSAR : CPU Security Attribution Register

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x170

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUSA1	CPUSA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CPUSA0	CPU Security Attribution 0 (CPU0) 0: Secure 1: Non-secure	R/W
1	CPUSA1	CPU Security Attribution 1 (CPU1) 0: Secure 1: Non-secure	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1
P-TYPE-1

CPUSAn bit

Security attributes of registers for CPUn control. The target registers are as follow.

- CPU.CPUnLCKUPCR

- CPU.CPUnCRPT

CPUSA0 bit

The CPUSA0 bit controls the security attributes of the following registers.

- CPU0RSTCR
- CPU0LMECR

CPUSA1 bit

The CPUSA1 bit controls the security attributes of the following registers.

- CPU1INITVTOR
- CPU1WAITCR
- CPU1ACTCSR

2.9.1.2 CPUnLCKUPCR : CPUn Lockup Control Register (n = 0, 1)

Base address: CPU_CTRL = 0x4000_F000
CPU_CTRL_NS = 0x5000_F000

Offset address: 0x030 + 0x004 × n

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

Bit	Symbol	Function	R/W
0	OAD	Operation after detection of CPUn lockup 0: Non-maskable interrupt 1: System reset	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3 (CPUSAR.CPUSAn)
P-TYPE-2
n = 0, 1

When CPUn enters the lockup state, it can be selected to generate either a system reset or an interrupt by writing to CPUnLCKUPCR.OAD.

The CPUnLCKUPCR security attribution is controlled by CPUSAR.

Writing to CPUnLCKUPCR.OAD bit is protected by CPU Control Register Protection Register (CPUnCRPT).

2.9.1.3 CPUnCRPT : CPUn Control Register Protection Register (n = 0, 1)

Base address: CPU_CTRL = 0x4000_F000
CPU_CTRL_NS = 0x5000_F000

Offset address: 0x840 + 0x004 × n

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

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: Writing to CPUnLCKUPCR register is available. 1: Writing to CPUnLCKUPCR register is not available. Read access is available.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
15:8	KEY[7:0]	The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 in KEY[7:0] bits at the same time. When values other than 0xA5 are written to KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.	W

Note: S-TYPE-3 (CPUSAR.CPUSAn)
P-TYPE-2
n = 0, 1

The CPUnCRPT.PROTECT bit enables writing to CPUnLCKUPCR, CPUnWAITCR, and CPUnLMECR registers (CPU1LMECR is not available).

The security attribution of CPUnCRPT is controlled by CPUSAR.

2.9.1.4 CPU0LOCKCR : CPU0 Function Lock Control Register

Base address: CPU_CTRL = 0x4000_F000

Offset address: 0x400

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	LCKDCAIC	LCKDTGU	LCKITGU	LCKSAU	LCKSMPU	LCKSVTAIR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LCKSVTAIR	Disables writes to the following secure registers from software or from a debug agent that is connected to the processor. VTOR_S, AIRCR.PRIS, AIRCR.BFHFNMINS	R/W*1
1	LCKSMPU	Disables writes to registers that are associated with the Secure MPU region from software or from a debug agent connected to the processor. MPU_CTRL, MPU_RNR, MPU_RBAR, MPU_RLAR, MPU_RBAR_An, MPU_RLAR_An	R/W*1
2	LCKSAU	This signal disables writes to registers that are associated with the SAU region from software or from a debug agent connected to the processor. SAU_CTRL, SAU_RNR, SAU_RBAR, SAU_RLAR	R/W*1
3	LCKITGU	Disables writes to registers that are associated with the ITCM interface security gating from software or from a debug agent connected to the processor. ITGUCTRL, ITGU_LUTn	R/W*1
4	LCKDTGU	Disables writes to registers that are associated with the DTCM interface security gating from software or from a debug agent connected to the processor. DTGUCTRL, DTGU_LUTn	R/W*1
5	LCKDCAIC	Disables access to the instruction cache direct cache access registers DCAICLR and DCAICRR	R/W*1
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-6
P-TYPE-2
Note 1. Writing 0 is ignored.

The CPU0LOCKCR register controls some miscellaneous signals of CPU0.

Reset is the only clear condition.

2.9.1.5 CPU1LOCKCR : CPU1 Function Lock Control Register

Base address: CPU_CTRL = 0x4000_F000

Offset address: 0x404

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

Bit	Symbol	Function	R/W
0	LCKSVTAIR	Disables writes to the following secure registers from software or from a debug agent that is connected to the processor. VTOR_S, AIRCR.PRIS, AIRCR.BFHFNMINS	R/W*1
1	LCKSMPU	Disables writes to registers that are associated with the Secure MPU region from software or from a debug agent connected to the processor. MPU_CTRL, MPU_RNR, MPU_RBAR, MPU_RLAR, MPU_RBAR_An, MPU_RLAR_An	R/W*1
2	LCKSAU	This signal disables writes to registers that are associated with the SAU region from software or from a debug agent connected to the processor. SAU_CTRL, SAU_RNR, SAU_RBAR, SAU_RLAR	R/W*1
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-6
P-TYPE-2
Note 1. Writing 0 is ignored.

The CPU1LOCKCR register controls some miscellaneous signals of CPU1.

Reset is the only clear condition.

2.9.1.6 CPUnLOCKCRNS : CPUn Non-secure Function Lock Control Register (n = 0, 1)

Base address: CPU_CTRL_NS = 0x5000_F000

Offset address: 0x500 + 0x004 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	LCKN SMPU	LCKN SVTO R
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LCKNSVTOR	Disables writes to the VTOR_NS register.	R/W*1
1	LCKNSMPU	Disables writes to registers that are associated with the Non-secure MPU region from software or from a debug agent connected to the processor. MPU_CTRL_NS, MPU_RNR_NS, MPU_RBAR_NS, MPU_RLAR_NS, MPU_RBAR_A_NSn, MPU_RLAR_A_NSn	R/W*1
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-7
P-TYPE-2
Note 1. Writing 0 is ignored.

The CPUnLOCKCRNS register controls some miscellaneous signals of CPUn.

Reset is the only clear condition.

2.9.1.7 CPUnINITVTOR : CPU Initial Vector Base Address Register (n = 0, 1)

Base address: CPU_CTRL = 0x4000_F000
CPU_CTRL_NS = 0x5000_F000

Offset address: 0x040 + 0x004 × n

Bit position: 31 ... 0

Bit field: CPUINITVTOR[31:0]

Value after reset: 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	CPUINITVTOR[31:0]	CPUn Initial Vector Base Address Writing to CPUINITVTOR[6:0] is discarded and read as 0.	R/W

Note: S-TYPE-6 (CPU0INITVTOR) / S-TYPE-3 (CPU1INITVTOR)
P-TYPE-2 (CPUnINITVTOR)

The CPUnINITVTOR register controls Vector Base Address bit field in the Secure/Non-secure Vector Table Offset.

2.9.1.8 CPUnWAITCR : CPUn CPUWAIT Control Register (n = 0, 1)

Base address: CPU_CTRL = 0x4000_F000
CPU_CTRL_NS = 0x5000_F000

Offset address: 0x050 + 0x004 × n

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

Bit	Symbol	Function	R/W
0	CPUWAIT	Writing 1 to stall the CPUn when it is out of reset. This bit is valid only when CPUn is the secondary CPU. 0: CPUn starts instruction execution out of reset 1: CPUn is forced into a quiescent state out of reset	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-6 (CPU0WAITCR) / S-TYPE-3 (CPU1WAITCR)
P-TYPE-2 (CPUnWAITCR)

The CPUnWAITCR register controls the execution state of the secondary CPU when it is out of reset.

Writing to CPUnWAITCR.CPUWAIT bit is protected by CPU Control Register Protection Register (CPUnCRPT).

2.9.1.9 CPUnACTCSR : CPUn Activation Control and Status Register (n = 0, 1)

Base address: CPU_CTRL = 0x4000_F000
CPU_CTRL_NS = 0x5000_F000

Offset address: 0x060 + 0x004 × n

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

Bit	Symbol	Function	R/W
0	ACTREQ	CPUn activation request This bit is read as 0. This bit is valid only when CPUn is the secondary CPU. 0: No effect 1: Request CPUn activation if ACT bit is 0	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	ACT	CPUn activation status 0: CPUn is in inactive state (power gating/reset state) 1: CPUn is in active state (out of reset)	R
15:8	KEY[7:0]	Key Code These bits control the write access to this register. To modify this register, write 0xA5 to the upper 8 bits and the target value to the lower 8 bits as a 16-bit unit. These bits are read as 0.	R/W

Note: The initial value of ACT bit is:
1: If CPUn is the primary CPU.
0: If CPUn is the secondary CPU.

Note: S-TYPE-6 (CPU0ACTCSR) / S-TYPE-3 (CPU1ACTCSR)
P-TYPE-2 (CPUnACTCSR)

The CPUnACTCSR register controls the activation of the secondary CPU after a system reset. It also indicates the activation status of CPUn.

2.9.1.10 CPU0LMECR : CPU0 Local Memory Error Control Register

Base address: CPU_CTRL = 0x4000_F000
CPU_CTRL_NS = 0x5000_F000

Offset address: 0x070

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

Bit	Symbol	Function	R/W
0	SYRSTEN	System Reset Request Enable 0: Disables system reset request when multiple bits error occurs in data cache or TCM 1: Enables system reset request when multiple bits error occurs in data cache or TCM	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3
P-TYPE-2

The CPU0LMECR register controls a system reset request when multiple bits error occurs in data cache or in TCM of CPU0.

Writing to CPU0LMECR.SYRSTEN bit is protected by CPU Control Register Protection Register (CPU0CRPT).

TCM must be initialized before setting CPU0LMECR.SYRSTEN bit because ECC error may occur if CPU0 and CPU1 issue a speculative access to the uninitialized TCM.

2.9.1.11 CPUIDR : CPU Identification Register

Base address: CPU_CTRL = 0x4000_F000
CPU_CTRL_NS = 0x5000_F000

Offset address: 0x078

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

Bit	Symbol	Function	R/W
0	CPUID	CPU Identification 0: CPU0 1: CPU1	R
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5
P-TYPE-5

The CPUIDR register shows the ID of CPUn. This register is expected to be read by CPUn. The read value is undefined if read by a bus master other than CPUn.

2.9.1.12 CPU0STATM : CPU0 Status Monitor Register

Base address: CPU_CTRL = 0x4000_F000
CPU_CTRL_NS = 0x5000_F000

Offset address: 0x080

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

Bit	Symbol	Function	R/W
0	SLEEPING	CPU0 Sleeping State 0: CPU0 is running 1: CPU0 is at Sleep mode	R
1	SLEEPDEEP	CPU0 SLEEPDEEP state Indicates that the processor is at a Deep Sleep mode.	R
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	SAHBSTP	CPU0 S-AHB Status Flag 0: CPU0 S-AHB is active 1: CPU0 S-AHB is stopped	R
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5
P-TYPE-5

The CPU0STATM register shows status of CPU0 related to Sleep, SleepDeep, and S-AHB Stop.

2.9.1.13 CPU1STATM : CPU1 Status Monitor Register

Base address: CPU_CTRL = 0x4000_F000
CPU_CTRL_NS = 0x5000_F000

Offset address: 0x084

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

Bit	Symbol	Function	R/W
0	SLEEPING	CPU1 Sleeping State 0: CPU1 is running 1: CPU1 is at Sleep mode	R
1	SLEEPDEEP	CPU1 SLEEPDEEP state Indicates that the processor is at a Deep Sleep mode.	R
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5
P-TYPE-5

The CPU1STATM register shows status of CPU1 related to Sleep and SleepDeep.

2.9.1.14 SECEXTMON : CPU SECEXT Monitor Register

Base address: CPU_CTRL = 0x4000_F000

Offset address: 0x090

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SECEXT1	SECEXT0
Value after reset:	0	0	0	0	0	0	0/1	1

Bit	Symbol	Function	R/W
0	SECEXT0	CPU0 Security Extension CPU0 (Cortex-M85) always includes Security Extension. 0: CPU0 does not include Security Extension 1: CPU0 includes Security Extension	R
1	SECEXT1	CPU1 Security Extension 0: CPU1 does not include Security Extension 1: CPU1 includes Security Extension	R
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-6
P-TYPE-2

The SECEXTMON register shows status of CPU0/1 related to Security Extension.

2.9.1.15 NSCPUCR : Non-secure CPU Control Register

Base address: CPU_CTRL = 0x4000_F000

Offset address: 0x094

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

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

Bit	Symbol	Function	R/W
0	RSTREQEN	System Reset Request Enable 0: System reset request caused by non-secure CPU without SECEXT is not allowed 1: System reset request caused by non-secure CPU without SECEXT is allowed	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-6
P-TYPE-2

The NSCPUCR register manages the system reset request of the CPU which does not have Security Extension.

2.9.2 Address Spaces for External Debugger

The MCU debug system includes three CoreSight Access Ports (AP):

- AHB-APn, which is connected to the CPUn bus matrix, has the same accessibility to the system address space as CPUn (n = 0, 1)
- APB-AP, which has OCD address space, is connected to the CoreSight components and OCDREG registers.

Figure 2.2 shows a block diagram of the AP connection and address spaces.

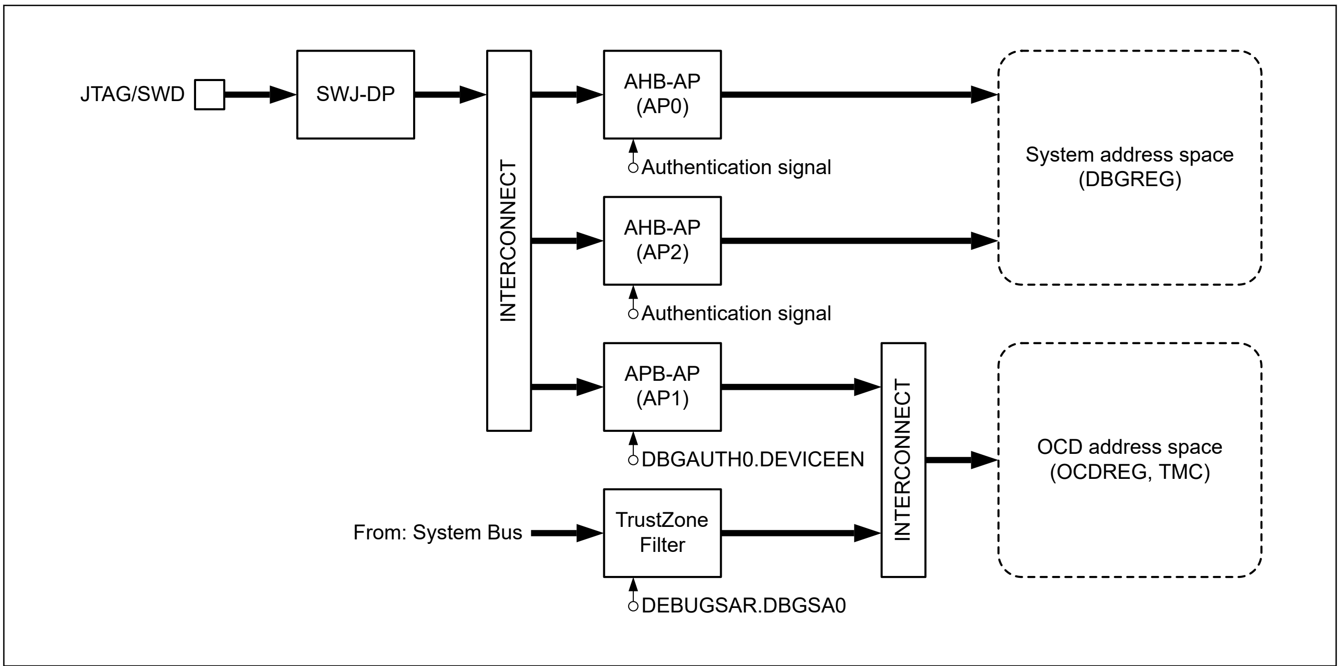


Figure 2.2 Block diagram of the AP connection and address spaces

For debugging purpose, there are two address spaces, DBGREG and OCDREG.

DBGREG is located in the system address space and can be accessed from either OCD emulator, CPUs or other bus masters in the MCU.

OCDREG is located in the OCD address space and can be accessed from either OCD emulator, CPUs or other bus masters in the MCU. See Table 2.15 for details.

2.9.3 CPU Peripheral

Each CPU core has its Private Peripheral Bus (PPB) which can be accessed only from that CPU and OCD emulator. CoreSight components are accessible from either CPUs or OCD emulator. Table 2.13 and Table 2.14 show dedicated Peripheral address of CPUn (n = 0, 1). Table 2.15 describes CoreSight component address.

Table 2.13 Peripheral address of CPU0 (1 of 2)

Component name	Start address	End address	Note
ITM	0xE000_0000	0xE000_0FFF	See reference [2] ARM Limited., Arm(R) Cortex-M85 Processor Technical Reference Manual (ARM 101924)
DWT	0xE000_1000	0xE000_1FFF	See reference [2] ARM Limited., Arm(R) Cortex-M85 Processor Technical Reference Manual (ARM 101924)
BPU	0xE000_2000	0xE000_2FFF	See reference [2] ARM Limited., Arm(R) Cortex-M85 Processor Technical Reference Manual (ARM 101924)
PMU	0xE000_3000	0xE000_3FFF	See reference [2] ARM Limited., Arm(R) Cortex-M85 Processor Technical Reference Manual (ARM 101924)
SCS	0xE000_E000	0xE000_EFFF	See reference [1] ARM Limited., ARMv8-M Architecture Reference Manual (ARM DDI 0553B)
SCS Non-Secure Alias	0xE002_E000	0xE002_EFFF	See reference [2] ARM Limited., Arm(R) Cortex-M85 Processor Technical Reference Manual (ARM 101924)
TPIU	0xE004_0000	0xE004_0FFF	See reference [2] ARM Limited., Arm(R) Cortex-M85 Processor Technical Reference Manual (ARM 101924)
ETM	0xE004_1000	0xE004_1FFF	See section 2.11. CoreSight ATB Funnel and reference [1] ARM Limited., ARMv8-M Architecture Reference Manual (ARM DDI 0553B)
CTI0	0xE004_2000	0xE004_2FFF	See reference [2] ARM Limited., Arm(R) Cortex-M85 Processor Technical Reference Manual (ARM 101924)

Table 2.13 Peripheral address of CPU0 (2 of 2)

Component name	Start address	End address	Note
EPPB ROM Table	0xE00F_E000	0xE00F_EFFF	—
Processor ROM Table	0xE00F_F000	0xE00F_FFFF	See reference [2] ARM Limited., Arm(R) Cortex-M85 Processor Technical Reference Manual (ARM 101924)

Table 2.14 Peripheral address of CPU1

Component name	Start address	End address	Note
ITM	0xE000_0000	0xE000_0FFF	See reference [9] ARM Limited., Arm(R) Cortex-M33 Processor Technical Reference Manual (ARM 100230)
DWT	0xE000_1000	0xE000_1FFF	See reference [9] ARM Limited., Arm(R) Cortex-M33 Processor Technical Reference Manual (ARM 100230)
BPU	0xE000_2000	0xE000_2FFF	See reference [9] ARM Limited., Arm(R) Cortex-M33 Processor Technical Reference Manual (ARM 100230)
SCS	0xE000_E000	0xE000_EFFF	See reference [1] ARM Limited., ARMv8-M Architecture Reference Manual (ARM DDI 0553B)
SCS Non-Secure Alias	0xE002_E000	0xE002_EFFF	See reference [9] ARM Limited., Arm(R) Cortex-M33 Processor Technical Reference Manual (ARM 100230)
TPIU	0xE004_0000	0xE004_0FFF	See reference [11] ARM Limited., Arm(R) CoreSight ETM-M33 Technical Reference Manual (ARM 100232)
ETM	0xE004_1000	0xE004_1FFF	See section 2.11. CoreSight ATB Funnel and reference [1] ARM Limited., ARMv8-M Architecture Reference Manual (ARM DDI 0553B)
CTI1	0xE004_2000	0xE004_2FFF	See reference [9] ARM Limited., Arm(R) Cortex-M33 Processor Technical Reference Manual (ARM 100230)
EPPB ROM Table	0xE00F_E000	0xE00F_EFFF	—
Processor ROM Table	0xE00F_F000	0xE00F_FFFF	See reference [9] ARM Limited., Arm(R) Cortex-M33 Processor Technical Reference Manual (ARM 100230)

Table 2.15 CoreSight component address

Component name	Start address*1	End address	Note
System ROM Table	0x4001_0000	0x4001_0FFF	See reference [3]
OCDREG	0x4001_1000	0x4001_1FFF	See reference [4] ARM Limited., Arm(R) CoreSight SoC-400 Technical Reference Manual (ARM 0302-01 2nd release for r3p2)
CTI	0x4001_2000	0x4001_2FFF	See reference [4] ARM Limited., Arm(R) CoreSight SoC-400 Technical Reference Manual (ARM 0302-01 2nd release for r3p2)
Funnel	0x4001_3000	0x4001_3FFF	See section 2.11. CoreSight ATB Funnel and reference [4] ARM Limited., Arm(R) CoreSight SoC-400 Technical Reference Manual (ARM 0302-01 2nd release for r3p2)
TMC (ETF configuration)	0x4001_4000	0x4001_4FFF	See reference [12] ARM Limited., Arm(R) CoreSight Trace Memory Controller Technical Reference Manual (ARM DDI0461B)
Time Stamp Generator	0x4001_5000	0x4001_5FFF	See section 3.1. Overview and reference [2] ARM Limited., Arm(R) Cortex-M85 Processor Technical Reference Manual (ARM 101924)
TPIU	0x4001_6000	0x4001_6FFF	See reference [4] ARM Limited., Arm(R) CoreSight SoC-400 Technical Reference Manual (ARM 0302-01 2nd release for r3p2)
Reserved area	0x4001_7000	0x4001_AFFF	—

Note: Do not access TPIU if trace clock is not supplied.
Note 1. The base address 0x4001_0000 is dedicated for CPU secure access.
The base address for CPU non-secure access is 0x5001_0000.
The base address for OCD emulator access is 0x8001_0000.
Do not access reserved area (secure: 0x4001_7000-0x4001_AFFF, non-secure: 0x5001_7000-0x5001_AFFF).
If reserved area is accessed, the access is not guaranteed.

2.9.4 ROM Table

There are five ROM Tables in the MCU:

- CPU0 processor ROM table
- CPU0 EPPB ROM table
- CPU1 processor ROM table
- CPU1 EPPB ROM table
- System ROM Table.

The CPUn EPPB ROM Table points to the TPIU component and processor ROM table inside each CPU. The EPPB ROM table does not point to or be pointed by the System ROM Table.

The Processor ROM Table contains the entries which hold a list of debug components implemented in CPUn (n = 0, 1). The System ROM Table contains the entries which hold a list of debug components implemented as CoreSight debug components of the MCU.

2.9.4.1 ROM Entries

ROM entries holds a list of CoreSight debug components in the system. OCD emulator can use the ROM entries to determine which components are implemented in the system.

The entries of debug components are described as follow:

- Table 2.16: System ROM entries
- Table 2.17: CPU0 EPPB ROM entries
- Table 2.18: CPU0 Processor ROM entries (see reference [2] ARM Limited., Arm(R) Cortex-M85 Processor Technical Reference Manual (ARM 101924))
- Table 2.19: CPU1 EPPB ROM entries
- Table 2.20: CPU1 Processor ROM entries (see reference [9] ARM Limited., Arm(R) Cortex-M33 Processor Technical Reference Manual (ARM 100230)).

Table 2.16 System ROM entries

#	Address*1	Access size	R/W	Value	Target module
0	0x4001_0000	32-bit	Read-only	0x00001003	OCDREG
1	0x4001_0004	32-bit	Read-only	0x00002003	CTI
2	0x4001_0008	32-bit	Read-only	0x00003003	Funnel
3	0x4001_000C	32-bit	Read-only	0x00004003	ETF
4	0x4001_0010	32-bit	Read-only	0x00005003	TSG
5	0x4001_0014	32-bit	Read-only	0x00006003	TPIU
6	0x4001_00148	32-bit	Read-only	0x00000000	End of entries

Note 1. The base address 0x4001_0000 is dedicated for secure access of CPUs. The base address for non-secure access of CPUs is 0x5001_0000. The base address for OCD emulator access is 0x8001_0000.

Table 2.17 CPU0 EPPB ROM entries

#	Address	Access size	R/W	Value	Target module
0	0xE00F_E000	32-bit	Read-only	0x00001003	CPU0 Processor ROM Table
1	0xE00F_E004	32-bit	Read-only	0xFFF42003	CPU0 TPIU
2	0xE00F_E008	32-bit	Read-only	0x00000000	End of entries

Note: The base address of CPU0 EPPB ROM Table is 0xE00F_E000.
Note: TPIU clock enable should be supplied before reading ROM table.

Table 2.18 CPU0 Processor ROM entries

#	Address	Access size	R/W	Value	Target module
0	0xE00F_F000	32-bit	Read-only	0xFFF0F003	SCS
1	0xE00F_F004	32-bit	Read-only	0xFFF02003	DWT
2	0xE00F_F008	32-bit	Read-only	0xFFF03003	BPU
3	0xE00F_F00C	32-bit	Read-only	0xFFF01003	ITM
5	0xE00F_F014	32-bit	Read-only	0xFFF42003	ETM
6	0xE00F_F018	32-bit	Read-only	0xFFF04003	PMU
7	0xE00F_F01C	32-bit	Read-only	0xFFF43003	CTI0
8	0xE00F_F020	32-bit	Read-only	0xFFF47002	PMC
9	0xE00F_F024	32-bit	Read-only	0x00000000	End of entries

Note: The base address of CPU0 Processor ROM Table is 0xE00F_F000.

Table 2.19 CPU1 EPPB ROM entries

#	Address	Access size	R/W	Value	Target module
0	0xE00F_E000	32-bit	Read-only	0x00001003	CPU1 Processor ROM Table
1	0xE00F_E004	32-bit	Read-only	0xFFF42003	CPU1 TPIU
2	0xE00F_E008	32-bit	Read-only	0x00000000	End of entries

Note: The base address of CPU1 EPPB ROM Table is 0xE00F_E000.
Note: TPIU clock enable should be supplied before reading ROM table.

Table 2.20 CPU1 Processor ROM entries

#	Address	Access size	R/W	Value	Target module
0	0xE00F_F000	32-bit	Read-only	0xFFF0F003	SCS
1	0xE00F_F004	32-bit	Read-only	0xFFF02003	DWT
2	0xE00F_F008	32-bit	Read-only	0xFFF03003	BPU
3	0xE00F_F00C	32-bit	Read-only	0xFFF01003	ITM
5	0xE00F_F014	32-bit	Read-only	0xFFF42003	ETM
6	0xE00F_F018	32-bit	Read-only	0xFFF43003	CTI1
7	0xE00F_F01C	32-bit	Read-only	0xFFF44002	MTB
8	0xE00F_F20	32-bit	Read-only	0x00000000	End of entries

Note: The base address of CPU1 Processor ROM Table is 0xE00F_F000.

2.9.4.2 CoreSight Component Registers

ROM Table has CoreSight Component registers defined in Arm® CoreSight™ Architecture.

Table 2.21 shows the registers of System ROM Table.

Table 2.22 shows CPU0 EPPB ROM Table.

Table 2.23 shows CPU0 Processor ROM Table. See reference [2] ARM Limited., Arm(R) Cortex-M85 Processor Technical Reference Manual (ARM 101924).

Table 2.24 shows CPU1 EPPB ROM Table.

Table 2.25 shows CPU1 Processor ROM Table. See reference [9] ARM Limited., Arm(R) Cortex-M33 Processor Technical Reference Manual (ARM 100230).

Table 2.21 Registers of System ROM Table (1 of 2)

Name	Address	Access size	R/W	Initial value
PID4	0x8001_0FD0	32-bit	Read-only	0x00000004

Table 2.21 Registers of System ROM Table (2 of 2)

Name	Address	Access size	R/W	Initial value
PID5	0x8001_0FD4	32-bit	Read-only	0x00000000
PID6	0x8001_0FD8	32-bit	Read-only	0x00000000
PID7	0x8001_0FDC	32-bit	Read-only	0x00000000
PID0	0x8001_0FE0	32-bit	Read-only	0x00000050
PID1	0x8001_0FE4	32-bit	Read-only	0x00000030
PID2	0x8001_0FE8	32-bit	Read-only	0x0000000A
PID3	0x8001_0FEC	32-bit	Read-only	0x00000000
CID0	0x8001_0FF0	32-bit	Read-only	0x0000000D
CID1	0x8001_0FF4	32-bit	Read-only	0x00000010
CID2	0x8001_0FF8	32-bit	Read-only	0x00000005
CID3	0x8001_0FFC	32-bit	Read-only	0x000000B1

Table 2.22 CPU0 EPPB ROM Table

Name	Address	Access size	R/W	Initial value
PID4	0xE00F_EFD0	32-bit	Read-only	0x00000004
PID5	0xE00F_EFD4	32-bit	Read-only	0x00000000
PID6	0xE00F_EFD8	32-bit	Read-only	0x00000000
PID7	0xE00F_EFDC	32-bit	Read-only	0x00000000
PID0	0xE00F_EFE0	32-bit	Read-only	0x00000050
PID1	0xE00F_EFE4	32-bit	Read-only	0x00000030
PID2	0xE00F_EFE8	32-bit	Read-only	0x0000000A
PID3	0xE00F_EFEC	32-bit	Read-only	0x00000000
CID0	0xE00F_EFF0	32-bit	Read-only	0x0000000D
CID1	0xE00F_EFF4	32-bit	Read-only	0x00000010
CID2	0xE00F_EFF8	32-bit	Read-only	0x00000005
CID3	0xE00F_EFFC	32-bit	Read-only	0x000000B1

Table 2.23 CPU0 processor ROM Table

Name	Address	Access size	R/W	Initial value
PID4	0xE00F_FFD0	32-bit	Read-only	0x00000004
PID5	0xE00F_FFD4	32-bit	Read-only	0x00000000
PID6	0xE00F_FFD8	32-bit	Read-only	0x00000000
PID7	0xE00F_FFDC	32-bit	Read-only	0x00000000
PID0	0xE00F_FFE0	32-bit	Read-only	0x000000D4
PID1	0xE00F_FFE4	32-bit	Read-only	0x000000B4
PID2	0xE00F_FFE8	32-bit	Read-only	0x0000000B
PID3	0xE00F_FFEC	32-bit	Read-only	0x00000000
CID0	0xE00F_FFF0	32-bit	Read-only	0x0000000D
CID1	0xE00F_FFF4	32-bit	Read-only	0x00000010
CID2	0xE00F_FFF8	32-bit	Read-only	0x00000005
CID3	0xE00F_FFFC	32-bit	Read-only	0x000000B1

Table 2.24 CPU1 EPPB ROM Table

Name	Address	Access size	R/W	Initial value
PID4	0xE00F_EFD0	32-bit	Read-only	0x00000004
PID5	0xE00F_EFD4	32-bit	Read-only	0x00000000
PID6	0xE00F_EFD8	32-bit	Read-only	0x00000000
PID7	0xE00F_EFDC	32-bit	Read-only	0x00000000
PID0	0xE00F_EFE0	32-bit	Read-only	0x00000050
PID1	0xE00F_EFE4	32-bit	Read-only	0x00000030
PID2	0xE00F_EFE8	32-bit	Read-only	0x0000000A
PID3	0xE00F_EFEC	32-bit	Read-only	0x00000000
CID0	0xE00F_EFF0	32-bit	Read-only	0x0000000D
CID1	0xE00F_EFF4	32-bit	Read-only	0x00000010
CID2	0xE00F_EFF8	32-bit	Read-only	0x00000005
CID3	0xE00F_EFFC	32-bit	Read-only	0x000000B1

Table 2.25 CPU1 processor ROM Table

Name	Address	Access size	R/W	Initial value
PID4	0xE00F_FFD0	32-bit	Read-only	0x00000004
PID5	0xE00F_FFD4	32-bit	Read-only	0x00000000
PID6	0xE00F_FFD8	32-bit	Read-only	0x00000000
PID7	0xE00F_FFDC	32-bit	Read-only	0x00000000
PID0	0xE00F_FFE0	32-bit	Read-only	0x000000C9
PID1	0xE00F_FFE4	32-bit	Read-only	0x000000B4
PID2	0xE00F_FFE8	32-bit	Read-only	0x0000000B
PID3	0xE00F_FFEC	32-bit	Read-only	0x00000000
CID0	0xE00F_FFF0	32-bit	Read-only	0x0000000D
CID1	0xE00F_FFF4	32-bit	Read-only	0x00000010
CID2	0xE00F_FFF8	32-bit	Read-only	0x00000005
CID3	0xE00F_FFFC	32-bit	Read-only	0x000000B1

2.9.5 DBGREG

Base_Address: 0x4001_B000 (Secure), 0x5001_B000 (Non-secure)

The DBGREG module controls the debug functionalities.

Table 2.26 shows a list of DBGREG registers.

Table 2.26 DBGREG registers (1 of 2)

Name		DAP port	Address	Access size	R/W
Debug Status Register	DBGSTR	Port 0/2	0x4001_B000 0x5001_B000	32-bit	Read/Write
Debug Stop Control Register	DBGSTOPCR	Port 0/2	0x4001_B010 0x5001_B010	32-bit	Read/Write
Debug Authentication Control Register 0	DBGAUTH0	Port 0/2	0x4001_B020 0x5001_B020	32-bit	Read/Write
Trace Port Control Register	TRPORTCR	Port 0/2	0x4001_B030 0x5001_B030	32-bit	Read/Write

Table 2.26 DBGREG registers (2 of 2)

Name		DAP port	Address	Access size	R/W
Trace Port Size Control Register	TRPORTSZ	Port 0/2	0x4001_B038 0x5001_B038	32-bit	Read/Write
Cache Debug Control Register	CACHEDBGCR	Port 0/2	0x4001_B040 0x5001_B040	32-bit	Read/Write
Debug Non-volatile Memory Control Register	DBGNVMCR	Port 0/2	0x4001_B050 0x5001_B050	32-bit	Read/Write
Authentication Level Control Register	ALCTRL	Port 0/2	0x4001_B100 0x5001_B100	32-bit	Read/Write
First Stage Boot Loader Status Register	FSBLSTAT	Port 0/2	0x4001_B200 0x5001_B200	32-bit	Read/Write

2.9.5.1 DEBUGSAR : Debug Security Attribution Register

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x180

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DBGSA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DBGSA0	Debug Security Attribution 0 0: Secure 1: Non-secure	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1
P-TYPE-1

DBGSA0 bit (Debug Security Attribution 0)

Security attributes of registers for debug control. The target registers are as follow:

- DBGREG.DBGAUTH0
- The CoreSight registers connected to APB-AP and mapped to the system address space.

Note: This is not affected to the access from external by APB-AP.

2.9.5.2 DBGSTR : Debug Status Register

Base address: CPU_DBG = 0x4001_B000
CPU_DBG_NS = 0x5001_B000

Offset address: 0x000

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

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

Bit	Symbol	Function	R/W
27:0	—	These bits are read as 0.	R
28	CDBGPWRUPREQ	Debug power-up request 0: OCD is not requesting debug power-up 1: OCD is requesting debug power-up	R
29	CDBGPWRUPACK	Debug power-up acknowledge 0: Debug power-up request is not acknowledged 1: Debug power-up request is acknowledged	R
31:30	—	These bits are read as 0.	R

Note: S-TYPE-5
P-TYPE-2

The DBGSTR register is a status register to notify OCD status to programs. This register monitors the debug connecting state.

2.9.5.3 DBGSTOPCR : Debug Stop Control Register

Base address: CPU_DBG = 0x4001_B000
CPU_DBG_NS = 0x5001_B000

Offset address: 0x010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	DBGSTOP_CTERR1	DBGSTOP_CTERR0	—	DBGSTOP_NVMERR	—	—	—	—	—	—	—	—	DBGSTOP_PVD	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	DBGSTOP_WDT1	DBGSTOP_WDT0	DBGSTOP_IWDT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	Symbol	Function	R/W
0	DBGSTOP_IWDT	Mask bit for IWDT reset/interrupt in the OCD run mode In the OCD break mode, the reset/interrupt is masked and IWDT counter is stopped, regardless of this bit value. 0: Enable IWDT reset/interrupt 1: Mask IWDT reset/interrupt and stop IWDT counter	R/W

Bit	Symbol	Function	R/W
1	DBGSTOP_WDT0	Mask bit for WDT0 reset/interrupt in the OCD run mode In the OCD break mode, the reset/interrupt is masked and WDT0 counter is stopped, regardless of this bit value. 0: Enable WDT0 reset/interrupt 1: Mask WDT0 reset/interrupt and stop WDT0 counter	R/W
2	DBGSTOP_WDT1	Mask bit for WDT1 reset/interrupt in the OCD run mode In the OCD break mode, the reset/interrupt is masked and WDT1 counter is stopped, regardless of this bit value. 0: Enable WDT1 reset/interrupt 1: Mask WDT1 reset/interrupt and stop WDT1 counter	R/W
16:3	—	These bits are read as 0. The write value should be 0.	R/W
17	DBGSTOP_PVD	Mask bit for PVDn (n! = 0) reset/interrupt 0: Enable PVDn (n! = 0) reset/interrupt 1: Mask PVDn (n! = 0) reset/interrupt	R/W
25:18	—	These bits are read as 0. The write value should be 0.	R/W
26	DBGSTOP_NVMERR	Mask bit for MRAM ECC error reset/interrupt 0: Enable MRAM ECC error reset/interrupt 1: Mask MRAM ECC error reset/interrupt	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
28	DBGSTOP_CTERR0	Mask bit for Cache/TCM ECC error reset of CPU0 0: Enable Cache/TCM ECC error reset 1: Mask Cache/TCM ECC error reset	R/W
29	DBGSTOP_CTERR1	This bit is reserved. It can be R/W but has no effect.	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5
P-TYPE-2

The DBGSTOPCR register specifies functional stop during debugging.

This register is valid only when any of the following conditions are true:

- CDBGPWRUPREQ is 1, and AL is AL2 or AL1
- OFS1.SWDBG is 0, and DBGAUTH0.DBGEN0 is 1
- OFS1.SWDBG is 0, and DBGAUTH0.DBGEN1 is 1.

When none of the above conditions are true, the MCU ignores this register setting and regards all bits as 0.

2.9.5.4 DBGAUTH0 : Debug Authentication Control Register0

Base address: CPU_DBG = 0x4001_B000
CPU_DBG_NS = 0x5001_B000

Offset address: 0x020

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SWDBG	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEVICEEN
Value after reset:	OFS1.SWDBG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	NIDEN1	NIDEN0	—	—	DBGEN1	DBGEN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DBGEN0[*1]	CPU0 invasive debug enable 0: Disabled 1: Enabled	R/W
1	DBGEN1[*1]	CPU1 invasive debug enable 0: Disabled 1: Enabled	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	NIDEN0	CPU0 non-invasive debug enable 0: Disabled 1: Enabled	R/W
5	NIDEN1	CPU1 non-invasive debug enable 0: Disabled 1: Enabled	R/W
15:6	—	These bits are read as 0. The write value should be 0.	R/W
16	DEVICEEN[*2]	APB-AP (AP1) authentication 0: Disabled 1: Enabled	R/W
30:17	—	These bits are read as 0. The write value should be 0.	R/W
31	SWDBG	Software control of debug function 0: Disabled 1: Enabled	R

Note: S-TYPE-3(DEBUGSAR.DBGSA0)
P-TYPE-2

Note: MCU behaves according to the setting of DBGSTOPCR, SYOCDCR, TRCKCR, TRACECTR, and MCUCTRL when DBGENn = 1 or NIDENn = 1 is TRUE or when CDBGPWRUPREQ = 1 and AL ≠ AL0 is TRUE (n = 0, 1).

Note 1. When AL2 or AL1, the MCU ignores this bit and regards as if it is set to 1.

Note 2. When AL2, the MCU ignores this bit and regards as if it is set to 1.

When software control of debug function (OFS1.SWDBG = 0) is enabled, non-secure invasive and non-invasive debug function of CPUn can be controlled by DBGENn and NIDENn bits of this register (n = 0, 1).

Access from APB-AP to the CoreSight components through Debug APB can be enabled by DEVICEEN bit of this register.

When OFS1.SWDBG is 1, the MCU ignores this register and regards it as if it is set to its initial value.

2.9.5.5 CACHEDBGCR : Cache Debug Control Register

Base address: CPU_DBG = 0x4001_B000
CPU_DBG_NS = 0x5001_B000

Offset address: 0x040

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	L1RST DIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	L1RSTDIS	Disable L1 cache automatic invalidation of CPU0 0: Enable automatic invalidation of the L1 cache 1: Disable automatic invalidation of the L1 cache	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-6
P-TYPE-2

Note: The setting of the register is valid only if any of the following conditions apply:

- CDBGPWRUPREQ = 1 and AL is AL2 or AL1
- OFS1.SWDBG is 0 and DBGAUTH0.DBGEN0 or DBGAUTH0.NIDEN0 is 1.

The CACHEDBGCR register provides capability of control Disable L1 cache invalidation out of reset during debugging.

2.9.5.6 TRPORTCR : Trace Port Control Register

Base address: CPU_DBG = 0x4001_B000
CPU_DBG_NS = 0x5001_B000

Offset address: 0x030

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

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

Bit	Symbol	Function	R/W
0	OE	Data Out Enable bit indicates whether trace clock, trace data and SWO outputs are enabled or not. OE bit should be enabled before trace clock is enabled. 0: Output disabled 1: Output enabled	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
3:2	DRV[1:0]	Port drive capability control indicates trace port buffer speed. 0 0: Low 0 1: Middle 1 0: High-Speed high-drive 1 1: High	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
9:8	PORTSEL[1:0]	Port select. Only A port setting is supported. 0 0: A port 0 1: B port	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	SWOSEL	Select SWO between CPU0 and CPU1 0: SWO of CPU0 is selected (default) 1: SWO of CPU1 is selected	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5
P-TYPE-2

Note: The setting of the register is valid only when any of the following conditions applies:

- CDBGPWRUPREQ = 1 and AL is AL2 or AL1
- OFS1.SWDBG is 0 and DBGAUTH0.DBGEN0 or DBGAUTH0.NIDEN0 is 1
- OFS1.SWDBG is 0 and DBGAUTH0.DBGEN1 or DBGAUTH0.NIDEN1 is 1.

Trace data control when OE bit = 1.

Target CPU	Trace Protocol	TPIU_SPPR (CPU0)	TPIU_SPPR (CPU1)	SWOSEL
CPU0	SWO	01b or 10b	Don't care	0 (CPU0)
	Parallel	00b	Don't care	0 (CPU0)

Target CPU	Trace Protocol	TPIU_SPPR (CPU0)	TPIU_SPPR (CPU1)	SWOSEL
CPU1	SWO	Don't care	01b or 10b	1 (CPU1)
	Parallel	Don't care	00b	1 (CPU1)
CPU0 and CPU1	SWO	01b or 10b	01b or 10b	Select CPU output SWO
	Parallel	00b to CPU0 or CPU1		Select CPU when TPIU_SPPR = 00b

For more information about TPIU_SPPR register (CPU0: bit PProtocol, CPU1: bit TXMODE), see [1] ARM Limited., ARMv8-M Architecture Reference Manual (ARM DDI 0553B).

2.9.5.7 ALCTRL : Authentication Level Control Register

Base address: CPU_DBG = 0x4001_B000
CPU_DBG_NS = 0x5001_B000

Offset address: 0x100

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

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

Note 1. Determined by PL value.

Bit	Symbol	Function	R/W
7:0	AL[7:0]	AL monitor Power-on reset initializes the AL to AL0. When the current device lifecycle is not OEM, AL is updated to the level corresponding to the device lifecycle. When the current device lifecycle is OEM: ● Without debugger connection: AL is updated to the same level as the PL ● With debugger connection: AL is updated to the same level as the PL only if PL is higher than AL The Renesas firmware updates the AL in Boot mode when the debug authentication is successful. The Renesas firmware updates the AL to a lower level in Boot mode regardless of authentication status. A debugger disconnection updates the AL to the same level as the PL. When FAILCNT[1:0] = 11, AL[7:0] stops updating (it is controlled by software). 0x00: AL0 0x05: AL1 0xA5: AL2	R/W
29:8	—	These bits are read as 0. The write value should be 0.	R/W
31:30	FAILCNT[1:0]	Number of times responding to incorrect response data 0 0: No responding to incorrect response data 0 1: First time of responding to incorrect response data 1 0: Second time of responding to incorrect response data 1 1: Third time of responding to incorrect response data	R/W

Note: S-TYPE-5
P-TYPE-5

Note:
- When CDBGPWRUPREQ changes from 1 to 0, the AL value is initialized to PL value
- Exiting from Boot mode when DBGPWRUPREQ = 0 also initializes the AL value
- Only accessible in Boot mode.

The ALCTRL register monitors the Authentication Level.

2.9.5.8 FSBLSTAT : First Stage Boot Loader Status Register

Base address: CPU_DBG = 0x4001_B000
CPU_DBG_NS = 0x5001_B000

Offset address: 0x200

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

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

Bit	Symbol	Function	R/W
0	CS	FSBL completion status 0: FSBL is executing 1: FSBL execution is complete	R/W
1	RS	FSBL result status 0: FSBL failed 1: FSBL passed	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
10:8	FSBLCLK[2:0]	System clock frequency selection during FSBL execution The selection reflects the value of FSBLCTRL0.FSBLCLK[2:0]. FSBLCTRL0 specifications are described in section 7, Option-Setting Memory.	R
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5
P-TYPE-5

The FSBLSTAT register notifies FSBL completion to the debugger.

This register is initialized by a system reset and is accessible only while FSBL is executing.

2.9.5.9 TRPORTSZ : Trace Port Size Control Register

Base address: CPU_DBG = 0x4001_B000
CPU_DBG_NS = 0x5001_B000

Offset address: 0x038

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

Bit	Symbol	Function	R/W
31:0	PORTSIZE[31:0]	Indicates how many pins of TRACEDATA are available. Only set 1 bit to this register, all others must be 0. Writing value with more than 1 bit set can cause Unpredictable behavior. The value set to this register must be the same as the value setting for CoreSight TPIU.CPSR register. Writing different value can cause Unpredictable behavior. Bit N is set: N + 1 bit(s) of TRACEDATA is supported (N = 0 to 31).	R/W

Note: S-TYPE-5
P-TYPE-2

Note: The setting of the register is valid only if any of the following conditions apply:

- CDBGPWRUPREQ = 1 and AL is AL2 or AL1
- OFS1.SWDBG is 0 and DBGAUTH0.DBGEN0 or DBGAUTH0.NIDEN0 is 1
- OFS1.SWDBG is 0 and DBGAUTH0.DBGEN1 or DBGAUTH0.NIDEN1 is 1.

The TRPORTSZ register controls how many pins of trace data are used.

2.9.5.10 DBGNVMCR : Debug Non-volatile Memory Control Register

Base address: CPU_DBG = 0x4001_B000
CPU_DBG_NS = 0x5001_B000

Offset address: 0x050

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

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

Bit	Symbol	Function	R/W
0	NVMWE	Non-volatile memory write enable 0: MRAM write is not allowed 1: MRAM write is allowed	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5
P-TYPE-2

Note: NVMWE is valid only when CPU is in halted state

2.9.5.11 CoreSight Component Registers

DBGREG has CoreSight Component registers defined in ARM® CoreSight™ Architecture. Table 2.27 shows a list of DBGREG CoreSight Component registers.

Table 2.27 CoreSight Component registers of DBGREG

Name	Address	Access size	R/W	Initial value
PID4	0x4001_BFD0	32-bit	Read-only	0x00000004
PID5	0x4001_BFD4	32-bit	Read-only	0x00000000
PID6	0x4001_BFD8	32-bit	Read-only	0x00000000
PID7	0x4001_BFDC	32-bit	Read-only	0x00000000
PID0	0x4001_BFE0	32-bit	Read-only	0x00000005
PID1	0x4001_BFE4	32-bit	Read-only	0x00000030
PID2	0x4001_BFE8	32-bit	Read-only	0x0000000A
PID3	0x4001_BFEC	32-bit	Read-only	0x00000000
CID0	0x4001_BFF0	32-bit	Read-only	0x0000000D
CID1	0x4001_BFF4	32-bit	Read-only	0x000000F0
CID2	0x4001_BFF8	32-bit	Read-only	0x00000005
CID3	0x4001_BFFC	32-bit	Read-only	0x000000B1

2.9.6 OCDREG

OCDREG is a register module provided for OCD (On-Chip Debug) Emulator functionalities. OCDREG is implemented as a CoreSight compliant component. Table 2.28 shows a list of OCDREG registers.

Note: Accesses to OCDREG from CPU and other bus masters is only permitted by privileged access, and one of the following conditions must be satisfied:

- Debugger is connected

- Self-hosted debug is enabled.

Accesses to OCDREG from external debug through APB-AP is permitted by either privileged access or unprivileged access.

Base address:

- CPU 0x4001_1000 (Secure), 0x5001_1000 (Non-Secure)
- OCD: 0x8001_1000

Table 2.28 OCDREG registers list

Name		DAP Port	OCD address	CPU address	Access size	R/W
MCU Status Register*1	MCUSTAT	Port 1	0x8000_0400	Cannot be accessed from CPU	32-bit	Read-only
MCU Error Status Register	MCUERRSTAT	Port 1	0x8001_1000	0x4001_1000 0x5001_1000	32-bit	Read-only
MCU Control Register	MCUCTRL	Port 1	0x8001_1004	0x4001_1004 0x5001_1004	32-bit	Read/Write
JTAG Boot Mode Entry Register	JBMDR	Port 1	0x8001_1100	0x4001_1100 0x5001_1100	32-bit	Read/Write
JTAG Boot Receive Data Register	JBRDR	Port 1	0x8001_1120	0x4001_1120 0x5001_1120	32-bit	Read/Write
JTAG Boot Transmit Data Register	JBTDR	Port 1	0x8001_1130	0x4001_1130 0x5001_1130	32-bit	Read/Write
JTAG Boot Status Register	JBSTR	Port 1	0x8001_1140	0x4001_1140 0x5001_1140	32-bit	Read/Write
JTAG Boot Interrupt Control Register	JBICR	Port 1	0x8001_1150	0x4001_1150 0x5001_1150	32-bit	Read/Write
First Stage Boot Loader Status Monitor Register	FSBLSTATM	Port 1	0x8001_1300	0x4001_1300 0x5001_1300	32-bit	Read-only

Note: Do not write to MCUCTRL using JTAG/SWD while the CPU is accessing MCUCTRL, JBMDR, JBRDR, JBTDR, and JBICR.
Do not write to JBMDR using JTAG/SWD except during RES pin reset.
Do not write to JBRDR using JTAG/SWD while the CPU is accessing MCUCTRL, JBMDR, JBRDR, JBTDR, and JBICR.
Do not write to JBTDR using JTAG/SWD.
Do not write to JBSTR using JTAG/SWD.
Do not write to JBICR using JTAG/SWD.

Note 1. See section 2.9.6.1. MCUSTAT : MCU Status Register.

2.9.6.1 MCUSTAT : MCU Status Register

Address: 0x8000_0400

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	AP[1:0]		—	—	—	—	—	—	—	DSTBY3	DSTBY2	DSTBY1	—	—	STBY2	—
Value after reset:	0	x	0	0	0	0	0	0	0	x	x	x	0	0	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	AL[1:0]		—	—	—	SECEXT	—	—	—	CPU1STOPCLK	CPU1SLEEP	CPU0STOPCLK	CPU0SLEEP	—
Value after reset:	1	0	0	0	0	0	0	1	0	0	0	x	x	x	x	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0.	R
1	CPU0SLEEP	Indicates CPU0 is in Sleep mode. 0: CPU0 is not in Sleep mode 1: CPU0 is in Sleep mode	R

Bit	Symbol	Function	R/W
2	CPU0STOPCLK	0: CPU0 clock is not stopped. It indicates that the CPU0 is in Normal or Sleep mode. 1: CPU0 clock is stopped. It indicates that the CPU0 is in Deep Sleep, Or the MCU is in Software Standby mode or Deep Software Standby power mode.	R
3	CPU1SLEEP	Indicates CPU1 is in Sleep mode. 0: CPU1 is not in Sleep mode 1: CPU1 is in Sleep mode	R
4	CPU1STOPCLK	0: CPU1 clock is not stopped. It indicates CPU1 is in Normal or Sleep mode. 1: CPU1 clock is stopped. It indicates CPU1 is in Deep Sleep, or the MCU is in Software Standby mode or Deep Software Standby power mode.	R
7:5	—	These bits are read as 0.	R
8	SECEXT	This bit is read as 1. 0: Secure debug is not supported 1: Secure debug is supported	R
11:9	—	These bits are read as 0.	R
13:12	AL[1:0]	Authentication Level monitor 0x0: AL0 0x1: AL1 0x3: AL2	R
14	—	This bit is read as 0.	R
15	—	This bit is read as 1.	R
16	—	The read value is undefined.	R
17	STBY2	Indicates the MCU is in Software Standby mode. 0: MCU is not in Software Standby mode 1: MCU is in Software Standby mode	R
19:18	—	These bits are read as 0.	R
20	DSTBY1	Indicates the MCU is in Deep Software Standby 1 mode. 0: MCU is not in Deep Software Standby 1 mode 1: MCU is in Deep Software Standby 1 mode	R
21	DSTBY2	Indicates the MCU is in Deep Software Standby 2 mode. 0: MCU is not in Deep Software Standby 2 mode 1: MCU is in Deep Software Standby 2 mode	R
22	DSTBY3	Indicates the MCU is in Deep Software Standby 3 mode. 0: MCU is not in Deep Software Standby 3 mode 1: MCU is in Deep Software Standby 3 mode	R
29:23	—	These bits are read as 0.	R
31:30	AP[1:0]	Indicates which access ports are available. An external host must check AP[1:0] and must not access AP that is unavailable because unavailable AP cannot respond. 0 0: AP0 and AP1 are available 0 1: AP0, AP1 and AP2 are available	R

Note: MCUSTAT register address is independent of OCDREG space. Only OCD emulator can access this register.

The MCUSTAT register indicates MCU status and authentication status. All bits in the register are monitoring bits so cannot be reset. The register has no security protection.

2.9.6.2 MCUERRSTAT : MCU Error Status Register

Base address: CPU_OCD = 0x4001_1000
CPU_OCD_NS = 0x5001_1000

Offset address: 0x000

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

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

Bit	Symbol	Function	R/W
0	ZERO	Zeroization status flag 0: Zeroization caused by tamper detection has not been performed 1: Zeroization caused by tamper detection has been performed	R
31:1	—	These bits are read as 0.	R

Note: S-TYPE-5
P-TYPE-2

The MCUERRSTAT register indicates the error status of the MCU caused by tamper attack. All bits in the register are monitoring bits so cannot be reset. The register has no security protection.

2.9.6.3 MCUCTRL : MCU Control Register

Base address: CPU_OCD = 0x4001_1000
CPU_OCD_NS = 0x5001_1000

Offset address: 0x004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUWAIT1	CPUWAIT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EDBGRQ1	EDBGRQ0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EDBGRQ0	External Debug Request for CPU0. Writing 1 to the bit causes CPU Halt or Debug Monitor exception request. 0: Not request debug event 1: Request debug event	R/W
1	EDBGRQ1	External Debug Request for CPU1. Writing 1 to the bit causes CPU Halt or Debug Monitor exception request. 0: Not request debug event 1: Request debug event	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
16	CPUWAIT0	CPU0 WAIT SETTING. Write 1 to assert CPUWAIT0, write 0 to deassert CPUWAIT0. 0: Clear CPUWAIT0 to low 1: Set CPUWAIT0 to high	R/W

Bit	Symbol	Function	R/W
17	CPUWAIT1	CPU1 WAIT SETTING. Write 1 to assert CPUWAIT1, write 0 to deassert CPUWAIT1. 0: Clear CPUWAIT1 to low 1: Set CPUWAIT1 to high	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note: This register is valid only when any of the following conditions are true:

- CDBGPWRUPREQ is 1, and AL is AL2 or AL1
- OFS1.SWDBG is 0, and DBGAUTH0.DBGEN0 is 1
- OFS1.SWDBG is 0, and DBGAUTH0.DBGEN1 is 1.
  When none of the above conditions are true, the MCU ignores this register setting and regards all bits as 0.

Note: S-TYPE-5
P-TYPE-2

The MCUCTRL register provides external debugger ability to wake up system from low power state, and ability to enter debug state by external event.

2.9.6.4 Authentication Control Registers

Authentication registers accessible from the debugger are placed in OCDREG.

2.9.6.4.1 JBMDR : JTAG Boot Mode Entry Register

Base address: CPU_OCD = 0x4001_1000
CPU_OCD_NS = 0x5001_1000

Offset address: 0x100

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

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

Bit	Symbol	Function	R/W
7:0	KEY	Mode entry key Pin reset releases after 0xA5 is set, then system enters the JTAG boot mode. MDSR.JBOTS = 1 when system transitions to JTAG boot mode. Setting this value does not have any effect when debugger is not connected.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5
P-TYPE-2

The JBMDR register sets command from the debugger.

This register is initialized by the debugger disconnection.

2.9.6.4.2 JBRDR : JTAG Boot Receive Data Register

Base address: CPU_OCD = 0x4001_1000
CPU_OCD_NS = 0x5001_1000

Offset address: 0x120

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

Bit	Symbol	Function	R/W
31:0	RDAT[31:0]	Received data register R/W is possible for both the external host and CPU, but the following usage is recommended: W: External host R: CPU (Boot Firmware) When JBSTR.RDF = 1, write is not possible and an error occurs.	R/W

Note: S-TYPE-5
P-TYPE-2

The JBRDR register is used for receiving data from the debugger.

2.9.6.4.3 JBTDR : JTAG Boot Transmit Data Register

Base address: CPU_OCD = 0x4001_1000
CPU_OCD_NS = 0x5001_1000

Offset address: 0x130

Bit position:	31 … 0
Bit field:	TDAT[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TDAT[31:0]	Transmitted data register R/W is possible for both the external host and CPU, but the following usage is recommended: W: CPU (Boot Firmware) R: External host When JBSTR.TDE = 1, read is not possible and an error occurs.	R/W

Note: S-TYPE-5
P-TYPE-2

The JBTDR register is used for transmitting data from the debugger.

2.9.6.4.4 JBSTR : JTAG Boot Status Register

Base address: CPU_OCD = 0x4001_1000
CPU_OCD_NS = 0x5001_1000

Offset address: 0x140

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

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

Bit	Symbol	Function	R/W
0	RDF	Receive buffer full [Set conditions] • Write access to JBRDR • Write 1 to JBSTR.RDF [Clear conditions] • Read access to JBRDR • Write 0 to JBSTR.RDF 0: No receiving data 1: There is receiving data	R/W

Bit	Symbol	Function	R/W
1	TDE	Transmit data empty [Set conditions] ● Read access to JBTDR ● Write 1 to JBSTR.TDE [Clear conditions] ● Write access to JBTDR ● Write 0 to JBSTR.TDE 0: There is data transmission 1: No data transmission	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5
P-TYPE-2

The JBSTR register is used for monitoring boot status.

2.9.6.4.5 JBICR : JTAG Boot Interrupt Control Register

Base address: CPU_OCD = 0x4001_1000
CPU_OCD_NS = 0x5001_1000

Offset address: 0x150

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

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

Bit	Symbol	Function	R/W
0	RDFIE	Receive buffer full interrupt enabled. 0: Interrupt request disabled by RDF = 1 1: Enable interrupt request by RDF = 1	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5
P-TYPE-2

The JBICR register is used for controlling interrupt during JTAG boot.

2.9.6.4.6 FSBLSTATM : First Stage Boot Loader Status Monitor Register

Base address: CPU_OCD = 0x4001_1000
CPU_OCD_NS = 0x5001_1000

Offset address: 0x300

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

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

Bit	Symbol	Function	R/W
0	CS	FSBL completion status 0: FSBL is not complete 1: FSBL is complete	R
1	RS	FSBL result status 0: FSBL failed 1: FSBL passed	R
31:2	—	These bits are read as 0.	R

Note: S-TYPE-5
P-TYPE-2

The FSBLSTATM register is mirror of the FSBLSTAT register.

2.9.6.5 CoreSight Component Registers

OCDREG has CoreSight Component registers defined in ARM® CoreSight™ Architecture. Table 2.29 shows a list of OCDREG CoreSight Component registers.

See reference [6] ARM Limited., Arm(R) CoreSight Architecture Specification (ARM IHI 0029D) for the detail of each register.

Table 2.29 CoreSight Component registers of OCDREG

Name	Address	Access size	R/W	Initial value
PID4	0x8001_1FD0	32-bit	Read-only	0x00000004
PID5	0x8001_1FD4	32-bit	Read-only	0x00000000
PID6	0x8001_1FD8	32-bit	Read-only	0x00000000
PID7	0x8001_1FDC	32-bit	Read-only	0x00000000
PID0	0x8001_1FE0	32-bit	Read-only	0x00000004
PID1	0x8001_1FE4	32-bit	Read-only	0x00000030
PID2	0x8001_1FE8	32-bit	Read-only	0x0000000A
PID3	0x8001_1FEC	32-bit	Read-only	0x00000000
CID0	0x8001_1FF0	32-bit	Read-only	0x0000000D
CID1	0x8001_1FF4	32-bit	Read-only	0x000000F0
CID2	0x8001_1FF8	32-bit	Read-only	0x00000005
CID3	0x8001_1FFC	32-bit	Read-only	0x000000B1

Note: S-TYPE-5
P-TYPE-2

2.10 CoreSight Cross Trigger Interface

Table 2.30 Cross Trigger Interface (1 of 2)

Number of CTI channel	CTITRIGIN		CTITRIGOUT	
CTI	0	ETF ACQCOMP	0	TPIU FLUSHIN
	1	ETF FULL	1	TPIU TRIGIN
	2	—	2	ETF FLUSHIN
	3	—	3	ETF TRIGIN
	4	—	4	—
	5	—	5	—
	6	—	6	—
	7	—	7	—

Table 2.30 Cross Trigger Interface (2 of 2)

Number of CTI channel	CTITRIGIN		CTITRIGOUT	
CTI0	0	Processor Halted	0	Processor debug request
	1	DWT Comparator Output 0	1	Processor Restart
	2	DWT Comparator Output 1	2	CTIIRQ[0] (Connected to ICU0)
	3	DWT Comparator Output 2	3	CTIIRQ[1] (Connected to ICU0)
	4	ETM Event Output 0	4	ETM Event Input 0
	5	ETM Event Output 1	5	ETM Event Input 1
	6	—	6	ETM Event Input 2
	7	—	7	ETM Event Input 3
CTI1	0	Processor Halted	0	Processor debug request
	1	DWT Comparator Output 0	1	Processor Restart
	2	DWT Comparator Output 1	2	CTIIRQ[0] (Connected to ICU1)
	3	DWT Comparator Output 2	3	CTIIRQ[1] (Connected to ICU1)
	4	ETM Event Output 0	4	ETM Event Input 0
	5	ETM Event Output 1	5	ETM Event Input 1
	6	—	6	ETM Event Input 2
	7	—	7	ETM Event Input 3

The MCU supports processor CTI0/CTI1 and CoreSight Common CTI which is implemented in Debug module. The ICU setting is necessary when CTIIRQ is used.

2.11 CoreSight ATB Funnel

There is one CoreSight ATB Funnel in the MCU. The funnel has four ATB slaves and one ATB master, and it is used to select the debug trace sources from ETMn and ITMn to ETF. Figure 2.3 shows the CoreSight ATB connection in the MCU (n = 0, 1).

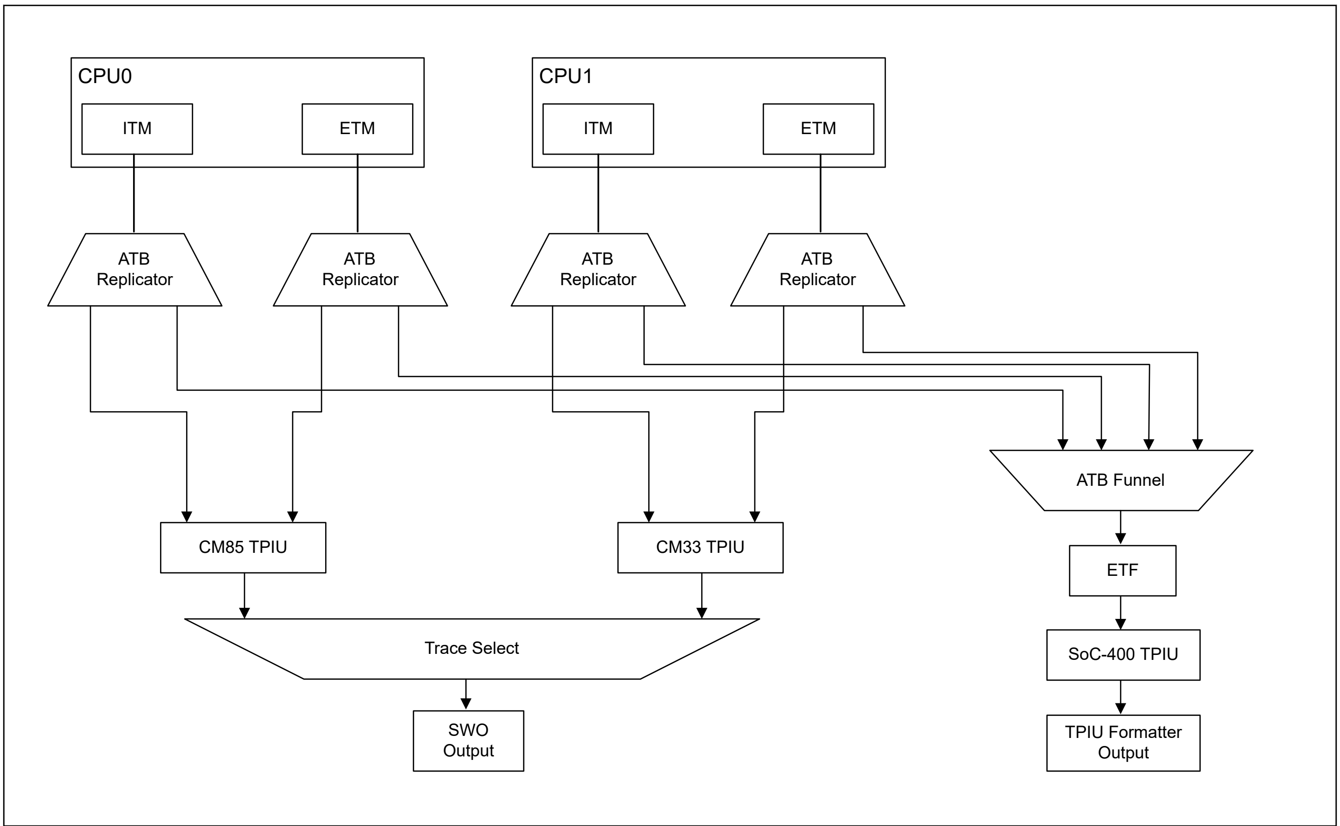


Figure 2.3 CoreSight ATB connection

Table 2.31 shows funnel ATB slave connection.

Table 2.31 ATB slave connection

ATB Slave No.	Connected Trace Source
#0	CPU0-ITM
#1	CPU0-ETM
#2	CPU1-ITM
#3	CPU1-ETM

See reference [4] ARM Limited., Arm(R) CoreSight SoC-400 Technical Reference Manual (ARM 0302-01 2nd release for r3p2) for the detail of ATB and Funnel.

2.12 SysTick System Timer

The MCU has two SysTick system timers for each CPUn (n = 0, 1), secure SysTick and non-secure SysTick.

The SysTick source is CPUCLKn or 1 MHz clock generated from the on-chip oscillator. When using 1 MHz clock generated from the on-chip oscillator for the SysTick source, the CPUCLKn frequency must be 2 MHz or higher (n = 0, 1).

2.13 CoreSight Timestamp Generator

The MCU has a CoreSight Timestamp Generator to provide timestamp to ITM and ETM for each CPUn (n = 0, 1).

The timestamp is generated by a 64-bit counter operating.

See reference [4] ARM Limited., Arm(R) CoreSight SoC-400 Technical Reference Manual (ARM 0302-01 2nd release for r3p2) for the detail.

2.14 OCD Emulator Connection

In this product, the MCU confirms access permission for MCU resources by checking current Authentication Level stored in OCDREG. There are two ways to control Authentication Level, through:

- JTAG/SWD. See Figure 2.4.
- Software. See Figure 2.5.

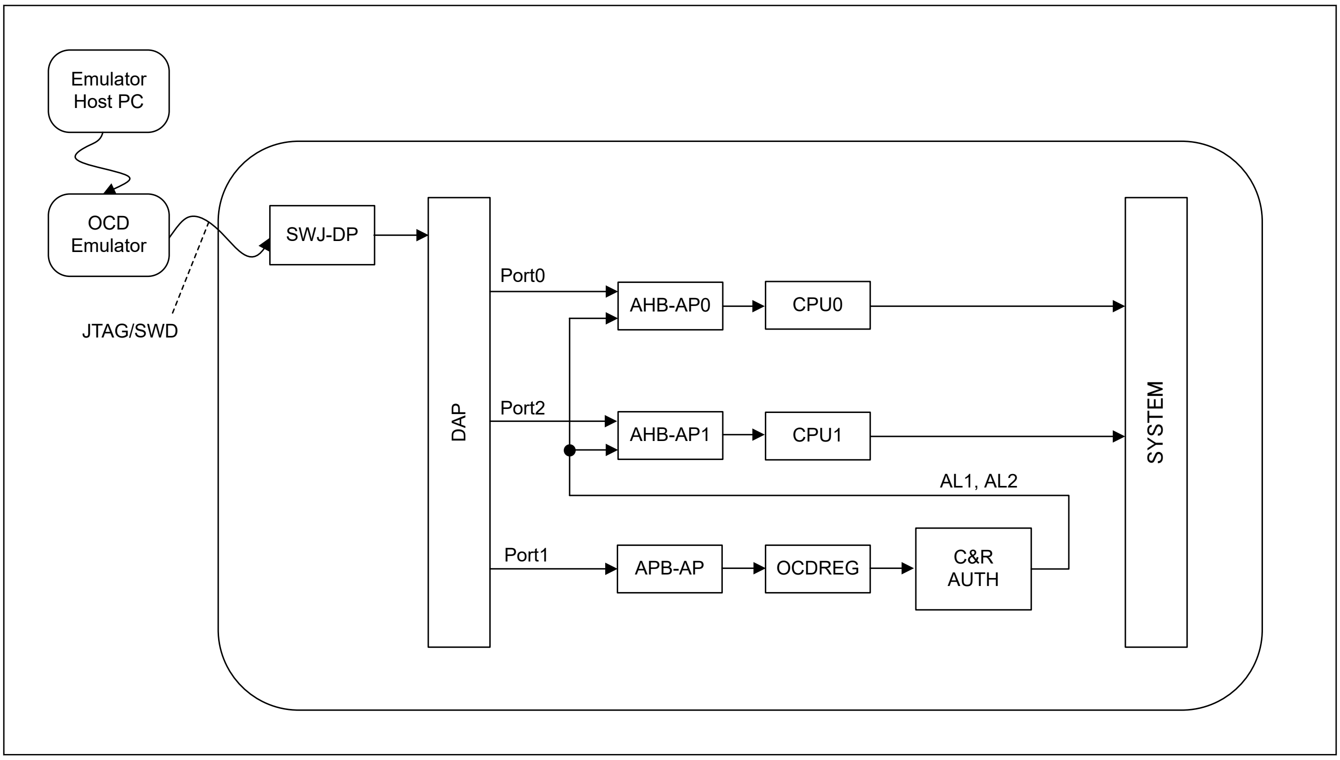


Figure 2.4 Using JTAG/SWD to control authentication level

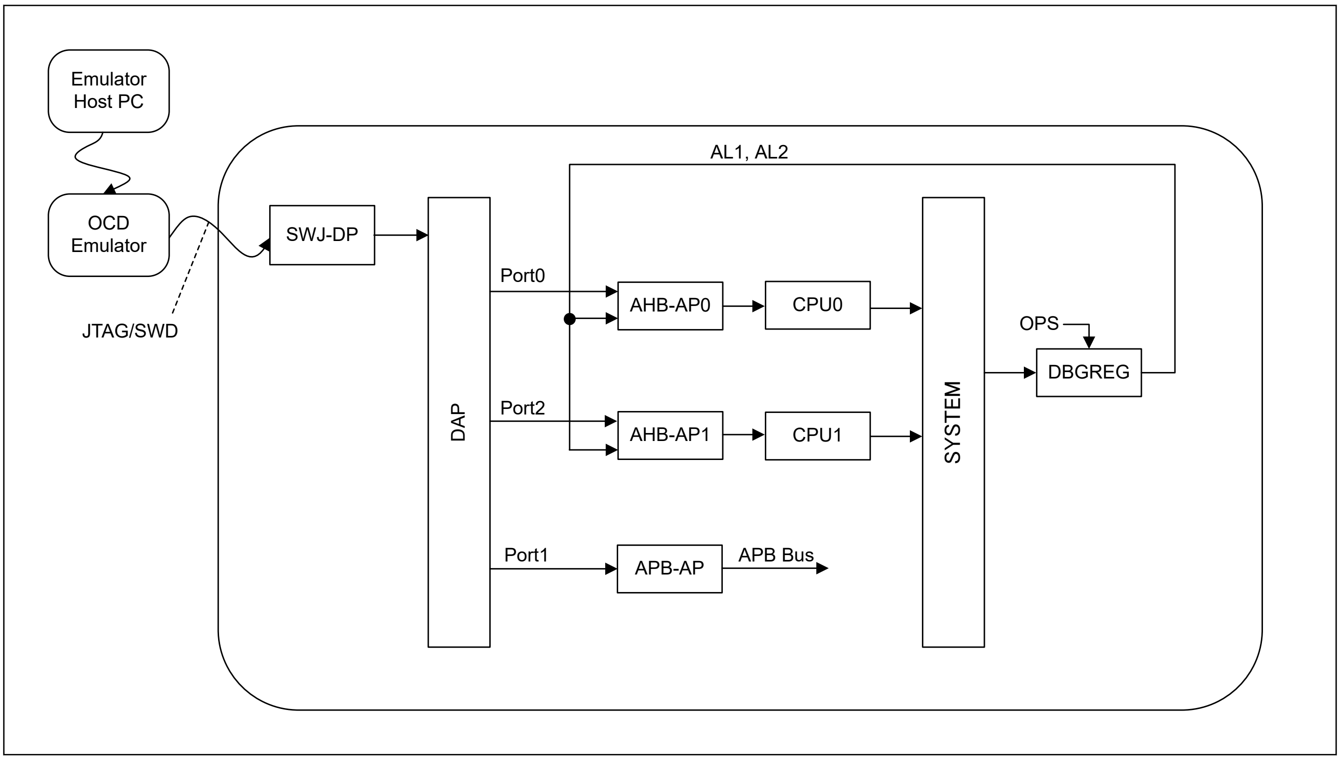


Figure 2.5 Using software to control authentication level

2.14.1 Restrictions in OCD Emulator Connecting

There are some restrictions regarding emulator access. This section describes the restrictions.

2.14.1.1 Starting Connection While in Low Power Mode

When starting a JTAG/SWD connection from an OCD emulator, the MCU must be in Normal, Sleep mode or Deep Sleep mode. If the MCU is in Software Standby or Deep Software Standby mode, the OCD emulator can cause the MCU to hang.

2.14.1.2 Low Power Mode Change During Debugging

When the MCU is in OCD mode, the low power mode can be changed. However, system bus access from AHB-APn (n = 0, 1) is prohibited in Deep Sleep, Software Standby, Deep Software Standby mode. Only SWJ-DP, APB-AP and OCDREG can be accessed from OCD emulator in these modes. Table 2.32 shows the limitations.

Table 2.32 Limitations of SWJ-DP, APB-AP and OCDREG in low power mode

Current mode	Start OCD emulator connection	Change low power mode	Access AHB-APn (n = 0, 1) and system bus	Access APB-AP and OCDREG
Normal	Available	Available	Available	Available
Sleep	Available	Available	Available	Available
Deep Sleep	Available	Available	N/A	Available
Software Standby	N/A	Available	N/A	Available
Deep Software Standby	N/A	Available	N/A	Available

2.14.2 Connecting Sequence and JTAG/SWD Authentication

This section describes how to authenticate the MCU. section 2.14.2.1. JTAG/SWD Authentication Mechanism introduces which mechanism is used for authentication, and section 2.14.2.2. Connecting Sequence and JTAG/SWD Authentication provides steps for connecting sequence and authenticating the MCU using JTAG/SWD. See section 2.9.5. DBGREG and section 2.9.6. OCDREG for details about the registers control JTAG/SWD authentication.

2.14.2.1 JTAG/SWD Authentication Mechanism

The MCU supports challenge-response authentication using the boot firmware.

2.14.2.1.1 JTAG Boot Mode and Authentication Request

JTAG boot mode request is asserted if the debugger sets 0xA5 command to JBMDR while RES pin is asserted and MD pin is high when RES pin is deasserted. After entered JTAG boot mode, the debugger can issue authentication procedure by sending Authentication command to JBRDR register.

2.14.2.1.2 Auth Circuit

Auth circuit updates initial Authentication Level (AL) by Protection Level (PL).

When challenge-response authentication is complete, the authentication level is updated by the boot firmware

(1) Initial Authentication Level Update

Auth circuit updates initial authentication level by protection level.

Table 2.33 shows auth circuit update of initial authentication level.

Other than OEM, the initial authentication level is not dependent on the protection level.

Table 2.33 Auth circuit update of initial authentication level (1 of 2)

Input		Output
Lifecycle	Protection level	Authentication level
OEM	PL0	AL0
	PL1	AL1
	PL2	AL2
RMA_REQ	*	AL0

Table 2.33 Auth circuit update of initial authentication level (2 of 2)

Input		Output
Lifecycle	Protection level	Authentication level
RMA_ACK	*	AL2
RMA_RET	*	AL0
LCK_BOOT	*	AL0

(2) Authentication at Challenge-response Authentication

The boot firmware authenticates and writes authentication result to authentication control registers.

2.14.2.2 Connecting Sequence and JTAG/SWD Authentication

Challenge-response authentication

1. Assert RES pin by connecting the OCD debugger to the MCU through the JTAG or SWD interface.
2. Set up SWJ-DP to access DAP bus.
   In the setup, the OCD emulator must assert CDBGPWRUPREQ in the SWJDP Control Status Register (CTRL/STAT), and then wait until CDBGPWRUPACK in the same register is asserted.
3. Set the APB-AP to access OCDREG. This APB-AP is connected to DAP bus port 1.
4. Set boot mode request command (0xA5) to JBMDR and set MD pin to 1.
5. Negate RES pin.
6. Check MCUSTAT.AL:
   - If MCUSTAT.AL is higher than or equal to request AL, there is no need to authenticate. Go to step 11..
7. Set Authentication command to JBRDR.
8. Read challenge value from JBTDR.
9. Set the response data to JBRDR.
10. Read JBTDR and check MCUSTAT.AL.
    - If MCUSTAT.AL is not the same as the requested AL, authentication failed
    - If retry authentication: go to step 7..
11. Set up the AHB-APn to access the system address space. The AHB-APn is connected to DAP bus port 0, 2 (n = 0, 1).
12. Start accessing the CPU debug resources using the AHB-APn (n = 0, 1).

For CPU individual reset, FSBL is never executed.

For FSBL execution, IWDT and WDT0/1 function as the settings for OEM selected mode.

2.15 References

[1] ARM Limited., ARMv8-M Architecture Reference Manual (ARM DDI 0553B)

[2] ARM Limited., Arm® Cortex-M85 Processor Technical Reference Manual (ARM 101924)

[3] ARM Limited., Arm® Cortex-M85 Processor User Guide (ARM 101927)

[4] ARM Limited., Arm® CoreSight SoC-400 Technical Reference Manual (ARM 0302-01 2nd release for r3p2)

[5] ARM Limited., Arm® CoreSight ETM-M85 Technical Reference Manual (ARM 101926)

[6] ARM Limited., Arm® CoreSight Architecture Specification (ARM IHI 0029D)

[7] ARM Limited., Arm® Embedded Trace Macrocell Architecture Specification (ARM IHI 0064H.a)

[8] ARM Limited., Arm® CoreSight SoC-400 User Guide (ARM 100490)

[9] ARM Limited., Arm® Cortex-M33 Processor Technical Reference Manual (ARM 100230)

[10] ARM Limited., Arm® Cortex-M33 Processor User Guide (ARM 100234)

[11] ARM Limited., Arm® CoreSight ETM-M33 Technical Reference Manual (ARM 100232)

[12] ARM Limited., Arm® CoreSight Trace Memory Controller Technical Reference Manual (ARM DDI0461B)

2.16 CM33 CACHE (C-Cache and S-Cache)

2.16.1 Overview

There are two types of caches:

- C-Cache on Cortex-M33 Code AHB interface (C-AHB)
- S-Cache on Cortex-M33 System AHB interface (S-AHB).

2.16.1.1 Specifications of Cache

Table 2.34 Specifications of cache

Item	C-Cache	S-Cache
Capacity	16 KB	16 KB
Way	4-way set associative	4-way set associative
Line size	256 bits	256 bits
Number of entry	128 entries/ways (4-way set associative)	128 entries/ways (4-way set associative)
Write way	Write-through/write-back Non write-allocate/write-allocate	Write-through/write-back Non write-allocate/write-allocate
Replace way	LRU	LRU
Error detection	ECC Data memory: SECDED Tag memory: SEDDED	ECC Data memory: SECDED Tag memory: SEDDED
Cache support area	0x0000_0000 – 0x1FFF_FFFF	0x2000_0000 – 0xDFFF_FFFF

Note: Transient, non-transient and shareability of memory attributes (set by Arm MPU) do not affect cache behavior.

2.16.1.2 Block Diagram

Figure 2.6 shows the block structure of cache.

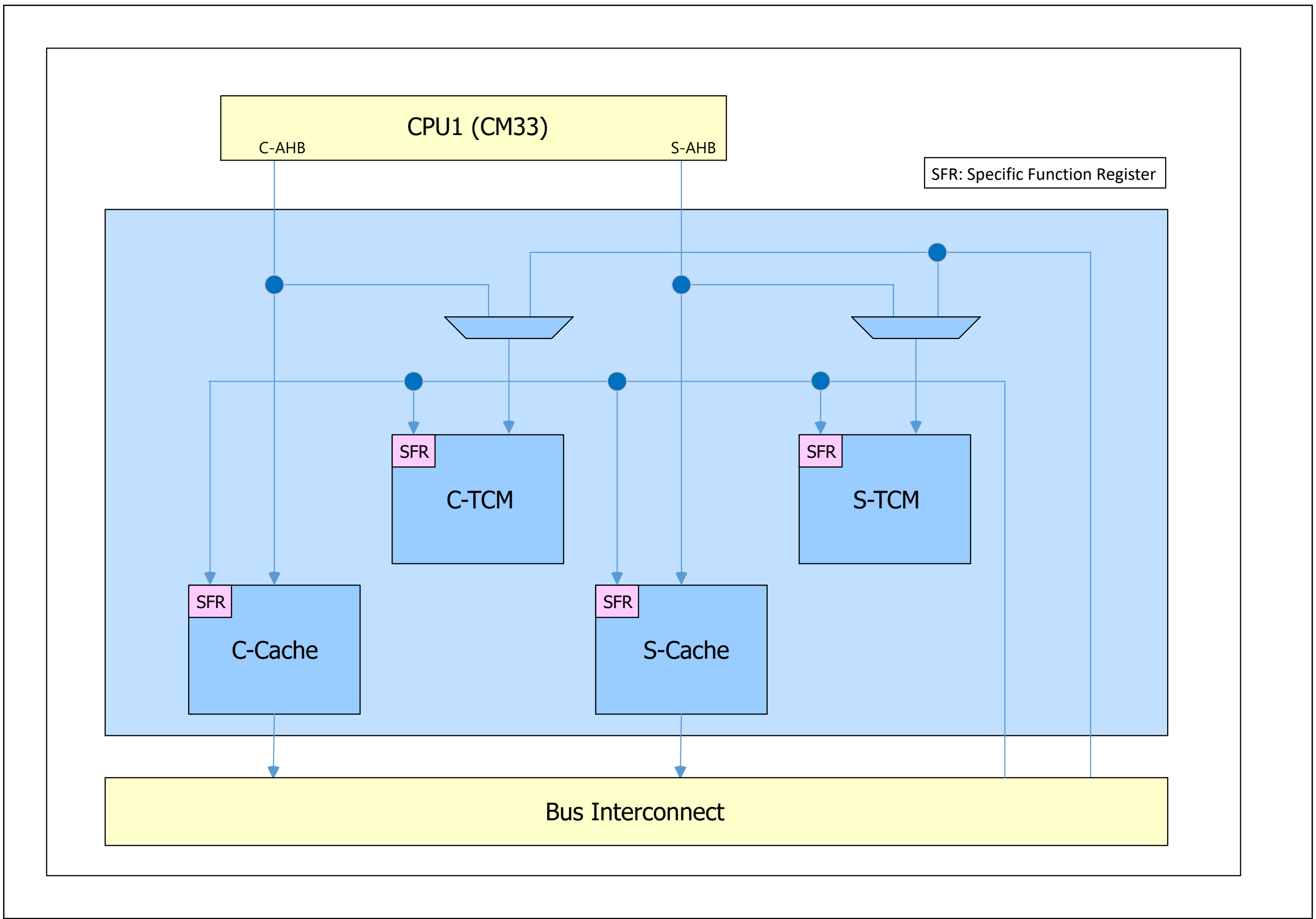


Figure 2.6 Cache block structure

Figure 2.7 shows the detailed cache structure.

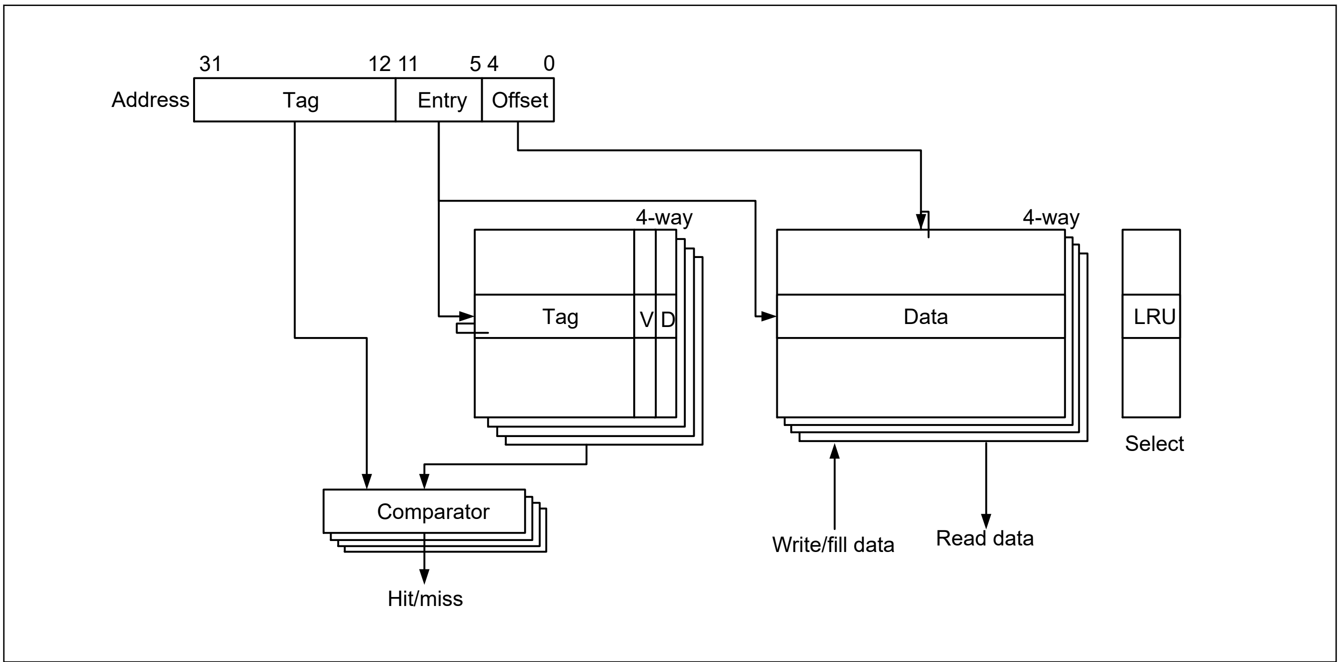


Figure 2.7 Detailed cache structure

Tag

Tag holds the cached address [31:12]. Tag also holds the ECC code for 32 bits with zero-extended address[31:12] and valid bit V.

Tag is cleared by the cache flush function.

V

V is a valid bit and exists in each cache line. It shows the validated data is stored in the cache line.

V bit is cleared by the cache flush function.

D

D is a dirty bit and exists in each cache line. It shows the validated data stored in the cache line is changed from the original value, and requires to be written back to bus.

Data

Data holds 256-bit data in each cache line.

LRU

LRU (Least Recently Used) is a way to replace each cache line.

Comparator

Comparator determines whether the data to be accessed is stored in the cache line.

If V is set and the address equals to the address stored in the tag, the cache determines the access is a hit. Otherwise, it is a miss. If there is an ECC error in the Hit tag memory line, it is treated as a miss.

2.16.2 Register List

Cache has registers which control the cache functions. Table 2.35 shows all the cache related registers.

Table 2.35 Register list

Register name	Symbol	Address (for secure/non-secure access)
Cache Security Attribution Register	CACHESAR	CPSCU_BASE + 0x500
C-Cache Control Register	CCACTL	CACHE_BASE + 0x000
C-Cache Flush Control Register	CCAFCT	CACHE_BASE + 0x004
C-Cache Write Attribute	CCAWTA	CACHE_BASE + 0x00C
C-Cache Error Detection Status	CCAEDST	CACHE_BASE + 0x010
C-Cache Test Access Address	CCATAA	CACHE_BASE + 0x014
C-Cache Test Access Data	CCATAD	CACHE_BASE + 0x018
S-Cache Control Register	SCACTL	CACHE_BASE + 0x040
S-Cache Flush Control Register	SCAFCT	CACHE_BASE + 0x044
S-Cache Write Attribute	SCAWTA	CACHE_BASE + 0x04C
S-Cache Error Detection Status	SCAEDST	CACHE_BASE + 0x050
S-Cache Test Access Address	SCATAA	CACHE_BASE + 0x054
S-Cache Test Access Data	SCATAD	CACHE_BASE + 0x058
Cache ECC Error Operation After Detection Register	CAPOAD	CACHE_BASE + 0x200
Cache Protection Register	CAPRCR	CACHE_BASE + 0x204

CPSCU_BASE (Base address of CPSCU) is 0x4000_8000 (Secure) or 0x5000_8000 (Non-secure).

CACHE_BASE (Base address of Cache) is 0x4001_C000 (Secure) or 0x5001_C000 (Non-secure).

2.16.3 Register Descriptions

2.16.3.1 CACHESAR : Cache Security Attribution Register

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x500

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

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

Bit	Symbol	Function	R/W
0	CACHESA	Security attributes of registers for cache control 0: Secure 1: Non-Secure	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	CACHEESA	Security attributes of registers for cache error 0: Secure 1: Non-Secure	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: Security: S-TYPE-1
Privilege: P-TYPE-1
This register is write-protected by PRCR_S.PRC4.

Only Secure access can write to this register. Non-secure read access is allowed, non-secure write access is denied, and TrustZone access error is not generated.

CACHESA bit (Security attributes of registers for cache control)

Security attributes of registers for cache control. The target registers are as follows.

- CCACTL
- CCAWTA
- CCAEDST
- SCACTL
- SCAWTA
- SCAEDST

CACHEESA bit (Security attributes of registers for cache error)

Security attributes of registers for cache error. The target registers are as follows.

- CAPOAD
- CAPRCR

CACHESA and CACHEESA must be set to 0 when secure program/secure data is cacheable.

2.16.3.2 CCACTL : C-Cache Control Register

Base address: CACHE = 0x4001_C000
CACHE_NS = 0x5001_C000

Offset address: 0x000

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

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

Bit	Symbol	Function	R/W
0	ENC	C-cache enable bit Set C-Cache enable. 0: C-Cache disable 1: C-Cache enable	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	FC	C-Cache flush bit Set C-Cache line flush. 0: No action 1: C-Cache line flush (all lines invalidate)	R/W
9	WB	C-cache write-back Trigger to write back C-Cache. 0: No action 1: C-Cache write-back	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: Security: S-TYPE-3 (CACHESAR.CACHESA)
Privilege: P-TYPE-2

ENC bit (C-cache enable bit)

The ENC bit controls the cache enable of C-Cache.

When C-Cache set to enable, whether it caches data or not depends on the access attribute setting for cacheable of CPU1.

FC bit (C-Cache flush bit)

The FC bit is an alias for CCAFCT.FC. See section 2.16.3.3. CCAFCT : C-Cache Flush Control Register for details.

WB bit (C-cache write-back)

The WB bit is an alias for CCAFCT.WB. See section 2.16.3.3. CCAFCT : C-Cache Flush Control Register for details.

2.16.3.3 CCAFCT : C-Cache Flush Control Register

Base address: CACHE = 0x4001_C000
CACHE_NS = 0x5001_C000

Offset address: 0x004

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

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

Bit	Symbol	Function	R/W
0	FC	C-Cache flush bit Set C-Cache line flush. 0: No action 1: C-Cache line flush (all lines invalidate)	R/W
1	WB	C-Cache write-back Trigger to write back C-Cache. 0: No action 1: C-Cache write-back	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: Security: S-TYPE-5
Privilege: P-TYPE-2

FC bit (C-Cache flush bit)

The FC bit controls the cache flush of C-Cache.

[Clearing condition]

- This bit is cleared automatically when a cache flush is completed.

[Setting condition]

- Writing 1 to this bit.

WB bit (C-Cache write-back)

The WB bit triggers a write-back of C-Cache.

[Clearing condition]

- This bit is cleared automatically when write back is completed.

[Setting condition]

- Writing 1 to this bit.

Note: Setting FC to 1 when CCAWTA.WT is 0 performs both write back and flush.

If flush or write back has already started (CCAFCT.FC = 1 or CCAFCT.WB = 1), CCAFCT and CCACTL cannot be written until CCAFCT.FC and CCAFCT.WB are cleared automatically, that is, the process completes.

Cache is disabled after reset.

Cache does not perform write back and flush cache memory automatically when cache is disabled or reset.

If there is a dirty line left in the cache while the cache is disabled, inconsistencies can occur between cache and memory.

To disable the cache, use the following procedure.

1. Write 0x0100 (if CCAWTA.WT = 1) or 0x0300 (if CCAWTA.WT = 0) to CCACTL.

2. Make sure that CCAFCT.FC and CCAFCT.WB are both 0.

To enable the cache after reset, use the following procedure.

1. Set CCAFCT.FC to 1 to flush cache memory.
2. Wait for the flush to complete until CCAFCT.FC is 0.
3. Set CCACTL.ENC to 1 to enable the cache.
4. Writing to CCACTL or CCAFCT after reset automatically flushes the cache.

To enable the cache after cache is disabled, use the following procedure.

1. Make sure CCAFCT.WB and CCAFCT.FC is 0, that is, the cache does not perform write back or flush.
2. Set CCACTL.ENC to 1 to enable the cache.

2.16.3.4 CCAWTA : C-Cache Write Attribute

Base address: CACHE = 0x4001_C000
CACHE_NS = 0x5001_C000

Offset address: 0x00C

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

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

Bit	Symbol	Function	R/W
0	WT	C-Cache write-through Setting the attribute of C-Cache for write-through or write-back. 0: Write-through or write-back depends on the cache attribute of the area 1: Always write-through mode	R/W
1	WA	C-Cache write-allocation Setting the attribute of C-Cache for write-allocation. 0: Always disable write-allocation 1: Write-allocation enable depends on the cache attribute of the area	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: Security: S-TYPE-3 (CACHESAR.CACHESA)
Privilege: P-TYPE-2

WT bit (C-Cache write-through)

The WT bit specifies the write attribute of C-Cache for write-through or write-back.

C-Cache must be disabled (CCACTL.ENC = 0) when changing this bit value.

When WT is 0, the cache attribute is determined by the default memory map or MPU of the Cortex-M33 processor.

WA bit (C-Cache write-allocation)

The WA bit enables C-Cache write-allocation.

C-Cache must be disabled (CCACTL.ENC = 0) when changing this bit value.

2.16.3.5 CCAEDST : C-Cache Error Detection Status

Base address: CACHE = 0x4001_C000
CACHE_NS = 0x5001_C000

Offset address: 0x010

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	EST2	ESTD	ESTC	ESD1	ESD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ESD0	C-Cache data error status 0 Indicates status of whether ECC 1-bit error was detected when reading C-Cache data memory. Write value must be 0. Writing 0 clears this status. 0: ECC 1-bit error was not detected 1: ECC 1-bit error was detected and corrected	R/W
1	ESD1	C-Cache data error status 1 Indicates status of whether ECC 2-bit error was detected when reading C-Cache data memory. Write value must be 0. Writing 0 clears this status. 0: ECC 2-bit error was not detected 1: ECC 2-bit error was detected	R/W
2	ESTC	C-Cache tag clean line invalidate status Indicates status of whether clean line was invalidated by ECC 1-bit error of C-Cache tag memory. Write value must be 0. Writing 0 clears this status. 0: Clean line was not invalidated 1: Clean line was invalidated	R/W
3	ESTD	C-Cache tag dirty line invalidate status Indicates status of whether dirty line was invalidated by ECC 1-bit error of C-Cache tag memory. Write value must be 0. Writing 0 clears this status. 0: Dirty line was not invalidated 1: Dirty line was invalidated	R/W
4	EST2	C-Cache tag 2-bit error status Indicates status of whether ECC 2-bit error was detected when reading C-Cache tag memory. Write value must be 0. Writing 0 clears this status. 0: ECC 2-bit error was not detected 1: ECC 2-bit error was detected	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: Security: S-TYPE-3 (CACHESAR.CACHESA)
Privilege: P-TYPE-2

ESD0, ESD1 bit

These bits indicate the status of whether ECC errors were detected when reading C-Cache data memory.

Writing 0 clears this status. Resets other than Bus Error Reset or Memory Error Reset clear this status. If an error occurs and the status is cleared at the same time, the status is cleared, and the error is ignored.

ESTC, ESTD, EST2 bit

These bits indicate the status of whether uncorrectable ECC errors were detected when reading C-Cache tag memory.

ESTC : Invalidate clean line by ECC 1-bit error (correctable)

ESTD : invalidate dirty line by ECC 1-bit error (uncorrectable)

EST2 : Detect ECC 2-bit error (uncorrectable)

Writing 0 clears this status. If an error occurs and the status is cleared at the same time, the status is cleared, and the error is ignored.

If ESTD is 1, dirty line is invalidated without write back.

2.16.3.6 CCATAA : C-Cache Test Access Address

Base address: CACHE = 0x4001_C000
CACHE_NS = 0x5001_C000

Offset address: 0x014

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

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

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
4:2	OFFSET[2:0]	Address offset C-Cache test access address lower 3 bits [4:2] corresponds to upper 3 bits of 5-bit offset [4:0]. These bits are valid when cache data read/write and ECC code read/write.	R/W
11:5	ENTRY[6:0]	Address entry C-Cache test access address middle 7 bits [11:5] corresponds to entry.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
18:16	TARGET[2:0]	Access target These bits specify the C-Cache test access target. Access type is specified by bit RW[23]. 0 0 0: Cache data read or write 0 0 1: Data ECC code read or write 0 1 0: Tag, V, D read or write 0 1 1: LRU read or write 1 0 0: Tag ECC code read or write 1 0 1 - 1 1 1: Setting prohibited	R/W
22:19	—	These bits are read as 0. The write value should be 0.	R/W
23	RW	Read write This bit specifies whether C-Cache test access type is read or write. 0: Read 1: Write	R/W
29:24	—	These bits are read as 0. The write value should be 0.	R/W
31:30	WAY[1:0]	Address way C-Cache test access address higher 2 bits corresponds to way. These bits are valid when cache data/ecc code/tag read/write. These bits are invalid when LRU read/write.	R/W

Note: It is necessary to write by word (32-bit) access. Half word/byte write access is prohibited.
Note: Security: S-TYPE-6
Privilege: P-TYPE-2

WAY[1:0], ENTRY[6:0], OFFSET[2:0] bit

These bits indicate the address for C-Cache test access.

WAY[1:0], ENTRY[6:0], OFFSET[2:0] are valid for ECC code/cache data read or write.

WAY[1:0], ENTRY[6:0] are valid, and OFFSET[2:0] is invalid for Tag and Tag ECC code read or write.

ENTRY[6:0] is valid and WAY[1:0], OFFSET[2:0] are invalid for LRU read or write.

TARGET[2:0] bits (Access target)

These bits specify the C-Cache test access target. The truth table is below.

RW bit (Read write)

The RW bit specifies whether C-Cache test access type is read or write. The truth table is below.

CCATAA.RW[23]	CCATAA.TARGET[18:16]	Command
0	000	Cache data read
1	000	Cache data write
0	001	Data ECC code read
1	001	Data ECC code write
0	010	Tag read
1	010	Tag write
0	011	LRU read
1	011	LRU write
0	100	Tag ECC code read
1	100	Tag ECC code write

Writing to CCATAA register triggers the test access. During test access writing, write to register CCATAD before writing to register CCATAA. During test access reading, read register CCATAD after writing to register CCATAA.

To perform read access, follow these steps:

1. Write to CCATAA with CCATAA.RW = 0.
2. Read CCATAA to check.
3. Read CCATAD.

To perform write access, follow these steps:

1. Write to CCATDA.
2. Write to CCATAA with CCATAA.RW = 1.

If CCATAD register is updated by the test access reading and the CCATAD register is written at the same time, CCATAD register update by the test access reading is ignored and CCATAD register is written.

2.16.3.7 CCATAD_DATA : C-Cache Test Access Data (DATA)

Base address: CACHE = 0x4001_C000
CACHE_NS = 0x5001_C000

Offset address: 0x018

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

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

Bit	Symbol	Function	R/W
31:0	DATA[31:0]	C-Cache test access data These bits are valid when C-Cache test access cache data read/write (CCATAA[18:16] = 000).	R/W

Note: It is necessary to write by word (32-bit) access. Half word/byte write access is prohibited.
Note: Security: S-TYPE-6
Privilege: P-TYPE-2

DATA[31:0] bits

These bits indicate each data when C-Cache test access.

Which bits are valid depends on the type of C-Cache test access.

C-Cache test access should be done with C-Cache turned off (CCACTL.ENC = 0).

If CCATAD register is updated by the test access reading and the CCATAD register is written at the same time, CCATAD register update by the test access reading is ignored and CCATAD register is written.

2.16.3.8 CCATAD_ECC : C-Cache Test Access Data (ECC)

Base address: CACHE = 0x4001_C000
CACHE_NS = 0x5001_C000

Offset address: 0x018

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

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

Bit	Symbol	Function	R/W
6:0	ECC[6:0]	C-Cache test access ECC code These bits are valid when C-Cache test access data ECC code of cache data read/write (CCATAA[18:16] = 001).	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

Note: It is necessary to write by word (32-bit) access. Half word/byte write access is prohibited.
Note: Security: S-TYPE-6
Privilege: P-TYPE-2

ECC[6:0] bits

The ECC[6:0] bits indicate each data when C-Cache test access.

Which bits are valid depends on the type of C-Cache test access.

C-Cache test access should be done with C-Cache turned off (CCACTL.ENC = 0).

If CCATAD register is updated by the test access reading and the CCATAD register is written at the same time, CCATAD register update by the test access reading is ignored and CCATAD register is written.

2.16.3.9 CCATAD_TAG : C-Cache Test Access Data (TAG)

Base address: CACHE = 0x4001_C000
CACHE_NS = 0x5001_C000

Offset address: 0x018

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TAG[19:4]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	D	C-Cache test access Dirty This bit is valid when C-Cache test access Tag, V, D read/write (CCATAA[18:16] = 010).	R/W
1	V	C-Cache test access Valid This bit is valid when C-Cache test access Tag, V, D read/write (CCATAA[18:16] = 010).	R/W
11:2	—	These bits are read as 0. The write value should be 0.	R/W
31:12	TAG[19:0]	C-Cache test access Tag data These bits are valid when C-Cache test access Tag, V, D read/write (CCATAA[18:16] = 010).	R/W

Note: It is necessary to write by word (32-bit) access. Half word/byte write access is prohibited.
Note: Security: S-TYPE-6
Privilege: P-TYPE-2

TAG[19:0], V, D bits

These bits indicate each data when C-Cache test access.

Which bits are valid depends on the type of C-Cache test access.

C-Cache test access should be done with C-Cache turned off (CCACTL.ENC = 0).

If CCATAD register is updated by the test access reading and the CCATAD register is written at the same time, CCATAD register update by the test access reading is ignored and CCATAD register is written.

2.16.3.10 CCATAD_LRU : C-Cache Test Access Data (LRU)

Base address: CACHE = 0x4001_C000
CACHE_NS = 0x5001_C000

Offset address: 0x018

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

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

Bit	Symbol	Function	R/W
4:0	LRU[4:0]	C-Cache test access LRU data These bits are valid when C-Cache test access LRU read/write (CCATAA[18:16] = 011).	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: It is necessary to write by word (32-bit) access. Half word/byte write access is prohibited.
Note: Security: S-TYPE-6
Privilege: P-TYPE-2

LRU[4:0] bits

The LRU[4:0] bits indicate each data when C-Cache test access.

Which bits are valid depends on the type of C-Cache test access.

C-Cache test access should be done with C-Cache turned off (CCACTL.ENC = 0).

If CCATAD register is updated by the test access reading and the CCATAD register is written at the same time, CCATAD register update by the test access reading is ignored and CCATAD register is written.

2.16.3.11 CCATAD_TAGECC : C-Cache Test Access Data (TAGECC)

Base address: CACHE = 0x4001_C000
CACHE_NS = 0x5001_C000

Offset address: 0x018

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

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

Bit	Symbol	Function	R/W
6:0	TAG ECC[6:0]	C-Cache test access Tag ECC code These bits are valid when C-Cache test access Tag ECC code read/write (CCATAA[18:16] = 100).	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

Note: It is necessary to write by word (32-bit) access. Half word/byte write access is prohibited.
Note: Security: S-TYPE-6
Privilege: P-TYPE-2

TAG ECC[6:0] bits (C-Cache test access Tag ECC code)

These bits indicate each data when C-Cache test access.

Which bits are valid depends on the type of C-Cache test access.

C-Cache test access should be done with C-Cache turned off (CCACTL.ENC = 0).

If CCATAD register is updated by the test access reading and the CCATAD register is written at the same time, CCATAD register update by the test access reading is ignored and CCATAD register is written.

2.16.3.12 SCACTL : S-Cache Control Register

Base address: CACHE = 0x4001_C000
CACHE_NS = 0x5001_C000

Offset address: 0x040

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

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

Bit	Symbol	Function	R/W
0	ENS	S-Cache enable bit Set S-Cache enable. 0: S-Cache disable 1: S-Cache enable	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	FS	S-Cache flush bit Set S-Cache line flush. 0: No action 1: S-Cache line flush (all lines invalidate)	R/W
9	WB	S-Cache write-back Trigger to write back S-Cache. 0: No action 1: S-Cache write-back	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: Security: S-TYPE-3 (CACHESAR.CACHESA)
Privilege: P-TYPE-2

ENS bit (S-Cache enable bit)

The ENS bit controls the cache enable of S-Cache.

When S-Cache set to enable, whether it caches the data or not depends on the access attribute setting of cacheable for CPU1.

FS bit (S-Cache flush bit)

The FS bit is an alias for SCAFCT.FS. See section 2.16.3.13. SCAFCT : S-Cache Flush Control Register for details.

WB bit (S-Cache write-back)

The WB bit is an alias for SCAFCT.WB. See section 2.16.3.13. SCAFCT : S-Cache Flush Control Register for details.

2.16.3.13 SCAFCT : S-Cache Flush Control Register

Base address: CACHE = 0x4001_C000
CACHE_NS = 0x5001_C000

Offset address: 0x044

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

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

Bit	Symbol	Function	R/W
0	FS	S-Cache flush bit Set S-Cache line flush. 0: No action 1: S-Cache line flush (all lines invalidate)	R/W
1	WB	S-Cache write-back Trigger to write back S-Cache. 0: No action 1: S-Cache write-back	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: Security: S-TYPE-5
Privilege: P-TYPE-2

FS bit (S-Cache flush bit)

The FS bit controls the cache flush of S-Cache.

[Clearing condition]

- This bit is cleared automatically when a cache flush is completed.

[Setting condition]

- Writing 1 to this bit.

WB bit (S-Cache write-back)

The WB bit triggers a write-back of S-Cache.

[Clearing condition]

- This bit is cleared automatically when write back is completed.

[Setting condition]

- Writing 1 to this bit.

Note: Setting FS to 1 when SCAWTA.WT is 0 performs both write back and flush.

If flush or write back has already started (SCAFCT.FS = 1 or SCAFCT.WB = 1), SCAFCT and SCACTL cannot be written until SCAFCT.FS and SCAFCT.WB are cleared automatically, that is, the process completes.

Cache is disabled after reset.

Cache does not write back and flush cache memory automatically when cache is disabled or reset.

If there is a dirty line left in the cache while cache is disabled, inconsistencies can occur between the cache and memory.

To disable the cache, use the following procedure.

1. Write 0x0100 (if SCAWTA.WT = 1) or 0x0300 (if SCAWTA.WT = 0) to SCACTL.

2. Make sure that SCAFCT.FC and SCAFCT.WB are both 0.

To enable the cache after reset, use the following procedure.

1. Set SCAFCT.FS to 1 to flush cache memory.
2. Wait for the flush to complete until SCAFCT.FS is 0.
3. Set SCACTL.ENS to 1 to enable the cache.
4. Writing to SCACTL or SCAFCT after reset automatically flushes the cache.

To enable the cache after cache is disabled, use the following procedure.

1. Make sure SCAFCT.WB and SCAFCT.FS is 0, that is, the cache does not perform write back or flush.
2. Set SCACTL.ENS to 1 to enable the cache.

2.16.3.14 SCAWTA : S-Cache Write Attribute

Base address: CACHE = 0x4001_C000
CACHE_NS = 0x5001_C000

Offset address: 0x04C

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

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

Bit	Symbol	Function	R/W
0	WT	S-Cache write-through Setting the attribute of S-Cache for write-through or write-back. 0: Write-through or write-back depends on the cache attribute of the area 1: Always write-through mode	R/W
1	WA	S-Cache write-allocation Setting the attribute of S-Cache write-allocation. 0: Always disable write-allocation 1: Write-allocation enable depends on the cache attribute of the area	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R

Note: Security: S-TYPE-3 (CACHESAR.CACHESA)
Privilege: P-TYPE-2

WT bit (S-Cache write-through)

The WT bit specifies the write attribute of S-Cache for write-through or write-back.

S-Cache must be disabled (SCACTL.ENS = 0) when changing this bit value.

When WT is 0, the cache attribute is determined by the default memory map or MPU of the Cortex-M33 processor.

WA bit (S-Cache write-allocation)

The WA bit enables S-Cache write-allocation.

S-Cache must be disabled (SCACTL.ENS = 0) when changing this bit value.

2.16.3.15 SCAEDST : S-Cache Error Detection Status

Base address: CACHE = 0x4001_C000
CACHE_NS = 0x5001_C000

Offset address: 0x050

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	EST2	ESTD	ESTC	ESD1	ESD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ESD0	S-Cache data error status 0 Indicates status of whether ECC 1-bit error was detected when reading S-Cache data memory. Write value must be 0. Writing 0 clears this status. 0: ECC 1-bit error was not detected 1: ECC 1-bit error was detected, and corrected	R/W
1	ESD1	S-Cache data error status 1 Indicates status of whether ECC 2-bit error was detected when reading S-Cache data memory. Write value must be 0. Writing 0 clears this status. 0: ECC 2-bit error was not detected 1: ECC 2-bit error was detected	R/W
2	ESTC	S-Cache Tag clean line invalidate status Indicates status of whether clean line was invalidated by ECC 1-bit error of S-Cache tag memory. Write value must be 0. Writing 0 clears this status. 0: Clean line was not invalidated 1: Clean line was invalidated	R/W
3	ESTD	S-Cache Tag dirty line invalidate status Indicates status of whether dirty line was invalidated by ECC 1-bit error of S-Cache tag memory. Write value must be 0. Writing 0 clears this status. 0: Dirty line was not invalidated 1: Dirty line was invalidated	R/W
4	EST2	S-Cache Tag 2-bit error status Indicates status of whether ECC 2-bit error was detected when reading S-Cache tag memory. Write value must be 0. Writing 0 clears this status. 0: ECC 2-bit error was not detected 1: ECC 2-bit error was detected	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: Security: S-TYPE-3 (CACHESAR.CACHESA)
Privilege: P-TYPE-2

ESD0, ESD1

These bits indicate the status of whether ECC errors were detected when reading S-Cache data memory.

Writing 0 clears this status. Resets other than Bus Error Reset or Memory Error Reset clear this status. If an error occurs and the status is cleared at the same time, the status is cleared, and the error is ignored.

ESTC, ESTD, EST2

These bits indicate the status of whether uncorrectable ECC error were detected when reading S-Cache tag memory.

ESTC : Invalidate clean line by ECC 1-bit error (correctable)

ESTD : Invalidate dirty line by ECC 1-bit error (uncorrectable)

EST2 : Detect ECC 2-bit error (uncorrectable)

Writing 0 clears this status. If an error occurs and the status is cleared at the same time, the status is cleared, and the error is ignored.

If ESTD is 1, dirty line is invalidated without write back.

2.16.3.16 SCATAA : S-Cache Test Access Address

Base address: CACHE = 0x4001_C000
CACHE_NS = 0x5001_C000

Offset address: 0x054

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

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

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
4:2	OFFSET[2:0]	Address offset S-Cache test access address lower 3-bit [4:2] corresponds to upper 3-bit of 5-bit offset [4:0]. These bits are valid when cache data read/write and ECC code read/write.	R/W
11:5	ENTRY[6:0]	Address entry S-Cache test access address middle 7-bit [11:5] corresponds to entry.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
18:16	TARGET[2:0]	Access target These bits specify the S-Cache test access target. Access type is specified by bit RW[23]. 0 0 0: Cache data read or write 0 0 1: Data ECC code read or write 0 1 0: Tag, V, D read or write 0 1 1: LRU read or write 1 0 0: Tag ECC code read or write 1 0 1 - 1 1 1: Setting prohibited	R/W
22:19	—	These bits are read as 0. The write value should be 0.	R/W
23	RW	Read write This bit specifies whether S-Cache test access type is read or write. 0: read 1: write	R/W
29:24	—	These bits are read as 0. The write value should be 0.	R/W
31:30	WAY[1:0]	Address way S-Cache test access address higher 2-bit corresponds to way. These bits are valid when cache data/ecc code/tag read/write. These bits are invalid when LRU read/write.	R/W

Note: It is necessary to write by word (32-bit) access. Half word/byte write access is prohibited.
Note: Security: S-TYPE-6
Privilege: P-TYPE-2

WAY[1:0], ENTRY[6:0], OFFSET[2:0] bit

These bits indicate the address when S-Cache test access.

WAY[1:0], ENTRY[6:0], OFFSET[2:0] are valid for ECC code/cache data read or write.

WAY[1:0], ENTRY[6:0] are valid, and OFFSET[2:0] is invalid for TAG and TAG ECC code read or write.

ENTRY[6:0] is valid and WAY[1:0], OFFSET[2:0] are invalid for LRU read or write.

TARGET[2:0] bits

The TARGET[2:0] bits specify the S-Cache test access target. The truth table is below.

RW bit

The RW bit specifies whether S-Cache test access type is read or write. The truth table is below.

SCATAA.RW[23]	SCATAA.TARGET[18:16]	Command
0	000	Cache data read
1	000	Cache data write
0	001	Data ECC code read
1	001	Data ECC code write
0	010	Tag read
1	010	Tag write
0	011	LRU read
1	011	LRU write
0	100	Tag ECC code read
1	100	Tag ECC code write

Writing to SCATAA register triggers the test access. When the test access is writing, write to register SCATAD before writing to register SCATAA. When the test access is reading, read register SCATAD after writing to register SCATAA.

To perform read access, follow these steps:

1. Write to SCATAA with SCATAA.RW = 0.
2. Read SCATAA to check.
3. Read SCATAD.

To perform write access, follow these steps:

1. Write to SCATDA.
2. Write to SCATAA with SCATAA.RW = 1.

If SCATAD register is updated by the test access reading and the SCATAD register is written at the same time, SCATAD register update by the test access reading is ignored and SCATAD register is written.

2.16.3.17 SCATAD_DATA : S-Cache Test Access Data (DATA)

Base address: CACHE = 0x4001_C000
CACHE_NS = 0x5001_C000

Offset address: 0x058

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

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

Bit	Symbol	Function	R/W
31:0	DATA[31:0]	S-Cache test access data These bits are valid when S-Cache test access cache data read/write (SCATAA[18:16] = 000).	R/W

Note: It is necessary to write by word (32-bit) access. Half word/byte write access is prohibited.
Note: Security: S-TYPE-6
Privilege: P-TYPE-2

DATA[31:0] bits

These bits indicate each data when S-Cache test access.

Which bits are valid depend on the type of S-Cache test access.

S-Cache test access should be done with S-Cache turned off (SCACTL.ENS = 0).

If SCATAD register is updated by the test access reading and the SCATAD register is written at the same time, SCATAD register update by the test access reading is ignored and SCATAD register is written.

2.16.3.18 SCATAD_ECC : S-Cache Test Access Data (ECC)

Base address: CACHE = 0x4001_C000
CACHE_NS = 0x5001_C000

Offset address: 0x058

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

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

Bit	Symbol	Function	R/W
6:0	ECC[6:0]	S-Cache test access ECC code These bits are valid when S-Cache test access data ECC code of cache data read/write (SCATAA[18:16] = 001).	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

Note: It is necessary to write by word (32-bit) access. Half word/byte write access is prohibited.
Note: Security: S-TYPE-6
Privilege: P-TYPE-2

ECC[6:0] bits

These bits indicate each data when S-Cache test access.

Which bits are valid depend on the type of S-Cache test access.

S-Cache test access should be done with S-Cache turned off (SCACTL.ENS = 0).

If SCATAD register is updated by the test access reading and the SCATAD register is written at the same time, SCATAD register update by the test access reading is ignored and SCATAD register is written.

2.16.3.19 SCATAD_TAG : S-Cache Test Access Data (TAG)

Base address: CACHE = 0x4001_C000
CACHE_NS = 0x5001_C000

Offset address: 0x058

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TAG[19:4]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	D	S-Cache test access Dirty This bit is valid when S-Cache test access Tag, V, D read/write (SCATAA[18:16] = 010).	R/W
1	V	S-Cache test access Valid This bit is valid when S-Cache test access Tag, V, D read/write (SCATAA[18:16] = 010).	R/W
11:2	—	These bits are read as 0. The write value should be 0.	R/W
31:12	TAG[19:0]	S-Cache test access Tag data These bits are valid when S-Cache test access Tag, V, D read/write (SCATAA[18:16] = 010).	R/W

Note: It is necessary to write by word (32-bit) access. Half word/byte write access is prohibited.
Note: Security: S-TYPE-6
Privilege: P-TYPE-2

TAG, V, D bits

These bits indicate each data when S-Cache test access.

Which bits are valid depend on the type of S-Cache test access.

S-Cache test access should be done with S-Cache turned off (SCACTL.ENS = 0).

If SCATAD register is updated by the test access reading and the SCATAD register is written at the same time, SCATAD register update by the test access reading is ignored and SCATAD register is written.

2.16.3.20 SCATAD_LRU : S-Cache Test Access Data (LRU)

Base address: CACHE = 0x4001_C000
CACHE_NS = 0x5001_C000

Offset address: 0x058

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

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

Bit	Symbol	Function	R/W
4:0	LRU[4:0]	S-Cache test access LRU data These bits are valid when S-Cache test access LRU read/write (SCATAA[18:16] = 011).	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: It is necessary to write by word (32-bit) access. Half word/byte write access is prohibited.
Note: Security: S-TYPE-6
Privilege: P-TYPE-2

LRU[4:0] bits

The LRU[4:0] bits indicate each data when S-Cache test access.

Which bits are valid depend on the type of S-Cache test access.

S-Cache test access should be done with S-Cache turned off (SCACTL.ENS = 0).

If SCATAD register is updated by the test access reading and the SCATAD register is written at the same time, SCATAD register update by the test access reading is ignored and SCATAD register is written.

2.16.3.21 SCATAD_TAGECC : S-Cache Test Access Data

Base address: CACHE = 0x4001_C000
CACHE_NS = 0x5001_C000

Offset address: 0x058

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

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

Bit	Symbol	Function	R/W
6:0	TAGECC[6:0]	S-Cache test access Tag ECC code These bits are valid when S-Cache test access Tag ECC code read/write (CCATAA[18:16] = 100).	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

Note: It is necessary to write by word (32-bit) access. Half word/byte write access is prohibited.
Note: Security: S-TYPE-6
Privilege: P-TYPE-2

TAGECC[6:0] bits (S-Cache test access Tag ECC code)

The TAGECC[6:0] bits indicate each data when S-Cache test access.

Which bits are valid depend on the type of S-Cache test access.

S-Cache test access should be done with S-Cache turned off (SCACTL.ENC = 0).

If SCATAD register is updated by the test access reading and the SCATAD register is written at the same time, SCATAD register update by the test access reading is ignored and SCATAD register is written.

2.16.3.22 CAPOAD : Cache ECC Error Operation After Detection Register

Base address: CACHE = 0x4001_C000
CACHE_NS = 0x5001_C000

Offset address: 0x200

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	E1STSEN	ECCMOD1	—	—	OAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OAD	Operation after detection bit Selects the operation after error detection. 0: Interrupt 1: Reset	R/W
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	ECCMOD1	ECC enable Enable ECC (error correction code) on C-Cache and S-Cache. 0: Disable ECC 1: Enable ECC	R/W
4	E1STSEN	ECC correctable error information update enable Correctable ECC error is: ● 1-bit ECC error of Cache Data memory ● Clean line of Cache Tag memory was invalidated by ECC 1-bit error 0: Disable updating of correctable ECC error information 1: Enable updating of correctable ECC error information	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: Security: S-TYPE-3 (CACHESAR.CACHEESA)
Privilege: P-TYPE-2

The CAPOAD register is write-protected by CAPRCR. Set CAPRCR.PRCR before writing to this register. Do not write to this register when the cache is enabled.

OAD bit (Operation after detection bit)

The OAD bit specifies the operation after ECC error detection.

ECCMOD1 bit (ECC enable)

The ECCMOD1 bit enables detecting and correcting error with ECC on C-Cache and S-Cache.

E1STSEN bit (ECC correctable error information update enable)

The E1STSEN bit enables or disables updating of the correctable ECC error status CCAEDST.ESD0/SCAEDST.ESD0/CCAEDST.ESTC/SCAEDST.ESTC.

When CAPOAD.E1STSEN = 0, even if a correctable ECC error occurs, the error status is not updated and no actions such as interrupt or reset are taken after an error is detected, regardless of the value of CAPOAD.OAD.

2.16.3.23 CAPRCR : Cache Protection Register

Base address: CACHE = 0x4001_C000
CACHE_NS = 0x5001_C000

Offset address: 0x204

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

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

Bit	Symbol	Function	R/W
0	PRCR	Register Write Control bit Set Register Write Control 0: Writing to the protected register is disabled 1: Writing to the protected register is enabled	R/W
7:1	KW[6:0]	Write Key Code bit These bits are used to enable or disable writing of the PRCR bit.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: Security: S-TYPE-3 (CACHESAR.CACHEESA)
Privilege: P-TYPE-2

PRCR bit (Register Write Control bit)

The PRCR bit controls the writing of the CAPOAD register.

When the PRCR bit is set to 1, writing to the CAPOAD register is enabled.

When writing to this bit, write 0x78 in KW[6:0] at the same time.

KW[6:0]bits (Write Key Code bit)

The KW[6:0] bits are used to enable or disable writing of the PRCR bit.

When writing to PRCR bit, write 0x78 in KW[6:0] at the same time.

When the value of KW[6:0] bits except 0x78 is written in, PRCR bit is not updated.

The KW[6:0] bits are always read as 0x00.

2.16.4 Operation

2.16.4.1 C-Cache/S-Cache

Figure 2.8 shows the function flow to access C-Cache/S-Cache from CPU.

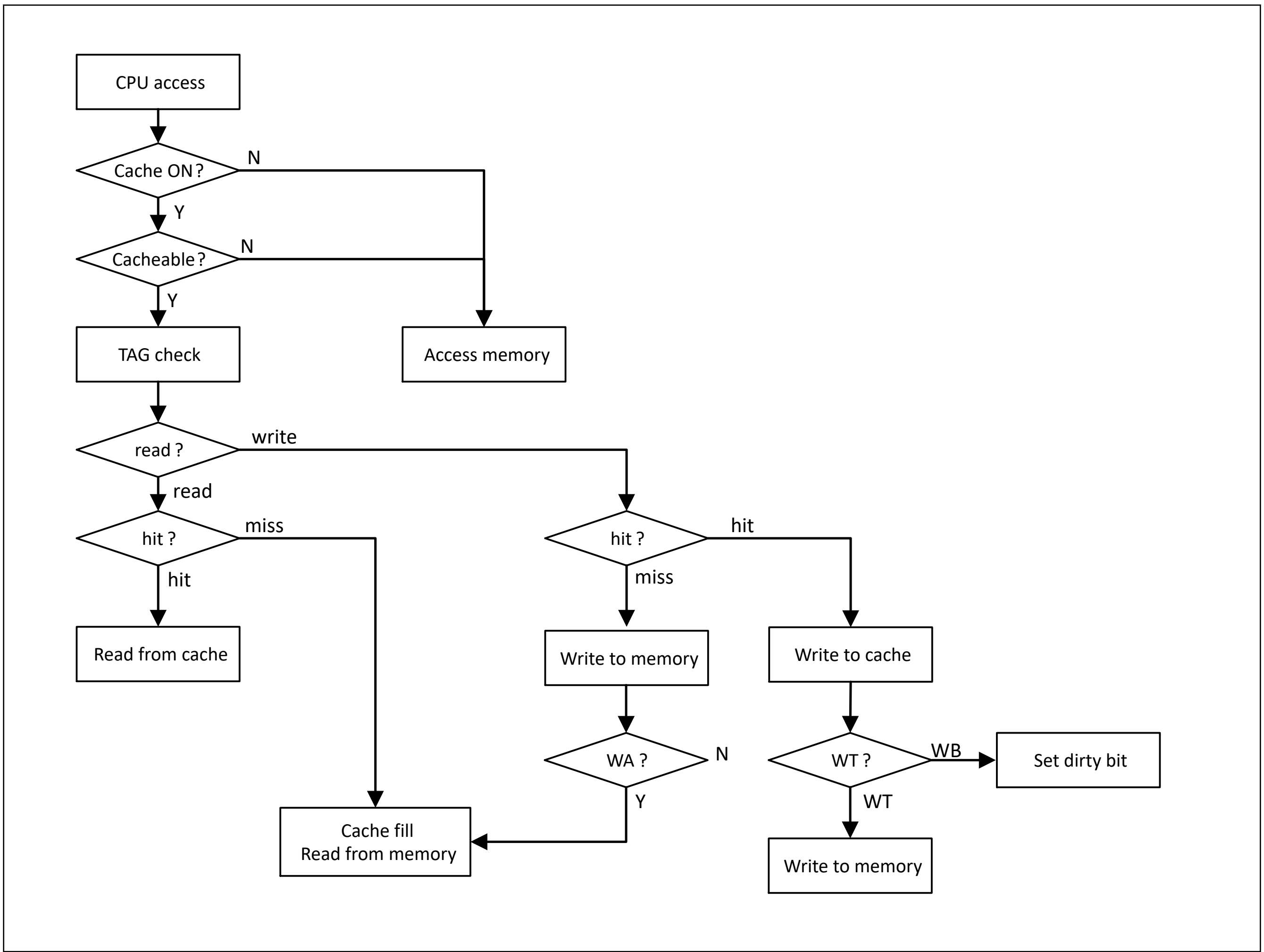


Figure 2.8 CPU access function flow of C-Cache/S-Cache

The cache function works when the cache is enabled (CCACTL.ENC = 1, SCACTL.ENS = 1) and cacheable access from the CPU. Cache checks the address from which the CPU access request and the one in the cache tag, and determines that the CPU access is a hit or miss.

- Read miss: Cache reads one cache line data from memory and stores it into the cache data. Then the cache returns the required data to the CPU.
- Read hit: Cache reads required data from the cache data and returns to the CPU. Then the access cycle is determined as a hit of zero-wait cycle.
- Write miss: Cache writes data to memory. When write-allocation is set (CCAWTA.WA = 1, SCAWTA.WA = 1) and the attribute of area is write-allocation, the cache fills a line from memory.
- Write hit: Cache writes the data to the cache. When write-through is set (CCAWTA.WT = 1, SCAWTA.WT = 1) or attribute of area is write-through, the cache writes the data to memory. However, when write-through is not set (CCAWTA.WT = 0, SCAWTA.WT = 0) and attribute of area is write-back, the cache sets the Dirty bit of the data in Tag.

2.16.4.2 Cache Flush

In order to flush the cache, set 1 to CCAFCT.FC and SCAFCT.FS bits. The Valid bit, Dirty bit and Tag data in Tag are then cleared. The Tag ECC code is initialized to the code generated by the value 0. Cache data is not cleared.

Note: If the same address is accessed with different cacheability (set by the Arm MPU), write back and flush the cache before any access after cacheability changes. If CCAWTA.WT and SCAWTA.WT are set to 1, no write back is required.

If the same address is accessed with different cacheability without write back and flush, the read/write data is undefined.

2.16.4.3 LRU and Replace

Cache uses LRU (Least Recently Used) mechanism as cache replacement algorithm. If a CPU access is determined as a hit or miss, the cache replaces the cache data which is not the last restored. The cache also marks as the latest data in LRU of the cache data. Therefore, even if cache line in cache ways are full, the cache can replace the cache data using LRU that shows the older cache.

2.16.4.4 ECC

2.16.4.4.1 Correction of ECC Errors of Cache Data Memory

Enabling and disabling of ECC error correction can be selected through the setting by the CAPOAD.ECCMOD1 bit. In the initial state, ECC error correction is disabled. The ECC check type is SEC-DED (Single-Error Correction and Double-Error Detection Code).

When ECC function is enabled, 7-bit ECC is generated from 32-bit data, 32-bit data and 7-bit ECC code are written to the cache data memory. For reading, 32-bit data and 7-bit ECC code are read out from the cache data memory.

When ECC function is enabled, if a 1-bit error occurs, error correction is performed and the CCAEDST.ESD0 or SCAEDST.ESD0 bit is set to 1, if the CAPOAD.E1STSEN bit is 1. If a 2-bit error occurs, error detection is performed and the CCAEDST.ESD1 or SCAEDST.ESD1 bit is set to 1, though error correction is not performed.

When ECC error is detected in cache data, it only performs error correction and CCAEDST, SCAEDST setting. It does not invalidate the cache like Tag ECC error.

When ECC function is disabled, error correction or error detection is not performed even though 1-bit or 2-bit error occurs. So the CCAEDST.ESD0 bit, the CCAEDST.ESD1 bit, the SCAEDST.ESD0 bit and the SCAEDST.ESD1 bit are not updated.

2.16.4.4.2 ECC Errors of Cache Tag Memory

Enabling and disabling of Tag ECC error detection can be selected through the setting by the CAPOAD.ECCMOD1 bit (the same as cache data). The ECC check type is SED-DED (Single-Error Detection and Double-Error Detection Code).

When ECC function is enabled, 7-bit ECC code is generated from 32-bit data (upper bit of tag/V extended 0), tag data and 7-bit ECC code are written to the cache tag memory. For read, tag data and 7-bit ECC code are read out from the cache tag memory.

When ECC function is disabled, no error detection is performed even though 1-bit or 2-bit error occurs. So, the CCAEDST.ESTC bit, the CCAEDST.ESTD bit, the CCAEDST.EST2 bit, the SCAEDST.ESTC and the SCAEDST.ESTD bits, and the SCAEDST.EST2 bit are not updated.

If Tag ECC error is detected, the line is invalidated and treated as a miss. For clean line error, the cache can be recovered by cache refilling. However, for dirty line error, the system cannot be recovered because dirty data cannot write back.

2.16.4.4.3 ECC Error Interrupt Function

When ECC function is enabled, error checking is possible in the cache tag/data memory.

For a 2-bit error of the cache data, the CCAEDST.ESD1 bit or the SCAEDST.ESD1 bit is set to 1. For a 1-bit error of the cache data, the CCAEDST.ESD0 bit or the SCAEDST.ESD0 bit is set to 1 when CAPOAD.E1STSEN bit is 1.

For a 2-bit error of the cache tag, the CCAEDST.EST2 bit or the SCAEDST.EST2 bit is set to 1. For invalidate of the cache tag dirty line by ECC 1-bit error, the CCAEDST.ESTD bit or the SCAEDST.ESTD bit is set to 1.

For invalidate of the cache tag clean line by ECC 1-bit error, the CCAEDST.ESTC bit or the SCAEDST.ESTC bit is set to 1 when CAPOAD.E1STSEN bit is 1.

When the correctable error is to be masked, it is necessary to set the CAPOAD.E1STSEN bit to 0 to disable updating the following bits to 1.

- CCAEDST.ESD0
- CCAEDST
- ESTC
- SCAEDST.ESD0

- SCAEDST.ESTC

An ECC error is not generated while ECC function is disabled

ECC error may result request of a reset or an interrupt. When an ECC error occurs and when the CAPOAD.OAD bit is 1, a reset request is output. When it is 0, an interrupt request is output to the ICU.

ECC Error interrupt occurs when there is at least one error status in the CCAEDST register and the SCAEDST register is set to 1. ECC Error interrupt continues to occur until all error status in the CCAEDST register and SCAEDST register are cleared.

2.16.4.4.4 ECC Decoder Testing

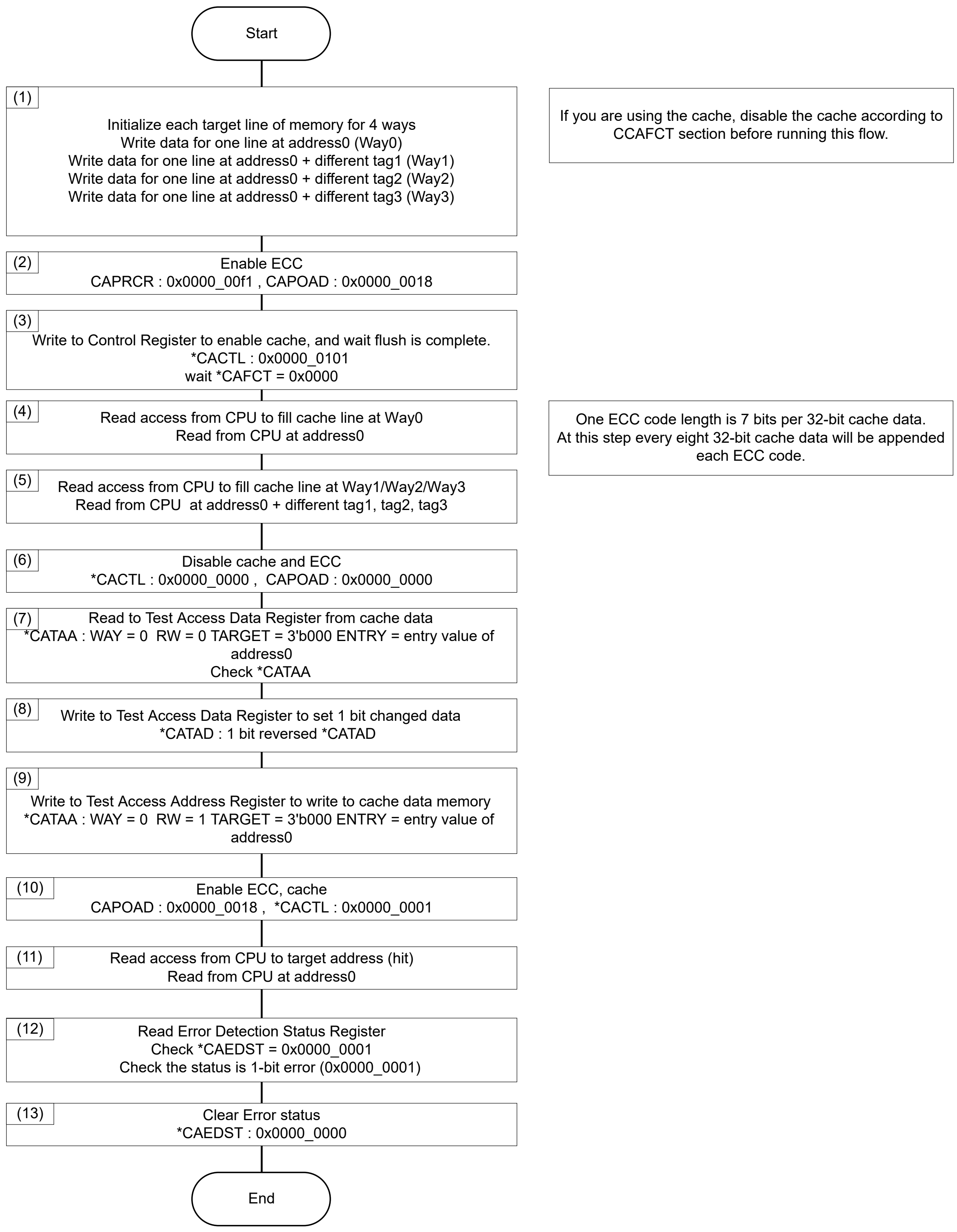
There are six ECC decoders in total, one for each way of the cache tag, and one each for the CPU read path and write-back path of cache data.

To test the ECC decoder in cache data, perform the flow of CPU read (section 2.16.4.4.4.1. ECC Decoder of CPU Read) and the flow of write-back (section 2.16.4.4.4.2. ECC Decoder of Write-back). To test the ECC decoder in cache tag, perform all of Way0 through Way3 of tag in the flow of CPU read (section 2.16.4.4.4.1. ECC Decoder of CPU Read).

Since the ECC decoder for cache data is shared between Way0 and Way3, it is possible to test the ECC decoder without testing both ways. For reference, both ways test are described.

2.16.4.4.4.1 ECC Decoder of CPU Read

Figure 2.9 shows the testing flow of cache ECC decoder for CPU reading from the cache or decoder of cache tag.



Note: If you are using the cache, disable the cache according to section 2.16.3.3. CCAFCT : C-Cache Flush Control Register before running this flow.

Figure 2.9 Test flow of cache ECC decoder for CPU reading from decoder of cache tag

Notes for each step in the operation flow are described as follows.

(1) To initialize the target line, write data at the address that offset value is from 0x00 to 0x1F.

To initialize data for Way1, write at address0 + different tag1 address.

Initialize data for Way2 and Way3 as well.

For example: address0 = 0x2100_0000

address0 + different tag1 = 0x2101_0000

address0 + different tag2 = 0x2102_0000

address0 + different tag3 = 0x2103_0000

(7)(9) The flow for these steps is about ECC decoder in Way0 of the cache data. For ECC decoders test except write-back ECC decoder (see section 2.16.4.4.4.2. ECC Decoder of Write-back), repeat steps (1) to (13) with the following item combination value written to *CATAA.

	*CATAA.TARGET
Data	000
Data ECC	001
Tag	010
Tag ECC	100

	*CATAA.WAY
Way0	00
Way1	01
Way2	10
Way3	11

Test all of Way0 through Way3 in TARGET = 3’b010 or 3’b100, because Tag ECC decoders are in each of all ways.

(8) To test 2-bit ECC error detection, write 2 bits reversed *CATAD.

(11) Read from CPU at address0 + different tag1 when testing Way1.

Read from CPU at address0 + different tag2 when testing Way2.

Read from CPU at address0 + different tag3 when testing Way3.

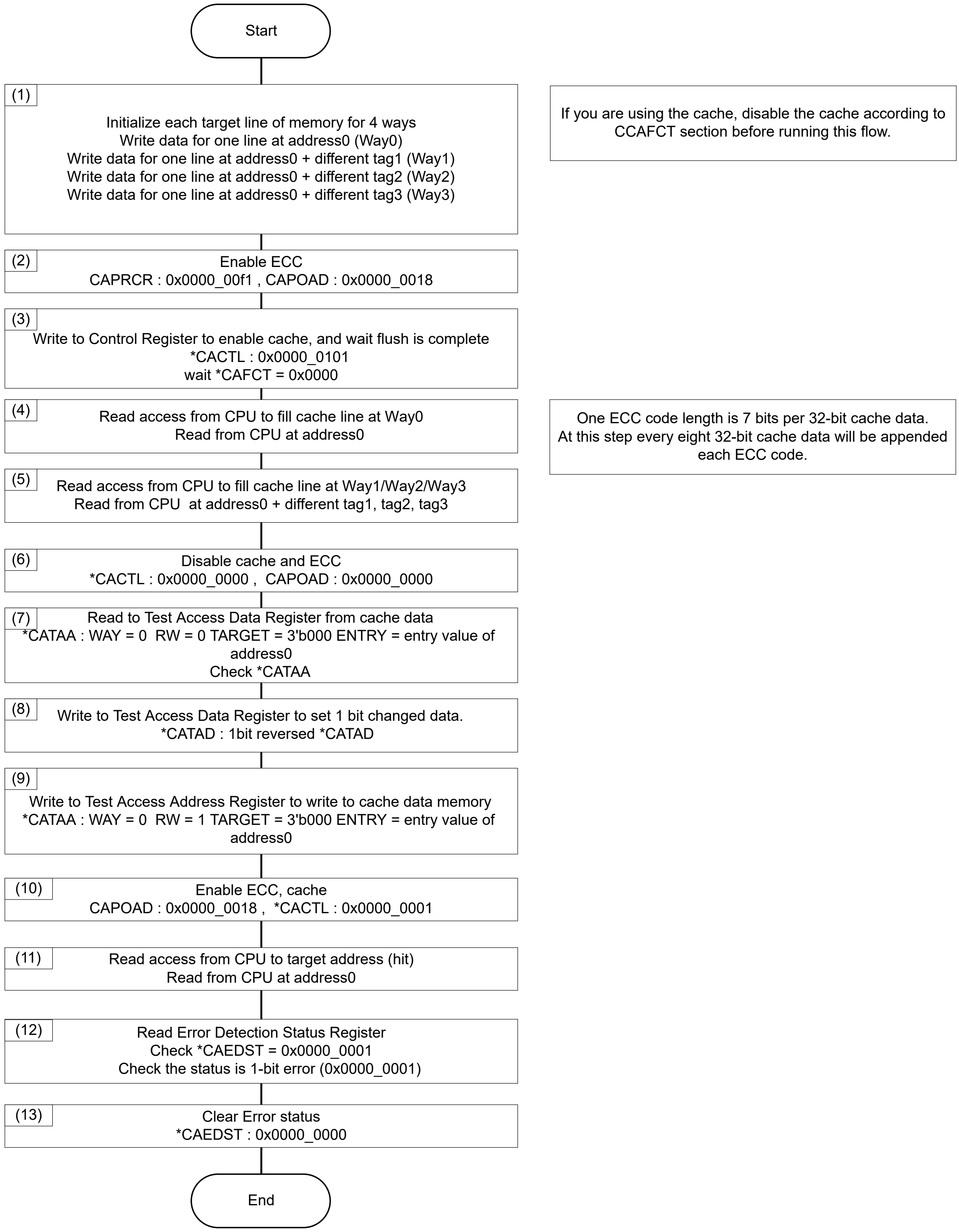
(12) To test 1-bit ECC error correction, check that the read value has been corrected.

Check *CAEDST is of the following value in each test.

Target	Error	Value of *CAEDST
Data	ECC 1-bit error	0x0000_0001
	ECC 2-bit error	0x0000_0002
Data ECC	ECC 1-bit error	0x0000_0001
	ECC 2-bit error	0x0000_0002
Tag	ECC 1-bit error	0x0000_0004 (in this flow, the cache line is clean)
	ECC 2-bit error	0x0000_0010
Tag ECC	ECC 1-bit error	0x0000_0004 (in this flow, the cache line is clean)
	ECC 2-bit error	0x0000_0010

2.16.4.4.4.2 ECC Decoder of Write-back

Figure 2.10 shows the testing flow of cache ECC decoder for write-back.



Note: If you are using the cache, disable the cache according to section 2.16.3.3. CCAFCT : C-Cache Flush Control Register before running this flow.

Figure 2.10 Test flow of cache ECC decoder for write-back

Notes for each step in the operation flow are described as follows.

(1) To initialize the target line, write data at the address that offset value is from 0x00 to 0x1F.

To initialize data for Way1, write at address0 + different tag1 address.

Initialize data for Way2 and Way3 as well.

For example: address0 = 0x2100_0000

address0 + different tag1 = 0x2101_0000

address0 + different tag2 = 0x2102_0000

address0 + different tag3 = 0x2103_0000

(7)(9) The flow for these steps is about ECC decoder in Way0 of the cache data. For decoder in write-back, repeat steps (1) to (15) with the following item combination value written to *CATAA.

	*CATAA.TARGET
Data	000
Data ECC	001

	*CATAA.WAY
Way0	00
Way1	01
Way2	10
Way3	11

(8) To test 2-bit ECC error detection, write 2 bits reversed *CATAD.

(10)(11) The flow for these steps is about ECC decoder in Way0 of the cache data. For decoder in path of other ways, repeat steps (1) to (15) with the following value of Way0 through Way3.

	Way0	Way1	Way2	Way3
*CATAA.WAY	00	01	10	11
*CATAD of (11)	0x0000_0014	0x0000_0009	0x0000_001E	0x0000_0013

(13) Different tag (not in cache) has a different tag address from each Way address in (1).

For example: address0 + different tag (not in cache) = 0x2110_0000

(14) To test 1-bit ECC error, check that the read value at address0 (write-back value) has been corrected.

To test 2-bit ECC error, check that the read value at address0 (write-back value) has not been corrected.

Check *CAEDST is of the following value.

Target	Error	*CAEDST
Data	ECC 1-bit error	0x0000_0001
	ECC 2-bit error	0x0000_0002
Data ECC	ECC 1-bit error	0x0000_0001
	ECC 2-bit error	0x0000_0002

2.16.4.4.5 Note of ECC

When switching to enable ECC, follow the procedure (section 2.16.3.3. CCAFCT : C-Cache Flush Control Register and section 2.16.3.13. SCAFCT : S-Cache Flush Control Register) to disable C-Cache and S-Cache before switching. Switching to enable ECC while the cache is enabled can cause ECC errors due to incorrect ECC code.

2.16.4.5 Bus Error

The correspondence to a bus error from a bus slave is shown in the section that follows:

- For cache off: Cache returns the bus error to the CPU.
- For non-cacheable access: Cache returns the bus error to the CPU.

- For cache fill: Cache returns the bus error to the CPU. The cache does not write the line contains error to cache memory.
- For write hit status: Cache cannot return the bus error to the CPU. The cache does not clear validation.
- For write miss status: Cache cannot return the bus error to the CPU.

2.16.5 Usage Notes

2.16.5.1 Coherency

The coherency between the cache and the external memory and internal SRAM must be guaranteed by software. To share memory between multiple bus masters (such as between multiple CPUs or between CPU and DMAC) in the cache support area, perform write back and flush the cache data as needed.

2.16.5.2 Cache Flush/WB Operation

When CPU1 operates Cache Flush/WB by *CAFCT/*CACTL, the CPU1 must check that Flush/WB completes (*CAFCT = 0, * denotes C-Cache, or S-Cache, that is, CCAFCT/CCACTL or SCAFCT/SCACTL).

Do not operate Flush/WB from a bus master other than CPU1.

When Cache Flush/WB operation is executed, the priority of Cache Flush/WB operation is higher than that of access from CPU1.

2.16.5.3 Restrictions Relating to Security Attribution of C-Cache and S-Cache

When using C-Cache and S-Cache, one of the following conditions must be observed:

- CPU1 should be used only in secure state.
- CPU1 should be used only in non-secure state. It is only possible if SECEXT (Security Extension) of CPU1 is disabled.
- C-Cache and S-Cache should be used only in non-secure program. In addition, all of the following conditions must be observed:
    1. Set the MPU so that all secure memory space used by the secure program is non-cacheable.
    2. When secure program accesses non-secure memory space, execute WB and Flush of the cache before accessing it.
    3. When using on-chip WDT instead of an external WDT, the on-chip WDT should be set as a secure attribution and also set to apply interrupt to CPU0 or reset when a WDT error occurs.

2.17 CM33 TCM (Tightly Coupled Memory)

2.17.1 Overview

The MCU provides an on-chip TCM (tightly coupled memory) with Error Correction Code (ECC).

2.17.1.1 Specifications of TCM

Table 2.36 Specifications of TCM

Item	C-TCM	S-TCM
Capacity	64 KB	64 KB
Error detection	ECC (SECDED)	ECC (SECDED)
Security	Secure region/Non-secure region	Secure region/Non-secure region

Table 2.37 Address map of TCM (1 of 2)

Address	Usage	Master	SA
0x0000_0000 to 0x0000_FFFF	C-TCM	CPU1 (CM33)	Secure
0x1000_0000 to 0x1000_FFFF	C-TCM	CPU1 (CM33)	Non-secure

Table 2.37 Address map of TCM (2 of 2)

Address	Usage	Master	SA
0x2000_0000 to 0x2000_FFFF	S-TCM	CPU1 (CM33)	Secure
0x2A00_0000 to 0x2A00_FFFF	C-TCM	Bus master (example, CPU0)	Secure
0x2A01_0000 to 0x2A01_FFFF	S-TCM	Bus master (example, CPU0)	Secure
0x3000_0000 to 0x3000_FFFF	S-TCM	CPU1 (CM33)	Non-secure
0x3A00_0000 to 0x3A00_FFFF	C-TCM	Bus master (example, CPU0)	Non-secure
0x3A01_0000 to 0x3A01_FFFF	S-TCM	Bus master (example, CPU0)	Non-secure

Each C-TCM or S-TCM memory is one entity which capacity is 64 KB can be accessed from different master through different address.

If the access of bus master conflicts with the access of CPU1, the bus interface of TCM returns a wait cycle to the bus. See section 2.17.4.4. Wait Cycle.

2.17.1.2 Block Diagram

Figure 2.11 shows the block structure of Cache and TCM.

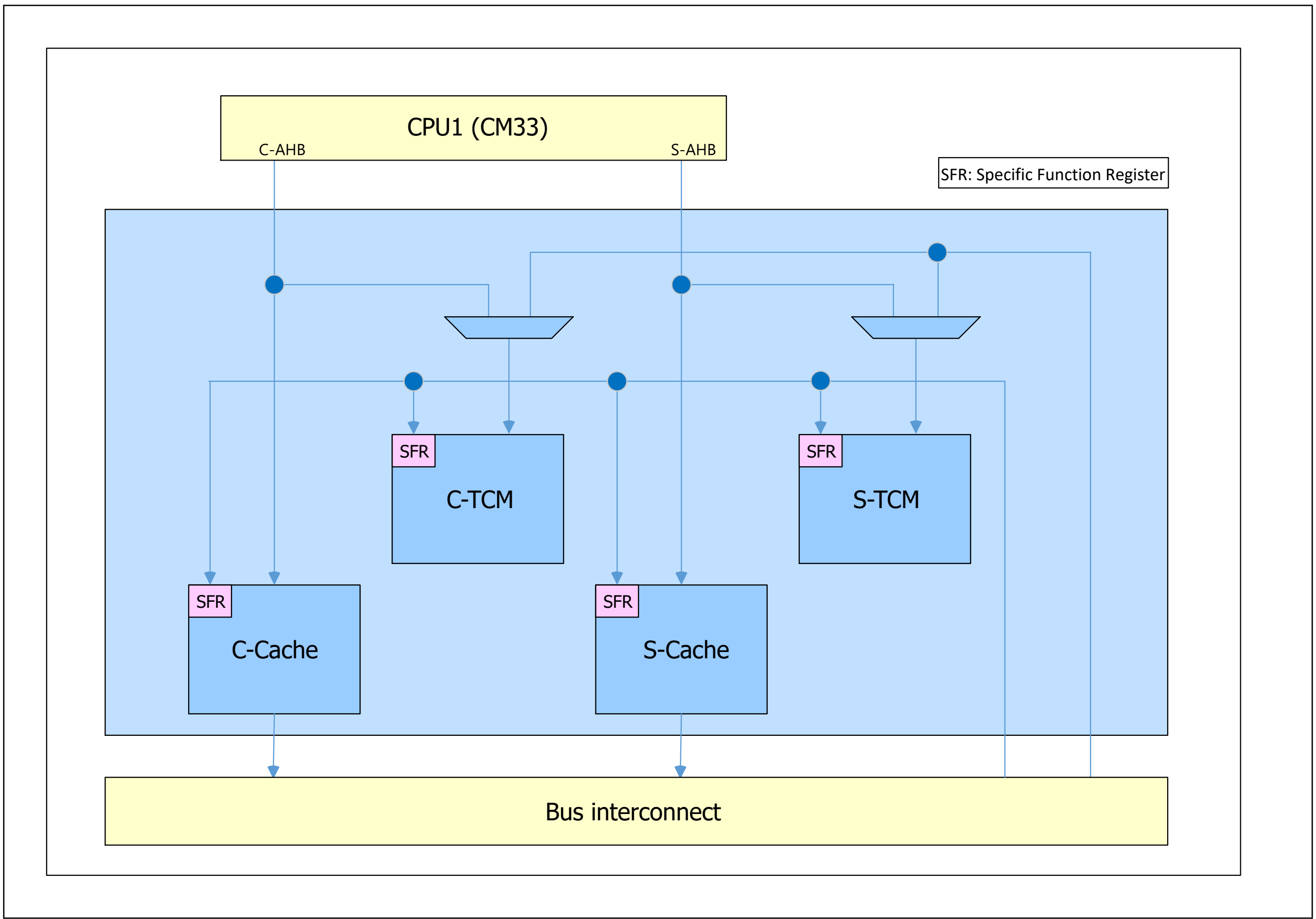


Figure 2.11 Block structure of Cache and TCM

2.17.2 Register List

TCM_BASE_S(Base address of TCM) is 0x4001_C800

TCM_BASE_NS(Base address of TCM) is 0x5001_C800

CPSCU_BASE_S (Base address of CPSCU) is 0x4000_8000.

CPSCU_BASE_NS (Base address of CPSCU) is 0x5000_8000.

Security Attribution Register (TCMSAR) is write protected by PRCR.PRCR4. TCMSAR controls the security attributes of all TCM registers. TCMPRCR_S and TCMPRCR_NS protects TCMCRn from secure and non-secure write accesses.

Table 2.38 Register list

Register name	Address (for secure access)		Address (for non-secure access)	
TCMSAR	CPSCU_BASE_S +	0x504	—	—
TCMSABARC TCMSABARS	CPSCU_BASE_S +	0x508 0x50C	—	—
TCMPRCR_S	TCM_BASE_S +	0x00	—	—
TCMPRCR_NS	—	—	TCM_BASE_NS +	0x04
TCMCRC TCMCRS	TCM_BASE_S +	0x10 0x14	TCM_BASE_NS +	0x10 0x14
TCMESR	TCM_BASE_S +	0x40	TCM_BASE_NS +	0x40
TCMESCLR	TCM_BASE_S +	0x48	TCM_BASE_NS +	0x48
TCMEARC0 TCMEARC1 TCMEARS0 TCMEARS1	TCM_BASE_S +	0x50 0x54 0x58 0x5C	TCM_BASE_NS +	0x50 0x54 0x58 0x5C

2.17.3 Register Descriptions

2.17.3.1 TCMSAR : TCM Security Attribution Register

Base address: CPSCU = 0x4000_8000

Offset address: 0x504

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

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

Bit	Symbol	Function	R/W
0	TCMSA	Security attributes of registers for TCM control 0: Secure 1: Non-Secure	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: This register is write-protected by PRCR_S.PRC4.

Security: S-TYPE-1

Privilege: P-TYPE-1

Only secure access can write to this register. Non-secure read access is allowed, non-secure write access is denied, TrustZone access error is not generated.

TCMSA bit (Security attributes of registers for TCM control)

Security attributes of registers for TCM control. The target registers are as follow.

- TCMCRx
- TCMESCLR

The TCMSA bit must be set to 0 when secure program/secure data is located at TCM.

2.17.3.2 TCMSABARx : TCM Security Attribute Boundary Address Register (x = C or S)

Base address: CPSCU = 0x4000_8000

Offset address: 0x508 or 0x50C

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

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

Bit	Symbol	Function	R/W
12:0	—	These bits are read as 0. The write value should be 0.	R/W
18:13	TCMSABA	Boundary address between secure and non-secure (start address of non-secure region).	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

Note: This register is write-protected by PRCR_S.PRC4.

Security: S-TYPE-1

Privilege: P-TYPE-1

TCMSABARx specifies the boundary address between the secure and non-secure regions of each TCM. TCMSABARx indicates the start address of non-secure regions. TCMSABARx 32-bit value corresponds to the absolute address. When writing to TCMSABARx, the write value of bits [31:19] and bits [12:0] should be 0.

The region lower than the boundary address is marked as Secure, and the region higher than or equal to the boundary address is marked as Non-secure.

TCMSABARC

When the boundary address is 0x0000_0000, the whole C-TCM is marked as Non-secure.

When the boundary address is 0x0001_0000 or higher, the whole C-TCM is marked as Secure.

Since the initial boundary address is 0x0007_E000, the whole of C-TCM (64 KB) is marked as Secure after reset.

The boundary address setting affects not only the CPU1 address but also the bus master address (alias address).

Table 2.39 Address map of C-TCM

Address	Master	SA
0x0000_0000 to 0x0000_0000 + TCMSABARC - 0x1	CPU1 (CM33)	Secure
0x0000_0000 + TCMSABARC to 0x0000_FFFF	CPU1 (CM33)	Non-secure
0x1000_0000 to 0x1000_0000 + TCMSABARC - 0x1	CPU1 (CM33)	Secure
0x1000_0000 + TCMSABARC to 0x1000_FFFF	CPU1 (CM33)	Non-secure
0x2A00_0000 to 0x2A00_0000 + TCMSABARC - 0x1	Bus master	Secure
0x2A00_0000 + TCMSABARC to 0x2A00_FFFF	Bus master	Non-secure
0x3A00_0000 to 0x3A00_0000 + TCMSABARC - 0x1	Bus master	Secure
0x3A00_0000 + TCMSABARC to 0x3A00_FFFF	Bus master	Non-secure

TCMSABARS

When the boundary address is 0x0000_0000, the whole S-TCM is marked as Non-secure.

When the boundary address is 0x0001_0000 or higher, the whole S-TCM is marked as Secure.

Since the initial boundary address is 0x0007_E000, the whole of S-TCM (64 KB) is marked as Secure after reset.

The boundary address setting affects not only the CPU1 address but also the bus master address (alias address).

Table 2.40 Address map of S-TCM

Address	Master	SA
0x2000_0000 to 0x2000_0000 + TCMSABARS - 0x1	CPU1 (CM33)	Secure
0x2000_0000 + TCMSABARS to 0x2000_FFFF	CPU1 (CM33)	Non-secure
0x2A01_0000 to 0x2A01_0000 + TCMSABARS - 0x1	Bus master	Secure
0x2A01_0000 + TCMSABARS to 0x2A01_FFFF	Bus master	Non-secure
0x3000_0000 to 0x3000_0000 + TCMSABARS - 0x1	CPU1 (CM33)	Secure
0x3000_0000 + TCMSABARS to 0x3000_FFFF	CPU1 (CM33)	Non-secure
0x3A01_0000 to 0x3A01_0000 + TCMSABARS - 0x1	Bus master	Secure
0x3A01_0000 + TCMSABARS to 0x3A01_FFFF	Bus master	Non-secure

2.17.3.3 TCMPRCR_S : TCM Protection Control Register for Secure

Base address: TCM = 0x4001_C800

Offset address: 0x00

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

Bit	Symbol	Function	R/W
0	PR	Register Write Control 0: Write registers are disabled 1: Write registers are enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KW[7:0]	Write Key Code Key code protection to the PR bit.	R/W

Note: It is necessary to write by half word access.
Byte write access is prohibited. When byte write access is executed, operation is not guaranteed.

Security: S-TYPE-1

Privilege: P-TYPE-1

PR bit (Register Write Control)

The PR bit controls the write mode of the TCMCRx registers that are marked as Secure by TCMSAR register. When this bit is set to 1, writing to the TCMCRx marked as Secure is enabled.

KW[7:0] bits (Write Key Code)

The KW[7:0] bits enable or disable writes to the PR bit. When writing to the PR bit, write 0xA5 to the KW[7:0] bits simultaneously. When a value other than 0xA5 is written to KW[7:0], the PR bit is not updated. The KW[7:0] bits are always read as 0x00.

2.17.3.4 TCMPRCR_NS : TCM Protection Control Register for Non-Secure

Base address: TCM_NS = 0x5001_C800

Offset address: 0x04

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

Bit	Symbol	Function	R/W
0	PR	Register Write Control 0: Write registers are disabled 1: Write registers are enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KW[7:0]	Write Key Code Key code protection to the PR bit.	R/W

Note: It is necessary to write by half word access.
Byte write access is prohibited. When byte write access is executed, operation is not guaranteed.

Security: S-TYPE-1

Privilege: P-TYPE-1

PR bit (Register Write Control)

The PR bit controls the write mode of the TCMCRx registers that are marked as Non-secure by TCMSAR register. When this bit is set to 1, writing to the TCMCRx marked as Non-secure is enabled.

KW[7:0] bits (Write Key Code)

The KW[7:0] bits enable or disable writes to the PR bit. When writing to the PR bit, write 0xA5 to the KW[7:0] bits simultaneously. When a value other than 0xA5 is written to KW[7:0], the PR bit is not updated. The KW[7:0] bits are always read as 0x00.

2.17.3.5 TCMCRx : TCM Control Register (x = C or S)

Base address: TCM = 0x4001_C800
TCM_NS = 0x5001_C800

Offset address: 0x10 or 0x14

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TSTBYP	—	—	E1STSEN	ECCMOD[1:0]		—	OAD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OAD	Operation after ECC error detection 0: Interrupt 1: Reset	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
3:2	ECCMOD[1:0]	ECC Operating Mode Select 0 0: Disable ECC function 0 1: Setting prohibited 1 0: Enable ECC function without error checking 1 1: Enable ECC function with error checking	R/W
4	E1STSEN	ECC 1-Bit Error Information Update Enable 0: Disable updating of 1-bit ECC error information 1: Enable updating of 1-bit ECC error information	R/W

Bit	Symbol	Function	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	TSTBYP	ECC Test Enable/ECC Bypass Select 0: Disable ECC bypass 1: Enable ECC bypass	R/W

Security: S-TYPE-3 (TCMSAR.TCMSA)

Privilege: P-TYPE-2

The protection register (TCMPRCR_S or TCMPRCR_NS) protects this register against writing. This register can be written only when the PR bit in the TCMPRCR_S or TCMPRCR_S register is 1.

Do not write to this register while access to TCM is in progress.

If CPU0 and CPU1 issue speculative access to an uninitialized TCM with ECC enabled, it may trigger an interrupt or reset. After enabling ECC function with error checking, initialize the TCM before setting the interrupt in the IELSRn register or setting the reset in the OAD bit.

OAD bit (Operation after ECC error detection)

The OAD bit selects whether to generate a reset or an interrupt when ECC error is detected.

ECCMOD[1:0] bits (ECC Operating Mode Select)

The ECCMOD[1:0] bits set the access mode to the ECC area.

E1STSEN bit (ECC 1-Bit Error Information Update Enable)

The E1STSEN bit enables or disables updating of the ECC 1-bit error status TCMESR.ERRC0, TCMESR.ERRS0.

When TCMCRx.E1STSEN = 0, the operation after error (such as NMI or reset) is not performed regardless of the value TCMCRx.OAD, even when ECC 1-bit error correction is done.

TSTBYP bit (ECC Test Enable/ECC Bypass Select)

The TSTBYP bit enables direct access to ECC code by bypassing the ECC function.

The ECC bypass function is used along with, setting the ECCMOD[1:0] bits in the same register to 00b.

The ECC code length is 7 bits for 32-bit data. When TSTBYP bit is 1, bits [6:0] of read data is ECC code, bits [31:7] of read data is undefined, bits [6:0] of write data are written to the ECC region of the address, bits [31:7] of write data are ignored.

For details about ECC test, see section 2.17.4.1. ECC Function.

2.17.3.6 TCMESR : TCM Error Status Register

Base address: TCM = 0x4001_C800
TCM_NS = 0x5001_C800

Offset address: 0x40

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	ERRS1	ERRS0	ERRC1	ERRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ERRC0	C-TCM 1-bit ECC Error Status 0: 1-bit ECC error has not occurred 1: 1-bit ECC error has occurred	R
1	ERRC1	C-TCM 2-bit ECC Error Status 0: 2-bit ECC error has not occurred 1: 2-bit ECC error has occurred	R

Bit	Symbol	Function	R/W
2	ERRS0	S-TCM 1-bit ECC Error Status 0: 1-bit ECC error has not occurred 1: 1-bit ECC error has occurred	R
3	ERRS1	S-TCM 2-bit ECC Error Status 0: 2-bit ECC error has not occurred 1: 2-bit ECC error has occurred	R
15:4	—	These bits are read as 0.	R

Note: This register is initialized after returning from Software Standby mode while the debugger is connected.

Security: S-TYPE-5

Privilege: P-TYPE-2

The TCMESR register is cleared by the corresponding bit in TCMESCLR register or resets other than Bus Error Reset and Memory Error Reset. Also during access from the debugger, the TCMESR register is not updated.

ERRx0 bit (x-TCM 1-bit ECC Error Status) (x = C or S)

The ERRx0 bit shows whether there is a 1-bit ECC error in x-TCM.

When ECC function with error checking is enabled (TCMCRx.ECCMOD = 2’b11) and, updating of the 1-bit error information is enabled (TCMCRx.E1STSEN = 1), the ERRx0 bit is set to 1 if a 1-bit error is detected. When ERRx0 bit is set to 1, a reset or an interrupt request is generated according to the TCMCRx.OAD.

ERRx1 bit (x-TCM 2-bit ECC Error Status) (x = C or S)

The ERRx1 bit shows whether there is a 2-bit ECC error in x-TCM.

When ECC function with error checking is enabled (TCMCRx.ECCMOD = 2’b11), the ERRx1 bit is set to 1 if a 2-bit error is detected.

When ERRx1 bit is set to 1, a reset or an interrupt request is generated according to the TCMCRx.OAD.

If the status is cleared and an error occurs at the same time, the error is ignored and the status is cleared.

2.17.3.7 TCMESCLR : TCM Error Status Clear Register

Base address: TCM = 0x4001_C800
TCM_NS = 0x5001_C800

Offset address: 0x48

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	CLRS1	CLRS0	CLRC1	CLRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CLRC0	C-TCM 1-bit ECC Error Status Clear Writing to the CLRC0 clears TCMESR.ERRC0 and TCMEARC0.EAR. 1: Clear error	R/W*1
1	CLRC1	C-TCM 2-bit ECC Error Status Clear Writing to the CLRC1 clears TCMESR.ERRC1 and TCMEARC1.EAR. 1: Clear error	R/W*1
2	CLRS0	S-TCM 1-bit ECC Error Status Clear Writing to the CLRS0 clears TCMESR.ERRS0 and TCMEARS0.EAR. 1: Clear error	R/W*1
3	CLRS1	S-TCM 2-bit ECC Error Status Clear Writing to the CLRS1 clears TCMESR.ERRS1 and TCMEARS1.EAR. 1: Clear error	R/W*1
15:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only writing 1 is valid. Writing value is not retained, read value is 0.

Security: S-TYPE-3 (TCMSAR.TCMSA)

Privilege: P-TYPE-2

CLRx0 bit (x-TCM 1-bit ECC Error Status Clear) (x = C or S)

The CLRx0 bit can clear 1-bit ECC error status bit in the TCMESR.

TCMESR.ERRx0 can be cleared when writing 1 to CLRx0.

CLRx1 bit (x-TCM 2-bit ECC Error Status Clear) (x = C or S)

The CLRx1 bit can clear 2-bit ECC error status bit in the TCMESR.

TCMESR.ERRx1 can be cleared when writing 1 to CLRx1.

If the status is cleared and an error occurs at the same time, the error is ignored and the status is cleared.

2.17.3.8 TCMEARxm : TCM Error Address Register (x = C or S) (m = 0 to 1)

Base address: TCM = 0x4001_C800
TCM_NS = 0x5001_C800

Offset address: 0x50, 0x54, 0x58 or 0x5C

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EAR[17:2]														—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
17:2	EAR[17:2]	When an SRAM error occurs, an error address is stored.	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note: This register is initialized after returning from Software Standby mode while the debugger is connected.

Security: S-TYPE-5

Privilege: P-TYPE-2

This register is cleared by the corresponding bit in TCMESCLR register, or resets other than Bus Error Reset and Memory Error Reset. Also during access from the debugger, this register is not updated.

TCMEARx0 bits

This register stores the error address where 1-bit ECC error is detected. These bits hold the error address that occurred first. These bits are cleared by clearing 1-bit ECC error from TCMESCLR.CLRx0.

TCMEARx1 bits

This register stores the error address where 2-bit ECC error is detected. These bits hold the error address that occurred first. These bits are cleared by clearing 2-bit ECC error from TCMESCLR.CLRx1.

If the status is cleared and an error occurs at the same time, the error is ignored and the status is cleared.

2.17.4 Operation

2.17.4.1 ECC Function

2.17.4.1.1 Correction of ECC Errors

Enabling and disabling of ECC error correction can be selected through the setting by the TCMCRx.ECCMOD[1:0] bits. In the initial state, ECC error correction is disabled. The ECC check type is SEC-DED (Single-Error Correction and Double-Error Detection Code).

When the ECC function is enabled, 7-bit ECC code is generated from 32-bit data. Writes to the TCM are 32-bit data and 7-bit ECC code. For reads, 32-bit data and 7-bit ECC code are read out from the TCM.

When ECC function with error checking is enabled, if a 1-bit error occurs, error correction is performed and the TCMESR.ERRx0 bit is set to 1, if the TCMCRx.E1STSEN bit is 1. If a 2-bit error occurs, error detection is performed and the TCMESR.ERRx1 bit is set to 1, although error correction is not performed.

When ECC function without error checking is enabled, if a 1-bit error occurs, error correction is performed but the TCMESR.ERRx0 bit is not updated even though the TCMCRx.E1STSEN bit is 1. If a 2-bit error occurs, error detection is performed but the TCMESR.ERRx1 bit is not updated, and error correction is not performed.

When the ECC function is disabled, neither error correction nor error detection is performed even though 1-bit or 2-bit error occurs. Therefore, the TCMESR.ERRx0 bit and the TCMESR.ERRx1 bit are not updated.

When updating the data after the occurrence of an error, the only support of 32-bit data writing. Since the TCM data is undefined after power on and release from Deep Software Standby mode, accessing the TCM when the ECC function is enabled and error checking is selected causes an ECC error to occur. Therefore, before using ECC function, initial writing with 32-bit data size, or initial writing with ECC function enabled and error checking disabled, to the area to be used in the TCM should be performed.

2.17.4.1.2 ECC Error Interrupt Function

When the ECC function with error checking is enabled, error checking is possible in the TCM. For a 2-bit error, the TCMESR.ERRx1 bit set to 1. For a 1-bit error, the TCMESR.ERRx0 bit set to 1 when TCMCRx.E1STSEN bit is 1.

When the ECC 1-bit error is to be masked, it is necessary to set the TCMCRx.E1STSEN bit to 0 to disable the setting of TCMESR.ERRx0 bit to 1. An ECC error is not generated while the ECC function is disabled or the ECC function without error checking is enabled.

An ECC error may result in a request for a reset or an interrupt. When an ECC error occurs and when the TCMCRx.OAD bit is 1, a reset request is output. When the TCMCRx.OAD bit is 0, an interrupt request is output to the ICU.

An ECC error interrupt occurs when an error status in the TCMESR register is set to 1. ECC error interrupt continues to occur until the error status in the TCMESR register is cleared.

2.17.4.1.3 ECC Decoder Testing

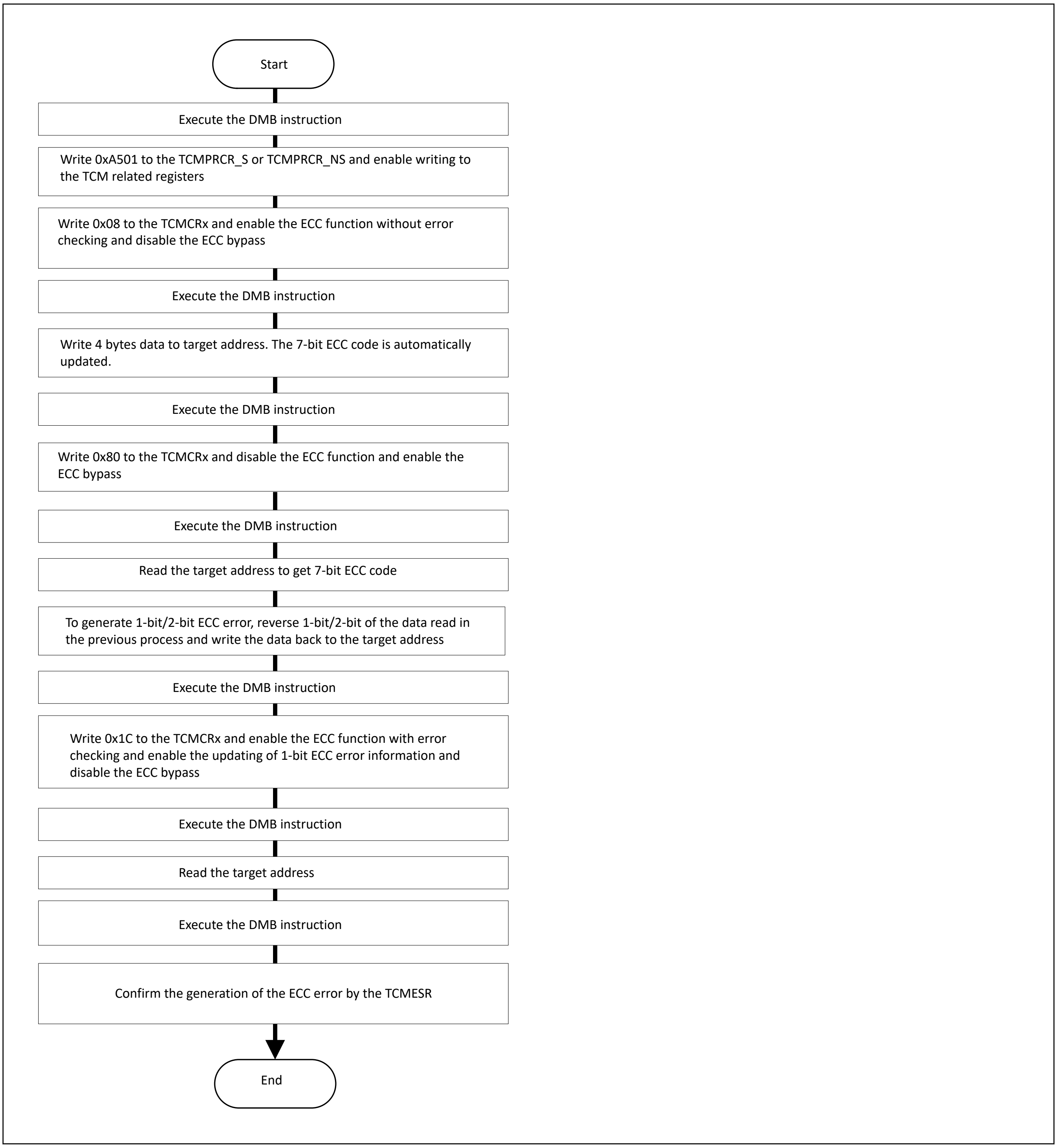


Figure 2.12 ECC decoder testing flowchart

2.17.4.2 TrustZone Filter Function

2.17.4.2.1 TrustZone Filter for TCM Registers

TCM registers can be protected with a Security Attribution (SA) from non-secure or secure transaction. When SA indicates that TCM registers are secure status, non-secure transaction cannot access them because TrustZone Filter detects an error and protects the access. When SA indicates that TCM registers are non-secure status, secure transaction cannot access them because TrustZone Filter detects an error and protects the access. SA for TCM registers is one to be used commonly among TCM registers.

For a failed access due to a TrustZone error, an error response is generated.

SA	Transaction	Write access	Read access
Secure	Secure	Permitted	Permitted
	Non-secure	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)
Non-secure	Secure	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)
	Non-secure	Permitted	Permitted

2.17.4.2.2 TrustZone Filter for TCM Memory Regions

C-TCM and S-TCM regions can divide into secure, non-secure independently. The access permissions for these regions are as follows.

SA	Transaction	Write access	Read access
Secure	Secure	Permitted	Permitted
	Non-secure	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)
Non-secure	Secure	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)
	Non-secure	Permitted	Permitted

A TrustZone Filter error for TCM memory, generates an error notification.

For details, see section 51, Security Features.

2.17.4.3 Interrupt Source

The TCM interrupt source includes an ECC error and a TrustZone Filter error.

ECC error can choose a reset request or an interrupt request by the TCMCRx.OAD bit.

The TCM interrupt occurs when an error status in the TCMESR register is set to 1. The TCM interrupt continues to occur until the error status in the TCMESR register is cleared.

When local memory error occurs (NMISR.LMST = 1 or RSTSR1.LM1RF = 1), read TCMESR and check the TCM interrupt source. When access is from the debugger, the error is detected and corrected, but no error status is set, and no reset request or interrupt request is output.

For more details on Debug mode, see section 2, CPU.

Table 2.41 TCM interrupt source

Name	Interrupt source	DTC activation	DMAC activation
ECCERR	ECC error (x-TCM with ECC)	Not possible	Not possible
TZFLT	TrustZone Filter error	Not possible	Not possible

2.17.4.4 Wait Cycle

For write access with ECC function enabled (TCMCRx.ECCMOD[1] = 1) and access size is other than 32-bit (for example, 8-bit or 16-bit), a wait cycle occurs.

If access of the bus master conflicts with access of CPU1, the bus interface of TCM returns a wait cycle to the bus master until CPU1 finishes the access, because CPU1 has priority.

If there is a security breach in CPU1 or access of the bus master to TCM, TCM responds by inserting a wait cycle with an AHB error. In other case, no wait cycle is required.

2.17.5 Usage Notes

2.17.5.1 Note on TCM Memory Cell Test

Because the TCM has a write buffer, if reading from the same address after writing, data in the write buffer may be read instead of the TCM memory cell. When reading data written for testing of the TCM memory cell, insert the write according to the following procedure to avoid reading from a write buffer.

1. Write data to the test target address.
2. Inside the same TCM, write to an address 4 or more away from the address in step 1.
3. Read the data from the address in step 1.

3. Inter-Processor Communication (IPC)

3.1 Overview

Inter-Processor Communication (IPC) supports hardware sharing and communication between two processors. IPC can generate interrupt events to support communication between each processor.

Several semaphore mechanisms are required to achieve mutually exclusive access between processors. To achieve this, IPC implements multiple locks. Application developers select to use separate or shared hardware semaphores between the two CPUs. A mechanism is also required for simple data communication between CPUs. For example, a FIFO that CPU0 uses for write-only data and CPU1 uses for read-only data. In addition, for mutual use, a FIFO is required with CPU1 as write-only data and CPU0 as read-only data. Table 3.1 lists the IPC specifications.

Table 3.1 IPC specifications

Item		Description
Semaphore	Lock information	• A total of 16 factors • Security attributes and privileged attributes are selected every 8 factors
Inter-Processor interrupt	Non-maskable	• IPC0: one factor, one non-maskable interrupt, security attributes and privileged attributes are selected. • IPC1: one factor, one non-maskable interrupt, security attributes and privileged attributes are selected.
	Maskable	• IPC00: 8 factors, one maskable interrupt, security attributes and privileged attributes are selected[*1] • IPC01: 8 factors, one maskable interrupt, security attributes and privileged attributes are selected[*1] • IPC10: 8 factors, one maskable interrupt, security attributes and privileged attributes are selected[*1] • IPC11: 8 factors, one maskable interrupt, security attributes and privileged attributes are selected[*1]
Message-FIFO	FIFO	• Message FIFO 00 (CPU1 -> CPU0), one maskable interrupt, security attributes and privileged attributes are selected[*1] • Message FIFO 01 (CPU1 -> CPU0), one maskable interrupt, security attributes and privileged attributes are selected[*1] • Message FIFO 10 (CPU0 -> CPU1), one maskable interrupt, security attributes and privileged attributes are selected[*1] • Message FIFO 11 (CPU0 -> CPU1), one maskable interrupt, security attributes and privileged attributes are selected[*1] • 4 FIFO stages • Transfer data size 32 bits
	Error information	• Write with FIFO FULL[*1] • Read with FIFO Empty[*1]

Note 1. The interrupts for IPC00, Message FIFO 00 and Error information are common. The same applies to others.

3.2 Register Descriptions

IPC_BASE (base address of IPC) is 0x4002_0000 (Secure) and 0x5002_0000 (Non-secure).

CPSCU_BASE (base address of CPSCU) is 0x4000_8000 (Secure) and 0x5000_8000 (Non-secure).

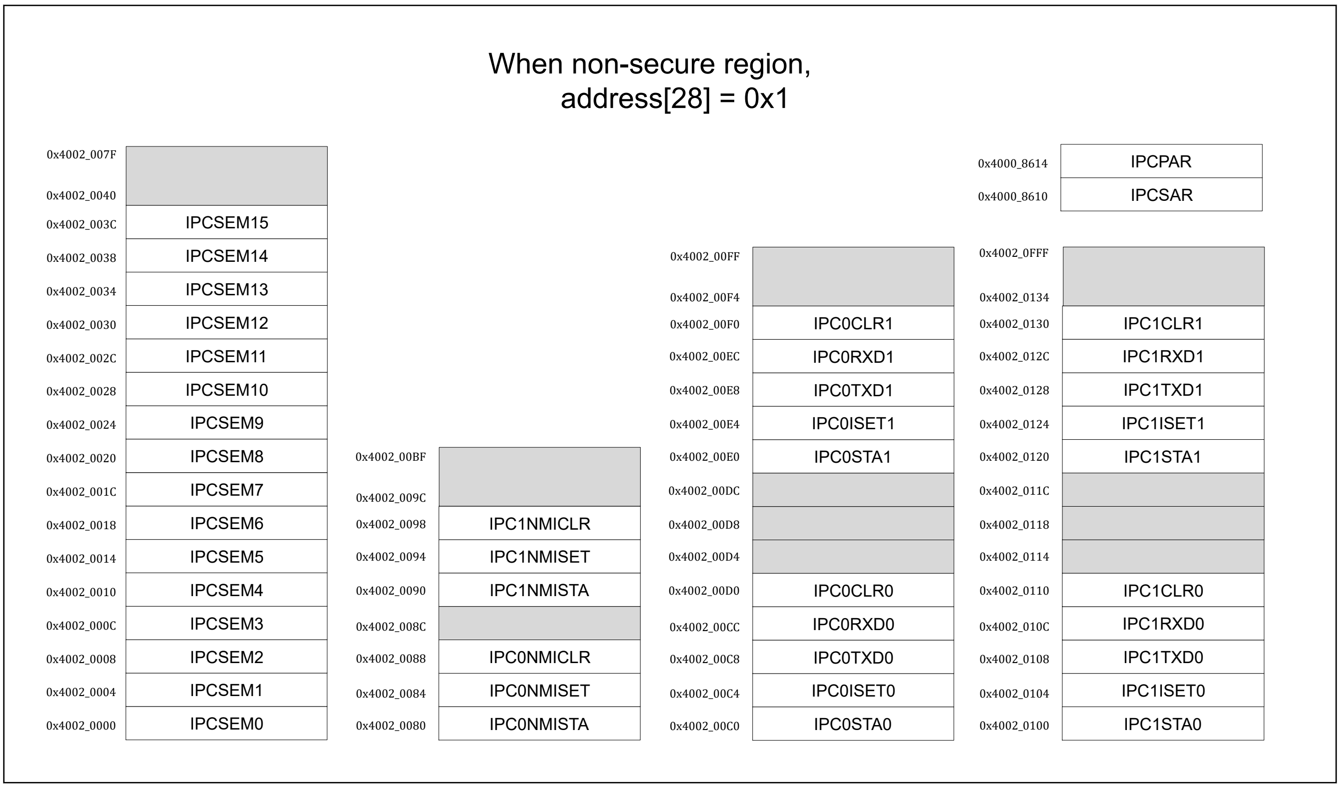


Figure 3.1 IPC register address map

3.2.1 IPCSAR : IPC Security Attribution Register

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x610

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	SAIPCIR3	SAIPCIR2	SAIPCIR1	SAIPCIR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SAIPCNMI1	SAIPCNMI0	—	—	—	—	—	—	SAIPCSEM1	SAIPCSEM0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SAIPCSEM0	Security attributes of registers for IPCSEMn (n = 0 to 7) Security attributes of IPCPAR.PAIPCSEM0 0: Secure 1: Non-secure	R/W
1	SAIPCSEM1	Security attributes of registers for IPCSEMn (n = 8 to 15) Security attributes of IPCPAR.PAIPCSEM1 0: Secure 1: Non-secure	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	SAIPCNMI0	Security attributes of IPC0NMISTA, IPC0NMISET and IPC0NMICLR Security attributes of IPCPAR.PAIPCNMI0 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
9	SAIPCNMI1	Security attributes of IPC1NMISTA, IPC1NMISET and IPC1NMICLR Security attributes of IPCPAR.PAIPCNMI1 0: Secure 1: Non-secure	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	SAIPCIR0	Security attributes of registers for IPC0STA0, IPC0ISET0, IPC0TXD0, IPC0RXD0 and IPC0CLR0 Security attributes of IPCPAR.PAIPCIR0 0: Secure 1: Non-secure	R/W
17	SAIPCIR1	Security attributes of registers for IPC0STA1, IPC0ISET1, IPC0TXD1, IPC0RXD1 and IPC0CLR1 Security attributes of IPCPAR.PAIPCIR1 0: Secure 1: Non-secure	R/W
18	SAIPCIR2	Security attributes of registers for IPC1STA0, IPC1ISET0, IPC1TXD0, IPC1RXD0 and IPC1CLR0 Security attributes of IPCPAR.PAIPCIR2 0: Secure 1: Non-secure	R/W
19	SAIPCIR3	Security attributes of registers for IPC1STA1, IPC1ISET1, IPC1TXD1, IPC1RXD1 and IPC1CLR1 Security attributes of IPCPAR.PAIPCIR3 0: Secure 1: Non-secure	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

This register is write-protected by PRCR_S.PRC4.

SAIPCSEM0 bit

Security attributes of registers. The target registers are as follow.

- IPCSEM0 to IPCSEM7 registers
- IPCPAR.PAIPCSEM0 bit

SAIPCSEM1 bit

Security attributes of registers. The target registers are as follow.

- IPCSEM8 to IPCSEM15 registers
- IPCPAR.PAIPCSEM1 bit

SAIPCNMI0 bit

Security attributes of registers. The target registers are as follow.

- IPC0NMISTA register
- IPC0NMISET register
- IPC0NMICLR register
- IPCPAR.PAIPCNMI0 bit

SAIPCNMI1 bit

Security attributes of registers. The target registers are as follow.

- IPC1NMISTA register
- IPC1NMISET register
- IPC1NMICLR register
- IPCPAR.PAIPCNMI1 bit

SAIPCIR0 bit

Security attributes of registers. The target registers are as follow.

- IPC0STA0 register
- IPC0ISET0 register
- IPC0TXD0 register
- IPC0RXD0 register
- IPC0CLR0 register
- IPCPAR.PAIPCIR0 bit

SAIPCIR1 bit

Security attributes of registers. The target registers are as follow.

- IPC0STA1 register
- IPC0ISET1 register
- IPC0TXD1 register
- IPC0RXD1 register
- IPC0CLR1 register
- IPCPAR.PAIPCIR1 bit

SAIPCIR2 bit

Security attributes of registers. The target registers are as follow.

- IPC1STA0 register
- IPC1ISET0 register
- IPC1TXD0 register
- IPC1RXD0 register
- IPC1CLR0 register
- IPCPAR.PAIPCIR2 bit

SAIPCIR3 bit

Security attributes of registers. The target registers are as follow.

- IPC1STA1 register
- IPC1ISET1 register
- IPC1TXD1 register
- IPC1RXD1 register
- IPC1CLR1 register
- IPCPAR.PAIPCIR3 bit

3.2.2 IPCPAR : IPC Privileged Attribution Register

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x614

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	PAIPC IR3	PAIPC IR2	PAIPC IR1	PAIPC IR0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PAIPC NMI1	PAIPC NMI0	—	—	—	—	—	—	PAIPC SEM1	PAIPC SEM0
Value after reset:	x	x	x	x	x	x	1	1	x	x	x	x	x	x	1	1

Bit	Symbol	Function	R/W
0	PAIPCSEM0	Privileged attributes of registers for IPCSEMn (n = 0 to 7) 0: Privileged 1: Unprivileged	R/W
1	PAIPCSEM1	Privileged attributes of registers for IPCSEMn (n = 8 to 15) 0: Privileged 1: Unprivileged	R/W
7:2	—	The read values are undefined. The write value should be 1.	R/W
8	PAIPCNMI0	Privileged attributes of registers for IPC0NMISTA, IPC0NMISET and IPC0NMICLR 0: Privileged 1: Unprivileged	R/W
9	PAIPCNMI1	Privileged attributes of registers for IPC1NMISTA, IPC1NMISET and IPC1NMICLR 0: Privileged 1: Unprivileged	R/W
15:10	—	The read values are undefined. The write value should be 1.	R/W
16	PAIPCIR0	Privileged attributes of registers for IPC0STA0, IPC0ISET0, IPC0TXD0, IPC0RXD0 and IPC0CLR0 0: Privileged 1: Unprivileged	R/W
17	PAIPCIR1	Privileged attributes of registers for IPC0STA1, IPC0ISET1, IPC0TXD1, IPC0RXD1 and IPC0CLR1 0: Privileged 1: Unprivileged	R/W
18	PAIPCIR2	Privileged attributes of registers for IPC1STA0, IPC1ISET0, IPC1TXD0, IPC1RXD0 and IPC1CLR0 0: Privileged 1: Unprivileged	R/W
19	PAIPCIR3	Privileged attributes of registers for IPC1STA1, IPC1ISET1, IPC1TXD1, IPC1RXD1 and IPC1CLR1 0: Privileged 1: Unprivileged	R/W
31:20	—	The read values are undefined. The write value should be 1.	R/W

Note: S-TYPE-2, P-TYPE-1

The Security Attribution of bits of this register is determined by the corresponding bits of IPCSAR.

This register is the write protected bit by bit according to the security attribute setting.

Secure bits is write-protected by PRCR_S.PRC4.

Non-secure bits is write-protected by PRCR_NS.PRC4.

PAIPCSEM0 bit

Privileged attributes of registers. The target registers are as follow.

- IPCSEM0 to IPCSEM7 registers

PAIPCSEM1 bit

Privileged attributes of registers. The target registers are as follow.

- IPCSEM8 to IPCSEM15 registers

PAIPCNMI0 bit

Privileged attributes of registers. The target registers are as follow.

- IPC0NMISTA register
- IPC0NMISET register
- IPC0NMICLR register

PAIPCNMI1 bit

Privileged attributes of registers. The target registers are as follow.

- IPC1NMISTA register
- IPC1NMISET register
- IPC1NMICLR register

PAIPCIR0 bit

Privileged attributes of registers. The target registers are as follow.

- IPC0STA0 register
- IPC0ISET0 register
- IPC0TXD0 register
- IPC0RXD0 register
- IPC0CLR0 register

PAIPCIR1 bit

Privileged attributes of registers. The target registers are as follow.

- IPC0STA1 register
- IPC0ISET1 register
- IPC0TXD1 register
- IPC0RXD1 register
- IPC0CLR1 register

PAIPCIR2 bit

Privileged attributes of registers. The target registers are as follow.

- IPC1STA0 register
- IPC1ISET0 register
- IPC1TXD0 register
- IPC1RXD0 register
- IPC1CLR0 register

PAIPCIR3 bit

Privileged attributes of registers. The target registers are as follow.

- IPC1STA1 register
- IPC1ISET1 register
- IPC1TXD1 register

- IPC1RXD1 register
- IPC1CLR1 register

3.2.3 IPCSEMn : Semaphore Register n (n = 0 to 15)

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x000 + 0x4 × n

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

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

Bit	Symbol	Function	R/W
0	LOCK	This bit indicates the shared resource is locked. Set condition : Reading this register. Clear condition : Writing 1 to this bit. 0: Unlocked 1: Locked	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: The LOCK bit is not set for reads with halfword access or byte access that do not include the b0.
Note: S-TYPE-3, P-TYPE-3

The IPC has the registers to perform semaphore for synchronization and exclusive control between CPU0 and CPU1. This register only indicates the status and does not have any hardware protection of shared memory. Thus, the exclusive control must be done by software that uses this register.

The security of this register depends on IPCSAR. The privilege of this register depends on IPCPAR.

LOCK bit (Shared resource is locked for Semaphore Control)

The LOCK bit of this register is used to achieve mutually exclusive access between processors. By using this bit, the application developer can select whether to use the hardware semaphore separately or in common between the two processors.

3.2.4 IPC0NMISTA : Inter-Processor0 NMI Request Status Register

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x080

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

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

Bit	Symbol	Function	R/W
0	NMI	This bit indicates the status of interrupt request. 0: Not requesting interrupt 1: Requesting interrupt	R
31:1	—	These bits are read as 0. The write value should be 0.	R

Note: S-TYPE-3, P-TYPE-3

The IPC0NMISTA register indicates that a non-maskable interrupt is issued from CPU0 to CPU1. The security of this register depends on IPCSAR. The privilege of this register depends on IPCPAR.

NMI bit (NMI Request Set)

The NMI bit indicates an NMI interrupt request is issued from CPU0 to CPU1.

3.2.5 IPC0NMISET : Inter-Processor0 NMI Request Set Register

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x084

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

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

Bit	Symbol	Function	R/W
0	SET	Writing 1 to the bit sets IPC0NMISTA.NMI.	W
31:1	—	These bits are read as 0. The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The IPC0NMISET register is used to issue a non-maskable interrupt from CPU0 to CPU1. The security of this register depends on IPCSAR. The privilege of this register depends on IPCPAR.

SET bit (NMI Request Set)

When 1 is written, IPC0NMISTA.NMI is set to 1 and an NMI interrupt is issued to CPU0.

3.2.6 IPC0NMICLR : Inter-Processor0 NMI Request Clear Register

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x088

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

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

Bit	Symbol	Function	R/W
0	CLR	Writing 1 to this bit clears IPC0NMISTA.NMI.	W
31:1	—	These bits are read as 0. The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The IPC0NMICLR register is used for canceling non-maskable interrupt from CPU1 to CPU0. The security of this register depends on IPCSAR. The privilege of this register depends on IPCPAR.

CLR bit (NMI Request Clear)

When 1 is written, IPC0NMISTA.NMI is clear to 0 and the NMI interrupt to CPU0 is canceled.

3.2.7 IPC1NMISTA : Inter-Processor1 NMI Request Status Register

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x090

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

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

Bit	Symbol	Function	R/W
0	NMI	This bit indicates the status of interrupt request. 0: Not requesting interrupt 1: Requesting interrupt	R
31:1	—	These bits are read as 0. The write value should be 0.	R

Note: S-TYPE-3, P-TYPE-3

The IPC1NMISTA register indicates that a non-maskable interrupt is issued from CPU0 to CPU1. The security of this register depends on IPCSAR. The privilege of this register depends on IPCPAR.

NMI bit (NMI Request Status)

The NMI bit indicates an NMI interrupt request is issued from CPU0 to CPU1.

3.2.8 IPC1NMISET : Inter-Processor1 NMI Request Set Register

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x094

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

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

Bit	Symbol	Function	R/W
0	SET	Writing 1 to this bit sets IPC1NMISTA.NMI.	W
31:1	—	These bits are read as 0. The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The
IPC1NMISET

register is used to issue a non-maskable interrupt from CPU0 to CPU1. The security of this register depends on IPCSAR. The privilege of this register depends on IPCPAR.

SET bit (NMI Request Set)

When 1 is written, IPC1NMISTA.NMI is set to 1 and an NMI interrupt is issued to CPU1.

3.2.9 IPC1NMICLR : Inter-Processor1 NMI Request Clear Register

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x098

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

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

Bit	Symbol	Function	R/W
0	CLR	Writing 1 to this bit clears IPC1NMISTA.NMI.	W
31:1	—	These bits are read as 0. The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The
IPC1NMICLR

register is used to cancel non-maskable interrupt from CPU0 to CPU1. The security of this register depends on IPCSAR. The privilege of this register depends on IPCPAR.

CLR bit (NMI Request Clear)

When 1 is written, IPC1NMISTA.NMI is clear to 0 and the NMI interrupt to CPU1 is canceled.

3.2.10 IPC0STA0 : Inter-Processor0 Status Register0

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x0C0

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	IRQ7 to IRQ0	This bit indicates the status of interrupt request. 0: Not requesting interrupt 1: Requesting interrupt	R
15:8	—	These bits are read as 0. The write value should be 0.	R
16	RDY	This bit is set when FIFO is not empty. This bit is used for the interrupt request. 0: FIFO 00 data is not ready 1: FIFO 00 data is ready	R
17	FULL	0: FIFO 00 is not full 1: FIFO 00 is full	R
23:18	—	These bits are read as 0. The write value should be 0.	R
24	RERR	Indicates that the message FIFO 00 tried to read data despite being empty. 0: Not Error. Not requesting interrupt. 1: Error. Requesting interrupt.	R
25	FERR	Indicates that the message FIFO 00 tried to send more data even though it was full. 0: Not Error. Not requesting interrupt. 1: Error. Requesting interrupt.	R
31:26	—	These bits are read as 0. The write value should be 0.	R

Note: S-TYPE-3, P-TYPE-3

This register indicates the status of the maskable interrupt IPC0IRQ0 from CPU1 to CPU0. When any of IRQn (n = 0 to 7), RDY, RERR, or FERR is 1, the IPC0IRQ0 interrupt is issued. Read this register in the interrupt handler to determine which interrupt has occurred. The security attributes of this register are controlled by IPCSAR. The privileged attribute of this register is controlled by IPCPAR.

IRQn bit (Interrupt Request Status n (n = 0 to 7))

These bits indicate an interrupt to CPU0. Writing "1" to IPC0ISET0.SETn sets IRQn to "1". Writing "1" to IPC0CLR0.CLRn clears IRQn to "0". If any bit is "1", an IPC0IRQ0 interrupt request is issued.

RDY bit (Receive Data Ready)

When this bit is "1", it indicates that there is data received in the message FIFO 00 for CPU0. When the message FIFO is empty, this bit is "0". When this bit is "1", the IPC0IRQ0 interrupt request is issued.

FULL bit (FIFO Full)

The depth of message FIFO 00 is 4. When this bit is "1", it indicates that the message FIFO 00 is full.

RERR bit (RDY Error Status)

When reading IPC0RXD0 while the RDY bit is "0", this bit sets to "1". Writing "1" to the IPC0CLR0.RCLR bit clears this bit to "0". When this bit is "1", the IPC0IRQ0 interrupt request is issued.

FERR bit (FULL Error Status)

When writing to IPC0TXD0 while the FULL bit is "1", this bit sets to "1". Writing "1" to the IPC0CLR0.FCLR bit clears this bit to "0". When this bit is "1", the IPC0IRQ0 interrupt request is issued.

3.2.11 IPC0ISET0 : Inter-Processor0 IRQ Request Set Register0

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x0C4

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	SET7	SET6	SET5	SET4	SET3	SET2	SET1	SET0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	SETn (n = 0 to 7)	Writing 1 to this bit sets IPC0STA0.IRQn	W
31:8	—	These bits are read as 0. The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The
IPC0ISET0

register is used to issue a maskable interrupt from CPU0 to CPU1. The IRQ is assigned eight events, but there is only one interrupt event. The security of this register depends on IPCSAR. The privilege of this register depends on IPCPAR.

SETn bit (Interrupt Request Set n (n = 0 to 7))

When 1 is written, IPC0STA0.IRQn is set to 1 and a maskable interrupt to CPU0 is issued.

3.2.12 IPC0TXD0 : Inter-Processor0 FIFO Transfer Data Register0

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x0C8

Bit position: 31 … 0

Bit field: TXD[31:0]

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TXD[31:0]	Transmit data for the Message FIFO 00	W

Note: Only writing with a 32-bit data size is valid. Halfword or byte access is ignored.
Note: S-TYPE-3, P-TYPE-3

This is a register for writing data to message FIFO 00 from CPU1 to CPU0. Write to this register by CPU1. The security attributes of this register are controlled by IPCSAR. The privileged attribute of this register is controlled by IPCPAR.

TXD[31:0] bits (Transmit data for the Message FIFO 00)

Writing to this register sets the transmit data in message FIFO 00 and the IPC0STA0.RDY bit becomes "1". If you write to this register while the IPC0STA0.FULL bit is "1", the write operation is ignored, and the IPC0STA0.FERR bit becomes "1".

3.2.13 IPC0RXD0 : Inter-Processor0 FIFO Receive Data Register0

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x0CC

Bit position:	31–0
Bit field:	RXD[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	RXD[31:0]	Received data from the message FIFO 00 Reading this register updates it with the next received data in the message FIFO 00.	R

Note: If a TRZ error occurs, the read is 0 and it does not update to the next data.
S-TYPE-3, P-TYPE-3

This register is for reading received data from the message FIFO 00 from CPU1 to CPU0. Read this register by CPU0. The security attributes of this register are controlled by IPCSAR. The privileged attribute of this register is controlled by IPCPAR.

RXD[31:0] bits (Received data from the message FIFO 00)

If this register is read while the IPC0STA0.RDY bit is "0", the IPC0STA0.RERR bit is set to "1".

3.2.14 IPC0CLR0 : Inter-Processor0 Clear Register0

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x0D0

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	CLR7	CLR6	CLR5	CLR4	CLR3	CLR2	CLR1	CLR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CLRn (n = 0 to 7)	Writing 1 to this bit clears IPC0STA0.IRQn.	W
15:8	—	These bits are read as 0. The write value should be 0.	W
16	RST	Writing 1 resets Message FIFO 00.	W
23:17	—	These bits are read as 0. The write value should be 0.	W
24	RCLR	Writing 1 resets IPC0STA0.RERR.	W
25	FCLR	Writing 1 resets IPC0STA0. FERR.	W
31:26	—	These bits are read as 0. The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The

IPC0CLR0

register is used to cancel the maskable interrupt from CPU1 to CPU0. The register can then be used to cancel the FIFO transfer. It can also clear the Error flag that occurred in the FIFO. The IRQ is assigned eight events, but there is only one interrupt event. The security of this register depends on IPCSAR. The privilege of this register depends on IPCPAR.

CLRn bits (Interrupt Request Clear n (n = 0 to 7))

When 1 is written, IPC0STA0.IRQn is clear to 0. All bits of IPC0STA0 are 0 and maskable interrupts to CPU0 are cancelled.

RST bit (FIFO Reset)

When 1 is written, the RST bit resets Message FIFO 00. The data stored in the FIFO becomes invalid and the following registers are cleared:

- IPC0STA0.FULL
- IPC0STA0.RDY

If there is no interrupt event other than the message FIFO, the interrupt request is canceled.

RCLR bit (Clear RDY Error Status 00)

The RCLR bit clears IPC0STA0.RERR.

FCLR bit (Clear FULL Error Status 00)

The FCLR bit clears IPC0STA0. FERR.

3.2.15 IPC0STA1 : Inter-Processor0 Status Register1

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x0E0

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	IRQn (n = 0 to 7)	This bit indicates the status of interrupt request. 0: Not requesting interrupt 1: Requesting interrupt	R
15:8	—	These bits are read as 0. The write value should be 0.	R
16	RDY	This bit is set when FIFO is not empty. This bit is used for the interrupt request. 0: FIFO 01 data is not ready 1: FIFO 01 data is ready	R
17	FULL	0: FIFO 01 is not full 1: FIFO 01 is full	R
23:18	—	These bits are read as 0. The write value should be 0.	R
24	RERR	Indicates that the message FIFO 01 tried to read data despite being empty. 0: Not Error. Not requesting interrupt. 1: Error. Requesting interrupt.	R

Bit	Symbol	Function	R/W
25	FERR	Indicates that the message FIFO 01 tried to send more data even though it was full. 0: Not Error. Not requesting interrupt. 1: Error. Requesting interrupt.	R
31:26	—	These bits are read as 0. The write value should be 0.	R

Note: S-TYPE-3, P-TYPE-3

This register indicates the status of the maskable interrupt IPC0IRQ1 from CPU1 to CPU0. When any of IRQn (n = 0 to 7), RDY, RERR, or FERR is '1', the IPC0IRQ1 interrupt is issued. Read this register in the interrupt handler to determine which interrupt has occurred. The security attributes of this register are controlled by IPCSAR. The privileged attribute of this register is controlled by IPCPAR.

IRQn bit (Interrupt Request Status n (n = 0 to 7))

These bits indicate an interrupt to CPU0. Writing "1" to IPC0ISET1.SETn sets IRQn to "1". Writing "1" to IPC0CLR1.CLRn clears IRQn to "0". If any bit is "1", an IPC0IRQ1 interrupt request is issued.

RDY bit (Receive Data Ready)

When this bit is "1", it indicates that there is data received in the message FIFO 01 for CPU0. When the message FIFO is empty, this bit is "0". When this bit is "1", the IPC0IRQ1 interrupt request is issued.

FULL bit (FIFO Full)

The depth of message FIFO 01 is 4. When this bit is "1", it indicates that the message FIFO 01 is full.

RERR bit (RDY Error Status)

When reading IPC0RXD1 while the RDY bit is "0", this bit sets to "1". Writing "1" to the IPC0CLR1.RCLR bit clears this bit to "0". When this bit is "1", the IPC0IRQ1 interrupt request is issued.

FERR bit (FULL Error Status)

When writing to IPC0TXD1 while the FULL bit is "1", this bit sets to "1". Writing "1" to the IPC0CLR1.FCLR bit clears this bit to "0". When this bit is "1", the IPC0IRQ1 interrupt request is issued.

3.2.16 IPC0ISET1 : Inter-Processor0 IRQ Request Set Register1

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x0E4

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	SET7	SET6	SET5	SET4	SET3	SET2	SET1	SET0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	SETn (n = 0 to 7)	Writing 1 to this bit sets IPC0STA1.IRQn.	W
31:8	—	These bits are read as 0. The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The
IPC0ISET1

register is used to issue a maskable interrupt from CPU1 to CPU0. The IRQ is assigned eight events, but there is only one interrupt event. The security of this register depends on IPCSAR. The privilege of this register depends on IPCPAR.

SETn bit (Interrupt Request Set n (n = 0 to 7))

When 1 is written, IPC0STA1.IRQn is set to 1 and a maskable interrupt to CPU0 is issued.

3.2.17 IPC0TXD1 : Inter-Processor0 FIFO Transfer Data Register1

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x0E8

Bit position:	31–0
Bit field:	TXD[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TXD[31:0]	Transmit data for the Message FIFO 01	W

Note: Only writing with a 32-bit data size is valid. Halfword or byte access is ignored.
Note: S-TYPE-3, P-TYPE-3

This is a register for writing data to message FIFO 01 from CPU1 to CPU0. Write to this register by CPU1. The security attributes of this register are controlled by IPCSAR. The privileged attribute of this register is controlled by IPCPAR.

TXD[31:0] bits (Transmit data for the Message FIFO 01)

Writing to this register sets the transmit data in message FIFO 01 and the IPC0STA1.RDY bit becomes "1". If you write to this register while the IPC0STA1.FULL bit is "1", the write operation is ignored, and the IPC0STA1.FERR bit becomes "1".

3.2.18 IPC0RXD1 : Inter-Processor0 FIFO Receive Data Register1

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x0EC

Bit position:	31–0
Bit field:	RXD[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	RXD[31:0]	Received data from the message FIFO 01 Reading this register updates it with the next received data in the message FIFO 01.	R

Note: If a TRZ error occurs, the read is 0 and it does not update to the next data.
S-TYPE-3, P-TYPE-3

This register is for reading received data from the message FIFO 01 from CPU1 to CPU0. Read this register by CPU0. The security attributes of this register are controlled by IPCSAR. The privileged attribute of this register is controlled by IPCPAR.

RXD[31:0] bits (Received data from the message FIFO 01)

If this register is read while the IPC0STA1.RDY bit is "0", the IPC0STA1.RERR bit is set to "1".

3.2.19 IPC0CLR1 : Inter-Processor0 Clear Register1

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x0F0

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	CLR7	CLR6	CLR5	CLR4	CLR3	CLR2	CLR1	CLR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CLRn (n = 0 to 7)	Writing 1 to this bit clears IPC0STA1.IRQn.	W
15:8	—	These bits are read as 0. The write value should be 0.	W
16	RST	Writing 1, then resets Message FIFO 01.	W
23:17	—	These bits are read as 0. The write value should be 0.	W
24	RCLR	Writing 1 resets IPC0STA1.RERR.	W
25	FCLR	Writing 1 resets IPC0STA1. FERR.	W
31:26	—	These bits are read as 0. The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The
IPC0CLR1

register is used to cancel the maskable interrupt from CPU1 to CPU0. The register can then be used to cancel the FIFO transfer. This register can also clear the Error flag that occurred in the FIFO. The IRQ is assigned eight events, but there is only one interrupt event. The security of this register depends on IPCSAR. The privilege of this register depends on IPCPAR.

CLRn bit (Interrupt Request Clear n (n = 0 to 7))

When 1 is written, IPC0STA1.IRQn is cleared to 0. All bits of IPC0STA1 are 0, and maskable interrupts to CPU0 are cancelled.

RST bit (FIFO Reset)

Writing 1 to the RST bit resets message FIFO 01. The data stored in the FIFO becomes invalid and the following registers are cleared:

- IPC0STA1.FULL
- IPC0STA1.RDY

If there is no interrupt event other than the message FIFO, the interrupt request is canceled.

RCLR bit (Clear RDY Error Status 01)

The RCLR bit clears IPC0STA1.RERR.

FCLR bit (Clear FULL Error Status 01)

The FCLR bit clears IPC0STA1. FERR.

3.2.20 IPC1STA0 : Inter-Processor1 Status Register0

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x100

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	IRQn (n = 0 to 7)	This bit indicates the status of interrupt request. 0: Not requesting interrupt 1: Requesting interrupt	R
15:8	—	These bits are read as 0. The write value should be 0.	R
16	RDY	This bit is set when FIFO is not empty. This bit is used for the interrupt request. 0: FIFO 10 data is not ready 1: FIFO 10 data is ready	R
17	FULL	0: FIFO 10 is not full 1: FIFO 10 is full	R
23:18	—	These bits are read as 0. The write value should be 0.	R
24	RERR	This bit indicates that the message FIFO 10 tried to read data despite being empty. 0: Not Error. Not requesting interrupt. 1: Error. Requesting interrupt.	R
25	FERR	This bit indicates that the message FIFO 10 tried to send more data even though it was full. 0: Not Error. Not requesting interrupt. 1: Error. Requesting interrupt.	R
31:26	—	These bits are read as 0. The write value should be 0.	R

Note: S-TYPE-3, P-TYPE-3

This register indicates the status of the maskable interrupt IPC1IRQ0 from CPU0 to CPU1. When any of IRQn (n = 0 to 7), RDY, RERR, or FERR is '1', the IPC1IRQ0 interrupt is issued. Read this register in the interrupt handler to determine which interrupt has occurred. The security attributes of this register are controlled by IPCSAR. The privileged attribute of this register is controlled by IPCPAR.

IRQn bit (Interrupt Request Status n (n = 0 to 7))

These bits indicate an interrupt to CPU1. Writing "1" to IPC1ISET0.SETn sets IRQn to "1". Writing "1" to IPC1CLR0.CLRn clears IRQn to "0". If any bit is "1", an IPC1IRQ0 interrupt request is issued.

RDY bit (Receive Data Ready)

When this bit is "1", it indicates that there is data received in the message FIFO 10 for CPU1. When the message FIFO is empty, this bit is "0". When this bit is "1", the IPC1IRQ0 interrupt request is issued.

FULL bit (FIFO Full)

The depth of message FIFO 10 is 4. When this bit is "1", it indicates that the message FIFO 10 is full.

RERR bit (RDY Error Status)

When reading IPC1RXD0 while the RDY bit is "0", this bit sets to "1". Writing "1" to the IPC1CLR0.RCLR bit clears this bit to "0". When this bit is "1", the IPC1IRQ0 interrupt request is issued.

FERR bit (FULL Error Status)

When writing to IPC1TXD0 while the FULL bit is "1", this bit sets to "1". Writing "1" to the IPC1CLR0.FCLR bit clears this bit to "0". When this bit is "1", the IPC1IRQ0 interrupt request is issued.

3.2.21 IPC1ISET0 : Inter-Processor1 IRQ Request Set Register0

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x104

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	SET7	SET6	SET5	SET4	SET3	SET2	SET1	SET0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	SETn (n = 0 to 7)	Writing 1 to this bit sets IPC1STA0.IRQn.	W
31:8	—	These bits are read as 0. The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The
IPC1ISET0

register is used to issue a maskable interrupt from CPU0 to CPU1. The IRQ is assigned eight events, but there is only one interrupt event. The security of this register depends on IPCSAR. The privilege of this register depends on IPCPAR.

SETn bit (Interrupt Request Set n (n = 0 to 7))

When 1 is written, IPC1STA0.IRQn is set to 1 and a maskable interrupt to CPU1 is issued.

3.2.22 IPC1TXD0 : Inter-Processor1 FIFO Transfer Data Register0

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x108

Bit position:	31 … 0
Bit field:	TXD[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TXD[31:0]	Transmit data for the Message FIFO 10	W

Note: Only writing with a 32-bit data size is valid. Halfword or byte access is ignored.
Note: S-TYPE-3, P-TYPE-3

This is a register for writing data to message FIFO 10 from CPU0 to CPU1. Write to this register by CPU0. The security attributes of this register are controlled by IPCSAR. The privileged attribute of this register is controlled by IPCPAR.

TXD[31:0] bits (Transmit data for the Message FIFO 10)

Writing to this register sets the transmit data in message FIFO 10 and the IPC1STA0.RDY bit becomes "1". If you write to this register while the IPC1STA0.FULL bit is "1", the write operation is ignored, and the IPC1STA0.FERR bit becomes "1".

3.2.23 IPC1RXD0 : Inter-Processor1 FIFO Receive Data Register0

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x10C

Bit position:	31 ... 0
Bit field:	RXD[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	RXD[31:0]	Received data from the message FIFO 10 Reading this register updates it with the next received data in the message FIFO 10.	R

Note: If a TRZ error occurs, the read is 0 and it does not update to the next data.
S-TYPE-3, P-TYPE-3

This register is for reading received data from the message FIFO 10 from CPU0 to CPU1. Read this register by CPU1. The security attributes of this register are controlled by IPCSAR. The privileged attribute of this register is controlled by IPCPAR.

RXD[31:0] bits (Received data from the message FIFO 10)

If this register is read while the IPC1STA0.RDY bit is "0", the IPC1STA0.RERR bit is set to "1".

3.2.24 IPC1CLR0 : Inter-Processor1 Request Clear Register0

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x110

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	CLR7	CLR6	CLR5	CLR4	CLR3	CLR2	CLR1	CLR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CLRn (n = 0 to 7)	Writing 1 to this bit clears IPC1STA0.IRQn.	W
15:8	—	These bits are read as 0. The write value should be 0.	W
16	RST	Writing 1 resets Message FIFO 10.	W
23:17	—	These bits are read as 0. The write value should be 0.	W
24	RCLR	Writing 1 resets IPC1STA0.RERR.	W
25	FCLR	Writing 1 resets IPC1STA0. FERR.	W
31:26	—	These bits are read as 0. The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The
IPC1CLR0

register is used to cancel the maskable interrupt from CPU0 to CPU1. The register can then be used to cancel the FIFO transfer. It can also clear the Error flag that occurred in the FIFO. The IRQ is assigned eight events, but there is only one interrupt event. The security of this register depends on IPCSAR. The privilege of this register depends on IPCPAR.

CLRn bit (Interrupt Request Clear n (n = 0 to 7))

When 1 is written, IPC1STA0.IRQn is cleared to 0. All bits of IPC1STA0 are 0, and maskable interrupts to CPU1 are cancelled.

RST bit (FIFO Reset)

Writing 1 to the RST bit resets message FIFO 10. The data stored in the FIFO becomes invalid and the following registers are cleared:

- IPC1STA0.FULL
- IPC1STA0.RDY

If there is no interrupt event other than the message FIFO, the interrupt request is canceled.

RCLR bit (Clear RDY Error Status 10)

The RCLR bit clears IPC1STA0.RERR.

FCLR bit (Clear FULL Error Status 10)

The FCLR bit clears IPC1STA0. FERR.

3.2.25 IPC1STA1 : Inter-Processor1 Request Status Register1

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x120

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	IRQn (n = 0 to 7)	This bit indicates the status of interrupt request. 0: Not requesting interrupt 1: Requesting interrupt	R
15:8	—	These bits are read as 0. The write value should be 0.	R
16	RDY	This bit is set when FIFO is not empty. This bit is used for the interrupt request. 0: FIFO 11 data is not ready 1: FIFO 11 data is ready	R
17	FULL	0: FIFO 11 is not full 1: FIFO 11 is full	R
23:18	—	These bits are read as 0. The write value should be 0.	R
24	RERR	Indicates that the message FIFO 11 tried to read data despite being empty. 0: Not Error. Not requesting interrupt. 1: Error. Requesting interrupt.	R
25	FERR	Indicates that the message FIFO 11 tried to send more data even though it was full. 0: Not Error. Not requesting interrupt. 1: Error. Requesting interrupt.	R
31:26	—	These bits are read as 0. The write value should be 0.	R

Note: S-TYPE-3, P-TYPE-3

This register indicates the status of the maskable interrupt IPC1IRQ1 from CPU0 to CPU1. When any of IRQn (n = 0 to 7), RDY, RERR, or FERR is '1', the IPC1IRQ1 interrupt is issued. Read this register in the interrupt handler to determine which interrupt has occurred. The security attributes of this register are controlled by IPCSAR. The privileged attribute of this register is controlled by IPCPAR.

IRQn bit (Interrupt Request Status n (n = 0 to 7))

These bits indicate an interrupt to CPU1. Writing "1" to IPC1ISET1.SETn sets IRQn to "1". Writing "1" to IPC1CLR1.CLRn clears IRQn to "0". If any bit is "1", an IPC1IRQ1 interrupt request is issued.

RDY bit (Receive Data Ready)

When this bit is "1", it indicates that there is data received in the message FIFO 11 for CPU1. When the message FIFO is empty, this bit is "0". When this bit is "1", the IPC1IRQ1 interrupt request is issued.

FULL bit (FIFO Full)

The depth of message FIFO 11 is 4. When this bit is "1", it indicates that the message FIFO 11 is full.

RERR bit (RDY Error Status)

When reading IPC1RXD1 while the RDY bit is "0", this bit sets to "1". Writing "1" to the IPC1CLR1.RCLR bit clears this bit to "0". When this bit is "1", the IPC1IRQ1 interrupt request is issued.

FERR bit (FULL Error Status)

When writing to IPC1TXD1 while the FULL bit is "1", this bit sets to "1". Writing "1" to the IPC1CLR1.FCLR bit clears this bit to "0". When this bit is "1", the IPC1IRQ1 interrupt request is issued.

3.2.26 IPC1ISET1 : Inter-Processor1 IRQ Request Set Register1

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x124

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	SET7	SET6	SET5	SET4	SET3	SET2	SET1	SET0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	SETn (n = 0 to 7)	Writing 1 to this bit sets IPC1STA1.IRQn.	W
31:8	—	These bits are read as 0. The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The
IPC1ISET1

register is used to issue a maskable interrupt from CPU0 to CPU1. The IRQ is assigned eight events, but there is only one interrupt event The security of this register depends on IPCSAR. The privilege of this register depends on IPCPAR.

SETn bit (Interrupt Request Set n (n = 0 to 7))

When 1 is written, IPC1STA1.IRQn is set to 1 and a maskable interrupt to CPU1 is issued.

3.2.27 IPC1TXD1 : Inter-Processor1 FIFO Transfer Data Register1

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x128

Bit position:	31	...	0
Bit field:	TXD[31:0]		
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		

Bit	Symbol	Function	R/W
31:0	TXD[31:0]	Transmit data for the Message FIFO 11	W

Note: Only writing with a 32-bit data size is valid. Halfword or byte access is ignored.
Note: S-TYPE-3, P-TYPE-3

This is a register for writing data to message FIFO 11 from CPU0 to CPU1. Write to this register by CPU0. The security attributes of this register are controlled by IPCSAR. The privileged attribute of this register is controlled by IPCPAR.

TXD[31:0] bits (Transmit data for the Message FIFO 11)

Writing to this register sets the transmit data in message FIFO 11 and the IPC1STA1.RDY bit becomes "1". If you write to this register while the IPC1STA1.FULL bit is "1", the write operation is ignored, and the IPC1STA1.FERR bit becomes "1".

3.2.28 IPC1RXD1 : Inter-Processor1 FIFO Receive Data Register1

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x12C

Bit position:	31	...	0
Bit field:	RXD[31:0]		
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		

Bit	Symbol	Function	R/W
31:0	RXD[31:0]	Received data from the message FIFO 11 Reading this register updates it with the next received data in the message FIFO 11.	R

Note: If a TRZ error occurs, the read is 0 and it does not update to the next data.
S-TYPE-3, P-TYPE-3

This register is for reading received data from the message FIFO 11 from CPU0 to CPU1. Read this register by CPU1. The security attributes of this register are controlled by IPCSAR. The privileged attribute of this register is controlled by IPCPAR.

RXD[31:0] bits (Received data from the message FIFO 11)

If this register is read while the IPC1STA1.RDY bit is "0", the IPC1STA1.RERR bit is set to "1".

3.2.29 IPC1CLR1 : Inter-Processor1 Clear Register1

Base address: IPC = 0x4002_0000
IPC_NS = 0x5002_0000

Offset address: 0x130

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	CLR7	CLR6	CLR5	CLR4	CLR3	CLR2	CLR1	CLR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CLRn (n = 0 to 7)	Writing 1 to this bit clears IPC0IRQSTA1.IRQn.	W
15:8	—	These bits are read as 0. The write value should be 0.	W
16	RST	Writing 1 resets message FIFO 11.	W
23:17	—	These bits are read as 0. The write value should be 0.	W
24	RCLR	Writing 1 resets IPC1STA1.RERR.	W
25	FCLR	Writing 1resets IPC1STA1. FERR.	W
31:26	—	These bits are read as 0. The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The
IPC1CLR1

register is used to cancel the maskable interrupt from CPU0 to CPU1. The register can then be used to cancel the FIFO transfer. It can also clear the Error flag that occurred in the FIFO. The IRQ is assigned eight events, but there is only one interrupt event. The security of this register depends on IPCSAR. The privilege of this register depends on IPCPAR.

CLRn bit (Interrupt Request Clear n (n = 0 to 7))

When 1 is written, IPC1STA1.IRQn is cleared to 0. All bits of IPC1STA1 are 0, and maskable interrupts to CPU1 are cancelled.

RST bit (FIFO Reset)

Writing 1 resets the message FIFO 11. The data stored in the FIFO becomes invalid and the following registers are cleared:

- IPC1STA1.FULL
- IPC1STA1.RDY

If there is no interrupt event other than the message FIFO, the interrupt request is canceled.

RCLR bit (Clear RDY Error Status 11)

The RCLR bit clears IPC1STA1.RERR.

FCLR bit (Clear FULL Error Status 11)

The FCLR bit clears IPC1STA1. FERR.

3.3 Operation

3.3.1 Semaphore

The IPC has registers to perform semaphore for synchronization and exclusive control between CPU0 and CPU1.

Figure 3.2 shows the structure of the IPCSEM register used for semaphore control.

This register only indicates the status and does not have any hardware protection of shared memory. Thus, the exclusive control must be done by software that uses this register.

Security attributes and privileged attributes must be set in common for IPCSEM0 to IPCSEM7, and IPCSEM8 to IPCSEM15.

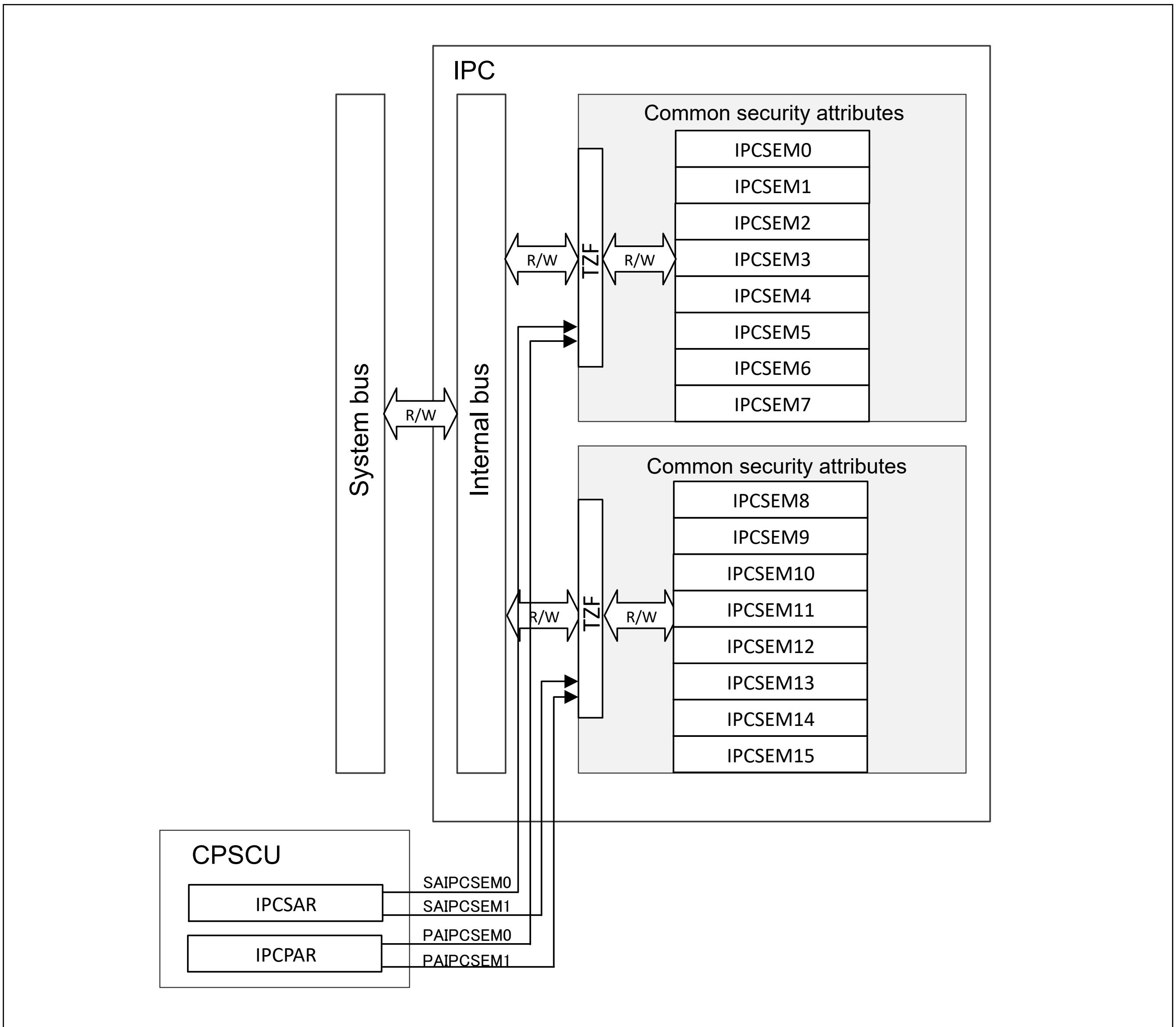


Figure 3.2 IPCSEM register

3.3.2 Inter-processor Non-maskable Interrupt

IPC supports the Inter-processor Non-maskable Interrupt feature between CPU0 and CPU1. Figure 3.3 shows the configuration diagram of the "Inter-processor Non-maskable Interrupt" function. IPC0NMI is the non-maskable interrupt from CPU1 to CPU0 and IPC1NMI is the one from CPU0 to CPU1. IPCiNMI (i = 0, 1) functions are controlled by IPCiNMISTA, IPCiNMISET, and IPCiNMICLR. The security attributes of these bits are controlled by IPCSAR and privileged attributes of these bits are controlled by IPCPAR.

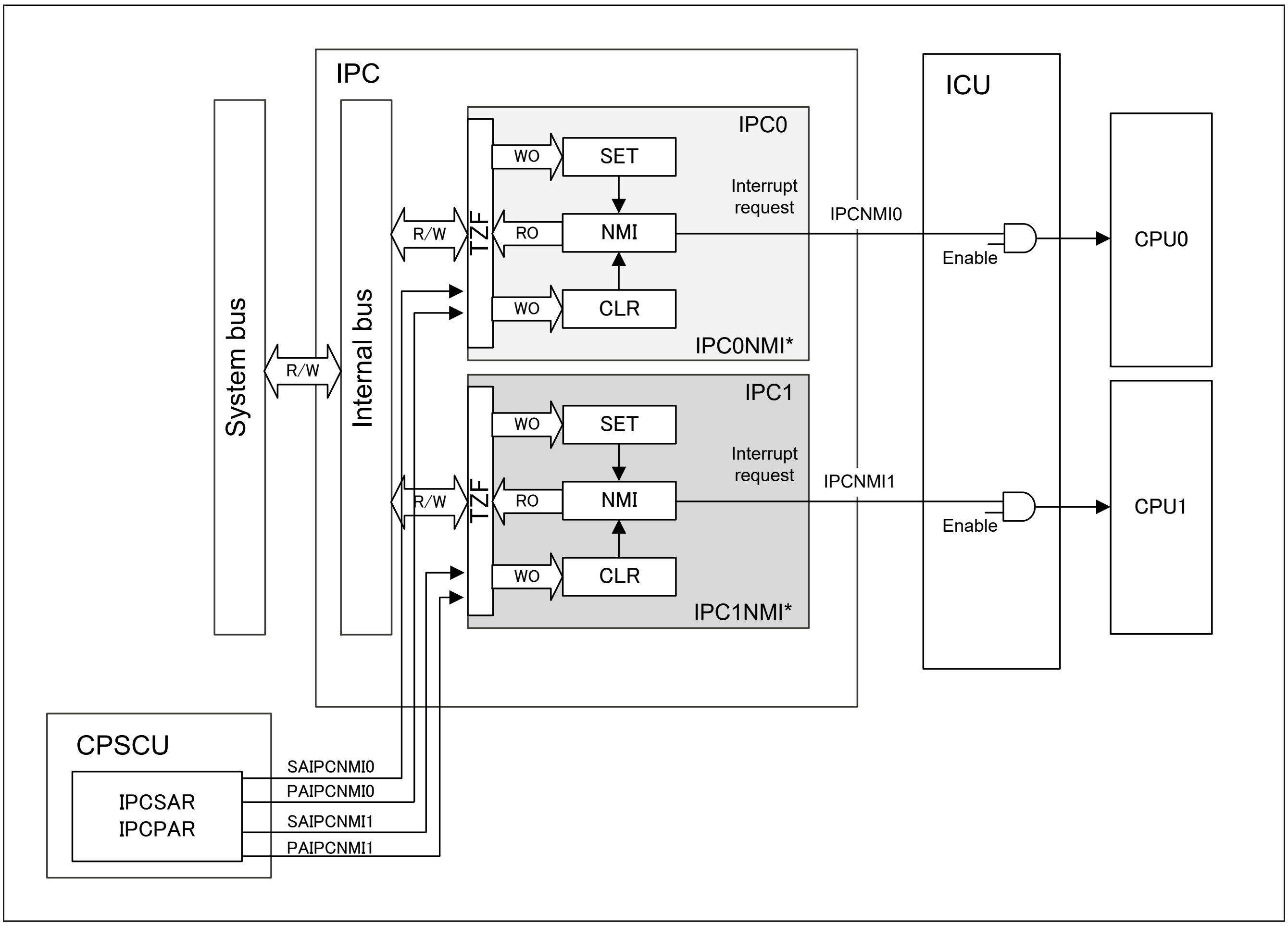


Figure 3.3 Inter-processor non-maskable interrupt

The CPU can only select either Secure or Non-secure for NMI. The IPC NMI security attribute must also be set accordingly. See section 14, Interrupt Controller Unit (ICU) for controls on mixing Secure and Non-secure NMIs. If the IPC does not use NMI interrupt, mask it in the ICU register.

3.3.3 Inter-processor Maskable Interrupt

IPC supports the Inter-processor Maskable Interrupt feature between CPU0 and CPU1. Figure 3.4 shows the configuration diagram of the "Inter-processor Maskable Interrupt" function. IPC0IRQj (j = 0, 1) is the maskable interrupt from CPU1 to CPU0 and IPC1IRQj (j = 0, 1) is the maskable interrupt from CPU0 to CPU1. IPCiIRQj (i = 0, 1), (j = 0, 1) are controlled by IPCiSTAj.IRQn, IPCiISETj.SETn, IPCiCLRj.CLRn (n = 0 to 7) respectively. The security attributes of the registers related to this function are controlled by IPCSAR, and the privileged attributes are controlled by IPCPAR. Security attributes and privileged attributes can be set for each IPCiIRQj (i = 0, 1), (j = 0, 1). Each IPCiIRQj has eight interrupt request sources, IPCiSTAj.IRQn (n = 0 to 7).

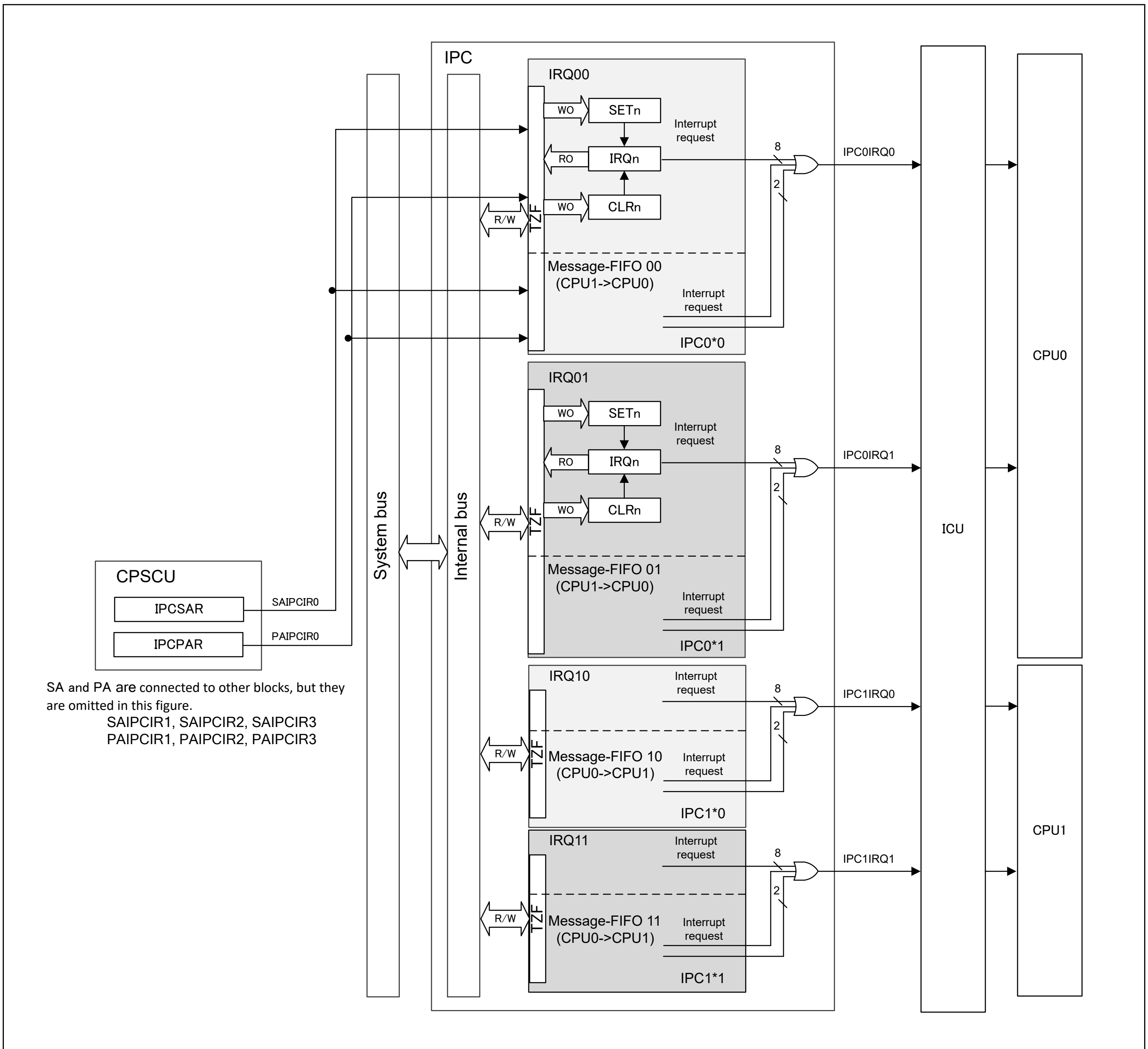


Figure 3.4 Inter-processor maskable interrupt

3.3.4 Exclusive Control Flow

Figure 3.5 shows an example of the exclusive access control flow when semaphore is executed between two CPUs.

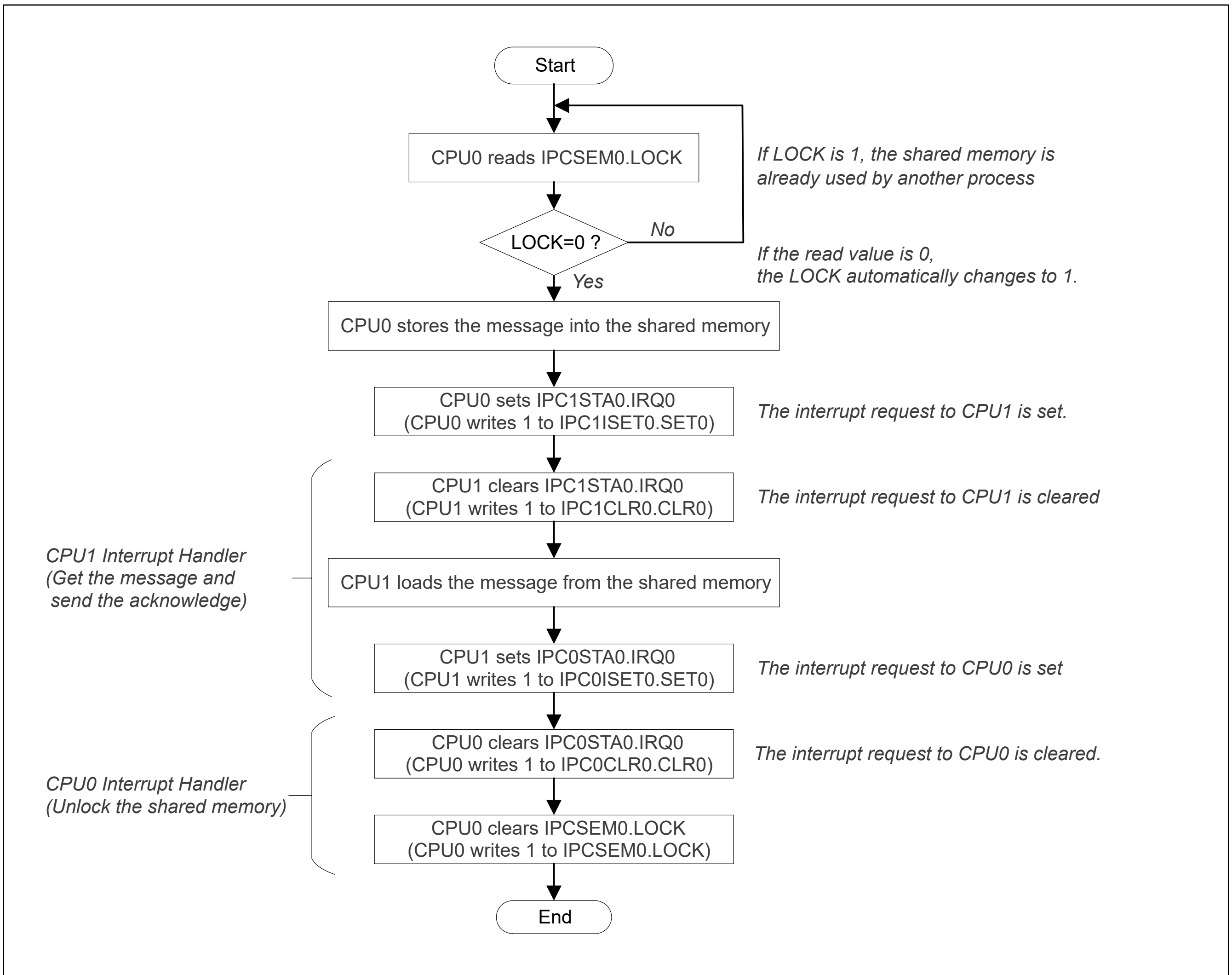


Figure 3.5 Example flow of CPU0 sending a message to CPU1

3.3.5 Message-FIFO

The IPC has four FIFOs. IPC00 and IPC01 are FIFOs that transmit data from CPU1 to CPU0, and IPC10 and IPC11 are FIFOs that transmit data from CPU0 to CPU1. Each FIFO issues the maskable interrupt IPCiIRQj (i = 0, 1), (j = 0, 1) as shown in the following figure. IPCiIRQj is issued when there is a request in the Message-FIFO (RDY = 1), when the Message-FIFO is full and you write to it (FERR = 1), or when the Message-FIFO is empty and you read from it (RERR = 1) in IRQij (i = 0, 1; j = 0, 1). The security attributes of this register are controlled by IPCSAR. The privileged attribute of this register is controlled by IPCPAR. Security attributes and privileged attributes can be set for each FIFO. Figure 3.6 shows an image diagram of the message-FIFOs.

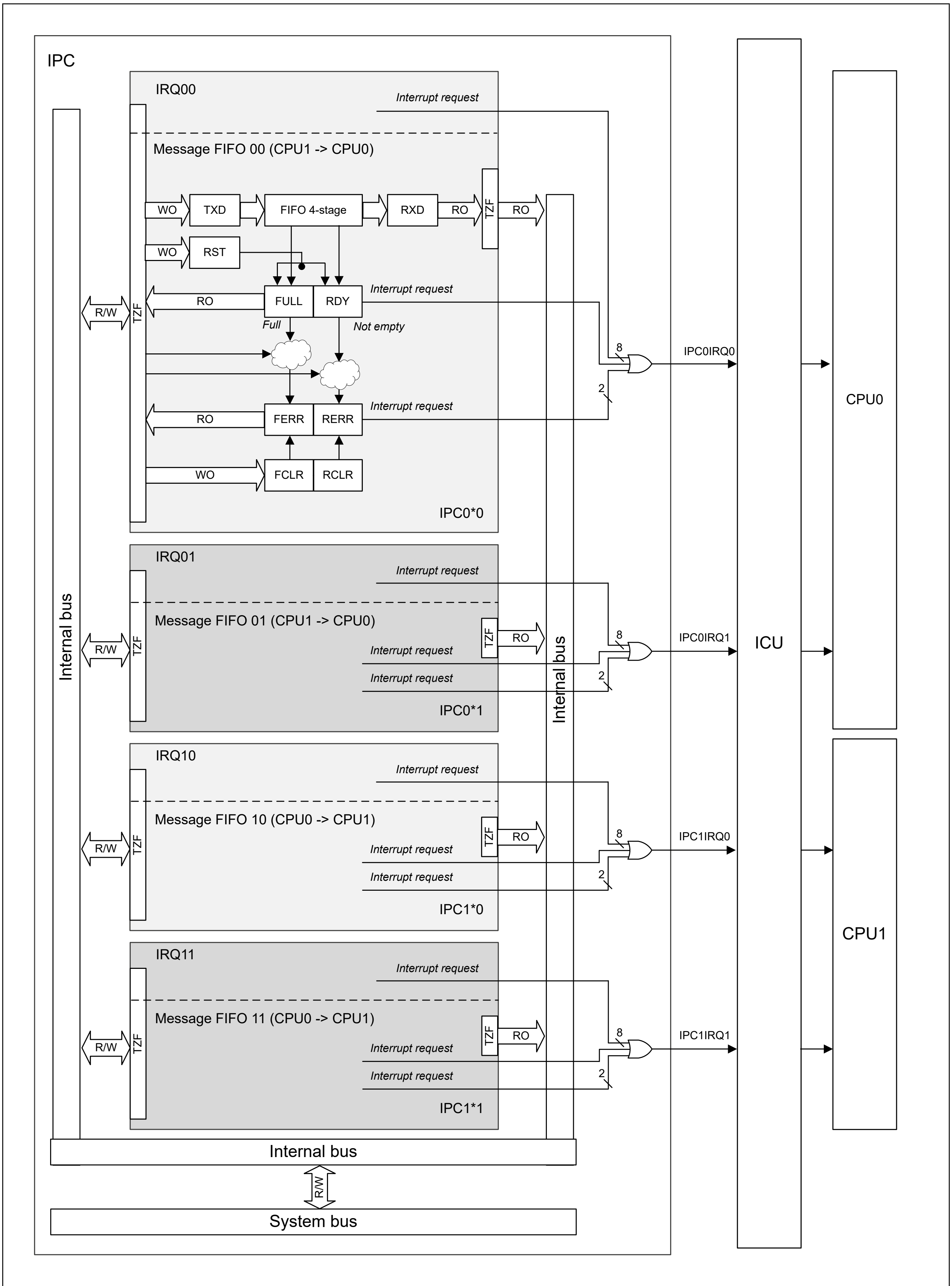


Figure 3.6 Inter-processor message FIFO

See Figure 3.4 for the security and privileged attributions.

Figure 3.7 shows an example of sending data from CPU1 to CPU0 using message FIFO 00.

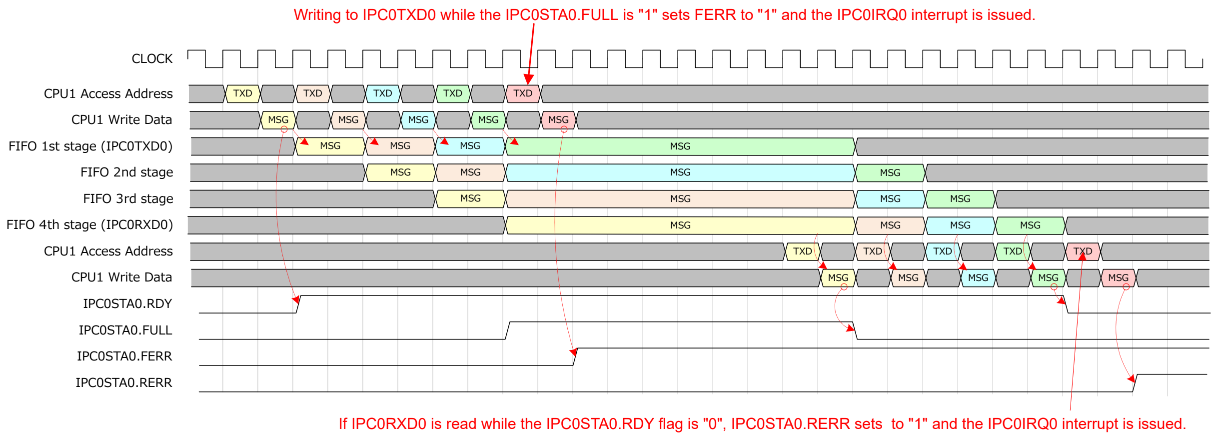


Figure 3.7 Inter-processor message FIFO timing

3.3.6 IPC Interrupt

CPU Sleep mode and CPU Deep Sleep mode can be canceled by the interrupts issued by IPC functions.

4. Operating Modes

4.1 Overview

Table 4.1 shows the selection of operating modes by the mode-setting pin(MD pin) and JTAG accessible command. For details, see section 4.3. Details of Operating Modes.

Table 4.1 Selection of operating modes by the mode-setting pin

Mode-setting pin (MD)	Operating mode	On-chip MRAM	External bus
1	Single chip mode/ JTAG Boot Mode	Enable	Disable
0	SCI / USB boot mode	Enable	Disable

4.2 Operating Mode Types and Selection

Table 4.1 show the relationship between levels on the mode-setting pins (MD) on release from the reset state and the operating mode selected at that time. For details on each of the operating modes, see section 4.3. Details of Operating Modes. Operation starts with the on-chip MRAM enabled and the external bus disabled, regardless of the mode in which operation started.

4.3 Details of Operating Modes

4.3.1 Single-Chip Mode

In single-chip mode, all I/O pins are available for use as input or output port, inputs or outputs for peripheral functions, or as interrupt inputs. When a reset is released while the MD pin is high, the MCU starts in single-chip mode and the on-chip MRAM is enabled.

4.3.2 JTAG Boot Mode

In this mode, the on-chip MRAM programming routine (JTAG boot program), stored in a dedicated area within the MCU, is used. The on-chip MRAM, including code MRAM and extra MRAM, can be modified from outside the MCU by using the JTAG and SWD interface. CSW.DbgSwEnable bit in APB-AP Control/Status Word register must be 1 in this mode.

To enter this mode, it is necessary to input the request from the JTAG and SWD-I/F during RES-pin reset.

4.3.3 SCI Boot Mode

In this mode, the on-chip MRAM programming routine (SCI boot program), stored in a dedicated area within the MCU, is used. The on-chip MRAM, including the code MRAM and extra MRAM, can be modified from outside the MCU by using a universal asynchronous receiver/transmitter (UART) SCI. For details, see section 59, MRAM. The MCU starts in SCI boot mode if the MD pin is held low on release from the reset state.

4.3.4 USB Boot Mode

In this mode, the on-chip MRAM programming routine (USB boot program), stored in a dedicated area within the MCU, is used. The on-chip MRAM, including the code MRAM and extra MRAM, can be modified from outside the MCU by using the USB. For details, see section 59, MRAM. The MCU starts in USB boot mode if the MD pin is held low on release from the reset state.

4.4 Operating Modes Transitions

4.4.1 Operation Modes and the Relationship of Mode Transition

Figure 4.1 shows that the operation modes and the relationship of mode transition. Operating mode can be transitioned in the direction of the arrow at the figure.

JTAG Boot mode and SCI/USB Boot mode is not possible using POR.

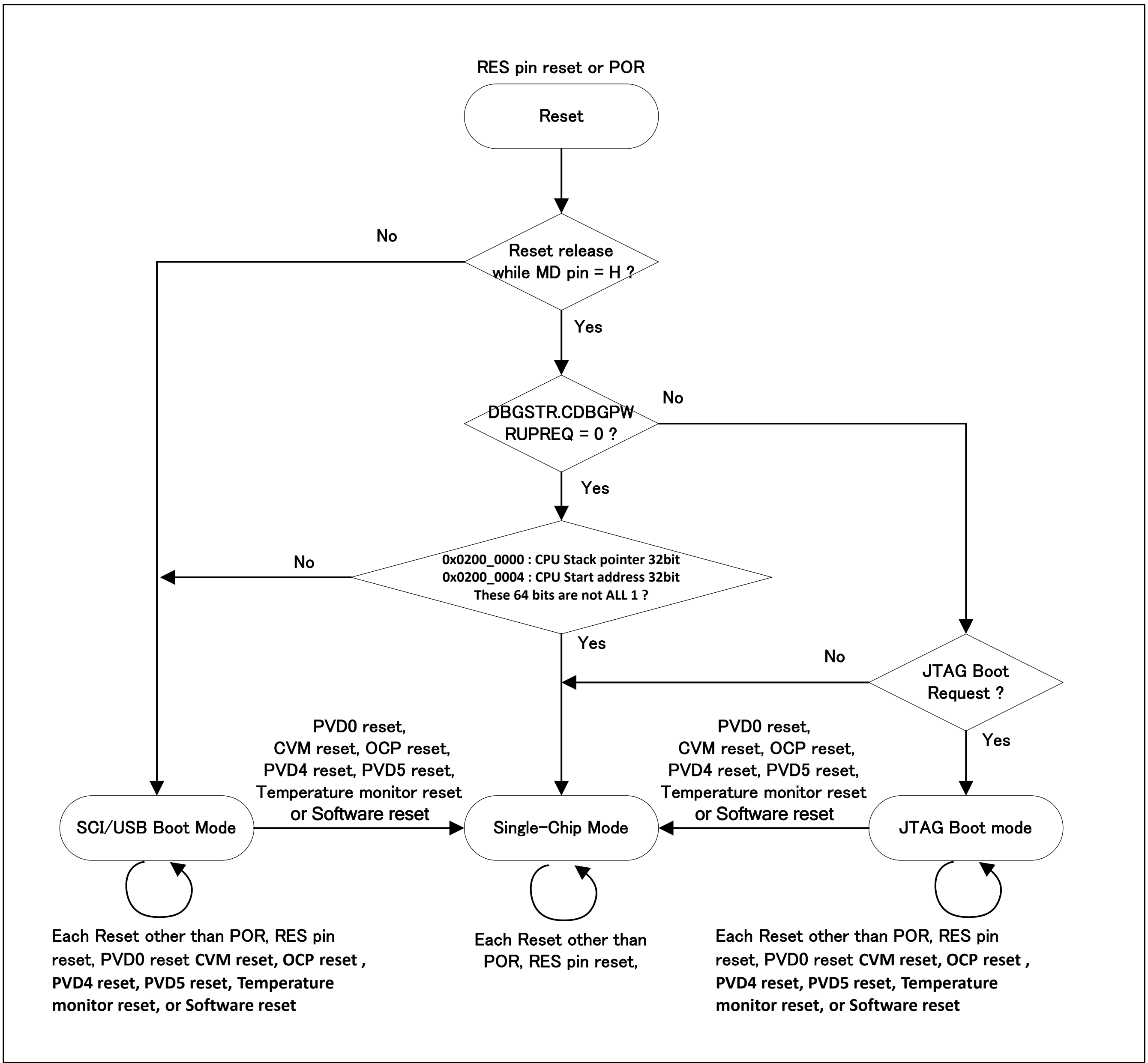


Figure 4.1 Operation modes and the relationship of mode transition

5. Address Space

5.1 Address Space

The MCU supports a 4 GB linear address space ranging from 0x0000_0000 to 0xFFFF_FFFF that can contain both program and data. In the address from 0x0000_0000 to 0x5FFF_FFFF, secure and non-secure region are isolated. Figure 5.1 and Figure 5.2 show the memory map example of 1 MB MRAM product.

Address	CPU0 (CM85)	CPU1 (CM33)	the others	IDAU/MSAU Security_Attribution
0xE010_0000 – 0xFFFF_FFFF	System for CM85	System for CM33	Reserved area*1	
0xE000_0000 – 0xE010_0000	Private Peripheral bus	Private Peripheral bus	Reserved area*1	
0xA000_0000 – 0xE000_0000	Reserved area*1			Non-secure
0x7000_0000 – 0xA000_0000	External address space (OSPI area)			Non-secure
0x6800_0000 – 0x7000_0000	External address space (SDRAM area)			Non-secure
0x6000_0000 – 0x6800_0000	External address space (CS area)			Non-secure
0x5050_0000 – 0x6000_0000	Reserved area*1			Non-secure
0x5000_0000 – 0x5050_0000	Peripheral IO registers			Non-secure
0x4050_0000 – 0x5000_0000	Reserved area*1			Secure
0x4000_0000 – 0x4050_0000	Peripheral IO registers			Secure
0x3A02_0000 – 0x4000_0000	Reserved area*1			Non-secure
0x3A01_0000 – 0x3A02_0000	CPU1-STCM alias*9	Reserved area*1	CPU1-STCM alias*9	Non-secure
0x3A00_0000 – 0x3A01_0000	CPU1-CTCM alias*8	Reserved area*1	CPU1-CTCM alias*8	Non-secure
0x3804_0000 – 0x3A00_0000	Reserved area*1			Non-secure
0x3802_0000 – 0x3804_0000	Reserved area*1	CPU0-DTCM alias*7		Non-secure
0x3800_0000 – 0x3802_0000	Reserved area*1	CPU0- ITCM alias*6		Non-secure
0x321D_4000 – 0x3800_0000	Reserved area*1			Non-secure
0x321A_0000 – 0x321D_4000	On-chip SRAM (SRAME)*10			Non-secure
0x3200_0000 – 0x321A_0000	On-chip SRAM			Non-secure
0x3002_0000 – 0x3200_0000	Reserved area*1			Non-secure
0x3001_0000 – 0x3002_0000	CPU0-DTCM*7	Reserved area*1	Reserved area*1	Non-secure
0x3000_0000 – 0x3001_0000		CPU1-STCM*9		Non-secure
0x2A02_0000 – 0x3000_0000	Reserved area*1			Non-secure callable for CPU Secure for other bus masters
0x2A01_0000 – 0x2A02_0000	CPU1-STCM alias*5	Reserved area*1	CPU1-STCM alias*5	Non-secure callable for CPU Secure for other bus masters
0x2A00_0000 – 0x2A01_0000	CPU1-CTCM alias*4	Reserved area*1	CPU1-CTCM alias*4	Non-secure callable for CPU Secure for other bus masters
0x2804_0000 – 0x2A00_0000	Reserved area*1			Non-secure callable for CPU Secure for other bus masters
0x2802_0000 – 0x2804_0000	Reserved area*1	CPU0-DTCM alias*3		Non-secure callable for CPU Secure for other bus masters
0x2800_0000 – 0x2802_0000	Reserved area*1	CPU0-ITCM alias*2		Non-secure callable for CPU Secure for other bus masters
0x221D_4000 – 0x2800_0000	Reserved area*1			Non-secure callable for CPU Secure for other bus masters
0x221A_0000 – 0x221D_4000	On-chip SRAM (SRAME)*10			Non-secure callable for CPU Secure for other bus masters
0x2200_0000 – 0x221A_0000	On-chip SRAM			Non-secure callable for CPU Secure for other bus masters
0x2002_0000 – 0x2200_0000	Reserved area*1			Non-secure callable for CPU Secure for other bus masters
0x2001_0000 – 0x2002_0000	CPU0-DTCM*3	Reserved area*1	Reserved area*1	Non-secure callable for CPU Secure for other bus masters
0x2000_0000 – 0x2001_0000		CPU1-STCM*5		Non-secure callable for CPU Secure for other bus masters
0x1880_0000 – 0x2000_0000	Reserved area*1			Non-secure
0x1800_0000 – 0x1880_0000	SiP-Flash area (OSPI1 CS1 area)			Non-secure
0x1300_0000 – 0x1800_0000	Reserved area*1			Non-secure
0x12E0_0000 – 0x1300_0000	Extra MRAM (option-setting memory)			Non-secure
0x12D0_0000 – 0x12E0_0000	Reserved area*1			Non-secure
0x12A0_0000 – 0x12D0_0000	Extra MRAM (option-setting memory)			Non-secure
0x1210_0000 – 0x12A0_0000	Reserved area*1			Non-secure
0x1200_0000 – 0x1210_0000	Code MRAM			Non-secure
0x1002_0000 – 0x1200_0000	Reserved area*1			Non-secure
0x1001_0000 – 0x1002_0000	CPU0-ITCM*6	Reserved area*1	Reserved area*1	Non-secure
0x1000_0000 – 0x1001_0000		CPU1-CTCM*8		Non-secure
0x0880_0000 – 0x1000_0000	Reserved area*1			Non-secure callable for CPU Secure for other bus masters
0x0800_0000 – 0x0880_0000	SiP-Flash area (OSPI1 CS1 area)			Non-secure callable for CPU Secure for other bus masters
0x0300_0000 – 0x0800_0000	Reserved area*1			Non-secure callable for CPU Secure for other bus masters
0x02E0_0000 – 0x0300_0000	Extra MRAM (option-setting memory)			Non-secure callable for CPU Secure for other bus masters
0x02D0_0000 – 0x02E0_0000	Reserved area*1			Non-secure callable for CPU Secure for other bus masters
0x02A0_0000 – 0x02D0_0000	Extra MRAM (option-setting memory)			Non-secure callable for CPU Secure for other bus masters
0x0210_0000 – 0x02A0_0000	Reserved area*1			Non-secure callable for CPU Secure for other bus masters
0x0200_0000 – 0x0210_0000	Code MRAM			Non-secure callable for CPU Secure for other bus masters
0x0002_0000 – 0x0200_0000	Reserved area*1			Non-secure callable for CPU Secure for other bus masters
0x0001_0000 – 0x0002_0000	CPU0- ITCM*2	Reserved area*1	Reserved area*1	Non-secure callable for CPU Secure for other bus masters
0x0000_0000 – 0x0001_0000		CPU1-CTCM*4		Non-secure callable for CPU Secure for other bus masters

Note: See Table 5.1. The capacity of the Code MRAM, On-chip SRAM differs depending on the product.
Note 1. Reserved areas should not be accessed.
Note 2. CPU0-ITCM for Secure has different addresses for CPU1 and other bus masters.
Note 3. CPU0-DTCM for Secure has different addresses for CPU1 and other bus masters.
Note 4. CPU1-CTCM for Secure has different addresses for CPU0 and other bus masters.
Note 5. CPU1-STCM for Secure has different addresses for CPU0 and other bus masters.
Note 6. CPU0-ITCM for Non-Secure has different addresses for CPU1 and other bus masters.
Note 7. CPU0-DTCM for Non-Secure has different addresses for CPU1 and other bus masters.
Note 8. CPU1-CTCM for Non-Secure has different addresses for CPU0 and other bus masters.
Note 9. CPU1-STCM for Non-Secure has different addresses for CPU0 and other bus masters.
Note 10. These areas are accessible only when SRAM ECC function is disabled.

Figure 5.1 1 MB MRAM product for SiP Product

Address	CPU0 (CM85)	CPU1 (CM33)	the others	IDAU/MSAU Security_Attribution
0xFFFF_FFFF	System for CM85	System for CM33	Reserved area*1	
0xE010_0000	Private Peripheral bus	Private Peripheral bus	Reserved area*1	
0xE000_0000	Reserved area*1			Non-secure
0xA000_0000	External address space (OSPI area)			
0x7000_0000	External address space (SDRAM area)			
0x6800_0000	External address space (CS area)			
0x6000_0000	Reserved area*1			
0x5050_0000	Peripheral IO registers			
0x5000_0000	Reserved area*1			Secure
0x4050_0000	Peripheral IO registers			
0x4000_0000	Reserved area*1			Non-secure
0x3A02_0000	CPU1-STCM alias*9	Reserved area*1	CPU1-STCM alias*9	
0x3A01_0000	CPU1-CTCM alias*8	Reserved area*1	CPU1-CTCM alias*8	
0x3A00_0000	Reserved area*1			
0x3804_0000	Reserved area*1	CPU0-DTCM alias*7		
0x3802_0000	Reserved area*1	CPU0- ITCM alias*6		
0x3800_0000	Reserved area*1			
0x321D_4000	On-chip SRAM (SRAME)*10			
0x321A_0000	On-chip SRAM			
0x3200_0000	Reserved area*1			
0x3002_0000	CPU0-DTCM*7	Reserved area*1	Reserved area*1	
0x3001_0000		CPU1-STCM*9		
0x3000_0000	Reserved area*1			Non-secure callable for CPU Secure for other bus masters
0x2A02_0000	CPU1-STCM alias*5	Reserved area*1	CPU1-STCM alias*5	
0x2A01_0000	CPU1-CTCM alias*4	Reserved area*1	CPU1-CTCM alias*4	
0x2A00_0000	Reserved area*1			
0x2804_0000	Reserved area*1	CPU0-DTCM alias*3		
0x2802_0000	Reserved area*1	CPU0-ITCM alias*2		
0x2800_0000	Reserved area*1			
0x221D_4000	On-chip SRAM (SRAME)*10			
0x221A_0000	On-chip SRAM			
0x2200_0000	Reserved area*1			
0x2002_0000	CPU0-DTCM*3	Reserved area*1	Reserved area*1	
0x2001_0000		CPU1-STCM*5		
0x2000_0000	Reserved area*1			Non-secure
0x1800_0000	Reserved area*1			
0x1300_0000	Extra MRAM (option-setting memory)			
0x12E0_0000	Reserved area*1			
0x12D0_0000	Extra MRAM (option-setting memory)			
0x12A0_0000	Reserved area*1			
0x1210_0000	Code MRAM			
0x1200_0000	Reserved area*1			
0x1002_0000	CPU0-ITCM*6	Reserved area*1	Reserved area*1	
0x1001_0000		CPU1-CTCM*8		
0x1000_0000	Reserved area*1			Non-secure callable for CPU Secure for other bus masters
0x0300_0000	Extra MRAM (option-setting memory)			
0x02E0_0000	Reserved area*1			
0x02D0_0000	Extra MRAM (option-setting memory)			
0x02A0_0000	Reserved area*1			
0x0210_0000	Code MRAM			
0x0200_0000	Reserved area*1			
0x0002_0000	CPU0- ITCM*2	Reserved area*1	Reserved area*1	
0x0001_0000		CPU1-CTCM*4		
0x0000_0000				

Note: See Table 5.1. The capacity of the Code MRAM, On-chip SRAM differs depending on the product.
Note 1. Reserved areas should not be accessed.
Note 2. CPU0-ITCM for Secure has different addresses for CPU1 and other bus masters.
Note 3. CPU0-DTCM for Secure has different addresses for CPU1 and other bus masters.
Note 4. CPU1-CTCM for Secure has different addresses for CPU0 and other bus masters.
Note 5. CPU1-STCM for Secure has different addresses for CPU0 and other bus masters.
Note 6. CPU0-ITCM for Non-Secure has different addresses for CPU1 and other bus masters.
Note 7. CPU0-DTCM for Non-Secure has different addresses for CPU1 and other bus masters.
Note 8. CPU1-CTCM for Non-Secure has different addresses for CPU0 and other bus masters.
Note 9. CPU1-STCM for Non-Secure has different addresses for CPU0 and other bus masters.
Note 10. These areas are accessible only when SRAM ECC function is disabled.

Figure 5.2 1 MB MRAM product for Standard Product

Table 5.1 Capacity of the Code MRAM, SRAM and SiP-Flash

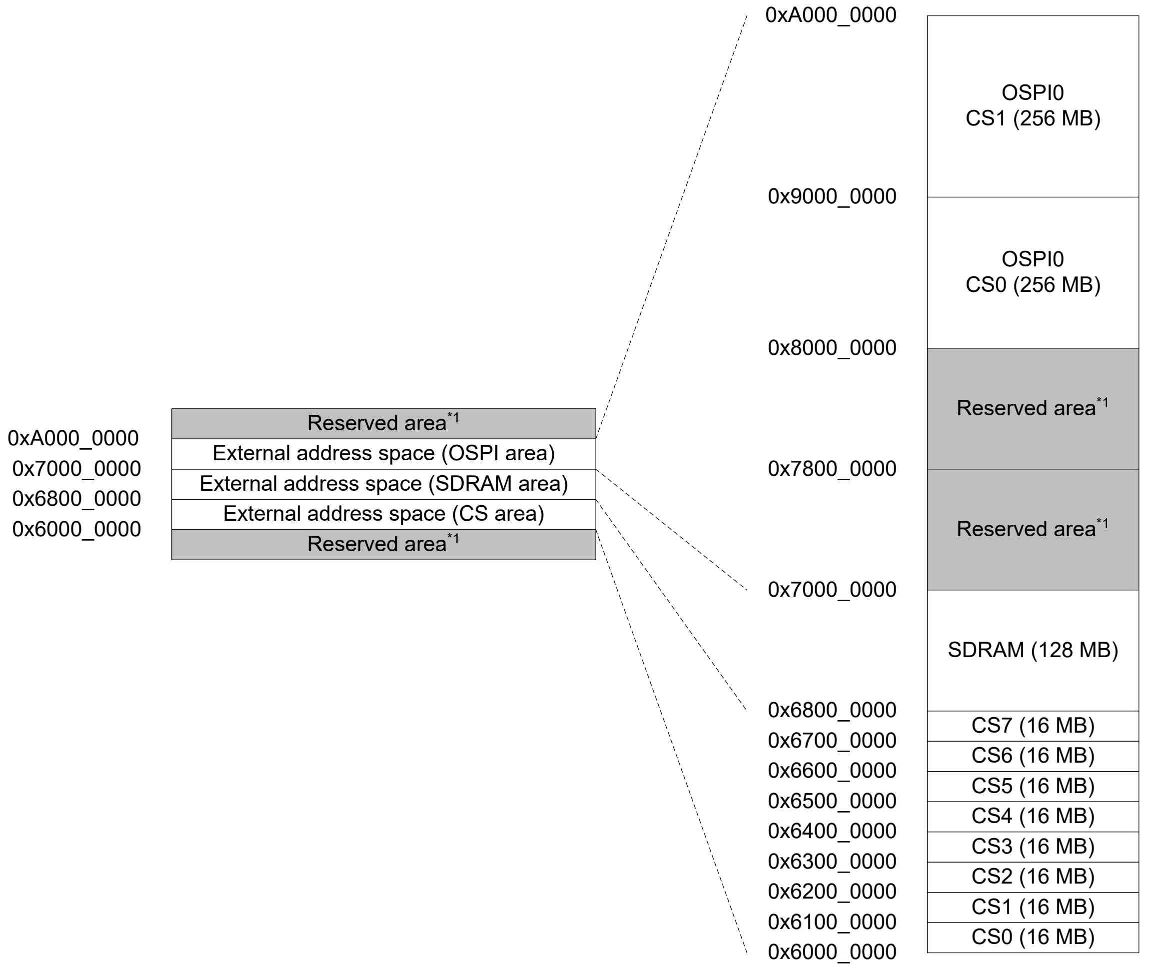
CPU	Code MRAM			On-chip SRAM			SiP-Flash		
	Capacity	Address		Capacity	Address		Capacity	Address	
		Secure alias	Non-Secure alias		Secure alias	Non-Secure alias		Secure alias	Non-Secure alias
Dual	1 MB	0x0200_0000 to 0x0210_0000	0x1200_0000 to 0x1210_0000	1.6 MB	0x2200_0000 to 0x221A_0000	0x3200_0000 to 0x321A_0000	4 MB	0x0800_0000 to 0x0840_0000	0x1800_0000 to 0x1840_0000
							8 MB	0x0800_0000 to 0x0880_0000	0x1800_0000 to 0x1880_0000
Single	1 MB	0x0200_0000 to 0x0210_0000	0x1200_0000 to 0x1210_0000	1.6 MB + 128 KB*1	0x2200_0000 to 0x221A_0000 0x2A00_0000 to 0x2A02_0000	0x3200_0000 to 0x321A_0000 0x3A00_0000 to 0x3A02_0000	—		
	512 KB	0x0200_0000 to 0x0208_0000	0x1200_0000 to 0x1208_0000						

Note 1. In single-core, C-TCM and S-TCM at addresses 0x2A00_0000 to 0x2A01_FFFFF can be used as SRAM. For details of TCM, see section 2, CPU.

5.2 External Address Space

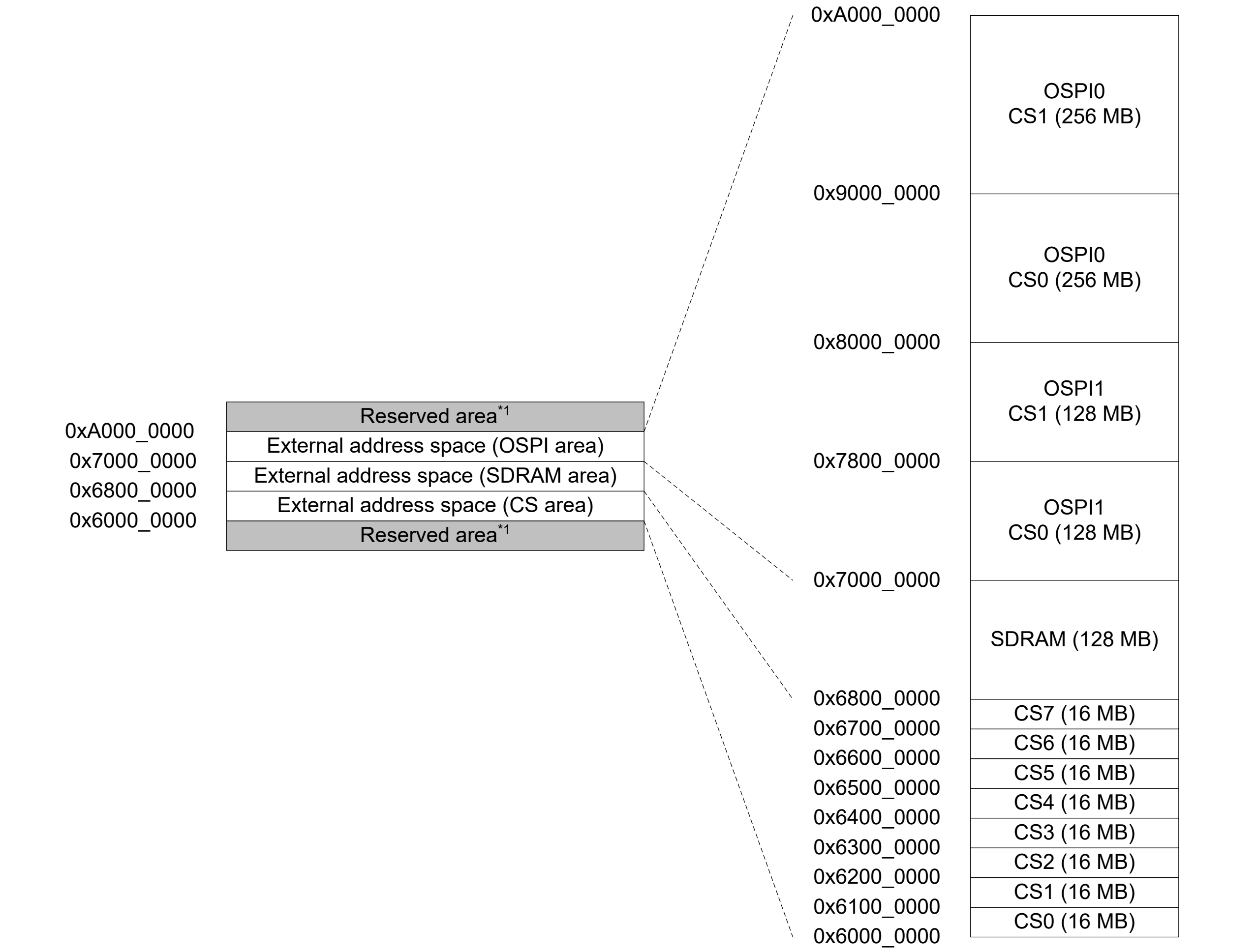
The external address space is divided into CS areas (CS0 to CS7), SDRAM area (SDCS), and OSPI areas (CS0 and CS1). The eight CS areas (CS0 to CS7) each correspond to the CSn signal output from a CSn (n = 0 to 7) pin. The two OSPI areas (CS0 and CS1) each correspond to the OM_CSn signal output from an OM_CSn (n = 0, 1) pin.

Figure 5.3 and Figure 5.4 show the address ranges associated with the individual CS areas (CS0 to CS7), SDRAM area (SDCS), and OSPI areas (CS0 and CS1).



Note 1. Reserved areas should not be accessed.

Figure 5.3 Detailed address map for CS area, SDRAM area, and OSPI area for SiP Product



Note 1. Reserved areas should not be accessed.

Figure 5.4 Detailed address map for CS area, SDRAM area, and OSPI area for Standard Product

5.3 Peripheral I/O Register Address Space

The peripheral I/O register address space is classified into two types. Figure 5.5 shows the address space types and Table 5.2 shows the address space types for each module.

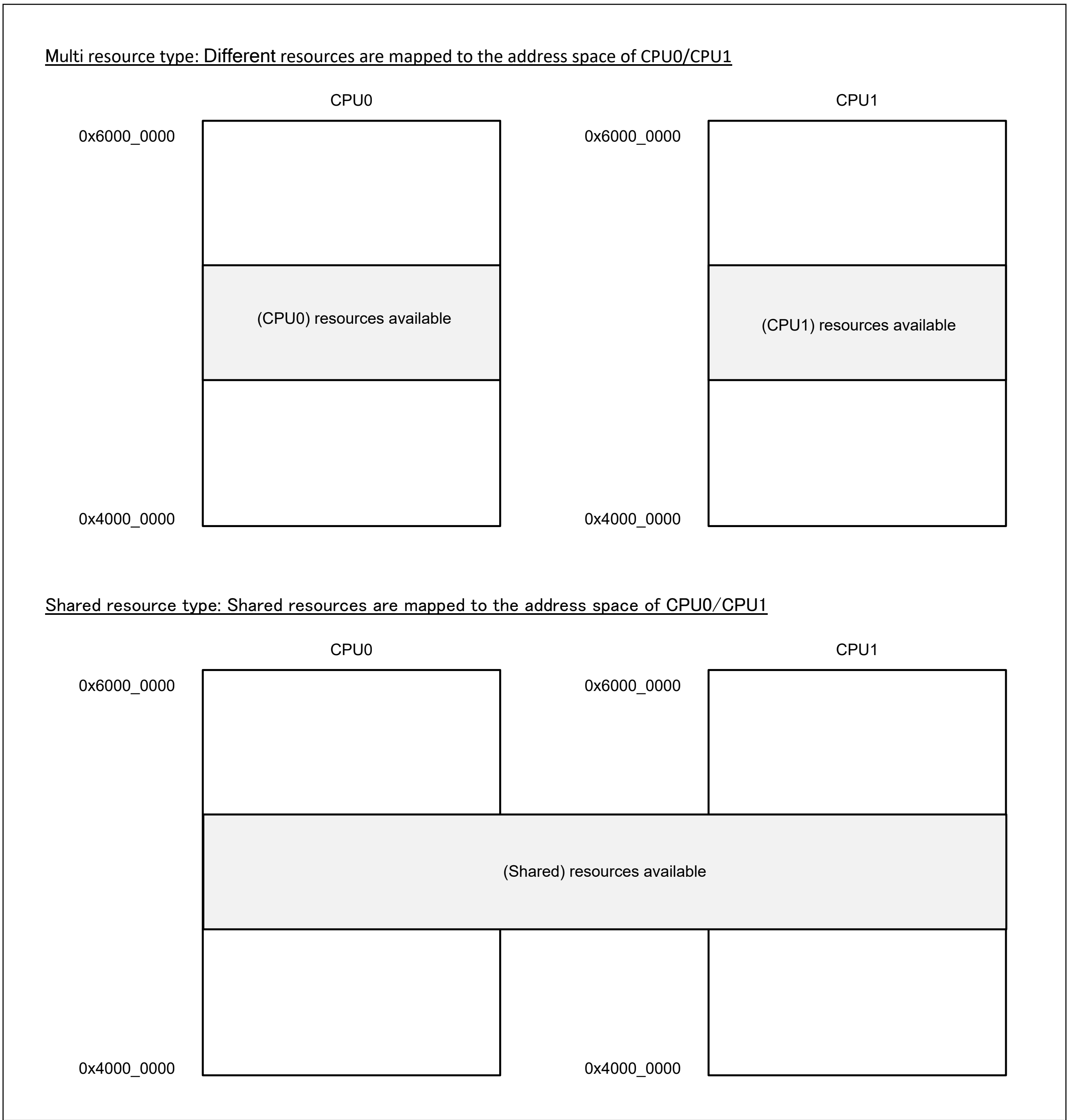


Figure 5.5 Address space types

Table 5.2 Address space types of each module (1 of 2)

Module name	Register name	Space type
ICU	NMIER, NMICLR, NMISR, WUPEN0, WUPEN1, DSLPWUPIRQENj, DELSRm, IELSRn	Multi resource type
	Other than the above	Shared resource types
DMAC	DMACCHSAR, DMACCHPAR, DMACSAR	Shared resource types
	Other than the above	Multi resource type
DTC	DTCSAR	Shared resource types
	Other than the above	Multi resource type

Table 5.2 Address space types of each module (2 of 2)

Module name	Register name	Space type
Others	All	Shared resource types

6. Resets

6.1 Overview

This MCU provides the following 21 types of reset.

It also has a function to control the reset generation by only secure program access.

System reset is all factors such that can initialize CPU core. All reset factors except VBATT_POR are system reset for this product.

On the other hand, CPU has a reset that initializes itself individually. The reset is not a system reset. See section 2, CPU.

Table 6.1 lists the reset names and sources.

Table 6.1 Reset names and sources

Reset Name		Source
System reset	RES Pin reset	Voltage input to the RES pin is driven low
	Power-on reset (POR)	VCC falls (voltage detection: V_{POR})*1
	Voltage Monitor 0 reset	VCC falls (voltage detection: V_{det0})*1
	Voltage Monitor 1 reset	VCC falls (voltage detection: V_{det1})*1 VCC rises (voltage detection: V_{det1})*1
	Voltage Monitor 2 reset	VCC falls (voltage detection: V_{det2})*1 VCC rises (voltage detection: V_{det2})*1
	Voltage Monitor 4 reset	VCC falls (voltage detection: V_{det4})*1 VCC rises (voltage detection: V_{det4})*1
	Voltage Monitor 5 reset	VCC falls (voltage detection: V_{det5})*1 VCC rises (voltage detection: V_{det5})*1
	Core Voltage Monitor reset	VDD falls (voltage detection: V_{det}_VDDL)*1 VDD rises (voltage detection: V_{det}_VDDH)*1
	Temperature Monitor reset	Detect temperature outside a certain range
	Independent Watchdog Timer reset	IWDT underflow or refresh error
	CPU0 Watchdog Timer reset	WDT underflow or refresh error for CPU0
	CPU1 Watchdog Timer reset	WDT underflow or refresh error for CPU1
	CPU0 Lockup reset	This reset is generated when CPU0 encounters lockup
	CPU1 Lockup reset	This reset is generated when CPU1 encounters lockup
	Bus Error reset	BUS error (MSAU error, MMPU error, Illegal address error, TZF error, Slave Bus error, Bufferable write error)
	Common Memory Error reset	RAM error (ECC error of SRAM)
	Local Memory 0 Error reset	RAM error (ECC error of CPU0 Cache or TCM)
	Local Memory 1 Error reset	RAM error (ECC error of CPU1 Cache or TCM)
	Deep Software Standby reset	Deep Software Standby mode is canceled by an interrupt
	Software reset	Register setting (use the software reset bit AIRCR.SYSRESETREQ)
VBATT_POR reset		VBATT_R falls (voltage detection: $V_{PORBATT}$)*1

Note 1. For details on the voltages to be monitored (V_{POR}, V_{det0}, V_{det1}, V_{det2}, V_{det4}, V_{det5}, V_{det}_VDDL, V_{det}_VDDH and $V_{PORBATT}$), see section 8, Programmable Voltage Detection (PVD), section 12, Battery Backup Function, and section 62, Electrical Characteristics.

The internal state and pins are initialized by a reset. Table 6.2 and Table 6.3 list the targets initialized by resets.

Table 6.2 Reset detect flags initialized by each reset source (1 of 4)

Flag to be initialized	Reset source: RES pin reset	Power-on reset	Voltage monitor 0 reset	Core voltage monitor reset	Voltage monitor 4 reset	Voltage monitor 5 reset	Temperature monitor reset	Independent watchdog timer reset	CPU0 watchdog timer reset	CPU1 watchdog timer reset	CPU0 lockup reset
Power-On Reset Detect Flag (RSTSR0.PORF)	✓	—	—	—	—	—	—	—	—	—	—
Voltage Monitor 0 Reset Detect Flag (RSTSR0.PVD0RF)	✓	✓	—	—	—	—	—	—	—	—	—
Core Voltage Monitor Reset Detect Flag (RSTSR3.CVMRF)	✓	✓	✓	—	—	—	—	—	—	—	—
Temperature Monitor Reset Detect Flag (RSTSR3.TEMPRF)	✓	✓	✓	—	—	—	—	—	—	—	—
Independent Watchdog Timer Reset Detect Flag (RSTSR1.IWDTRF)	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
CPU0 Watchdog Timer Reset Detect Flag (RSTSR1.WDT0RF)	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
CPU1 Watchdog Timer Reset Detect Flag (RSTSR1.WDT1RF)	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
CPU0 Lockup Reset Detect Flag (RSTSR1.CLU0RF)	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
CPU1 Lockup Reset Detect Flag (RSTSR1.CLU1RF)	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
Voltage Monitor 1 Reset Detect Flag (RSTSR0.PVD1RF)	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
Voltage Monitor 2 Reset Detect Flag (RSTSR0.PVD2RF)	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
Voltage Monitor 4 Reset Detect Flag (RSTSR0.PVD4RF)	✓	✓	✓	—	—	—	—	—	—	—	—
Voltage Monitor 5 Reset Detect Flag (RSTSR0.PVD5RF)	✓	✓	✓	—	—	—	—	—	—	—	—
Software Reset Detect Flag (RSTSR1.SWRF)	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
Bus error Reset Detect Flag (RSTSR1.BUSRF)	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
Common Memory Error Reset Detect Flag (RSTSR1.CMRF)	✓	✓	✓	✓	✓	✓	✓	—	—	—	—

Table 6.2 Reset detect flags initialized by each reset source (2 of 4)

Flag to be initialized	Reset source: RES pin reset	Power-on reset	Voltage monitor 0 reset	Core voltage monitor reset	Voltage monitor 4 reset	Voltage monitor 5 reset	Temperature monitor reset	Independent watchdog timer reset	CPU0 watchdog timer reset	CPU1 watchdog timer reset	CPU0 lockup reset
Local Memory 0 Error Reset Detect Flag (RSTSR1.LM0RF)	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
Local Memory 1 Error Reset Detect Flag (RSTSR1.LM1RF)	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
Deep Software Standby Reset Detect Flag (RSTSR0.DPSRSTF)	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
Cold Start/Warm Start Determination Flag (RSTSR2.CWSF)	—	✓	—	—	—	—	—	—	—	—	—

Table 6.2 Reset detect flags initialized by each reset source (3 of 4)

Flag to be initialized	Reset source											
	CPU1 Lockup reset	Voltage monitor 1 reset	Voltage monitor 2 reset	Software reset	Bus error reset	Common memory error reset	Local memory 0 error reset	Local memory 1 error reset	Deep software standby reset: Deep software standby mode 1	Deep software standby mode 2	Deep software standby mode 3	VBATT_POR*1
Power-On Reset Detect Flag (RSTSR0.PORF)	—	—	—	—	—	—	—	—	—	—	—	—
Voltage Monitor 0 Reset Detect Flag (RSTSR0.PVD0RF)	—	—	—	—	—	—	—	—	—	—	—	—
Core Voltage Monitor Reset Detect Flag(RSTSR3.CVMRF)	—	—	—	—	—	—	—	—	—	—	—	—
Temperature Monitor Reset Detect Flag (RSTSR3.TEMPRF)	—	—	—	—	—	—	—	—	—	—	—	—
Independent Watchdog Timer Reset Detect Flag (RSTSR1.IWDTRF)	—	—	—	—	—	—	—	—	—	✓	✓	—
CPU0 Watchdog Timer Reset Detect Flag (RSTSR1.WDT0RF)	—	—	—	—	—	—	—	—	—	✓	✓	—
CPU1 Watchdog Timer Reset Detect Flag (RSTSR1.WDT1RF)	—	—	—	—	—	—	—	—	—	✓	✓	—

Table 6.2 Reset detect flags initialized by each reset source (4 of 4)

Flag to be initialized	Reset source											
									Deep software standby reset			
	CPU1 Lockup reset	Voltage monitor 1 reset	Voltage monitor 2 reset	Software reset	Bus error reset	Common memory error reset	Local memory 0 error reset	Local memory 1 error reset	Deep software standby mode 1	Deep software standby mode 2	Deep software standby mode 3	VBATT_POR*1
CPU0 Lockup Reset Detect Flag (RSTSR1.CLU0RF)	—	—	—	—	—	—	—	—	—	✓	✓	—
CPU1 Lockup Reset Detect Flag (RSTSR1.CLU1RF)	—	—	—	—	—	—	—	—	—	✓	✓	—
Voltage Monitor 1 Reset Detect Flag (RSTSR0.PVD1RF)	—	—	—	—	—	—	—	—	—	—	—	—
Voltage Monitor 2 Reset Detect Flag (RSTSR0.PVD2RF)	—	—	—	—	—	—	—	—	—	—	—	—
Voltage Monitor 4 Reset Detect Flag (RSTSR0.PVD4RF)	—	—	—	—	—	—	—	—	—	—	—	—
Voltage Monitor 5 Reset Detect Flag (RSTSR0.PVD5RF)	—	—	—	—	—	—	—	—	—	—	—	—
Software Reset Detect Flag (RSTSR1.SWRF)	—	—	—	—	—	—	—	—	—	✓	✓	—
Bus error Reset Detect Flag (RSTSR1.BUSRF)	—	—	—	—	—	—	—	—	—	✓	✓	—
Common Memory Error Reset Detect Flag (RSTSR1.CMRF)	—	—	—	—	—	—	—	—	—	✓	✓	—
Local Memory 0 Error Reset Detect Flag (RSTSR1.LM0RF)	—	—	—	—	—	—	—	—	—	✓	✓	—
Local Memory 1 Error Reset Detect Flag (RSTSR1.LM1RF)	—	—	—	—	—	—	—	—	—	✓	✓	—
Deep Software Standby Reset Detect Flag (RSTSR0.DPSRSTF)	—	—	—	—	—	—	—	—	—	—	—	—
Cold Start/Warm Start Determination Flag (RSTSR2.CWSF)	—	—	—	—	—	—	—	—	—	—	—	—

Note: ✓ : Initialized to 0
— : Not initialized

Note 1. For VBATT_POR details, see section 12, Battery Backup Function.

Table 6.3 Module-related registers initialized by each reset source (1 of 7)

Registers to be initialized		Reset source										
		RES# pin reset	Power-on reset	Voltage monitoring 0 reset	Core Voltage monitor reset	Voltage monitor 4 reset	Voltage monitor 5 reset	Temperature monitor reset	Independent watchdog timer reset	CPU0 watchdog timer reset	CPU1 watchdog timer reset	CPU0 lockup reset
Voltage Monitor Function 1 registers	PVD1CR0, PVD1CMPCR, PVD1FCR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
	PVD1CR1, PVD1SR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
Voltage Monitor Function 2 registers	PVD2CR0, PVD2CMPCR, PVD2FCR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
	PVD2CR1, PVD2SR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
Voltage Monitor Function 4 registers	PVD4CR0, PVD4CMPCR, PVD4FCR	✓	✓	✓	—	—	—	—	—	—	—	—
	PVD4CR1, PVD4SR	✓	✓	✓	—	—	—	—	—	—	—	—
Voltage Monitor Function 5 registers	PVD5CR0, PVD5CMPCR, PVD5FCR	✓	✓	✓	—	—	—	—	—	—	—	—
	PVD5CR1, PVD5SR	✓	✓	✓	—	—	—	—	—	—	—	—
Temperature Reset Control Register	TEMPRCR	✓	✓	✓	—	—	—	—	—	—	—	—
SOSC registers	SOSCCR, SOSTDR, SOSTDSR, SOMCR	—	—	—	—	—	—	—	—	—	—	—
LOCO registers	LOCOUTCR	—	✓	✓	✓	✓	✓	✓	—	—	—	—
MOSC registers	MOMCR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
HOCO registers	HOCOCR2	—	✓	✓	✓	✓	✓	✓	—	—	—	—
Pin states (except XCIN/XCOUT)		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Pin states (XCIN/XCOUT)		—	—	—	—	—	—	—	—	—	—	—
IO capture and tamper detection such as VBAT (sampling timing for RTC) (RTCIC0-2)	VBTICTLR, VBTICTLR2, VBTADSR, VBTADCR1, VBTADCR2, VBTADCR3, VBTNCWCR	—	—	—	—	—	—	—	—	—	—	—

Table 6.3 Module-related registers initialized by each reset source (2 of 7)

Registers to be initialized		Reset source										
		RES# pin reset	Power-on reset	Voltage monitoring 0 reset	Core Voltage monitor reset	Voltage monitor 4 reset	Voltage monitor 5 reset	Temperature monitor reset	Independent watchdog timer reset	CPU0 watchdog timer reset	CPU1 watchdog timer reset	CPU0 lockup reset
VBATT Battery Power Supply Switch Control Register 1	VBTBPCR1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
VBATT Battery Power Supply Switch Control Register 2	VBTBPCR2	—	—	—	—	—	—	—	—	—	—	—
VBATT Backup Enable Register	VBTBER	—	✓	—	—	—	—	—	—	—	—	—
Battery Backup Register	VBTBKRn	—	—	—	—	—	—	—	—	—	—	—
Independent Watchdog Timer registers	IWDTRR, IWDTCR, IWDTSR, IWDTRCR, IWDTCSTPR, IWDTTS, IWDTPR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Realtime Clock*1 Register		—	—	—	—	—	—	—	—	—	—	—
Low-Power Timer registers	ULPTCNT, ULPTCMA, ULPTCMB, ULPTCR, ULPTMR1, ULPTMR2, ULPTMR3, ULPTIOC, ULPTISR, ULPTCMSR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
USBFS registers	DPUSR0R, DPUSR1R	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
USBHS registers	DPUSR0R, DPUSR1R, DPUSR2R, DPUSRCR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Reset Flag	BUSERRSTAT<Master Name>, MBWERRSTAT, SRAMESR, SRAMEARnm (n = 0 to 3, m = 0 to 1), CCAEDST	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 6.3 Module-related registers initialized by each reset source (3 of 7)

Registers to be initialized		Reset source										
		RES# pin reset	Power-on reset	Voltage monitoring 0 reset	Core Voltage monitor reset	Voltage monitor 4 reset	Voltage monitor 5 reset	Temperature monitor reset	Independent watchdog timer reset	CPU0 watchdog timer reset	CPU1 watchdog timer reset	CPU0 lockup reset
Reset Flag		See Table 6.2										
Low-Power Function registers	DPSBYCR, DPSWCR, DPSIER0 to DPSIER5, DPSIFR0 to DPSIFR5, DPSIEGR0 to DPSIEGR4 LPSCR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Low-Power Function registers	SYOCDCR	—	✓	✓	✓	✓	✓	✓	—	—	—	—
System Reset Mask Control Register	SYRSTMSK0, SYRSTMSK1, SYRSTMSK2	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
ARM Debug function	MCUSTAT, MCUCTRL, JBMDR, DBGSTR, DBGSTOPCR, DBGAUTH0, CACHEDBGCR, TRPORTCR, ALCTRL, TRPORTSZ, DBGNVMCR, Cortex-M85 registers*4, Cortex-M33 Registers*5, CoreSight Registers*6	—	✓	✓	✓	✓	✓	✓	—	—	—	—
TRCLK Control registers	TRCKCR	—	✓	✓	✓	✓	✓	✓	—	—	—	—
Power Gating Control registers	PDRAMSCR0, PDRAMSCR1, PGSCR, PSSTCRn (n = 2 to 5)	—	✓	✓	✓	✓	✓	✓	—	—	—	—
Other than the above		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 6.3 Module-related registers initialized by each reset source (4 of 7)

Registers to be initialized		Reset source											
										Deep software standby reset			
		CPU1 Lockup reset	Voltage monitoring 1 reset	Voltage monitoring 2 reset	Software reset	Bus error reset	Common memory error reset	Local memory 0 error reset	Local memory 1 error reset	Deep software standby mode 1 reset	Deep software standby mode 2 reset	Deep software standby mode 3 reset	VBATT-selected voltage power-on reset
Voltage Monitor Function 1 registers	PVD1CR0, PVD1CMPCR, PVD1FCR	—	—	—	—	—	—	—	—	—	—	—	—
	PVD1CR1, PVD1SR	—	—	—	—	—	—	—	—	✓	✓	✓	—
Voltage Monitor Function 2 registers	PVD2CR0, PVD2CMPCR, PVD2FCR	—	—	—	—	—	—	—	—	—	—	—	—
	PVD2CR1, PVD2SR	—	—	—	—	—	—	—	—	✓	✓	✓	—
Voltage Monitor Function 4 registers	PVD4CR0, PVD4CMPCR, PVD4FCR	—	—	—	—	—	—	—	—	—	—	—	—
	PVD4CR1, PVD4SR	—	—	—	—	—	—	—	—	✓	✓	✓	—
Voltage Monitor Function 5 registers	PVD5CR0, PVD5CMPCR, PVD5FCR	—	—	—	—	—	—	—	—	—	—	—	—
	PVD5CR1, PVD5SR	—	—	—	—	—	—	—	—	✓	✓	✓	—
Temperature Reset Control Register	TEMPRCR	—	—	—	—	—	—	—	—	—	—	—	—
SOSC registers	SOSCCR, SOSTDR, SOSTDSR, SOMCR	—	—	—	—	—	—	—	—	—	—	—	✓
LOCO registers	LOCOUTCR	—	—	—	—	—	—	—	—	—	✓	✓	—
MOSC registers	MOMCR	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
HOCO registers	HOCOCR2	—	—	—	—	—	—	—	—	✓*3	✓*3	✓*3	—
Pin states (except XCIN/XCOUT)		✓	✓	✓	✓	✓	✓	✓	✓	*2	*2	*2	—
Pin states (XCIN/XCOUT)		—	—	—	—	—	—	—	—	—	—	—	✓

Table 6.3 Module-related registers initialized by each reset source (5 of 7)

Registers to be initialized		Reset source											
										Deep software standby reset			
		CPU1 Lockup reset	Voltage monitoring 1 reset	Voltage monitoring 2 reset	Software reset	Bus error reset	Common memory error reset	Local memory 0 error reset	Local memory 1 error reset	Deep software standby mode 1 reset	Deep software standby mode 2 reset	Deep software standby mode 3 reset	VBATT-selected voltage power-on reset
IO capture and tamper detection such as VBAT (sampling timing for RTC) (RTCIC0-2)	VBTICTLR, VBTICTLR2, VBTADSR, VBTADCR1, VBTADCR2, VBTADCR3, VBTNCWCR	—	—	—	—	—	—	—	—	—	—	—	✓
VBATT Battery Power Supply Switch Control Register 1	VBTBPCR1	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
VBATT Battery Power Supply Switch Control Register 2	VBTBPCR2	—	—	—	—	—	—	—	—	—	—	—	✓
VBATT Backup Enable Register	VBTBER	—	—	—	—	—	—	—	—	—	—	—	—
Battery Backup Register	VBTBKRn	—	—	—	—	—	—	—	—	—	—	—	✓
Independent Watchdog Timer registers	IWDTRR, IWDTCR, IWDTSR, IWDTRCR, IWDTCSTPR, IWDTTS, IWDTPR	✓	✓	✓	✓	✓	✓	✓	✓	—	✓	✓	—
Realtime Clock*1 Register		—	—	—	—	—	—	—	—	—	—	—	—

Table 6.3 Module-related registers initialized by each reset source (6 of 7)

Registers to be initialized		Reset source											
										Deep software standby reset			
		CPU1 Lockup reset	Voltage monitoring 1 reset	Voltage monitoring 2 reset	Software reset	Bus error reset	Common memory error reset	Local memory 0 error reset	Local memory 1 error reset	Deep software standby mode 1 reset	Deep software standby mode 2 reset	Deep software standby mode 3 reset	VBATT-selected voltage power-on reset
Low-Power Timer registers	ULPTCNT, ULPTCMA, ULPTCMB, ULPTCR, ULPTMR1, ULPTMR2, ULPTMR3, ULPTIOC, ULPTISR, ULPTCMSR	✓	✓	✓	✓	✓	✓	✓	✓	—	✓	✓	—
USBFS registers	DPUSR0R, DPUSR1R	✓	✓	✓	✓	✓	✓	✓	✓	—	✓	✓	—
USBHS registers	DPUSR0R, DPUSR1R, DPUSR2R, DPUSRCR	✓	✓	✓	✓	✓	✓	✓	✓	—	✓	✓	—
Reset Flag	BUSERRSTAT<Master Name>, MBWERRSTAT, SRAMESR, SRAMEARnm (n = 0 to 3, m = 0 to 1), CCAEDST	✓	✓	✓	✓	—	—	—	—	✓	✓	✓	—
Reset Flag		See Table 6.2.											
Low-Power Function registers	DPSBYCR, DPSWCR, DPSIER0 to DPSIER5, DPSIFR0 to DPSIFR5, DPSIEGR0 to DPSIEGR4, LPSCR	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
Low-Power Function registers	SYOCDCR	—	—	—	—	—	—	—	—	—	—	—	—
System Reset Mask Control Register	SYRSTMSK0, SYRSTMSK1, SYRSTMSK2	—	—	—	—	—	—	—	—	—	—	—	—

Table 6.3 Module-related registers initialized by each reset source (7 of 7)

Registers to be initialized		Reset source											
										Deep software standby reset			
		CPU1 Lockup reset	Voltage monitoring 1 reset	Voltage monitoring 2 reset	Software reset	Bus error reset	Common memory error reset	Local memory 0 error reset	Local memory 1 error reset	Deep software standby mode 1 reset	Deep software standby mode 2 reset	Deep software standby mode 3 reset	VBATT-selected voltage power-on reset
ARM Debug function	MCUSTAT, MCUCTRL, JBMDR, DBGSTR, DBGSTOPCR, DBGAUTH0, CACHEDBGCR, TRPORTCR, ALCTRL, TRPORTSZ, DBGNVMCR, Cortex-M85 registers*4, Cortex-M33 registers*5, CoreSight registers*6	—	—	—	—	—	—	—	—	✓*3	✓*3	✓*3	—
TRCLK Control registers	TRCKCR	—	—	—	—	—	—	—	—	✓*3	✓*3	✓*3	—
Power Gating Control registers	PDRAMSCR0, PDRAMSCR1, PGSCR, PSSTCRn (n = 2 to 5)	—	—	—	—	—	—	—	—	✓*3	✓*3	✓*3	—
Other than the above		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—

Note: ✓ : Initialized
— : Not initialized

Note 1. RTC has Software reset for RTC. Some control bits are not initialized by all types of resets. For details on the target bits, see section 9, Clock Generation Circuit.

Note 2. Depend on setting of DPSBYCR.IOKEEP.

Note 3. After the debugger authentication passes, setting SYOCDCR.DBGEN to 1 does not initialize this register or the function. For details, see section 2, CPU.

Note 4. Cortex-M85 registers initialized on cold reset.

Note 5. Cortex-M33 registers initialized on cold reset.

Note 6. CoreSight SoC-400 TPIU registers, CoreSight SoC-400 ATB Funnel registers, CoreSight SoC-400 ATB Replicator registers, CoreSight SoC-400 CTI Replicator registers, CoreSight SoC-400 AHB-AP registers, CoreSight SoC-400 APB-AP registers, CoreSight SoC-400 Debug Port registers, CoreSight SoC-400 Timestamp Generator registers, CoreSight Trace Memory Controller registers.

The RTC is not initialized by any reset source. SOSC and LOCO can be selected as the clock sources of the RTC.

Table 6.4 and Table 6.5 show the states of SOSC and LOCO when a reset occurs.

Table 6.4 States of SOSC when a reset occurs

		Reset source	
		VBAT_POR reset	Other
SOSC	Enable or disable	Initialized to enable	Continue with the state that was selected before the reset occurred
	Drive capability	Initialized to standard drive capability	

Table 6.5 States of LOCO when a reset occurs

		Reset source	
		Power-on reset, Voltage-Monitor 0, 4, 5 reset, Core Voltage-Monitor reset, Temperature-Monitor reset and Deep software standby 2, 3 reset	Other
LOCO	Enable or disable	Initialized to enable	
	Oscillation accuracy*1	Initialized to accuracy before trimming by LOCOUTCR (accuracy: ± 15%)	Continue with the accuracy that was trimmed by LOCOUTCR

Note 1. If the LOCO that is trimmed by LOCOUTCR is selected as the RTC source clock, please be careful that LOCO oscillation accuracy will be initialized by Power-on reset, Voltage-Monitor 0, 4, 5 reset, Core Voltage-Monitor reset, Temperature-Monitor reset and Deep software standby 2, 3 reset.

When a reset is released, reset exception handling starts. For details on the reset exception handling, see section 6.3.14. Determination of Reset Generation Source.

Table 6.6 lists the pin related to the reset function.

Table 6.6 Pin related to reset

Pin name	I/O	Function
RES	Input	Reset pin

6.2 Register Descriptions

6.2.1 RSTSAR : Reset Security Attribution Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x3C4

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	NONSEC3	NONSEC2	NONSEC1	NONSEC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NONSEC0	Non-secure Attribute bit 0 Target register: Reset Status Register 0 0: Secure 1: Non-secure	R/W
1	NONSEC1	Non-secure Attribute bit 1 Target register: Reset Status Register 1 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
2	NONSEC2	Non-secure Attribute bit 2 Target register: Reset Status Register 2 0: Secure 1: Non-secure	R/W
3	NONSEC3	Non-secure Attribute bit 3 Target register: Reset Status Register 3 0: Secure 1: Non-secure	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

NONSEC0 bit (Non-secure Attribute bit 0)

This bit controls the security attribute of RSTSR0.

NONSEC1 bit (Non-secure Attribute bit 1)

This bit controls the security attribute of RSTSR1.

NONSEC2 bit (Non-secure Attribute bit 2)

This bit controls the security attribute of RSTSR2.

NONSEC3 bit (Non-secure Attribute bit 3)

This bit controls the security attribute of RSTSR3.

6.2.2 RSTSR0 : Reset Status Register 0

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA40

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DPSR STF	PVD5 RF	PVD4 RF	—	PVD2 RF	PVD1 RF	PVD0 RF	PORF
Value after reset:	x[*1]	x[*1]	x[*1]	0	x[*1]	x[*1]	x[*1]	x[*1]

Bit	Symbol	Function	R/W
0	PORF	Power-On Reset Detect Flag 0: Power-on reset not detected 1: Power-on reset detected	R/W[*2]
1	PVD0RF	Voltage Monitor 0 Reset Detect Flag 0: Voltage monitor 0 reset not detected 1: Voltage monitor 0 reset detected	R/W[*2]
2	PVD1RF	Voltage Monitor 1 Reset Detect Flag 0: Voltage monitor 1 reset not detected 1: Voltage monitor 1 reset detected	R/W[*2]
3	PVD2RF	Voltage Monitor 2 Reset Detect Flag 0: Voltage monitor 2 reset not detected 1: Voltage monitor 2 reset detected	R/W[*2]
4	—	This bit is read as 0. The write value should be 0.	R/W
5	PVD4RF	Voltage Monitor 4 Reset Detect Flag 0: Voltage monitor 4 reset not detected 1: Voltage monitor 4 reset detected	R/W[*2]
6	PVD5RF	Voltage Monitor 5 Reset Detect Flag 0: Voltage monitor 5 reset not detected 1: Voltage monitor 5 reset detected	R/W[*2]

Bit	Symbol	Function	R/W
7	DPSRSTF	Deep Software Standby Reset Flag 0: Deep Software Standby Mode cancellation not requested by an interrupt or a reset*3. 1: Deep Software Standby Mode cancellation requested by an interrupt or a reset*3.	R/W*2

Note: S-TYPE-3, P-TYPE-2
Note 1. The value after reset depends on the reset source.
Note 2. Only 0 can be written to clear the flag. The flag must be cleared by writing 0 after 1 is read.
Note 3. Independent watchdog timer reset, Voltage Monitor 1 reset, Voltage Monitor 2 reset.

PORF flag (Power-On Reset Detect Flag)

The PORF flag indicates that a power-on reset occurred.

[Setting condition]

- When a power-on reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 0 is written after 1 is read from PORF flag.

PVD0RF flag (Voltage Monitor 0 Reset Detect Flag)

The PVD0RF flag indicates that the voltage monitor 0 reset has occurred.

[Setting condition]

- When a voltage monitor 0 reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When 0 is written after 1 is read from PVD0RF flag.

PVD1RF flag (Voltage Monitor 1 Reset Detect Flag)

The PVD1RF flag indicates that the voltage monitor 1 reset has occurred.

[Setting condition]

- When a voltage monitor 1 reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When 0 is written after 1 is read from PVD1RF flag.

PVD2RF flag (Voltage Monitor 2 Reset Detect Flag)

The PVD2RF flag indicates that the voltage monitor 2 reset has occurred.

[Setting condition]

- When a voltage monitor 2 reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When 0 is written after 1 is read from PVD2RF flag.

PVD4RF flag (Voltage Monitor 4 Reset Detect Flag)

The PVD4RF flag indicates that the voltage monitor 4 reset has occurred.

[Setting condition]

- When a voltage monitor 4 reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When 0 is written after 1 is read from PVD4RF flag.

PVD5RF flag (Voltage Monitor 5 Reset Detect Flag)

The PVD5RF flag indicates that the voltage monitor 5 reset has occurred.

[Setting condition]

- When a voltage monitor 5 reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When 0 is written after 1 is read from PVD5RF flag.

DPSRSTF flag (Deep Software Standby Reset Flag)

The DPSRSTF flag indicates that Deep Software Standby Mode is canceled by an external or internal interrupt, and that an internal reset (Deep Software Standby reset) occurs when an exception from Deep Software Standby Mode occurs.

[Setting condition]

- When Deep Software Standby Mode is cancelled by an external or an internal interrupt. For details, see section 11, Low Power Mode.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When 0 is written after 1 is read from DPSRSTF flag.

6.2.3 RSTSR1 : Reset Status Register 1

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x0C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	LM1RF	CLU1RF	—	—	WDT1RF	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	x*1	x*1	0	0	x*1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	CMRF	—	—	—	BUSRF	—	—	—	—	LM0RF	CLU0RF	—	SWRF	WDT0RF	IWDTRF
Value after reset:	0	x*1	0	0	0	x*1	0	0	0	0	x*1	x*1	0	x*1	x*1	x*1

Bit	Symbol	Function	R/W
0	IWDTRF	Independent Watchdog Timer Reset Detect Flag 0: Independent watchdog timer reset not detected 1: Independent watchdog timer reset detected	R/W*2
1	WDT0RF	Watchdog Timer 0 Reset Detect Flag 0: Watchdog timer 0 reset not detected 1: Watchdog timer 0 reset detected	R/W*2
2	SWRF	Software Reset Detect Flag 0: Software reset not detected 1: Software reset detected	R/W*2
3	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
4	CLU0RF	CPU0 Lockup Reset Detect Flag 0: CPU0 lockup reset not detected 1: CPU0 lockup reset detected	R/W
5	LM0RF	Local memory 0 error Reset Detect Flag 0: Local memory 0 error reset not detected 1: Local memory 0 error reset detected	R/W*2
9:6	—	These bits are read as 0. The write value should be 0.	R/W
10	BUSRF	Bus Error Reset Detect Flag 0: Bus error reset not detected 1: Bus error reset detected	R/W*2
13:11	—	These bits are read as 0. The write value should be 0.	R/W
14	CMRF	Common Memory Error Reset Detect Flag 0: Common memory error reset not detected 1: Common memory error reset detected	R/W*2
16:15	—	These bits are read as 0. The write value should be 0.	R/W
17	WDT1RF	Watchdog Timer 1 Reset Detect Flag 0: Watchdog timer 1 reset not detected 1: Watchdog timer 1 reset detected	R/W*2
19:18	—	These bits are read as 0. The write value should be 0.	R/W
20	CLU1RF	CPU1 Lockup Reset Detect Flag 0: CPU1 lockup reset not detected 1: CPU1 lockup reset detected	R/W*2
21	LM1RF	Local memory 1 error Reset Detect Flag 0: Local memory 1 error reset not detected 1: Local memory 1 error reset detected	R/W*2
31:22	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note 1. The value after reset depends on the reset source.
Note 2. Only 0 can be written to clear the flag. The flag must be cleared by writing 0 after 1 is read.

IWDTRF flag (Independent Watchdog Timer Reset Detect Flag)

The IWDTRF flag indicates that an Independent watchdog timer reset occurred.

[Setting condition]

- When an independent watchdog timer reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When 0 is written after 1 is read from IWDTRF flag.

WDT0RF flag (Watchdog Timer 0 Reset Detect Flag)

The WDT0RF flag indicates that a CPU0 watchdog timer reset occurred.

[Setting condition]

- When a CPU0 watchdog timer reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When 0 is written after 1 is read from WDT0RF flag.

SWRF flag (Software Reset Detect Flag)

The SWRF flag indicates that a Software reset occurred.

[Setting condition]

- When a software reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When 0 is written after 1 is read from SWRF flag.

CLU0RF flag (CPU0 Lockup Reset Detect Flag)

The CLU0RF flag indicates that a CPU0 lockup reset occurred.

[Setting condition]

- When a CPU0 Lockup reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When 0 is written after 1 is read from CLU0RF flag.

LM0RF flag (Local memory 0 error Reset Detect Flag)

The LM0RF flag indicates that a Local memory 0 error reset (ECC error of CPU0 Cache or TCM) occurred.

[Setting condition]

- When a Local memory 0 error reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When 0 is written after 1 is read from LM0RF flag.

BUSRF flag (Bus Error Reset Detect Flag)

The BUSRF flag indicates that BUS error reset (MSAU error, MMPU error, Illegal address error, TZF error, Slave Bus error, Bufferable write error) occurred.

[Setting condition]

- When a bus slave MPU error reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 0 is written after 1 is read from BUSRF flag.

There are several causes for setting this reset flag.

If you want to identify the reset factor, see section 15, Buses.

CMRF flag (Common Memory Error Reset Detect Flag)

The CMRF flag indicates that a Common memory error reset (ECC error of SRAM) occurred.

[Setting condition]

- When a Common memory error reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When 0 is written after 1 is read from CMRF flag.

There are several causes for setting this reset flag.

If you want to identify the reset factor, see section 58, SRAM.

WDT1RF flag (Watchdog Timer 1 Reset Detect Flag)

The WDT1RF flag indicates that a CPU1 watchdog timer reset occurred.

[Setting condition]

- When a CPU1 watchdog timer reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When 0 is written after 1 is read from WDT1RF flag.

CLU1RF flag (CPU1 Lockup Reset Detect Flag)

The CLU1RF flag indicates that a CPU1 lockup reset occurred.

[Setting condition]

- When a CPU1 lockup reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When 0 is written after 1 is read from CLU1RF flag.

LM1RF flag (Local memory 1 error Reset Detect Flag)

The LM1RF flag indicates that a Local memory 1 error reset (ECC error of CPU1 Cache or TCM) occurred.

[Setting condition]

- When a Local memory 1 error reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When 0 is written after 1 is read from LM1RF flag.

6.2.4 RSTSR2 : Reset Status Register 2

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA44

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

Bit	Symbol	Function	R/W
0	CWSF	Cold/Warm Start Determination Flag 0: Cold start 1: Warm start	R/W*2
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note 1. The value after reset depends on the reset source.
Note 2. Only 1 can be written to set the flag.

RSTSR2 determines whether a power-on reset caused the reset processing (cold start) or a reset signal input during operation caused the reset processing (warm start).

CWSF flag (Cold/Warm Start Determination Flag)

The CWSF flag indicates the type of reset processing, either cold start or warm start. CWSF flag is initialized by a power-on reset. It is not initialized by a reset signal generated by the RES pin.

[Setting condition]

- When 1 is written by software. Writing 0 to CWSF does not set it to 0.

[Clearing condition]

- When a reset listed in Table 6.2 occurs.

6.2.5 RSTSR3 : Reset Status Register 3

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA48

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TEMPRF	—	—	—	—	—	—	CVMRF
Value after reset:	x*1	0	0	0	0	0	0	x*1

Bit	Symbol	Function	R/W
0	CVMRF	Core Voltage Monitor Reset Detect Flag 0: Core voltage monitor reset not detected 1: Core voltage monitor reset detected	R/W*2
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	TEMPRF	Temperature Monitor Reset Detect Flag 0: Temperature monitor reset not detected 1: Temperature monitor reset detected	R/W*2

Note: S-TYPE-3, P-TYPE-2
Note 1. The value after reset depends on the reset source.
Note 2. Only 0 can be written to clear the flag. The flag must be cleared by writing 0 after 1 is read.

CVMRF flag (Core Voltage Monitor Reset Detect Flag)

The CVMRF indicates that Core voltage fell below V_{det}_VDDL.

The CVMRF indicates that Core voltage rised above V_{det}_VDDH.

[Setting condition]

- When a Core voltage monitor reset occurs.

[Clearing condition]

- When a reset listed in Table 6.3 occurs.
- When 0 is written after 1 is read from CVMRF flag.

TEMPRF flag (Temperature Monitor Reset Detect Flag)

The TEMPRF indicates that Temperature monitor reset has occurred.

[Setting condition]

- When a Temperature monitor reset occurs.

[Clearing condition]

- When a reset listed in Table 6.3 occurs.
- When 0 is written after 1 is read from TEMPRF flag.

6.2.6 SYRSTMSK0 : System Reset Mask Control Register 0

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xAD0

Bit position:	7	6	5	4	3	2	1	0
Bit field:	BUSMASK	CMMASK	LM0MASK	CLU0MASK	—	SWMASK	WDT0MASK	IWDTMASK
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IWDTMASK	Independent Watchdog Timer Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W
1	WDT0MASK	CPU0 Watchdog Timer Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W
2	SWMASK	Software Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	CLU0MASK	CPU0 Lockup Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W
5	LM0MASK	Local Memory 0 Error Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W
6	CMMASK	Common Memory Error Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W
7	BUSMASK	Bus Error Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W

Note: S-TYPE-6, P-TYPE-2
Note: Set the PRCR.PRC5 bit to 1 (write enabled) before rewriting this register.

The SYRSTMSK0 is a register that controls the occurrence of reset.

The IWDTMASK bit cannot be rewritten while the independent watchdog timer is operating. The WDT0MASK bit cannot be rewritten while the CPU0 watchdog timer is operating.

6.2.7 SYRSTMSK1 : System Reset Mask Control Register 1

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xAD4

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	LM1MASK	CLU1MASK	—	—	WDT1MASK	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	WDT1MASK	CPU1 Watchdog Timer Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W

Bit	Symbol	Function	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	CLU1MASK	CPU1 Lockup Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W
5	LM1MASK	Local Memory 1 Error Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-6, P-TYPE-2
Note: Set the PRCR.PRC5 bit to 1 (write enabled) before rewriting this register.

The SYRSTMSK1 is a register that controls the occurrence of reset.

6.2.8 SYRSTMSK2 : System Reset Mask Control Register 2

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xAD8

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PVD2 MASK	PVD1 MASK
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PVD1MASK	Voltage Monitor 1 Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W
1	PVD2MASK	Voltage Monitor 2 Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-6, P-TYPE-2
Note: Set the PRCR.PRC5 bit to 1 (write enabled) before rewriting this register.

The SYRSTMSK2 is a register that controls the occurrence of reset.

6.2.9 TEMPRCR : Temperature Monitor Reset Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xADC

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	TSNK EEP	CMPE N	TSNE N	TEMP REN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TEMPREN	Temperature Reset Enable 0: Temperature reset is disabled 1: Temperature reset is enabled	R/W
1	TSNEN	Temperature Sensor Enable 0: Temperature sensor is disabled 1: Temperature sensor is enabled	R/W

Bit	Symbol	Function	R/W
2	CMPEN	Comparator Enable 0: Comparator is disabled 1: Comparator is enabled	R/W
3	TSNKEEP	Temperature Sensor Latch Control 0: Temperature sensor latch is opened 1: Temperature sensor latch is closed	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-6, P-TYPE-2
Note: Set the PRCR.PRC5 bit to 1 (write enabled) before rewriting this register.

6.2.10 TEMPRLR : Temperature Monitor Reset Lock Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xAE0

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

Bit	Symbol	Function	R/W
0	LOCK	Temperature Monitor Reset Control Register Lock 0: Temperature monitor reset control register is unlocked 1: Temperature monitor reset control register is locked	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-6, P-TYPE-2
Note: Set the PRCR.PRC5 bit to 1 (write enabled) before rewriting this register.
Note: This register accepts write accesses only twice. If you write to TEMPRLR 3 times or more, please reset it and re-write.

6.3 Operation

6.3.1 RES Pin Reset

The RES pin generates this reset. When the RES pin is driven low, all the processing in progress is aborted and the MCU enters a reset state. To successfully reset the MCU, the RES pin must be held low for the power supply stabilization time specified at power-on.

When the RES pin is driven high from low, the internal reset is canceled after the post-RES cancellation wait time (t_{RESWT}) elapses. The CPU then starts the reset exception handling.

For details, see section 62, Electrical Characteristics.

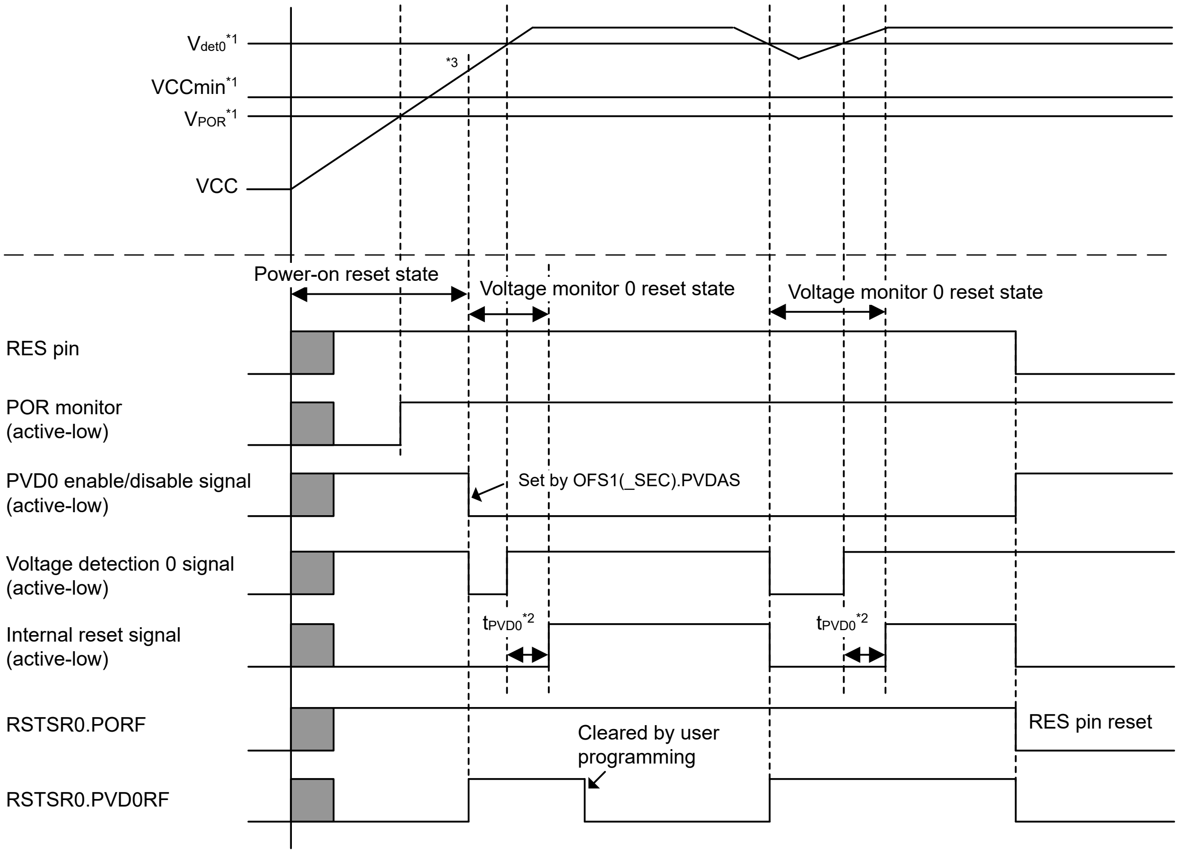
6.3.2 Power-On Reset

The power-on reset circuit generates this internal reset. If the RES pin is in a high-level state when power is supplied, a power-on reset is generated.

After VCC exceeds V_{POR} and the specified period (power-on reset time) elapses, the internal reset is canceled, and the CPU starts the reset exception handling. The power-on reset time is the stabilization period for the external power supply and the MCU circuit.

After a power-on reset is generated, the PORF flag in the RSTSR0 is set to 1. The PORF flag is initialized by the RES pin reset.

Figure 6.1 shows examples of operations during a power-on reset and voltage monitor 0 reset.



Note: For details on the electrical characteristics, see section 62, Electrical Characteristics.

Note 1. V_{det0} shows a voltage monitor 0 reset detection level, V_{POR} shows a power-on reset detection level, and VCCmin shows minimum guaranteed voltage of MCU.

Note 2. tPVD0 shows the reset time of Voltage Monitor 0.

Note 3. At power-on, VCC should rise to the minimum guaranteed voltage before the power-on reset is released.

Figure 6.1 Example of operations during a power-on reset and Voltage Monitor 0 Reset

6.3.3 Voltage Monitor Reset

The voltage monitor 0 reset is an internal reset generated by the voltage monitor circuit. If the Voltage Detection 0 Circuit Start bit (PVDAS) in Option Function Select Register 1 (OFS1) is 0 (voltage monitor 0 reset is enabled after a reset) and VCC falls below V_{det0}, the RSTSR0.PVD0RF flag is set to 1 and the voltage detection circuit generates a voltage monitor 0 reset. Clear the OFS1.PVDAS bit to 0 if the voltage monitor 0 reset is to be used. After VCC exceeds V_{det0} and the voltage monitor 0 reset time (t_{PVD0}) elapses, the internal reset is canceled, and the CPU starts the reset exception handling. The V_{det0} voltage detection level can be changed by the setting in the VDSEL[1:0] bits in Option Function Select Register 1 (OFS1).

- Case of PVDiFCR.RHSEL = 0 and PVDjFCR.RHSEL = 0 (i = 1, 2, j = 4, 5)

When the Voltage Monitor 1 Interrupt/Reset Enable bit (RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the Voltage Monitor 1 Circuit Mode Select bit (RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in Voltage Monitor 1 Circuit Control Register 0 (PVD1CR0), the RSTSR0.PVD1RF flag is set to 1 and the voltage detection circuit generates a Voltage Monitor 1 reset if VCC falls to or below V_{det1}.

Timing for release from the Voltage Monitor 1 reset state is selectable in the Voltage Monitor 1 Reset Negate Select bit (RN) in PVD1CR0. When the RN bit is 0 and VCC falls to or below Vdet1, the CPU is released from the internal reset state and starts reset exception handling when the PVD1 reset time (tPVD1) elapses after VCC rises above V_{det1}. When the PVD1CR0.RN bit is 1 and VCC falls to or below Vdetm, the CPU is released from the internal reset state and starts reset exception handling when the PVD1 reset time (tPVD1) elapses.

Likewise, when the Voltage Monitor 2 Interrupt/Reset Enable bit (RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the Voltage Monitor 2 Circuit Mode Select bit (RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in Voltage Monitor 2 Circuit Control Register 0 (PVD2CR0), the RSTSR0.PVD2RF flag is set to 1 and the voltage detection circuit generates a Voltage Monitor 2 reset if VCC falls to or below V_{det2}.

Timing for release from the Voltage Monitor 2 reset state is selectable in the Voltage Monitor 2 Reset Negate Select bit (RN) in PVD2CR0. When the RN bit is 0 and VCC falls to or below V_{det2}, the CPU is released from the internal reset state and starts reset exception handling when the PVD2 reset time (tPVD2) elapses after VCC rises above V_{det2}. When the PVD2CR0.RN bit is 1 and VCC falls to or below Vdetm, the CPU is released from the internal reset state and starts reset exception handling when the PVD2 reset time (tPVD2) elapses.

When the Voltage Monitor 4 Reset Enable bit (RE) is set to 1 (enabling generation of a reset by the voltage detection circuit) in Voltage Monitor 4 Circuit Control Register 0 (PVD4CR0), the RSTSR0.PVD4RF flag is set to 1 and the voltage detection circuit generates a Voltage Monitor 4 reset if VCC falls to or below V_{det4}.

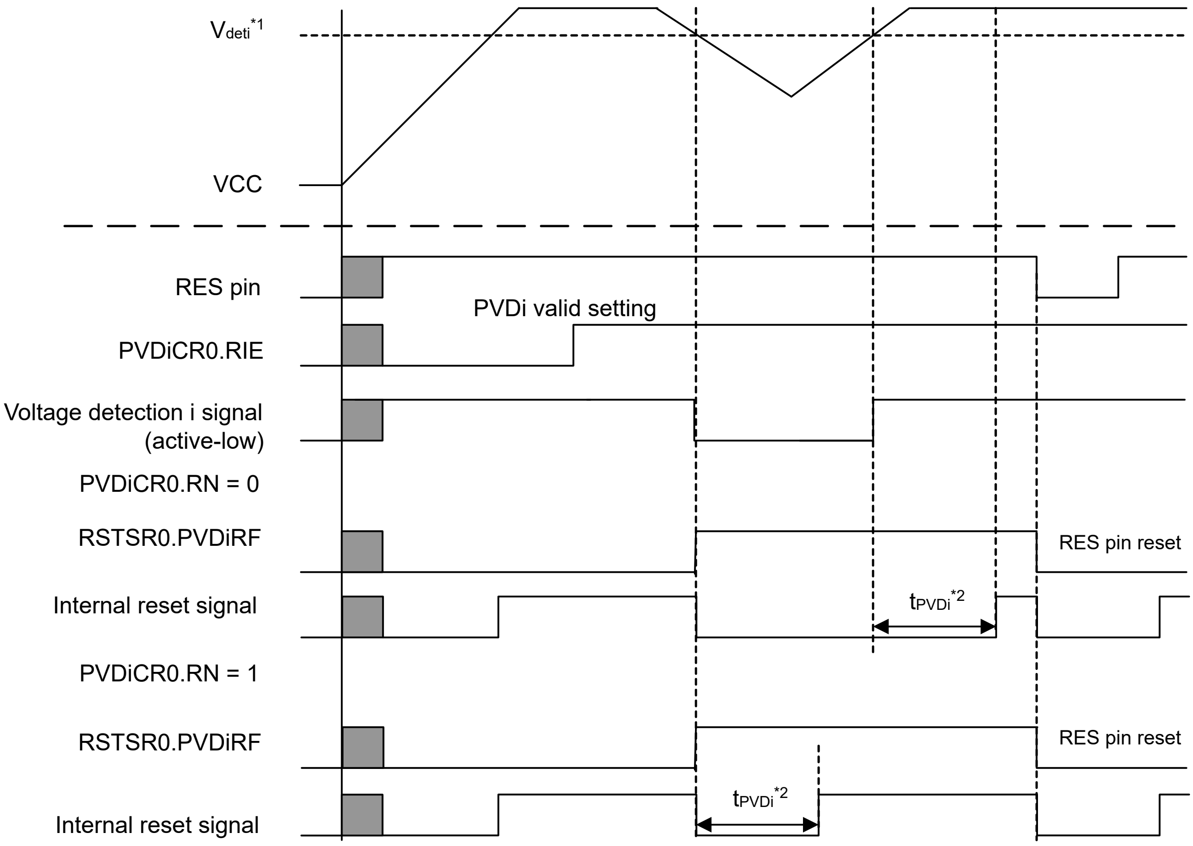
The CPU is then released from the internal reset state and starts reset exception handling when the PVD4 reset time (tPVD4) elapses after VCC rises above V_{det4}.

When the Voltage Monitor 5 Reset Enable bit (RE) is set to 1 (enabling generation of a reset by the voltage detection circuit) in Voltage Monitor 5 Circuit Control Register 0 (PVD5CR0), the RSTSR0.PVD5RF flag is set to 1 and the voltage detection circuit generates a Voltage Monitor 5 reset if VCC falls to or below V_{det5}.

The CPU is then released from the internal reset state and starts reset exception handling when the PVD5 reset time (tPVD5) elapses after VCC rises above V_{det5}.

Figure 6.2 shows examples of operations during the voltage monitor 1 and 2 reset.

Figure 6.3 shows examples of operations during the voltage monitor 4 and 5 reset.

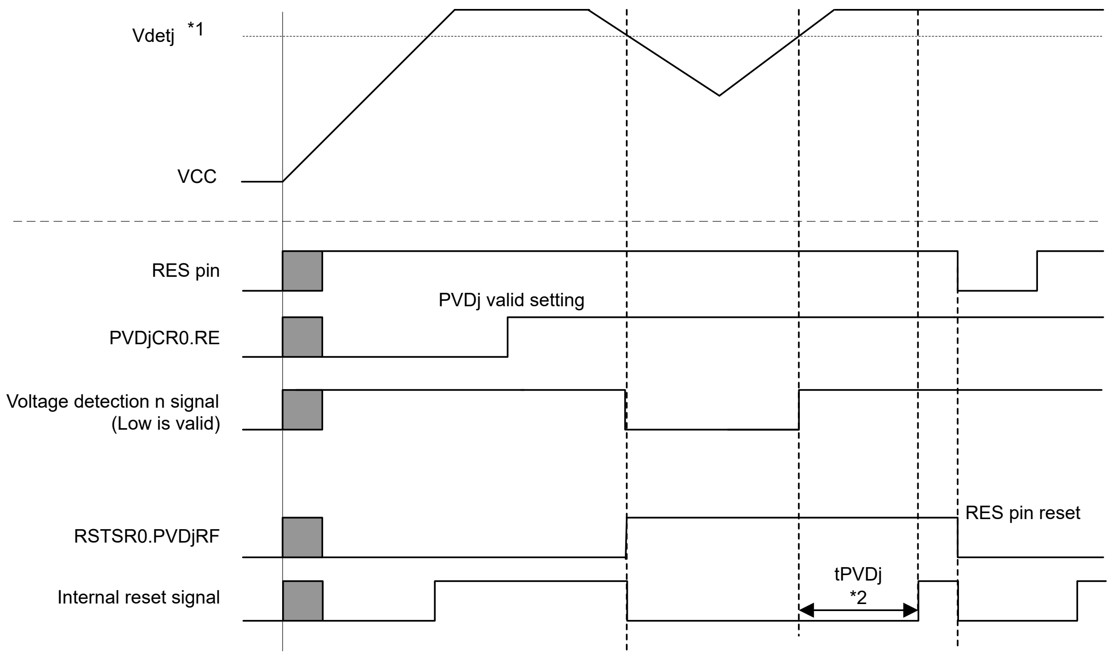


Note: For details on the electrical characteristics, see section 62, Electrical Characteristics.

Note 1. V_{deti} indicates the detection level of the voltage monitor i reset. (i = 1, 2)

Note 2. t_{PVDi} indicates the time for the voltage monitor i reset. (i = 1, 2)

Figure 6.2 Example of operations during voltage monitor i reset (i = 1, 2)



Note: For details on the electrical characteristics, see section 62, Electrical Characteristics.
Note 1. Vdetj shows a detection level of voltage monitor j reset. (j = 4, 5)
Note 2. tPVDj shows a time for voltage monitor j reset. (j = 4, 5)

Figure 6.3 Example of operations during voltage monitor j reset (j = 4, 5)

- Case of PVDiFCR.RHSEL = 1 and PVDjFCR.RHSEL = 1 (i = 1, 2, j = 4, 5)

When the Voltage Monitor 1 Interrupt/Reset Enable bit (RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the Voltage Monitor 1 Circuit Mode Select bit (RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in Voltage Monitor 1 Circuit Control Register 0 (PVD1CR0), the RSTSR0.PVD1RF flag is set to 1 and the voltage detection circuit generates a Voltage Monitor 1 reset if VCC rises to or above V_{det1}.

The CPU is then released from the internal reset state and starts reset exception handling when the PVD1 reset time (tPVD1) elapses after VCC falls below V_{det1}.

Likewise, when the Voltage Monitor 2 Interrupt/Reset Enable bit (RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the Voltage Monitor 2 Circuit Mode Select bit (RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in Voltage Monitor 2 Circuit Control Register 0 (PVD2CR0), the RSTSR0.PVD2RF flag is set to 1 and the voltage detection circuit generates a Voltage Monitor 2 reset if VCC rises to or above V_{det2}.

The CPU is then released from the internal reset state and starts reset exception handling when the PVD2 reset time (tPVD2) elapses after VCC falls below V_{det2}.

When the Voltage Monitor 4 Reset Enable bit (RE) is set to 1 (enabling generation of a reset by the voltage detection circuit) in Voltage Monitor 4 Circuit Control Register 0 (PVD4CR0), the RSTSR0.PVD4RF flag is set to 1 and the voltage detection circuit generates a Voltage Monitor 4 reset if VCC rises to or above V_{det4}.

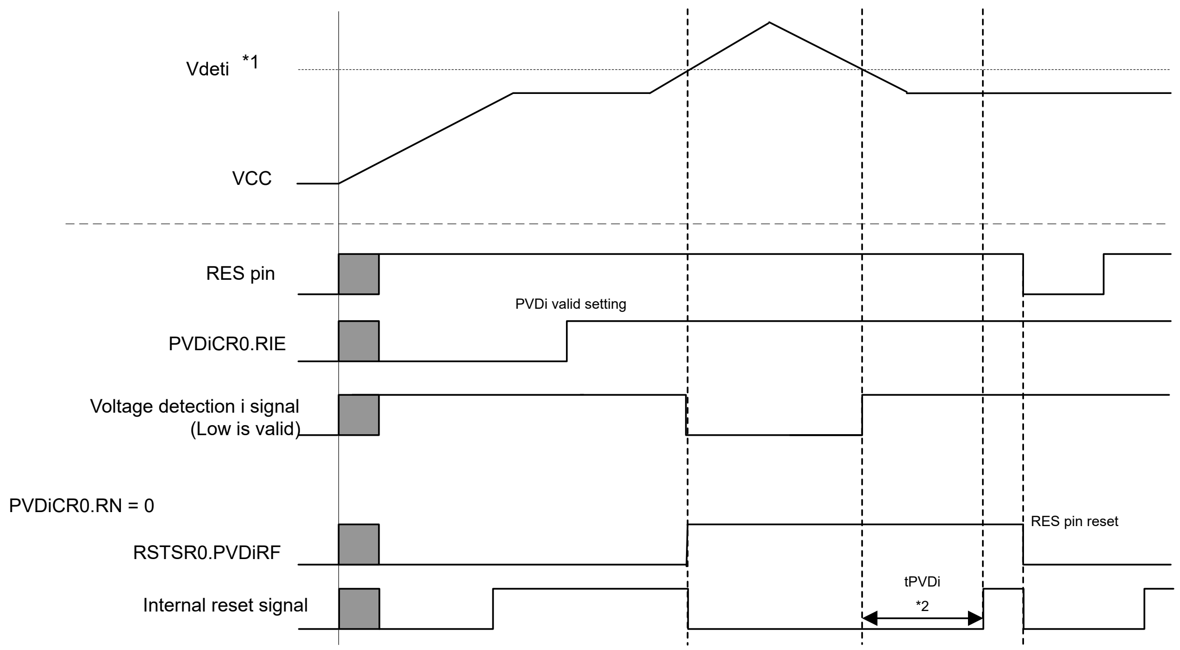
The CPU is then released from the internal reset state and starts reset exception handling when the PVD4 reset time (tPVD4) elapses after VCC falls below V_{det4}.

When the Voltage Monitor 5 Reset Enable bit (RE) is set to 1 (enabling generation of a reset by the voltage detection circuit) in Voltage Monitor 5 Circuit Control Register 0 (PVD5CR0), the RSTSR0.PVD5RF flag is set to 1 and the voltage detection circuit generates a Voltage Monitor 5 reset if VCC rises to or above V_{det5}.

The CPU is then released from the internal reset state and starts reset exception handling when the PVD5 reset time (tPVD5) elapses after VCC falls below V_{det5}.

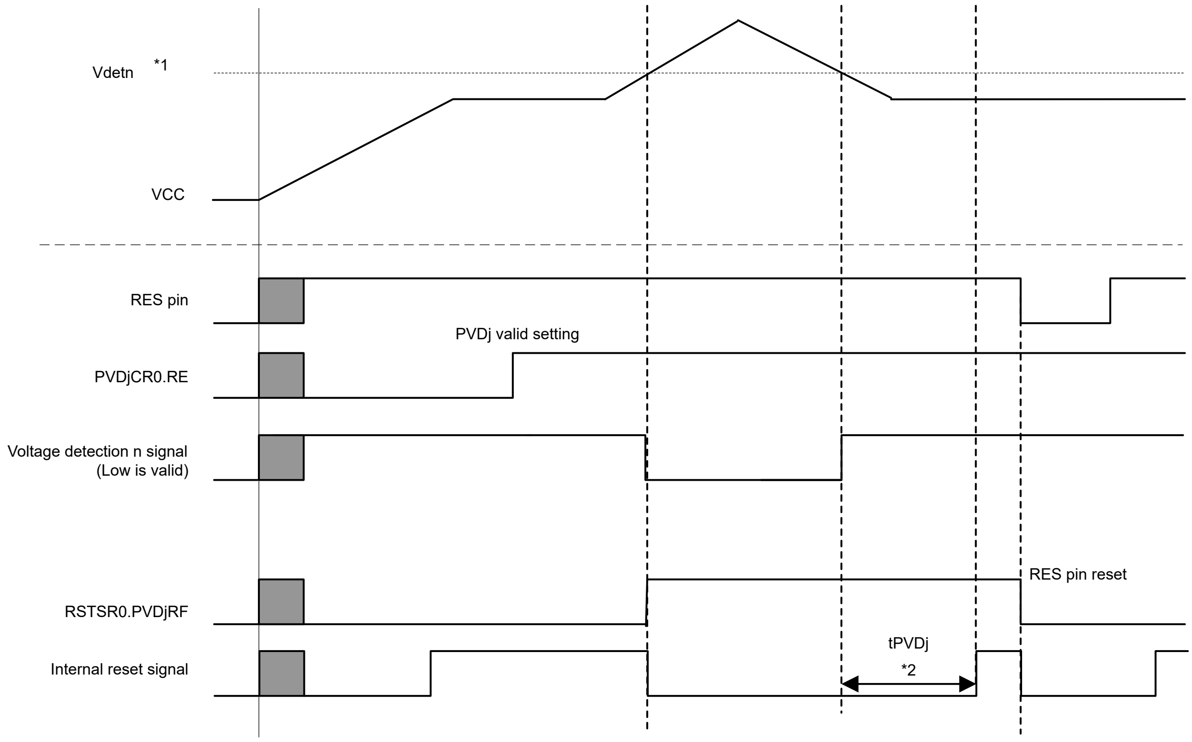
Figure 6.4 shows examples of operations during the voltage monitor 1 and 2 reset.

Figure 6.5 shows examples of operations during the voltage monitor 4 and 5 reset.



Note: For details on the electrical characteristics, see section 62, Electrical Characteristics.
Note 1. V_{deti} shows a detection level of voltage monitor i reset (i = 1 to 2).
Note 2. t_{PVDi} shows a time for voltage monitor i reset (i = 1 to 2).

Figure 6.4 Example of operations during voltage monitor i reset (i = 1, 2)



Note: For details on the electrical characteristics, see section 62, Electrical Characteristics.
Note 1. Vdetj shows a detection level of voltage monitor j reset. (j = 4, 5)
Note 2. tPVDj shows a time for voltage monitor j reset. (j = 4, 5)

Figure 6.5 Example of operations during voltage monitor j reset (j = 4, 5)

The Core Voltage Monitor reset is an internal reset generated by the Core Voltage monitor circuit.

If the Core Voltage Detection Circuit disable bit (CVMRDIS) in Option Function Select Register 0 (OFS2) is 0 (Core Voltage monitor reset is enabled after a reset) and VDD falls below V_{det}_VDDL or rises above V_{det}_VDDH, the RSTSR3.CVMRF flag is set to 1 and the Core Voltage detection circuit generates a Core Voltage monitor reset.

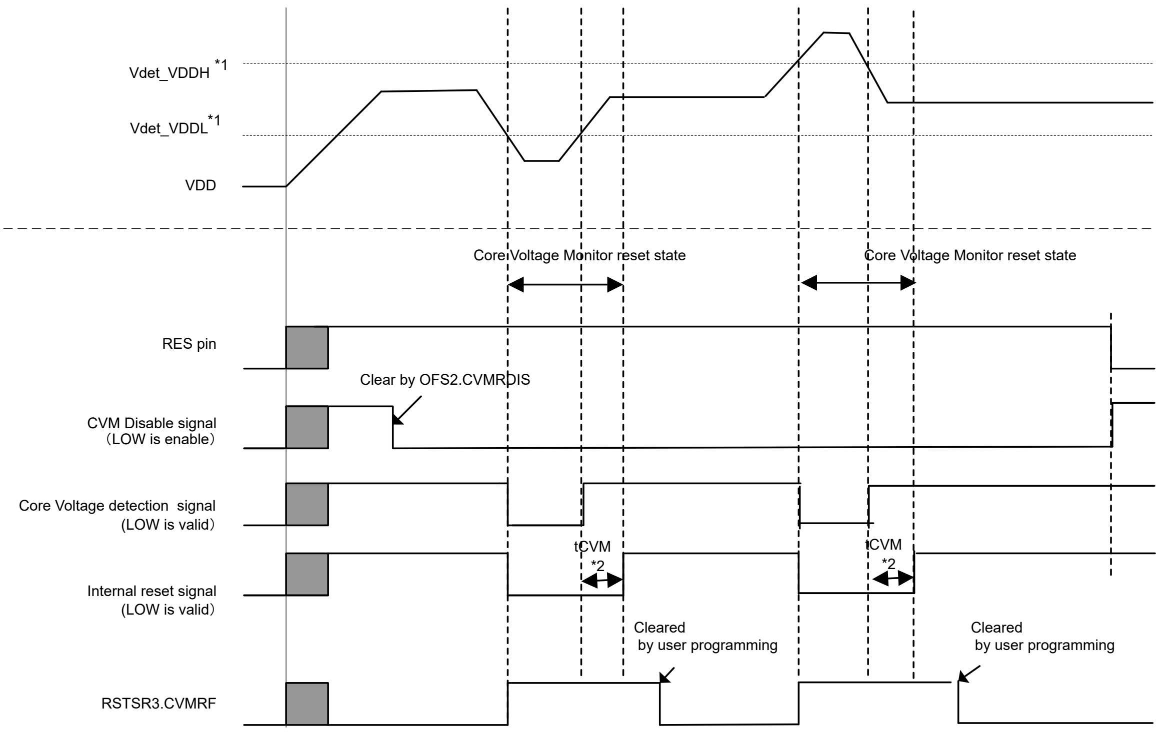
After VDD is between V_{det}_VDDL and V_{det}_VDDH and the Core Voltage monitor reset time (tCVM) elapses, the internal reset is canceled, and the CPU starts the reset exception handling.

The Core Voltage Monitor reset function does not work during DSTBY mode.

When RES pin reset is asserted while issuing CVM reset, the CVM reset is not negated.

When the Core Voltage Monitor reset occurs, the values of register and SRAM cannot be guaranteed due to the response time until the reset occurs.

Figure 6.6 shows examples of operations during the core voltage monitor reset.



Note 1. For details on the electrical characteristics, see section 62, Electrical Characteristics.
Note 2. tCVM shows the reset time of Core Voltage Monitor.

Figure 6.6 Operation Examples during Core Voltage Monitor Reset

6.3.4 Deep Software Standby Reset

These internal resets are generated when either of the Deep Software Standby mode are canceled by an associated interrupt.

The Deep Software Standby reset is canceled after t_{DSBY} (return time after Deep Software Standby mode cancellation) elapses. At the same time, Deep Software Standby mode is also canceled.

When t_{DSBYWT} (wait time after Deep Software Standby mode cancellation) elapses after Deep Software Standby mode has been canceled, the internal reset is canceled, and the CPU starts the reset exception handling.

For details of the Deep Software Standby reset, see section 11, Low Power Mode.

6.3.5 Independent Watchdog Timer Reset

The independent watchdog timer reset is an internal reset generated from the Independent Watchdog Timer (IWDT). Output of the reset from the IWDT can be selected in the Option Function Select Register 0 (OFS0).

When output of the independent watchdog timer reset is selected, the reset is generated if the IWDT underflows, or if data is written when refresh operation is disabled. When the internal reset time (t_{RESW2}) elapses after the independent watchdog timer reset is generated, the internal reset is canceled and the CPU starts the reset exception handling.

For details on the independent watchdog timer reset, see section 28, Independent Watchdog Timer (IWDT).

6.3.6 CPU0/CPU1 Watchdog Timer Reset

The CPU0/CPU1 watchdog timer reset is an internal reset generated from the Watchdog Timer (WDT0/1). Output of the reset from the WDT0/1 can be selected in the WDT Reset Control Register (WDTRCR) or Option Function Select register 0 (OFS0).

When output of the watchdog timer reset is selected, the reset is generated if the WDT underflows, or if data is written when refresh operation is disabled. When the internal reset time (t_{RESW2}) elapses after the watchdog timer reset is generated, the internal reset is canceled and the CPU starts the reset exception handling.

For details on the watchdog timer reset, see section 27, Watchdog Timer (WDT).

6.3.7 CPU0/CPU1 Lockup Reset

The CPU0 lockup reset is an internal reset generated by Arm core. Output of the reset from Arm core can be selected by CPU0LCKUPCR.OAD.

When output of the CPU0 lockup reset is selected, CPU0 generated if Arm core is in the lockup state. When the internal reset time (tRESW2) elapses after the CPU0 lockup reset is generated, the internal reset is cancelled, and the CPU starts the reset exception handling.

Similarly, CPU1 has lockup reset. Output of the reset from Arm core can be selected by CPU1LCKUPCR.OAD.

For details on the CPU lockup reset, see section 2, CPU.

6.3.8 Software Reset

This internal reset is generated by a software setting of the SYSRESETREQ bit in the AIRCR register in the Arm core. When the SYSRESETREQ bit is set to 1, a software reset is generated. When the internal reset time (t_{RESW2}) elapses after the software reset is generated, the internal reset is canceled, and the CPU starts the reset exception handling. For dual CPU systems, SYSRESETREQ of CPU0 will always generate a software reset but CPU1 reset request can be gated off by setting a control register accessible by secure software running on CPU0.

6.3.9 Bus Error Reset

Bus error reset is an internal reset generated by buses.

It is an integrated reset generated by buses, and it consists of the following:

- Bus error reset (MSAU error, MMPU error, Illegal address error, Slave TrustZone Filter error, Slave Bus error, Bufferable write error)

Output of Bus error reset can be selected by MSAOAD.OAD or MMPUOAD.OAD or BUSOAD.ILERROAD or BUSOAD.SLERROAD or BUSOAD.BWERROAD.

When the internal reset time (t_{RESW2}) elapses after the Bus error reset is generated, the internal reset is cancelled, and the CPU starts the reset exception handling.

For details on the Bus error reset, see section 15, Buses. Determination of reset generation source in Bus error reset is similar to determination of interrupt generation source, see section 14, Interrupt Controller Unit (ICU).

6.3.10 Common Memory Error Reset

Common memory error reset is an internal reset generated by SRAMs.

It is an integrated reset generated by SRAMs and consists of the following:

- SRAM error reset (ECC error)

Output of SRAM error reset can be selected by SRAMCRn.OAD.

When the internal reset time (tRESW2) elapses after the Common memory error reset is generated, the internal reset is cancelled, and the CPU starts the reset exception handling.

For details on the Common memory error reset, see section 58, SRAM. Determination of reset generation source in Common memory error reset is similar to determination of interrupt generation source, see section 14, Interrupt Controller Unit (ICU).

6.3.11 Local Memory 0 / Local Memory 1 Error Reset

Local memory 0/ Local memory 1 error reset is an internal reset generated by SRAMs.

It is an integrated reset generated by SRAMs, and it consists of the following:

- Local Memory 0 error reset (ECC error of CPU0 Cache or TCM)
- Local Memory 1 error reset (ECC error of CPU1 Cache or TCM)

Output of Local Memory 1 error reset can be selected by CAPOAD.OAD and TCMCRC.OAD or TCMCRS.OAD.

When the internal reset time (tRESW2) elapses after the Local Memory 0/1 Error Reset is generated, the internal reset is cancelled, and the CPU starts the reset exception handling.

For details on the Local Memory 0 and Local Memory 1 error reset, see section 2, CPU. For determination of reset generation source in Local Memory 0/Local Memory 1 Error Reset, it is similar to determination of interrupt generation source and see section 14, Interrupt Controller Unit (ICU).

6.3.12 Temperature Monitor Reset

Temperature monitor reset is an internal reset generated by temperature sensor (TSN). TSN detects ambient temperature on surrounding environment of MCU. If it is outside from the specified range, reset is issued.

The function of temperature monitor reset is enabled via specific procedure which is described in Figure 6.7.

When TSN is activated and reset function enabled, temperature monitoring is executed. Representative behavior of temperature monitor reset is shown in Figure 6.8 and Figure 6.9.

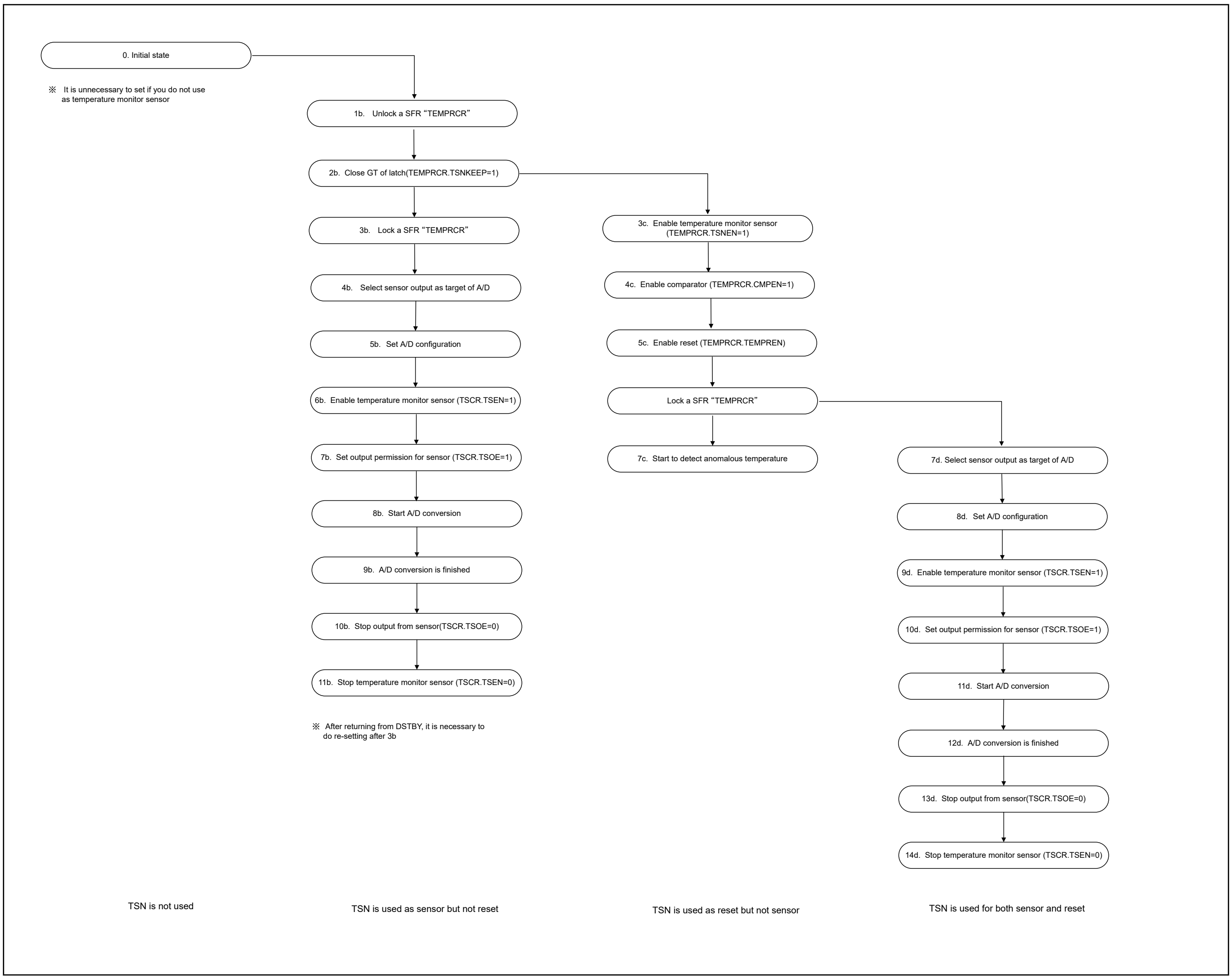


Figure 6.7 Setting procedure for temperature monitor sensor

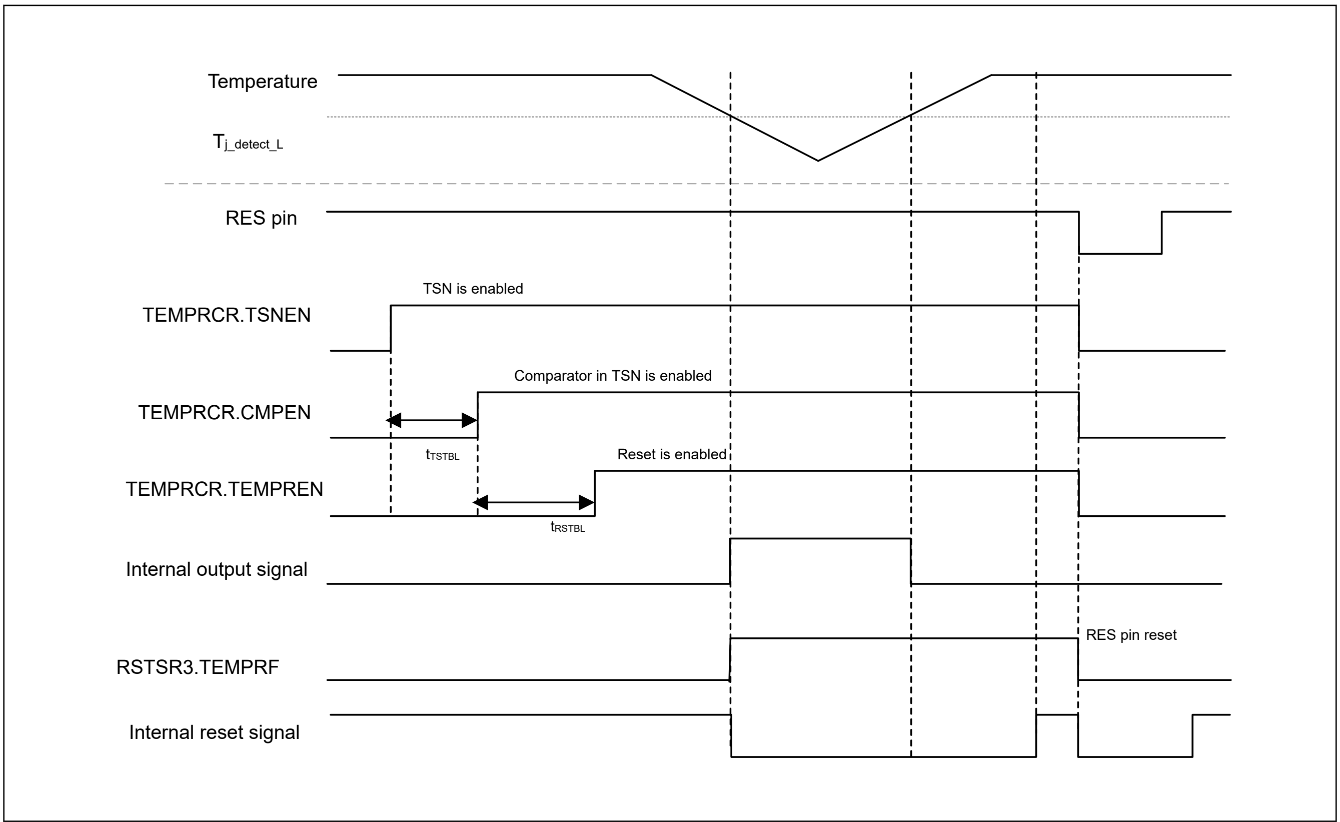


Figure 6.8 Example of temperature monitor reset operation (low detection)

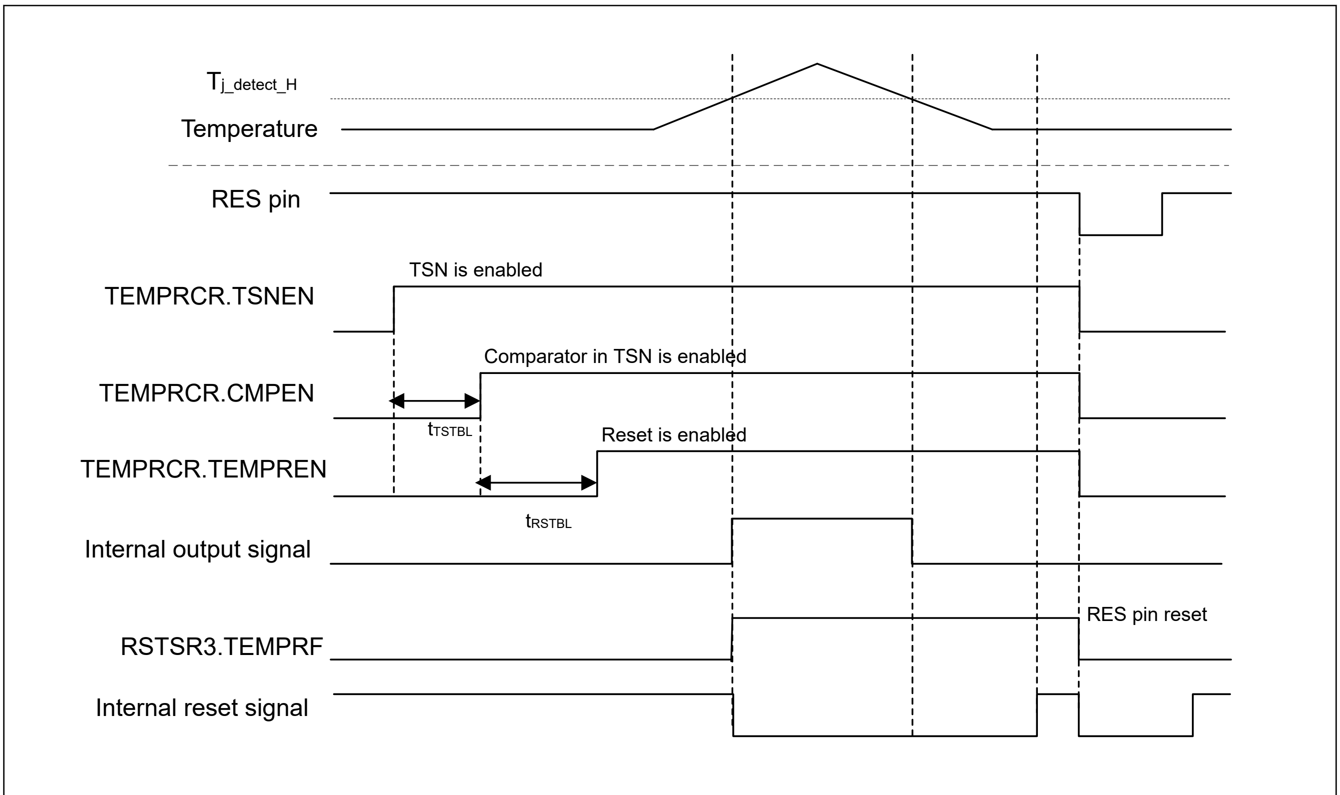


Figure 6.9 Example of temperature monitor reset operation (high detection)

6.3.13 Determination of Cold/Warm Start

Read the CWSF flag in RSTSR2 to determine the cause of reset processing. This flag indicates whether a power-on reset caused the reset processing (cold start) or a reset signal input during operation caused the reset processing (warm start).

The CWSF flag is set to 0 when a power-on reset occurs (cold start), otherwise the flag is not set to 0. The flag is set to 1 when 1 is written to it through software. It is not set to 0 even on writing 0 to it.

Figure 6.10 shows an example of cold/warm start determination operation.

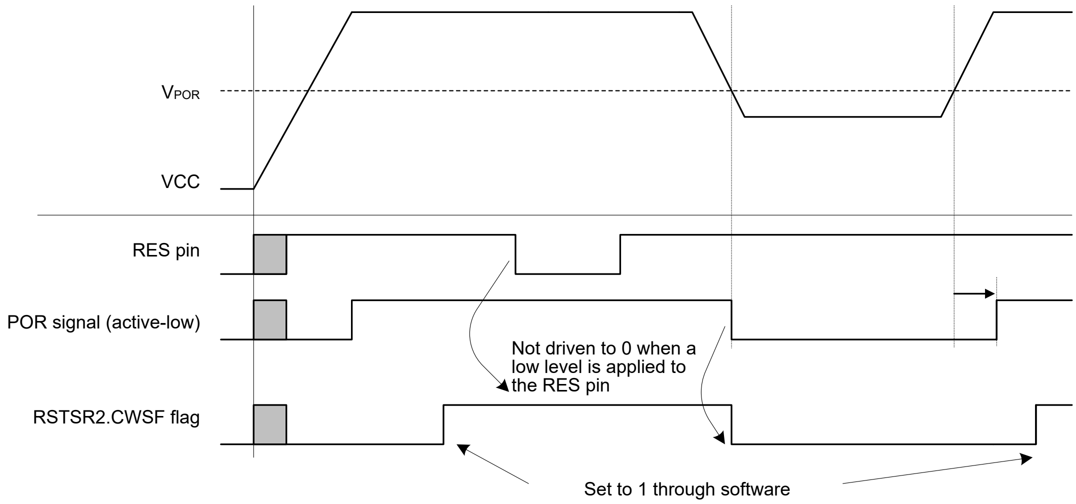
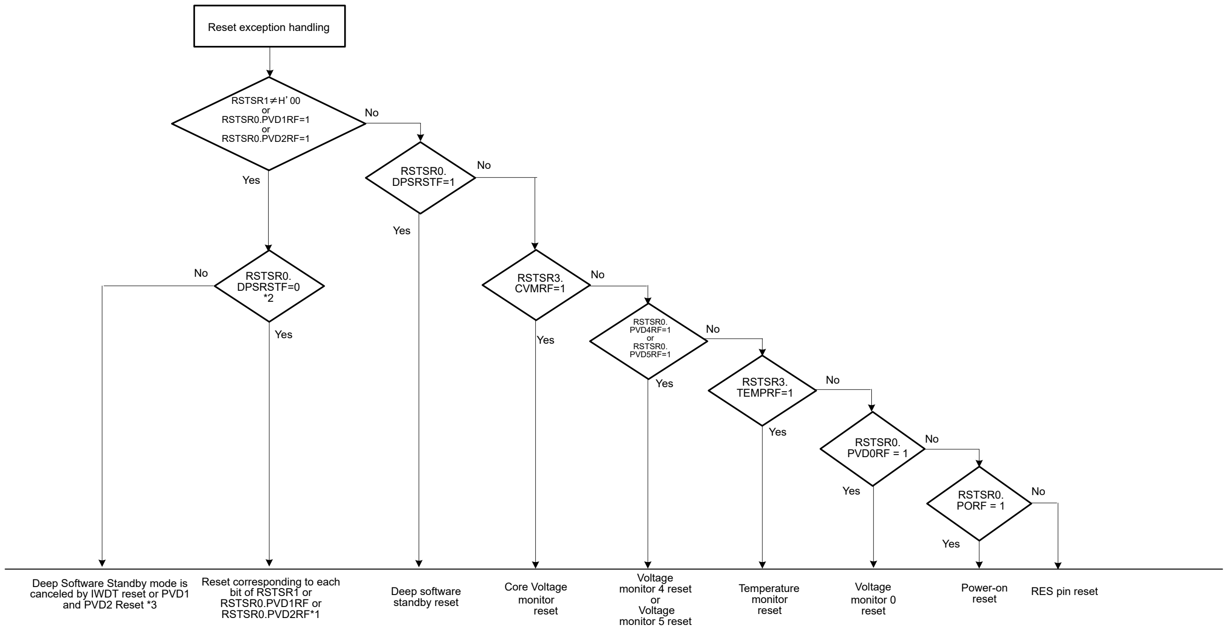


Figure 6.10 Example of cold/warm start determination operation

6.3.14 Determination of Reset Generation Source

Read RSTSR0, RSTSR1, and RSTSR3 to determine which reset executes the reset exception handling.

Figure 6.11 shows an example of the flow to identify a reset generation source. The reset flag must be written with 0 after 1 is read.



Note 1. If a reset associated with each bit of RSTSR1, RSTSR0.PVD1RF, or RSTSR0.PVD2RF occurred at the same time, all occurrences of the reset flags are set to 1.

Note 2. After checking RSTSR1 ≠ 0x00, RSTSR0.PVD1RF = 1 or RSTSR0.PVD2RF = 1, check RSTSR0.DPSRSTF to determine whether it caused the cancellation of Deep Software Standby mode.

Note 3. The Reset Status registers (RSTSR0, RSTSR1, RSTSR2, and RSTSR3) should be cleared before the Deep Software Standby transition. If the reset flag has not been cleared, the return factor cannot be correctly identified.

Figure 6.11 Example of reset generation source determination flow

6.4 Reset Protection from Non-Secure Programs

The reset occurrence can be controlled. A reset is generated according to the setting of SYRSTMSK0, SYRSTMSK1 and SYRSTMSK2 that can only be accessed from a secure program.

The reset source that is set to disable is generated but is not propagated to the MCU. The reset flags of RSTSR0 and RSTSR1 bits are not set.

The resets that can be controlled are as follows:

- Independent watchdog timer reset
- CPU0/CPU1 Watchdog timer reset
- CPU0/CPU1 lockup reset
- Common memory error reset
- Local memory 0 error reset
- Local memory 1 error reset
- Bus error reset
- Voltage monitor 1 reset
- Voltage monitor 2 reset
- Software reset

7. Option-Setting Memory

7.1 Overview

The option-setting memory determines the state of the MCU after a reset. The option-setting memory is allocated to the configuration setting area or the OTP area of the extra MRAM memory.

Figure 7.1 shows the option-setting memory area. The option-setting memory area has Secure region and Non-secure region. Table 7.1 shows the programming condition of the option-setting memory area.

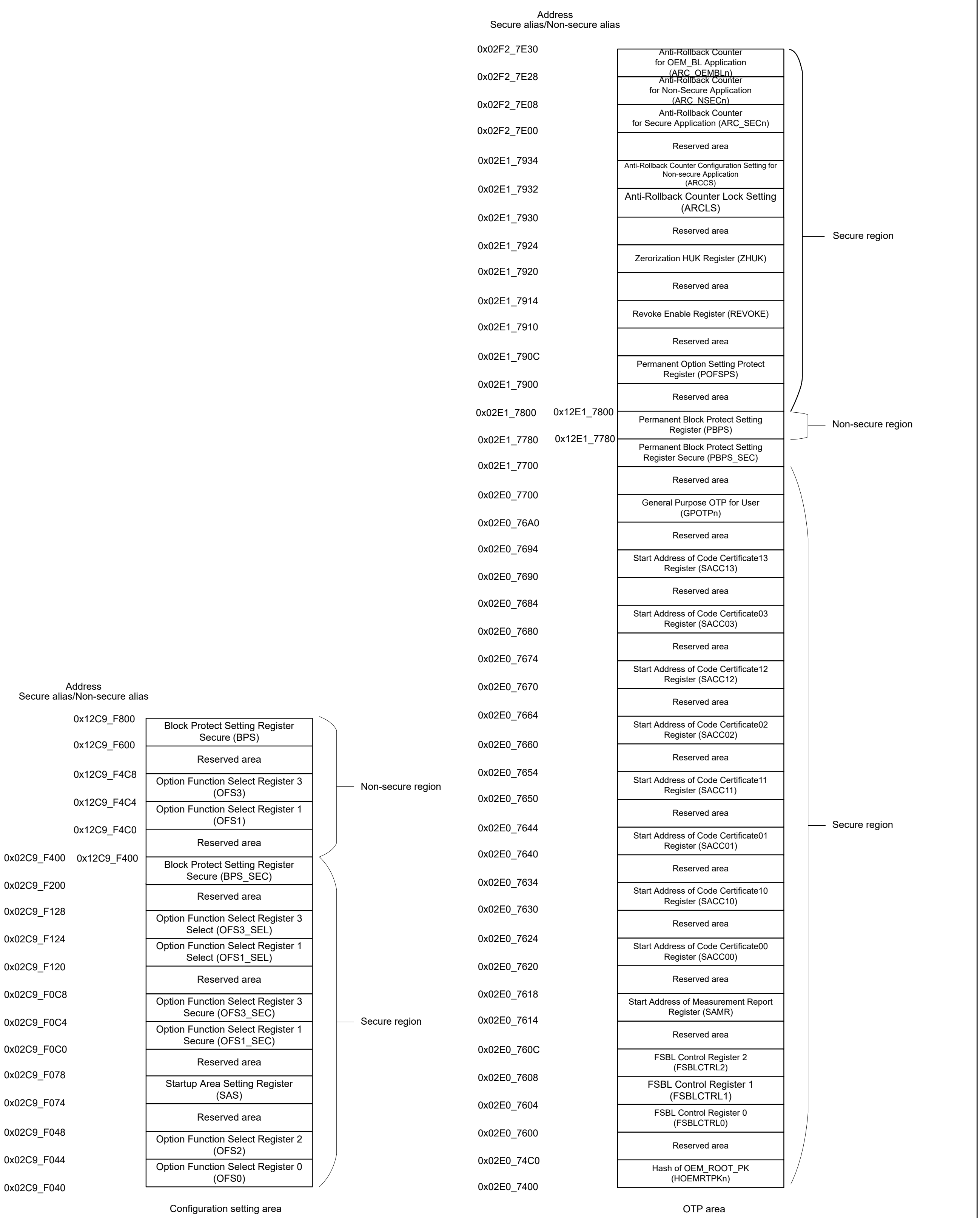


Figure 7.1 Option-setting memory area

Table 7.1 The programming condition of the option-setting memory area

	Self programming	Serial programming	Programming by the on-chip debugger
Secure region	Programming commands issued via Secure alias access.	Programming commands issued when the authentication level is AL2.	Programming commands issued when the authentication level is AL2.
Non-secure region	Programming commands issued via Non-secure alias access.	Programming commands issued when the authentication level is AL2 or AL1.	Programming commands issued when the authentication level is AL2 or AL1.

7.2 Register Descriptions

7.2.1 OFS0 : Option Function Select Register 0

Address: 0x02C9_F040

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	WDT0STPCTL	—	WDT0RSTIRQS	WDT0RPSS[1:0]		WDT0RPES[1:0]		WDT0CKS[3:0]				WDT0TOPS[1:0]		WDT0STRT	—
Value after reset:	User setting[*1]															

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	IWDTSTPCTL	—	IWDTRSTIRQS	IWDTRPSS[1:0]		IWDTRPES[1:0]		IWDTCKS[3:0]				IWDTTOPS[1:0]		IWDTSTRT	—
Value after reset:	User setting[*1]															

Bit	Symbol	Function	R/W
0	—	The program value is read from this bit.[*2]	R
1	IWDTSTRT	IWDT Start Mode Select 0: Automatically activate IWDT after a reset (auto start mode) 1: Stop IWDT after a reset (register start mode)	R
3:2	IWDTTOPS[1:0]	IWDT Timeout Period Select 0 0: 128 cycles (0x007F) 0 1: 512 cycles (0x01FF) 1 0: 1024 cycles (0x03FF) 1 1: 2048 cycles (0x07FF)	R
7:4	IWDTCKS[3:0]	IWDT-Dedicated Clock Frequency Division Ratio Select 0x0: × 1 0x2: × 1/16 0x3: × 1/32 0x4: × 1/64 0xF: × 1/128 0x5: × 1/256 Others: Reserved	R
9:8	IWDTRPES[1:0]	IWDT Window End Position Select 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (no window end position setting)	R
11:10	IWDTRPSS[1:0]	IWDT Window Start Position Select 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (no window start position setting)	R
12	IWDTRSTIRQS	IWDT Reset Interrupt Request Select 0: Interrupt 1: Reset	R
13	—	The program value is read from this bit.[*2]	R

Bit	Symbol	Function	R/W
14	IWDTSTPCTL	IWDT Stop Control 0: Continue counting 1: Stop counting when primary CPU is in CPU Sleep mode or CPU Deep Sleep mode, or MCU is in Software Standby mode or Deep Software Standby mode 1	R
16:15	—	The program value is read from these bits.*2	R
17	WDT0STRT	WDT0 Start Mode Select 0: Automatically activate WDT0 after a reset (auto start mode) 1: Stop WDT0 after a reset (register start mode)	R
19:18	WDT0TOPS[1:0]	WDT0 Timeout Period Select 0 0: 1024 cycles (0x03FF) 0 1: 4096 cycles (0x0FFF) 1 0: 8192 cycles (0x1FFF) 1 1: 16384 cycles (0x3FFF)	R
23:20	WDT0CKS[3:0]	WDT0 Clock Frequency Division Ratio Select 0x1: PCLKB divided by 4 0x4: PCLKB divided by 64 0xF: PCLKB divided by 128 0x6: PCLKB divided by 512 0x7: PCLKB divided by 2048 0x8: PCLKB divided by 8192 Others: Reserved	R
25:24	WDT0RPES[1:0]	WDT0 Window End Position Select 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (no window end position setting)	R
27:26	WDT0RPSS[1:0]	WDT0 Window Start Position Select 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (no window start position setting)	R
28	WDT0RSTIRQS	WDT0 Reset Interrupt Request Select 0: Interrupt 1: Reset	R
29	—	The program value is read from this bit.*2	R
30	WDT0STPCTL	WDT0 Stop Control 0: Continue counting 1: Stop counting when CPU0 is in CPU Sleep mode or CPU Deep Sleep mode	R
31	—	The program value is read from this bit.*2	R

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.
Note 2. This register can only be programmed by MACI command. When programming, the set value should be 1.

IWDTSTRT bit (IWDT Start Mode Select)

The IWDTSTRT bit selects the mode in which the IWDT is activated after a reset (stopped state or activated state). When IWDT is activated in auto start mode, the OFS0 register setting for the IWDT is valid.

IWDTTOPS[1:0] bits (IWDT Timeout Period Select)

The IWDTTOPS[1:0] bits specify the timeout period, that is, the time it takes for the down counter to underflow, as 128, 512, 1024, or 2048 cycles of the frequency-divided clock set in the IWDTCKS[3:0] bits. The time it takes for the counter to underflow after a refresh operation is determined by the combination of the IWDTCKS[3:0] and IWDTTOPS[1:0] bits.

For details, see section 28, Independent Watchdog Timer (IWDT).

IWDTCKS[3:0] bits (IWDT-Dedicated Clock Frequency Division Ratio Select)

The IWDTCKS[3:0] bits specify the division ratio of the prescaler for dividing the frequency of the clock for the IWDT as 1/1, 1/16, 1/32, 1/64, 1/128, and 1/256. Using this setting combined with the IWDTTOPS[1:0] bits setting, the IWDT counting period can be set from 128 to 524,288 IWDT clock cycles.

For details, see section 28, Independent Watchdog Timer (IWDT).

IWDTRPES[1:0] bits (IWDT Window End Position Select)

The IWDTRPES[1:0] bits specify the position where the window for the down counter ends as 0%, 25%, 50%, or 75% of the count value. The value of the window end position must be smaller than the value of the window start position, otherwise only the value for the window start position is valid.

The counter values associated with the settings for the start and end positions of the window in the IWDTRPSS[1:0] and IWDTRPES[1:0] bits vary with the setting in the IWDTTOPS[1:0] bits.

For details, see section 28, Independent Watchdog Timer (IWDT).

IWDTRPSS[1:0] bits (IWDT Window Start Position Select)

The IWDTRPSS[1:0] bits specify the position where the window for the down counter starts as 25%, 50%, 75%, or 100% of the counted value. The point at which counting starts is 100% and the point at which an underflow occurs is 0%. The interval between the window starts and ends positions becomes the period in which a refresh is possible. Refresh is not possible outside this period.

For details, see section 28, Independent Watchdog Timer (IWDT).

IWDTRSTIRQS bit (IWDT Reset Interrupt Request Select)

The IWDTRSTIRQS bit selects the operation on an underflow of the down counter or generation of a refresh error. The operation is selectable to an independent watchdog timer reset, a non-maskable interrupt request, or an interrupt request.

For details, see section 28, Independent Watchdog Timer (IWDT).

IWDTSTPCTL bit (IWDT Stop Control)

The IWDTSTPCTL bit specifies whether to stop counting when entering CPU Sleep mode, CPU Deep Sleep mode, Software Standby mode or Deep Software Standby mode1.

For details, see section 28, Independent Watchdog Timer (IWDT).

WDT0STRT bit (WDT0 Start Mode Select)

The WDT0STRT bit selects the mode in which the WDT is activated after a reset (stopped state or activated in auto start mode). When WDT is activated in auto start mode, the OFS0 register setting for the WDT is valid.

WDT0TOPS[1:0] bits (WDT0 Timeout Period Select)

The WDT0TOPS[1:0] bits specify the timeout period, that is, the time it takes for the down counter to underflow as 1024, 4096, 8192, or 16384 cycles of the frequency-divided clock set in the WDT0CKS[3:0] bits. The number of PCLKB cycles that takes to underflow after a refresh operation is determined by a combination of the WDT0CKS[3:0] and WDT0TOPS[1:0] bits.

For details, see section 27, Watchdog Timer (WDT).

WDT0CKS[3:0] bits (WDT0 Clock Frequency Division Ratio Select)

The WDT0CKS[3:0] bits specify the division ratio of the prescaler for dividing the frequency of PCLKB as 1/4, 1/64, 1/128, 1/512, 1/2048, and 1/8192. Using this setting combined with the WDT0TOPS[1:0] bits setting, the WDT counting period can be set from 4,096 to 134,217,728 PCLKB cycles.

For details, see section 27, Watchdog Timer (WDT).

WDT0RPES[1:0] bits (WDT0 Window End Position Select)

The WDT0RPES[1:0] bits specify the position where the window on the down counter ends as 0%, 25%, 50%, or 75% of the counted value. The value of the window end position must be smaller than the value of the window start position, otherwise only the value for the window start position is valid.

The counter values associated with the settings for the start and end positions of the window in the WDT0RPSS[1:0] and WDT0RPES[1:0] bits vary with the setting of the WDT0TOPS[1:0] bits.

For details, see section 27, Watchdog Timer (WDT).

WDT0RPSS[1:0] bits (WDT0 Window Start Position Select)

The WDT0RPSS[1:0] bits specify the position where the window for the down counter starts as 25%, 50%, 75%, or 100% of the counted value. The point at which counting starts is 100% and the point at which an underflow occurs is 0%. The interval between the positions where the window starts and ends becomes the period in which a refresh is possible.

Refresh is not possible outside this period.

For details, see section 27, Watchdog Timer (WDT).

WDT0RSTIRQS bit (WDT0 Reset Interrupt Request Select)

The WDT0RSTIRQS bit selects the operation on an underflow of the down-counter or generation of a refresh error. The operation is selectable to a watchdog timer reset, a non-maskable interrupt request, or an interrupt request.

For details, see section 27, Watchdog Timer (WDT).

WDT0STPCTL bit (WDT0 Stop Control)

The WDT0STPCTL bit specifies whether to stop counting when CPU0 is in CPU Sleep mode or CPU Deep Sleep mode.

For details, see section 27, Watchdog Timer (WDT).

7.2.2 OFS2 : Option Function Select Register 2

Address: 0x02C9_F044

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	EXVDDSEL[1:0]	EXVDDSEL[1:0]	—	—	CVMRDIS	DCDCEN
Value after reset:	User setting[*1]															

Bit	Symbol	Function	R/W
0	DCDCEN	DCDC enable 0: Disable DCDC 1: Enable DCDC	R
1	CVMRDIS	CVM reset disable 0: Enable CVM reset 1: Disable CVM reset	R
3:2	—	The program value is read from these bits.[*2]	R
5:4	EXVDDSEL[1:0]	External VDD voltage select for SRAM 1 0: 0.9V ± 10% Others: Setting prohibit	R
31:6	—	The program value is read from these bits.[*2]	R

Note 1. The value in a blank product is 0xFFFFFFEF. It is set to the value written by your application.
Note 2. This register can only be programmed by MACI command. When programming, the set value should be 1.

DCDCEN bit (DCDC enable)

The DCDCEN bit selects whether the DCDC is enabled or disabled.

CVMRDIS bit (CVM reset disable)

The CVMRDIS bit selects whether the CVM reset is enabled or disabled.

EXVDDSEL[1:0] bits (External VDD voltage select for SRAM)

The EXVDDSEL[1:0] bit selects bits selects the voltage for SRAM in external VDD mode.

7.2.3 SAS : Startup Area Setting Register

Address: 0x02C9_F074

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

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

Bit	Symbol	Function	R/W
14:0	—	The program value is read from these bits.*2	R
28:15	—	The program value is read from these bits.*2	R
30:29	BTSIZE[1:0]	Size of Start-Up Area select 0 0: Reserved 0 1 Start-Up area Size is set to 32KB 1 0 Start-Up area Size is set to 16KB 1 1: Start-Up area Size is set to 8KB	R
31	BTFLG	Startup Area Select Flag This bit specifies whether the address of the startup area is exchanged for the boot swap function or not. Example of 8KB startup area 0: First 8 KB area (0x0200_0000 to 0x0200_1FFF) and second 8 KB area (0x0200_2000 to 0x0200_3FFF) are exchanged. 1: First 8 KB area (0x0200_0000 to 0x0200_1FFF) and second 8 KB area (0x0200_2000 to 0x0200_3FFF) are not exchanged.	R

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application

Note 2. This register can only be programmed by MACI command. When programming, the set value should be 1.

7.2.4 OFS1, OFS1_SEC : Option Function Select Register 1 for Non-Secure and Secure

Address: OFS1: 0x12C9_F4C0
OFS1_SEC: 0x02C9_F0C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	INITECCEN	SWDBG	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	HOCOFRQ0[2:0]			HOCOEN	—	—	PVDLPSEL	—	PVDAS	VDSEL[2:0]		
Value after reset:	The value set by the user*1															

Bit	Symbol	Function	R/W
2:0	VDSEL[2:0]	Voltage Detection 0 Level Select 0 0 0: Select 2.85 V 0 0 1: Select 2.58 V 0 1 0: Select 2.15 V 0 1 1: Select 2.00 V 1 0 0: Select 1.90 V 1 0 1: Select 1.80 V 1 1 0: Select 1.67 V 1 1 1: Select 1.56 V*3	R

Bit	Symbol	Function	R/W
3	PVDAS	Voltage Detection 0 Circuit Start 0: Enable voltage monitor 0 reset after a reset 1: Disable voltage monitor 0 reset after a reset	R
4	—	The program value is read from this bit.*2	R
5	PVDLPSEL	Low Power Consumption Function of PVD0 Select at Deep Software standby mode 0: Enable low power consumption function by selecting PVD0 for the battery power supply switch control during DSTBY1 and DSTBY2. 1: Disable low power consumption function of PVD0 for the battery power supply switch control during DSTBY1 and DSTBY2.	R
7:6	—	The program value is read from these bits.*2	R
8	HOCOEN	HOCO Oscillation Enable 0: Enable HOCO oscillation after a reset 1: Disable HOCO oscillation after a reset	R
11:9	HOCOFRQ0[2:0]	HOCO Frequency Setting 0 0 0 0: 16 MHz 0 0 1: 18 MHz 0 1 0: 20 MHz 1 0 0: 32 MHz 1 1 1: 48 MHz Others: Reserved.	R
23:12	—	The program value is read from these bits.*2	R
24	SWDBG	Software Debug Control 0: Enable software debug control. The MCU behaves according to DBGCTRL0 and DBGCTRL1, and IWDT and WDT automatically stops when CPU is in the debug state. 1: Disable software debug control. The MCU ignores DBGCTRL0 and DBGCTRL1, and IWDT and WDT continue the operation even when CPU is in the debug state. However, when CDBGPWRUPREQ is 1 and AL is not AL0, IWDT and WDT automatically stops when CPU is in the debug state.	R
25	INITECCEN	Initial ECC Enable 0: Disable ECC function of TCM and CACHE of CPU0. 1: Enable ECC function of TCM and CACHE of CPU0.	R
31:26	—	The program value is read from these bits.*2	R

Note 1. The value of OFS1 and OFS1_SEC in a blank product is 0xFFFFFFFF. It is set to the value written by your application.
Note 2. This register can only be programmed by MACI command. When programming, the set value should be 1.
Note 3. Setting prohibited when VBATT function is enabled, OFS1(_SEC).PVDAS and PVDLPSEL bits are 0 and at least one of Deep Software Standby modes 1 and 2 is used.

OFS1 register is for Non-secure developer, and OFS1_SEC register is for Secure developer.

VDSEL[2:0] bits (Voltage Detection 0 Level Select)

The VDSEL[2:0] bits select the voltage detection level of the voltage detection 0 circuit.

PVDAS bits (Voltage Detection 0 Circuit Start)

The PVDAS bit selects whether the voltage monitor 0 reset is enabled or disabled after a reset.

PVDLPSEL bits (Low Power Consumption Function of PVD0 Select at Deep Software standby mode)

The PVDLPSEL selects whether the low power consumption function of PVD0 for the battery power supply switch control is enabled or disabled during DSTBY1 and DSTBY2 mode.

When disable this function, the battery power supply switch is controlled by VCC voltage drop detection circuit dedicated for VBATT, the response time is faster instead of increasing current consumption. When enable this function, the battery power supply switch is controlled by PVD0, function, the current consumption is smaller instead of response time delay.

See section 62, Electrical Characteristics

HOCOEN bit (HOCO Oscillation Enable)

The HOCOEN bit selects whether the HOCO oscillation is enabled or disabled after a reset. Setting this bit to 0 allows the HOCO oscillation to start before the CPU starts operation, which reduces the wait time for oscillation stabilization.

Note: When the HOCOEN bit is set to 0, the system clock source is not switched to HOCO. The system clock source is only switched to HOCO by setting the Clock Source Select bits (SCKSCR.CKSEL[2:0]). To use the HOCO clock, Set the OFS1(_SEC).HOCOFRQ0[2:0] bits to an optimum value.

Note: OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1.HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits when OFS1(_SEC).HOCOEN = 1.

HOCOFRQ0[2:0] bits (HOCO Frequency Setting 0)

The HOCOFRQ0[2:0] bits specify the HOCO frequency after a reset as 16 to 48 MHz.

SWDBG bit (Software Debug Control)

The SWDBG bit selects whether software debug control is enabled or disabled.

INITECCEN bit (Initial ECC Enable)

The INITECCEN bit selects whether ECC function of TCM and CACHE of CPU0 is enabled or disabled.

When the INITECCEN bit is changed from 1 to 0, be sure to perform a power-on reset after the change.

7.2.5 OFS1_SEL : Option Function Select Register 1 for Security Attribution

Address: OFS1_SEL: 0x02C9_F120

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	INITECCEN	SWDBG	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	HOCOFRQ0[2:0]			HOCOEN	—	—	PVDLPSEL	—	PVDAS	VDSEL[2:0]		
Value after reset:	The value set by the user*1															

Bit	Symbol	Function	R/W
2:0	VDSEL[2:0]	Security attributes of Voltage Detection 0 Level Select 0 0 0: Select OFS1_SEC.VDSEL0[2:0] 1 1 1: Select OFS1.VDSEL0[2:0] Others: Select Reserved.	R
3	PVDAS	Security attributes of Voltage Detection 0 Circuit Start 0: Select OFS1_SEC.PVDAS 1: Select OFS1.PVDAS	R
4	—	The program value is read from this bit.*2	R
5	PVDLPSEL	Security attributes of Low Power Consumption Function of PVD0 Select at Deep Software standby mode 0: Select OFS1_SEC.PVDLPSEL 1: Select OFS1.PVDLPSEL.	R
7:6	—	The program value is read from these bits.*2	R
8	HOCOEN	Security attributes of HOCO Oscillation Enable 0: Select OFS1_SEC.HOCOEN 1: Select OFS1.HOCOEN	R

Bit	Symbol	Function	R/W
11:9	HOCOFRQ0[2:0]	Security attributes of HOCO Frequency Setting 0 0 0 0: Select OFS1_SEC.HOCOFRQ0[2:0] 1 1 1: Select OFS1.HOCOFRQ0[2:0] Others: Reserved	R
23:12	—	The program value is read from these bits.*2	R
24	SWDBG	Security attributes of Software Debug Control 0: Select OFS1_SEC.SWDBG 1: Select OFS1.SWDBG	R
25	INITECCEN	Security attributes of Initial ECC Enable 0: Select OFS1_SEC.INITECCEN 1: Select OFS1.INITECCEN	R
31:26	—	The program value is read from these bits.*2	R

Note 1. The value of OFS1_SEL in a blank product is 0x00000000. It is set to the value written by your application

Note 2. This register can only be programmed by MACI command. When programming, the set value should be 0.

The OFS1_SEL register selects the security attribute. Which security attribute is applied, OFS or OFS_SEC, is determined by the setting value of the corresponding bit in the OFS1_SEL register. For details, see section 7.3.3. Security Attribution of Extra MRAM Option-Setting Memory

7.2.6 OFS3, OFS3_SEC : Option Function Select Register 3

Address: OFS3: 0x12C9_F4C4
OFS3_SEC: 02C9_F0C4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	WDT1STPCTL	—	WDT1RSTIRQS	WDT1RPSS[1:0]		WDT1RPES[1:0]		WDT1CKS[3:0]				WDT1TOPS[1:0]		WDT1STRT	—
Value after reset:	User setting*1															

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

Bit	Symbol	Function	R/W
16:0	—	The program value is read from these bits. *2	R
17	WDT1STRT	WDT1 Start Mode Select 0: Automatically activate WDT after a reset (auto start mode) 1: Stop WDT after a reset (register start mode).	R
19:18	WDT1TOPS[1:0]	WDT1 Timeout Period Select 0 0: 1024 cycles (0x03FF) 0 1: 4096 cycles (0x0FFF) 1 0: 8192 cycles (0x1FFF) 1 1: 16384 cycles (0x3FFF)	R
23:20	WDT1CKS[3:0]	WDT1 Clock Frequency Division Ratio Select 0 0 0 1: PCLKB divided by 4 0 1 0 0: PCLKB divided by 64 1 1 1 1: PCLKB divided by 128 0 1 1 0: PCLKB divided by 512 0 1 1 1: PCLKB divided by 2048 1 0 0 0: PCLKB divided by 8192 Other settings are prohibited.	R

Bit	Symbol	Function	R/W
25:24	WDT1RPES[1:0]	WDT1 Window End Position Select 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (No window end position setting)	R
27:26	WDT1RPSS[1:0]	WDT1 Window Start Position Select 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (No window start position setting)	R
28	WDT1RSTIRQS	WDT1 Reset Interrupt Request Select WDT Behavior Select 0: NMI 1: Reset	R
29	—	The program value is read from these bits. *2	R
30	WDT1STPCTL	WDT1 Stop Control 0: Continue counting 1: Stop counting when CPU1 is in CPU Sleep mode or CPU Deep Sleep mode	R
31	—	The program value is read from these bits. *2	R

Note 1. The value of OFS3 and OFS3_SEC in a blank product is 0xFFFFFFFF. The value of OFS3_SEL in factory shipment product is 0x00000000. It is set to the value written by your application.

Note 2. This register can only be programmed by MACI command. When programming, the set value should be 0.

OFS3 register is for Non-secure developer, and OFS3_SEC register is for Secure developer. OFS3 register is for non-secure developer.

WDT1STRT bit (WDT1 Start Mode Select)

The WDT1STRT bit selects the mode in which the WDT is activated after a reset (stopped state or activated state). When the WDT is activated in auto start mode, the OFS0 register setting for the WDT is valid.

WDT1TOPS[1:0] bits (WDT1 Timeout Period Select)

The WDT1TOPS[1:0] bits select the timeout period, the time it takes for the down counter to underflow, as 1024, 4096, 8192, or 16384 cycles of the frequency-divided clock set in the WDT1CKS[3:0] bits. The number of PCLKB cycles that the counter takes to underflow after a refresh operation is determined by the combination of the WDT1CKS[3:0] and WDT1TOPS[1:0] bits.

For details, see section 27, Watchdog Timer (WDT).

WDT1CKS[3:0] bits (WDT1 Clock Frequency Division Ratio Select)

The WDT1CKS[3:0] bits select the division ratio of the prescaler to divide the PCLKB frequency as 1/4, 1/64, 1/128, 1/512, 1/2048, or 1/8192. Using this setting combined with the WDT1TOPS[1:0] bit setting, the WDT counting period can be set from 4,096 to 134,217,728 PCLKB cycles.

For details, see section 27, Watchdog Timer (WDT).

WDT1RPES[1:0] bits (WDT1 Window End Position Select)

The WDT1RPES[1:0] bits select the position where the window for the down counter ends as 0%, 25%, 50%, or 75% of the counted value. The value of the window end position must be smaller than the value of the window start position. Otherwise, only the value for the window start position is valid.

The counter values associated with the settings for the start and end positions of the window, in the WDT1RPSS[1:0] and WDT1RPES[1:0] bits, vary with the setting of the WDT1TOPS[1:0] bits.

For details, see section 27, Watchdog Timer (WDT).

WDT1RPSS[1:0] bits (WDT1 Window Start Position Select)

The WDT1RPSS[1:0] bits select the position where the window for the down counter starts as 25%, 50%, 75%, or 100% of the counted value. The point at which counting starts is 100% and the point at which an underflow occurs is 0%.

The interval between the positions where the window starts and ends becomes the period in which a refresh is possible. However, refresh is not possible outside this period.

For details, see section 27, Watchdog Timer (WDT).

WDT1RSTIRQS bit (WDT1 Reset Interrupt Request Select)

The WDT1RSTIRQS bit selects the operation on an underflow of the down counter or generation of a refresh error. The selected operation can be a watchdog timer reset, non-maskable interrupt request, or interrupt request.

For details, see section 27, Watchdog Timer (WDT).

WDT1STPCTL bit (WDT1 Stop Control)

The WDT1STPCTL bit specifies whether to stop counting when CPU1 is in CPU Sleep mode or CPU Deep Sleep mode. For details, see section 27, Watchdog Timer (WDT).

7.2.7 OFS3_SEL : Option Function Select Register 3 for Security

Address: OFS3_SEL: 02C9 F124h

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	WDT1STPCTL	—	WDT1RSTIRQS	WDT1RPSS[1:0]		WDT1RPES[1:0]		WDT1CKS[3:0]				WDT1TOPS[1:0]		WDT1STRT	—
Value after reset:	User setting*1															

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

Bit	Symbol	Function	R/W
16:0	—	The program value is read from these bits. *2	R
17	WDT1STRT	Security attributes of WDT1 Start Mode Select 0: Select OFS3_SEC.WDT1STRT 1: Select OFS3.WDT1STRT.	R
19:18	WDT1TOPS[1:0]	Security attributes of WDT1 Timeout Period Select 0 0: Select OFS3_SEC.WDT1TOPS[1:0] 1 1: Select OFS3.WDT1TOPS[1:0] Others: Setting prohibit	R
23:20	WDT1CKS[3:0]	Security attributes of WDT1 Clock Frequency Division Ration Select 0 0 0 0: Select OFS3_SEC.WDT1CKS[3:0] 1 1 1 1: Select OFS3.WDT1CKS[3:0 Others: Setting prohibit	R
25:24	WDT1RPES[1:0]	Security attributes of WDT1 Window End Position Select 0 0: Select OFS3_SEC.WDT1RPES[1:0] 1 1: Select OFS3.WDT1RPES[1:0] Others: Setting prohibit	R
27:26	WDT1RPSS[1:0]	Security attributes of WDT1 Window Start Position Select 0 0: Select OFS3_SEC.WDT1RPSS[1:0] 1 1: Select OFS3.WDT1RPSS[1:0] Others: Setting prohibit	R
28	WDT1RSTIRQS	Security attributes of WDT1 Reset Interrupt Request Select 0: Select OFS3_SEC.WDT1RSTIRQS 1: Select OFS3.WDT1RSTIRQS	R
29	—	The program value is read from these bits. *2	R

Bit	Symbol	Function	R/W
30	WDT1STPCTL	Security attributes of WDT1 Stop Control 0: Select OFS3_SEC.WDT1STPCTL 1: Select OFS3.WDT1STPCTL	R
31	—	The program value is read from these bits. *2	R

Note 1. The value of OFS3 and OFS3_SEC in a blank product is 0xFFFFFFFF. The value of OFS3_SEL in factory shipment product is 0x00000000. It is set to the value written by your application.

Note 2. This register can only be programmed by MACI command. When programming, the set value should be 0.

The OFS3_SEL register selects the security attribute. Which security attribute is applied, OFS or OFS_SEC, is determined by the setting value of the corresponding bit in the OFS3_SEL register. For details, see section 59, MRAM

7.2.8 BPS, BPS_SEC : Block Protect Setting Register

Address:

BPS: 0x12C9_F600 + 0x04 × n (n = 0 to 127) (Non-secure)
BPS_SEC: 0x02C9_F200 + 0x04 × n (n = 0 to 127) (Secure)

Bit position:	31 … 0
Bit field:	
Value after reset:	User setting*1

Note 1. The value of BPS and BPS_SEC in a blank product is 0xFFFFFFFF. It is set to the value written by your application.

BPS_SEC register is for Secure developer, and BPS register is for Non-secure developer.

The BPS and BPS_SEC registers invalidate the writing to the code MRAM memory. When the bit of this register is set to 0, the writing to the corresponding block are invalid. For details, see section 59, MRAM.

7.2.9 PBPS, PBPS_SEC : Permanent Block Protect Setting Register

Address: PBPS: 0x12E1_7780 + 0x04 × n (n = 0 to 31) (Non-Secure)
PBPS_SEC: 0x02E1_7700 + 0x04 × n (n = 0 to 31) (Secure)

Bit position:	31 … 0
Bit field:	
Value after reset:	User setting*1

Note 1. The value of PBPS and PBPS_SEC in a blank product is 0xFFFFFFFF. It is set to the value written by your application.

PBPS_SEC register is for secure developer, and PBPS register is for non-secure developer. The PBPS and PBPS_SEC registers invalidate writes to bits of BPS and BPS_SEC. The bit of this register can be programmed 0 when corresponding bit of BPS and BPS_SEC is programmed to 0. When the bit of this register is programmed to 0, writing the corresponding bit of BPS and BPS_SEC register is invalid. Once the bit of this register is set to 0, it is impossible to change the bit to 1.

For details, see section 59, MRAM.

7.2.10 Permanent Option Function Select Protect Setting Register (POFSPS)

Address: 0x02E1_7900

Bit position:	31 … 0
Bit field:	POFSPS[31:0]
Value after reset:	User setting*1

Address: 0x02E1_7904

Bit position:	63–32
Bit field:	POFSPS[63:32]
Value after reset:	User setting*1

Address: 0x02E1_7908

Bit position:	95–80	79–64
Bit field:	Reserved bits	POFSPS[79:64]
Value after reset:	User setting*1	

Note 1. The value of POFSPS in a blank product is 0xFFFFFFFF. It is set to the value written by your application.

The POFSPS register invalidate the programming and erasure to the option function select area. When the bit of this register is set to 0, the programming and erasure to the corresponding area is invalid. For details, see section 59, MRAM.

7.2.11 FSBLCTRL0 : FSBL Control Register 0

Address: 0x02E0_7600

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	FSBLCLK[2:0]			FSBLSKIPDS[2:0]			FSBLSKIPSW[2:0]			FSBLEN[2:0]		
Value after reset:	User setting*1															

Bit	Symbol	Function	R/W
2:0	FSBLEN[2:0]	FSBL Enable 0 0 0: Enable FSBL 1 1 1: Disable FSBL Others: Reserved	R
5:3	FSBLSKIPSW[2:0]	FSBL Skip Enable for Software Reset 0 0 0: Enable skip 1 1 1: Disable skip Others: Reserved	R
8:6	FSBLSKIPDS[2:0]	FSBL Skip Enable for Deep Software Standby Reset 0 0 0: Enable skip 1 1 1: Disable skip Others: Reserved	R
11:9	FSBLCLK[2:0]	Clock Frequency Selection during FSBL Execution See Table 7.2.	R
31:12	—	The program value is read from these bits.*2	R

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.
Note 2. This register can only be programmed by MACI command. When programming, the set value should be 1.

Table 7.2 The relationship between FSBLCLK setting and system clock frequency (1 of 2)

System clock	FSBLCLK[2:0] = 111b (VDD = VSCR_1 or voltage range 1)	FSBLCLK[2:0] = 110b (VDD = VSCR_1 or voltage range 1)	FSBLCLK[2:0] = 101b (VDD = VSCR_2 or voltage range 2)	FSBLCLK[2:0] = 100b (VDD = VSCR_2 or voltage range 2)
CPUCLK0	250 MHz	200 MHz	150 MHz	133 MHz

Table 7.2 The relationship between FSBLCLK setting and system clock frequency (2 of 2)

System clock	FSBLCLK[2:0] = 111b (VDD = VSCR_1 or voltage range 1)	FSBLCLK[2:0] = 110b (VDD = VSCR_1 or voltage range 1)	FSBLCLK[2:0] = 101b (VDD = VSCR_2 or voltage range 2)	FSBLCLK[2:0] = 100b (VDD = VSCR_2 or voltage range 2)
CPUCLK1	250 MHz	200 MHz	150 MHz	133 MHz
ICLK	250 MHz	200 MHz	150 MHz	133 MHz
MRICLK	250 MHz	200 MHz	150 MHz	133 MHz
MRPCLK	125 MHz	100 MHz	75 MHz	66 MHz
PCLKA	125 MHz	100 MHz	75 MHz	66 MHz
PCLKB	62.5 MHz	50 MHz	37.5 MHz	33 MHz

Note: Other FSBLCLK settings are prohibited.

7.2.12 FSBLCTRL1 : FSBL Control Register 1

Address: 0x02E0_7604

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

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

Bit	Symbol	Function	R/W
1:0	FSBLEXMD[1:0]	FSBL Execution Mode 0 0: CRC boot and not report measurement 0 1: CRC boot and report measurement 1 0: Secure boot and not report measurement 1 1: Secure boot and report measurement	R
31:2	—	The program value is read from these bits.*2	R

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.
Note 2. This register can only be programmed by MACI command. When programming, the set value should be 1.

7.2.13 FSBLCTRL2 : FSBL Control Register 2

Address: 0x02E0_7608

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

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

Bit	Symbol	Function	R/W
3:0	PORTPN[3:0]	FSBL Error Notification Port Pin Number*2 0 0 0 0: PORTn00 0 0 0 1: PORTn01 ⋮ 1 1 1 1: PORTn15 (n = 0 to 9, A, B)	R
8:4	PORTGN[4:0]	FSBL Error Notification Port Group Name*2 0 0 0 0 0: PORT0m 0 0 0 0 1: PORT1m ⋮ 0 1 0 1 1: PORTBm Others: Reserved (m = 0 to 15)	R
31:9	—	The program value is read from these bits.*3	R

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.
Note 2. When non-existent pin is selected, FSBL error notification is not output.
Note 3. This register can only be programmed by MACI command. When programming, the set value should be 1.

FSBLCTRL2 register selects the port to output when an Error occurs during FSBL execution. The PORTGN bits select a group of ports, and the PORTPN bits select the pins of that group. It thereby determines the unique port.

7.2.14 SACC0n : Start Address of Code Certificate 0n (n = 0 to 3)

Address: SACC0n: 0x02E0_7620 + 0x20 × n (n = 0 to 3)

Bit position:	31 … 0
Bit field:	
Value after reset:	User setting*1

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.

See section 51.7. Secure Boot for details.

7.2.15 SACC1n : Start Address of Code Certificate 1n (n= 0 to 3)

Address: SACC1n: 0x02E0_7630 + 0x20 × n (n = 0 to 3)

Bit position:	31 … 0
Bit field:	
Value after reset:	User setting*1

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.

See section 51.7. Secure Boot for details.

7.2.16 SAMR : Start Address of Measurement Report

Address: 0x02E0_7614

Bit position:	31 … 0
Bit field:	
Value after reset:	User setting*1

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.

The SAMR register specifies the starting address of measurement report. The measurement report is stored in the SRAM shown in . Specify the address of SRAM0 to SRAM3. In case SRAM0 is used for measurement report, do not specify the addresses from 0x22007780 to 0x22007BFF. When reading the measurement report, disable the ECC function by setting SRAMCRn.ECCMOD[1:0] = 00b. See section 51.7. Secure Boot for details.

7.2.17 REVOKE : REVOKE Enable

Address: 0x02E1_7910

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

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

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.

Bit	Symbol	Function	R/W
3:0	REVOKE[3:0]	Configuration setting for OEM_ROOT_KEY revoke xxx0b: OEM_ROOT_PK0 is revoked, and read data is masked to 0 xxx1b: OEM_ROOT_PK0 is enabled and can be read xx0xb: OEM_ROOT_PK1 is revoked, and read data is masked to 0 xx1xb: OEM_ROOT_PK1 is enabled and can be read x0xxb: OEM_ROOT_PK2 is revoked, and read data is masked to 0 x1xxb: OEM_ROOT_PK2 is enabled and can be read 0xxxb: OEM_ROOT_PK3 is revoked, and read data is masked to 0 1xxxb: OEM_ROOT_PK3 is enabled and can be read	R
31:4	—	The program value is read from this bit.*1	R

Note 1. This register can only be programmed by MACI command. When programming, the set value should be 1.

7.2.18 HOEMRTRKn : Hash of OEM_ROOT_PKn (n= 0 to 3)

Address: HOEMRTPK0: 0x02E0_7400 +0x4 × i (i = 0 to 11)
HOEMRTPK1: 0x02E0_7430 +0x4 × i (i = 0 to 11)
HOEMRTPK2: 0x02E0_7460 +0x4 × i (i = 0 to 11)
HOEMRTPK3: 0x02E0_7490 +0x4 × i (i = 0 to 11)

Bit position:	31 ... 0
Bit field:	
Value after reset:	User setting*1

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.

This register can be programmed only by boot firmware.

7.2.19 ZHUK : Zeroization HUK Enable

Address: 0x02E1_7920

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	ZHUKEN3	ZHUKEN2	ZHUKEN1	ZHUKEN0
Value after reset:	User setting*1															

Bit	Symbol	Function	R/W
0	ZHUKEN0	Configuration setting for RTCIC0 0: W-HUK zeroization by RTCIC0 tamper event is enabled 1: W-HUK zeroization by RTCIC0 tamper event is disabled	R
1	ZHUKEN1	Configuration setting for RTCIC1 0: W-HUK zeroization by RTCIC1 tamper event is enabled 1: W-HUK zeroization by RTCIC1 tamper event is disabled	R
2	ZHUKEN2	Configuration setting for RTCIC2 0: W-HUK zeroization by RTCIC2 tamper event is enabled 1: W-HUK zeroization by RTCIC2 tamper event is disabled	R
3	ZHUKEN3	Configuration setting for MREZC register 0: W-HUK zeroization by MREZC is enabled 1: W-HUK zeroization by MREZC is disabled	R
31:4	—	The program value is read from this bit.*2	R

Note: When ZHUKEN3 bit is 0, set ZHUKENn (n = 0 to 2) bits setting same as VBTADCR3 (see section 12, Battery Backup Function). For example, when ZHUKENn (n = 0 to 2) is 101b, set VBTADCR3.VBATADZEn (n = 0 to 2) to 010b.

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.

Note 2. This register can only be programmed by MACI command. When programming, the set value should be 1.

7.2.20 ARCLS : Anti-Rollback Counter Lock Setting

Address: 0x02E1_7930

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	ARCBL_LK	ARCNS_LK[3:0]				ARCS_LK
Value after reset:	User setting*1															

Bit	Symbol	Function	R/W
0	ARCS_LK	ARC_SEC Lock 0: ARC_SEC protection is enabled 1: ARC_SEC protection is disabled	R
4:1	ARCNS_LK[3:0]	ARC_NSEC Lock In case of 64 bits x 4 counter setting x x x 0: ARC_NSEC[63:0] protection is enabled x x 0 x: ARC_NSEC[127:64] protection is enabled x 0 x x: ARC_NSEC[191:128] protection is enabled 0 x x x : ARC_NSEC[255:192] protection is enabled 1 1 1 1: ARC_NSEC[255:0] protection is disabled Others Reserved. In case of 256 bits counter setting 0 0 0 0: ARC_NSEC[255:0] protection is enabled 1 1 1 1: ARC_NSEC[255:0] protection is enabled Others Reserved.	R

Bit	Symbol	Function	R/W
5	ARCBL_LK	ARC_OENBL Lock 0: ARC_OEMBL protection is enabled 1: ARC_OEMBL protection is disabled	R
15:6	—	The program value is read from these bits*2	R

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application

Note 2. This register can only be programmed by MACI command. When programming, the set value should be 1.

Anti-rollback counter Lock bits setting are located in the OTP area. Anti-rollback counter Lock bits setting can be set to "0" by the "Program" command and cannot be changed from "0" to "1". When the lock bit is 0, the anti-rollback counter corresponding to each bit cannot be updated by the "Increment Counter" command.

7.2.21 ARCCS : Anti-Rollback Counter Configuration Setting for Non-secure Application

Address: 0x02E1_7932

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

Bit	Symbol	Function	R/W
1:0	CNF_ARCNS[1:0]	Configuration setting for ARC_NSEC 0 0: ARC_NSEC configuration is 4 × 64 bits 0 1: ARC_NSEC configuration is 1 × 256 bits Others ARC_NSEC configuration is disabled	R
15:2	—	The program value is read from these bits*2	R

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application

Note 2. This register can only be programmed by MACI command. When programming, the set value should be 1.

ARC_NSEC configuration setting is located in the OTP area. These bits specify two types of ARC_NSEC configuration: 4 × 64 bits or 1 × 256 bits. "Increment Counter" or "Read Counter" command cannot be issued to ARC_NSEC when counter configuration setting is disabled.

7.2.22 ARC_SECn : Anti-Rollback Counter for Secure Application n (n = 0, 1)

Address: 0x02F2 7E00 + 4 × n (n=0, 1)

Bit position:	31	0
Bit field:	ARC_SEC[32 × n + 31 : 32 × n]	
Value after reset:	User setting*1	

Bit	Symbol	Function	R/W
31:0	ARC_SEC[32 × n + 31 : 32 × n]	Anti-Rollback Counter for Secure Application The counter value is obtained by arranging the read values from the upper register (n = 1) to the lower register (n = 0).	R

Note 1. The value in a blank product is 0x0. It is set to the value written by your application

7.2.23 ARC_NSECn : Anti-Rollback Counter for Non-secure Application n (n = 0 to 7)

Address: 0x02F2 7E08 + 4 × n (n = 0 to 7)

Bit position:	31	0
Bit field:	ARC_NSECn[32 × n + 31 : 32 × n]	
Value after reset:	User setting*1	

Bit	Symbol	Function	R/W
31:0	ARC_NSECn[32 × n + 31 : 32 × n]	Anti-Rollback Counter for Non-secure Application The counter value is obtained by arranging the read values from the upper register (n = 7) to the lower register (n = 0).	R

Note 1. The value in a blank product is 0x0. It is set to the value written by your application

7.2.24 ARC_OEMBLn : Anti-Rollback Counter for OEMBL (n = 0, 1)

Address: 0x02F2 7E28 + 4 × n (n = 0, 1)

Bit position:	31	0
Bit field:	ARC_OEMBL[32 × n + 31 : 32 × n]	
Value after reset:	User setting*1	

Bit	Symbol	Function	R/W
31:0	ARC_OEMBL[32 × n + 31 : 32 × n]	Anti-Rollback Counter for OEM_BL Application The counter value is obtained by arranging the read values from the upper register (n = 1) to the lower register (n = 0).	R

Note 1. The value in a blank product is 0x0. It is set to the value written by your application

7.2.25 GPOTPn : General Purpose OTP (n = 0 to 23)

Address: 02E0 76A0h + 4 x n (n = 0 to 23)

Bit position:	31	0
Bit field:	GPOTP[31:0]	
Value after reset:	User setting*1	

Bit	Symbol	Function	R/W
31:0	GPOTP[31:0]	General Purpose OTP This is an OTP area that can be used for general purpose. Secure user can write any value.	R

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.

7.3 Setting Option-Setting Memory

7.3.1 Allocation of Data in Option-Setting Memory

Programming data is allocated to the addresses in the option-setting memory shown in Figure 7.1. The allocated data is used by FSBL or tools such as a flash programming software or an on-chip debugger.

Note: Programming formats vary depending on the compiler. See the compiler manual for details.

7.3.2 Setting Data for Programming Option-Setting Memory

Allocating data according to the procedure described in section 7.3.1. Allocation of Data in Option-Setting Memory, alone does not actually write the data to the option-setting memory. You must also follow one of the actions described in this section.

(1) Changing the option-setting memory by self-programming

Use the configuration setting command to write data to the option-setting memory in the configuration setting area.

The extra MRAM option-setting memory support background operations (BGO). When write the option-setting memory, the code MRAM can be read.

For details of the configuration setting command, see section 59, MRAM.

(2) Debugging through an OCD or programming by a flash writer

This procedure depends on the tool in use, see the tool manual for details.

The MCU provides two setting procedures:

- Read the data allocated as described in section 7.3.1. Allocation of Data in Option-Setting Memory, from an object file or Motorola S-format file generated by the compiler, and write the data to the MCU
- Use the GUI interface of the tool to program the same data as allocated in section 7.3.1. Allocation of Data in Option-Setting Memory.

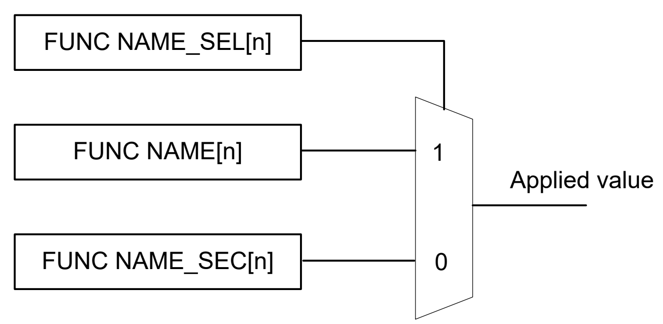
7.3.3 Security Attribution of Extra MRAM Option-Setting Memory

Some functionality has 3 registers for Non-secure (FUNC NAME), and Secure (FUNC NAME_SEC), and security attribution (FUNC NAME_SEL). Only Secure developer can set the registers for Secure and security attribution. As shown in Figure 7.2, when the bit of security attribution register is set to 0, the corresponding bit of Secure register is applied. When the bit of security attribution register is set to 1, the corresponding bit of Non-secure register is applied.

For example, if the Secure developer wants to configure PVD of OFS1 as Secure, HOCO of OFS1 as Non-secure, the Secure developer needs to set OFS1_SEL as follows.

OFS1_SEL = 0xFFFF_FFF0

By this setting, PVDAS and VDSEL[2:0] values of OFS1_SEC and HOCOFRQ0[2:0] and HOCOEN values of OFS1 are applied to MCU. The reserved bits of the security attribution register (FUNC NAME_SEL) should be set to 0.



Note: FUNC_NAME is corresponded to OFS1 and OFS3.

Figure 7.2 Selection of applied value

Block protect setting (BPS or BPS_SEC) and permanent block protect setting (PBPS or PBPS_SEC) are selected by the code MRAM security attribution.

7.3.4 Timing of the Setting Value

For SAS, BPS, BPS_SEC, PBPS and PBPS_SEC registers, the setting value of the related startup area and block protection is applied immediately after programming. For other registers, the setting value is applied after the MCU is reset.

7.4 Usage Notes

7.4.1 Data for Programming Reserved Areas and Reserved Bits in the Option-Setting Memory

- Security attribution region (0x02C9_F120 to 0x02C9_F1FF)
  When reserved areas and reserved bits in the option-setting memory are within the scope of programming, write 0 to all bits in reserved areas and all reserved bits. Operation is not guaranteed if 1 is written to these bits.
- Other region
  When reserved areas and reserved bits in the option-setting memory are within the scope of programming, write 1 to all bits of reserved areas and all reserved bits. If 0 is written to these bits, normal operation cannot be guaranteed.

7.4.2 FSBL Execution After System Reset

When FSBLCTRL0.FSBLEN bit is set to 000b, FSBL is always executed after system reset is released. However, setting FSBLCTRL0.FSBLSKIPSW or FSBLCTRL0.FSBLSKIPDS bit adds a condition to Skip FSBL. See Table 7.3.

Table 7.3 FSBL execution conditions

FSBLEN	FSBLSKIPSW	FSBLSKIPDS	FSBL execution after system reset
111b	*	*	Not executed
000b	111b	111b	Executed
000b	111b	000b	Deep Software Standby reset : Not executed Other reset than above : Executed
000b	000b	111b	Software reset : Not executed Other reset than above : Executed
000b	000b	000b	Software reset : Not executed Deep Software Standby reset : Not executed Other reset than above : Executed

When using the FSBL skip function, the reset status registers (RSTSR0, RSTSR1, and RSTSR3) should be cleared before the system reset is released.

Note: The FSBL will always run if at least one of the reset flags that allows the FSBL to run is set.

7.4.3 FSBL Skip Function

To skip FSBL execution after a software reset release, RSTSR0, RSTSR1, and RSTSR3 should be cleared before executing software reset. If any flags other than RSTSR1.SWRF is set, FSBL may be executed.

To skip FSBL execution after a Deep Software Standby Reset release, RSTSR0, RSTSR1, and RSTSR3 should be cleared before the transition to Deep Software Standby mode. If any flags other than RSTSR0.DPSRSTF is set, FSBL may be executed.

8. Programmable Voltage Detection (PVD)

8.1 Overview

The Programable Voltage Detection (PVD) module monitors the voltage level input to the VCC pin and the detection level can be selected using a software program. The PVD module consists of several separate voltage level detectors which measure the voltage level input to the VCC pin. PVD registers allow your application to configure detection of VCC change at various voltage thresholds.

Each voltage level detector has a voltage monitor associated with it. Voltage monitor registers are used to configure the PVD to trigger an interrupt, event link output, or reset when the thresholds are crossed.

Table 8.1 lists the PVD specifications. Figure 8.1 shows a block diagram of the voltage monitor 0 reset generation circuit. Figure 8.2 and Figure 8.3 show a block diagram of the voltage monitor m/n interrupt and reset circuit (m = 1, 2, n = 4, 5).

The PVD module consists of low voltage detection (LVD) and High voltage detection (HVD). LVD detects that VCC is dropping below the voltage threshold. HVD detects that VCC is rising above the voltage threshold.

PVDn are implemented for security purposes. Compared to PVDm, the functions available are more limited. PVDn has a register lock function for security.

The voltage level detector for core voltage is implemented. It monitors the voltage level input to the VDD, and the detection level is pre-defined by hardware. The voltage level detector for core voltage has a voltage monitor circuit associated with it. The voltage monitor circuit for core voltage can assert the reset (Core Voltage Monitor Reset) when the core voltage crossed the threshold as under Vdet_VDDL and over Vdet_VDDH.

The function of Core Voltage Monitor Reset (CVMR) is enabled when OFS2.CVMRDIS = 0.

Table 8.1 PVD specifications (1 of 2)

Item		Programable voltage monitor 0	Programable voltage monitor m (m = 1, 2)	Programable voltage monitor n (n = 4, 5)
Voltage Monitoring	Monitored pin	VCC	VCC	VCC
	Detected Event	Voltage falls past V_{det0}	Voltage rises or falls past V_{detm}	Voltage rises or falls past V_{detn}
	Detection Voltage	Selectable from among 8 different levels by using OFS1.VDSEL0 [2:0] bits as V_{det0}	Selectable from among 16 different levels by using PVDmCMPCR.PVDLVL[4:0] bits as V_{detm}	Selectable from among 16 different levels by using PVDnCMPCR.PVDLVL[4:0] bits as V_{detn}
	Monitor Flag	None	PVDmSR.MON flag: Monitors whether voltage is higher or lower than V_{detm}	None
			PVDmSR.DET flag: V_{detm} passage detection	
Process on voltage detection	Reset	Voltage monitor 0 reset	Voltage monitor m reset	Voltage monitor n reset
		Reset when V_{det0} > VCC CPU restart after specified time with VCC > V_{det0}	Reset condition selectable: Reset when V_{detm} > VCC (LVD) or V_{detm} < VCC (HVD) CPU restart timing selectable: after specified time with LVD or HVD	Reset condition selectable: Reset when V_{detn} > VCC (LVD) or V_{detn} < VCC (HVD)
	Interrupt	No interrupt	Voltage monitor m interrupt	No interrupt
			Non-maskable interrupt or maskable interrupt selectable	
			Interrupt request issued in response to either or both LVD and HVD	
Digital filter	Enable/ Disable Switching	Digital filter function not available	Available	Available
	Sampling Time	—	1/s LOCO frequency × 2 (s: 2, 4, 8, 16)	1/s LOCO frequency × 2 (s: 2, 4, 8, 16)
Event linking		None	Available Output of event signals on detection of V_{detm} crossings	None

Table 8.1 PVD specifications (2 of 2)

Item	Programable voltage monitor 0	Programable voltage monitor m (m = 1, 2)	Programable voltage monitor n (n = 4, 5)
Register lock	None	None	PVDLR

Table 8.2 CVM specifications

Item		Core voltage monitor
Voltage Monitoring	Monitored pin	VDD
	Detected Event	Voltage falls past V_{det}_VDDL Voltage rises past V_{det}_VDDH
	Detection Voltage	Pre-defined level*1
Process on voltage detection	Reset	Core Voltage monitor reset*2
		Reset when V_{det}_VDDL > VDD or V_{det}_VDDH < VDD CPU restart after specified time with V_{det}_VDDL ≤ VDD ≤ V_{det}_VDDH

Note 1. It means V_{det}_VDDL and V_{det}_VDDH

Note 2. See the chapter of Reset about core voltage monitor reset.

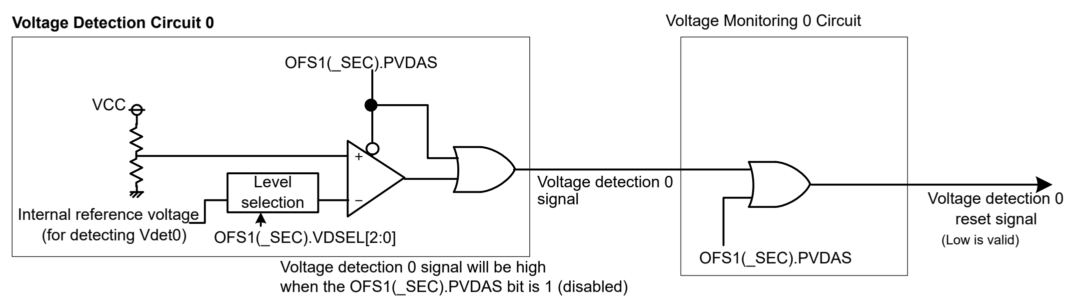


Figure 8.1 Block diagram of voltage monitor 0 reset circuit

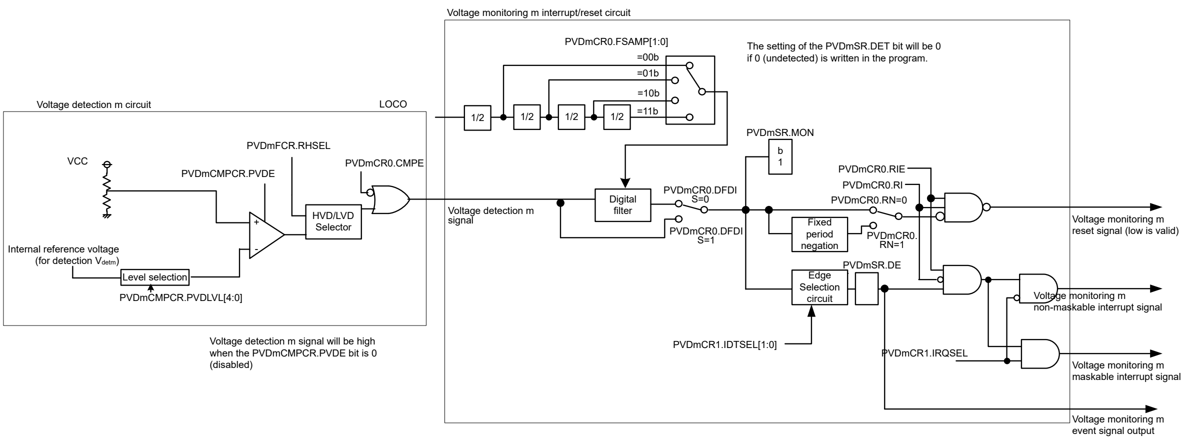


Figure 8.2 Block diagram of voltage monitor m interrupt/reset circuit (m = 1, 2. The m is integer).

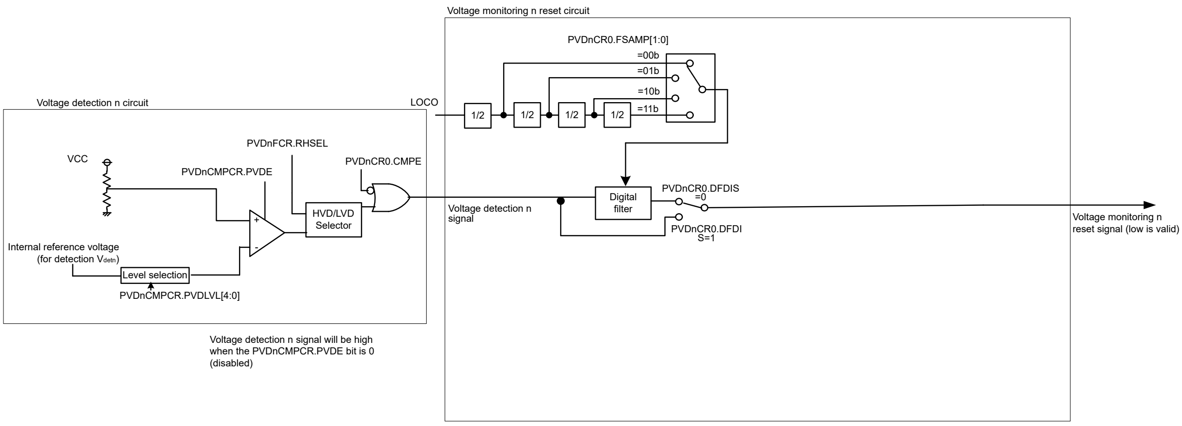


Figure 8.3 Block diagram of voltage monitor n reset circuit (n = 4, 5. The n is integer).

8.2 Register Descriptions

8.2.1 PVDSAR : Programable Voltage Detection Security Attribution Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x3CC

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	NONSEC1	NONSEC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NONSEC0	Non Secure Attribute bit 0 Target register: registers for PVD1 0: Secure 1: Non Secure	R/W
1	NONSEC1	Non Secure Attribute bit 1 Target register: registers for PVD2 0: Secure 1: Non Secure	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1
Note: This register is write-protected by PRCR register.

The PVDSAR register controls the secure attribute of PVD registers.

NONSEC0 bit (Non Secure Attribute bit 0)

This bit controls the security attribute of PVD1CMPCR, PVD1CR0, PVD1CR1, PVD1SR.

NONSEC1 bit (Non Secure Attribute bit 1)

This bit controls the security attribute of PVD2CMPCR,PVD2CR0, PVD2CR1, PVD2SR.

8.2.2 PVDmCMPCR : Voltage Monitor m Comparator Control Register (m = 1, 2)

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA58 + 0x4 × (m - 1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PVDE	—	—	PVDLVL[4:0]				
Value after reset:	0	0	0	0	1	1	1	1

Bit	Symbol	Function	R/W
4:0	PVDLVL[4:0]	Detection Voltage m Level Select Settings other than the following are prohibited. These are standard voltage level during the fall of voltage. For details on standard voltage level during the rise of voltage, see section 62, Electrical Characteristics. 0x03: 3.86 V (Vdetm_3) 0x04: 3.14 V (Vdetm_4) 0x05: 3.10 V (Vdetm_5) 0x06: 3.08 V (Vdetm_6) 0x07: 2.85 V (Vdetm_7) 0x08: 2.83 V (Vdetm_8) 0x09: 2.80 V (Vdetm_9) 0x0A: 2.62V (Vdetm_10) 0x0B: 2.33V (Vdetm_11) 0x0C: 1.90V (Vdetm_12) 0x0D: 1.86V (Vdetm_13) 0x0E: 1.74V (Vdetm_14) 0x0F: 1.71V (Vdetm_15) Others: Setting prohibited	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	PVDE	Voltage Detection m Enable 0: Voltage detection m circuit disabled 1: Voltage detection m circuit enabled	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

PVDLVL[4:0] bits (Detection Voltage m Level Select)

The PVDmCMPCR.PVDLVL can be changed only if all the PVDmCMPCR.PVDE bits and all the PVDnCMPCR.PVDE bits are 0. All the voltage detection circuits should not be set at the same voltage detection level.

Do not change PVDmCMPCR.PVDLVL and PVDmCMPCR.PVDE at the same time.

PVDE bit (Voltage Detection m Enable)

When using voltage detection m interrupt/reset or the PVDmSR.MON bit, set the PVDmCMPCR.PVDE bit to 1. The voltage detection m circuit starts once td(E-A) passes after the PVDmCMPCR.PVDE bit value is changed from 0 to 1. When using the voltage detection m circuit in Deep Software Standby mode, do not transit to Deep Software Standby mode 3.

8.2.3 PVDnCMPCR : Voltage Monitor n Comparator Control Register (n = 4, 5)

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA64+ 0x4 × (n - 4)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PVDE	—	—	PVDLVL[4:0]				
Value after reset:	0	0	0	0	1	1	1	1

Bit	Symbol	Function	R/W
4:0	PVDLVL[4:0]	Detection Voltage n Level Select Settings other than above are prohibited. (These are standard voltage level during fall of voltage. About standard voltage level during rise of voltage, refer to chapter of electrical characteristics.) 0 0 0 1 1 : 3.86V (Vdetn_3) 0 0 1 0 0 : 3.14V (Vdetn_4) 0 0 1 0 1 : 3.10V (Vdetn_5) 0 0 1 1 0 : 3.08V (Vdetn_6) 0 0 1 1 1 : 2.85V (Vdetn_7) 0 1 0 0 0 : 2.83V (Vdetn_8) 0 1 0 0 1 : 2.80V (Vdetn_9) 0 1 0 1 0 : 2.62V (Vdetn_10) 0 1 0 1 1 : 2.33V (Vdetn_11) 0 1 1 0 0 : 1.90V (Vdetn_12) 0 1 1 0 1 : 1.86V (Vdetn_13) 0 1 1 1 0 : 1.74V (Vdetn_14) 0 1 1 1 1 : 1.71V (Vdetn_15)	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	PVDE	Voltage Detection n Enable 0: Voltage detection n circuit disabled 1: Voltage detection n circuit enabled	R/W

Note: S-TYPE-6, P-TYPE-2
Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

The PVDnCMPCR.PVDLVL can be changed only if all the PVDmCMPCR.PVDE bits and all the PVDnCMPCR.PVDE bits are 0. All the voltage detection circuits should not be set at the same voltage detection level.

Please do not change PVDnCMPCR.PVDLVL and PVDnCMPCR.PVDE at the same time.

PVDE bit (Voltage Detection n Enable)

When using voltage detection n interrupt/reset or the PVDnSR.MON bit, set the PVDnCMPCR.PVDE bit to 1. The voltage detection n circuit starts once td(E-A) passes after the PVDnCMPCR.PVDE bit value is changed from 0 to 1. When using the voltage detection n circuit in deep software standby mode, do not transit to deep software standby mode 3.

8.2.4 PVDmCR0 : Voltage Monitor m Circuit Control Register 0 (m = 1, 2)

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA70 + 0x4 × (m - 1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RN	RI	FSAMP[1:0]		—	CMPE	DFDIS	RIE
Value after reset:	1	0	0	0	x	0	1	0

Bit	Symbol	Function	R/W
0	RIE	Voltage Monitor m Interrupt/Reset Enable 0: Disable 1: Enable	R/W
1	DFDIS	Voltage monitor m Digital Filter Disabled Mode Select 0: Enable the digital filter 1: Disable the digital filter	R/W
2	CMPE	Voltage Monitor m Circuit Comparison Result Output Enable 0: Voltage monitor m circuit comparison result output disabled 1: Voltage monitor m circuit comparison result output enabled	R/W
3	—	The read value is undefined. The write value should be 1.	R/W
5:4	FSAMP[1:0]	Sampling Clock Select 0 0: 1/2 LOCO frequency 0 1: 1/4 LOCO frequency 1 0: 1/8 LOCO frequency 1 1: 1/16 LOCO frequency	R/W
6	RI	Voltage Monitor m Circuit Mode Select (1) case of PVDmFCR.RHSEL = 0 0: Voltage monitor m interrupt during V_{detm} passage 1: Voltage monitor m reset enabled when the voltage falls to and below V_{detm} (2) case of PVDmFCR.RHSEL = 1 0: Prohibited 1: Voltage monitoring m reset enabled when the voltage rises to and up V_{detm}	R/W
7	RN	Voltage Monitor m Reset Negate Select (1) case of PVDmFCR.RHSEL = 0 0: Negation follows a stabilization time (t_{PVDm}) after VCC > V_{detm} is detected. 1: Negation follows a stabilization time (t_{PVDm}) after assertion of the PVDm reset. (2) case of PVDmFCR.RHSEL = 1 0: Negation follows a stabilization time (t_{PVDm}) after VCC < V_{detm} is detected. 1: Prohibited	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

RIE bit (Voltage Monitor m Interrupt/Reset Enable)

The RIE bit enables or disables voltage monitor m interrupt/reset. Ensure that neither a voltage monitor m interrupt nor a voltage monitor m reset is generated during programming or erasure of the MRAM.

DFDIS bit (Voltage monitor m Digital Filter Disabled Mode Select)

The DFDIS bit disables the digital filter circuit. Set the LOCOCR.LCSTP bit to 0 (the LOCO operates) if the DFDIS bit is 0 (enabling the digital filter circuit).Set the DFDIS bit to 1 (digital filter circuit disabled) when using voltage monitor m circuit in software standby mode or Deep Software Standby mode.

CMPE bit (Voltage Monitor m Circuit Comparison Result Output Enable)

The CMPE bit enables or disables voltage monitor m circuit comparison result output. Set the CMPE bit to 1 after the voltage detection m circuit enables and stabilization time ($t_{d(E-A)}$) elapses. When stopping the voltage detection m circuit, disable the voltage detection m circuit after setting the CMPE bit is 0.

FSAMP[1:0] bits (Sampling Clock Select)

The FSAMP[1:0] bits can be rewritten only when the PVDmCR0.DFDIS bit is 1 (digital filter circuit disabled). The FSAMP[1:0] bits should not be vmodified when the PVDmCR0.DFDIS bit is 0 (digital filter circuit enabled).

RI bit (Voltage Monitor m Circuit Mode Select)

When the PVDmCR0.RI bit is 1 (voltage monitor m reset selected), a transition to Deep Software Standby mode 2 or 3 cannot be made, instead a transition to Deep Software Standby mode 1 is made. To enter Deep Software Standby mode 2 or 3, set all the PVDmCR0.RI bits to 0 (voltage monitor m interrupt selected).

RN bit (Voltage Monitor m Reset Negate Select)

Case of PVDmFCR.RHSEL = 0

If the RN bit is set to 1 (negation follows a stabilization time after assertion of the PVDm reset signal), set the LOCOCR.LCSTP bit to 0 (the LOCO operates). Furthermore, if a transition to Software Standby mode or Deep Software Standby mode is to be made, the only possible value for the RN bit is 0 (negation follows a stabilization time after VCC > V_{detm} is detected). Do not set the RN bit to 1 (negation follows a stabilization time after assertion of the PVDm reset signal) when this is the case.

Case of PVDmFCR.RHSEL = 1

Do not set the RN bit to 1.

8.2.5 PVDnCR0 : Voltage Monitor n Circuit Control Register 0 (n = 4, 5)

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA7C + 0x4 × (n - 4)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	FSAMP[1:0]		—	CMPE	DFDIS	RE
Value after reset:	0	1	0	0	x	0	1	0

Bit	Symbol	Function	R/W
0	RE	Voltage Monitor n Reset Enable 0: Disabled 1: Enabled	R/W
1	DFDIS	Voltage Monitor n Digital Filter Disable Mode Select 0: Digital filter enabled 1: Digital filter disabled	R/W
2	CMPE	Voltage Monitor n Circuit Comparison Result Output Enable 0: Voltage monitor n circuit comparison result output disabled. 1: Voltage monitor n circuit comparison result output enabled.	R/W
3	—	The read value is undefined. The write value should be 1.	R/W
5:4	FSAMP[1:0]	Sampling Clock Select 0 0: 1/2 LOCO frequency 0 1: 1/4 LOCO frequency 1 0: 1/8 LOCO frequency 1 1: 1/16 LOCO frequency	R/W
6	—	This bit is read as 1. The write value should be 1.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-6, P-TYPE-2
Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

RE bit (Voltage Monitor n Reset Enable)

RE bit decides whether to enable Voltage Monitor n Reset. When RE bit is 1, this function is activated. Ensure that a voltage Monitor n reset isn't generated during programming or erasure of the MRAM.

DFDIS bit (Voltage Monitor n Digital Filter Disable Mode Select)

Set the LOCOCR.LCSTP bit to 0 (the LOCO operates) if the DFDIS bit is 0 (enabling the digital filter circuit).

Set the DFDIS bit to 1 (digital filter circuit disabled) when using voltage monitor n circuit in software standby mode or deep software standby mode.

FSAMP[1:0] bits (Sampling Clock Select)

The FSAMP[1:0] bits can be modified only when the PVDnCR0.DFDIS bit is 1 (digital filter circuit disabled). The FSAMP[1:0] bits should not be modified when the PVDnCR0.DFDIS bit is 0 (digital filter circuit enabled).

8.2.6 PVDmCR1 : Voltage Monitor m Circuit Control Register (m = 1, 2)

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x0E0 + 0x2 × (m-1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	IRQSEL	IDTSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
1:0	IDTSEL[1:0]	Voltage Monitor m Interrupt Generation Condition Select (1) case of PVDmFCR.RHSEL = 0 0 0: When VCC ≥ V_{detm} (rise) is detected 0 1: When VCC < V_{detm} (fall) is detected 1 0: When fall and rise are detected 1 1: Settings prohibited (2) case of PVDmFCR.RHSEL = 1 This bit is not applicable.	R/W
2	IRQSEL	Voltage Monitor m Interrupt Type Select 0: Non-maskable interrupt 1: Maskable interrupt*1	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC3 bit to 1 (writing enabled) before rewriting this register.
Note 1. When enabling maskable interrupts, do not change the value of the NMIER.PVDmEN bit on the ICU side from the reset state.

8.2.7 PVDmSR : Voltage Monitor m Circuit Status Register (m = 1, 2)

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x0E1 + 0x2 × (m - 1)

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

Bit	Symbol	Function	R/W
0	DET	Voltage Monitor m Voltage Change Detection Flag*1 (1) case of PVDmFCR.RHSEL = 0 0: Not detected 1: V_{detm} passage detection (2) case of PVDmFCR.RHSEL = 1 This bit is not applicable.*2	R/W
1	MON	Voltage Monitor 1 Signal Monitor Flag (1) case of PVDmFCR.RHSEL = 0 0: VCC ≤ V_{detm} 1: VCC > V_{detm} or MON is disabled (2) case of PVDmFCR.RHSEL = 1 0: VCC ≥ V_{detm} 1: VCC < V_{detm} or MON is disabled	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.
Note 1. Only 0 can be written to this bit. After writing 0 to this bit, 2 system clock cycles are required for the bit to be read as 0.
Note 2. DET is depended on PVDmCR1.IDETSEL[1:0].
PVDmCR1.IDETSEL[1:0] = 0 0: When VCC < Vdetm (fall) is detected, DET is set.

PVDmCR1.IDETSEL[1:0] = 0 1: When VCC ≥ Vdetm (rise) is detected, DET is set.
PVDmCR1.IDETSEL[1:0] = 1 0: When fall or rise are detected, DET is set.
PVDmCR1.IDETSEL[1:0] = 1 1: Settings prohibited

DET flag (Voltage Monitor m Voltage Change Detection Flag)

The DET flag is enabled when the PVDmCMPCR.PVDE bit is 1 (voltage detection m circuit enabled) and the PVDmCR0.CMPE bit is 1 (voltage monitor m circuit comparison result output enabled).

The DET flag should be set to 0 after PVDmCR0.RIE is set to 0 (disabled). PVDmCR0.RIE can be set to 1 (enabled) after a period of two or more cycles of PCLKB has elapsed.

Depending on the number of cycles of PCLKB defined for access to read an I/O register, two or more cycles than PCLKB may have to be secured as waiting time.

MON flag (Voltage Monitor 1 Signal Monitor Flag)

The MON flag is enabled when the PVDmCMPCR.PVDE bit is 1 (voltage detection m circuit enabled) and the PVDmCR0.CMPE bit is 1 (voltage monitor m circuit comparison result output enabled).

8.2.8 PVDmFCR : Voltage Monitor m Function Control Register (m = 1, 2)

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xB20 + 0x4 × (m - 1)

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

Bit	Symbol	Function	R/W
0	RHSEL	Rise Hysteresis Select 0: Hysteresis level for VCC-fall detection is selected. 1: Hysteresis level for VCC-rise detection is selected.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

RHSEL bit (Rise Hysteresis Select)

When RHSEL = 0 for VCC-fall detection, the hysteresis level is set above the voltage detection level set by PVDLVL[4:0]. When RHSEL = 1 for VCC-rise detection, the hysteresis level is set under the voltage detection level set by PVDLVL[4:0]. See the chapter of electrical characteristics about the detail of hysteresis level of PVD.

When PVD reset generated by the VCC-fall detection is required, you should set RHSEL to 0.

When PVD reset generated by the VCC-rise detection is required, you should clear RHSEL to 1.

RHSEL must not be set to 1 when PVDmCR0.RI = 0.

RHSEL can be changed only if all the PVDmCMPCR.PVDE bits and all the PVDnCMPCR.PVDE bits are 0.

8.2.9 PVDnFCR : Voltage Monitor n Function Control Register (n = 4, 5)

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xB2C + 0x4 × (n - 1)

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

Bit	Symbol	Function	R/W
0	RHSEL	Rise Hysteresis Select 0: Hysteresis level for VCC-fall detection is selected. 1: Hysteresis level for VCC-rise detection is selected.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-6, P-TYPE-2

RHSEL bit (Rise Hysteresis Select)

When RHSEL = 0 for VCC-fall detection, the hysteresis level is set above the voltage detection level set by PVDLVL[4:0]. When RHSEL = 1 for VCC-rise detection, the hysteresis level is set under the voltage detection level set by PVDLVL[4:0] . See the chapter of electrical characteristics about the detail of hysteresis level of PVD.

When PVD reset generated by the VCC-fall detection is required, you should set RHSEL to 0.

When PVD reset generated by the VCC-rise detection is required, you should clear RHSEL to 1.

RHSEL can be changed only if all the PVDmCMPCR.PVDE bits and all the PVDnCMPCR.PVDE bits are 0.

8.2.10 PVDLR : Voltage Monitor Lock Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xB34

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

Bit	Symbol	Function	R/W
0	LOCK	LOCK control 0: The control registers for PVD4 and PVD5 can be written. 1: The control registers for PVD4 and PVD5 can not be written.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-6, P-TYPE-2

This register is initialized by POR, RES pin reset, Voltage Monitor 0 reset (PVD0 reset).

LOCK bit (LOCK control)

The initial value of LOCK bit is 1. When the LOCK bit is 1, the control registers for PVD4 and PVD5 can not be written. If the LOCK bit is written to 0 after the LOCK is initialized by the resets which can initialize the LOCK, the control registers for PVD4 and PVD5 can be written.

After that, if you write an arbitrary value to the LOCK, the LOCK bit is fixed to 1. Then the control registers for PVD4 and PVD5 can not be written.

Here, the control registers for PVD4 and PVD5 are PVD4CMPCR, PVD5CMPCR, PVD4CR0, PVD5CR0, PVD4FCR, PVD5FCR.

8.3 VCC Input Voltage Monitor

8.3.1 Monitoring V_{det0}

The comparison results from voltage monitor 0 are not available for reading.

8.3.2 Monitoring V_{detm}

After the settings are completed, results of comparison by voltage Monitor m can be monitored by using the PVDmSR.MON flag.

Table 8.3 Procedures to set up monitoring against V_{detm}

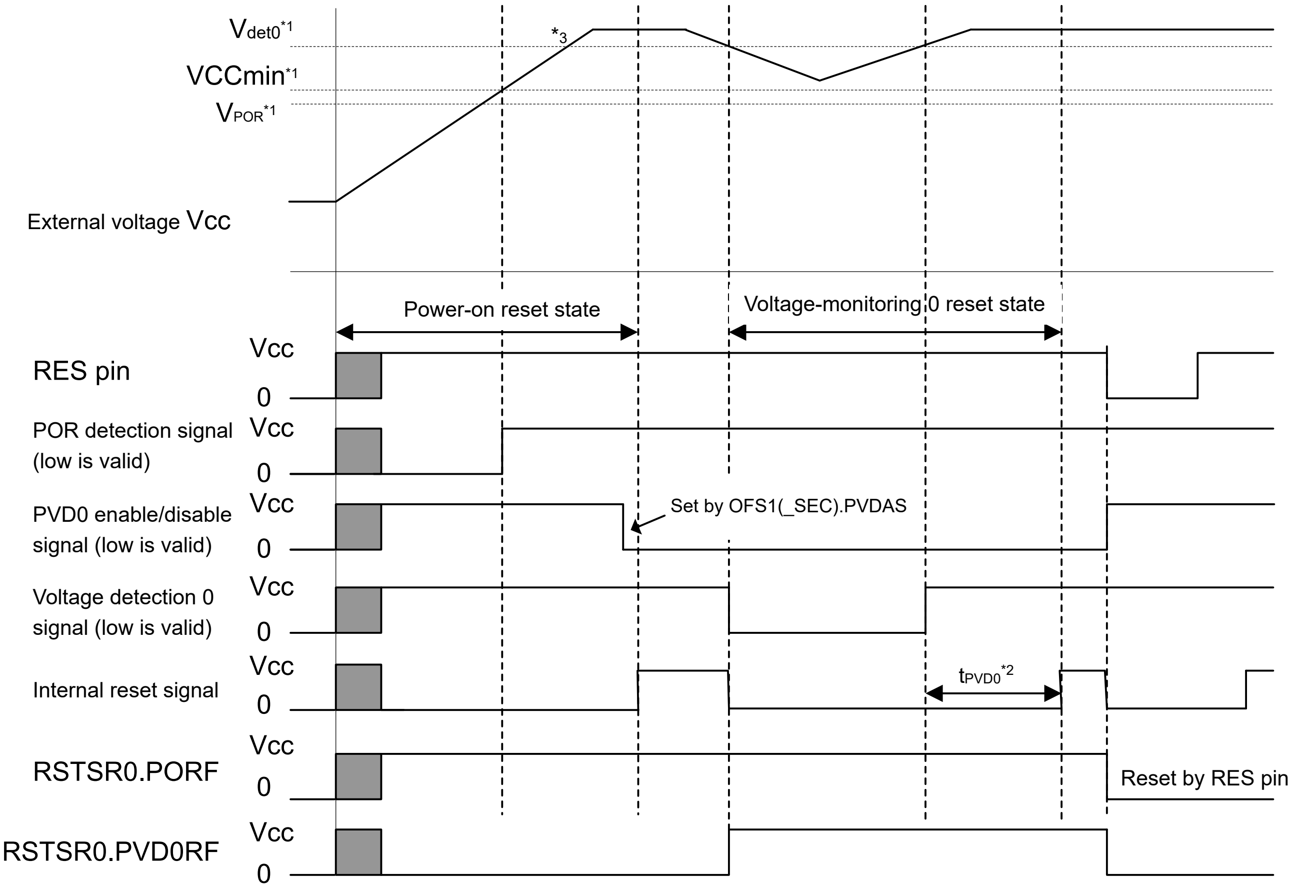
Step		Monitoring the comparison results from voltage monitor 1
Setting the voltage detection m circuit	1	Set PVDmCMPCR.PVDE = 0 to disable voltage detection m before writing to the PVDmCMPCR.PVDLVL [4:0] bits.
	2	Select the detection voltage in the PVDmCMPCR.PVDLVL[4:0] bits.
	3	Set PVDmCMPCR.PVDE = 1 to enable the voltage detection m circuit.
	4	Wait for at least $t_{d\ (E-A)}$ *1.
Setting the digital filter*2	5	Select the sampling clock for the digital filter in the PVDmCR0.FSAMP[1:0] bits.
	6	Set PVDmCR0.DFDIS = 0 to enable the digital filter.
	7	Wait for at least 2s + 3 cycles of the LOCO (where s = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by s).
Enabling output	8	Set PVDmCR0.CMPE = 1 to enable output of the comparison results from voltage monitor m.

Note 1. Steps 5 to 7 can be performed during the waiting time of step 4.
Note 2. Steps 5 to 7 are not required if the digital filter is not in use.

8.4 Reset from Voltage Monitor 0

When using the reset from voltage monitor 0, clear the OFS1(_SEC).PVDAS bit to 0. (enabling the voltage monitor 0 reset after a reset)

Figure 8.4 shows an example of operations for a voltage monitor 0 reset.



Note: For details of the electrical characteristics, see section 62, Electrical Characteristics.
Note 1. V_{POR} indicates the detection level for a power-on reset and V_{det0} indicates the detection level for a voltage monitor 0 reset and VCC_{min} shows minimum guaranteed voltage of MCU.
Note 2. t_{PVD0}: the period until the voltage monitor 0 reset is released.
Note 3. At the time the power-supply voltage rises, VCC must rise to at least the minimum guaranteed voltage before release from the POR reset state.

Figure 8.4 Example of voltage monitor 0 reset operation

8.5 Interrupt and Reset from Voltage Monitor m (m = 1, 2)

An interrupt or reset can be generated in response to the results of comparison by the voltage detection m circuit

Table 8.4 shows the procedures for setting bits related to the voltage monitor m interrupt and voltage monitor m reset so that voltage monitoring operates. Table 8.5 shows the procedures for setting bits related to the voltage monitor m interrupt and voltage monitor m reset so that voltage monitoring stops. Figure 8.5 shows an example of operations for a voltage monitor m interrupt. For the operation of the voltage monitor m reset, see Figure 6.2 in section 6, Resets.

Furthermore, if you intend to use the voltage Monitor m circuit in software standby or Deep Software Standby mode, make settings for the voltage Monitor m circuit according to the following procedures.

(1) Setting in Software Standby mode

- Disable the digital filter (PVDmCR0.DFDIS = 1).
- After VCC > V_{detm} is detected, negate the voltage monitor m reset signal (PVDmCR0.RN = 0) following a stabilization time.

(2) Settings in Deep Software Standby mode

- Disable the digital filter (PVDmCR0.DFDIS = 1).
- Enable the voltage monitor m interrupt (PVDmCR0.RI = 0). If the voltage monitor m reset is enabled (PVDmCR0.RI = 1), a transition to Deep Software Standby mode 2 or 3 is not possible, and the transition will be to Deep Software Standby mode 1 instead.
- During Deep Software Standby mode 3, the voltage Monitor m circuit is stopped. If you intend to use the voltage Monitor m circuit in Deep Software Standby mode, do not transit to Deep Software Standby mode 3.

(3) Disabling at on-chip debug mode

- The function of the voltage Monitor m circuit is disabled by setting DBGSTOPCR.DBGSTOP_PVD at on-chip debug mode.

Table 8.4 Procedures for setting bits related to voltage monitor m interrupt and voltage monitor m reset so that voltage monitoring operates (m = 1, 2)

Step		Voltage monitor m interrupt (voltage monitor m ELC event output)	Voltage monitor m reset
Setting up the voltage detection m circuit	1	Set PVDmCMPCR.PVDE = 0 (disable voltage detection m before writing to the PVDmCMPCR register.)	
	2	Select the detection voltage by setting the PVDmCMPCR.PVDLVL bits.	
	3	—	Select the type of the reset condition by setting the PVDmFCR.RHSEL bit.
	4	Set PVDmCMPCR.PVDE = 1 (enabling the voltage detection m circuit).	
	5	Wait for at least $t_{d\ (E\text{-}A)}$ *1.	
Setting the digital filter*2	6	Select the sampling clock for the digital filter by the setting PVDmCR0.FSAMP[1:0] bits.	
	7	Set PVDmCR0.DFDIS = 0 (enabling the digital filter).	
	8	Wait for at least 2s + 3 cycles of the LOCO (where s = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by s) *4.	
Setting up the voltage monitor m interrupt or reset	9	Set PVDmCR0.RI = 0 to select the voltage monitor m interrupt.	• Set PVDmCR0.RI = 1 (selecting the voltage monitor m reset). • Select the type of the reset negation by setting the PVDmCR0.RN bit.
	10	• Select the timing of interrupt requests by setting the PVDmCR1.IDTSEL[1:0] bits. • Select the type of interrupt by setting the PVDmCR1.IRQSEL bit.	—
Enabling output	11	Set PVDmSR.DET = 0.	
	12	Set PVDmCR0.RIE = 1 (enabling the voltage Monitor m interrupt or reset) *3.	
	13	Set PVDmCR0.CMPE = 1 (enabling output of the results of comparison by voltage monitor m).	

Note 1. Steps 6 to 12 can be performed during the wait time in step 5.
Note 2. Steps 6 to 8 are not required if the digital filter is not in use.
Note 3. Step 12 is not required if only the ELC event signal is to be output.
Note 4. Steps 9 to 12 can be performed during the waiting time of step 8.

Table 8.5 Procedures for setting bits related to voltage monitor m interrupt and voltage monitor m reset so that voltage monitoring stops (m = 1, 2) (1 of 2)

Step		Voltage monitor m interrupt (voltage monitor m ELC event output), voltage monitor m reset
Stopping the enabling output	1	Set PVDmCR0.CMPE = 0 (disabling output of the results of comparison by voltage Monitor m).
	2	Wait for at least 2s + 3 cycles of the LOCO (where s = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by s) *1.
	3	Set PVDmCR0.RIE = 0 (disabling the voltage monitor m interrupt or reset)*3.
Stopping the digital filter	4	Set PVDmCMPCR.DFDIS = 1 (disabling the digital filter) *1 *2.

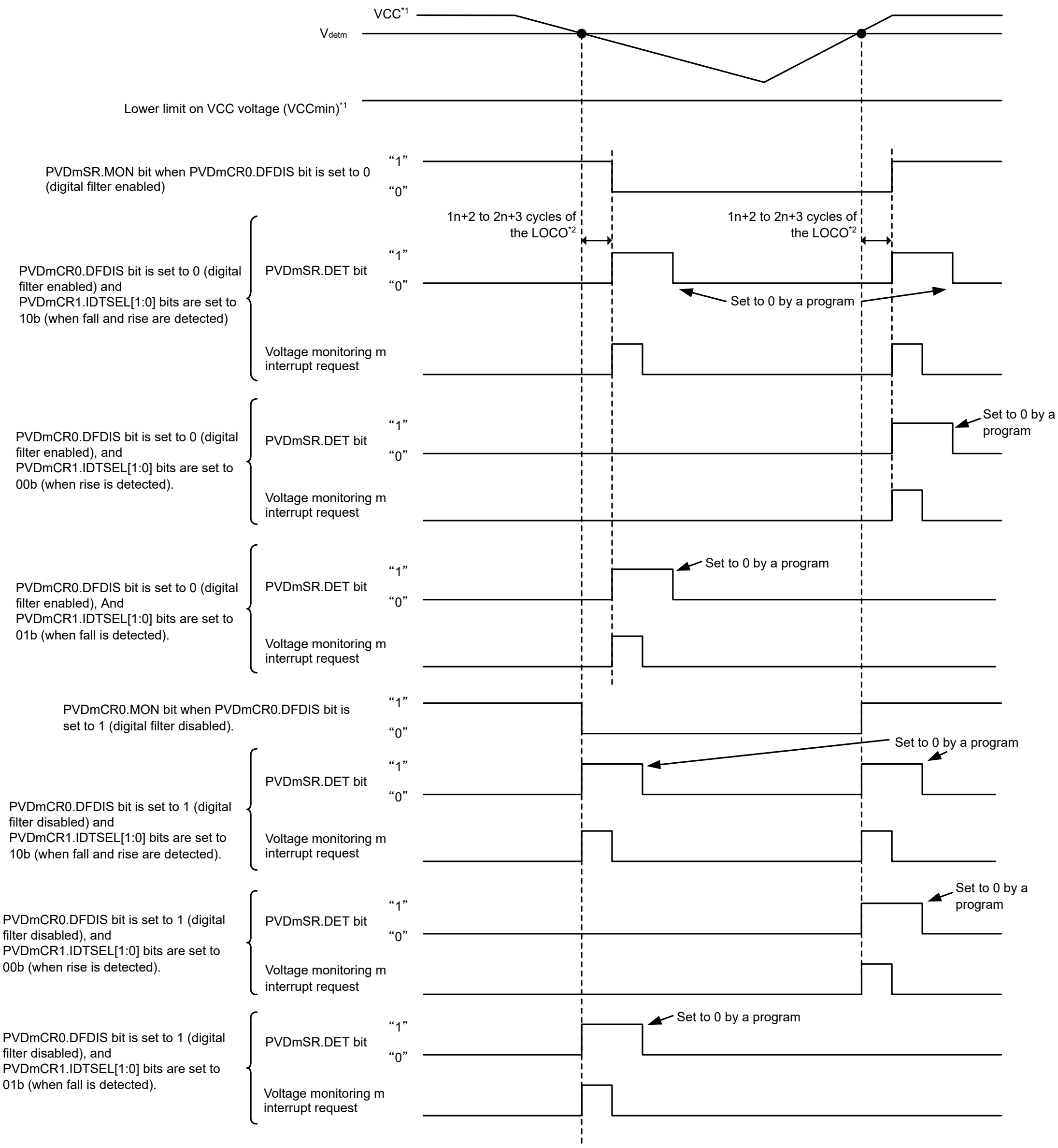
Table 8.5 Procedures for setting bits related to voltage monitor m interrupt and voltage monitor m reset so that voltage monitoring stops (m = 1, 2) (2 of 2)

Step		Voltage monitor m interrupt (voltage monitor m ELC event output), voltage monitor m reset
Stopping the voltage detection m circuit	5	Set PVDmCMPCR.PVDE = 0 (disabling the voltage detection m circuit).

Note 1. Steps 2 and 4 are not required if the digital filter is not in use.
Note 2. To disable the digital filter from its enabled state and then re-enable it, disable it and wait for at least two cycles of the LOCO before re-enabling it.
Note 3. Step 3 is not required if only the ELC event signal is to be output.

If the voltage monitor m interrupt or voltage monitor m reset setting is to be made again after it has been used and stopped once, the following steps in the procedures for stopping and making the setting can be omitted according to the condition:

- Setting or stopping the voltage detection m circuit is not required if the setting for the voltage detection m circuit is not to be changed
- Setting or stopping the digital filter is not required if the setting for the digital filter is not to be changed
- Setting the voltage monitor m interrupt or reset is not required if the setting for the voltage monitor m interrupt or voltage monitor m reset is not to be changed.



Note 1. When the voltage monitor 0 reset is not in use, use at VCC ≥ VCCmin.
Note 2. Frequency of the sampling clock for the digital filter is the LOCO frequency divided by n.

Figure 8.5 Example of voltage monitor m interrupt operation (m = 1, 2)

8.6 Reset from Voltage Monitor n = 4, 5

A reset can be generated in response to the results of comparison by the voltage detection n circuit.

Table 6.6 lists the procedures for setting bits related to the voltage monitor n reset so that voltage monitoring operates. Table 6.7 shows the procedure for setting bits related to the voltage monitor n reset so that voltage monitoring stops. For the operation of the voltage monitor n reset, see section 6.3.3. Voltage Monitor Reset.

Furthermore, if you intend to use the voltage Monitor n circuit in software standby or deep software standby mode, make settings for the voltage Monitor n circuit according to the following procedures

1. Setting in software standby mode
   - Disable the digital filter (PVDnCR0.DFDIS = 1).

- After VCC > Vdetn is detected, negate the voltage Monitor n reset signal following a stabilization time.

2. Settings in deep software standby mode
   - Disable the digital filter (PVDnCR0.DFDIS = 1)
   - During deep software standby mode 3, the voltage Monitor n circuit is stopped. If you intend to use the voltage Monitor n circuit in deep software standby mode, do not transit to deep software standby mode 3.

Table 8.6 Procedures for setting bits related to the voltage monitor n reset so that voltage monitoring operates

Procedure	Step	Voltage Monitor n Reset
Setting the voltage detection n circuit	1	Set PVDnCMPCR.PVDE = 0 (disable voltage detection n before writing to PVD nCMPCR.PVDLVL)
	2	Select the detection voltage by setting the PVDnCMPCR.PVDLVL[4:0] bits.
	3	Select the type of the reset condition by setting the PVDnFCR.RHSEL bit.
	4	Set PVDnCMPCR.PVDE = 1 (enabling the voltage detection n circuit).
	5	Wait for at least td (E-A).*1
Setting the digital filter*2	6	Select the sampling clock for the digital filter by setting the PVDnCR0.FSAMP[1:0] bits.
	7	Set PVDnCR0.DFDIS = 0 (enabling the digital filter).
	8	Wait for at least 2s + 3 cycles of the LOCO (where s = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by s).*3
Enabling output	9	Set PVDnCR0.RE = 1 (enabling the voltage Monitor n reset).
	10	Set PVDnCR0.CMPE = 1 (enabling output of the results of comparison by voltage Monitor n).

Note 1. Steps 5 to 10 can be performed during the waiting time of step 4.
Note 2. Steps 5 to 8 are not required if the digital filter is not in use.
Note 3. Steps 9 can be performed during the waiting time of step 8.

Table 8.7 Procedures for setting bits related to the voltage monitor n reset so that voltage monitoring stops

Procedure	Step	Voltage Monitor n Reset
Settings to stop enabling of output	1	Set PVDnCR0.CMPE = 0 (disabling output of the results of comparison by voltage Monitor n).
	2	Wait for at least 2s + 3 cycles of the LOCO (where s = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by s).*1
	3	Set PVDnCR0.RIE = 0 (disabling the voltage Monitor n reset).
Stopping the digital filter	4	Set PVDnCR0.DFDIS = 1 (disabling the digital filter).*1*2
Stopping the voltage detection n circuit	5	Set PVDnCMPCR.PVDE = 0 (disabling the voltage detection n circuit).

Note 1. Steps 2 and 4 are not required if the digital filter is not in use.
Note 2. To disable the digital filter from its enabled state and then re-enable it, disable it and wait for at least two cycles of the LOCO before re-enabling it.

If the voltage Monitor n reset setting is to be made again after it has been used and stopped once, the following steps in the procedures for stopping and making the setting can be omitted according to the condition.

- Setting or stopping the voltage detection n circuit is not required if the setting for the voltage detection n circuit is not to be changed.
- Setting or stopping the digital filter is not required if the setting for the digital filter is not to be changed.
- Setting the voltage Monitor n interrupt or reset is not required if the setting for the voltage Monitor m interrupt or voltage Monitor n reset is not to be changed.

8.7 Event Link Controller (ELC) Output

The PVDm (m = 1, 2) can output the event signals to the Event Link Controller (ELC).

(1) V_{detm} passage detection event

The PVDm outputs the event signal when it is detected that the voltage has passed the V_{detm} voltage while both the voltage detection m circuit and the voltage monitor m circuit comparison result output are enabled.

When enabling the event link output function of the PVDm, be sure to specify settings to enable the PVDm before enabling the PVDm event link function of the ELC. To stop the event link output function of the PVDm, be sure to specify settings to stop the PVDm before disabling the PVDm event link function of the ELC.

8.7.1 Interrupt Handling and Event Linking

Each PVD has bits to separately enable or disable the voltage monitor m interrupts. When an interrupt source is generated and the interrupt is enabled by the interrupt enable bit, the interrupt signal is output to the CPU.

On the contrary, as soon as an interrupt source is generated, the event link signal is output as the event signal to the other module through the ELC regardless of the state of the interrupt enable bit.

It is possible to output voltage monitor m interrupts in Software Standby mode, Deep Software Standby mode 1, and Deep Software Standby mode 2. The event signals for the ELC in Software Standby mode, Deep Software Standby mode 1, and Deep Software Standby mode 2 are output as follows:

- When the event V_{detm} is detected in Software Standby mode, no event signals are generated for the ELC because no clock is presented in Software Standby mode. Because V_{detm} passage detection flag is preserved, when the supply of the clock is resumed after restoring from Software Standby mode, the event signals for the ELC is output according to the state of the V_{detm} passage detection flag.
- If event of passing V_{detm} is detected in Deep Software Standby mode, no event signal is generated for the ELC.

9. Clock Generation Circuit

9.1 Overview

The MCU provides a clock generation circuit. Table 9.1 and Table 9.2 list the clock generation circuit specifications. Figure 9.1 show a block diagram, and Table 9.3 lists the I/O pins.

Table 9.1 Clock generation circuit specifications for the clock sources (1 of 2)

Clock source	Description	Specification
Main clock oscillator (MOSC)	Resonator frequency	8 MHz to 48 MHz
	External clock input frequency	Up to 48 MHz
	Connectable resonator or additional circuit: ceramic resonator, crystal Connection pins: EXTAL, XTAL	Available
	Drive capability switching	Available
	Automatic Gain Control Function	Available
	Oscillation stop detection function	Available
Sub-clock oscillator (SOSC)	Resonator frequency	32.768 kHz
	Connectable resonator or additional circuit: crystal resonator Connection pin: XCIN, XCOUT	Available
	External clock input	Available
	Drive capability switching	Available
	Oscillation stop detection function	Available
PLL1 circuit PLL2 circuit	Input clock source	MOSC/HOCO
	Input pulse frequency division ratio	Selectable from 1/2/3
	Input clock frequency	8 MHz to 48 MHz
	Input clock frequency (After input frequency division)	8 MHz to 24 MHz
	Frequency multiplication ratio	Selectable from 40 to 300 (after the decimal point : 0/0.33/0.50/0.66)
	VCO frequency	960 MHz to 2400 MHz
	Number of output clocks	Output 3 different clocks
	Output pulse frequency division ratio	Number of output clocks Output 3 different clocks
	PLL Output clock P	60 to 1200 MHz (output division ratio : 2/3/4/6/8/16)
	PLL Output clock Q	106.6 to 1200 MHz (output division ratio : 1.5/2/3/4/5/6/8/9)
	PLL Output clock R	106.6 to 1200 MHz (output division ratio : 1.5/2/3/4/5/6/8/9)
High-speed on-chip oscillator (HOCO)	Oscillation frequency	16/18/20/32/48/ MHz
	FLL function	Available
	User trimming	Available
Middle-speed on-chip oscillator (MOCO)	Oscillation frequency	8 MHz
	User trimming	Available
Low-speed on-chip oscillator (LOCO)	Oscillation frequency	32.768 kHz
	User trimming	Available

Table 9.1 Clock generation circuit specifications for the clock sources (2 of 2)

Clock source	Description	Specification
External clock input for JTAG (TCK)	Input clock frequency	Up to 25 MHz
External clock input for SWD (SWCLK)	Input clock frequency	Up to 25 MHz

Table 9.2 Clock generation circuit specifications for the internal clocks (1 of 3)

Item	Clock source	Maximum Frequency of Clock Source	Clock supply	Specification
CPU0 clock (CPUCLK0)	MOSC/SOSC/HOCO/ MOCO/PLL1P	1 GHz	CPU0	Up to 1 GHz Division ratio: 1/2/3/4/6/8/12/16/24/32/64
CPU1 clock (CPUCLK1)	MOSC/SOSC/HOCO/ MOCO/PLL1P	1 GHz	CPU1	Up to 250 MHz Division ratio: 1/2/3/4/6/8/12/16/24/32/64
MRAM bus clock (MRICLK)	MOSC/SOSC/HOCO/ MOCO/PLL1P	1 GHz	Code MRAM, MRAM code bus	Up to 250 MHz Division ratio: 1/2/3/4/6/8/12/16/24/32/64
MRAM clock (MRPCLK)	MOSC/SOSC/HOCO/ MOCO/PLL1P	1 GHz	Extra MRAM, MRAM peripheral bus	Up to 125 MHz Division ratio: 1/2/3/4/6/8/12/16/24/32/64
System clock (ICLK)	MOSC/SOSC/HOCO/ MOCO/PLL1P	1 GHz	DMAC, DTC, SRAM, System BUS, I/O Ports, ICU	Up to 250 MHz Division ratio: 1/2/3/4/6/8/12/16/24/32/64
Debugger clock (DCLK)	ICLK	—	Debug Subsystem	Up to 250 MHz
Peripheral module clock A (PCLKA)	MOSC/SOSC/HOCO/ MOCO/PLL1P	1 GHz	High-speed peripheral bus (ESWM, , USBHS, SCI, I3C, CANFD, CNECC, SPI, CRC, DOC, ADC16H, RSIP-E50D, GPT, PDG, DOTF AES clock, CEU)	Up to 125 MHz Division ratio: 1/2/3/4/6/8/12/16/24/32/64
Peripheral module clock B (PCLKB)	MOSC/SOSC/HOCO/ MOCO/PLL1P	1 GHz	Low-speed peripheral bus (CAC, ELC, POEG, RTC, WDT, IWDT, AGT, ULPT, IIC, USBFS, SSIE, SDHI, TSN, ACMPHS, OSPI, DOTF bus clock, DAC12, PDMIF)	Up to 62.5 MHz Division ratio: 1/2/3/4/6/8/12/16/24/32/64
Peripheral module clock C (PCLKC)	MOSC/SOSC/HOCO/ MOCO/PLL1P	1 GHz	Not used	Up to 125 MHz Division ratio: 1/2/3/4/6/8/12/16/24/32/64
Peripheral module clock D (PCLKD)	MOSC/SOSC/HOCO/ MOCO/PLL1P	1 GHz	Peripheral module(GPT count clock, PDG)	Up to 250 MHz Division ratio: 1/2/3/4/6/8/12/16/24/32/64
Peripheral module clock E (PCLKE)	MOSC/SOSC/HOCO/ MOCO/PLL1P	1 GHz	Peripheral module (CANFDRAM)	Up to 250 MHz Division ratio: 1/2/3/4/6/8/12/16/24/32/64
External bus clock (BCLK)	MOSC/SOSC/HOCO/ MOCO/PLL1P	1 GHz	External bus	Up to 125 MHz Division ratio: 1/2/3/4/6/8/12/16/24/32/64
EBCLK pin output (EBCLK)	BCLK or 1/2 BCLK or BCLKA or 1/2 BCLKA	—	EBCLK pin	BCLK: Up to 60 MHz Division ratio: 1 or 2 BCLKA: Up to 60 MHz Division ratio: 1 or 2
SDCLK pin output (SDCLK)	BCLK or BCLKA	—	SDCLK pin	BCLK: Up to 125 MHz BCLKA: Up to 133MHz
Trace clock (TRCLK)	MOSC/SOSC/HOCO/ MOCO/PLL1P	1 GHz	CPU-OCD	Up to 125 MHz Division ratio: 1/2/3/4/6/8/12/16/24/32/64/128/256
SCI clock (SCICLK)	MOSC/SOSC/HOCO/ MOCO/LOCO/PLL1P/ PLL1Q/PLL1R/PLL2P/ PLL2Q/PLL2R	480 MHz	SCI	Up to 120 MHz Division ratio: 1/2/3/4/5/6/8/10/16/32

Table 9.2 Clock generation circuit specifications for the internal clocks (2 of 3)

Item	Clock source	Maximum Frequency of Clock Source	Clock supply	Specification
SPI clock (SPICLK)	MOSC/SOSC/HOCO/ MOCO/ LOCO/PLL1P/ PLL1Q/PLL1R/PLL2P/ PLL2Q/PLL2R	333 MHz	SPI	Up to 333.33 MHz Division ratio: 1/2/3/4/5/6/8/10/16/32
Octal-SPI clock (OCTACLK)	MOSC/SOSC/HOCO/ MOCO/ LOCO/PLL1P/ PLL1Q/PLL1R/PLL2P/ PLL2Q/PLL2R	333 MHz	Octal-SPI	Up to 333.33 MHz Division ratio: 1/2/3/4/5/6/8/10/16/32
Octal-SPI divide clock (OCTADIVCLK)	1/2 OCTACLK	—	Octal-SPI	Up to 166.67 MHz
CANFD core clock (CANFDCLK)	MOSC/SOSC/HOCO/ MOCO/ LOCO/PLL1P/ PLL1Q/PLL1R/PLL2P/ PLL2Q/PLL2R	480 MHz	CANFD	Up to 80 MHz Division ratio: 1/2/3/4/5/6/8/10/16/32
ADC clock (ADCCLK)	MOSC/SOSC/HOCO/ MOCO/LOCO/PLL1P/ PLL1Q/PLL1R/PLL2P/ PLL2Q/PLL2R	480 MHz	ADC16H	Up to 120 MHz Division ratio: 1/2/3/4/5/6/8/10/16/32
GPT clock (GPTCLK)	MOSC/HOCO/MOCO/ PLL1P/PLL1Q/PLL1R/ PLL2P/PLL2Q/PLL2R	600 MHz	GPT, PDG, ADC16H	Up to 300 MHz Division ratio: 1/2/3/4/5/6/8/10/16/32
USB clock (USBCLK)	MOSC/HOCO/MOCO/ PLL1P/ PLL1Q/PLL1R/ PLL2P/PLL2Q/PLL2R	480 MHz	USBFS, USBHS	48 MHz Division ratio: 1/2/3/4/5/6/8/10/16/32
USB clock (USB60CLK)	MOSC/HOCO/MOCO/ PLL1P/ PLL1Q/PLL1R/ PLL2P/PLL2Q/PLL2R	480 MHz	USBHS	60 MHz Division ratio: 1/2/3/4/5/6/8/10/16/32
I3C clock (I3CCLK)	MOCO/PLL1P/PLL1Q/ PLL1R/ PLL2P/PLL2Q/ PLL2R	800 MHz	I3C	Up to 200 MHz Division ratio: 1/2/3/4/5/6/8/10/16/32
Asynchronous external bus clock (BCLKA)	MOCO/PLL1P/PLL1Q/ PLL1R/PLL2P/ PLL2Q/ PLL2R	800 MHz	External bus	Up to 133 MHz Division ratio: 1/2/3/4/5/6/8/10/16/32
EtherSW clock (ESWCLK)	MOCO/PLL1P/PLL1Q/ PLL1R/PLL2P/ PLL2Q/ PLL2R	1 GHz	ESWM	Up to 250 MHz Division ratio: 1/2/3/4/5/6/8/10/16/32
EtherSW-PHY clock (ESWPHYCLK)	MOCO/PLL1P/PLL1Q/ PLL1R/PLL2P/ PLL2Q/ PLL2R	1 GHz	ESWM	500 MHz Division ratio: 1/2/3/4/5/6/8/10/16/32
Ether-PHY clock (ETHPHYCLK)	MOSC/MOCO/PLL1P/ PLL1Q/PLL1R/PLL2P/ PLL2Q/PLL2R	800 MHz	ETHPHYCLK pin	25/50 MHz Division ratio: 1/2/3/4/5/6/8/10/16/32
Clock/buzzer output (CLKOUT)	MOSC/SOSC/HOCO/ MOCO/LOCO	—	CLKOUT pin	Up to 60 MHz Division ratio: 1/2/4/8/16/32/64/128
MOSC clock for USBHS (USBMCLK)	MOSC	—	USBHS	12/20/24/48 MHz No division
CAN clock (CANMCLK)	MOSC	—	CANFD	8 MHz to 48 MHz No division
ULPT LOCO clock (ULPTLCLK)	LOCO	—	ULPT	32.768 kHz No division
ULPT Sub-clock (ULPTSCLK)	SOSC	—	ULPT	32.768 kHz No division
AGT LOCO clock (AGTLCLK)	LOCO	—	AGT	32.768 kHz No division
AGT Sub clock (AGTSCLK)	SOSC	—	AGT	32.768 kHz No division

Table 9.2 Clock generation circuit specifications for the internal clocks (3 of 3)

Item	Clock source	Maximum Frequency of Clock Source	Clock supply	Specification
PDMIF clock (PDMIFCLK)	MOCO	—	PDMIF	8 MHz No division
CAC Main clock (CACMCLK)	MOSC	—	CAC	Up to 48 MHz No division
CAC Sub clock (CACSCLK)	SOSC	—	CAC	32.768 kHz No division
CAC HOCO clock (CACHCLK)	HOCO	—	CAC	16/18/20/32/48 MHz No division
CAC MOCO clock (CACMOCLK)	MOCO	—	CAC	8 MHz No division
CAC LOCO clock (CACLCLK)	LOCO	—	CAC	32.768 kHz No division
RTC LOCO clock (RTCLCLK)	LOCO	—	RTC	32.768 kHz No division
RTC Sub clock (RTCSCLK)	SOCO	—	RTC	32.768 kHz No division
IWDT clock (IWDTCLK)	1/2 LOCO	—	IWDT	16.384 kHz No division
SysTick timer clock (SYSTICKCLK)	1/8 MOCO	—	SysTick timer	1 MHz No division
JTAG clock (JTAGTCK)	TCK	—	JTAG	Up to 25 MHz No division
Serial wire clock (SWCLK)	TCK	—	OCD	Up to 25 MHz No division
TCLK pin output (TCLK)	1/2 TRCLK	—	TCLK pin	Up to 60 MHz

Note: When selecting PLL as the clock source, set the PLL output frequency to 1 GHz or less.

Note: If the PLL reference clock source is HOCO, the PLL multiplication setting must be set within output frequency range of PLL in consideration of HOCO frequency (minimum/maximum).

Note: Restrictions on setting clock frequency: CPUCLK0 ≥ ICLK, CPUCLK1 ≥ ICLK, MRICLK ≥ ICLK, CPUCLK0 ≥ MRICLK, CPUCLK1 ≥ MRICLK, ICLK ≥ MRPCLK, ICLK ≥ PCLKA ≥ PCLKB, ICLK ≥ BCLK, GPTCLK > PCLKA, PCLKD ≥ PCLKA ≥ PCLKB
Restrictions on clock frequency ratio: (N: integer, and up to 64)
CPUCLK0 : ICLK = N : 1, CPUCLK1 : ICLK = N : 1, MRICLK : ICLK = N : 1, CPUCLK0 : MRICLK = N : 1, CPUCLK1 : MRICLK = N : 1, ICLK : MRPCLK = N : 1, ICLK : BCLK = N : 1, ICLK : PCLKA = N : 1, ICLK : PCLKB = N : 1, ICLK : PCLKC = N : 1 or 1 : N, ICLK : PCLKD = N : 1 or 1 : N, ICLK : PCLKE = N : 1 or 1 : N
When one clock selects 3, 6, 12, 24-divisions, the other clocks set by SCKDIVCR and SCKDIVCR2 registers must not select 2, 4, 8, 16, 32, 64- divisions.
If the CAN-FD is used, clock frequency ratio is constrained to be PCLKA:PCLKE = 1:2.

Note: The multiplication of PLL1 and PLL2 should be set to be within the output frequency range of PLL1 and PLL2, taking the frequency of HOCO into consideration when not using the FLL function.

Note: Clocks have a permissible frequency range (See Table 9.2).
MRAM and SRAM also have a permissible operating frequency range in each wait cycle setting. (See section 58, SRAM, section 59, MRAM)
Those clock frequency ranges must be satisfied even if the HOCO has its maximum or minimum frequency when not using FLL function. (See section 62, Electrical Characteristics).

Note: PLLCCR.PLSRCSEL bit must be set to 0 when CPUCLK0 is set to over 960 MHz.

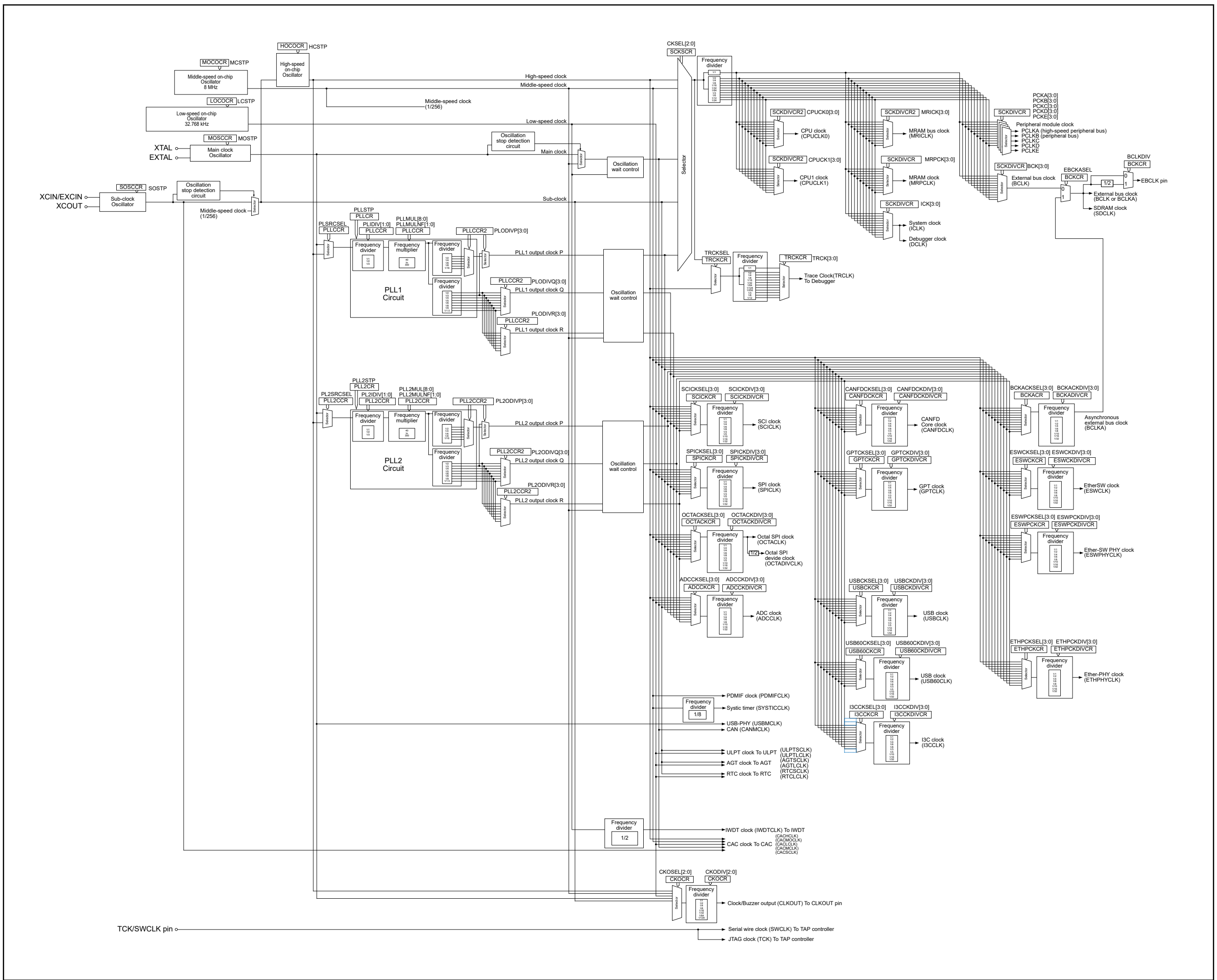


Figure 9.1 Clock generation circuit block diagram

Table 9.3 lists the input/output pins of the clock generation circuit.

Table 9.3 Input/Output Pins of Clock Generation Circuit

Pin name	I/O	Description
XTAL	Output	These pins are used to connect a ceramic resonator or crystal resonator. The EXTAL pin can also be used to input an external clock. For details, see section 9.3.2. External Clock Input.
EXTAL	Input	
XCIN/EXCIN	Input	These pins are used to connect a 32.768-kHz crystal resonator. The XCIN pin can also be used to input an external clock. For details, section 9.4.2. External Clock Input.
XCOUT	Output	
TCK/SWCLK	Input	This pin is used to input the clock for the JTAG/SWD
EBCLK	Output	This pin is used to supply external devices with the external bus clock (EBCLK)
SDCLK	Output	SDRAM clock (SDCLK) supply for external devices
CLKOUT	Output	This pin is used to output the CLKOUT/BUZZER clock
TRCLK	Output	TRACE clock output.

9.2 Register Descriptions

9.2.1 CGFSAR : Clock Generation Function Security Attribute Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x3C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	NONSEC28	NONSEC27	—	NONSEC25	NONSEC24	—	NONSEC22	NONSEC21	NONSEC20	NONSEC19	NONSEC18	NONSEC17	NONSEC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	NONSEC14	NONSEC13	NONSEC12	NONSEC11	NONSEC10	NONSEC09	NONSEC08	NONSEC07	NONSEC06	NONSEC05	NONSEC04	NONSEC03	NONSEC02	—	NONSEC00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NONSEC00	Non-secure Attribute bit 00 Target register: SCKDIVCR, SCKDIVCR2, SCKSCR Target factor: system clock control 0: Secure 1: Non-secure	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	NONSEC02	Non-secure Attribute bit 02 Target register: HOCOCR, HOCOCR2, FLLCR1, FLLCR2, HOCOUTCR, HOCOSCR, USBCKSELR Target factor: HOCO 0: Secure 1: Non-secure	R/W
3	NONSEC03	Non-secure Attribute bit 03 Target register: MOCOCR, MOCOUTCR, MOCOSCR Target factor: MOCO 0: Secure 1: Non-secure	R/W
4	NONSEC04	Non-secure Attribute bit 04 Target register: LOCOCR, LOCOUTCR Target factor: LOCO 0: Secure 1: Non-secure	R/W
5	NONSEC05	Non-secure Attribute bit 05 Target register: MOSCCR, MOSCWTCR, MOMCR, MOMCR2, MOSCSCR Target factor: MOSC 0: Secure 1: Non-secure	R/W
6	NONSEC06	Non-secure Attribute bit 06 Target register: OSTDCR, OSTDSR Target factor: oscillation stop detection control 0: Secure 1: Non-secure	R/W
7	NONSEC07	Non-secure Attribute bit 07 Target register: SOSCCR, SOMCR Target factor: SOSC 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
8	NONSEC08	Non-secure Attribute bit 08 Target register: PLLCCR, PLLCCR2, PLLCR Target factor: PLL 0: Secure 1: Non-secure	R/W
9	NONSEC09	Non-secure Attribute bit 09 Target register: PLL2CCR, PLL2CCR2, PLL2CR Target factor: PLL2 0: Secure 1: Non-secure	R/W
10	NONSEC10	Non-secure Attribute bit 10 Target register:SOSTDCR, SOSTDSR Target factor: Sub oscillation stop detection control 0: Secure 1: Non-secure	R/W
11	NONSEC11	Non-secure Attribute bit 11 Target register: CKOCR Target factor: CLKOUT control 0: Secure 1: Non-secure	R/W
12	NONSEC12	Non-secure Attribute bit 12 Target register: BCKCR, EBCKOCR Target factor: EBCLK 0: Secure 1: Non-secure	R/W
13	NONSEC13	Non-secure Attribute bit 13 Target register: SDCKOCR Target factor: SDCLK 0: Secure 1: Non-secure	R/W
14	NONSEC14	Non-secure Attribute bit 14 Target register: BCKADIVCR, BCKACR Target factor: BCLKA 0: Secure 1: Non-secure	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
16	NONSEC16	Non-secure Attribute bit 16 Target register: USBCKDIVCR, USBCKCR Target factor: USBCLK 0: Secure 1: Non-secure	R/W
17	NONSEC17	Non-secure Attribute bit 17 Target register: OCTACKDIVCR, OCTACKCR Target factor: OCTACLK 0: Secure 1: Non-secure	R/W
18	NONSEC18	Non-secure Attribute bit 18 Target register: CANFDCKDIVCR, CANFDCKCR Target factor: CANFDCLK 0: Secure 1: Non-secure	R/W
19	NONSEC19	Non-secure Attribute bit 19 Target register: USB60CKDIVCR, USB60CKCR Target factor: USB60CLK 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
20	NONSEC20	Non-secure Attribute bit 20 Target register:I3CCKDIVCR, I3CCKCR Target factor: I3CCLK 0: Secure 1: Non-secure	R/W
21	NONSEC21	Non-secure Attribute bit 21 Target register: SCICKDIVCR, SCICKCR Target factor: SCICLK 0: Secure 1: Non-secure	R/W
22	NONSEC22	Non-secure Attribute bit 22 Target register: SPICKDIVCR, SPICKCR Target factor: SPICLK 0: Secure 1: Non-secure	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
24	NONSEC24	Non-secure Attribute bit 24 Target register: ADCCKDIVCR, ADCCKCR Target factor: ADCCLK 0: Secure 1: Non-secure	R/W
25	NONSEC25	Non-secure Attribute bit 25 Target register: GPTCKDIVCR, GPTCKCR Target factor: GPTCLK 0: Secure 1: Non-secure	R/W
26	—	This bit is read as 0. The write value should be 0.	R/W
27	NONSEC27	Non-secure Attribute bit 27 Target register: ESWCKDIVCR, ESWCKCR, ESWPCKDIVCR, ESWPCKCR Target factor: ESWCLK, ESWPHYCLK 0: Secure 1: Non-secure	R/W
28	NONSEC28	Non-secure Attribute bit 28 Target register: ETHPCKDIVCR, ETHPCKCR Target factor: ETHPHYCLK 0: Secure 1: Non-secure	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1
Note: Set the PRCR.PRC4 bit to 1 (write enabled) before rewriting this register.

CGFSAR register controls the secure attribute of Clock Generation Function registers.

NONSEC00 bit (Non-secure Attribute bit 00)

This bit controls the security attribute of SCKDIVCR, SCKDIVCR2, SCKSCR.

NONSEC02 bit (Non-secure Attribute bit 02)

This bit controls the security attribute of HOCOCR, HOCOCR2, FLLCR1, FLLCR2, HOCOUTCR, HOCOSCR, USBCKSELR.

NONSEC03 bit (Non-secure Attribute bit 03)

This bit controls the security attribute of MOCOCR, MOCOUTCR, MOCOSCR.

NONSEC04 bit (Non-secure Attribute bit 04)

This bit controls the security attribute of LOCOCR, LOCOUTCR.

NONSEC05 bit (Non-secure Attribute bit 05)

This bit controls the security attribute of MOSCCR, MOSCWTCR, MOMCR, MOMCR2, MOSCSCR.

NONSEC06 bit (Non-secure Attribute bit 06)

This bit controls the security attribute of OSTDCR, OSTDSR.

NONSEC07 bit (Non-secure Attribute bit 07)

This bit controls the security attribute of SOSCCR, SOMCR.

NONSEC08 bit (Non-secure Attribute bit 08)

This bit controls the security attribute of PLLCCR, PLLCCR2, PLLCR.

NONSEC09 bit (Non-secure Attribute bit 09)

This bit controls the security attribute of PLL2CCR, PLL2CCR2, PLL2CR.

NONSEC10 bit (Non-secure Attribute bit 10)

This bit controls the security attribute of SOSTDCR, SOSTDSR.

NONSEC11 bit (Non-secure Attribute bit 11)

This bit controls the security attribute of CKOCR.

NONSEC12 bit (Non-secure Attribute bit 12)

This bit controls the security attribute of BCKCR, EBCKOCR.

NONSEC13 bit (Non-secure Attribute bit 13)

This bit controls the security attribute of SDCKOCR.

NONSEC14 bit (Non-secure Attribute bit 14)

This bit controls the security attribute of BCKADIVCR, BCKACR.

NONSEC16 bit (Non-secure Attribute bit 16)

This bit controls the security attribute of USBCKDIVCR, USBCKCR.

NONSEC17 bit (Non-secure Attribute bit 17)

This bit controls the security attribute of OCTACKDIVCR, OCTACKCR.

NONSEC18 bit (Non-secure Attribute bit 18)

This bit controls the security attribute of CANFDCKDIVCR, CANFDCKCR.

NONSEC19 bit (Non-secure Attribute bit 19)

This bit controls the security attribute of USB60CKDIVCR, USB60CKCR.

NONSEC20 bit (Non-secure Attribute bit 20)

This bit controls the security attribute of I3CCKDIVCR, I3CCKCR.

NONSEC21 bit (Non-secure Attribute bit 21)

This bit controls the security attribute of SCICKDIVCR, SCICKCR.

NONSEC22 bit (Non-secure Attribute bit 22)

This bit controls the security attribute of SPICKDIVCR, SPICKCR.

NONSEC24 bit (Non-secure Attribute bit 24)

This bit controls the security attribute of ADCCKDIVCR, ADCCKCR.

NONSEC25 bit (Non-secure Attribute bit 25)

This bit controls the security attribute of GPTCKDIVCR, GPTCKCR.

NONSEC27 bit (Non-secure Attribute bit 27)

This bit controls the security attribute of ESWCKDIVCR, ESWCKCR, ESWPCKDIVCR, ESWPCKCR

NONSEC28 bit (Non-secure Attribute bit 28)

This bit controls the security attribute of ETHPCKDIVCR, ETHPCKCR.

9.2.2 SCKDIVCR : System Clock Division Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x020

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

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

Bit	Symbol	Function	R/W
3:0	PCKD[3:0]*4	Peripheral Module Clock D (PCLKD) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 1 0 1 1: × 1/24 Others: Setting prohibited.	R/W
7:4	PCKC[3:0]*4	Peripheral Module Clock C (PCLKC) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 1 0 1 1: × 1/24 Others: Setting prohibited.	R/W
11:8	PCKB[3:0]*3	Peripheral Module Clock B (PCLKB) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 1 0 1 1: × 1/24 Others: Setting prohibited.	R/W

Bit	Symbol	Function	R/W
15:12	PCKA[3:0]*3	Peripheral Module Clock A (PCLKA) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 1 0 1 1: × 1/24 Others: Setting prohibited.	R/W
19:16	BCK[3:0]*2	External Bus Clock (BCLK) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 1 0 1 1: × 1/24 Others: Settings prohibited	R/W
23:20	PCKE[3:0]*4	Peripheral Module Clock E (PCLKE) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 1 0 1 1: × 1/24 Others: Settings prohibited	R/W
27:24	ICK[3:0]*1*2*3*4*5*6*7	System Clock (ICLK) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 1 0 1 1: × 1/24 Others: Setting prohibited.	R/W

Bit	Symbol	Function	R/W
31:28	MRPCK[3:0][*1*8]	MRAM Clock (MRPCLK) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 1 0 1 1: × 1/24 Others: Setting prohibited.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note: When one clock selects 3, 6, 12, 24-divisions, the other clocks must not select 2, 4, 8, 16, 32, 64- divisions.
Note 1. The following relation is required between the frequencies of the system clock (ICLK) and the MRAM Clock (MRPCLK).
ICLK:MRPCLK = N:1 (N: integer)
Note 2. The following relation is required between the frequencies of the system clock (ICLK) and the external bus clock (BCLK).
ICLK:BCLK = N:1 (N: integer)
Note 3. The following relation is required between the frequencies of the system clock (ICLK) and the peripheral module clocks (PCLKA, PCLKB)
ICLK:PCLKA = N:1, ICLK:PCLKB = N:1 (N: integer)
Note 4. The following relation is required between the frequencies of the system clock (ICLK) and the peripheral module clocks (PCLKC, PCLKD, PCLKE):
ICLK:PCLKC, PCLKD PCLKE = N:1 or 1:N (N: integer)
Note 5. The following relation is required between the frequencies of the CPU0 clock (CPUCLK0) and the system clock (ICLK):
CPUCLK0:ICLK = N:1 (N: integer)
Note 6. The following relation is required between the frequencies of the CPU1 clock (CPUCLK1) and the system clock (ICLK):
CPUCLK1:ICLK = N:1 (N: integer)
Note 7. The following relation is required between the frequencies of the MRAM bus clock (MRICLK), the CPU0 clock (CPUCLK0), the CPU1 clock (CPUCLK1) and the system clock (ICLK):
MRICLK:ICLK = N:1 (N: integer)
CPUCLK0:MRICLK = N:1 (N: integer)
CPUCLK1:MRICLK = N:1 (N: integer)
Note 8. The frequency of the MRAM clock (MRPCLK) is limited to the MRE frequency notifications register (MREFREQ). See section 59, MRAM.

SCKDIVCR selects the frequencies of the system clock (ICLK), peripheral module clock (PCLKA, PCLKB, PCLKC, PCLKD, PCLKE), MRAM Clock (MRPCLK), and external bus clock (BCLK).

9.2.3 SCKDIVCR2 : System Clock Division Control Register 2

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x024

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MRICK[3:0]				—				CPUCK1[3:0]				CPUCK0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CPUCK0[3:0]*1	CPU0 Clock (CPUCLK0) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 1 0 1 1: × 1/24 Others: Setting prohibited.	R/W
7:4	CPUCK1[3:0]*2*5	CPU1 Clock (CPUCLK1) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 1 0 1 1: × 1/24 Others: Setting prohibited.	R/W
11:8	—	Set these bits to the same value as MRICK[3:0]	R/W
15:12	MRICK[3:0]*3*4	MRAM bus Clock (MRICLK) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 1 0 1 1: × 1/24 Others: Setting prohibited.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: When one clock selects 3, 6, 12, 24-divisions, the other clocks must not select 2, 4, 8, 16, 32, 64- divisions.
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. The following relation is required between the frequencies of the CPU0 clock (CPUCLK0) and the system clock (ICLK): CPUCLK0:ICLK = N:1 (N: integer)
Note 2. The following relation is required between the frequencies of the CPU1 clock (CPUCLK1) and the system clock (ICLK):CPUCLK1:ICLK = N:1 (N: integer)
Note 3. The following relation is required between the frequencies of the MRAM bus clock (MRICLK), the CPU0 clock (CPUCLK0), the CPU1 clock (CPUCLK1)and the system clock (ICLK):
MRICLK:ICLK = N:1 (N: integer)
CPUCLK0:MRICLK = N:1 (N: integer)
CPUCLK1:MRICLK = N:1 (N: integer)
Note 4. The frequency of the MRAM bus clock (MRICLK) is limited to the MRC frequency notifications register (MRCFREQ). See section 59, MRAM.
Note 5. Set the same value as MRICK[3:0] for single core product.

SCKDIVCR2 selects the frequencies of the CPU0 clock (CPUCLK0), the CPU1 clock (CPUCLK1) and the MRAM bus clock (MRICLK).

9.2.4 SYRACCR : System Register Access Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xCC

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

Bit	Symbol	Function	R/W
0	BUSY	Access Ready monitor 0: Ready to read/write access 1: Writing in progress	R
7:1	—	These bits are read as 0. The write value should be 0.	R

Note: S-TYPE-5, P-TYPE-5
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

SYRACCR register monitor that the target registers can be accessed.

The target registers are LOCOCR and LOCOUTCR.

BUSY bit (Access Ready monitor)

When SYRACCR.BUSY is 0, you can access the target registers.

Make sure that SYRACCR.BUSY is 0 before accessing the target registers.

When SYRACCR.BUSY is 1, the write access to one of the target registers is in progress. If SYRACCR.BUSY is 1, further writing to one of the target registers is ignored, and the read value from the target registers is not guaranteed.

9.2.5 SCKSCR : System Clock Source Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x026

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

Bit	Symbol	Function	R/W
2:0	CKSEL[2:0]	Clock Source Select 0 0 0: HOCO 0 0 1: MOCO (Value after reset) 0 1 0: Setting prohibited 0 1 1: Main clock oscillator (MOSC) 1 0 0: Sub-clock oscillator (SOSC) 1 0 1: PLL1 output clock P (PLL1P) 1 1 0: Setting prohibited 1 1 1: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note: S-TYPE-3, P-TYPE-2

The SCKSCR register selects the clock source for the system clock.

CKSEL[2:0] bits (Clock Source Select)

The CKSEL[2:0] bits select the source for the following modules:

- CPU0 clock (CPUCLK0)
- CPU1 clock (CPUCLK1)
- MRAM bus clock (MRICLK)
- MRAM clock (MRPCLK)
- System clock (ICLK)
- Peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD and PCLKE)
- External bus clock (BCLK)
- SDRAM clock (SDCLK)

The bits select from one of the following sources:

- Middle-speed on-chip oscillator (MOCO)
- High-speed on-chip oscillator (HOCO)
- Main clock oscillator (MOSC)
- Sub-clock oscillator (SOSC)
- PLL1 output clock P (PLL1P)

The operating state of each clock source is controlled not only by the clock oscillation enable settings but also by the operating modes of the product. Some clock sources might be forcibly stopped depending on the product operating mode being used.

Check the operation state of clock sources in each product operating mode, and do not select the clock source to be stopped in SCKSCR. The clock sources should be switched when there are no occurring internal asynchronous interrupt. For details, see section 11, Low Power Mode.

9.2.6 PLLCCR : PLL Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x0AC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLLMUL[8]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
1:0	PLIDIV[1:0]*1	PLL1 Input Frequency Division Ratio Select 0 0: 1/1 0 1: 1/2 1 0: 1/3 Others: Setting prohibited.	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	PLSRCSEL	PLL1 Clock Source Select 0: Main clock oscillator*3 1: HOCO*4	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7:6	PLLMULNF[1:0]*2	PLL1 Frequency Multiplication Fractional Factor Select 0 0: 0.00 (Value after reset) 0 1: 0.33 (1/3) 1 0: 0.66 (2/3) 1 1: 0.50 (1/2)	R/W
16:8	PLLMUL[8:0]*2	PLL1 Frequency Multiplication Factor Select 0x27: × 40 (Value after reset) 0x28: × 41 0x29: × 42 ⋮ 0x12A: × 299 0x12B: × 300 Others: Setting prohibited.	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Note 1. PLIDIV[1:0] should be set so that the frequency of PLL1 input signal is within the range of section 9.1. Overview.

Note 2. PLLMUL[8:0] and PLLMULNF[1:0] should be set so that the frequency of PLL1 output signal is within the range of section 9.1. Overview.

Note 3. PLSRCSEL must be set to 0 when CPUCLK0 is set to over 960MHz.

Note 4. The FLL function must be enabled when using USBCLK.

The PLLCCR register sets the operation of the PLL1 circuit.

Writing to the PLLCCR is prohibited when the PLLCR.PLLSTP bit is 0 (the PLL1 operates).

PLIDIV[1:0] bits (PLL1 Input Frequency Division Ratio Select)

These bits select the frequency division ratio of the PLL1 clock source.

PLSRCSEL bit (PLL1 Clock Source Select)

This bit selects the clock source for the PLL1.

PLLMULNF[1:0] bits (PLL1 Frequency Multiplication Fractional Factor Select)

These bits select the fractional part of the frequency multiplication factor for the PLL1 circuit.

PLLMUL[8:0] bits (PLL1 Frequency Multiplication Factor Select)

These bits select the frequency multiplication factor of the PLL1 circuit.

9.2.7 PLLCCR2 : PLL Clock Control Register 2

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x4C

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

Bit	Symbol	Function	R/W
3:0	PLODIVP[3:0]*1	PLL1 Output Frequency Division Ratio Select for output clock P 0001: × 1/2 0010: × 1/3 0011: × 1/4 0101: × 1/6 (Value after reset) 0111: × 1/8 1111: × 1/16 Others: Setting prohibited.	R/W

Bit	Symbol	Function	R/W
7:4	PLODIVQ[3:0]*1	PLL1 Output Frequency Division Ratio Select for output clock Q 0001: × 1/2 0010: × 1/3 0011: × 1/4 0100: × 1/5 0101: × 1/6 (Value after reset) 0111: × 1/8 1000: × 1/9 1001: × 1/1.5 Others: Setting prohibited.	R/W
11:8	PLODIVR[3:0]*1	PLL1 Output Frequency Division Ratio Select for output clock R 0001: × 1/2 0010: × 1/3 0011: × 1/4 0100: × 1/5 0101: × 1/6 (Value after reset) 0111: × 1/8 1000: × 1/9 1001: × 1/1.5 Others: Setting prohibited.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. It must be set so that the frequency of PLL1 output signal is within the range listed in Table 9.1.

The PLLCCR2 register sets up the operation of the PLL1 circuit. Writing to the PLLCCR2 is prohibited when the PLLCR.PLLSTP bit is 0 (the PLL1 operates).

Even if only one of the PLL1 output clock P, Q, or R is used, PLLCCR2 must be set so that the frequency of PLL1 output signal is within the range listed in Table 9.1.

PLODIVP[3:0] bit (PLL1 Output Frequency Division Ratio Select for output clock P)

These bits select the output frequency division ratio for PLL1 output clock P (PLL1P).

PLODIVQ[3:0] bit (PLL1 Output Frequency Division Ratio Select for output clock Q)

These bits select the output frequency division ratio for PLL1 output clock Q (PLL1Q).

PLODIVR[3:0] bit (PLL1 Output Frequency Division Ratio Select for output clock R)

These bits select the output frequency division ratio for PLL1 output clock R (PLL1R).

9.2.8 PLLCR : PLL Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x02A

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

Bit	Symbol	Function	R/W
0	PLLSTP	PLL1 Stop Control 0: PLL1 is operating 1: PLL1 is stopped.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The PLLCR register controls the operation of the PLL1 circuit.

PLLSTP bit (PLL1 Stop Control)

This bit runs or stops the PLL1 circuit.

If the main clock oscillator is to be selected as the clock source for the PLL1 by the PLLCCR.PLSRCSEL bit, the Main Clock Oscillator Wait Control Register (MOSCWTCR) must be set.

After the PLLSTP bit setting is changed to run the PLL1, only use the PLL1 clock after confirming that the OSCSF.PLLSF bit is set to 1. That is, a fixed time for stabilization is required after starting the PLL1 operation. A fixed time is also required for oscillation to stop after stopping the PLL1 operation. Additionally, apply the following limitations when starting and stopping the PLL1 operation by the PLLSTP bit:

- After stopping the PLL1, confirm that the OSCSF.PLLSF bit is 0 before restarting the PLL1.
- Confirm that the PLL1 is operating and that the OSCSF.PLLSF bit is 1 before stopping the PLL1.
- Regardless of whether the PLL1 clock is selected as the system clock source, confirm that the OSCSF.PLLSF is set to 1 before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode after operating the PLL1.
- When transitioning to Software Standby or Deep Software Standby mode after stopping the PLL1, confirm that the OSCSF.PLLSF bit is cleared to 0 before executing a WFI instruction.

Writing 1 to the PLLSTP bit is prohibited when SCKSCR.CKSEL[2:0] = 101 (system clock source = PLL1 output clock P (PLL1P)).

Confirm the following conditions before writing 0 to PLLSTP:

- When PLL1 source clock = MOSC: MOSCCR.MOSTP = 0 (MOSC is enabled)
- When PLL1 source clock = HOCO: HOCOCR.HCSTP = 0 (HOCO is enabled).

9.2.9 PLL2CCR : PLL2 Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x0C8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL2MUL[8]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PLL2MUL[7:0]								PLL2MULNF[1:0]		—	PL2SRCSEL	—		PL2IDIV[1:0]	
Value after reset:	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PL2IDIV[1:0][*1]	PLL2 Input Frequency Division Ratio Select 0 0: 1/1 (Value after reset) 0 1: 1/2 1 0: 1/3 Others: Setting prohibited.	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	PL2SRCSEL	PLL2 Clock Source Select 0: Main clock oscillator 1: HOCO[*3]	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
7:6	PLL2MULNF[1:0][*2]	PLL2 Frequency Multiplication Fractional Factor Select 00: 0.00 (Value after reset) 01: 0.33 (1/3) 10: 0.66 (2/3) 11: 0.50 (1/2)	R/W

Bit	Symbol	Function	R/W
16:8	PLL2MUL[8:0]*2	PLL2 Frequency Multiplication Factor Select 0x27: × 40 (Value after reset) 0x28: × 41 0x29: × 42 ⋮ 0x12A: × 299 0x12B: × 300 Others: Setting prohibited.	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. PL2IDIV[1:0] should be set so that the frequency of PLL2 input signal is within the range of section 9.1. Overview.
Note 2. PLL2MUL[8:0] and PLL2MULNF[1:0] should be set so that the frequency of PLL2 output signal is within the range of section 9.1. Overview.
Note 3. The FLL function must be enabled when using USBCLK.

The PLL2CCR register sets the operation of the PLL2 circuit.

Writing to the PLL2CCR register is prohibited when the PLL2CR.PLL2STP bit is 0 (the PLL2 operates).

PL2IDIV[1:0] bits (PLL2 Input Frequency Division Ratio Select)

These bits select the frequency division ratio of the PLL2 clock source.

PL2SRCSEL bit (PLL2 Clock Source Select)

This bit selects the clock source for the PLL2.

PLL2MULNF[1:0] bit (PLL2 Frequency Multiplication Fractional Factor Select)

These bits select the fractional part of the frequency multiplication factor for the PLL2 circuit.

PLL2MUL[8:0] bits (PLL2 Frequency Multiplication Factor Select)

These bits select the frequency multiplication factor of the PLL2 circuit.

9.2.10 PLL2CCR2 : PLL2 Clock Control Register 2

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x4E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—				PL2ODIVR[3:0]				PL2ODIVQ[3:0]				PL2ODIVP[3:0]			
Value after reset:	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1

Bit	Symbol	Function	R/W
3:0	PL2ODIVP[3:0]*1	PLL2 Output Frequency Division Ratio Select for output clock P 0001: × 1/2 0010: × 1/3 0011: × 1/4 0101: × 1/6 (Value after reset) 0111: × 1/8 1111: × 1/16 Others: Setting prohibited.	R/W

Bit	Symbol	Function	R/W
7:4	PL2ODIVQ[3:0]*1	PLL2 Output Frequency Division Ratio Select for output clock Q 0001: × 1/2 0010: × 1/3 0011: × 1/4 0100: × 1/5 0101: × 1/6 (Value after reset) 0111: × 1/8 1000: × 1/9 1001: × 1/1.5 Others: Setting prohibited.	R/W
11:8	PL2ODIVR[3:0]*1	PLL2 Output Frequency Division Ratio Select for output clock R 0001: × 1/2 0010: × 1/3 0011: × 1/4 0100: × 1/5 0101: × 1/6 (Value after reset) 0111: × 1/8 1000: × 1/9 1001: × 1/1.5 Others: Setting prohibited.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. It must be set so that the frequency of PLL2 output signal is within the range listed in Table 9.1.

The PLL2CCR2 register sets up the operation of the PLL2 circuit. Writing to the PLL2CCR2 is prohibited when the PLL2CR.PLL2STP bit is 0 (the PLL2 operates).

Even if only one of the PLL2 output clock P, Q, or R is used, PLL2CCR2 must be set so that the frequency of PLL2 output signal is within the range listed in Table 9.1.

PL2ODIVP[3:0] bit (PLL2 Output Frequency Division Ratio Select for output clock P)

These bits select the output frequency division ratio for PLL2 output clock P (PLL2P).

PL2ODIVQ[3:0] bit (PLL2 Output Frequency Division Ratio Select for output clock Q)

These bits select the output frequency division ratio for PLL2 output clock Q (PLL2Q).

PL2ODIVR[3:0] bit (PLL2 Output Frequency Division Ratio Select for output clock R)

These bits select the output frequency division ratio for PLL2 output clock R (PLL2R).

9.2.11 PLL2CR : PLL2 Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x04A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	PLL2STP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	PLL2STP	PLL2 Stop Control 0: PLL2 is operating 1: PLL2 is stopped.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The PLL2CR register controls the operation of the PLL2 circuit.

PLL2STP bit (PLL2 Stop Control)

This bit runs or stops the PLL2 circuit.

If the main clock oscillator is to be selected as the clock source for the PLL2 by the PLL2CCR.PL2SRCSEL bit, the Main Clock Oscillator Wait Control Register (MOSCWTCR) must be set.

After the PLL2STP bit setting is changed to run the PLL2, only use the PLL2 clock after confirming that the OSCSF.PLL2SF bit is set to 1. That is, a fixed time for stabilization is required after starting the PLL2 operation. A fixed time is also required for oscillation to stop after stopping the PLL2 operation. Additionally, apply the following limitations when starting and stopping the PLL2 operation by the PLL2STP bit:

- After stopping the PLL2, confirm that the OSCSF.PLL2SF bit is 0 before restarting the PLL2.
- Confirm that the PLL2 is operating and that the OSCSF.PLL2SF bit is 1 before stopping the PLL2.
- Confirm that the OSCSF.PLL2SF bit is set to 1 before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode after operating the PLL2.
- When transitioning to Software Standby or Deep Software Standby mode after stopping the PLL2, confirm that the OSCSF.PLL2SF bit is cleared to 0 before executing a WFI instruction.

Confirm the following conditions before writing 0 to PLL2STP:

- When the PLL2 source clock = MOSC: MOSCCR.MOSTP = 0 (MOSC is enabled)
- When the PLL2 source clock = HOCO: HOCOCR.HCSTP = 0 (HOCO is enabled).

9.2.12 BCKCR : External Bus Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x030

Bit position:	7	6	5	4	3	2	1	0
Bit field:	EBCK ASEL	—	—	—	—	—	—	BCLK DIV
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BCLKDIV	BCLK Pin Output Select 0: BCLK or BCLKA 1: BCLK/2 or BCLKA/2	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	EBCKASEL[*1]	External Bus Asynchronous Select 0: Synchronous (BCLK) 1: Asynchronous (BCLKA)	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. Before changing this bit, set BCKACR.BCKACKSREQ = 1 and then change the value while BCKACR.BCKACKSRDY = 1.

The BCKCR register controls the external bus clock.

This register must be selected with BCKACR. For detail, see section 9.2.62. BCKACR : Asynchronous External Bus Clock Control Register.

BCLKDIV bit (BCLK Pin Output Select)

This bit selects the clock signal for output from the BCLK pin.

Either the BCLK clock with the frequency selected by the BCK[2:0] bits in SCKDIVCR or the BCLK clock divided by 2 can be selected.

EBCKASEL bit (External Bus Asynchronous Select)

This bit selects the clock signal for External Bus, EBCLK and SDCLK.

9.2.13 MOSCCR : Main Clock Oscillator Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x032

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

Bit	Symbol	Function	R/W
0	MOSTP	Main Clock Oscillator Stop 0: Operate the main clock oscillator*1 1: Stop the main clock oscillator	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note: S-TYPE-3, P-TYPE-2
Note 1. MOMCR register must be set before setting MOSTP to 0.

The MOSCCR register controls the main clock oscillator.

MOSTP bit (Main Clock Oscillator Stop)

The MOSTP bit starts or stops the main clock oscillator.

When changing the value of the MOSTP bit, execute subsequent instructions only after reading the bit to check that the value is updated.

When using the main clock, the Main Clock Oscillator Mode Oscillation Control Register (MOMCR) and the Main Clock Oscillator Wait Control Register (MOSCWTCR) must be set before setting MOSTP to 0. After setting the MOSTP bit to 0, confirm that the OSCSF.MOSCSF bit is set to 1 before using the main clock oscillator.

A fixed stabilization wait time is required after setting the main clock oscillator to start operation. A fixed wait time is also required for oscillation to stop after stopping the main clock oscillator.

The following restrictions apply when starting and stopping operation:

- After stopping the main clock oscillator, confirm that the OSCSF.MOSCSF bit is 0 before restarting the main clock oscillator
- Confirm that the main clock oscillator operates and that the OSCSF.MOSCSF bit is 1 before stopping the main clock oscillator
- Regardless of whether the main clock oscillator is selected as the system clock source, confirm that the OSCSF.MOSCSF bit is set to 1 before executing a WFI instruction to place the MCU in Software standby after operating the main clock oscillator or Deep Software Standby mode.
- When a transition to Software Standby or Deep Software Standby mode is to follow the setting to stop the main clock oscillator, confirm that the OSCSF.MOSCSF bit is set to 0 before executing the WFI instruction.

Writing 1 to MOSTP is prohibited under the following condition:

- SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC).
- PLLCCR.PLSRCSEL = 0 (PLL1 source clock = MOSC) and SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL1P)
- PLLCCR.PLSRCSEL = 0 (PLL1 source clock = MOSC) and PLLCR.PLLSTP = 0 (PLL1 is operating)
- PLL2CCR.PL2SRCSEL = 0 (PLL2 source clock = MOSC) and PLL2CR.PLL2STP = 0 (PLL2 is operating)

9.2.14 SOSCCR : Sub-Clock Oscillator Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC00

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

Bit	Symbol	Function	R/W
0	SOSTP	Sub-Clock Oscillator Stop 0: Operate the sub-clock oscillator*1 1: Stop the sub-clock oscillator	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. The SOMCR register must be set before setting SOSTP to 0.

The SOSCCR register controls the sub-clock oscillator.

SOSTP bit (Sub-Clock Oscillator Stop)

The SOSTP bit starts or stops the sub-clock oscillator. When changing the value of the SOSTP bit, only execute subsequent instructions after reading the bit to check that the value is updated. Use the SOSTP bit when using the sub-clock oscillator as the source for a peripheral module, for example the RTC. When using the sub-clock oscillator, set the Sub-Clock Oscillator Mode Control Register (SOMCR) before setting SOSTP to 0.

When selecting External clock input by SOMCR.SOSEL, set SOMCR.SOSEL to 1 and wait 50us, then set SOSTP to 0.

The following restrictions apply when starting and stopping the operation:

- After stopping the sub-clock oscillator, allow a stop interval of at least 5 SOSC clock cycles before restarting it
- After setting the SOSTP bit to 0, use the sub-clock only after the sub-clock oscillation stabilization time ($t_{SUBOSCWT}$) has elapsed.
- Regardless of whether the sub-clock oscillator is selected as the system clock, confirm that the sub-clock oscillation is stable before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode
- When a transition to Software Standby or Deep Software Standby mode is to follow the setting to stop the sub-clock oscillator, wait for at least 3 SOSC clock cycles before executing the WFI instruction.

Writing 1 to SOSTP is prohibited under the following condition:

- SCKSCR.CKSEL[2:0] = 100b (system clock source = SOSC).

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

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x400

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

Bit	Symbol	Function	R/W
0	LCSTP	LOCO Stop 0: Operate the LOCO clock 1: Stop the LOCO clock	R/W

Bit	Symbol	Function	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note: Make sure that SYRACCR.BUSY is 0 before accessing LOCOCR. When SYRACCR.BUSY is 1, the write access is ignored and the read value is not guaranteed.

The LOCOCR register controls the LOCO clock.

LCSTP bit (LOCO Stop)

The LCSTP bit starts or stops the LOCO clock.

When changing the value of the bit, only execute subsequent instructions after reading the bit and checking that its value has actually been updated.

9.2.16 HOCOCR : High-Speed On-Chip Oscillator Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x036

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

Bit	Symbol	Function	R/W
0	HCSTP	HOCO Stop 0: Operate the HOCO clock *2 *3 1: Stop the HOCO clock	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. The HCSTP bit value after a reset is 0 when the OFS1(_SEC).HOCOEN bit is 0. It is 1 when the OFS1(_SEC).HOCOEN bit is 1.
Note 2. If you are using the HOCO (HCSTP = 0), set the OFS1(_SEC).HOCOFRQ0[2:0] bit to an optimum value.
Note 3. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] even if OFS1(_SEC).HOCOFRQ0[2:0] is not appropriate value.

The HOCOCR register controls the HOCO clock.

HCSTP bit (HOCO Stop)

The HCSTP bit starts or stops the HOCO clock.

After setting the HCSTP bit to 0 to start the HOCO clock, confirm that the OSCSF.HOCOSF is set to 1 before using the clock. When OFS1(_SEC).HOCOEN is set to 0, confirm that OSCSF.HOCOSF is also set to 1 before using the HOCO clock. A fixed stabilization wait time is required after setting the HOCO clock to start operation. A fixed wait time is also required after setting the HOCO clock to stop.

The following limitations apply when starting and stopping operation:

- After stopping the HOCO clock, wait more than HOCO stop width time and confirm that the OSCSF.HOCOSF is 0 before restarting the HOCO clock. See section 62, Electrical Characteristics for HOCO stop width time.
- Confirm that the HOCO clock operates and that the OSCSF.HOCOSF is 1 before stopping the HOCO clock.
- Regardless of whether the HOCO clock is selected as the system clock source, confirm that the OSCSF.HOCOSF is set to 1 before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode after setting HOCO operation with the HCSTP bit.
- When a transition to Software Standby or Deep Software Standby mode is to follow the setting of the HOCO clock to stop, confirm that the OSCSF.HOCOSF is set to 0 after setting the HOCO clock and before executing the WFI instruction.

Writing 1 to HCSTP is prohibited under the following conditions:

- SCKSCR.CKSEL[2:0] = 000b (system clock source = HOCO).
- PLLCCR.PLSRCSEL = 1 (PLL1 source clock = HOCO) and SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL1P)
- PLLCCR.PLSRCSEL = 1 (PLL1 source clock = HOCO) and PLLCR.PLLSTP = 0 (PLL1 is operating)
- PLL2CCR.PL2SRCSEL = 1 (PLL2 source clock = HOCO) and PLL2CR.PLL2STP = 0 (PLL2 is operating)

9.2.17 HOCOCR2 : High-Speed On-Chip Oscillator Control Register2

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x037

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	HCFRQ0[2:0]		
Value after reset:	0	0	0	0	0	0/1*1	0/1*1	0/1*1

Bit	Symbol	Function	R/W
2:0	HCFRQ0[2:0]	HOCO Frequency Setting 0 0 0 0: 16 MHz 0 0 1: 18 MHz 0 1 0: 20 MHz 1 0 0: 32 MHz 1 1 1: 48 MHz Others: Setting prohibited.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. Value after reset of the HCFRQ0[2:0] bits depend on OFS1(_SEC).HOCOFRQ0[2:0] bits.

The HOCOCR2 register controls the HOCO clock.

Writing to the HOCOCR2 is prohibited when the HOCOCR.HCSTP bit is 0 (the HOCO operates)

HCFRQ0[2:0] bits (HOCO Frequency Setting 0)

These bits select the frequency of HOCO.

9.2.18 MOCOCR : Middle-Speed On-Chip Oscillator Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x038

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

Bit	Symbol	Function	R/W
0	MCSTP	MOCO Stop 0: MOCO clock is operating 1: MOCO clock is stopped	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The MOCOCR register controls the MOCO clock.

MCSTP bit (MOCO Stop)

This bit starts or stops the MOCO clock.

When changing the value of the bit, only execute subsequent instructions after reading the bit and checking that its value has actually been updated.

There are restrictions on MOCO operation conditions, MOCO stop conditions and the setting of the MCSTP bit. For details, see section 9.6. Middle Speed On-chip Oscillator (MOCO).

9.2.19 FLLCR1 : FLL Control Register1

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x039

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

Bit	Symbol	Function	R/W
0	FLLEN	FLL Enable 0: FLL function is disabled 1: FLL function is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: HOCO must be stopped (HOCOCR.HCSTP = 1) before FLLCR1.FLLEN is modified.
Note: SOSC must be operating with stabilization while FLL is enabled (FLLCR1.FLLEN = 1).
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The FLLCR1 register controls the FLL function of the HOCO.

FLLEN bit (FLL Enable)

This bit enables or disables the FLL function of the HOCO.

If FLL is enabled, the frequency accuracy is guaranteed after FLL is stabilized. The FLL stabilization can be checked by the CAC frequency measurement, but it must be executed after HOCO stabilization.

In addition, you must disable FLL by setting the FLLEN bit to 0 before transitioning to Software Standby mode.

Table 9.4 show an example flow of the FLL setting for each case.

Table 9.4 FLL setting flow (1 of 2)

Step		Operation
After reset release/ Deep Software Standby cancellation	1	Start (After reset release / Deep Software Standby cancellation)
	2	FLL setting (FLLCR2.FLLCNTL)
	3	Enable FLL (FLLCR1.FLLEN = 1) Note: SOSC must be running with the oscillation stabilization.
	4	Enable HOCO (HOCOCR.HCSTP = 0)
	5	Wait for the FLL stabilization (t_{FLLWT})
	6	Check the HOCO stabilization (OSCSF.HOCOSF = 1)
	7	End (HOCO can be used.)

Table 9.4 FLL setting flow (2 of 2)

Step		Operation
Software standby transition/ cancellation	1	Start (FLL is being used.)
	2	Stop HOCO (HOCOCR.HCSTP = 1) Note: If HOCO is used as the system clock or the PLL reference clock, these clock source must be changed to another clock before HOCO is stopped.
	3	Disable FLL (FLLCR1.FLLEN = 0)
	4	WFI instruction
	5	Software standby mode
	6	Software standby cancellation
	7	Enable FLL (FLLCR1.FLLEN = 1)
	8	Enable HOCO (HOCOCR.HCSTP = 0)
	9	Wait for the FLL stabilization (t_{FLLWT})
	10	Check the HOCO stabilization (OSCSF.HOCOSF = 1)
	11	End (HOCO can be used.)

9.2.20 FLLCR2 : FLL Control Register2

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x03A

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

Bit	Symbol	Function	R/W
10:0	FLLCNTL[10:0]	FLL Multiplication Control When OFS1(_SEC).HOCOFRQ0[2:0] is 000b (16 MHz) or 100b (32 MHz), these bits must be set to 0x1E8. When OFS1(_SEC).HOCOFRQ0[2:0] is 001b (18 MHz), these bits must be set to 0x225. When OFS1(_SEC).HOCOFRQ0[2:0] is 010b (20 MHz), these bits must be set to 0x262. When OFS1(_SEC).HOCOFRQ0[2:0] is 111b (48 MHz), these bits must be set to 0x1E8. Settings other than above are prohibited.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note: The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

The FLLCR2 register controls the FLL function of the HOCO.

FLLCNTL[10:0] bits (FLL Multiplication Control)

These bits select the multiplication ratio of the FLL reference clock.

These bits must be set before FLL is enabled (FLLCR1.FLLEN=1).

9.2.21 OSCSF : Oscillation Stabilization Flag Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x03C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	PLL2SF	PLLSF	—	MOSCSF	—	—	HOCOSF
Value after reset:	0	0	0	0	0	0	0	0/1*1

Bit	Symbol	Function	R/W
0	HOCOSF	HOCO Clock Oscillation Stabilization Flag 0: The HOCO clock is stopped or is not yet stable 1: The HOCO clock is stable, so is available for use as the system clock source	R
2:1	—	These bits are read as 0.	R
3	MOSCSF	Main Clock Oscillation Stabilization Flag 0: The main clock oscillator is stopped or is not yet stable*2 1: The main clock oscillator is stable, so is available for use as the system clock source	R
4	—	This bit is read as 0.	R
5	PLLSF	PLL1 Clock Oscillation Stabilization Flag 0: The PLL1 clock is stopped or is not yet stable. 1: The PLL1 clock is stable, so is available for use as the system clock source	R
6	PLL2SF	PLL2 Clock Oscillation Stabilization Flag 0: The PLL2 clock is stopped or is not yet stable. 1: The PLL2 clock is stable	R
7	—	This bit is read as 0.	R

Note: S-TYPE-5, P-TYPE-5

Note 1. The value after reset depends on the OFS1(_SEC).HOCOEN setting.
When OFS1(_SEC).HOCOEN = 1 (disable HOCO), the value after reset of HOCOSF is 0.
When OFS1(_SEC).HOCOEN = 0 (enable HOCO), the HOCOSF value is set to 0 immediately after reset is released, and the HOCOSF value is set to 1 after the HOCO oscillation stabilization wait time elapses.

Note 2. This is true when an appropriate value is set in the Wait Control register for the main clock oscillator. If the wait time value is not sufficient, the oscillation stabilization flag is set to 1 and supply of the clock signal to the internal circuits starts before oscillation is stable.

This register is not controlled by CGFSAR register.

The OSCSF register contains flags that indicate the operating status of the oscillation stabilization wait circuit of each oscillator. After oscillation starts, these circuits generate a waiting time until the output clock of each oscillator stabilizes. The status of each oscillation stabilization wait flag set to 1 indicates that the clock supply is stable and can be used in related circuits.

HOCOSF flag (HOCO Clock Oscillation Stabilization Flag)

The HOCOSF flag indicates the operating status of the counter that measures the wait time for the high-speed clock oscillator (HOCO). When OFS1(_SEC).HOCOEN is set to 0, confirm that OSCSF.HOCOSF is set to 1 before using the HOCO clock.

[Setting condition]

- After the HOCO clock stops and the HOCOCR.HCSTP bit is set to 0, HOCOSF is set to 1 after HOCO clock is stable to ±10% or less. If the HOCO clock is used at 1.5%, it must wait for wait time specified in the electrical characteristics. (For details, see section 62, Electrical Characteristics)
  The HOCO clock is supplied when HOCOCR.HCSTP bit is set to 0, regardless of the state of HOCOSF.

[Clearing condition]

- When the HOCO clock is operating and then is deactivated because the HOCOCR.HCSTP bit is set to 1.

Note: Even if OFS1(_SEC).HOCOEN has been set to 0, confirm that the OSCSF.HOCOSF has been set to 1 before starting to use the HOCO clock.

Note: In debugging mode, even if HOCO is oscillating, this flag is masked to 0 for the user program debugging when all of following conditions are met simultaneously:

- The Trace Clock is operating enabled by setting TRCKCR.TRCKEN bit to 1.
- The Trace Clock source is selected HOCO for oscillation in debug mode by setting TRCKCR.TRCKSEL bit to 1.
- HOCOCR.HCSTP bit is set to 1.

MOSCSF flag (Main Clock Oscillation Stabilization Flag)

The MOSCSF flag indicates the operating status of the counter that measures the wait time for the main clock oscillator.

[Setting condition]

- When the main clock oscillator is stopped and the MOSCCR.MOSTP bit is set to 0, and then the number of LOCO clock cycles corresponding to the setting of the MOSCWTCR register is counted and supply of the main clock within the MCU is started.

[Clearing condition]

- When the main clock oscillator is operating and then is deactivated because the MOSCCR.MOSTP bit is set to 1.

PLLSF flag (PLL1 Clock Oscillation Stabilization Flag)

The PLLSF flag indicates the oscillation stabilization status for the PLL1.

[Setting condition]

- After the PLL1 stops and the PLLCR.PLLSTP bit is set to 0, supply of the PLL1 clock starts after PLL1 clock is stable.
  If oscillation by the PLL1 clock source selected in the PLLCCR.PLSRCSEL bit is not stable when the PLLSTP bit is set to 0, waiting for the PLL1 oscillation stabilization after the PLL1 clock source oscillation is stabilized.

[Clearing condition]

- When the PLL1 is operating and then is deactivated because the PLLCR.PLLSTP bit is set to 1.

PLL2SF flag (PLL2 Clock Oscillation Stabilization Flag)

This flag indicates the oscillation stabilization status for the PLL2.

[Setting condition]

- After the PLL2 stops and the PLL2CR.PLL2STP bit is set to 0, supply of the PLL2 clock starts after PLL2 clock is stable.
  If oscillation by the PLL2 clock source selected in the PLLCCR.PLSRCSEL is not stable when the PLL2STP bit is set to 0, waiting for the PLL2 oscillation stabilization after the oscillation of the PLL2 clock source is stabilized.

[Clearing condition]

- When the PLL2 is operating and then is deactivated because the PLL2CR.PLL2STP bit is set to 1.

9.2.22 OSCMONR : Oscillator Monitor Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x43

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

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
1	MOCOMON	MOCO operation monitor 0: MOCO is set to operate. 1: MOCO is set to stop.	R
2	LOCOMON	LOCO operation monitor 0: LOCO is set to operate. 1: LOCO is set to stop.	R
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5, P-TYPE-5

This register is not controlled by CGFSAR register.

MOCOMON bit (MOCO operation monitor)

MOCOMON indicates the operation or stop setting status for MOCO.

[Clearing condition]

- When MOCO operating conditions are met. For details, see section 9.6.1. MOCO Operating Conditions.

[Setting condition]

- When MOCO stop conditions are met. For details, see section 9.6.2. MOCO Stop Conditions.

LOCOMON bit (LOCO operation monitor)

LOCOMON indicates the operation or stop setting status for LOCO.

[Clearing condition]

- When LOCO operating conditions are met. For details, see section 9.5.1. LOCO Operating Conditions.

[Setting condition]

- When LOCO stop conditions are met. For details, see section 9.5.2. LOCO Stop Conditions.

9.2.23 OSTDCR : Oscillation Stop Detection Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x040

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

Bit	Symbol	Function	R/W
0	OSTDIE	Oscillation Stop Detection Interrupt Enable 0: Disable oscillation stop detection interrupt (do not notify the POEG) 1: Enable oscillation stop detection interrupt (notify the POEG)	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	OSTDE	Oscillation Stop Detection Function Enable 0: Disable oscillation stop detection function 1: Enable oscillation stop detection function	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The OSTDCR register controls main-clock oscillation stop detection function.

OSTDIE bit (Oscillation Stop Detection Interrupt Enable)

The OSTDIE bit enables the oscillation stop detection function interrupt. It also controls whether oscillation stop detection is reported to the POEG.

If the Oscillation Stop Detection flag in the Oscillation Stop Detection Status Register (OSTDSR.OSTDF) requires clearing, set the OSTDIE bit to 0 before clearing OSTDF. Wait for at least 2 PCLKB cycles before setting the OSTDIE bit to 1. By reading the I/O register whose access cycle number is defined by PCLKB, it is possible to secure waiting time of 2 or more cycles of PCLKB.

OSTDE bit (Oscillation Stop Detection Function Enable)

The OSTDE bit enables the oscillation stop detection function.

When the OSTDE bit is 1 (enabled), the MOCO stop bit (MOCOCR.MCSTP) is set to 0 and the MOCO operation starts. The MOCO clock cannot be stopped while the oscillation stop detection function is enabled. Writing 1 to the MOCOCR.MCSTP bit (MOCO stopped) is invalid.

When the Oscillation Stop Detection flag in the Oscillation Stop Detection Status Register (OSTDSR.OSTDF) is 1 (main clock oscillation stop detected), writing 0 to the OSTDE bit is invalid.

9.2.24 OSTDSR : Oscillation Stop Detection Status Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x041

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

Bit	Symbol	Function	R/W
0	OSTDF	Oscillation Stop Detection Flag 0: Main clock oscillation stop not detected 1: Main clock oscillation stop detected	R/W*1
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. This bit can only be set to 0. This bit is cleared to 0 by writing 0 after reading it as 1.

The OSTDSR register indicates the stop detection status of the main clock oscillator.

OSTDF flag (Oscillation Stop Detection Flag)

The OSTDF flag indicates the main clock oscillator status. When this flag is 1, it indicates that the main clock oscillation stop was detected. After this stop is detected, the OSTDF flag is not set to 0 even when the main clock oscillation is restarted. The OSTDF bit is cleared to 0 by writing 0 after reading it as 1.

At least 3 ICLK cycles of wait time are required between writing 0 to OSTDF and reading it as 0. If the OSTDF flag is set to 0 when the main clock oscillation is stopped, the OSTDF flag becomes 0 then returns to 1.

The OSTDF flag cannot be set to 0 under the following conditions:

- SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC).
- PLLCCR.PLSRCSEL = 0 (PLL1 source clock = MOSC) and SCKSCR.CKSEL[2:0] = 101b (System clock source = PLL1P)

The OSTDF flag must be set to 0 after switching the clock source to sources other than the main clock oscillator and PLL1P.

[Setting condition]

- The main clock oscillator is stopped when OSTDCR.OSTDE = 1 (oscillation stop detection function enabled).

[Clearing condition]

- 1 is read and then 0 is written when the SCKSCR.CKSEL[2:0] bits are neither 011b (system clock source is MOSC) nor, 101b (system clock source is PLL1P) and PLLCCR.PLSRCSEL bit is not 0 (PLL1 source clock is MOSC).

9.2.25 SOSTDCR : Sub-clock Oscillation Stop Detection Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC04

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

Bit	Symbol	Function	R/W
0	SOSTDIE	Sub-clock Oscillation Stop Detection Interrupt Enable 0: The oscillation stop detection interrupt is disabled. 1: The oscillation stop detection interrupt is enabled.	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	SOSTDE	Sub-clock Oscillation Stop Detection Function Enable 0: Oscillation stop detection function is disabled. 1: Oscillation stop detection function is enabled.	R/W

Note: S-TYPE-3, P-TYPE-2

SOSTDCR Register controls Sub-clock oscillation-stop detection function.

This register is only initialized by VBATT_POR reset.

SOSTDIE bit (Sub-clock Oscillation Stop Detection Interrupt Enable)

This bit enables or disables the Sub-clock oscillation stop detection function interrupt.

If the Sub-clock oscillation stops detection flag in the Sub-clock oscillation stop detection status register (SOSTDSR.SOSTDF) requires clearing, clear the SOSTDF after the SOSTDIE bit is cleared to 0. Make sure SOSTDF is cleared before setting the SOSTDIE bit to 1 again.

SOSTDE bit (Sub-clock Oscillation Stop Detection Function Enable)

This bit enables or disables the sub-clock oscillation stop detection function.

When the SOSTDE bit is 1 (sub-clock oscillation stop detection function enabled), the MOCO stop bit (MOCOCR.MCSTP) is cleared to 0 and the MOCO operation is started. The MOCO cannot be stopped while the oscillation stop detection function is enabled except in deep software standby mode and VBAT mode. Writing 1 to the MOCOCR.MCSTP bit (MOCO stopped) is invalid.

When the sub-clock oscillation stops detection flag in the sub-clock oscillation stop detection status register (SOSTDSR.SOSTDF) is 1 (sub-clock oscillation stop detected), writing 0 to the SOSTDE bit is invalid.

To enable the sub-clock oscillation stop detection function, the SOSTDSR.SOSTDF flag must be cleared to 0 after setting the SOSTDCR.SOSTDE bit to 1. However, after setting the SOSTDCR.SOSTDE bit to 1, be sure to wait for the start-up time of the sub-clock oscillation stop detector (Tsosdup) before clearing the SOSTDSR.SOSTDF flag. For SOSTDSR, please see section 9.2.26. SOSTDSR : Sub-clock Oscillation Stop Detection Status Register.

Be sure to set the system clock to other clock sources than the sub-clock oscillator before enabling the sub-clock oscillation stop detection function.

9.2.26 SOSTDSR : Sub-clock Oscillation Stop Detection Status Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC05

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

Bit	Symbol	Function	R/W
0	SOSTDF	Sub-clock Oscillation Stop Detection Flag 0: The sub-clock oscillation stop has not been detected. 1: The sub-clock oscillation stop has been detected.	R/W*1
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note 1. This bit can only be set to 0.

SOSTDSR Register indicates status of the sub-clock oscillation stop detection.

This register is only initialized by VBATT_POR reset.

SOSTDF flag (Sub-clock Oscillation Stop Detection Flag)

This bit is a flag to indicate the sub-clock status. When the SOSTDF flag is 1, it indicates that the sub-clock oscillation stop has been detected.

Once the sub-clock oscillation stop is detected, the OSTDF bit is not cleared to 0 even though the sub-clock oscillation is restarted. The SOSTDF bit is cleared to 0 by reading 1 from the bit and then writing 0.

At least 3 PCLKB cycles of wait time is necessary between writing 0 to SOSTDF and reading SOSTDF as 0.

If the SOSTDF bit is cleared to 0 while the sub-clock oscillation is stopped, the SOSTDF bit is immediately set to 1.

SOSTDSR.SOSTDF cannot be cleared to 0 while the following conditions:

- SCKSCR.CKSEL[2:0] = 100 (System clock source = SOSC)

[Setting condition]

- The sub-clock oscillation is stopped with the SOSTDCR.SOSTDE being 1 (sub-clock oscillation stop detection function enabled).

[Clearing condition]

- 1 is read and then 0 is written when the SCKSCR.CKSEL[2:0] bits are not 100b (System clock is SOSC).

9.2.27 MOSCWTCR : Main Clock Oscillator Wait Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x0A2

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

Bit	Symbol	Function	R/W
3:0	MSTS[3:0]	Main Clock Oscillator Wait Time Setting 0x0: Wait time = 3 cycles (11.4 μs) 0x1: Wait time = 35 cycles (133.5 μs) 0x2: Wait time = 67 cycles (255.6 μs) 0x3: Wait time = 131 cycles (499.7 μs) 0x4: Wait time = 259 cycles (988.0 μs) 0x5: Wait time = 547 cycles (2086.6 μs) 0x6: Wait time = 1059 cycles (4039.8 μs) 0x7: Wait time = 2147 cycles (8190.2 μs) 0x8: Wait time = 4291 cycles (16368.9 μs) 0x9: Wait time = 8163 cycles (31139.4 μs) Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

MSTS[3:0] bits (Main Clock Oscillator Wait Time Setting)

The MSTS[3:0] bits specify the oscillation stabilization wait time for the main clock oscillator.

Set the main clock oscillation stabilization time to a period longer than or equal to the stabilization time recommended by the oscillator manufacturer. When the main clock is input externally, set these bits to 0x0 because the oscillation stabilization time is not required.

The wait time set in these bits is counted using: 1 cycle (μs) = $1/(f_{LOCO}[MHz] \times 8) = 1/(0.032768 \times 8) = 3.81$ (μs) (min.) The LOCO clock automatically oscillates when necessary, regardless of the value of the LOCO.LCSTP bit. After the specified wait time elapses, supply of the main clock starts internally in the MCU, and the OSCSF.MOSCSF flag is set to 1. If the specified wait time is short, supply of the main clock starts before oscillation of the clock becomes stable.

Only rewrite the MOSCWTCR register when the MOSCCR.MOSTP bit is 1 and the OSCSF.MOSCSF flag is 0. Do not rewrite this register under any other conditions.

9.2.28 MOMCR : Main Clock Oscillator Mode Oscillation Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA50

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	MOSEL	—	—	MODRV0[2:0]			—
Value after reset:	0	0	0	1	1	0	1	0

Bit	Symbol	Function	R/W
0	—	These bits are read as 0. The write value should be 0.	R/W
3:1	MODRV0[2:0]	Main Clock Oscillator Drive Capability 0 Switching 0 0 0: 8 MHz 0 1 1: 8MHz to 24MHz 1 0 1: 8MHz to 48MHz Others: Setting prohibited	R/W
4	—	This bit is read as 1. The write value should be 1.	R/W
5	—	These bits are read as 0. The write value should be 0.	R/W
6	MOSEL	Main Clock Oscillator Switching 0: Resonator 1: External clock input	R/W
7	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: The EXTAL/XTAL pins are also used as ports. In the initial state, the pin is set as a port.
Note: The MOSCCR.MOSTP bit must be 1 (MOSC is stopped) before changing this register.
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

MODRV0[2:0] bit (Main Clock Oscillator Drive Capability 0 Switching)

The MODRV0[2:0] bit switches the drive capability of the main clock oscillator. It is recommended to set 101b, other setting is possible to reduce current consumption. See section 62, Electrical Characteristics for current consumption.

MOSEL bit (Main Clock Oscillator Switching)

The MOSEL bit switches the source for the main clock oscillator.

9.2.29 SOMCR : Sub-Clock Oscillator Mode Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC01

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	SOSEL	—	—	—	—	SODRV[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	SODRV[1:0]	Sub-Clock Oscillator Drive Capability Switching 0 0: Standard (12.5pf) 0 1: Lowpower mode 1 (9pf) 1 0: Lowpower mode 2 (7pf) 1 1: Lowpower mode 3 (4pf)	R/W
5:2	—	These bits are read as 0. The write value should be 0.	R/W
6	SOSEL	Sub-Clock Oscillator Switching 0: Resonator 1: External clock input	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The SOMCR register must be modified when SOSCCR.SOSTP is 1 (SOSC is stopped).

This register is initialized by only VBATT_POR reset.

SODRV[1:0] bits (Sub-Clock Oscillator Drive Capability Switching)

The SODRV[1:0] bits switch the drive capability of the sub-clock oscillator.

SOSEL bit (Sub-Clock Oscillator Switching)

The SOSEL bit switches the source for the Sub-clock oscillator.

For details, see section 9.4. Sub-Clock Oscillator.

9.2.30 CKOCR : Clock Out Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x03E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CKOEN	CKODIV[2:0]			—	CKOSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
2:0	CKOSEL[2:0]	Clock Out Source Select 0 0 0: HOCO 0 0 1: MOCO (Value after reset) 0 1 0: LOCO 0 1 1: MOSC 1 0 0: SOSC Others: Setting prohibited	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
6:4	CKODIV[2:0]	Clock Output Frequency Division Ratio 0 0 0: × 1/1 0 0 1: × 1/2 0 1 0: × 1/4 0 1 1: × 1/8 1 0 0: × 1/16 1 0 1: × 1/32 1 1 0: × 1/64 1 1 1: × 1/128	R/W
7	CKOEN	Clock Out Enable 0: Disable clock out 1: Enable clock out	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

CKOSEL[2:0] bits (Clock Out Source Select)

The CKOSEL[2:0] bits select the source of the clock to be output from the CLKOUT pin. When changing the clock source, set the CKOEN bit to 0.

CKODIV[2:0] bits (Clock Output Frequency Division Ratio)

The CKODIV[2:0] bits specify the clock division ratio. Set the CKOEN bit to 0 when changing the division ratio.

CKOEN bit (Clock Out Enable)

The CKOEN bit enables output from the CLKOUT pin.

When this bit is set to 1, the selected clock is output. When this bit is set to 0, low is output. When changing this bit, confirm that the clock out source clock selected in the CKOSEL[2:0] bits is stable. Otherwise, a glitch might be generated in the output.

Clear this bit before entering Software Standby or Deep Software Standby mode if the selecting clock out source clock is stopped in that mode.

9.2.31 EBCKOCR : External Bus Clock Output Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x052

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

Bit	Symbol	Function	R/W
0	EBCKOEN	EBCLK Pin Output Control 0: EBCLK pin output is disabled (fixed high) 1: EBCLK pin output is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

9.2.32 SDCKOCR : SDRAM Clock Output Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x053

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

Bit	Symbol	Function	R/W
0	SDCKOEN	SDCLK Pin Output Control 0: SDCLK pin output is disabled. (Fixed high) 1: SDCLK pin output is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

SDCKOEN bit (SDCLK Pin Output Control)

This bit selects controls the output of the SDCLK pin.

9.2.33 LOCOUTCR : LOCO User Trimming Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x402

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

Bit	Symbol	Function	R/W
7:0	LOCOUTRM[7:0]	LOCO User Trimming 0x80: -128 0x81: -127 ⋮ 0xFF: -1 0x00: Center Code 0x01: +1 ⋮ 0x7E: +126 0x7F: +127	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note: Make sure that SYRACCR.BUSY is 0 before accessing LOCOUTCR. When SYRACCR.BUSY is 1, the write access is ignored and the read value is not guaranteed.
Note: MCU operation is not guaranteed when LOCOUTCR is set to a value that causes the LOCO frequency to be outside of the specification range.
Note: After changing LOCOUTCR, it takes certain time until the frequency to be stable. The time is same as the oscillation stabilization time in startup.

The LOCOUTCR register is added to the original LOCO trimming data.

Changing LOCOUTCR during RTC operation is prohibited.

9.2.34 MOCOUTCR : MOCO User Trimming Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x061

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

Bit	Symbol	Function	R/W
7:0	MOCOUTRM[7:0]	MOCO User Trimming 0x80: -128 0x81: -127 ⋮ 0xFF: -1 0x00: Center Code 0x01: +1 ⋮ 0x7E: +126 0x7F: +127	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note: MCU operation is not guaranteed when MOCOUTCR is set to a value that causes the MOCO frequency to be outside of the specification range.
Note: After changing MOCOUTCR, it takes certain time until the frequency to be stable. The time is same as the oscillation stabilization time in startup.

The MOCOUTCR register is added to the original MOCO trimming data.

9.2.35 HOCOUTCR : HOCO User Trimming Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x062

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

Bit	Symbol	Function	R/W
7:0	HOCOUTRM[7:0]	HOCO User Trimming 0x80: -128 0x81: -127 ⋮ 0xFF: -1 0x00: Center Code 0x01: +1 ⋮ 0x7E: +126 0x7F: +127	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note: MCU operation is not guaranteed when HOCOUTCR is set to a value that causes the HOCO frequency to be outside of the specification range.
Note: After changing HOCOUTCR, it takes certain time until the frequency to be stable. The time is same as the oscillation stabilization time in startup.
Note: These bits must be 0x00 when FLL is enabled (FLLCR1.FLLEN = 1).

The HOCOUTCR register is added to the original HOCO trimming data.

9.2.36 MOSCSCR : Main Clock Oscillator Standby Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x07C

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

Bit	Symbol	Function	R/W
0	MOSCSOKP	Main Clock Oscillator Standby Oscillation Keep select 0: Disable 1: Enable	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The MOSCSCR must be changed while MOSCCR.MOSTP bit is 1 (MOSC is stopped).

MOSCSOKP bit (Main Clock Oscillator Standby Oscillation Keep select)

This bit controls that Main Clock Oscillator keeps oscillating in Software Standby mode.

If this bit is disabled and MOSCCR.MOSTP bit is 0 (MOSC is run), the Main Clock Oscillator stops in Software Standby mode.

If this bit is enabled and MOSCCR.MOSTP bit is 0 (MOSC is run), the Main Clock Oscillator will keep oscillating in Software Standby mode.

If MOSCCR.MOSTP bit is 1 (MOSC is stopped), the Main Clock Oscillator stops in Software Standby mode regardless of the setting of this bit.

9.2.37 HOCOSCR : High-Speed On-Chip Oscillator Standby Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x07D

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

Bit	Symbol	Function	R/W
0	HOCOSOKP	HOCO Standby Oscillation Keep select. 0: Disable 1: Enable	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

HOCOSOKP bit (HOCO Standby Oscillation Keep select.)

This bit controls that HOCO keeps oscillating in Software Standby mode.

If this bit is disabled and HOCOCR.HCSTP bit is 0 (HOCO is run), the HOCO stops in Software Standby mode.

If this bit is enabled and HOCOCR.HCSTP bit is 0 (HOCO is run), the HOCO will keep oscillating in Software Standby mode.

If HOCOCR.HCSTP bit is 1 (HOCO is stopped), the HOCO stops in Software Standby mode regardless of the setting of this bit.

9.2.38 MOCOSCR : Middle-Speed On-Chip Oscillator Standby Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x084

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

Bit	Symbol	Function	R/W
0	MOCOSOKP	MOCO Standby Oscillation Keep select 0: Disable 1: Enable	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

MOCOSOKP bit (MOCO Standby Oscillation Keep select)

This bit controls that MOCO keeps oscillating in Software Standby mode.

If this bit is disabled and MOCOCR.MCSTP bit is 0 (MOCO is run), the MOCO stops in Software Standby mode.

If this bit is enabled and MOCOCR.MCSTP bit is 0 (MOCO is run), the MOCO will keep oscillating in Software Standby mode.

9.2.39 USBCKDIVCR : USB Clock Division Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x06C

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

Bit	Symbol	Function	R/W
3:0	USBCKDIV[3:0]	USB Clock (USBCLK) Division Select 0 0 0 0: 1/1 (value after reset) 0 0 0 1: 1/2 0 0 1 0: 1/4 0 0 1 1: 1/6 0 1 0 0: 1/8 0 1 0 1: 1/3 0 1 1 0: 1/5 0 1 1 1: 1/10 1 0 0 0: 1/16 1 0 0 1: 1/32 Others: Setting prohibited.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The USBCKDIVCR register controls the USB clock.

USBCKDIV[3:0] bits (USB Clock (USBCLK) Division Select)

These bits select the frequency of the USB clock (USBCLK) and must be modified when USBCKCR.USBCKSRDY = 1.

When switching the division ratio setting from 1/n (n ≠ 1), set MSTPCRB.MSTPB11 = 1 and MSTPCRB.MSTPB12 = 1 before setting USBCKCR.USBCKSREQ = 1.

9.2.40 OCTACKDIVCR : Octal-SPI Clock Division Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x06D

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

Bit	Symbol	Function	R/W
3:0	OCTACKDIV[3:0]	Octal-SPI Clock (OCTACLK) Division Select 0 0 0 0: 1/1 (Value after reset) 0 0 0 1: 1/2 0 0 1 0: 1/4 0 0 1 1: 1/6 0 1 0 0: 1/8 0 1 0 1: 1/3 0 1 1 0: 1/5 0 1 1 1: 1/10 1 0 0 0: 1/16 1 0 0 1: 1/32 Others: Setting prohibited.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

OCTACKDIVCR controls the Octal-SPI clock.

OCTACKDIV[3:0] bits (Octal-SPI Clock (OCTACLK) Division Select)

These bits select the frequency of the Octal-SPI clock (OCTACLK) and must be modified when OCTACKCR.OCTACKSRDY = 1.

When switching the division ratio setting from 1/n (n ≠ 1), set MSTPCRB.MSTPB16 = 1 and MSTPCRB.MSTPB17 = 1 before setting OCTACKCR.OCTACKSREQ = 1.

9.2.41 CANFDCKDIVCR : CANFD Core Clock Division Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x06E

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

Bit	Symbol	Function	R/W
3:0	CANFDCKDIV[3:0]	CANFD core clock (CANFDCLK) Division Select 0 0 0 0: 1/1 (Value after reset) 0 0 0 1: 1/2 0 0 1 0: 1/4 0 0 1 1: 1/6 0 1 0 0: 1/8 0 1 0 1: 1/3 0 1 1 0: 1/5 0 1 1 1: 1/10 1 0 0 0: 1/16 1 0 0 1: 1/32 Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

CANFDCKDIVCR controls the CANFD core clock.

CANFDCKDIV[3:0] bits (CANFD core clock (CANFDCLK) Division Select)

These bits select the frequency of the CANFD core clock (CANFDCLK) and must be modified when CANFDCKCR.CANFDCKSRDY = 1.

When switching the division ratio setting from 1/n (n ≠ 1), set MSTPCRC.MSTPC26 = 1 and MSTPCRC.MSTPC27 = 1 before setting CANFDCKCR.CANFDCKSREQ = 1.

9.2.42 USB60CKDIVCR : USB60 Clock Division Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x06F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	USB60CKDIV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	USB60CKDIV[3:0]	USB clock (USB60CLK) Division Select 0 0 0 0: 1/1 (Value after reset) 0 0 0 1: 1/2 0 0 1 0: 1/4 0 0 1 1: 1/6 0 1 0 0: 1/8 0 1 0 1: 1/3 0 1 1 0: 1/5 0 1 1 1: 1/10 1 0 0 0: 1/16 1 0 0 1: 1/32 Others: Setting prohibited.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

USB60CKDIVCR controls the USB clock.

USB60CKDIV[3:0] bits (USB clock (USB60CLK) Division Select)

These bits select the frequency of the USB clock USB60CKDIV and must be modified when USB60CKCR.USB60CKSRDY = 1.

When switching the division ratio setting from 1/n (n ≠ 1), set MSTPCRB.MSTPB12 = 1 before setting USB60CKCR.USB60CKSREQ = 1.

9.2.43 I3CCKDIVCR : I3C Clock Division Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x070

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	I3CCKDIV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	I3CCKDIV[3:0]	I3C clock (I3CCLK) Division Select 0 0 0 0: 1/1 (Value after reset) 0 0 0 1: 1/2 0 0 1 0: 1/4 0 0 1 1: 1/6 0 1 0 0: 1/8 0 1 0 1: 1/3 0 1 1 0: 1/5 0 1 1 1: 1/10 1 0 0 0: 1/16 1 0 0 1: 1/32 Others: Setting prohibited.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

I3CCKDIVCR controls the I3C clock.

I3CCKDIV[3:0] bits (I3C clock (I3CCLK) Division Select)

These bits select the frequency of the I3C clock (I3CCLK) and must be modified when I3CCKCR.I3CCKSRDY = 1.

When switching the division ratio setting from 1/n (n ≠ 1), set MSTPCRB.MSTPB4 = 1 before setting I3CCKCR.I3CCKSREQ = 1.

9.2.44 USBCKCR : USB Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x074

Bit position:	7	6	5	4	3	2	1	0
Bit field:	USBCKSRDY	USBCKSREQ	—	—	USBCKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	USBCKSEL[3:0]	USB Clock (USBCLK) Source Select 0000: HOCO 0001: MOCO (Value after reset) 0011: Main clock oscillator 0101: PLL1P 0110: PLL2P 0111: PLL1Q 1000: PLL1R 1001: PLL2Q 1010: PLL2R Others: Setting prohibited.	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
6	USBCKSREQ	USB Clock (USBCLK) Switching Request 0: No request 1: Request switching.	R/W
7	USBCKSRDY	USB Clock (USBCLK) Switching Ready state flag 0: Impossible to Switch 1: Possible to Switch	R

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The USBCKCR register controls the USB clock.

When switching the clock source, ensure that the clock before the switch and the clock after the switch generate stable output. To change the set value of USBCKDIVCR.USBCKDIV[3:0] and USBCKSEL[3:0], use the following procedure:

1. Write 1 to MSTPCRB.MSTPB11 and MSTPCRB.MSTPB12 only when switching the division ratio setting from 1/n (n ≠ 1).
2. Wait for two USBCLK only when switching the division ratio setting from 1/n (n ≠ 1).
3. Write 1 to USBCKSREQ.
4. Poll until USBCKSRDY is read as 1. While USBCKSRDY = 1, no clock is output to USBCLK.
5. Write to USBCKDIVCR.USBCKDIV[3:0] and USBCKSEL[3:0].
6. Write 0 to USBCKSREQ.
7. Poll until USBCKSRDY is read as 0.
8. When USBCKSRDY becomes 0, USBCLK starts to output. Clock switching is complete.

When transitioning to Software Standby or Deep Software Standby mode, do not execute the WFI instruction while performing clock switching. That is, do not execute the WFI instruction when USBCKSREQ = 1 and USBCKSRDY = 0, or when USBCKSREQ = 0 and USBCKSRDY = 1.

USBCKSEL[3:0] bits (USB Clock (USBCLK) Source Select)

These bits select the clock source of the USB clock (USBCLK) and must be modified when USBCKCR.USBCKSRDY = 1.

Do not stop the oscillator selected by these bits except when MOCO is selected.

USBCKSREQ bit (USB Clock (USBCLK) Switching Request)

This bit selects the USBCLK switching request.

USBCKSRDY flag (USB Clock (USBCLK) Switching Ready state flag)

This flag indicates the state of switching ready for the USBCLK. When USBCKSRDY = 1, no clock is output to USBCLK.

9.2.45 OCTACKCR : Octal-SPI Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x075

Bit position:	7	6	5	4	3	2	1	0
Bit field:	OCTACKSRDY	OCTACKSREQ	—	—	OCTACKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	OCTACKSEL[3:0]	Octal-SPI Clock (OCTACLK) Source Select 0 0 0 0: HOCO 0 0 0 1: MOCO (Value after reset) 0 0 1 0: LOCO 0 0 1 1: Main clock oscillator 0 1 0 0: Sub-clock oscillator 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Others: Setting prohibited.	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	OCTACKSREQ	Octal-SPI Clock (OCTACLK) Switching Request 0: No request 1: Request switching.	R/W
7	OCTACKSRDY	Octal-SPI Clock (OCTACLK) Switching Ready state flag 0: Impossible to Switch 1: Possible to Switch	R

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The OCTACKCR register controls the Octal-SPI clock.

When switching the clock source, ensure that the clock before the switch and the clock after the switch generate stable output.

To change the set value of OCTACKDIVCR.OCTACKDIV[3:0] and OCTACKSEL[3:0], use the following procedure:

1. Write 1 to MSTPCRB.MSTPB16 and MSTPCRB.MSTPB17 only when switching the division ratio setting from 1/n (n ≠ 1).
2. Confirm the reading value of both MSTPCRB.MSTPB16 and MSTPCRB.MSTPB17 is 1 only when switching the division ratio setting from 1/n (n ≠ 1).
3. Write 1 to OCTACKSREQ.
4. Poll until OCTACKSRDY is read as 1. While OCTACKSRDY = 1, no clock is output to OCTACLK.
5. Write to OCTACKDIVCR.OCTACKDIV[3:0] and OCTACKSEL[3:0].
6. Write 0 to OCTACKSREQ.
7. Poll until OCTACKSRDY is read as 0.
8. When OCTACKSRDY becomes 0, OCTACLK starts to output. Clock switching is complete.

When transitioning to Software Standby or Deep Software Standby mode, do not execute the WFI instruction while performing clock switching. That is, do not execute the WFI instruction when OCTACKSREQ = 1 and OCTACKSRDY = 0, or when OCTACKSREQ = 0 and OCTACKSRDY = 1.

OCTACKSEL[3:0] bits (Octal-SPI Clock (OCTACLK) Source Select)

These bits select the clock source of the Octal-SPI clock (OCTACLK) and must be modified when OCTACKCR.OCTACKSRDY = 1.

Do not stop the oscillator selected by these bits except when MOCO is selected.

OCTACKSREQ bit (Octal-SPI Clock (OCTACLK) Switching Request)

This bit selects the OCTACLK switching request.

OCTACKSRDY flag (Octal-SPI Clock (OCTACLK) Switching Ready state flag)

This flag indicates the state of switching ready for the OCTACLK. When OCTACKSRDY = 1, no clock is output to OCTACLK.

9.2.46 CANFDCKCR : CANFD Core Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x076

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CANF DCKS RDY	CANF DCKS REQ	—	—	CANFDCKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	CANFDCKSEL[3:0]	CANFD Core Clock (CANFDCLK) Source Select 0 0 0 0: HOCO 0 0 0 1: MOCO (Value after reset) 0 0 1 0: LOCO 0 0 1 1: Main clock oscillator 0 1 0 0: Sub-clock oscillator 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Others: Setting prohibited.	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	CANFDCKSREQ	CANFD Core Clock (CANFDCLK) Switching Request 0: No request 1: Request switching	R/W
7	CANFDCKSRDY	CANFD Core Clock (CANFDCLK) Switching Ready state flag 0: Impossible to Switch 1: Possible to Switch	R

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The CANFDCKCR register controls the CANFD Core Clock (CANFDCLK).

When switching the clock source, ensure that the clock before the switch and the clock after the switch generate stable output. To change the set value of CANFDCKDIVCR.CANFDCKDIV[3:0] and CANFDCKSEL[3:0], use the following procedure:

1. Write 1 to MSTPCRC.MSTPC26 and MSTPCRC.MSTPC27 only when switching the division ratio setting from 1/n (n ≠ 1).
2. Wait for two CANFDCLK only when switching the division ratio setting from 1/n (n ≠ 1).
3. Write 1 to CANFDCKSREQ.
4. Poll until CANFDCKSRDY is read as 1. While CANFDCKSRDY = 1, no clock is output to CANFDCLK.
5. Write to CANFDCKDIVCR.CANFDCKDIV[3:0] and CANFDCKSEL[3:0].
6. Write 0 to CANFDCKSREQ.
7. Poll until CANFDCKSRDY is read as 0.
8. When CANFDCKSRDY becomes 0, CANFDCLK starts to output. Clock switching is complete.

When transitioning to Software Standby or Deep Software Standby mode, do not execute the WFI instruction while performing clock switching. That is, do not execute the WFI instruction when CANFDCKSREQ = 1 and CANFDCKSRDY = 0, or when CANFDCKSREQ = 0 and CANFDCKSRDY = 1.

CANFDCKSEL[3:0] bits (CANFD Core Clock (CANFDCLK) Source Select)

These bits select the clock source of the CANFD Core clock (CANFDCLK) and must be modified when CANFDCKCR.CANFDCKSRDY = 1.

Do not stop the oscillator selected by these bits except when MOCO is selected.

CANFDCKSREQ bit (CANFD Core Clock (CANFDCLK) Switching Request)

This bit selects the CANFDCLK switching request.

CANFDCKSRDY flag (CANFD Core Clock (CANFDCLK) Switching Ready state flag)

This flag indicates the state of switching ready for the CANFDCLK. When CANFDCKSRDY = 1, no clock is output to CANFDCLK.

9.2.47 USB60CKCR : USB60 Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x077

Bit position:	7	6	5	4	3	2	1	0
Bit field:	USB60 CKSR DY	USB60 CKSR EQ	—	—	USB60CKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	USB60CKSEL[3:0]	USB60 Clock (USB60CK) Source Select 0 0 0 0: HOCO 0 0 0 1: MOCO (Value after reset) 0 0 1 1: Main clock oscillator 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Others: Setting prohibited.	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	USB60CKSREQ	USB60 Clock (USB60CK) Switching Request 0: No request 1: Request switching	R/W
7	USB60CKSRDY	USB60 Clock (USB60CK) Switching Ready state flag 0: Impossible to Switch 1: Possible to Switch	R

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The USB60CKCR register controls the USB60 Clock.

To change the set value of USB60CKDIVCR. USB60CKDIV[3:0] and USB60CKSEL[3:0], follow the procedure below.

When switching the clock source, it is necessary that the clock before switching and the clock after switching are stably output.

Clock selection switching procedure:

1. Write 1 to MSTPCRB.MSTPB12 only when switching the division ratio setting from 1/n (n ≠ 1).
2. Wait for two USB60CLK only when switching the division ratio setting from 1/n (n ≠ 1).
3. Write 1 to USB60CKSREQ.
4. Poll until USB60CKSRDYY is read as 1. While USB60CKSRDY = 1, no clock is output to USB60CLK.
5. Write to USB60CKDIVCR.USB60CKDIV[3:0] and USB60CKSEL[3:0].

6. Write 0 to USB60CKSREQ.
7. Poll until USB60CKSRDY is read as 0.
8. When USB60CKSRDY becomes 0, USB60CLK starts to output. Clock switching is complete.

When transitioning to Software Standby or Deep Software Standby mode, do not execute the WFI instruction while performing clock switching. That is, do not execute the WFI instruction when USB60CKSREQ = 1 and USB60CKSRDY = 0, or when USB60CKSREQ = 0 and USB60CKSRDY = 1.

USB60CKSEL[3:0] bits (USB60 Clock (USB60CK) Source Select)

These bits select the clock source of the USB Clock (USB60CLK) and must be modified when USB60CKCR.USB60CKSRDY = 1.

Do not stop the oscillator selected by these bits except when MOCO is selected.

USB60CKSREQ bit (USB60 Clock (USB60CK) Switching Request)

This bit selects the USB60CLK switching request.

USB60CKSRDY flag (USB60 Clock (USB60CK) Switching Ready state flag)

This flag indicates the state of switching ready for the USB60CLK. When USB60CKSRDY = 1, no clock is output to USB60CLK.

9.2.48 I3CCKCR : I3C Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x078

Bit position:	7	6	5	4	3	2	1	0
Bit field:	I3CCKSRDY	I3CCKSREQ	—	—	I3CCKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	I3CCKSEL[3:0]	I3C Clock (I3CCLK) Source Select 0001: MOCO (value after reset) 0101: PLL1P 0110: PLL2P 0111: PLL1Q 1000: PLL1R 1001: PLL2Q 1010: PLL2R Others: Setting prohibited	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	I3CCKSREQ	I3C Clock (I3CCLK) Switching Request 0: No request 1: Request switching	R/W
7	I3CCKSRDY	I3C Clock (I3CCLK) Switching Ready state flag 0: Impossible to switch 1: Possible to switch	R

Note: S-TYPE-3, P-TYPE-2

I3CCKCR controls the I3C clock.

To change the set value of I3CCKDIVCR.I3CCKDIV[3:0] and I3CCKSEL[3:0], follow the procedure below.

When switching the clock source, it is necessary that the clock before switching and the clock after switching are stably output.

Clock selection switching procedure:

1. Write 1 to MSTPCRB.MSTPB4 only when switching the division ratio setting from 1/n (n ≠ 1).

2. Wait for two I3CCLK only when switching the division ratio setting from 1/n (n ≠ 1).
3. Write 1 to I3CCKSREQ.
4. Polling until I3CCKSRDY is read as 1. While I3CCKSRDY = 1, no clock is output to I3CCLK.
5. Write the setting value to I3CCKDIVCR.I3CCKDIV[3:0] and I3CCKSEL[3:0].
6. Write 0 to I3CCKSREQ.
7. Polling until I3CCKSRDY is read as 0.
8. When I3CCKSRDY becomes 0, I3CCLK starts outputting. Clock switching complete.

When a transition to Software Standby or Deep Software Standby, do not execute the WFI instruction while performing clock selection switching. In other words, do not execute the WFI instruction with I3CCKSREQ = 1 and I3CCKSRDY = 0, or I3CCKSREQ = 0 and I3CCKSRDY = 1.

I3CCKSEL[3:0] bits (I3C Clock (I3CCLK) Source Select)

These bits select the clock source of the I3C clock (I3CCLK) and must be modified when I3CCKCR.I3CCKSRDY = 1.

Do not stop the oscillator selected by these bits except when MOCO is selected.

I3CCKSREQ bit (I3C Clock (I3CCLK) Switching Request)

This bit selects the I3CCLK switching request.

I3CCKSRDY bit (I3C Clock (I3CCLK) Switching Ready state flag)

This flag indicates the state of switching ready for the I3CCLK.

While I3CCKSRDY = 1, no clock is output to I3CCLK.

9.2.49 SCICKDIVCR : SCI Clock Division Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x054

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

Bit	Symbol	Function	R/W
3:0	SCICKDIV[3:0]	SCI clock (SCICLK) Division Select 0 0 0 0: 1/1 (Value after reset) 0 0 0 1: 1/2 0 0 1 0: 1/4 0 0 1 1: 1/6 0 1 0 0: 1/8 0 1 0 1: 1/3 0 1 1 0: 1/5 0 1 1 1: 1/10 1 0 0 0: 1/16 1 0 0 1: 1/32 Others: Setting prohibited.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

SCICKDIVCR controls the SCI clock.

SCICKDIV[3:0] bits (SCI clock (SCICLK) Division Select)

These bits select the frequency of the SCI clock (SCICLK) and must be modified when SCICKCR.SCICKSRDY = 1.

When switching the division ratio setting from 1/n (n ≠ 1), set all MSTPCRB.MSTPB22 to MSTPB31 = 1 before setting SCICKCR.SCICKSREQ = 1.

9.2.50 SPICKDIVCR : SPI Clock Division Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x056

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

Bit	Symbol	Function	R/W
3:0	SPICKDIV[3:0]	SPI clock (SPICLK) Division Select 0 0 0 0: 1/1 (Value after reset) 0 0 0 1: 1/2 0 0 1 0: 1/4 0 0 1 1: 1/6 0 1 0 0: 1/8 0 1 0 1: 1/3 0 1 1 0: 1/5 0 1 1 1: 1/10 1 0 0 0: 1/16 1 0 0 1: 1/32 Others: Setting prohibited.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

SPICKDIVCR controls the SPI clock.

SPICKDIV[3:0] bits (SPI clock (SPICLK) Division Select)

These bits select the frequency of the SPI clock (SPICLK) and must be modified when SPICKCR.SPICKSRDY = 1.

When switching the division ratio setting from 1/n (n ≠ 1), set MSTPCRB.MSTPB18 = 1 and MSTPCRB.MSTPB19 = 1 before setting SPICKCR.SPICKSREQ = 1.

9.2.51 ADCCKDIVCR : ADC Clock Division Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x05A

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

Bit	Symbol	Function	R/W
3:0	ADCCKDIV[3:0]	ADC Clock (ADCCLK) Division Select 0 0 0 0: 1/1 (value after reset) 0 0 0 1: 1/2 0 0 1 0: 1/4 0 0 1 1: 1/6 0 1 0 0: 1/8 0 1 1 1: 1/10 1 0 0 0: 1/16 1 0 0 1: 1/32 Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

ADCCKDIVCR controls the ADC clock.

ADCCKDIV[3:0] bits (ADC Clock (ADCCLK) Division Select)

These bits select the frequency of the ADC clock (ADCCLK).

These bits must change when ADCCKCR.ADCCKSRDY = 1.

When switching the division ratio setting from 1/n (n ≠ 1), set MSTPCRD.MSTPD21 = 1 before setting ADCCKCR.ADCCKSREQ = 1.

9.2.52 GPTCKDIVCR : GPT Clock Division Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x05C

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

Bit	Symbol	Function	R/W
3:0	GPTCKDIV[3:0]	GPT Clock (GPTCLK) Division Select 0 0 0 0: 1/1 (value after reset) 0 0 0 1: 1/2 0 0 1 0: 1/4 0 0 1 1: 1/6 0 1 0 0: 1/8 0 1 0 1: 1/3[*1] 0 1 1 0: 1/5[*1] 0 1 1 1: 1/10 1 0 0 0: 1/16 1 0 0 1: 1/32 Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note 1. Do not set when GPT clock is used for ADC16H or PDG.

GPTCKDIVCR controls the GPT clock.

GPTCKDIV[3:0] bits (GPT Clock (GPTCLK) Division Select)

These bits select the frequency of the GPT clock (GPTCLK).

These bits must change when GPTCKCR.GPTCKSRDY = 1.

When switching the division ratio setting from 1/n (n ≠ 1), set all MSTPCRE.MSTPE18 to MSTPE21 and MSTPE27 to MSTPE31 = 1 before setting GPTCKCR.GPTCKSREQ = 1.

9.2.53 SCICKCR : SCI Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x055

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SCICK SRDY	SCICK SREQ	—	—	SCICKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	SCICKSEL[3:0]	SCI Clock (SCICLK) Source Select 0000: HOCO 0001: MOCO (value after reset) 0010: LOCO 0011: Main clock oscillator 0100: Sub-clock oscillator 0101: PLL1P 0110: PLL2P 0111: PLL1Q 1000: PLL1R 1001: PLL2Q 1010: PLL2R Others: Setting prohibited	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	SCICKSREQ	SCI Clock (SCICLK) Switching Request 0: No request 1: Request switching	R/W
7	SCICKSRDY	SCI Clock (SCICLK) Switching Ready state flag 0: Impossible to switch 1: Possible to switch	R

Note: S-TYPE-3, P-TYPE-2

SCICKCR controls the SCI clock.

To change the set value of SCICKDIVCR.SCICKDIV[3:0] and SCICKSEL[3:0], follow the procedure below.

When switching the clock source, it is necessary that the clock before switching and the clock after switching are stably output.

Clock selection switching procedure:

1. Write 1 to all MSTPCRB.MSTPB22 to MSTPB31 only when switching the division ratio setting from 1/n (n ≠ 1).
2. Wait for two SCICLK only when switching the division ratio setting from 1/n (n ≠ 1).
3. Write 1 to SCICKSREQ.
4. Polling until SCICKSRDY is read as 1. While SCICKSRDY = 1, no clock is output to SCICLK.
5. Write the setting value to SCICKDIVCR.SCICKDIV[3:0] and SCICKSEL[3:0].
6. Write 0 to SCICKSREQ.
7. Polling until SCICKSRDY is read as 0.
8. When SCICKSRDY becomes 0, SCICLK starts outputting. Clock switching complete.

When a transition to Software Standby or Deep Software Standby, do not execute the WFI instruction while performing clock selection switching. In other words, do not execute the WFI instruction with SCICKSREQ = 1 and SCICKSRDY = 0, or SCICKSREQ = 0 and SCICKSRDY = 1.

SCICKSEL[3:0] bits (SCI Clock (SCICLK) Source Select)

These bits select the clock source of the SCI clock (SCICLK) and must be modified when SCICKCR.SCICKSRDY = 1.

Do not stop the oscillator selected by these bits except when MOCO is selected.

SCICKSREQ bit (SCI Clock (SCICLK) Switching Request)

This bit selects the SCICLK switching request.

SCICKSRDY bit (SCI Clock (SCICLK) Switching Ready state flag)

This flag indicates the state of switching ready for the SCICLK.

While SCICKSRDY = 1, no clock is output to SCICLK.

9.2.54 SPICKCR : SPI Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x057

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SPICK SRDY	SPICK SREQ	—	—	SPICKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	SPICKSEL[3:0]	SPI Clock (SPICLK) Source Select 0000: HOCO 0001: MOCO (value after reset) 0010: LOCO 0011: Main clock oscillator 0100: Sub-clock oscillator 0101: PLL1P 0110: PLL2P 0111: PLL1Q 1000: PLL1R 1001: PLL2Q 1010: PLL2R Others: Setting prohibited	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	SPICKSREQ	SPI Clock (SPICLK) Switching Request 0: No request 1: Request switching	R/W
7	SPICKSRDY	SPI clock (SPICLK) Switching Ready State Flag 0: Impossible to switch 1: Possible to switch	R

Note: S-TYPE-3, P-TYPE-2

SPICKCR controls the SPI clock.

To change the set value of SPICKDIVCR.SPICKDIV[3:0] and SPICKSEL[3:0], follow the procedure below.

When switching the clock source, it is necessary that the clock before switching and the clock after switching are stably output.

Clock selection switching procedure:

1. Write 1 to MSTPCRB.MSTPB18 and MSTPCRB.MSTPB19 only when switching the division ratio setting from 1/n (n ≠ 1).
2. Wait for two SPICLK only when switching the division ratio setting from 1/n (n ≠ 1).
3. Write 1 to SPICKSREQ.
4. Polling until SPICKSRDY is read as 1. While SPICKSRDY = 1, no clock is output to SPICLK.
5. Write the setting value to SPICKDIVCR.SPICKDIV[3:0] and SPICKSEL[3:0].
6. Write 0 to SPICKSREQ.
7. Polling until SPICKSRDY is read as 0.

8. When SPICKSRDY becomes 0, SPICLK starts outputting. Clock switching complete.

When a transition to Software Standby or Deep Software Standby, do not execute the WFI instruction while performing clock selection switching. In other words, do not execute the WFI instruction with SPICKSREQ = 1 and SPICKSRDY = 0, or SPICKSREQ = 0 and SPICKSRDY = 1.

SPICKSEL[3:0] bits (SPI Clock (SPICLK) Source Select)

These bits select the clock source of the SPI clock (SPICLK) and must be modified when SPICKCR.SPICKSRDY = 1.

Do not stop the oscillator selected by these bits except when MOCO is selected.

SPICKSREQ bit (SPI Clock (SPICLK) Switching Request)

This bit selects the SPICLK switching request.

SPICKSRDY bit (SPI clock (SPICLK) Switching Ready State Flag)

This flag indicates the state of switching ready for the SPICLK.

While SPICKSRDY = 1, no clock is output to SPICLK.

9.2.55 ADCCKCR : ADC Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x05B

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ADCCKSRDY	ADCCKSREQ	—	—	ADCCKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	ADCCKSEL[3:0]	ADC Clock (ADCCLK) Source Select 0 0 0 0: HOCO 0 0 0 1: MOCO (value after reset) 0 0 1 0: LOCO 0 0 1 1: Main clock oscillator 0 1 0 0: Sub-clock oscillator 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Others: Setting prohibited	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	ADCCKSREQ	ADC Clock (ADCCLK) Switching Request 0: No request 1: Request switching	R/W
7	ADCCKSRDY	ADC Clock (ADCCLK) Switching Ready State Flag 0: Impossible to switch 1: Possible to switch	R*1

Note: S-TYPE-3, P-TYPE-2
Note 1. This bit is read only.

ADCCKCR controls the ADCCLK.

To change the set value of ADCCKDIVCR.ADCCKDIV[3:0] and ADCCKSEL[3:0], follow the procedure below.

When switching the clock source, it is necessary that the clock before switching and the clock after switching are stably output.

Clock selection switching procedure:

1. Write 1 to MSTPCRD.MSTPD21 only when switching the division ratio setting from 1/n (n ≠ 1).
2. Wait for two ADCCLK only when switching the division ratio setting from 1/n (n ≠ 1).
3. Write 1 to ADCCKSREQ.
4. Polling until ADCCKSRDY is read as 1.
   While ADCCKSRDY = 1, no clock is output to the ADCCLK.
5. Write the setting value to ADCCKDIVCR.ADCCKDIV[3:0] and ADCCKSEL[3:0].
6. Write 0 to ADCCKSREQ.
7. Polling until ADCCKSRDY is read as 0.
   When ADCCKSRDY becomes 0, the ADCCLK starts outputting.
8. Clock switching complete.

When a transition to Software Standby or Deep Software Standby, do not execute the WFI instruction while performing clock selection switching. In other words, do not execute the WFI instruction with ADCCKSREQ = 1 and ADCCKSRDY = 0, or ADCCKSREQ = 0 and ADCCKSRDY = 1.

ADCCKSEL[3:0] bits (ADC Clock (ADCCLK) Source Select)

These bits select the clock source of the ADCCLK.

These bits must change when ADCCKCR.ADCCKSRDY = 1.

Do not stop the oscillator selected by these bits except when MOCO is selected.

ADCCKSREQ bit (ADC Clock (ADCCLK) Switching Request)

This bit selects the ADCCLK switching request.

ADCCKSRDY flag (ADC Clock (ADCCLK) Switching Ready State Flag)

This flag indicates the state of switching ready for the ADCCLK.

While ADCCKSRDY = 1, no clock is output to the ADCCLK.

9.2.56 GPTCKCR : GPT Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x05D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	GPTCKSRDY	GPTCKSREQ	—	—	GPTCKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	GPTCKSEL[3:0]	GPT Clock (GPTCLK) Source Select 0 0 0 0: HOCO 0 0 0 1: MOCO (value after reset) 0 0 1 1: Main clock oscillator 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Others: Setting prohibited.	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	GPTCKSREQ	GPT Clock (GPTCLK) Switching Request 0: No request 1: Request switching	R/W

Bit	Symbol	Function	R/W
7	GPTCKSRDY	GPT Clock (GPTCLK) Switching Ready State Flag 0: Impossible to switch 1: Possible to switch	R/W*1

Note: S-TYPE-3, P-TYPE-2
Note 1. This bit is read only.

GPTCKCR controls the GPTCLK.

To change the set value of GPTCKDIVCR.GPTCKDIV[3:0] and GPTCKSEL[3:0], follow the procedure below.

When switching the clock source, it is necessary that the clock before switching and the clock after switching are stably output.

Clock selection switching procedure:

1. Write 1 to all MSTPCRE.MSTPE18 to MSTPE21 and MSTPE27 to MSTPE31 only when switching the division ratio setting from 1/n (n ≠ 1).
2. Wait for two GPTCLK only when switching the division ratio setting from 1/n (n ≠ 1).
3. Write 1 to GPTCKSREQ.
4. Polling until GPTCKSRDY is read as 1.
   While GPTCKSRDY = 1, no clock is output to the GPTCLK.
5. Write the setting value to GPTCKDIVCR.GPTCKDIV[3:0] and GPTCKSEL[3:0].
6. Write 0 to GPTCKSREQ.
7. Polling until GPTCKSRDY is read as 0.
   When GPTCKSRDY becomes 0, the GPTCLK starts outputting.
8. Clock switching complete.

When a transition to Software Standby or Deep Software Standby, do not execute the WFI instruction while performing clock selection switching. In other words, do not execute the WFI instruction with GPTCKSREQ = 1 and GPTCKSRDY = 0, or GPTCKSREQ = 0 and GPTCKSRDY = 1.

GPTCKSEL[3:0] bits (GPT Clock (GPTCLK) Source Select)

These bits select the clock source of GPTCLK.

These bits must change when GPTCKCR.GPTCKSRDY = 1.

Do not stop the oscillator selected by these bits except when MOCO is selected.

GPTCKSREQ bit (GPT Clock (GPTCLK) Switching Request)

This bit selects the GPTCLK switching request.

GPTCKSRDY flag (GPT Clock (GPTCLK) Switching Ready State Flag)

This flag indicates the state of switching ready for the GPTCLK.

While GPTCKSRDY = 1, no clock is output to the GPTCLK.

9.2.57 TRCKCR : Trace Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x03F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TRCKEN	—	—	TRCKSEL	TRCK[3:0]			
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
3:0	TRCK[3:0]	Trace Clock Operating Frequency Select 0000: 1/1 0001: 1/2 0010: 1/4 (value after reset) 0011: 1/8 0100: 1/16 0101: 1/32 0110: 1/64 0111: 1/128 1000: 1/256 1001: 1/3 1010: 1/6 1011: 1/12 Others: Setting prohibited	R/W
4	TRCKSEL	Trace Clock Source Select 0: System clock source (value after reset) 1: HOCO (oscillation in debug mode)	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	TRCKEN	Trace Clock Operating Enable 0: Stop 1: Operation enable	R/W

Note: S-TYPE5, P-TYPE-2
Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note: Trace clock must be enabled when accessing TPIU registers.

Trace Clock Control Register controls switching the trace clock.

This register is not controlled by CGFSAR register.

When changing the frequency of the Trace Clock, the trace clock must be stopped (TRCKCR.TRCKEN = 0).

TRCK[3:0] bit (Trace Clock Operating Frequency Select)

These bits select the frequency of the trace clock.

This bit must change when TRCKCR.TRCKEN = 0.

TRCKSEL bit (Trace Clock Source Select)

This bit selects the clock source of the trace clock. According to this bit, the trace clock is as follows:

[When this bit = 0]

- The system clock source is selected as the clock source for the trace clock.
  The system clock source is selected by SCKSCR.CKSEL[2:0] bits.
- The clock supply to the trace clock is selected of MOCO divided by 1 when in Software Standby mode or Deep Software Standby mode.

[When this bit = 1]

- The HOCO is selected as the clock source for the trace clock.
- If TRCKSEL is set to 1, HOCO is operated depending on the state of HOCOCR.HCSTP. However, OSCSF does not confirm that the swing is stable, so wait for the stable wait time.
- HOCO continues to oscillate and supply the trace clock even if in Software Standby mode or Deep Software Standby mode.

This bit must change when TRCKCR.TRCKEN = 0.

TRCKEN bit (Trace Clock Operating Enable)

This bit enables or disables the clock supply of the trace clock.

TRCK[3:0] bits and TRCKSEL bit must be set before changing this bit from 0 (stop) to 1 (enable).

When changing this bit from 1 (enable) to 0 (stop), the trace function must be disabled.

The trace clock is output with this bit set to 1 and under the following conditions:

- CDBGPWRUPREQ = 1 and AL ≠ AL0
- DBGCTRL0.REGEN = 1 and OFS1.SWDBG = 0.

9.2.58 BCKADIVCR : Asynchronous External Bus Clock Division Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x0D4

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

Bit	Symbol	Function	R/W
3:0	BCKACKDIV[3:0]	Asynchronous External Bus Clock (BCLKA) Division Select 0 0 0 0: 1/1 (value after reset) 0 0 0 1: 1/2 0 0 1 0: 1/4 0 0 1 1: 1/6 0 1 0 0: 1/8 0 1 1 1: 1/10 1 0 0 0: 1/16 1 0 0 1: 1/32 Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

BCKADIVCR controls the asynchronous external bus clock.

BCKACKDIV[3:0] bits (Asynchronous External Bus Clock (BCLKA) Division Select)

These bits select the frequency of the asynchronous external bus clock (BCLKA).

These bits must change when BCKACR.BCKACKSRDY = 1.

When switching the division ratio setting from 1/n (n ≠ 1), set EBCKOCR.EBCKOEN = 0 and SDCKOCR.SDCKOEN = 0 before setting BCKACR.BCKACKSREQ = 1. In addition, set BCKCR.EBCKASEL = 0 to select the synchronous (BCLK) side before switching the division ratio.

9.2.59 ESWCKDIVCR : EtherSW Clock Division Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x0D5

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

Bit	Symbol	Function	R/W
3:0	ESWCKDIV[3:0]	EtherSW Clock (ESWCLK) Division Select 0 0 0 0: 1/1 (value after reset) 0 0 0 1: 1/2 0 0 1 0: 1/4 0 0 1 1: 1/6 0 1 0 0: 1/8 0 1 1 1: 1/10 1 0 0 0: 1/16 1 0 0 1: 1/32 Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

ESWCKDIVCR controls the EtherSW clock.

ESWCKDIV[3:0] bits (EtherSW Clock (ESWCLK) Division Select)

These bits select the frequency of the EtherSW clock (ESWCLK).

These bits must change when ESWCKCR.ESWCKSRDY = 1.

When switching the division ratio setting from 1/n (n ≠ 1), set MSTPCRC.MSTPC30 = 1 before setting ESWCKCR.ESWCKSREQ = 1.

9.2.60 ESWPCKDIVCR : EtherSW-PHY Clock Division Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x0D6

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

Bit	Symbol	Function	R/W
3:0	ESWPCKDIV[3:0]	EtherSW-PHY Clock (ESWPHYCLK) Division Select 0 0 0 0: 1/1 (value after reset) 0 0 0 1: 1/2 0 0 1 0: 1/4 0 0 1 1: 1/6 0 1 0 0: 1/8 0 1 1 1: 1/10 1 0 0 0: 1/16 1 0 0 1: 1/32 Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

ESWPCKDIVCR controls the EtherSW-PHY clock.

ESWPCKDIV[3:0] bits (EtherSW-PHY Clock (ESWPHYCLK) Division Select)

These bits select the frequency of the EtherSW-PHY clock (ESWPHYCLK).

These bits must change when ESWPCKCR.ESWPCKSRDY = 1.

When switching the division ratio setting from 1/n (n ≠ 1), set MSTPCRC.MSTPC30 = 1 before setting ESWPCKCR.ESWPCKSREQ = 1.

9.2.61 ETHPCKDIVCR : Ether-PHY Clock Division Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x0D8

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

Bit	Symbol	Function	R/W
3:0	ETHPCKDIV[3:0]	Ether-PHY Clock (ETHPHYCLK) Division Select 0 0 0 0: 1/1 (value after reset) 0 0 0 1: 1/2 0 0 1 0: 1/4 0 0 1 1: 1/6 0 1 0 0: 1/8 0 1 1 1: 1/10 1 0 0 0: 1/16 1 0 0 1: 1/32 Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

ETHPCKDIVCR controls the Ether-PHY clock.

ETHPCKDIV[3:0] bits (Ether-PHY Clock (ETHPHYCLK) Division Select)

These bits select the frequency of the Ether-PHY clock (ETHPHYCLK).

These bits must change when ETHPCKCR.ETHPCKSRDY = 1.

When switching the division ratio setting from 1/n (n ≠ 1), set MSTPCRC.MSTPC28 = 1 before setting ETHPCKCR.ETHPCKSREQ = 1.

9.2.62 BCKACR : Asynchronous External Bus Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x0DA

Bit position:	7	6	5	4	3	2	1	0
Bit field:	BCKACKSRDY	BCKACKSREQ	—	—	BCKACKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	BCKACKSEL[3:0]	Asynchronous External Bus Clock (BCLKA) 0 0 0 1: MOCO (value after reset) 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Others: Setting prohibited.	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	BCKACKSREQ	Asynchronous External Bus Clock (BCLKA) Switching Request 0: No request 1: Request switching	R/W

Bit	Symbol	Function	R/W
7	BCKACKSRDY	Asynchronous External Bus Clock (BCLKA) Switching Ready State Flag 0: Impossible to switch 1: Possible to switch	R*1

Note: S-TYPE-3, P-TYPE-2
Note 1. This bit is read only.

BCKACR controls the BCLKA.

To change the set value of BCKADIVCR.BCKACKDIV[3:0] and BCKACKSEL[3:0], follow the procedure below.

When switching the clock source, it is necessary that the clock before switching and the clock after switching are stably output.

Clock selection switching procedure:

1. Write 0 to EBCKOCR.EBCKOEN and SDCKOCR.SDCKOEN only when switching the division ratio setting from 1/n (n ≠ 1).
2. Write 1 to BCKACKSREQ.
3. Polling until BCKACKSRDY is read as 1.
   While BCKACKSRDY = 1, no clock is output to the BCLKA.
4. Write 0 to BCKCR.EBCKASEL only when switching the division ratio setting from 1/n (n ≠ 1).
5. Write the setting value to BCKCR.BCLKDIV, BCKCR.EBCKASEL, BCKADIVCR.BCKACKDIV[3:0] and BCKACKSEL[3:0].
6. Write 0 to BCKACKSREQ.
7. Polling until BCKACKSRDY is read as 0.
   When BCKACKSRDY becomes 0, the BCLKA starts outputting.
8. Clock switching complete.

When a transition to Software Standby or Deep Software Standby, do not execute the WFI instruction while performing clock selection switching. In other words, do not execute the WFI instruction with BCKACKSREQ = 1 and BCKACKSRDY = 0, or BCKACKSREQ = 0 and BCKACKSRDY = 1.

BCKACKSEL[3:0] bits (Asynchronous External Bus Clock (BCLKA))

These bits select the clock source of the BCLKA.

These bits must change when BCKACR.BCKACKSRDY = 1.

Do not stop the oscillator selected by these bits except when MOCO is selected

BCKACKSREQ bit (Asynchronous External Bus Clock (BCLKA) Switching Request)

This bit selects the BCLKA switching request.

BCKACKSRDY flag (Asynchronous External Bus Clock (BCLKA) Switching Ready State Flag)

This flag indicates the state of switching ready for the BCLKA.

While BCKACKSRDY = 1, no clock is output to the BCLKA.

9.2.63 ESWCKCR : EtherSW Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x0DB

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ESWCKSRDY	ESWCKSREQ	—	—	ESWCKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	ESWCKSEL[3:0]	EtherSW Clock (ESWCLK) Source Select 0 0 0 1: MOCO (value after reset) 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Others: Setting prohibited	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	ESWCKSREQ	EtherSW Clock (ESWCLK) Switching Request 0: No request 1: Request switching	R/W
7	ESWCKSRDY	EtherSW Clock (ESWCLK) Switching Ready State Flag 0: Impossible to switch 1: Possible to switch	R*1

Note: S-TYPE-3, P-TYPE-2
Note 1. This bit is read only.

ESWCKCR controls the ESWCLK.

To change the set value of ESWCKDIVCR.ESWCKDIV[3:0] and ESWCKSEL[3:0], follow the procedure below.

When switching the clock source, it is necessary that the clock before switching and the clock after switching are stably output.

Clock selection switching procedure:

1. Write 1 to MSTPCRC.MSTPC30 only when switching the division ratio setting from 1/n (n ≠ 1).
2. Wait for two ESWCLK only when switching the division ratio setting from 1/n (n ≠ 1).
3. Write 1 to ESWCKSREQ.
4. Polling until ESWCKSRDY is read as 1.
   While ESWCKSRDY = 1, no clock is output to the ESWCLK.
5. Write the setting value to ESWCKDIVCR.ESWCKDIV[3:0] and ESWCKSEL[3:0].
6. Write 0 to ESWCKSREQ.
7. Polling until ESWCKSRDY is read as 0.
   When ESWCKSRDY becomes 0, the ESWCLK starts outputting.
8. Clock switching complete.

When a transition to Software Standby or Deep Software Standby, do not execute the WFI instruction while performing clock selection switching. In other words, do not execute the WFI instruction with ESWCKSREQ = 1 and ESWCKSRDY = 0, or ESWCKSREQ = 0 and ESWCKSRDY = 1.

ESWCKSEL[3:0] bits (EtherSW Clock (ESWCLK) Source Select)

These bits select the clock source of the ESWCLK.

These bits must change when ESWCKCR.ESWCKSRDY = 1.

Do not stop the oscillator selected by these bits except when MOCO is selected.

ESWCKSREQ bit (EtherSW Clock (ESWCLK) Switching Request)

This bit selects the ESWCLK switching request.

ESWCKSRDY flag (EtherSW Clock (ESWCLK) Switching Ready State Flag)

This flag indicates the state of switching ready for the ESWCLK.

While ESWCKSRDY = 1, no clock is output to the ESWCLK.

9.2.64 ESWPCKCR : EtherSW-PHY Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x0DC

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ESWPCKSRDY	ESWPCKSREQ	—	—	ESWPCKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	ESWPCKSEL[3:0]	EtherSW-PHY Clock (ESWPHYCLK) Source Select 0 0 0 1: MOCO (value after reset) 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Others: Setting prohibited	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	ESWPCKSREQ	EtherSW-PHY Clock (ESWPHYCLK) Switching Request 0: No request 1: Request switching	R/W
7	ESWPCKSRDY	EtherSW-PHY Clock (ESWPHYCLK) Switching Ready State Flag 0: Impossible to switch 1: Possible to switch	R*1

Note: S-TYPE-3, P-TYPE-2
Note 1. This bit is read only.

ESWPCKCR controls the ESWPHYCLK.

To change the set value of ESWPCKDIVCR.ESWPCKDIV[3:0] and ESWPCKSEL[3:0], follow the procedure below.

When switching the clock source, it is necessary that the clock before switching and the clock after switching are stably output.

Clock selection switching procedure

1. Write 1 to MSTPCRC.MSTPC30 only when switching the division ratio setting from 1/n (n ≠ 1).
2. Wait for two ESWPHYCLK only when switching the division ratio setting from 1/n (n ≠ 1).
3. Write 1 to ESWPCKSREQ.
4. Polling until ESWPCKSRDY is read as 1.
   While ESWPCKSRDY = 1, no clock is output to the ESWPHYCLK.
5. Write the setting value to ESWPCKDIVCR.ESWPCKDIV[3:0] and ESWPCKSEL[3:0].
6. Write 0 to ESWPCKSREQ.
7. Polling until ESWPCKSRDY is read as 0.
   When ESWPCKSRDY becomes 0, the ESWPHYCLK starts outputting.
8. Clock switching complete.

When a transition to Software Standby or Deep Software Standby, do not execute the WFI instruction while performing clock selection switching. In other words, do not execute the WFI instruction with ESWPCKSREQ = 1 and ESWPCKSRDY = 0, or ESWPCKSREQ = 0 and ESWPCKSRDY = 1.

ESWPCKSEL[3:0] bits (EtherSW-PHY Clock (ESWPHYCLK) Source Select)

These bits select the clock source of the ESWPHYCLK.

These bits must change when ESWPCKCR.ESWPCKSRDY = 1.

Do not stop the oscillator selected by these bits except when MOCO is selected.

ESWPCKSREQ bit (EtherSW-PHY Clock (ESWPHYCLK) Switching Request)

This bit selects the ESWPHYCLK switching request.

ESWPCKSRDY flag (EtherSW-PHY Clock (ESWPHYCLK) Switching Ready State Flag)

This flag indicates the state of switching ready for the ESWPHYCLK.

While ESWPCKSRDY = 1, no clock is output to the ESWPHYCLK.

9.2.65 ETHPCKCR : Ether-PHY Clock Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x0DE

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ETHPCKSRDY	ETHPCKSREQ	—	—	ETHPCKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	ETHPCKSEL[3:0]	Ether-PHY Clock (ETHPHYCLK) Source Select 0 0 0 1: MOCO (value after reset) 0 0 1 1: Main clock oscillator 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Others: Setting prohibited	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	ETHPCKSREQ	Ether-PHY Clock (ETHPHYCLK) Switching Request 0: No request 1: Request switching	R/W
7	ETHPCKSRDY	Ether-PHY Clock (ETHPHYCLK) Switching Ready State Flag 0: Impossible to switch 1: Possible to switch	R*1

Note: S-TYPE-3, P-TYPE-2
Note 1. This bit is read only.

ETHPCKCR controls the ETHPHYCLK.

To change the set value of ETHPCKDIVCR.ETHPCKDIV[3:0] and ETHPCKSEL[3:0], follow the procedure below.

When switching the clock source, it is necessary that the clock before switching and the clock after switching are stably output.

Clock selection switching procedure:

1. Write 1 to MSTPCRC.MSTPC28 only when switching the division ratio setting from 1/n (n ≠ 1).
2. Wait for two ETHPHYCLK only when switching the division ratio setting from 1/n (n ≠ 1).
3. Write 1 to ETHPCKSREQ.
4. Polling until ETHPCKSRDY is read as 1.
   While ETHPCKSRDY = 1, no clock is output to the ETHPHYCLK.
5. Write the setting value to ETHPCKDIVCR.ETHPCKDIV[3:0] and ETHPCKSEL[3:0].
6. Write 0 to ETHPCKSREQ.
7. Polling until ETHPCKSRDY is read as 0.

When ETHPCKSRDY becomes 0, the ETHPHYCLK starts outputting.

8. Clock switching complete.

When a transition to Software Standby or Deep Software Standby, do not execute the WFI instruction while performing clock selection switching. In other words, do not execute the WFI instruction with ETHPCKSREQ = 1 and ETHPCKSRDY = 0, or ETHPCKSREQ = 0 and ETHPCKSRDY = 1.

ETHPCKSEL[3:0] bits (Ether-PHY Clock (ETHPHYCLK) Source Select)

These bits select the clock source of the ETHPHYCLK.

These bits must change when ETHPCKCR.ETHPCKSRDY = 1.

Do not stop the oscillator selected by these bits except when MOCO is selected.

ETHPCKSREQ bit (Ether-PHY Clock (ETHPHYCLK) Switching Request)

This bit selects the ETHPHYCLK switching request.

ETHPCKSRDY flag (Ether-PHY Clock (ETHPHYCLK) Switching Ready State Flag)

This flag indicates the state of switching ready for the ETHPHYCLK.

While ETHPCKSRDY = 1, no clock is output to the ETHPHYCLK.

9.2.66 USBCKSELR : USB Clock Selection Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x0F0

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

Bit	Symbol	Function	R/W
0	UCKSEL	USB Clock (USBCLK) Selection 0: Disable (value after reset) 1: Enable	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Rewrite the USBCKSELR register while the SYSCFG.SCKE bit is 0.

UCKSEL bit (USB Clock (USBCLK) Selection)

Set the UCKSEL bit depending on other register settings shown in Table 9.5.

The UCKSEL bit can only be set to 1 when USBFS is used as the device controller. Set the UCKSEL bit to 0 to use the host controller and the On-The-Go (OTG) function.

The HOCO user trimming function cannot be used when the UCKSEL bit is 1. Set the HOCOUTCR.HOCOUTRIM[7:0] bits to 0x00.

Table 9.5 UCKSEL bit setting

UCKSEL	USBCKCR.USBCKSEL[3:0]	FLLCR1.FLLEN
0	Other than 0000 (Other than HOCO)	—
	0000 (HOCO)	1 (FLL enabled)
1	0000 (HOCO)	0 (FLL disabled)

9.3 Main Clock Oscillator

To supply the clock signal to the main clock oscillator, use one of the following ways:

- Connect an oscillator
- Connect the input of an external clock signal.

9.3.1 Connecting a Crystal Resonator

Figure 9.2 shows an example of connecting a crystal resonator. A damping resistor (Rd) can be added, if required.

Because the resistor values vary according to the resonator and the oscillation drive capability, use values recommended by the resonator manufacturer. If the manufacturer recommends using an external feedback resistor (Rf), insert an Rf between EXTAL and XTAL by following the instructions.

When connecting a resonator to supply the clock, the frequency of the resonator must be in the frequency range of the resonator for the main clock oscillator as described in Table 9.1.

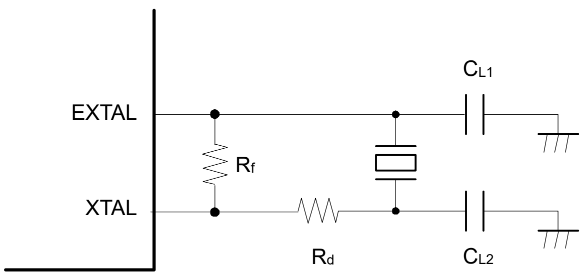


Figure 9.2 Example of crystal resonator connection

Figure 9.3 shows an equivalent circuit of the crystal resonator.

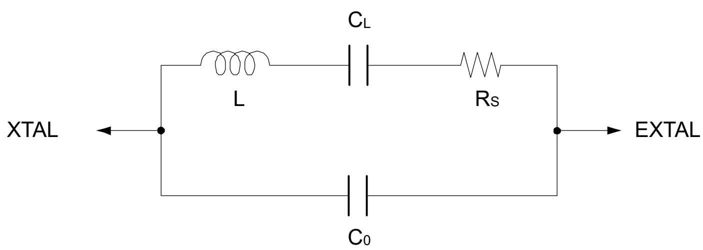


Figure 9.3 Equivalent circuit of the crystal resonator

9.3.2 External Clock Input

Figure 9.4 shows an example of connecting an external clock input. To operate the oscillator with an external clock signal, set the MOMCR.MOSEL bit to 1. The XTAL pin becomes high impedance.

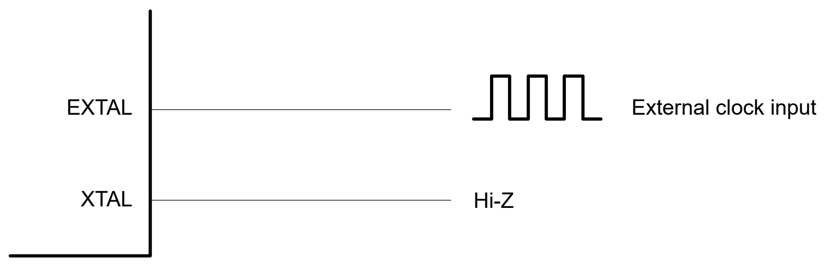


Figure 9.4 Equivalent circuit for external clock

9.3.3 Notes on External Clock Input

The frequency of the external clock input can only be changed when the main clock oscillator is stopped. Do not change the frequency of the external clock input when the setting of the Main Clock Oscillator Stop bit (MOSCCR.MOSTP) is 0.

9.3.4 Note on Oscillation Keep in Software Standby Mode

By using the oscillation keep function of the Main Clock Oscillator in Software Standby mode, the Main Clock Oscillator stabilization wait time after canceling Software standby mode can be shortened.

If MCU enters Software Standby mode under the MOSCSCR.MOSCSOKP bit is set to enabled (MOSCSCR.MOSCSOKP bit is 1) and under the Main Clock Oscillator is operating, the Main Clock Oscillator will keep oscillating in Software Standby mode. When canceling Software Standby mode in this state, the Main clock can be used without the oscillator stabilization wait time.

If using this function, confirm that the setting value is written to the register before entering the Software standby mode.

However, even if this function is enabled, the clock supply to the system and peripheral modules during Software Standby mode is stopped.

9.4 Sub-Clock Oscillator

To supply the clock signal to the sub-clock oscillator, use one of the following ways:

- Connect an oscillator
- Connect the input of an external clock signal.

9.4.1 Connecting a 32.768-kHz Crystal Resonator

To supply a clock to the sub-clock oscillator, connect a 32.768-kHz crystal resonator as shown in Figure 9.5. A damping resistor (Rd) can be added, if necessary. Because the resistor values vary according to the resonator and the oscillation drive capability, use values recommended by the resonator manufacturer. If the resonator manufacturer recommends the use of an external feedback resistor (Rf), insert an Rf between XCIN and XCOUT by following the instructions. When connecting a resonator to supply the clock, the frequency of the resonator must be in the frequency range of the resonator for the sub-clock oscillator as described in Table 9.1.

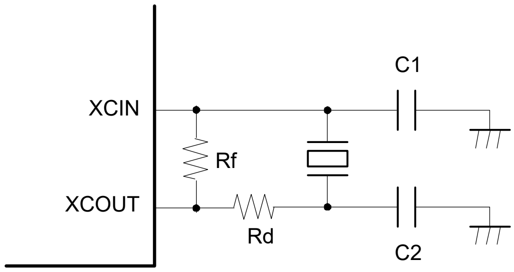


Figure 9.5 Connection example of 32.768-kHz crystal resonator

Figure 9.6 shows an equivalent circuit for the 32.768-kHz crystal resonator.

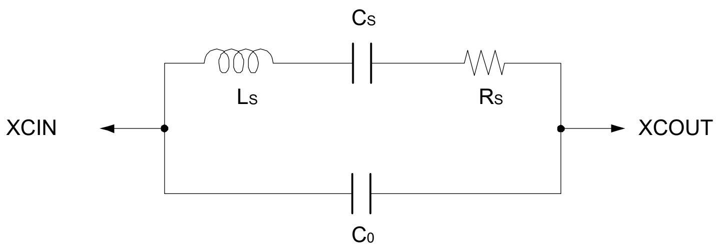


Figure 9.6 Equivalent circuit for the 32.768-kHz crystal resonator

9.4.2 External Clock Input

In the external clock input mode, clock is input from EXCIN pin.

9.5 Low Speed On-chip Oscillator (LOCO)

Low Speed On-chip Oscillator (LOCO) is the internal oscillator that supplies a low-speed clock.

9.5.1 LOCO Operating Conditions

There are restriction on LOCO operation setting. For detail, see section 9.5.3. Restrictions on LOCO Operation and Stopping

LOCO oscillates when any of the following conditions is met.*1

[LOCO operating conditions]

- When LOCO is set to operate in the LOCOCR register (LOCOCR.LCSTP bit is 0).
- When IWDT is set to the auto start mode. (OFS0.IWDTSTRT bit is 0)*2
- When IWDT is set to the register start mode (OFS0.IWDTSTRT bit is 1) and counting is started.*3
- \When waiting for the oscillation stabilization of the Main Clock Oscillator (while MOCOCR.MCSTP bit is 0 and OSCSF.MOSCSF bit is 0).*4

Note 1. LOCO stops at Deep Software Standby mode 2 or Deep Software Standby mode 3.
Note 2. When IWDT is set to the auto start mode, regardless of the setting of IWDTSTPCTL bit in OFS0 register, LOCO continues to oscillate.
Note 3. When IWDT is set to the register start mode, regardless of the setting of SLCSTP bit in IWDTCSTPR register, LOCO continues to oscillate once IWDT starts counting.
Note 4. Even if the Main Clock Oscillator is set to the external clock input mode, LOCO operates because the oscillation stabilization wait circuit operates when the Main Clock Oscillator starts operating.

9.5.2 LOCO Stop Conditions

There are restriction on LOCO stop setting. For detail, see section 9.5.3. Restrictions on LOCO Operation and Stopping.

There are two ways to top LOCO: stop by transition to Low Power mode or stop by LOCOCR register.

(A) LOCO stop conditions related to Low Power mode

LOCO stops when transitioning to the following Low Power mode

- Deep Software Standby mode 2
- Deep Software Standby mode 3

(B) LOCO stop conditions related to LOCOCR register

When LOCO is stopped using the LOCOCR register, even if the LCSTP bit set to stop (LOCOCR.LCSTP bit is 1), LOCO may not be stopped depending on the operating status of IWDT and oscillator.

LOCO can be stopped when all of the following conditions are met.

- LOCO is set to stop using the LOCOCR register (LOCOCR.LCSTP bit is 1)
- IWDT is not operating. (Include the case of stopping in CPU Sleep mode, CPU Deep Sleep mode, Software Standby mode, and Deep Software Standby mode 1 by setting the OFS0.IWDTSTPCTL bit or the IWDTCSTPR.SLCSTP bit.).
- Main Clock Oscillator is stopped or Main Clock oscillation is stable. (MOCOCR.MCSTP bit is 1 or OSCSF.MOSCSF bit is 1)

9.5.3 Restrictions on LOCO Operation and Stopping

After setting the LCSTP bit in LOCOCR to 0 to start the LOCO clock, only use the clock after the LOCO clock oscillation stabilization wait time (tLOCOWT) elapses. A fixed stabilization wait is required after setting the LOCO clock to start operation. A fixed wait for the oscillation to stop is also required.

The following restrictions apply when starting and stopping operation:

- After stopping the LOCO clock, read OSCMONR.LOCOMON, make sure LOCO is stopped, and then set the operation.
- Confirm that LOCO oscillation is stable before stopping the LOCO clock
- Confirm that LOCO oscillation is stable before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode.
- When a transition to Software Standby mode or Deep Software Standby mode after setting LOCO to stop, read OSCMONR.LOCOMON, make sure LOCO is stopped, and executing the WFI instruction.

The LOCO clock is used to measure the oscillation stabilization wait time of the main clock oscillator. While the main clock oscillator is waiting for oscillation stabilization, the LOCO clock oscillates regardless of the LCSTP bit setting. At this time, it is necessary to prevent conflicts between the LOCO stop (or operation) by the LCSTP bit in LOCOCR and the LOCO clock oscillation for the Main clock Oscillator stabilization wait time (or LOCO clock stop after Main clock oscillation stabilization). Therefore, do not change the LOCOCR.LCSTP bit during the period from the 3 LOCO clock cycle before setting the start of main clock oscillator operation until the main clock oscillator stabilization can be confirmed with the oscillation stabilization flag in the OSCSF register.

When using IWDT in the register start mode (OFS0.IWDTSTRT bit is 1), set the LOCO to run (LOCOCR.LCSTP = 0) before IWDT count operation, and use IWDT after LOCO oscillation has stabilized. When IWDT is used in register start mode, do not change the LOCOCR.LCSTP bit during the period from the 3 LOCO clock cycles before IWDT count operation starts until the start of IWDT count operation can be confirmed.

When using the Digital filter or the Reset negate select for Voltage monitor m (m = 1 to 2) in Programmable Voltage Detection (PVDm), set the LOCO to run (LOCOCR.LCSTP = 0) before enabling PVDm and use PVDm after LOCO oscillation has stabilized.

9.6 Middle Speed On-chip Oscillator (MOCO)

Middle Speed On-chip Oscillator (MOCO) is the internal oscillator that supplies a middle-speed clock.

9.6.1 MOCO Operating Conditions

There are restriction on MOCO operation setting. For detail, see section 9.6.3. Restrictions on MOCO Operation and Stopping.

MOCO oscillates when any of the following conditions is met.

[MOCO operating conditions]

- When MOCO is set to operate in the LOCOCR register (MOCOCR.MCSTP = 0)
- When waiting for the HOCO oscillation stabilization (while OFS1.HOCOEN bit is 0 or HOCOCR.HCSTP bit is 0, and OSCSF.HOCOSF bit is 0).
- When waiting for the PLL1 oscillation stabilization (while PLLCR.PLLSTP bit is 0, and OSCSF.PLLSF bit is 0).
- When waiting for the PLL2 oscillation stabilization (while PLL2CR.PLL2STP bit is 0, and OSCSF.PLL2SF bit is 0).
- When MOSC oscillation stop detection function is set to enable.
- When SOSC oscillation stop detection function is set to enable.

Note: MOCO stops at Software Standby mode or Deep Software Standby mode.

9.6.2 MOCO Stop Conditions

There are restriction on MOCO stop setting. For detail, see section 9.6.3. Restrictions on MOCO Operation and Stopping.

There are two ways to top MOCO: stop by transition to Low Power mode or stop by MOCOCR register.

(A) MOCO stop conditions related to Low Power mode

MOCO stops when transitioning to the following Low Power mode.

- Software Standby mode (MOCOSCR.MOCOSOKP=0)
- Deep Software Standby mode

Note: When the On-Chip Debugger function is enabled, MOCO does not stop when transitioning to each standby mode.

(B) MOCO stop conditions related to MOCOCR register

When MOCO is stopped using the MOCOCR register, even if the MCSTP bit set to stop (MOCOCR.MCSTP bit is 1), MOCO may not be stopped depending on the operating status of the system.

MOCO can be stopped when all of the following conditions are met.

[MOCO stop conditions]

- MOCO is set to stop using the MOCOCR register (MOCOCR.MCSTP bit is 1)
- When HOCO is stopped or HOCO oscillation is stable. (OFS1(_SEC).HOCOEN bit is 1 and HOCOCR.HCSTP bit is 1, or OSCSF.HOCOSF bit is 1)
- PLL1 circuit is stopped or PLL1 oscillation is stable. (PLLCR.PLSTP bit is 1 or OSCSF.PLLSF bit is 1)
- PLL2 circuit is stopped or PLL2 oscillation is stable. (PLL2CR.PL2STP bit is 1 or OSCSF.PLL2SF bit is 1)

9.6.3 Restrictions on MOCO Operation and Stopping

After setting the MCSTP bit to 0 to start the MOCO clock, only use the clock after the MOCO clock oscillation stabilization time (tMOCOWT) elapses. A fixed stabilization wait is required after setting the MOCO clock to start operation. A fixed wait for oscillation to stop is also required.

The following restrictions apply when starting and stopping operation.

- After stopping the MOCO clock, read OSCMONR.MOCOMON, make sure MOCO is stopped, and then set the operation.
- Confirm that MOCO oscillation is stable before stopping the MOCO clock.
- Regardless of whether the MOCO clock is selected as the system clock source, confirm that MOCO oscillation is stable before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode. If MOCO is stopped, operate the MOCO.

- Regardless of whether the MOCO clock is selected as the system clock source, confirm that MOCO oscillation is stable before executing the wait instruction to place the CPU in the CPU Deep Sleep mode. If MOCO is stopped, operate the MOCO.
- Confirm that MOCO oscillation is stable before executing a Power Control for each power domain by Software. If MOCO is stopped, operate the MOCO.

Writing 1 to MCSTP is prohibited under the following condition:

- SCKSCR.CKSEL[2:0] = 001b (system clock source = MOCO).
- Oscillation stop detection is enabled (OSTDCR.OSTDE bit is 1).
- While waiting for the HOCO oscillation stabilization (OFS1.HOCOEN bit is 0 or HOCOCR.HCSTP bit is 0, and OSCSF.HOCOSF bit is 0)
- While waiting for the PLL1 oscillation stabilization (PLLCR.PLLSTP bit is 0, and OSCSF.PLLSF bit is 0)
- While waiting for the PLL2 oscillation stabilization (PLL2CR.PLL2STP bit is 0, and OSCSF.PLL2SF bit is 0)
- While executing a Power Control for each power domain by software (During the Power Gating Control)

The MOCO clock is used to measure the oscillation stabilization wait time of the HOCO/PLL1/PLL2 clock. While waiting for the HOCO/PLL1/PLL2 clock to stabilize, the MOCO clock oscillates regardless of the MOCOCR.MCSTP bit setting. At this time, it is necessary to prevent conflicts between the MOCO stop (or operation) by the MCSTP bit in MOCOCR and the MOCO clock oscillation for HOCO/PLL1/PLL2 clock oscillation stabilization wait time (or MOCO clock stop after HOCO/PLL1/PLL2 clock oscillation stabilization). Therefore, do not change the MOCOCR.MCSTP bit during the period from the 3 MOCO clock cycles before the start of HOCO/PLL1/PLL2 clock operation setting until the HOCO/PLL1/PLL2 clock oscillation stability can be confirmed by the HOCOSF/PLLSF/PLL2SF flag in the OSCSF register.

9.7 High Speed On-chip Oscillator (HOCO)

High Speed On-chip Oscillator (HOCO) is the internal oscillator that supplies a high-speed clock.

9.7.1 Frequency Locked Loop Function

HOCO supports the Frequency Locked Loop (FLL) function that corrects the frequency accuracy of the HOCO clock using the Sub-Clock Oscillator. When the FLL function is enabled, the frequency accuracy of the HOCO clock can be improved compared to when the FLL function is disabled.

When using the FLL function, the oscillation of the Sub-Clock Oscillator must be stable. In addition, the sub-clock oscillator must meet the frequency accuracy specified in section 62, Electrical Characteristics.

When enabling the FLL function and using HOCO, enable the FLL function in advance and enable HOCO (HOCOCR.HCTSP=0) after SOSC has stabilized.

Set FLLCR2.FLLCNTL[10:0] as follows for each HOCO frequency setting. And the HOCO clock reaches the desired frequency after tFLLWT after enabling HOCO.

Table 9.6 FLLCNTL setting for each HOCO frequency setting

FLLCR2.FLLCNTL[10:0]	OFS1(_SEC).HOCOFRQ0[2:0]	Target Frequency (FHOCO)
0x1E8	000b	16 MHz
0x225	001b	18 MHz
0x262	010b	20 MHz
0x1E8	100b	32 MHz
0x1E8	111b	48 MHz

9.7.2 Note on Oscillation Keep in Software Standby mode

By using the oscillation keep function of the HOCO in Software Standby mode, the HOCO oscillation stabilization wait time after canceling Software standby mode can be shortened.

To keep the HOCO oscillation in Software Standby mode, make the following settings:

- Enable the HOCO Standby Oscillation Keep function in the HOCOSCR register. (HOCOSCR.HOCOSOKP bit is set to 1)
- The setting of power control for HOCO in HOCOLDOCR.

If MCU enters Software Standby mode under the above settings and under the HOCO is operating, the HOCO will keep oscillating in Software Standby mode. When canceling Software Standby mode in this state, the HOCO clock can be used without the oscillator stabilization wait time.

If using this function, confirm that the setting value is written to the register before entering the Software standby mode.

However, even if this function is enabled, the clock supply to the system and peripheral modules during Software Standby mode is stopped.

9.7.3 Note on using as TRACE clock source

If HOCO is selected as Trace clock source with TRCKCRSEL bit on TRCKCR register, and then Trace clock function is enable (TRCKCR.TRCKEN=1), HOCO oscillates regardless of the value of HCSTP bit on HOCOCR register. In this situation, HOCO will continue to oscillate even if MCU is in Software Standby mode, Deep Software Standby mode, or Reset (except POR and PVD0) to keep supplying the clock to the TRACE module.

If HOCO is used as Trace clock source, Trace clock should be enabled with TRCKEN bit on TRCKCR register while HOCO is stopped (OFS1(_SEC).HOCOEN=1 and HOCOCR.HCSTP=1) or HOCO is oscillating and stable (OSCSF.HOCOSF=1).

9.8 PLL Circuit

The PLL circuit has a function to multiply the frequency from the oscillator.

Figure 9.7 shows the block diagram of PLL1 circuit, and Figure 9.8 shows the PLL2 circuit. PLL1 and PLL2 have the same circuit configuration. The PLL1 circuit is explained the following as an example.

The input clock source selection circuit selects the clock source input to the PLL.

The input clock divider circuit divides the input clock and generates the PLL reference clock.

The frequency control circuit controls the VCO clock to be generated from the PLL reference clock according to the frequency multiplier setting. The frequency multiplication ratio is determined by the combination of the PLLMUL bits and PLLMULNF bits. The PLLMUL bit determines the integer part of the multiplication ratio, and the PLLMULNF bit determines the fractional part of the multiplication ratio. For example, if the PLLMUL bit setting is 26 and the PLLMULNF bit setting is 0.33, the multiplication ratio is 26.33.

The PLL circuit has the 3 output clocks. The output clock divider circuit divides the VCO clock to generate each PLL clock output. The division ratio for generating each output clock can be set independently with the PLODIVP, PLODIVQ, and PLODIVR bits.

The PLL Input clock frequency, the PLL Reference clock frequency, the VCO clock frequency, and each Output clock frequency must be within the specified in section 62, Electrical Characteristics.

The PLL frequency setting can only be changed the PLL circuit is stopped. Do not change the PLL frequency setting when PLL stop control bit (PLLSTP) is 0.

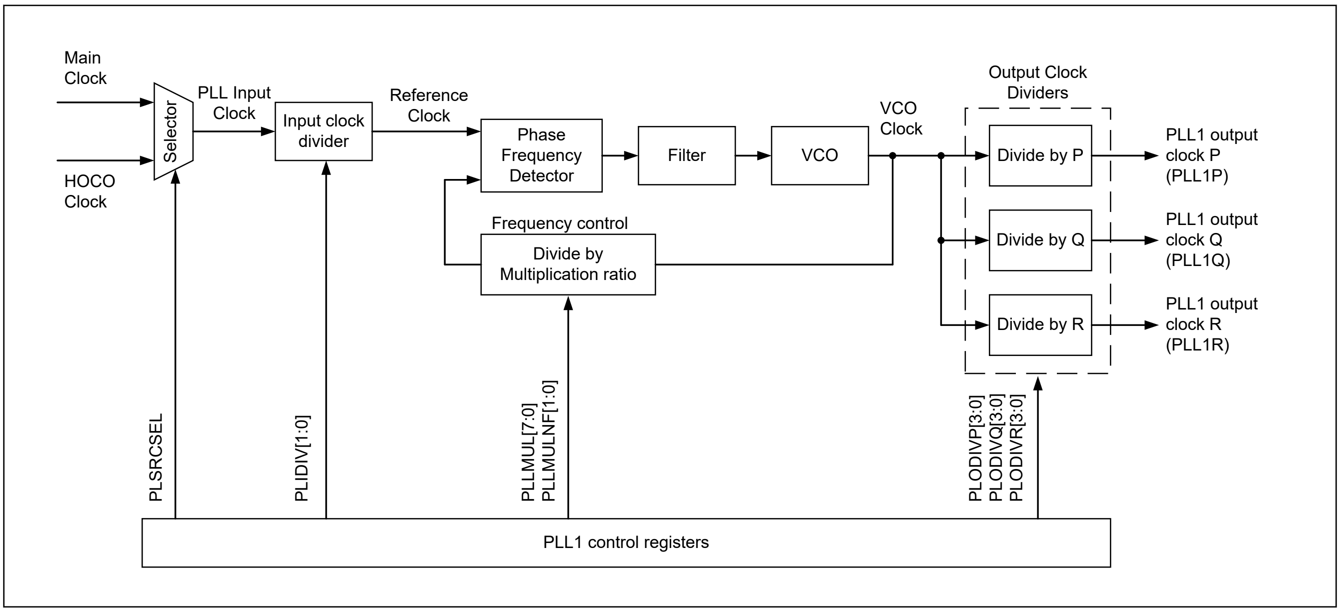


Figure 9.7 PLL1 block diagram

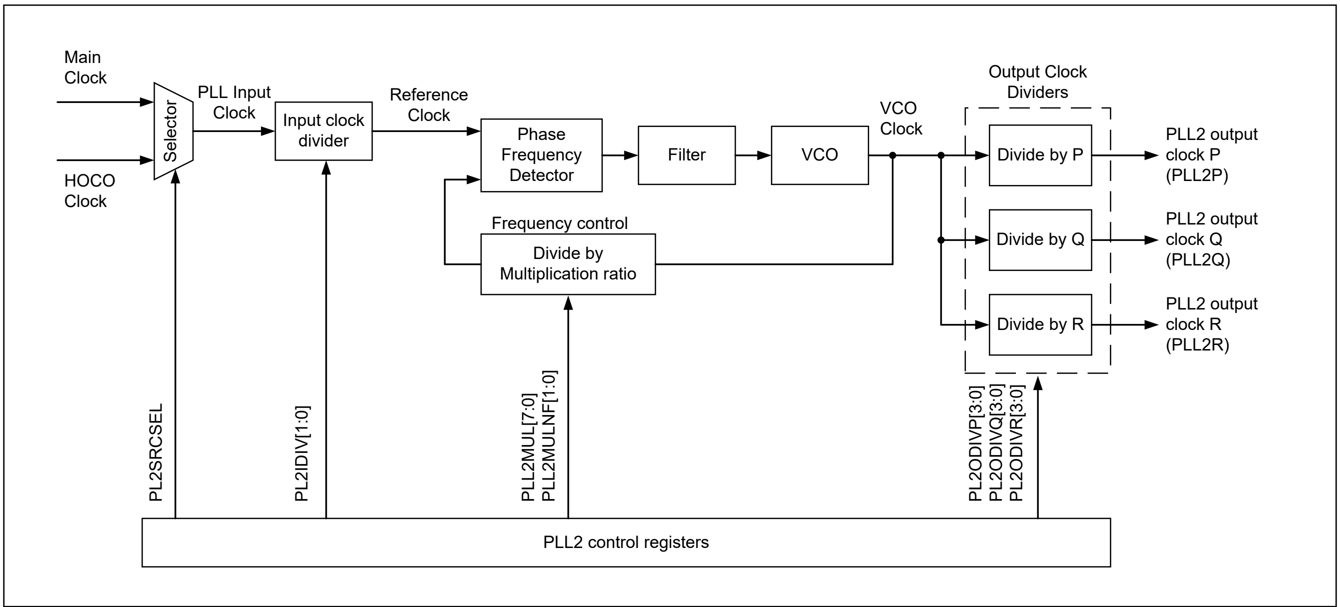


Figure 9.8 PLL2 block diagram

9.9 Oscillation Stop Detection Function

9.9.1 Oscillation Stop Detection and Operation after Detection

The oscillation stop detection function detects the main clock oscillator stop. When oscillation stop is detected, the system clock source switches as follows:

- If an oscillation stop is detected with SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC), the system clock source switches to the MOCO clock.
- If an oscillation stop is detected with PLLCCR.PLSRCSEL = 0 (PLL1 source clock = MOSC) and SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL1P), PLL1 clock remains the system clock source. The frequency becomes free-running, and the setting in the SCKSCR.CKSEL[2:0] bits does not change.

An oscillation stop detection interrupt request can be generated when an oscillation stop is detected. In addition, the General PWM Timer (GPT) output can be forced to stop state on detection.

The main clock oscillation stop is detected when the input clock remains at 0 or 1 for a certain period, for example, when a malfunction occurs in the main clock oscillator. See section 62, Electrical Characteristics.

Switching between the main clock oscillator and the MOCO clock or between the PLL1 clock and PLL1 free-running clock is controlled by the Oscillation Stop Detection Flag (OSTDSR.OSTDF).

OSTDF controls the switched clock as follows:

- When SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC):
  - When OSTDF changes from 0 to 1, the clock source switches to the MOCO clock.
  - When OSTDF changes from 1 to 0, the clock source switches back to MOSC.
- When PLLCCR.PLSRCSEL = 0 (PLL1 source clock = MOSC) and SCKSCR.CKSEL[2:0] = 101b (System clock source = PLL1P):
  - When OSTDF changes 0 to 1, the clock source switches to the PLL1 free-running oscillation clock.
  - When OSTDF changes 1 to 0, the clock source switches back to PLL1.

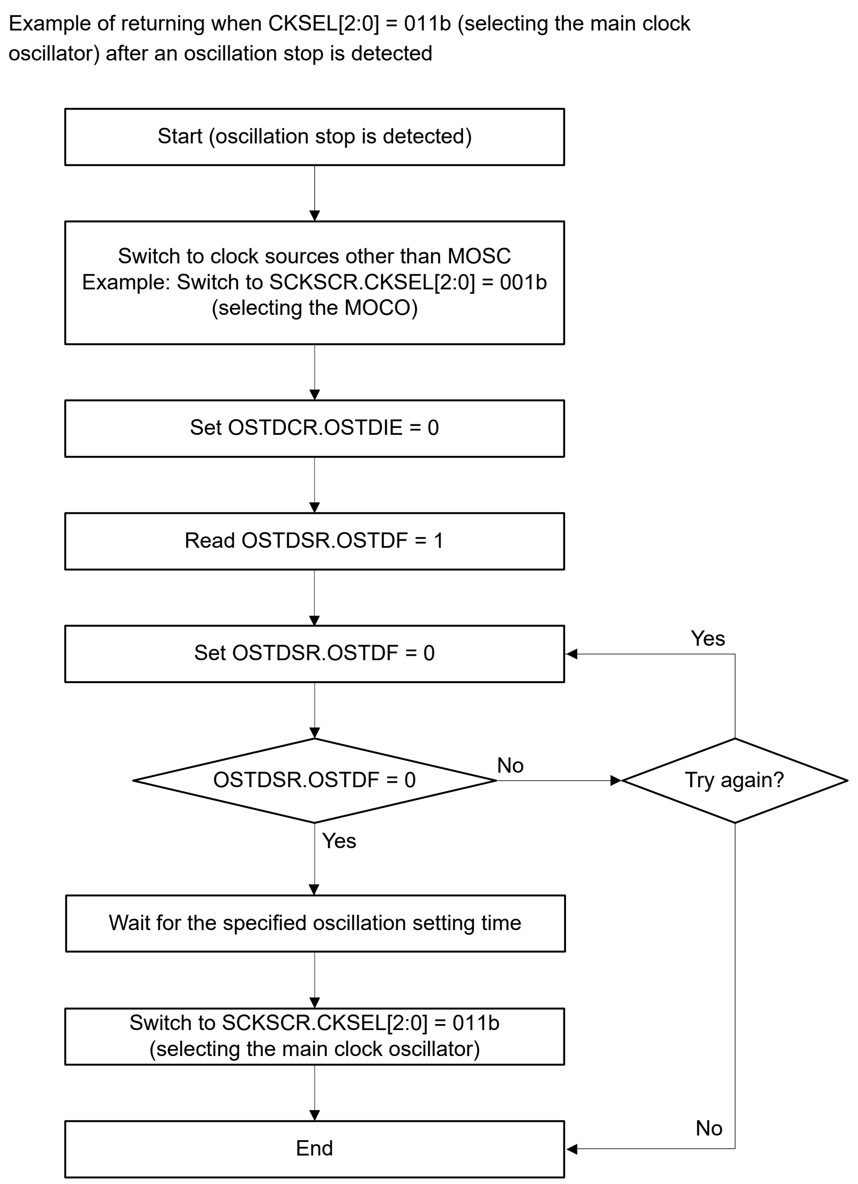
To switch the clock source to the main clock or PLL1 clock again after the oscillation stop detection, set the CKSEL[2:0] bits to a clock source other than the main clock or PLL1 clock and clear the OSTDF flag to 0. Also, check that the OSTDF flag is not 1, then set the CKSEL[2:0] bits to the main clock or PLL1 clock after the specified oscillation stabilization time elapses.

After a reset release, the main clock oscillator is stopped and the oscillation stop detection function is disabled. To enable the oscillation stop detection function, activate the main clock oscillator and write 1 to the oscillation stop detection function enable bit (OSTDCR.OSTDE) after a specified oscillation stabilization time elapses.

The oscillation stop detection function detects when the main clock is stopped by an external cause.

The oscillation stop detection function switches the following clocks to the MOCO clock (when the system clock source is MOSC) or the PLL1 free-running clock (when the system clock source is PLL1P) :

- All clocks that can be selected as the MOSC clock or PLL1 except CLKOUT, USBMCLK and CACMCLK.
- The system clock (ICLK) frequency during the MOCO (when the system clock source is MOSC) or PLL1 free-running (when the system clock source is PLL1P) operation is specified by the MOCO oscillation frequency and the division ratio set by the system clock select bits (SCKDIVCR.ICK[3:0])



Note: On return from the oscillation-stopped state, the factor responsible for stopping the main clock oscillation circuit must be removed from the system to allow oscillation to resume.

Figure 9.9 Flow of recovery on detection of oscillator stop

9.9.2 Oscillation Stop Detection Interrupts

An oscillation stop detection interrupt (MOSC_STOP) is generated when the Oscillation Stop Detection Flag (OSTDSR.OSTDF) is 1 and the Oscillation Stop Detection Interrupt Enable bit in the Oscillation Stop Detection Control Register (OSTDCR.OSTDIE) is 1 (enabled). The Port Output Enable for GPT (POEG) is notified of the main clock oscillator stop. On receiving the notification, the POEG sets the Oscillation Stop Detection Flag in the POEG Group n Setting Register (POEGGn.OSTPF) to 1 (n = A, B, C, D).

After the oscillation stop is detected, wait at least 10 PCLKB clock cycles before writing to the POEGGn.OSTPF flag. When the OSTDSR.OSTDF flag requires clearing, do so after clearing the Oscillation Stop Detection Interrupt Enable bit in the Oscillation Stop Detection Control Register (OSTDCR.OSTDIE). Wait at least 2 PCLKB clock cycles before setting the OSTDCR.OSTDIE bit to 1 again. A longer PCLKB wait time might be required, depending on the number of cycles required to read a given I/O register.

The oscillation stop detection interrupt is a non-maskable interrupt. Because non-maskable interrupts are disabled in the initial state after a reset release, enable non-maskable interrupts through software before using oscillation stop detection interrupts. For details, see section 14, Interrupt Controller Unit (ICU).

9.9.3 Sub-clock Oscillation Stop Detection and Operation after Detection

The sub-clock oscillation stop detection function is used to detect the sub-clock oscillator stop. When an oscillation stop is detected, the system clock switches as follows:

- If an oscillation stop is detected with SCKSCR.CKSEL[2:0] = 100b (system clock source = SOSC), the system clock source switches to the MOCO/256 clock.
  An oscillation stop detection interrupt request can be generated when an oscillation stop is detected.
  The sub-clock oscillation stop is detected when the input clock remains to be 0 or 1 for a certain period, for example, when a malfunction occurs in the sub-clock oscillator. See Electrical Characteristics.
  Switching between the sub-clock oscillator and the MOCO/256 clock is controlled by the Sub-clock Oscillation Stop Detection flag (SOSTDSR.SOSTDF).
  The SOSTDF flag controls the switched clock as the follows:
- SCKSCR.CKSEL[2:0] = 100b (System clock source = SOSC):
  - When SOSTDF changes from 0 to 1, the clock source switches to the MOCO/256 clock.
  - When SOSTDF changes from 1 to 0, the clock source switches to SOSC again.

To switch the clock source to the sub-clock oscillator again after the oscillation stop detection, set the CKSEL[2:0] bits to a clock source other than the sub-clock, and clear the SOSTDF flag to 0. Also, check that the SOSTDF flag is not 1, and then set the CKSEL[2:0] bits to the sub-clock oscillator after the specified oscillation stabilization time elapses.

After a VBAT_POR release, the sub-clock oscillator is stopped and the sub-clock oscillation stop detection function is disabled. To enable the sub-clock oscillation stop detection function, activate the sub-clock oscillator and write 1 to the Sub-clock Oscillation Stop Detection Function Enable bit (SOSTDCR.SOSTDE) after the specified oscillation stabilization time elapses.

After that, be sure to clear the Sub-clock Oscillation Stop Detection Flag (SOSTDSR.SOSTDF) to 0 after the start-up time of the sub-clock oscillation stop detector (Tsosdup) elapses.

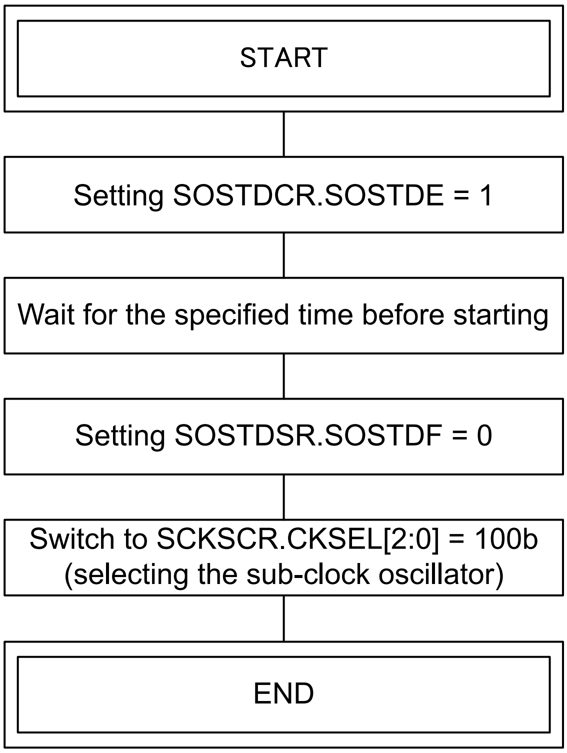
When selecting the sub-clock oscillator for the system clock, complete the above settings before switching.

The sub-clock oscillation stop detection function detects when the Sub-Clock Oscillator is stopped by an external cause.

The sub-clock oscillation stop detection function switches the following clocks to the MOCO/256 clock.

- All clocks that can be selected as the SOSC clock except CACSCLK.
- The system clock (ICLK) frequency during MOCO/256 operation is specified in the MOCO oscillation frequency and the division ratio set in the System Clock Select bits (SCKDIVCR.ICK[3:0]).

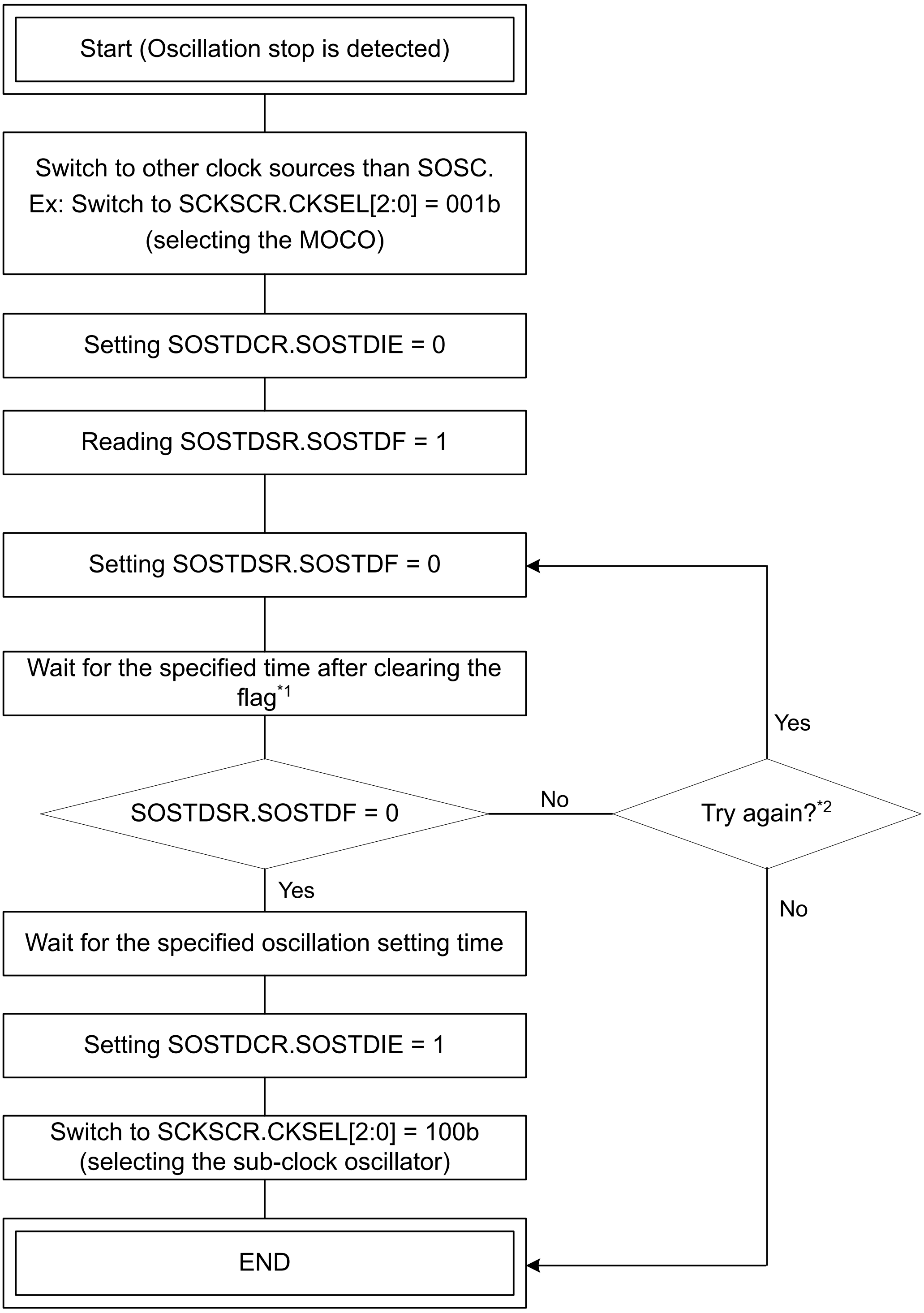
Example of activation of sub-clock oscillation stop detection function when CKSEL[2:0] = 100b (selecting the sub-clock oscillator)



Note: Set the system clock to the other clock sources than the sub-clock oscillator before executing this flow.

Figure 9.10 Flow of Activation of Sub Clock Oscillation Stop Detection Function

Example of re-returning when CKSEL[2:0]=100b (selecting the sub-clock oscillator) after an oscillation stop is detected.



Note: On return from the oscillation-stopped state, the factor responsible for stopping the sub-clock oscillation circuit must be removed on the user system to allow the return of oscillation.

Note 1. If the SOSTDSR.SOSTDF flag is cleared, wait at least 1.5 μs before reading the next SOSTDSR.SOSTDF flag.

Note 2. Retry at least once.

Figure 9.11 Flow of Recovery from Detection of Sub Clock Oscillator Stop

9.9.4 Sub-clock Oscillation Stop Detection Interrupts

A sub-clock oscillation stop detection interrupt (SOSTD) is generated when the Sub-clock Oscillation Stop Detection flag (SOSTDSR.SOSTDF) is 1 and the Sub-clock Oscillation Stop Detection Interrupt Enable bit in the Sub-clock Oscillation Stop Detection Control Register (SOSTDCR.SOSTDIE) is 1 (enabled).

When the SOSTDSR.SOSTDF flag requires clearing, do so after clearing the Sub-clock Oscillation Stop Detection Interrupt Enable bit in the Sub-clock Oscillation Stop Detection Control Register (SOSTDCR.SOSTDIE). Make sure SOSTDF is cleared before setting the SOSTDIE bit to 1 again.

9.10 Internal Clock

Clock sources for the internal clock signals include:

- Main clock oscillator
- Sub-clock oscillator
- HOCO clock
- MOCO clock
- LOCO clock
- PLL1 clock (PLL1P, PLL1Q and PLL1R)
- PLL2 clock (PLL2P, PLL2Q and PLL2R)
- External clock input for JTAG
- External clock input for SWD

The following internal clocks are produced from these sources.

- Operating clock of the CPU0 : CPU0 clock (CPUCLK0)
- Operating clock of the CPU1 : CPU1 clock (CPUCLK1)
- Operating clock of the DMAC, DTC, SRAM, System Bus, I/O Port and ICU: System clock (ICLK)
- Operating clock of the Debug Subsystem : Debug clock (DCLK)
- Operating clock for MRAM code bus : MRAM code bus clock (MRICLK)
- Operating clocks of peripheral modules : Peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD, and PCLKE)
- Operating clock for the MRAM peripheral bus : MRAM peripheral bus clock (MRPCLK)
- Clock for the external bus controller and external pin output: External bus clock (BCLK, EBCLK)
- Clock for the external bus controller and external pin output for the SDRAM : SDRAM clock (SDCLK)
- Clock for the trace function and external pin output : Trace clock (TRCLK)
- Operating clock for the SCI : SCI clock (SCICLK)
- Operating clock for the SPI : SPI clock (SPICLK)
- Operating clock for the Octal SPI : Octal-SPI clock (OCTACLK, OCTADIVCLK)
- Operating clock for the CANFD Core : CANFD Core clock (CANFDCLK)
- Operating clock for the USBFS and USBHS : USB clock (USBCLK)
- Operating clock for the USBHS : USB clock (USB60CLK)
- Operating clock for the ADC16H : ADC clock (ADCCLK)
- Operating clock for the GPT : GPT clock (GPTCLK)
- Operating clock for the I3C : I3C clock (I3CCLK)
- Clock for the external bus controller (Asynchronous) : External bus clock (BCLKA)
- Operating clock for the EtherSW : ESWM clock (ESWCLK)
- Operating clock for the EtherSW-PHY : ESWM clock (ESWPHYCLK)
- Operating clock for the ETHPHYCLK pin output : Ether-PHY clock (ETHPHYCLK)

- Clock for external pin output : Clock/Buzzer output (CLKOUT)
- Operating clock for the USBHS : MOSC clock for USBHS (USBMCLK)
- Operating clock for the CAN : CAN clock (CANMCLK)
- Operating clocks for the ULPT : ULPT LOCO clock (ULPTLCLK) and ULPT sub-clock (ULPTSCLK)
- Operating clock for the AGT : AGT LOCO clock (AGTLCLK) and AGT sub-clock (AGTSCLK)
- Operating clocks for the CAC : CAC clock (CACCLK)
- Operating clocks for the RTC : RTC LOCO clock (RTCLCLK) and RTC Sub-clock (RTCSCLK)
- Operating clock for the PDMIF : PDMIF clock (PDMIFCLK)
- Operating clock for the IWDT : IWDT clock (IWDTCLK)
- Operating clock for the SysTick timer : SysTick timer clock (SYSTICKCLK)
- Operating clock for the JTAG : JTAG clock (JTAGTCK)

For details on the registers used to set the frequencies of the internal clocks, see section 9.10.5. System Clock (ICLK) to section 9.10.34. JTAG Clock (JTAGTCK)

If the value of any of these bits is changed, subsequent operation is at the frequency determined by the new value.

9.10.1 CPU0 Clock (CPUCLK0)

The CPU0 clock (CPUCLK0) is the operating clock for the CPU0. Specify the frequency in the following bits:

- CPUCK0[3:0] bits in SCKDIVCR2
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[8:0], PLLMULNF[1:0], PLIDIV[1:0], and PLODIVP[3:0] bits in PLLCCR and PLLCCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.2 CPU1 Clock (CPUCLK1)

The CPU1 clock, CPUCLK1, is the operating clock for the CPU1. Specify the frequency in the following bits:

- CPUCK1[3:0] bits in SCKDIVCR2
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[8:0], PLLMULNF[1:0], PLIDIV[1:0], and PLODIVP[3:0] bits in PLLCCR and PLLCCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.3 MRAM Code Bus Clock (MRICLK)

The MRAM code bus clock, MRICLK, is the operating clock for the MRAM code bus interface. Specify the frequency in the following bits:

- MRICK[3:0] bits in SCKDIVCR2
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[8:0], PLLMULNF[1:0], PLIDIV[1:0], and PLODIVP[3:0] bits in PLLCCR and PLLCCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.4 MRAM Peripheral Bus Clock (MRPCLK)

The MRAM peripheral bus clock, MRPCLK, is the operating clock for the MRAM peripheral bus interface. Specify the frequency in the following bits:

- MRPCK[3:0] bits in SCKDIVCR
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[8:0], PLLMULNF[1:0], PLIDIV[1:0], and PLODIVP[3:0] bits in PLLCCR and PLLCCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.5 System Clock (ICLK)

The system clock (ICLK) is the operating clock of the DMAC, DTC, SRAM, System Bus, I/O Port, and ICU. Specify the frequency in the following bits:

- ICK[3:0] bits in SCKDIVCR
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[8:0], PLLMULNF[1:0], PLIDIV[1:0] and PLODIVP[3:0] bits in PLLCCR and PLLCCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

When the ICLK clock source is switched, the duration of the ICLK clock cycle becomes longer during the clock source transition period. See Figure 9.12 and Figure 9.13.

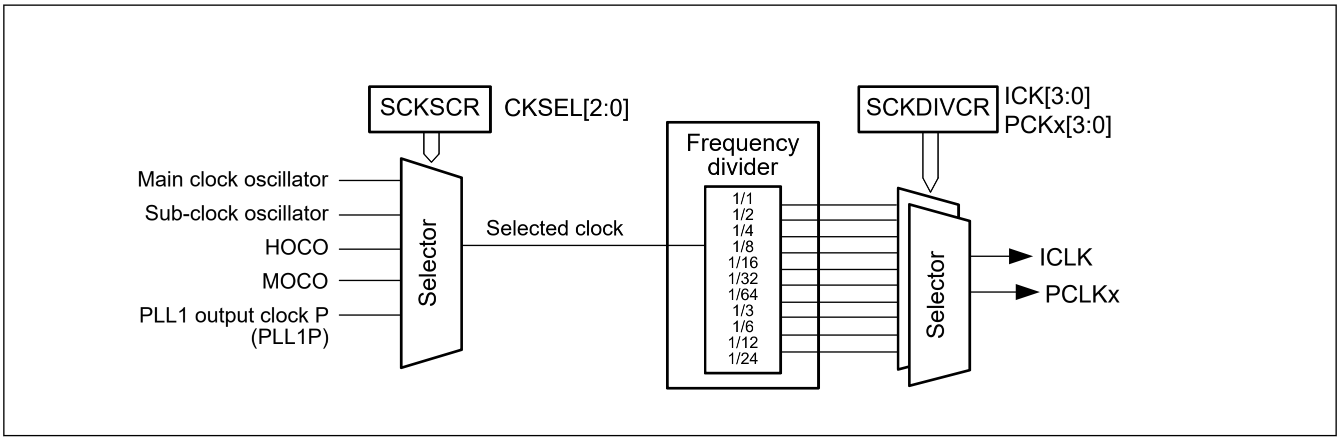


Figure 9.12 Block diagram of clock source selector

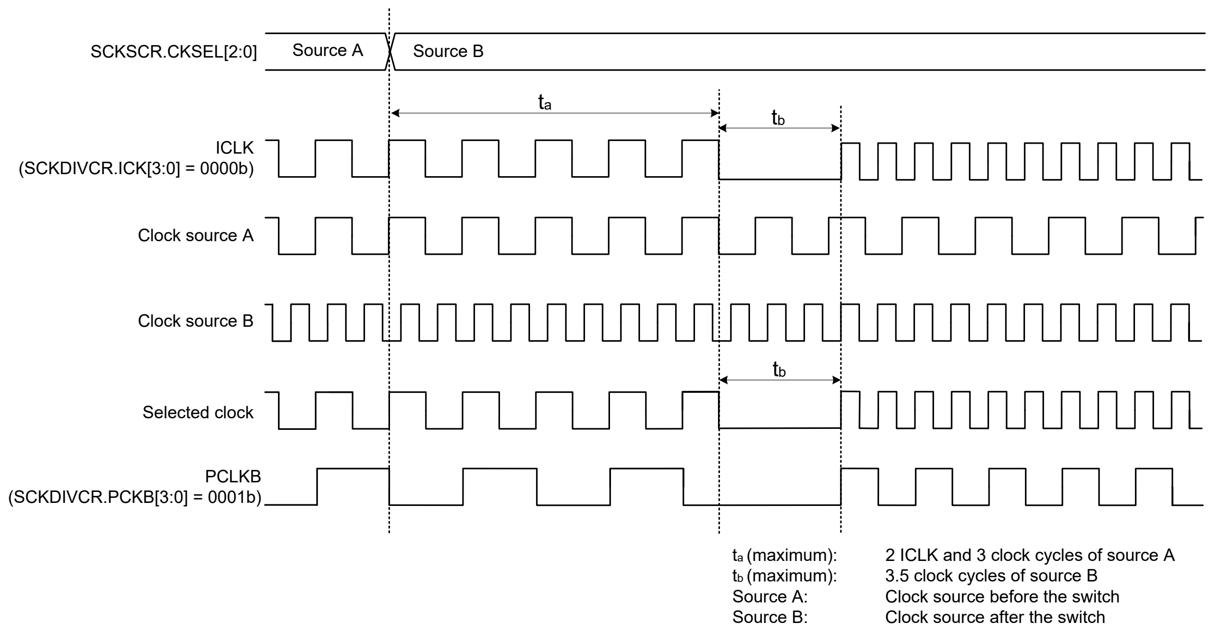


Figure 9.13 Timing of clock source switching

9.10.6 Debugger Clock (DCLK)

The Debugger clock (DCLK) is the operating clock for the Debug Subsystem. The frequency is same as ICLK.

9.10.7 Peripheral Module Clock (PCLKA, PCLKB, PCLKC, PCLKD, PCLKE)

The peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD and PCLKE) are the operating clocks for the peripheral modules.

The frequency of the given clock is specified in the following bits:

- PCKA[3:0], PCKB[3:0], PCKC[3:0], PCKD[3:0] and PCKE[3:0]bits in SCKDIVCR
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[8:0], PLLMULNF[1:0], PLIDIV[1:0] and PLODIVP[3:0] bits in PLLCCR and PLLCCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

When the clock source of the peripheral module clock is switched, the duration of the peripheral module clock cycle becomes longer during the clock source transition period. See Figure 9.12 and Figure 9.13.

9.10.8 External Bus Clock (BCLK, EBCLK)

The external bus clock (BCLK) is an operating clock for the external bus controller. It is also output externally from the EBCLK pin for the external connection bus.

BCLK can be output from the EBCLK pin by setting the EBCKOCR.EBCKOEN bit to 1 and setting the PmnPFS.PSEL[4:0] to 01011b. Make sure that modification of the PmnPFS.PSEL[4:0] to 01011b must always be performed while the EBCKOCR.EBCKOEN bit is 0.

When the BCKCR.BCLKDIV bit is set to 1, the BCLK clock divided by 2 is output from the EBCLK pin.

Specify the frequency in the following bits:

- BCK[3:0] bits in SCKDIVCR

- CKSEL[2:0] bits in SCKSCR
- PLLMUL[8:0], PLLMULNF[1:0], PLIDIV[1:0] and PLODIVP[3:0] bits in PLLCCR and PLLCCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

A frequency higher than the system clock (ICLK) should not be set for the BCLK.

9.10.9 SDRAM Clock (SDCLK)

The SDRAM clock (SDCLK) is an operating clock for the external bus controller. It is output externally from the SDCLK pin for the SDRAM that is connected to the external bus. To output SDCLK on the SDCLK pin, set the SDCKOCR.SDCKOEN bit to 1 and set the PmnPFS.PSEL[4:0] bits to 01011b(enabling SDCLK output). Only change the value in the PmnPFS.PSEL[4:0] bits when the SDCKOCR.SDCKOEN bit is 0. Specify the frequency in the following bits:

- BCK[3:0] bits in SCKDIVCR
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[8:0], PLLMULNF[1:0], PLIDIV[1:0] and PLODIVP[3:0] bits in PLLCCR and PLLCCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

Do not set SDCLK to a frequency higher than that of the system clock (ICLK).

9.10.10 Trace Clock (TRCLK)

The trace clock (TRCLK) is the CPU tracing clock for the On-Chip Debugger function.

The TRCLK frequency is specified by the TRCK[3:0] bits and TRCKSEL bit in TRCKCR (The Trace Clock frequency division ratio and the Trace Clock Source select). When changing the frequency of the Trace Clock, the Trace Clock must be stopped (TRCKCR.TRCKEN = 0).

9.10.11 SCI Clock (SCICLK)

The SCI clock (SCICLK) is the operating clock for the SCI module.

Specify the frequency in the following bits:

- SCICKDIV[3:0] bits in SCICKDIVCR
- SCICKSEL[3:0] bits in SCICKCR
- PLLMUL[8:0], PLLMULNF[1:0], PLIDIV[1:0], PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR and PLLCCR2
- PLL2MUL[8:0], PLL2MULNF[1:0], PL2IDIV[1:0], PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLL2CCR and PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.12 SPI Clock (SPICLK)

The SPI clock (SPICLK) is the operating clock for the SPI module.

Specify the frequency in the following bits:

- SPICKDIV[3:0] bits in SPICKDIVCR
- SPICKSEL[3:0] bits in SPICKCR
- PLLMUL[8:0], PLLMULNF[1:0], PLIDIV[1:0], PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR and PLLCCR2
- PLL2MUL[8:0], PLL2MULNF[1:0], PL2IDIV[1:0], PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLL2CCR and PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.13 Octal-SPI Clock (OCTACLK, OCTADIVCLK)

The Octal-SPI clock (OCTACLK) is the operating clock for the Octal-SPI module.

Specify the frequency in the following bits:

- OCTACKDIV[3:0] bits in OCTACKDIVCR
- OCTACKSEL[3:0] bits in OCTACKCR
- PLLMUL[8:0], PLLMULNF[1:0], PLIDIV[1:0], PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR and PLLCCR2
- PLL2MUL[8:0], PLL2MULNF[1:0], PL2IDIV[1:0], PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLL2CCR and PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

OCTADIVCLK always outputs 2-division of OCTACLK.

9.10.14 CANFD Core Clock (CANFDCLK)

The CANFD Core clock (CANFDCLK) is the operating clock for the CANFD module.

Specify the frequency in the following bits:

- CANFDCKDIV[3:0] bits in CANFDCKDIVCR
- CANFDCKSEL[3:0] bits in CANFDCKCR
- PLLMUL[8:0], PLLMULNF[1:0], PLIDIV[1:0], PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR and PLLCCR2
- PLL2MUL[8:0], PLL2MULNF[1:0], PL2IDIV[1:0], PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLL2CCR and PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.15 USB Clock (USBCLK)

The USB clock (USBCLK) is the operating clock for the USBFS and USBHS module.

A 48-MHz clock must be supplied when using the USBFS module or when using the USBHS module in CL-Only mode. USBCLK does not need to be supplied when not using USBHS module in CL-Only mode.

The USBCLK frequency is specified in the following bits:

- USBCKDIV[3:0] bits in USBCKDIVCR
- USBCKSEL[3:0] bits in USBCKCR
- PLLMUL[8:0], PLLMULNF[1:0], PLIDIV[1:0] and PLODIVP[3:0] bits in PLLCCR and PLLCCR2
- PLL2MUL[8:0], PLL2MULNF[1:0], PL2IDIV[1:0], PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLL2CCR and PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.16 USB Clock (USB60CLK)

The USB clock (USB60CLK) is the operating clock for the USBHS module. A 60-MHz clock must be supplied when using the USBHS module in CL-Only mode. USB60CLK does not need to be supplied when not using USBHS in CL-Only mode.

Specify the frequency in the following bits:

- USB60CKDIV[3:0] bits in USB60CKDIVCR
- USB60CKSEL[3:0] bits in USB60CKCR
- PLLMUL[8:0], PLLMULNF[1:0], PLIDIV[1:0], PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR and PLLCCR2
- PLL2MUL[8:0], PLL2MULNF[1:0], PL2IDIV[1:0], PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLL2CCR and PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.17 MOSC Clock for USBHS (USBMCLK)

The MOSC clock for USBHS (USBMCLK) is the operating clock for the USBHS when not using the module in CL-Only mode. USBMCLK does not need to be supplied when using USBHS in CL-Only mode.

The USBMCLK frequency is 12 MHz or 20 MHz or 24 MHz or 48 MHz supplied from the main clock oscillator.

9.10.18 ADC Clock (ADCCLK)

The ADC clock, ADCCLK, is the operating clock for the ADC16H module. Specify the frequency in the following bits:

- ADCCKDIV[3:0] bits in ADCCKDIVCR
- ADCCKSEL[3:0] bits in ADCCKCR
- PLLMUL[8:0], PLLMULNF[1:0], PLIDIV[1:0], PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR and PLLCCR2
- PLL2MUL[8:0], PLL2MULNF[1:0], PL2IDIV[1:0], PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLLCCR and PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.19 GPT Clock (GPTCLK)

The GPT clock, GPTCLK, is the operating clock for the GPT module. Specify the frequency in the following bits:

- GPTCKDIV[3:0] bits in GPTCKDIVCR
- GPTCKSEL[3:0] bits in GPTCKCR
- PLLMUL[8:0], PLLMULNF[1:0], PLIDIV[1:0], PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR and PLLCCR2
- PLL2MUL[8:0], PLL2MULNF[1:0], PL2IDIV[1:0], PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLLCCR and PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.20 I3C Clock (I3CCLK)

The I3C clock (I3CCLK) is the operating clock for the I3C module.

Specify the frequency in the following bits:

- I3CCKDIV[3:0] bits in I3CCKDIVCR
- I3CCKSEL[3:0] bits in I3CCKCR
- PLLMUL[8:0], PLLMULNF[1:0], PLIDIV[1:0], PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR and PLLCCR2
- PLL2MUL[8:0], PLL2MULNF[1:0], PL2IDIV[1:0], PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLL2CCR and PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.21 Asynchronous External Bus Clock (BCLKA)

The Asynchronous external bus clock, BCLKA, is the operating clock for the external bus controller. Specify the frequency in the following bits:

- BCKACKDIV[3:0] bits in BCKADIVCR
- BCKACKSEL[3:0] bits in BCKACR
- PLLMUL[8:0], PLLMULNF[1:0], PLIDIV[1:0], PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR and PLLCCR2
- PLL2MUL[8:0], PLL2MULNF[1:0], PL2IDIV[1:0], PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLLCCR and PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.22 EtherSW Clock (ESWCLK)

The EtherSW clock, ESWCLK, is the operating clock for the ESWM module. Specify the frequency in the following bits:

- ESWCKDIV[3:0] bits in ESWCKDIVCR
- ESWCKSEL[3:0] bits in ESWCKCR
- PLLMUL[8:0], PLLMULNF[1:0], PLIDIV[1:0], PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR and PLLCCR2
- PLL2MUL[8:0], PLL2MULNF[1:0], PL2IDIV[1:0], PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLLCCR and PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.23 EtherSW-PHY Clock (ESWPHYCLK)

The EtherSW-PHY clock, ESWPHYCLK, is the operating clock for the ESWM module. Specify the frequency in the following bits:

- ESWPCKDIV[3:0] bits in ESWPCKDIVCR
- ESWPCKSEL[3:0] bits in ESWPCKCR
- PLLMUL[8:0], PLLMULNF[1:0], PLIDIV[1:0], PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR and PLLCCR2
- PLL2MUL[8:0], PLL2MULNF[1:0], PL2IDIV[1:0], PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLLCCR and PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.24 Ether-PHY Clock (ETHPHYCLK)

The Ether-PHY clock, ETHPHYCLK, is the operating clock for the ETHPHYCLK pin output. Specify the frequency in the following bits:

- ETHPCKDIV[3:0] bits in ETHPCKDIVCR
- ETHPCKSEL[3:0] bits in ETHPCKCR
- PLLMUL[8:0], PLLMULNF[1:0], PLIDIV[1:0], PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR and PLLCCR2
- PLL2MUL[8:0], PLL2MULNF[1:0], PL2IDIV[1:0], PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLLCCR and PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(_SEC)*1

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.25 CAN Clock (CANMCLK)

The CAN clock (CANMCLK) is the operating clock for the CAN module. CANMCLK is generated by the main clock oscillator.

9.10.26 ULPT Clock (ULPTLCLK, ULPTSCLK)

The ULPT clocks (ULPTLCLK and ULPTSCLK) are the operating clock for the ULPT module. ULPTLCLK is generated by the LOCO clock and ULPTSCLK is generated by the Sub-clock oscillator.

9.10.27 CAC Clock (CACCLK)

The CAC clock (CACCLK) is the operating clock for the CAC. CACCLK includes the following clocks:

- CAC Main clock (CACMCLK) is generated by the Main clock oscillator
- CAC Sub oscillator clock (CACSCLK) is generated by the Sub-clock oscillator
- CAC HOCO clock (CACHCLK) is generated by the HOCO clock
- CAC MOCO clock (CACMOCLK) is generated by the MOCO clock
- CAC LOCO clock (CACLCLK) is generated by the LOCO clock

9.10.28 RTC Clock (RTCSCLK, RTCLCLK)

The RTC clock (RTCSCLK, RTCLCLK) is the operating clock for the RTC.

RTCSCLK is generated by the sub-clock oscillator, and RTCLCLK is generated by the LOCO clock.

9.10.29 IWDT Clock (IWDTCLK)

The IWDT clock (IWDTCLK) is the operating clock for the IWDT.

IWDTCLK is internally generated by the LOCO clock which is always divided by 2.

9.10.30 AGT Clock (AGTSCLK, AGTLCLK)

The AGT clocks (AGTSCLK and AGTLCLK) are the operating clocks for the AGT. AGTSCLK is generated by the sub-clock oscillator, and AGTLCLK is generated by the LOCO clock.

9.10.31 PDMIF Clock (PDMIFCLK)

The PDMIF clock, PDMIFCLK, are the operating clock for the PDMIF.

PDMIFCLK is generated by the MOCO clock.

9.10.32 SysTick Timer Clock (SYSTICKCLK)

The SysTick timer clock (SYSTICKCLK) is the operating clock for the SysTick timer. SYSTICKCLK is generated by the MOCO clock which is always divided by 8.

9.10.33 External Pin Output Clock (CLKOUT)

The CLKOUT is output externally from the CLKOUT pin for the clock or buzzer output. The CLKOUT is output to the CLKOUT pin when the CKOCR.CKOEN bit is set to 1. Only change the value in the CKODIV[2:0] bits or CKOSEL[2:0] bits in CKOCR when the CKOCR.CKOEN bit is 0.

The CLKOUT clock frequency is specified in the following bits:

- CKODIV[2:0] bits or CKOSEL[2:0] bits in CKOCR
- HOCOFRQ0[2:0] bits in OFS1(_SEC)[*1]

Note 1. OFS1 is for non-secure developers and OFS1_SEC is for secure developers. The applied setting value is determined by OFS1_SEL. The value of OFS1(_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

9.10.34 JTAG Clock (JTAGTCK)

The JTAG clock (JTAGTCK) is the clock for the JTAG.

JTAGTCK is generated by the JTAG external clock (TCK).

9.11 Clock Setting

9.11.1 System Clock Setting

9.11.1.1 Initial System Clock Setting

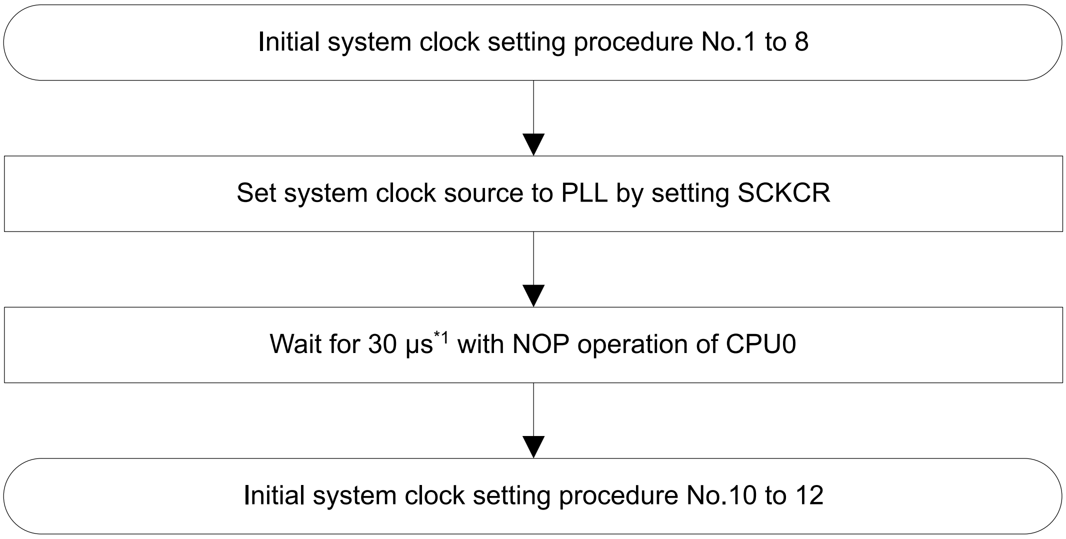
An example of the initial system clock setting procedure is shown in Table 9.7.

Table 9.7 Example of the initial system clock setting procedure after reset release/after Deep Software Standby cancellation

No.	Step	Description
1	Start	After reset release / Deep software standby cancellation (MOCO is selected as system clock source)
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 *1*2bit in PRCR register
3	Change Operating Power Control Mode to High-Speed mode*1	Change to High-Speed mode with OPCCR register. *1 For details, see section 11, Low Power Mode.
4	Set a clock source to oscillate	Set a clock source that switches to the system clock source to oscillate.
5	Set MRAM and SRAM Wait Cycle	Set an appropriate access wait count for the MRAM and SRAM according to the frequency at which it operates after the system clock (ICLK), MRICLK and MRPCLK are set. For details, see section 59, MRAM and section 58, SRAM.
6	Set voltage scaling	Set an appropriate target voltage of VDD according to the frequency of each clocks after switching the system clock. For details, see section 11, Low Power Mode.
7	Set internal clock division ratio	Set the internal clock division ratio with SCKDIVCR and SCKDIVCR2 registers.
8	Set External Bus clock output / SDRAM clock output*3	Set the EBCLK pin output setting with BCKCR. Set the SDCLK pin output setting with SDCKOCR.
9	Switch the System Clock source	Switch the system clock source with SCKCR register. Before switching the system clock source, confirm that the clock source oscillation is stable.
10	Change Operating Power Control Mode	Change the Operating Power Control Mode with OPCCR register. *2
11	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 *1*2bit in PRCR register
12	End	System clock setting is completed.

Note 1. Setting is not necessary if the operating power control mode is already the High-Speed mode.
Note 2. Setting is not necessary if the operating power control mode is not changed.
Note 3. Setting is not necessary if the External bus or SDRAM are not used.

When the PLL1 is selected as the system clock source, additional procedure is required for procedure No.9. This additional procedure is shown in Figure 9.14. It is recommended to use software to measure the wait time. Be sure to consider the worst-case conditions to ensure that the required wait time elapses. If an interrupt is unavoidably generated during the wait time, retry the measurement after return from the interrupt.



Note 1. 30 μs is in DCDC mode, 10 μs in External VDD mode.

Figure 9.14 Additional flow of initial system clock setting when PLL is used for the system clock source (assuming all module stop bits are initial value)

9.11.1.2 System Clock Setting for Faster Frequency

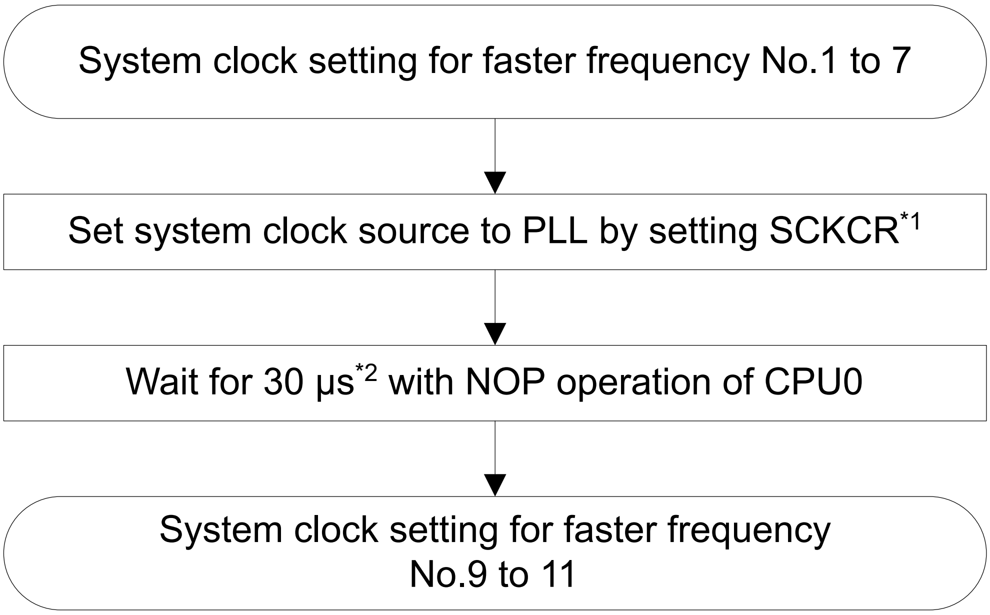
An example of setting procedure when changing the current system clock frequency to a faster frequency is shown in Table 9.8

Table 9.8 Example of setting procedure when changing the current system clock frequency to a faster frequency

No.	Step	Description
1	Start	
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1[*1*2] bit in PRCR register
3	Change Operating Power Control Mode to High-Speed mode[*1]	Change to High-Speed mode with OPCCR register.[*1] For details, see section 11, Low Power Mode.
4	Set a clock source to oscillate[*3]	Set a clock source that switches to the system clock source to oscillate.[*4]
5	Set MRAM and SRAM Wait Cycle	Set an appropriate access wait count for the MRAM and SRAM according to the frequency at which it operates after the system clock (ICLK), MRICLK and MRPCLK are set. For details, see section 59, MRAM and section 58, SRAM.
6	Set voltage scaling	Set an appropriate target voltage of VDD according to the frequency of each clocks after switching the system clock. For details, see section 11, Low Power Mode.
7	Set internal clock division ratio	Set the internal clock division ratio with SCKDIVCR and SCKDIVCR2 registers.
8	Switch the System Clock source[*3]	Switch the system clock source with SCKCR register. Before switching the system clock source, confirm that the clock source oscillation is stable.
9	Change Operating Power Control Mode	Change the Operating Power Control Mode with OPCCR register. [*2]
10	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1[*1*2] bit in PRCR register
11	End	System clock is changed to a Faster frequency.

Note 1. Setting is not necessary if the operating power control mode is already the High-Speed mode.
Note 2. Setting is not necessary if the operating power control mode is not changed.
Note 3. Setting is not necessary if the system clock source is not changed.
Note 4. To change the frequency of the PLL selected as the system clock source, it is necessary to change the system clock source to another clock source, stop the PLL, and then change the PLL setting.

When the PLL1 is selected as the system clock source, additional procedure is required for procedure No.8. This additional procedure is shown in Figure 9.15. It is recommended to use software to measure the wait time. Be sure to consider the worst-case conditions to ensure that the required wait time elapses. If an interrupt is unavoidably generated during the wait time, retry the measurement after return from the interrupt.



Note 1. Do not set SCKCR when the PLL is already selected.
Note 2. 30 μs is in DCDC mode, 10 μs in External VDD mode.

Figure 9.15 Additional flow of system clock setting when PLL is used for the system clock source and CPUCLK0 to be faster frequency

9.11.1.3 System Clock Setting for Slower frequency

An example of setting procedure when changing the current system clock frequency to a slower frequency is shown in Table 9.9.

Table 9.9 Example of setting procedure when changing the current system clock frequency to a slower frequency (1 of 2)

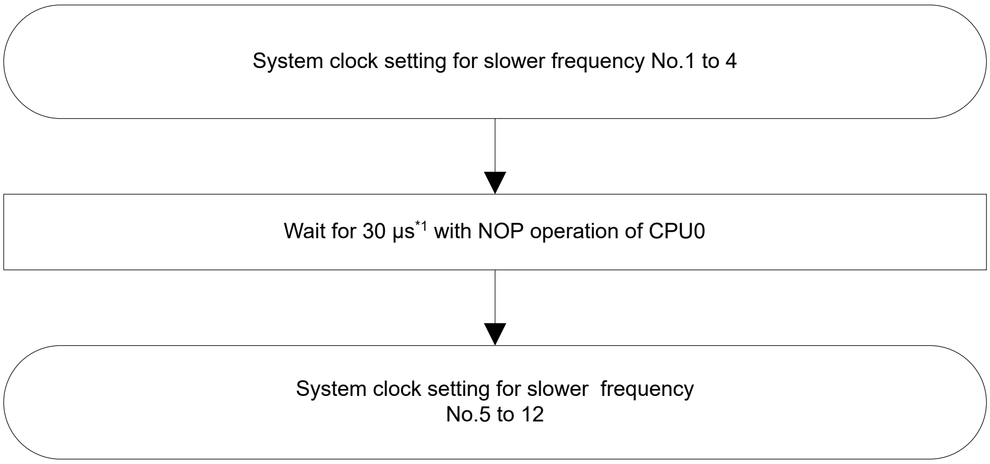
No.	Step	Description
1	Start	
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1*1*2 bit in PRCR register
3	Change Operating Power Control Mode to High-Speed mode*1	Change to High-Speed mode with OPCCR register.*1 For details, see section 11, Low Power Mode.
4	Set a clock source to oscillate*3	Set a clock source that switches to the system clock source to oscillate.*4
5	Set internal clock division ratio	Make the following settings so that the frequency does not exceed the electrical characteristics range when the system clock source is switched in the next step. ● Set the internal clock division ratio larger than the current division ratio. ● Set the internal clock division ratio so that the frequency after switching the system clock source is slower than the current frequency.*3
6	Switch the System Clock source*3	Switch the system clock source with SCKCR register. Before switching the system clock source, confirm that the clock source oscillation is stable.
7	Re-set internal clock division ratio	Re-set the internal clock division ratio with SCKDIVCR and SCKDIVCR2 registers according to the frequency after switching the system clock source.

Table 9.9 Example of setting procedure when changing the current system clock frequency to a slower frequency (2 of 2)

No.	Step	Description
8	Set voltage scaling	Set an appropriate target voltage of VDD according to the frequency of each clocks after switching the system clock. For details, see section 11, Low Power Mode.
9	Set MRAM and SRAM Wait Cycle	Set an appropriate access wait count for the MRAM and SRAM according to the frequency at which it operates after the system clock (ICLK), MRICLK and MRPCLK are set. For details, see section 59, MRAM and section 58, SRAM.
10	Change Operating Power Control Mode	Change the Operating Power Control Mode with OPCCR register. [*2]
11	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 [*1*2] bit in PRCR register
12	End	System clock is changed to a Slower frequency.

Note 1. Setting is not necessary if the operating power control mode is already the High-Speed mode.
Note 2. Setting is not necessary if the operating power control mode is not changed.
Note 3. Setting is not necessary if the system clock source is not changed.
Note 4. To change the frequency of the PLL selected as the system clock source, it is necessary to change the system clock source to another clock source, stop the PLL, and then change the PLL setting.

When the PLL1 is selected as the system clock source before switching the system clock source, additional procedure is required before procedure No.5. This additional procedure is shown in Figure 9.16. It is recommended to use software to measure the wait time. Be sure to consider the worst-case conditions to ensure that the required wait time elapses. If an interrupt is unavoidably generated during the wait time, retry the measurement after return from the interrupt.



Note 1. 30 μs is in DCDC mode, 10 μs in External VDD mode.

Figure 9.16 Additional flow of system clock setting when PLL is used for the system clock source and CPUCLK0 to be slower frequency

9.11.2 Peripheral Module-dedicated Clock Setting

This section describes the setting procedure of the operating clock for the peripheral module. The target clocks are as follows : SCICLK, SPICLK, OCTACLK, CANFDCLK, USBCLK,USB60CLK, ADCCLK, GPTCLK, I3CCLK, BCLKA, ESWCLK, ESWPHYCLK and ETHPHYCLK.

9.11.2.1 Initial Peripheral Module-dedicated Clock Setting

An example of the initial clock setting procedure for peripheral module is shown in Table 9.10.

Table 9.10 Example of the setting procedure of the operating clock for peripheral module after reset release/after Deep Software Standby cancellation

No.	Step	Description
1	Start	After reset release / Deep software standby cancellation (MOCO is selected as system clock source)
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 *1*2bit in PRCR register
3	Change Operating Power Control Mode to High-Speed mode*1	Change to High-Speed mode with OPCCR register. *1 For details, see section 11, Low Power Mode.
4	Set CKSREQ and wait for setting CKSRDY	Write 1 to CKSREQ bit in each CKCR register. Polling until CKSRDY bit in each CKCR register is read as 1 While CKSRDY is 1, the clock supply stops.
5	Set a clock source to oscillate	Set a clock source that switches to the peripheral module-dedicated clock source to oscillate.
6	Set the clock division ratio and switch the clock source	Write the setting value to CKDIV[3:0] in each CKDIVCR register. Write the setting value to CKSEL[3:0] in each CKCR register.
7	Clear CKSREQ and wait for clearing CKSRDY	Write 0 to CKSREQ bit in each CKCR register. Polling until CKSRDY bit in each CKCR register is read as 0. After CKSRDY is 0, the clock supply starts.
8	Cancel the Module Stop Control	Cancel the Module Stop Control in MSTPCRn register (n = A to E) and supply clock to peripheral module.
9	Change Operating Power Control Mode	Change the Operating Power Control Mode with OPCCR register.*2
10	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 bit in PRCR register
11	End	Clock setting is completed. And then, set the peripheral modules.

Note 1. Setting is not necessary if the operating power control mode is already the High-Speed mode.
Note 2. Setting is not necessary if the operating power control mode is not changed.

9.11.2.2 Peripheral Module-dedicated Clock Source Changing

Table 9.11 shows an example of the setting change procedure when switching the clock source of the operating clock for peripheral module in use.

Table 9.11 Example of the setting change procedure when switching the clock source of the operating clock for peripheral module in use (1 of 2)

No.	Step	Description
1	Start	The operating clock for the peripheral module is being used.
2	Stop the peripheral module	Stop the operation of the peripheral module that the operating clock is to be changed.
3	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1*1*2bit in PRCR register
4	Change Operating Power Control Mode to High-Speed mode *1	Change to High-Speed mode with OPCCR register.*1 For details, see section 11, Low Power Mode.
5	Set CKSREQ and wait for setting CKSRDY	Write 1 to CKSREQ bit in each CKCR register. Polling until CKSRDY bit in each CKCR register is read as 1. While CKSRDY is 1, the clock supplying stops.
6	Set a clock source to oscillate	Set a clock source that switches to the peripheral module-dedicated clock source to oscillate.
7	Set the clock division ratio and switch the clock source	Write the setting value to CKDIV[3:0] in each CKDIVCR register. Write the setting value to CKSEL[3:0] in each CKCR register.
8	Clear CKSREQ and wait for clearing CKSRDY	Write 0 to CKSREQ bit in each CKCR register. Polling until CKSRDY bit in each CKCR register is read as 0. After CKSRDY is 0, the clock supplying starts.
9	Change Operating Power Control Mode	Change the Operating Power Control Mode with OPCCR register. *2
10	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 bit in PRCR register

Table 9.11 Example of the setting change procedure when switching the clock source of the operating clock for peripheral module in use (2 of 2)

No.	Step	Description
11	Restart the peripheral module	Change the peripheral module settings according to the operating clock after the settings are changed. After that, restart the operation of the peripheral module.
12	End	Clock setting is completed.

Note 1. Setting is not necessary if the operating power control mode is already the High-Speed mode.
Note 2. Setting is not necessary if the operating power control mode is not changed.

9.11.2.3 Peripheral Module-dedicated Clock Division Ratio Changing

Table 9.12 shows an example of the setting change procedure when switching the division ratio of the operating clock for peripheral module in use.

Table 9.12 Example of the setting change procedure when switching the division ratio of the operating clock for peripheral module in use (without the clock source change)

No.	Step	Description
1	Start	The operating clock for the peripheral module is being used.
2	Stop the peripheral module	Stop the operation of the peripheral module that the operating clock is to be changed.
3	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 *1*2 bit in PRCR register
4	Change Operating Power Control Mode to High-Speed mode *1	Change to High-Speed mode with OPCCR register. *1 For details, see section 11, Low Power Mode.
5	Set the clock division ratio	Write the setting value to CKDIV[3:0] in each CKDIVCR register.
6	Change Operating Power Control Mode	Change the Operating Power Control Mode with OPCCR register.*2
7	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 bit in PRCR register
8	Restart the peripheral module	Change the peripheral module settings according to the operating clock after the settings are changed. After that, restart the operation of the peripheral module.
9	End	Clock setting is completed.

Note 1. Setting is not necessary if the operating power control mode is already the High-Speed mode.
Note 2. Setting is not necessary if the operating power control mode is not changed.

9.11.3 Main Clock Oscillator Setting

Table 9.13 shows an example of the initial setting procedure for the Main Clock Oscillator.

Table 9.13 Example of the initial setting procedure for Main Clock Oscillator after reset release / after Deep Software Standby cancellation (1 of 2)

No.	Step	Description
1	Start	The Main Clock Oscillator is stopped after reset release / Deep software standby cancellation
2	Cancel Register Write Protection	Set 1 to PRC0 bit in PRCR register
3	Set the mode of Main Clock Oscillator	Set the following Main Clock Oscillator mode with MOMCR register ● Resonator or Clock input ● Drive Capability
4	Set the oscillation stabilization wait time	Set the Main Clock Oscillator Wait Time in MOSCWTCR register.
5	Set the oscillation keep in Software Standby mode	Set MOSCSCR register if Main Clock Oscillator keeps oscillation in Software Standby mode.
6	Set the Main Clock Oscillator to operate	Set the Main Clock Oscillator to start oscillating with the MOSCCR register.
7	Wait for Main Clock oscillation to stabilize	Polling until MOSCSF bit in OSCSF register is read as 1 (Oscillation is stable)

Table 9.13 Example of the initial setting procedure for Main Clock Oscillator after reset release / after Deep Software Standby cancellation (2 of 2)

No.	Step	Description
8	Apply Register Write Protection	Set 0 to PRC0 bit in PRCR register
9	End	The Main Clock Oscillator setting is completed. The Main Clock is available.

9.11.4 Sub-Clock Oscillator Setting

Table 9.14 shows an example of the initial setting procedure for the Sub-Clock Oscillator.

Table 9.14 Example of the initial setting procedure for Sub-Clock Oscillator after VBAT Power On Reset release

No.	Step	Description
1	Start	The Sub-Clock Oscillator is stopped after VBAT Power On Reset release.
2	Cancel Register Write Protection	Set 1 to PRC0 bit in PRCR register
3	Setting the mode of Sub-Clock Oscillator	Set the following Sub-Clock Oscillator mode with SOMCR register ● Resonator or Clock input ● Drive Capability
4	Set the Sub-Clock Oscillator to operate	Set the Sub-Clock Oscillator to start oscillating with the SOSCCR register.
5	Wait for Sub-Clock oscillation to stabilize	Wait for sub-clock oscillation stabilization time (tSUBOSCWT). For the sub-clock oscillation stabilization time, see section 62, Electrical Characteristics.
6	Apply Register Write Protection	Set 0 to PRC0 bit in PRCR register
7	End	The Sub-Clock Oscillator setting is completed. The Sub-Clock is available.

9.11.5 High-speed On Chip Oscillator Setting

9.11.5.1 HOCO Initial Clock Setting

Table 9.15 shows an example of the initial setting procedure for the HOCO.

Table 9.15 Example of the HOCO initial setting procedure after reset release / after Deep Software Standby cancellation (OFS1(_SEC).HOCOEN = 1, without FLL)

No.	Step	Description
1	Start	The HOCO is stopped after reset release / Deep software standby cancellation when OFS1(_SEC).HOCOEN is 1.
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 bit in PRCR register
3	Check the HOCO power supply*1	Check the following bit in the HOCOLDOCR register ● LDOSTP bit is 0 (LDO is enabled)
4	Set the HOCO frequency	Set the HOCO frequency with HOCOCR2 register
5	Set the oscillation keep in Software Standby mode	If HOCO keeps oscillation in Software Standby mode, set the following: ● HOCOSOKP bit in HOCOSCR register ● SKEEP bit in HOCOLDOCR register
6	Set HOCO to operate	Set HOCO to start oscillating with the HOCOCR register.
7	Wait for HOCO clock oscillation to stabilize	Polling until HOCOSF bit in OSCSF register is read as 1 (Oscillation is stable)
8	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 bit in PRCR register
9	End	HOCO clock setting is completed.HOCO Clock is available.

Note 1. When HOCOLDOCR.LDOSTP bit is 1, do not operate HOCO. If HOCOLDOCR.LDOSTP is switched from 1 to 0, waiting for the HOCO power supply to stabilize is required. See section 62, Electrical Characteristics on waiting time for the HOCO power supply to stabilize.

9.11.5.2 HOCO Setting with FLL Function

Table 9.16 shows an example of the initial setting procedure for HOCO with FLL function.

Table 9.17 shows the FLL function setting flow for the Software Standby mode transition and cancellation.

Table 9.16 Example of the HOCO setting procedure with FLL function after reset release / after Deep Software Standby cancellation (OFS1(_SEC).HOCOEN = 1)

No.	Step	Description
1	Start	The HOCO is stopped after reset release / Deep software standby cancellation when OFS1(_SEC).HOCOEN is 1.
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 bit in PRCR register
3	Check the HOCO power supply*1	Check the following bit in the HOCOLDOCR register. ● LDOSTP bit is 0 (LDO is enabled)
4	Set the HOCO frequency	Set the HOCO frequency with HOCOCR2 register
5	Set the FLL function to enable*2	Set the FLL Multiplication Control with FLLCR2 register. Set the FLL function to enable with FLLCR1 register.
6	Set HOCO to operate	Set HOCO to start oscillating with the HOCOCR register.
7	Wait for HOCO clock oscillation to stabilize	Polling until HOCOSF bit in OSCSF register is read as 1 (Oscillation is stable)
8	Wait for FLL stabilization	Wait for FLL stabilization wait time (t_{FLLWT}), or wait until the HOCO clock is measured to confirm that the frequency accuracy is stable.
9	Check the HOCO stabilization	Check that HOCOSF bit in OSCSF register is read as 1
10	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 bit in PRCR register
11	End	FLL setting is completed. HOCO Clock is available.

Note 1. When HOCOLDOCR.LDOSTP bit is 1, do not operate HOCO.
If HOCOLDOCR.LDOSTP is switched from 1 to 0, waiting for the HOCO power supply to stabilize is required. See section 62, Electrical Characteristics on waiting time for the HOCO power supply to stabilize.
Note 2. Sub-Clock Oscillator must be oscillating with the stabilization.

Table 9.17 FLL setting flow for Software Standby mode transition and cancellation

No.	Step	Description
1	Start	The HOCO is oscillating with FLL is enabled
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 bit in PRCR register
3	Stop HOCO*1	Set HOCO to stop with HOCOCR register.
4	Disable FLL	Set the FLL function to disable with FLLCR1 register.
5	WFI instruction*2	Transition to Software Standby mode with WFI instruction
6	Software Standby mode*2	MCU is in Software Standby mode
7	Cancellation Software Standby mode*2	Software Standby mode is cancelled
8	Set the FLL function to enable*3	Set the FLL Multiplication Control with FLLCR2 register. Set the FLL function to enable with FLLCR1 register.
9	Set HOCO to operate	Set HOCO to start oscillating with the HOCOCR register.
10	Wait for FLL stabilization	Wait for FLL stabilization wait time (t_{FLLWT}), or wait until the HOCO clock is measured to confirm that the frequency accuracy is stable.
11	Check the HOCO stabilization	Check that HOCOSF bit in OSCSF register is read as 1.
12	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 bit in PRCR register
13	End	HOCO Clock is available.

Note 1. If HOCO is used as the system clock source or the clock source of PLL, these clock source must be changed to another clock before HOCO is stopped.
Note 2. See section 11, Low Power Mode for detail of Software Standby mode.
Note 3. Sub-Clock Oscillator must be oscillating with the stabilization.

9.11.6 PLL Setting

9.11.6.1 PLL Initial Clock Setting

Table 9.18 shows an example of the initial setting procedure for the PLL.

Table 9.18 Example of the PLL initial setting procedure after reset release / after Deep Software Standby cancellation (PLL1 or PLL2)

No.	Step	Description
1	Start	The PLL1 is stopped after reset release / Deep software standby cancellation. The PLL2 is stopped after reset release / Deep software standby cancellation.
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 bit in PRCR register.
3	Change Operating Power Control Mode to High-Speed mode*1	Change to High-Speed mode with OPCCR register. For details, see section 11, Low Power Mode.
4	Check the PLL power supply*2	[PLL1] Check the following bit in the PLL1LDOCR register. ● LDOSTP bit is 0 (LDO is enabled). [PLL2] Check the following bit in the PLL2LDOCR register. ● LDOSTP bit is 0 (LDO is enabled).
5	Configure the PLL settings	[PLL1] Configure the following PLL1 settings in PLLCCR and PLLCCR2 registers. [PLL2] Configure the following PLL2 settings in PLL2CCR and PLL2CCR2 registers. [Setting Items] ● PLL input frequency division ratio ● PLL clock source ● Frequency Multiplication Factor ● Frequency Multiplication Fractional Factor ● PLL Output Frequency Division Ratio (P/Q/R)
6	Set PLL to operate	[PLL1] Set PLL1 to start oscillating with the PLLCR register. [PLL2] Set PLL2 to start oscillating with the PLL2CR register.
7	Wait for PLL clock oscillation to stabilize	[PLL1] Polling until PLLSF bit in OSCSF register is read as 1 (Oscillation is stable). [PLL2] Polling until PLL2SF bit in OSCSF register is read as 1 (Oscillation is stable).
8	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 bit in PRCR register
9	End	PLL1 or PLL2 clock setting is completed. [PLL1] PLL1P, PLL1Q, PLL1R clocks are available. [PLL2] PLL2P, PLL2Q, PLL2R clocks are available.

Note 1. Setting is not necessary if the operating power control mode is already the High-Speed mode.
Note 2. When PLL1LDOCR.LDOSTP bit is 1, do not operate PLL1.
If PLL1LDOCR.LDOSTP is switched from 1 to 0, waiting for the PLL1 power supply to stabilize is required. See section 62, Electrical Characteristics on waiting time for the PLL1 power supply to stabilize.
When PLL2LDOCR.LDOSTP bit is 1, do not operate PLL2.
If PLL2LDOCR.LDOSTP is switched from 1 to 0, waiting for the PLL2 power supply to stabilize is required. See section 62, Electrical Characteristics on waiting time for the PLL2 power supply to stabilize.

9.11.6.2 PLL Clock Setting Change

Table 9.19 shows an example of the initial setting procedure for the PLL.

Table 9.19 Example of the PLL setting change procedure (PLL1 or PLL2)

No.	Step	Description
1	Start	PLL1 is oscillating and the PLL1 clocks (PLL1P, PLL1Q, PLL1R) are being used. PLL2 is oscillating and the PLL2 clocks (PLL2P, PLL2Q, PLL2R) are being used.
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 bit in PRCR register.
3	Stop PLL*1	[PLL1] Set PLL1 to stop with PLLCR register. [PLL2] Set PLL2 to stop with PLL2CR register.
4	Change the PLL settings	[PLL1] Configure the following PLL1 settings in PLLCCR and PLLCCR2 registers. [PLL2] Configure the following PLL2 settings in PLL2CCR and PLL2CCR2 registers. [Setting Items] ● PLL input frequency division ratio ● PLL clock source ● Frequency Multiplication Factor ● Frequency Multiplication Fractional Factor ● PLL Output Frequency Division Ratio (P/Q/R)
5	Set PLL to operate	[PLL1] Set PLL1 to start oscillating with the PLLCR register. [PLL2] Set PLL2 to start oscillating with the PLL2CR register.
6	Wait for PLL clock oscillation to stabilize	[PLL1] Polling until PLLSF bit in OSCSF register is read as 1 (Oscillation is stable). [PLL2] Polling until PLL2SF bit in OSCSF register is read as 1 (Oscillation is stable).
7	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 bit in PRCR register.
8	End	PLL1 or PLL2 clock setting change is completed. [PLL1] PLL1P, PLL1Q, PLL1R clocks are available. [PLL2] PLL2P, PLL2Q, PLL2R clocks are available.

Note 1. If PLL1P clock is used as the system clock source, the system clock source must be changed to another clock source before PLL1 is stopped.

9.12 Usage Notes

9.12.1 Notes on Clock Generation Circuit

The frequency of the following clocks supplied to each module changes according to the setting of the SCKDIVCR and SCKDIVCR2 register:

- CPU0 clock (CPUCLK0)
- CPU1 clock (CPUCLK1)
- MRAM code bus clock (MRICLK)
- System clock (ICLK)
- Peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD and PCLKE)
- MRAM peripheral bus clock (MRPCLK)
- external bus clock (BCLK)

Each frequency must meet the following conditions:

- Each frequency must be selected within the operation-guaranteed range of the operating frequency (f) specified in the AC characteristics. See section 62, Electrical Characteristics.

- Each clock must be set according to Table 9.2.
- Do not change the clock frequency during external bus access. Furthermore, when access via the external bus is to start after a change to the clock frequency, only start access via the bus after confirming that the change to the frequencies has been completed.

To ensure correct processing after the clock frequency changes, first write to the relevant Clock Control register to change the frequency, then read the value from the register, and finally perform the subsequent processing.

9.12.2 Notes on Resonator

Because various resonator characteristics relate closely to your board design, adequate evaluation is required before use. See the resonator connection example in Figure 9.5. The circuit constants for the resonator depend on the resonator to be used and the stray capacitance of the mounting circuit. Therefore, consult the resonator manufacturer when determining the circuit constants. The voltage to be applied between the resonator pins must be within the absolute maximum rating.

9.12.3 Notes on Board Design

When using a crystal resonator, place the resonator and its load capacitors as close to the XTAL and EXTAL pins as possible. Other signal lines should be routed away from the oscillation circuit as shown in Figure 9.17 to prevent electromagnetic induction from interfering with correct oscillation. Figure 9.17 shows the case which the main clock oscillator is used. In case of sub-clock oscillator, it is also same as Figure 9.17.

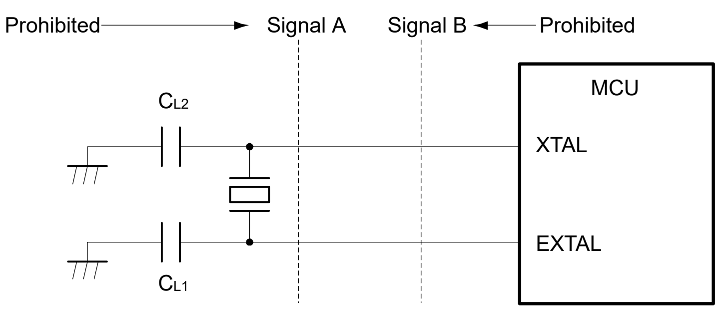


Figure 9.17 Signal routing in board design for oscillation circuit

9.12.4 Notes on Resonator Connect Pin

When the main clock is not used, the EXTAL and XTAL pins can be used as general ports. When these pins are used as general ports, the main clock must be stopped (MOSCCR.MOSTP bit should be set to 1).

When the sub-clock is not used, the XCIN and XCOUT pins can be used as general ports. When these pins are used as the general ports, the sub-clock must be stopped (SOSCCR.SOSTP bit should be set to 1).

9.12.5 Notes on Using Sub-Clock Oscillator

The output of the P212 (EXTAL), P213 (XTAL), P408 (for BGA-289pin) and P415 (for BGA-224pin) pins may affect the oscillation by the sub-clock oscillator.

If the sub-clock oscillator is used, implement board design so as not to affect the oscillation. Renesas strongly recommends setting the PmnPFS.DSCR[1:0] bits to 00b or 01b when using the P212 (EXTAL), P213 (XTAL), P408 (for BGA-289pin) and P415 (for BGA-224pin) as output pins and using the sub-clock oscillator.

In addition, when using the sub-clock oscillator in Low power mode drive capability (SOMCR.SODRV[1:0]= 01b, 10b and 11b), Renesas recommends setting the PmnPFS.DSCR[1:0] bits to 00b when using the P212 (EXTAL), P213 (XTAL), P408 (for BGA-289pin) and P415 (for BGA-224pin) as output pins and using the sub-clock oscillator.

9.12.6 Notes on Reset Conditions of SOSC and LOCO

RTC is not initialized by any reset source.

SOSC and LOCO can be selected as the clock source of RTC. The followings are the states of SOSC and LOCO when reset occurs.

Table 9.20 Reset Behavior of SOSC

Clock source	Item	Reset source	
		VBATT_POR reset	Other
SOSC	Enable or disable Drive capability	Initialized to enable	continue the state before the reset occurs

Table 9.21 Reset Behavior of LOCO

Clock source	Item	Reset source	
		Power-on reset/ Voltage-Monitoring 0, 4, 5 reset/ Core Voltage Monitor reset/ Deep software standby 2, 3 reset	Other
LOCO	Enable or disable	Initialized to enable	
	Oscillation accuracy*1	Initialized to accuracy before trimming by LOCOUTCR (accuracy : ±15 %)	Continue the accuracy that was trimmed by LOCOUTCR

Note 1. If the LOCO that is trimmed by LOCOUTCR is selected as the RTC source clock, please be careful that LOCO oscillation accuracy will be initialized by Power-on reset, Voltage-Monitor 0, 4, 5 reset, Core Voltage-Monitor reset and Deep software standby 2, 3 reset.

When a reset is canceled, a reset exception handling starts.

For details on the reset exception handling, see section 6, Resets

10. Clock Frequency Accuracy Measurement Circuit (CAC)

10.1 Overview

The Clock Frequency Accuracy Measurement Circuit (CAC) counts pulses of the clock to be measured (measurement target clock) within the time generated by the clock selected as the measurement reference (measurement reference clock), and determines the accuracy depending on whether the number of pulses is within the allowable range. When measurement is complete or the number of pulses within the time generated by the measurement reference clock is not within the allowable range, an interrupt request is generated.

Table 10.1 lists the CAC specifications, Figure 10.1 shows the CAC block diagram, and Table 10.2 lists the CAC I/O pin.

Table 10.1 CAC specifications

Parameter	Specifications
Measurement target clocks	Frequency can be measured for: ● Main clock oscillator (CACMCLK) ● Sub-clock oscillator (CACSCLK) ● HOCO clock (CACHCLK) ● MOCO clock (CACMOCLK) ● LOCO clock (CACLCLK) ● Peripheral module clock B (PCLKB)
Measurement reference clocks	Frequency can be referenced to: ● External clock input to the CACREF pin ● Main clock oscillator (CACMCLK) ● Sub-clock oscillator (CACSCLK) ● HOCO clock (CACHCLK) ● MOCO clock (CACMOCLK) ● LOCO clock (CACLCLK) ● Peripheral module clock B (PCLKB)
Selectable function	Digital filter
Interrupt sources	● Measurement end ● Frequency error ● Overflow
Module-stop function	Module-stop state can be set to reduce power consumption
TrustZone Filter	Security and Privilege attribution can be set

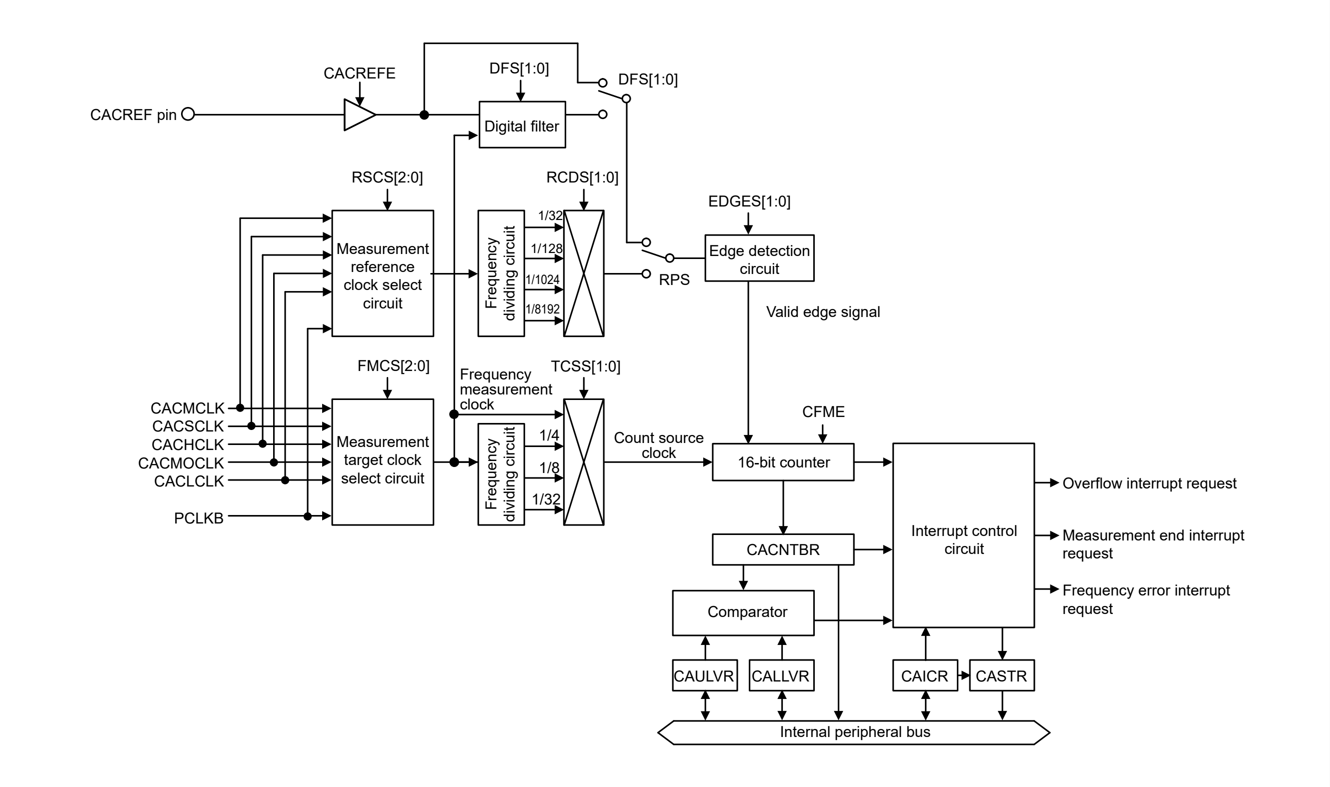


Figure 10.1 CAC block diagram

Table 10.2 CAC I/O pin

Function	Pin name	I/O	Description
CAC	CACREF	Input	Measurement reference clock input pin

10.2 Register Descriptions

10.2.1 CACR0 : CAC Control Register 0

Base address: CAC = 0x4020_2400
CAC_NS = 0x5020_2400

Offset address: 0x00

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

Bit	Symbol	Function	R/W
0	CFME	Clock Frequency Measurement Enable 0: Disable 1: Enable	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

CFME bit (Clock Frequency Measurement Enable)

The CFME bit enables clock frequency measurement. When this bit is rewritten, it takes some time for the change to be reflected in the internal circuit. If this bit is rewritten when the previous value has not been reflected in the internal circuit, the write is ignored. Read the bit to check if the rewrite has been reflected.

10.2.2 CACR1 : CAC Control Register 1

Base address: CAC = 0x4020_2400
CAC_NS = 0x5020_2400

Offset address: 0x01

Bit position:	7	6	5	4	3	2	1	0
Bit field:	EDGES[1:0]		TCSS[1:0]		FMCS[2:0]			CACREFE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CACREFE	CACREF Pin Input Enable 0: Disable 1: Enable	R/W
3:1	FMCS[2:0]	Measurement Target Clock Select 0 0 0: Main clock oscillator (CACMCLK) 0 0 1: Sub-clock oscillator (CACSCLK) 0 1 0: HOCO clock (CACHCLK) 0 1 1: MOCO clock (CACMOCLK) 1 0 0: LOCO clock (CACLCLK) 1 0 1: Peripheral module clock B (PCLKB) 1 1 0: Setting prohibited 1 1 1: Setting prohibited	R/W
5:4	TCSS[1:0]	Timer Count Clock Source Select 0 0: No division 0 1: × 1/4 clock 1 0: × 1/8 clock 1 1: × 1/32 clock	R/W
7:6	EDGES[1:0]	Valid Edge Select 0 0: Rising edge 0 1: Falling edge 1 0: Both rising and falling edges 1 1: Setting prohibited	R/W

Note: S-TYPE3, P-TYPE3
Note: Set the CACR1 register when the CACR0.CFME bit is 0.

CACREFE bit (CACREF Pin Input Enable)

The CACREFE bit enables the CACREF pin input.

FMCS[2:0] bits (Measurement Target Clock Select)

The FMCS[2:0] bits select the measurement target clock whose frequency is to be measured.

TCSS[1:0] bits (Timer Count Clock Source Select)

The TCSS[1:0] bits select the division ratio of the measurement target clock.

EDGES[1:0] bits (Valid Edge Select)

The EDGES[1:0] bits select the valid edge for the reference signal.

10.2.3 CACR2 : CAC Control Register 2

Base address: CAC = 0x4020_2400
CAC_NS = 0x5020_2400

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DFS[1:0]		RCDS[1:0]		RSCS[2:0]			RPS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RPS	Reference Signal Select 0: CACREF pin input 1: Internal clock (internally generated signal)	R/W
3:1	RSCS[2:0]	Measurement Reference Clock Select 0 0 0: Main clock oscillator (CACMCLK) 0 0 1: Sub-clock oscillator (CACSCLK) 0 1 0: HOCO clock (CACHCLK) 0 1 1: MOCO clock (CACMOCLK) 1 0 0: LOCO clock (CACLCLK) 1 0 1: Peripheral module clock B (PCLKB) 1 1 0: Setting prohibited 1 1 1: Setting prohibited	R/W
5:4	RCDS[1:0]	Measurement Reference Clock Frequency Division Ratio Select 0 0: × 1/32 clock 0 1: × 1/128 clock 1 0: × 1/1024 clock 1 1: × 1/8192 clock	R/W
7:6	DFS[1:0]	Digital Filter Select 0 0: Disable digital filtering 0 1: Use sampling clock for the digital filter as the frequency measuring clock 1 0: Use sampling clock for the digital filter as the frequency measuring clock divided by 4 1 1: Use sampling clock for the digital filter as the frequency measuring clock divided by 16.	R/W

Note: S-TYPE3, P-TYPE3
Note: Set the CACR2 register when the CACR0.CFME bit is 0.

RPS bit (Reference Signal Select)

The RPS bit selects whether to use the CACREF pin input or an internal clock (internally generated signal) as the reference signal.

RSCS[2:0] bits (Measurement Reference Clock Select)

The RSCS[2:0] bits select the reference clock for measurement.

RCDS[1:0] bits (Measurement Reference Clock Frequency Division Ratio Select)

The RCDS[1:0] bits select the frequency-divisor of the reference clock for measurement when an internal reference clock is selected. When RPS = 0 (CACREF pin is used as the reference clock source), the reference clock is not divided.

DFS[1:0] bits (Digital Filter Select)

The DFS[1:0] bits enable or disable the digital filter and selects its sampling clock.

10.2.4 CAICR : CAC Interrupt Control Register

Base address: CAC = 0x4020_2400
CAC_NS = 0x5020_2400

Offset address: 0x03

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	OVFFCL	MENDFCL	FERRFCL	—	OVFIE	MENDIE	FERRIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FERRIE	Frequency Error Interrupt Request Enable 0: Disable 1: Enable	R/W
1	MENDIE	Measurement End Interrupt Request Enable 0: Disable 1: Enable	R/W
2	OVFIE	Overflow Interrupt Request Enable 0: Disable 1: Enable	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	FERRFCL	FERRF Clear 0: No effect 1: The CASTR.FERRF flag is cleared	W
5	MENDFCL	MENDF Clear 0: No effect 1: The CASTR.MENDF flag is cleared	W
6	OVFFCL	OVFF Clear 0: No effect 1: The CASTR.OVFF flag is cleared.	W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

FERRIE bit (Frequency Error Interrupt Request Enable)

The FERRIE bit enables or disables the frequency error interrupt request.

MENDIE bit (Measurement End Interrupt Request Enable)

The MENDIE bit enables or disables the measurement end interrupt request.

OVFIE bit (Overflow Interrupt Request Enable)

The OVFIE bit enables or disables the overflow interrupt request.

FERRFCL bit (FERRF Clear)

Setting the FERRFCL bit to 1 clears the CASTR.FERRF flag.

MENDFCL bit (MENDF Clear)

Setting the MENDFCL bit to 1 clears the CASTR.MENDF flag.

OVFFCL bit (OVFF Clear)

Setting the OVFFCL bit to 1 clears the CASTR.OVFF flag.

10.2.5 CASTR : CAC Status Register

Base address: CAC = 0x4020_2400
CAC_NS = 0x5020_2400

Offset address: 0x04

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

Bit	Symbol	Function	R/W
0	FERRF	Frequency Error Flag 0: Clock frequency is within the allowable range 1: Clock frequency has deviated beyond the allowable range (frequency error).	R
1	MENDF	Measurement End Flag 0: Measurement is in progress 1: Measurement ended	R
2	OVFF	Overflow Flag 0: Counter has not overflowed 1: Counter overflowed	R
7:3	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3

FERRF flag (Frequency Error Flag)

The FERRF flag indicates a deviation of the clock frequency from the set value (frequency error).

[Setting condition]

- The clock frequency is outside the allowable range defined in the CAULVR and CALLVR registers.

[Clearing condition]

- 1 is written to the FERRFCL bit.

MENDF flag (Measurement End Flag)

The MENDF flag indicates the end of measurement.

[Setting condition]

- Measurement ends.

[Clearing condition]

- 1 is written to the MENDFCL bit.

OVFF flag (Overflow Flag)

The OVFF flag indicates that the counter overflowed.

[Setting condition]

- The counter overflows.

[Clearing condition]

- 1 is written to the CAICR.OVFFCL bit.

10.2.6 CAULVR : CAC Upper-Limit Value Setting Register

Base address: CAC = 0x4020_2400
CAC_NS = 0x5020_2400

Offset address: 0x06

Bit position:	15															0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	n/a	The Upper Value of the Allowable Range The CAULVR register is a 16-bit read/write register that specifies the upper value of the allowable range. When the counter value exceeds the value specified in this register, a frequency error is detected. Write to this register when the CACR0.CFME bit is 0. The counter value stored in CACNTBR can vary depending on the difference between the phases of the digital filter and edge-detection circuit, and the signal on the CACREF pin. Ensure that this setting allows an adequate margin.	R/W

Note: S-TYPE3, P-TYPE3

10.2.7 CALLVR : CAC Lower-Limit Value Setting Register

Base address: CAC = 0x4020_2400
CAC_NS = 0x5020_2400

Offset address: 0x08

Bit position:	15															0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	n/a	The Lower Value of the Allowable Range The CALLVR register is a 16-bit read/write register that specifies the lower value of the allowable range. When the counter value falls below the value specified in this register, a frequency error is detected. Write to this register when the CACR0.CFME bit is 0. The counter value stored in CACNTBR can vary depending on the difference between the phases of the digital filter and edge-detection circuit, and the signal on the CACREF pin. Ensure that this setting allows an adequate margin.	R/W

Note: S-TYPE3, P-TYPE3

10.2.8 CACNTBR : CAC Counter Buffer Register

Base address: CAC = 0x4020_2400
CAC_NS = 0x5020_2400

Offset address: 0x0A

Bit position:	15															0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	n/a	The Measurement Result The CACNTBR register is a 16-bit read-only register that stores the measurement result.	R

Note: S-TYPE3, P-TYPE3

10.3 Operation

10.3.1 Measuring Clock Frequency

The CAC measures the clock frequency using the CACREF pin input or an internal clock as a reference. Figure 10.2 shows an operating example of the CAC.

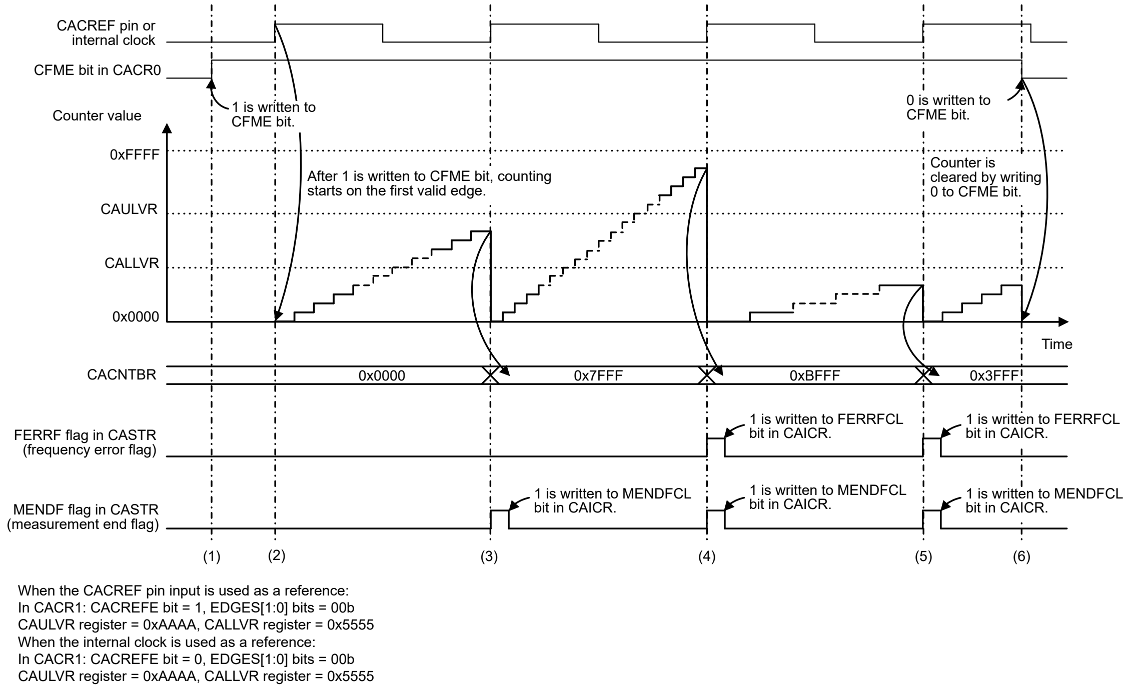


Figure 10.2 CAC operating example

The events in Figure 10.2 are:

1. When the CACREF pin input is used as reference (CACR1.CACREFE = 1), frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is set to 0 and the CACR1.CACREFE bit is set to 1. When the internal clock is used as reference (CACR1.CACREFE = 0), frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is set to 1.
2. When the CACREF pin input is used as reference, after 1 is written to the CFME bit, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits (rising edge (CACR1.EDGES[1:0] = 00b) in Figure 10.2) is input from the CACREF pin. When the internal clock is used as reference, after 1 is written to the CFME bit, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits (rising edge (CACR1.EDGES[1:0] = 00b) in Figure 10.2) is input based on the clock source selected by the CACR2.RSCS[2:0] bits.
3. When the next valid edge is input, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR. If both CACNTBR ≤ CAULVR and CACNTBR ≥ CALLVR are true, only the MENDF flag in CASTR is set to 1, because the clock frequency is correct. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.
4. When the next valid edge is input, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR. If CACNTBR > CAULVR, the FERRF flag in CASTR is set to 1, because the clock frequency is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. The MENDF flag in CASTR is set to 1 at the end of measurement. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.
5. When the next valid edge is input, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR. If CACNTBR < CALLVR, the FERRF flag in CASTR is set to 1, because the clock frequency

is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. The MENDF flag in CASTR is set to 1 at the end of measurement. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.

6. When the CFME bit in CACR0 is 1, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR every time a valid edge is input. Writing 0 to the CFME bit in CACR0 clears the counter and stops up-counting.

10.3.2 Digital Filtering of Signals on CACREF Pin

The CACREF pin has a digital filter, and levels on the CACREF pin are transmitted to the internal circuitry after three consecutive matches in the selected sampling interval. The same level continues to be transmitted internally until the level on the pin has three consecutive matches again. Enabling or disabling of the digital filter and its sampling clock are selectable.

The counter value transferred to CACNTBR might be in error by up to 1 cycle of the sampling clock because of the difference between the phases of the digital filter and the signal input to the CACREF pin. When a frequency dividing clock is selected as a count source clock, the counter value error is obtained using the following formula:

$$\text{Counter value error} = (1 \text{ cycle of the count source clock}) / (1 \text{ cycle of the sampling clock})$$

10.4 Interrupt Requests

The CAC generates three types of interrupt requests:

- Frequency error interrupt
- Measurement end interrupt
- Overflow interrupt

When an interrupt source is generated, the associated status flag is set to 1. Table 10.3 provides information on the CAC interrupt requests.

Table 10.3 CAC interrupt requests

Interrupt request	Interrupt enable bit	Status flag	Interrupt sources
Frequency error interrupt	CAICR.FERRIE	CASTR.FERRF	The result of comparing CACNTBR with CAULVR and CALLVR is either CACNTBR > CAULVR or CACNTBR < CALLVR
Measurement end interrupt	CAICR.MENDIE	CASTR.MENDF	● Valid edge is input from the CACREF pin or internal clock ● Measurement end interrupt does not occur at the first valid edge after writing 1 to the CACR0.CFME bit
Overflow interrupt	CAICR.OVFIE	CASTR.OVFF	Counter overflows

10.5 Usage Notes

10.5.1 Settings for the Module-Stop Function

The Module Stop Control Register C (MSTPCRC) can enable or disable CAC operation. The CAC module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Mode.

11. Low Power Mode

11.1 Overview

The MCU has several functions for reducing power consumption, such as setting clock dividers, EBCLK output control, SDCLK output control, stopping modules, power gating control, selecting operating power control modes, and transitions to low power modes.

Table 11.1 lists the specifications of functions to reduce power consumption. Table 11.2 to Table 11.4 lists the conditions to shift to low power modes, states of the CPU and peripheral modules, and the method for cancelling each mode. After a reset, the MCU enters the program execution state, but modules, except for the DMAC, DTC, and SRAM, do not operate.

Table 11.1 Specifications of the low-power mode functions

Parameter	Specification
Reducing power consumption by switching clock signals	The frequency division ratio can be selected independently for the system clock, peripheral module clock, external bus clock, and MRAM interface clock.*1
EBCLK output control	BCLK output or high-level output can be selected.
SDCLK output control	SDCLK output or high-level output can be selected.
Module-stop	Functions can be stopped independently for each peripheral module.
Power gating control	This function can be controlled the power state of the power domain. ● Control the turning On/OFF for the power domain ● Control the retention of specific circuits during power gating
Processor low power modes	● CPU Sleep mode ● CPU Deep Sleep mode
Low power modes	● Software Standby mode*2 ● Deep Software Standby mode1, 2, 3*2

Note 1. For details, see section 9, Clock Generation Circuit.
Note 2. This mode is not supported in external VDD mode.

Table 11.2 Operating state of processor low power mode (1 of 2)

Item	CPUn Sleep mode		CPUn Deep Sleep mode	
	n = 0 (CPU0)	n = 1 (CPU1)	n = 0 (CPU0)	n = 1 (CPU1)
Transition condition	When [Condition 1] or [Condition 2] after set CPU0.SCR.SLEEPDEEP = 0 [Condition 1] ● WFI or WFE instruction ● A valid interrupt request cannot be accepted to CPU*5 (including a transition from the time WFI instruction is executed to the time the transition to Sleep mode is completed) ● No event from CPU1 [Condition 2] ● CPU0.SCR.SLEEPONEXIT = 1 ● Complete execution of all exception handlers ● A valid interrupt request cannot be accepted to CPU*5 (including a transition from the time WFI instruction is executed to the time the transition to Sleep mode is completed) ● No event from CPU1	When [Condition 1] or [Condition 2] after set CPU1.SCR.SLEEPDEEP = 0 [Condition 1] ● WFI or WFE instruction ● A valid interrupt request cannot be accepted to CPU*5 (including a transition from the time WFI instruction is executed to the time the transition to Sleep mode is completed) ● No event from CPU0 [Condition 2] ● CPU1.SCR.SLEEPONEXIT = 1 ● Complete execution of all exception handlers ● A valid interrupt request cannot be accepted to CPU*5 (including a transition from the time WFI instruction is executed to the time the transition to Sleep mode is completed) ● No event from CPU0	When [Condition 1] or [Condition 2] after set CPU0.SCR.SLEEPDEEP = 1 [Condition 1] ● WFI instruction ● A valid interrupt request cannot be accepted to CPU*5 (including a transition from the time WFI instruction is executed to the time the transition to Deep Sleep mode is completed) [Condition 2] ● CPU0.SCR.SLEEPONEXIT = 1 ● Complete execution of all exception handlers ● A valid interrupt request cannot be accepted to CPU*5 (including a transition from the time WFI instruction is executed to the time the transition to Deep Sleep mode is completed)	When [Condition 1] or [Condition 2] after set CPU1.SCR.SLEEPDEEP = 1 [Condition 1] ● WFI instruction ● A valid interrupt request cannot be accepted to CPU*5 (including a transition from the time WFI instruction is executed to the time the transition to Deep Sleep mode is completed) [Condition 2] ● CPU1.SCR.SLEEPONEXIT = 1 ● Complete execution of all exception handlers ● A valid interrupt request cannot be accepted to CPU*5 (including a transition from the time WFI instruction is executed to the time the transition to Deep Sleep mode is completed)
Canceling method	All interrupts. Any reset available in the mode.	All interrupts. Any reset available in the mode.	Interrupts shown in Table 14.5 Any reset available in the mode.	Interrupts shown in Table 14.5 Any reset available in the mode.
State after cancellation by an interrupt	Program execution state	Program execution state	Program execution state	Program execution state
State after cancellation by a reset	Reset state	Reset state	Reset state	Reset state
CPU0	Stop (Retained)	—	Stop (Retained)	—
CPU0 TCM (SRAM)	—	—	Stop (Retained)*4	—
DMA Controller 0 (DMAC0)	Selectable	—	Selectable	—
Data Transfer Controller 0 (DTC0)	Selectable	—	Selectable	—
Watchdog Timer 0 (WDT0)	Selectable*1	—	Selectable*1	—
CPU1	—	Stop (Retained)	—	Stop (Retained)
DMA Controller 1 (DMAC1)	—	Selectable	—	Selectable
Data Transfer Controller 1 (DTC1)	—	Selectable	—	Selectable
Watchdog Timer 1 (WDT1)	—	Selectable*1	—	Selectable*1

Table 11.2 Operating state of processor low power mode (2 of 2)

Item	CPUn Sleep mode		CPUn Deep Sleep mode	
	n = 0 (CPU0)	n = 1 (CPU1)	n = 0 (CPU0)	n = 1 (CPU1)
Independent Watchdog Timer (IWDT)	Selectable*1	Selectable*1	Selectable*1	Selectable*1
ARM Debug function	Stop*2	Stop*2	Stop*2	Stop*2
Trace function	Stop*3	Stop*3	Stop*3	Stop*3
Other peripheral modules	—	—	—	—

Note: Selectable means that operating or not operating can be selected by setting the control registers before entering Processor Low Power Mode.
Stop (Retained) means that the contents of the internal registers are retained but the operations are suspended.
Stop (Undefined) means that the contents of the internal registers are undefined and power to the internal circuit is cut off.
"—" means that the operation is not affected.

Note 1. In IWDT, operating or stopping is selected by setting the IWDT Stop Control bit (IWDTSTPCTL) in Option Function Select register 0 (OFS0) in IWDT auto start mode. Also, operating or stopping is selected by setting the IWDT.IWDTCSTPR.SLCSTP bit in IWDT register start mode.
In WDT, operating or stopping is selected by setting the WDT Stop Control bit (WDTSTPCTL) in Option Function Select Register 0 (OFS0) in WDT auto start mode. Also, operating or stopping is selected by setting the WDT.WDTCSTPR.SLCSTP bit in WDT register start mode.

Note 2. This function is operating if debugger is connecting (DBGSTR.CDBGPWRUPREQ = 1 and SYOCDCR.DBGEN = 1) before entering this mode.

Note 3. This function is operating if debugger is connecting (DBGSTR.CDBGPWRUPREQ = 1 and SYOCDCR.DBGEN = 1) and TRCKCR.TRCKEN = 1 before entering this mode.

Note 4. If PDRAMSCR1.RKEEP0 bit is set to 0, the contents of the CPU0 TCM is not retained.

Note 5. Valid interrupt requests are any interrupt or exception that are not masked by the priority level of current exception and the priority level set by BASEPRI. In addition, if the interrupt request is based on IELSRn, the interrupt must be enabled by NVIC_ISERn.

Table 11.3 Operating state of each low power mode (1 of 3)

Item	Software Standby mode (SSTBY)	Deep Software Standby mode (DSTBY)		
	SSTBY	DSTBY 1	DSTBY 2	DSTBY 3
Transition condition	When [Condition 1] or [Condition 2] after set LPSCR and CPUn.SCR.SLEEPDEEP = 1 (n = 0, 1) [Condition 1] ● WFI instruction ● A valid interrupt request cannot be accepted to CPU*24 (including a transition from the time WFI instruction is executed to the time the transition to Software Standby mode is completed) [Condition 2] ● CPUn.SCR.SLEEPONEXIT = 1 ● Complete execution of all exception handlers ● A valid interrupt request cannot be accepted to CPU*24 (including a transition from the time WFI instruction is executed to the time the transition to Software Standby mode is completed)	When [Condition 1] or [Condition 2] after set LPSCR and CPUn.SCR.SLEEPDEEP = 1 (n = 0, 1) [Condition 1] ● WFI instruction ● A valid interrupt request cannot be accepted to CPU*24 (including a transition from the time WFI instruction is executed to the time the transition to Software Standby mode is completed) [Condition 2] ● CPUn.SCR.SLEEPONEXIT = 1 ● Complete execution of all exception handlers ● A valid interrupt request cannot be accepted to CPU*24 (including a transition from the time WFI instruction is executed to the time the transition to Software Standby mode is completed)		
Canceling method	Interrupts shown in Table 11.4. Any reset available in the mode.	Interrupts shown in Table 11.4. Any reset available in the mode.		
State after cancellation by an interrupt	Program execution state	Reset state		
State after cancellation by a reset	Reset state	Reset state		
Main clock oscillator	Selectable*10	Stop		
Sub-clock oscillator	Selectable	Selectable		

Table 11.3 Operating state of each low power mode (2 of 3)

Item	Software Standby mode (SSTBY)	Deep Software Standby mode (DSTBY)		
	SSTBY	DSTBY 1	DSTBY 2	DSTBY 3
High-speed on-chip oscillator	Selectable*11	Stop		
Middle-speed on-chip oscillator	Selectable*18	Stop*18		
Low-speed on-chip oscillator	Selectable*2	Selectable*2	Stop	
PLL1	Stop	Stop		
PLL2	Stop	Stop		
Oscillation stop detection function	Selectable*12	Stop		
Sub-clock Oscillation stop detection function	Selectable	Selectable		
Clock/buzzer output function	Selectable*3	Stop (Undefined)		
External Bus (EBCLK)	Stop (Retained)	Stop (Retained)		
CPU	Stop (Retained)	Stop (Undefined)		
TCM (SRAM)	Stop (Retained)*14	Stop (Undefined)		
User SRAM	Stop (Retained) *13	Stop (Undefined)		
Backup register	Stop (Retained)	Stop (Retained)		
MRAM	Stop (Retained)	Stop (Retained)		
Independent Watchdog Timer (IWDT)	Selectable*1	Selectable*1	Stop (Undefined)	
ARM Debug function	Stop*15	Stop*15		
Trace function	Stop*16	Stop*16		
Layer3 Ethernet Switch Module (ESWM)	Stop (Undefined)	Stop (Undefined)		
USB 2.0 Full-Speed (USBFS)	Stop (Retained)*22 Detection of USB resumption is possible.	Stop (Retained) Detection of USB resumption is possible.	Stop (Undefined)	
USB 2.0 High-Speed Module (USBHS)	Stop (Retained)*22 Detection of USB resumption is possible.	Stop (Retained) Detection of USB resumption is possible.	Stop (Undefined)	
Realtime Clock (RTC)	Selectable	Selectable	Selectable*4	
CAN with Flexible Data-rate (CANFD)	Stop (Undefined)	Stop (Undefined)		
SD/MMC Host Interface (SDHI)	Stop (Undefined)	Stop (Undefined)		
Ultra low power Timer (ULPT)	Selectable	Selectable	Stop (Undefined)	
Asynchronous General Purpose Timer (AGT)	Selectable*5	Stop (Undefined)		
I2C Bus Interface (IIC0)	Selectable*6	Stop (Undefined)		
I2C Bus Interface (IICn, n ≥1)	Stop (Retained)	Stop (Undefined)		
I3C Bus Interface (I3C)	Selectable*21	Stop (Undefined)		

Table 11.3 Operating state of each low power mode (3 of 3)

Item	Software Standby mode (SSTBY)	Deep Software Standby mode (DSTBY)		
	SSTBY	DSTBY 1	DSTBY 2	DSTBY 3
Capture Engine Unit (CEU)	Stop (Retained)*23	Stop (Undefined)		
12-Bit D/A Converter (DAC12)	Selectable	Selectable		
Temperature sensor (TSN)	Selectable*20	Selectable*20		
High-Speed Analog Comparator 0 (ACMPHS0)	Selectable	Stop (Undefined)		
High-Speed Analog Comparator n (ACMPHSn, n≥1)	Selectable*19	Stop (Undefined)		
Pulse Density Modulator Interface (PDMIF)	Selectable	Stop (Undefined)		
IRQn (n ≥ 0) pin interrupt	Selectable	Stop (Undefined)		
NMI, IRQn-DS (n ≥ 0) pin interrupt	Selectable	Selectable		
Programmable Voltage Detect (PVD)	Selectable	Selectable*7	Selectable*7	Stop (Undefined)*7
VCC voltage drop detection for VBATT	Selectable	Selectable*17	Selectable*17	Stop (Undefined)
Power-on reset circuit	Operating	Operating	Operating	Operating*8
CVM reset circuit	Selectable	Stop	Stop	Stop
Other peripheral modules	Stop (Retained)	Stop (Undefined)		
I/O Ports	Retained*9	Retained*9		

Note: **Selectable** means that operating or not operating can be selected by setting the control registers before low power mode.
Stop (Retained) means that the contents of the internal registers are retained but the operations are suspended.
Stop (Undefined) means that the contents of the internal registers are undefined and power to the internal circuit is cut off. After canceling from each low power mode, the internal registers are initialized.

Note 1. In IWDT, operating or stopping is selected by setting the IWDT Stop Control bit (IWDTSTPCTL) in Option Function Select register 0 (OFS0) in IWDT auto start mode. Also, operating or stopping is selected by setting the IWDT.IWDTCSTPR.SLCSTP bit in IWDT register start mode.

Note 2. If IWDT is not used and LOCOCR.LCSTP = 0, LOCO is not stopped in Software Standby mode or Deep Software Standby mode 1.
If IWDT is used and IWDT Stop Control bit is 0 (OFS.IWDTSTPCTL = 0 or IWDT.IWDTCSTPR.SLCSTP = 0), LOCO is not stopped regardless the value of LOCOCR.LCSTP in Software Standby mode or Deep Software Standby mode 1.
If IWDT is used and IWDT Stop Control bit is 1 (OFS.IWDTSTPCTL = 1 or IWDT.IWDTCSTPR.SLCSTP = 1), LOCO is not stopped with LOCOCR.LCSTP = 0 in Software Standby mode or Deep Software Standby mode 1.
In other cases, LOCO is stopped in Software Standby mode or Deep Software Standby mode 1.

Note 3. Stopped when the clock output source select bits (CKOCR.CKOSEL[2:0]) are set to a value other than 010 (LOCO) and 100 (SOSC).

Note 4. Only sub clock oscillator can be selected as the count source clock for RTC. When the RCR4.RCKSEL bit set to 1 (LOCO), the LPSCR must set to Deep Software Standby mode 1 before entering Deep Software Standby mode.

Note 5. AGT0 operation is possible when 100 (AGTLCLK) or 110 (AGTSCLK) is selected by the AGT0.AGTMR1.TCK[2:0] bits.
AGT1 operation is possible when 100 (AGTLCLK), 110 (AGTSCLK) or 101 (Underflow event signal from AGT0) is selected by the AGT1.AGTMR1.TCK[2:0] bits.

Note 6. Only IIC0 wakeup interrupt is available.

Note 7. When using PVD in Deep Software Standby mode, LPSCR must be set to DSTBY1 or DSTBY2 before entering Deep Software Standby mode.

Note 8. When the MCU enters Deep Software Standby mode 3, the PVD circuit stops.

Note 9. For the address bus and bus control signals (For SRAM: [A00 to A23, CS0 to CS7, RD, WR0 to WR3, WR, BC0 to BC3 and ALE], and for SDRAM: [A00 to A16, DQM0 to DQM3, SDCS, RAS, CAS, WE and CKE]), keeping the output state or changing to the high-impedance state can be selected by SBYCR.OPE bit.

Note 10. When MOSCSCR.MOSCSOKP = 1 while MOSC is oscillated, MOSC continues to oscillate in Software Standby mode.

Note 11. When HOCOSCR.HOCOSOKP = 1 while HOCO is oscillated, HOCO continues to oscillate in Software Standby mode.

Note 12. The function is depending on the setting value of the MOSCSCR.MOSCSOKP bit.

Note 13. If PDRAMSCR0.RKEEPn bit is set to 0, the contents of the target User SRAM is not retained.

Note 14. If PDRAMSCR1.RKEEPn bit is set to 0, the contents of the target TCM is not retained.

Note 15. This function is operating if debugger is connecting before entering this mode.

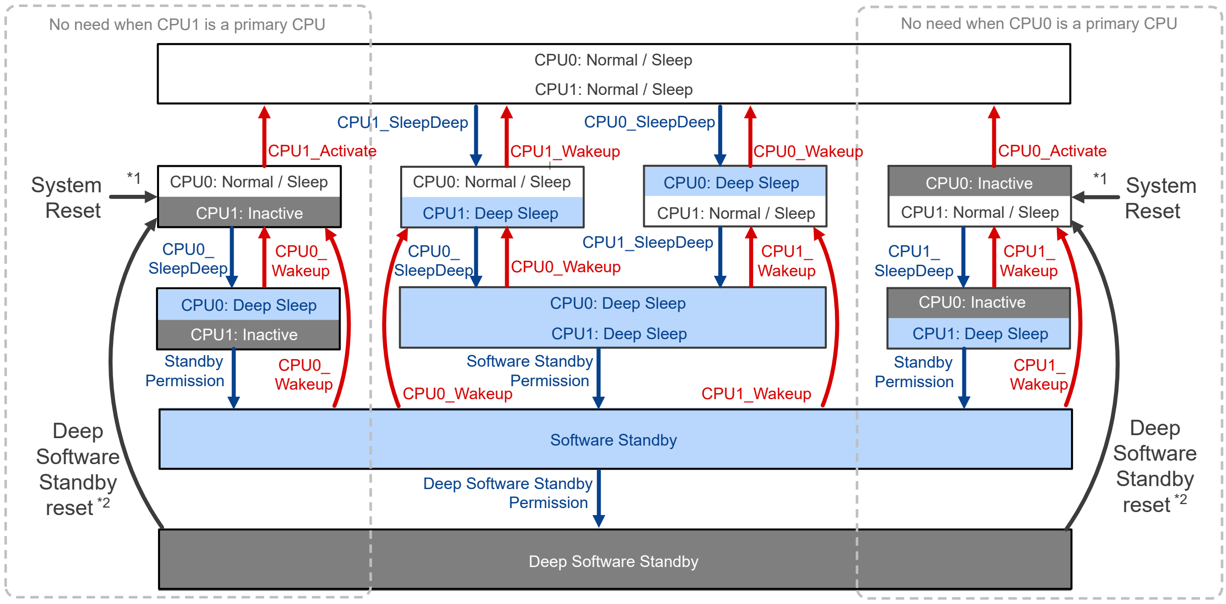
Note 16. This function is operating if debugger is connecting and TRCKCR.TRCKEN = 1 before entering this mode.
Note 17. When the MCU enters the Deep Software Standby mode 1 or 2 with that OFS1.PVDAS = 0 and OFS1.PVDLPSEL = 0, the battery power supply switch is controlled by the voltage monitor 0 which has lower power consumption than VCC voltage drop detection circuit dedicated for VBATT. When the MCU enters the Deep Software Standby mode 1 or 2 with that OFS1.PVDLPSEL = 1, the battery power supply switch is controlled by the VCC voltage drop detection circuit dedicated for VBATT. See section 12, Battery Backup Function and section 62, Electrical Characteristics for details.
Note 18. When the on-chip debugger function is valid, MOCO does not stop in this mode.
Note 19. Only VCOUT function is permitted. The VCOUT pin operates when ACMPHS uses no digital filter. For details on digital filter, see section 56, High-Speed Analog Comparator (ACMPHS).
Note 20. Only temperature monitor reset function can be operated.
Note 21. Only I3C wakeup interrupt is available.
Note 22. FIFO buffer is not retained.
Note 23. Write buffer is not retained.
Note 24. Valid interrupt requests are any interrupt or exception that are not masked by the priority level of current exception and the priority level set by BASEPRI. In addition, if the interrupt request is based on IELSRn, the interrupt must be enabled by NVIC_ISERn.

Table 11.4 Interrupt source for canceling Software Standby and Deep Software Standby modes

Interrupt source	Name	Software Standby mode	Deep Software Standby mode		
		SSTBY	DSTBY1	DSTBY2	DSTBY3
NMI		Yes	Yes	Yes	Yes
Port	PORT_IRQn	Yes	No	No	No
	PORT_IRQn-DS	Yes	Yes	Yes	Yes
PVD	PVD_PVDm (m = 1, 2)	Yes	Yes	Yes	No
IWDT	IWDT_NMIUNDF	Yes	Yes	No	No
USBFS	USBFS_USBR	Yes	Yes	No	No
USBHS	USBHS_USBIR	Yes	Yes	No	No
RTC	RTC_ALM	Yes	Yes	Yes	Yes
	RTC_PRD	Yes	Yes	Yes	Yes
ULPT0	ULPT0_ULPTI	Yes	Yes	No	No
	ULPT0_ULPTCMAI	Yes	No	No	No
	ULPT0_ULPTCMBI	Yes	No	No	No
ULPT1	ULPT1_ULPTI	Yes	Yes	No	No
	ULPT1_ULPTCMAI	Yes	No	No	No
	ULPT1_ULPTCMBI	Yes	No	No	No
AGT1	AGT1_AGTI	Yes	No	No	No
	AGT1_AGTCMAI	Yes	No	No	No
	AGT1_AGTCMBI	Yes	No	No	No
ACMPHS0	ACMP_HS0	Yes	No	No	No
IIC0	IIC0_WUI	Yes	No	No	No
I3C	I3C_WU	Yes	No	No	No
PDMIF	PDM_SDET	Yes	No	No	No
Sub-clock Oscillation stop detection function	SOSC_STOP	Yes	Yes	Yes	Yes
BBF (Battery Backup Function)	VBATT_TADI	Yes	Yes	Yes	Yes

Figure 11.1 lists the low power mode transitions.

For detail, see section 11.6. Low Power Modes.



Note: When an interrupt that acts as a trigger for cancellation of CPU Sleep mode or CPU Deep Sleep mode is received during a transition to CPU Sleep mode or CPU Deep Sleep mode, the CPU that receives the interrupt does not transit to CPU Sleep mode or CPU Deep Sleep mode but go back to Normal mode. When an interrupt that acts as a trigger for cancellation of CPU Sleep mode or CPU Deep Sleep mode is received during a transition to Software Standby mode, the MCU does not transit to Software Standby mode but go back to the state before the transition. When entering to Deep Software Standby mode, if an interrupt occurs after the state has transitioned, after mode transition is completed, returning with Deep Software Standby reset.

Note: When the transition to Normal mode is done because of an interrupt from CPU Sleep mode, CPU Deep Sleep mode or Software Standby mode, the clock source is the same as before entering each low power modes.

Note: For interrupt sources for canceling CPU Sleep mode and CPU Deep Sleep mode, see section 2.7. Sleep Modes for Low Power States.

Note: For interrupt sources for canceling Software Standby mode and Deep Software Standby mode, see Table 11.4.

Note 1. The MOCO clock is the source of the operating clock following a transition from the reset state to Normal mode.

Note 2. When an available interrupt request is generated, an internal reset (Deep Software Standby reset) is generated over a fixed period. Canceling of Deep Software Standby mode accompanies release from the internal reset state, and then the primary CPU transitions to Normal mode and executes reset exception processing with the MOCO clock as the source of the operating clock. The secondary CPU becomes inactive.

Figure 11.1 Low power mode transitions

11.2 Register Descriptions

11.2.1 LPMSAR : Low Power Mode Security Attribution Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x3C8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	NONSEC21	—	NONSEC19	NONSEC18	NONSEC17	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	NONSEC8	—	—	—	NONSEC4	—	NONSEC2	NONSEC1	NONSEC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NONSEC0	Non-secure Attribute bit 0 Target register: OPCCR 0: Secure 1: Non-secure	R/W
1	NONSEC1	Non-secure Attribute bit 1 Target register: LPSCR 0: Secure 1: Non-secure	R/W
2	NONSEC2	Non-secure Attribute bit 2 Target register: SBYCR, SSCR1, SVSCR 0: Secure 1: Non-secure	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	NONSEC4	Non-secure Attribute bit 4 Target register: VSCR 0: Secure 1: Non-secure	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	NONSEC8	Non-secure Attribute bit 8 Target register: DPSBYCR 0: Secure 1: Non-secure	R/W
16:9	—	These bits are read as 0. The write value should be 0.	R/W
17	NONSEC17	Non-secure Attribute bit 17 Target register: HOCOLDOCR 0: Secure 1: Non-secure	R/W
18	NONSEC18	Non-secure Attribute bit 18 Target register: PLL1LDOCR 0: Secure 1: Non-secure	R/W
19	NONSEC19	Non-secure Attribute bit 19 Target register: PLL2LDOCR 0: Secure 1: Non-secure	R/W
20	—	This bit is read as 0. The write value should be 0.	R/W
21	NONSEC21	Non-secure Attribute bit 21 Target register: LVOCR, MWMCR 0: Secure 1: Non-secure	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

The LPMSAR register controls the secure attribute of Low Power Mode registers.

NONSEC0 bit (Non-secure Attribute bit 0)

This bit controls the security attribute of OPCCR.

NONSEC1 bit (Non-secure Attribute bit 1)

This bit controls the security attribute of LPSCR.

NONSEC2 bit (Non-secure Attribute bit 2)

This bit controls the security attribute of SBYCR, SSCR1, SVSCR.

NONSEC4 bit (Non-secure Attribute bit 4)

This bit controls the security attribute of VSCR.

NONSEC8 bit (Non-secure Attribute bit 8)

This bit controls the security attribute of DPSBYCR.

NONSEC17 bit (Non-secure Attribute bit 17)

This bit controls the security attribute of HOCOLDOCR.

NONSEC18 bit (Non-secure Attribute bit 18)

This bit controls the security attribute of PLL1LDOCR.

NONSEC19 bit (Non-secure Attribute bit 19)

This bit controls the security attribute of PLL2LDOCR.

NONSEC21 bit (Non-secure Attribute bit 21)

This bit controls the security attribute of LVOCR and MWMCR.

11.2.2 PGCSAR : Power Gating Control Security Attribution Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x3D8

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	NONSEC7	NONSEC6	NONSEC5	NONSEC4	NONSEC3	—	—	NONSEC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NONSEC0	Non-secure Attribute bit 0 Target register: PGSCR 0: Secure 1: Non-secure	R/W
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	NONSEC3	Non-secure Attribute bit 3 Target register: PSSTCR2 0: Secure 1: Non-secure	R/W
4	NONSEC4	Non-secure Attribute bit 4 Target register: PSSTCR3 0: Secure 1: Non-secure	R/W
5	NONSEC5	Non-secure Attribute bit 5 Target register: PSSTCR4 0: Secure 1: Non-secure	R/W
6	NONSEC6	Non-secure Attribute bit 6 Target register: PSSTCR5, PDCTRESWM 0: Secure 1: Non-secure	R/W
7	NONSEC7	Non-secure Attribute bit 7 Target register: CPUDSCR 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

The PGCSAR register controls the secure attribute of power gating registers.

NONSEC0 bit (Non-secure Attribute bit 0)

This bit controls the security attribute of PGSCR.

NONSEC3 bit (Non-secure Attribute bit 3)

This bit controls the security attribute of PSSTCR2.

NONSEC4 bit (Non-secure Attribute bit 4)

This bit controls the security attribute of PSSTCR3.

NONSEC5 bit (Non-secure Attribute bit 5)

This bit controls the security attribute of PSSTCR4.

NONSEC6 bit (Non-secure Attribute bit 6)

This bit controls the security attribute of PSSTCR5, PDCTRESWM.

NONSEC7 bit (Non-secure Attribute bit 7)

This bit controls the security attribute of CPUDSCR.

11.2.3 DPFSAR : Deep Software Standby Interrupt Factor Security Attribution Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x3E0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DPFSA31	DPFSA30	DPFSA29	—	DPFSA27	DPFSA26	DPFSA25	DPFSA24	—	—	—	DPFSA20	DPFSA19	DPFSA18	DPFSA17	DPFSA16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
7:0	DPFSA7 to DPFSA0	Deep Software Standby Interrupt Factor Security Attribute bit n (n = 0 to 7) Target register: DPSIER0.bn, DPSIFR0.bn, DPSIEGR0.bn (n = 0 to 7) Target factor : IRQn-DS Pin (n = 0 to 7) 0: Secure 1: Non-secure	R/W
15:8	DPFSA15 to DPFSA8	Deep Software Standby Interrupt Factor Security Attribute bit n (n = 8 to 15) Target register: DPSIER1.bn, DPSIFR1.bn, DPSIEGR1.bn (n = 0 to 7) Target factor : IRQn-DS Pin (n = 8 to 15) 0: Secure 1: Non-secure	R/W
16	DPFSA16	Deep Software Standby Interrupt Factor Security Attribute bit 16 Target register: DPSIER2.b0, DPSIFR2.b0, DPSIEGR2.b0 Target factor: PVD1 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
17	DPFSA17	Deep Software Standby Interrupt Factor Security Attribute bit 17 Target register: DPSIER2.b1, DPSIFR2.b1, DPSIEGR2.b1 Target factor: PVD2 0: Secure 1: Non-secure	R/W
18	DPFSA18	Deep Software Standby Interrupt Factor Security Attribute bit 18 Target register: DPSIER2.b2, DPSIFR2.b2 Target factor: RTC Interval 0: Secure 1: Non-secure	R/W
19	DPFSA19	Deep Software Standby Interrupt Factor Security Attribute bit 19 Target register: DPSIER2.b3, DPSIFR2.b3 Target factor: RTC Alarm 0: Secure 1: Non-secure	R/W
20	DPFSA20	Deep Software Standby Interrupt Factor Security Attribute bit 20 Target register: DPSIER2.b4, DPSIFR2.b4, DPSIEGR2.b4 Target factor: NMI Pin 0: Secure 1: Non-secure	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
24	DPFSA24	Deep Software Standby Interrupt Factor Security Attribute bit 24 Target register: DPSIER3.b0, DPSIFR3.b0 Target factor: USBFS Suspend/Resume 0: Secure 1: Non-secure	R/W
25	DPFSA25	Deep Software Standby Interrupt Factor Security Attribute bit 25 Target register: DPSIER3.b1, DPSIFR3.b1 Target factor: USBHS Suspend/Resume 0: Secure 1: Non-secure	R/W
26	DPFSA26	Deep Software Standby Interrupt Factor Security Attribute bit 26 Target register: DPSIER3.b2, DPSIFR3.b2 Target factor: ULPT0 0: Secure 1: Non-secure	R/W
27	DPFSA27	Deep Software Standby Interrupt Factor Security Attribute bit 27 Target register: DPSIER3.b3, DPSIFR3.b3 Target factor: ULPT1 0: Secure 1: Non-secure	R/W
28	—	This bit is read as 0. The write value should be 0.	R/W
29	DPFSA29	Deep Software Standby Interrupt Factor Security Attribute bit 29 Target register: DPSIER3.b5, DPSIFR3.b5 Target factor: IWDT Overflow 0: Secure 1: Non-secure	R/W
30	DPFSA30	Deep Software Standby Interrupt Factor Security Attribute bit 30 Target register: DPSIER3.b6, DPSIFR3.b6 Target factor: Sub-clock Oscillation stop detection 0: Secure 1: Non-secure	R/W
31	DPFSA31	Deep Software Standby Interrupt Factor Security Attribute bit 31 Target register: DPSIER3.b7, DPSIFR3.b7 Target factor: Tamper Detection 0: Secure 1: Non-secure	R/W

Note: S-TYPE-1, P-TYPE-1

The DPFSAR register controls the secure attribute of Deep Software Standby Interrupt Factor control registers.

DPFSA bits (Deep Software Standby Interrupt Factor Security Attribute bit n (n = 0 to 7))

This bit controls the security attribute of DPSIER0.bn, DPSIFR0.bn, DPSIEGR0.bn (n = 0 to 7).

Target factor is IRQn-DS Pin (n = 0 to 7).

DPFSA bit (Deep Software Standby Interrupt Factor Security Attribute bit n (n = 8 to 15))

This bit controls the security attribute of DPSIER1.bn, DPSIFR1.bn, DPSIEGR1.bn (n = 0 to 7).

Target factor is IRQn-DS Pin (n = 8 to 15).

DPFSA16 bit (Deep Software Standby Interrupt Factor Security Attribute bit 16)

This bit controls the security attribute of DPSIER2.b0, DPSIFR2.b0, DPSIEGR2.b0.

Target factor is PVD1.

DPFSA17 bit (Deep Software Standby Interrupt Factor Security Attribute bit 17)

This bit controls the security attribute of DPSIER2.b1, DPSIFR2.b1, DPSIEGR2.b1.

Target factor is PVD2.

DPFSA18 bit (Deep Software Standby Interrupt Factor Security Attribute bit 18)

This bit controls the security attribute of DPSIER2.b2, DPSIFR2.b2.

Target factor is RTC Interval.

DPFSA19 bit (Deep Software Standby Interrupt Factor Security Attribute bit 19)

This bit controls the security attribute of DPSIER2.b3, DPSIFR2.b3.

Target factor is RTC Alarm.

DPFSA20 bit (Deep Software Standby Interrupt Factor Security Attribute bit 20)

This bit controls the security attribute of DPSIER2.b4, DPSIFR2.b4, DPSIEGR2.b4.

Target factor is NMI Pin.

DPFSA24 bit (Deep Software Standby Interrupt Factor Security Attribute bit 24)

This bit controls the security attribute of DPSIER3.b0, DPSIFR3.b0.

Target factor is USBFS Suspend/Resume.

DPFSA25 bit (Deep Software Standby Interrupt Factor Security Attribute bit 25)

This bit controls the security attribute of DPSIER3.b1, DPSIFR3.b1.

Target factor is USBHS Suspend/Resume.

DPFSA26 bit (Deep Software Standby Interrupt Factor Security Attribute bit 26)

This bit controls the security attribute of DPSIER3.b2, DPSIFR3.b2.

Target factor is ULPT0.

DPFSA27 bit (Deep Software Standby Interrupt Factor Security Attribute bit 27)

This bit controls the security attribute of DPSIER3.b3, DPSIFR3.b3.

Target factor is ULPT1.

DPFSA29 bit (Deep Software Standby Interrupt Factor Security Attribute bit 29)

This bit controls the security attribute of DPSIER3.b5, DPSIFR3.b5.

Target factor is IWDT Overflow.

DPFSA30 bit (Deep Software Standby Interrupt Factor Security Attribute bit 30)

This bit controls the security attribute of DPSIER3.b6, DPSIFR3.b6.

Target factor is Sub-clock Oscillation stop detection.

DPFSA31 bit (Deep Software Standby Interrupt Factor Security Attribute bit 31)

This bit controls the security attribute of DPSIER3.b7, DPSIFR3.b7.

Target factor is Tamper Detection.

11.2.4 DPFSAR1 : Deep Software Standby Interrupt Factor Security Attribution Register 1

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x3E8

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

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

Bit	Symbol	Function	R/W
7:0	DPFSA7 to DPFSA0	Deep Software Standby Interrupt Factor Security Attribute bit n (n = 0 to 7) Target register: DPSIER4.bn, DPSIFR4.bn, DPSIEGR3.bn (n =0 to 7) Target factor: IRQn-DS Pin (n = 16 to 23) 0: Secure 1: Non-secure	R/W
15:8	DPFSA15 to DPFSA8	Deep Software Standby Interrupt Factor Security Attribute bit n (n = 8 to 15) Target register: DPSIER5.bn, DPSIFR5.bn, DPSIEGR4.bn (n = 0 to 7) Target factor: IRQn-DS Pin (n = 24 to 31) 0: Secure 1: Non-secure	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

The DPFSAR1 register controls the secure attribute of Deep Software Standby Interrupt Factor control registers.

DPFSA bit (Deep Software Standby Interrupt Factor Security Attribute bit n (n = 0 to 7))

This bit controls the security attribute of DPSIER4.bn, DPSIFR4.bn, DPSIEGR3.bn (n = 0 to 7).

Target factor is IRQn-DS Pin (n = 16 to 23).

DPFSA bit (Deep Software Standby Interrupt Factor Security Attribute bit n (n = 8 to 15))

This bit controls the security attribute of DPSIER5.bn, DPSIFR5.bn, DPSIEGR4.bn.

Target factor is IRQn-DS Pin (n = 24 to 31).

11.2.5 RSCSAR : RAM Standby Control Security Attribution Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x3E4

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

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

Bit	Symbol	Function	R/W
12:0	RSCSA[12:0]	RAM Standby Control Security Attribute bit n (n =0 to 12) Target register: PDRAMSCR0.bn (n = 0 to 12) 0: Secure 1: Non-secure	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
17:16	RSCSA[17:16]	RAM Standby Control Security Attribute bit n (n = 16 to 17) Target register: PDRAMSCR1.bn (n = 0 to 1) 0: Secure 1: Non-secure	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

The RSCSAR register controls the secure attribute of RAM Standby Control registers.

RSCSA[12:0] bit (RAM Standby Control Security Attribute bit n (n =0 to 12))

This bit controls the security attribute of PDRAMSCR0.bn (n = 0 to 12).

RSCSA[17:16] bit (RAM Standby Control Security Attribute bit n (n = 16 to 17))

This bit controls the security attribute of PDRAMSCR1.bn (n = 0 to 1).

11.2.6 MSTPCRA : Module Stop Control Register A

Base address: MSTP = 0x4020_3000
MSTP_NS = 0x5020_3000

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	MSTPA23	MSTPA22	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MSTPA3	MSTPA2	MSTPA1	MSTPA0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0

Bit	Symbol	Function	R/W
0	MSTPA0	SRAM0 Module Stop*2 Target module: SRAM0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Bit	Symbol	Function	R/W
1	MSTPA1	SRAM1 Module Stop*2 Target module: SRAM1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
2	MSTPA2	SRAM2 Module Stop*2 Target module: SRAM2 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
3	MSTPA3	SRAM3 Module Stop*2 Target module: SRAM3 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
21:4	—	These bits are read as 1. The write value should be 1.	R/W
22	MSTPA22	DMA Controller 0 and Data Transfer Controller 0 Module Stop*1 Target modules: DMAC0, DTC0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
23	MSTPA23	DMA Controller 1 and Data Transfer Controller 1 Module Stop*1 Target module: DMAC1, DTC1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
31:24	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-4, P-TYPE-2
Note 1. When rewriting the MSTPAi bit from 0 to 1, disable the DMAC and DTC before setting MSTPAi bit (i = 22, 23).
Note 2. When changing the value of this bit, only execute subsequent instructions after reading this bit to check that the value was updated.

11.2.7 MSTPCRB : Module Stop Control Register B

Base address: MSTP = 0x4020_3000
MSTP_NS = 0x5020_3000

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MSTPB31	MSTPB30	MSTPB29	MSTPB28	MSTPB27	MSTPB26	MSTPB25	MSTPB24	MSTPB23	MSTPB22	—	—	MSTPB19	MSTPB18	MSTPB17	MSTPB16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	MSTPB12	MSTPB11	—	MSTPB9	MSTPB8	MSTPB7	—	—	MSTPB4	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 1. The write value should be 1.	R/W
4	MSTPB4	I3C Bus Interface Module Stop*1 Target module: I3C 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
6:5	—	These bits are read as 1. The write value should be 1.	R/W
7	MSTPB7	IIC Bus Interface 2 Module Stop Target module: IIC2 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Bit	Symbol	Function	R/W
8	MSTPB8	IIC Bus Interface 1 Module Stop Target module: IIC1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
9	MSTPB9	IIC Bus Interface 0 Module Stop Target module: IIC0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
10	—	This bit is read as 1. The write value should be 1.	R/W
11	MSTPB11	Universal Serial Bus 2.0 FS Interface 0 Module Stop*2 Target module: USBFS0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
12	MSTPB12	Universal Serial Bus 2.0 HS Interface Module Stop*2 Target module: USBHS 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
15:13	—	These bits are read as 1. The write value should be 1.	R/W
16	MSTPB16	Octal Serial Peripheral Interface 0 and Decryption On The Fly 0 Module Stop*3 Target module: OSPI0 and DOTF0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
17	MSTPB17	Octal Serial Peripheral Interface 1 and Decryption On The Fly 1 Module Stop*3 Target module: OSPI1 and DOTF1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
18	MSTPB18	Serial Peripheral Interface 1 Module Stop*4 Target module: SPI1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
19	MSTPB19	Serial Peripheral Interface 0 Module Stop*4 Target module: SPI0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
21:20	—	These bits are read as 1. The write value should be 1.	R/W
22	MSTPB22	Serial Communication Interface 9 Module Stop*5 Target module: SCI9 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
23	MSTPB23	Serial Communication Interface 8 Module Stop*5 Target module: SCI8 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
24	MSTPB24	Serial Communication Interface 7 Module Stop*5 Target module: SCI7 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
25	MSTPB25	Serial Communication Interface 6 Module Stop*5 Target module: SCI6 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
26	MSTPB26	Serial Communication Interface 5 Module Stop*5 Target module: SCI5 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Bit	Symbol	Function	R/W
27	MSTPB27	Serial Communication Interface 4 Module Stop*5 Target module: SCI4 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
28	MSTPB28	Serial Communication Interface 3 Module Stop*5 Target module: SCI3 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
29	MSTPB29	Serial Communication Interface 2 Module Stop*5 Target module: SCI2 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
30	MSTPB30	Serial Communication Interface 1 Module Stop*5 Target module: SCI1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
31	MSTPB31	Serial Communication Interface 0 Module Stop*5 Target module: SCI0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Note: S-TYPE-4, P-TYPE-4
Note 1. The MSTPB4 bit should be written after the I3C clock (I3CCLK) is stable. To enter Software Standby mode, the WFI instruction should be executed after 2 or more cycles of I3CCLK following the change of MSTPB4.
Note 2. The MSTPB11 and MSTPB12 bits should be written after the USB clock (USBCLK) is stable. To enter Software Standby mode, the WFI instruction should be executed after 2 or more cycles of USBCLK following the change of MSTPB11 and MSTPB12.
Note 3. The MSTPB16 and MSTPB17 bits should be written after the Octal-SPI clock (OCTACLK) is stable. To enter Software Standby mode, the WFI instruction should be executed after 2 or more cycles of OCTACLK following the change of MSTPB16 and MSTPB17. When changing the value of this bit, only execute subsequent instructions after reading this bit to check that the value was updated.
Note 4. The MSTPB18 and MSTPB19 bits should be written after the SPI clock (SPICLK) is stable. To enter Software Standby mode, the WFI instruction should be executed after 2 or more cycles of SPICLK following the change of MSTPB18 and MSTPB19.
Note 5. The MSTPBi bits should be written after the SCI clock (SCICLK) is stable. To enter Software Standby mode, the WFI instruction should be executed after 2 or more cycles of SCICLK following the change of MSTPBi (i = 22 to 31).

11.2.8 MSTPCRC : Module Stop Control Register C

Base address: MSTP = 0x4020_3000
MSTP_NS = 0x5020_3000

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MSTPC31	MSTPC30	—	MSTPC28	MSTPC27	MSTPC26	—	MSTPC24	—	—	—	—	—	—	—	MSTPC16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	MSTPC14	MSTPC13	MSTPC12	MSTPC11	—	—	MSTPC8	MSTPC7	—	—	—	—	—	MSTPC1	MSTPC0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	MSTPC0	Clock Frequency Accuracy Measurement Circuit Module Stop*1 Target module: CAC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
1	MSTPC1	Cyclic Redundancy Check Calculator Module Stop Target module: CRC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Bit	Symbol	Function	R/W
6:2	—	These bits are read as 1. The write value should be 1.	R/W
7	MSTPC7	Serial Sound Interface Enhanced 1 Module Stop Target module: SSIE1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
8	MSTPC8	Serial Sound Interface Enhanced 0 Module Stop Target module: SSIE0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
10:9	—	These bits are read as 1. The write value should be 1.	R/W
11	MSTPC11	Secure Digital Host IF 1 Module Stop Target module: SDHI1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
12	MSTPC12	Secure Digital Host IF 0 Module Stop Target module: SDHI0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
13	MSTPC13	Data Operation Circuit Module Stop Target module: DOC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
14	MSTPC14	Event Link Controller Module Stop Target module: ELC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
15	—	This bit is read as 1. The write value should be 1.	R/W
16	MSTPC16	Capture Engine Unit Module Stop Target module: CEU 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
23:17	—	These bits are read as 1. The write value should be 1.	R/W
24	MSTPC24	Pulse Density Modulation Interface Module Stop*2 Target module: PDMIF 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
25	—	This bit is read as 1. The write value should be 1.	R/W
26	MSTPC26	Controller Area Network with Flexible Data-Rate 1 Module Stop*3 Target module: CANFD1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
27	MSTPC27	Controller Area Network with Flexible Data-Rate 0 Module Stop*3 Target module: CANFD0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
28	MSTPC28	Ether-PHY clock Module Stop*4 Target module: ETHPHYCLK 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
29	—	This bit is read as 1. The write value should be 1.	R/W
30	MSTPC30	Layer 3 Ethernet Switch Module Module Stop*5 Target module: ESWM 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Bit	Symbol	Function	R/W
31	MSTPC31	Renesas Secure IP Module Stop Target module: RSIP-E50D 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Note: S-TYPE-4, P-TYPE-4

Note 1. The MSTPC0 bit should be written after the CAC clock is stable. To enter Software Standby mode, the WFI instruction should be executed after 2 or more cycles of the CAC clock following the change of MSTPC0.

Note 2. The MSTPC24 bit should be written after the PDM clock (PDMCLK) is stable. To enter Software Standby mode, the WFI instruction should be executed after 2 or more cycles of PDMCLK following the change of MSTPC24.

Note 3. The MSTPC26 and MSTPC27 bits should be written after the CANFD clock (CANFDCLK) is stable. To enter Software Standby mode, the WFI instruction should be executed after 2 or more cycles of CANFDCLK following the change of MSTPC26 and MSTPC27.

Note 4. The MSTPC28 bit should be written after the Ether-PHY clock (ETHPHYCLK) is stable. To enter Software Standby mode, the WFI instruction should be executed after 2 or more cycles of ETHPHYCLK following the change of MSTPC28.

Note 5. The MSTPC30 bit should be written after the EtherSW clock (ESWCLK) is stable. To enter Software Standby mode, the WFI instruction should be executed after 2 or more cycles of ESWCLK following the change of MSTPC30.

11.2.9 MSTPCRD : Module Stop Control Register D

Base address: MSTP = 0x4020_3000
MSTP_NS = 0x5020_3000

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	MSTPD28	MSTPD27	MSTPD26	MSTPD25	—	—	MSTPD22	MSTPD21	MSTPD20	MSTPD19	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	MSTPD14	MSTPD13	MSTPD12	MSTPD11	—	—	—	—	MSTPD6	MSTPD5	MSTPD4	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 1. The write value should be 1.	R/W
4	MSTPD4	Low power Asynchronous General Purpose Timer 1 Module Stop*1 Target module: AGT1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
5	MSTPD5	Low power Asynchronous General Purpose Timer 0 Module Stop*1 Target module: AGT0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
6	MSTPD6	PWM Delay Generation Circuit Module Stop Target module: PDG 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
10:7	—	These bits are read as 1. The write value should be 1.	R/W
11	MSTPD11	Port Output Enable for GPT Group D Module Stop Target module: POEG Group D 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
12	MSTPD12	Port Output Enable for GPT Group C Module Stop Target module: POEG Group C 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Bit	Symbol	Function	R/W
13	MSTPD13	Port Output Enable for GPT Group B Module Stop Target module: POEG Group B 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
14	MSTPD14	Port Output Enable for GPT Group A Module Stop Target module: POEG Group A 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
18:15	—	These bits are read as 1. The write value should be 1.	R/W
19	MSTPD19	12-bit D/A Converter 1 Module Stop Target module: DAC12_1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
20	MSTPD20	12-bit D/A Converter 0 Module Stop Target module: DAC12_0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
21	MSTPD21	16-Bit A/D Converter Module Stop*2 Target module: ADC16H 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
22	MSTPD22	Temperature Sensor Module Stop Target module: TSN 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
24:23	—	These bits are read as 1. The write value should be 1.	R/W
25	MSTPD25	High-Speed Analog Comparator 3 Module Stop Target module: ACMPHS3 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
26	MSTPD26	High-Speed Analog Comparator 2 Module Stop Target module: ACMPHS2 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
27	MSTPD27	High-Speed Analog Comparator 1 Module Stop Target module: ACMPHS1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
28	MSTPD28	High-Speed Analog Comparator 0 Module Stop Target module: ACMPHS0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
31:29	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-4, P-TYPE-4

Note 1. When the count source is sub-clock oscillator or LOCO, AGTn counting doesn't stop even if MSTPDi is set to 1. If the count source is the sub-clock oscillator or LOCO, this bit must be set to 1 except when accessing the AGTn registers. (n = 1 to 0) (i = 4 to 5)

Note 2. The MSTPD21 bit should be written after the ADC clock (ADCCLK) is stable. To enter Software Standby mode, the WFI instruction should be executed after 2 or more cycles of ADCCLK following the change of MSTPD21.

11.2.10 MSTPCRE : Module Stop Control Register E

Base address: MSTP = 0x4020_3000
MSTP_NS = 0x5020_3000

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MSTP E31	MSTP E30	MSTP E29	MSTP E28	MSTP E27	—	—	—	—	—	MSTP E21	MSTP E20	MSTP E19	MSTP E18	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	MSTP E9	MSTP E8	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 1. The write value should be 1.	R/W
8	MSTPE8	Ultra-Low Power Timer 1 Module Stop*1 Target module: ULPT1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
9	MSTPE9	Ultra-Low Power Timer 0 Module Stop*1 Target module: ULPT0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
17:10	—	These bits are read as 1. The write value should be 1.	R/W
18	MSTPE18	General PWM Timer 13 Module Stop*2 Target module: GPT13 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
19	MSTPE19	General PWM Timer 12 Module Stop*2 Target module: GPT12 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
20	MSTPE20	General PWM Timer 11 Module Stop*2 Target module: GPT11 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
21	MSTPE21	General PWM Timer 10 Module Stop*2 Target module: GPT10 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
26:22	—	These bits are read as 1. The write value should be 1.	R/W
27	MSTPE27	General PWM Timer 4-9 Module Stop*2 Target module: GPT4-9 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
28	MSTPE28	General PWM Timer 3 Module Stop*2 Target module: GPT3 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
29	MSTPE29	General PWM Timer 2 Module Stop*2 Target module: GPT2 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Bit	Symbol	Function	R/W
30	MSTPE30	General PWM Timer 1 Module Stop[*2] Target module: GPT1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
31	MSTPE31	General PWM Timer 0 Module Stop[*2] Target module: GPT0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Note: S-TYPE-4, P-TYPE-4

Note 1. When the count source is sub-clock oscillator or LOCO, ULPTn counting does not stop even if MSTPEi is set to 1. If the count source is the sub-clock oscillator or LOCO, this bit must be set to 1 except when accessing the ULPTn registers. (n = 1, 0) (i = 8, 9)

Note 2. The MSTPEi bits should be written after the GPT clock (GPTCLK) is stable. To enter Software Standby mode, the WFI instruction should be executed after 2 or more cycles of GPTCLK following the change of MSTPEi (i = 18 to 21, 27 to 31).

11.2.11 OPCCR : Operating Power Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x0A0

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	OPCMTSF	—	—	OPCM[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	OPCM[1:0]	Operating Power Control Mode Select 0 0: High-speed mode 0 1: Prohibited 1 0: Prohibited 1 1: Prohibited Settings other than above are prohibited.	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	OPCMTSF	Operating Power Control Mode Transition Status Flag • Read 0: Transition completed 1: During transition • Write The write value should be 0.	R
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

OPCCR is used to reduce power consumption in Normal mode and Processor low power mode. Do not change from the initial value.

11.2.12 PGSCR: Power Gating Shift Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x104

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

Bit	Symbol	Function	R/W
0	PGS	Power gating shift timing Set the power gating start timing for each domain. Set this bit to 0 before transitioning to Software Standby mode or Deep Software Standby mode. 0: Power gating control for each domain is started partial at the same time 1: Power gating control for each domain is started serially	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

11.2.13 PSSTCRn : Power Switch Control Start Time Control Register n (n = 2 to 5)

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x210 + 0x2 × n (n = 2 to 5)

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

Bit	Symbol	Function	R/W
12:0	PSST	Power switch control start time Set the power switch control start time when PGSCR.PGS = 0. Set 0x00B0 to this bit. Other setting is prohibited.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

11.2.14 PDCTRESWM : ESWM Power Domain Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x118

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

Bit	Symbol	Function	R/W
0	PDDE	Power control enable 0: Power on the target domain 1: Power off the target domain	R/W
5:1	—	These bits are read as 0. The write value should be 0.	R/W
6	PDCSF	Power control status flag 0: Power gating control is not executed (idle) 1: Power gating control is in progress	R
7	PDPGSF	Power gating status flag 0: Target domain is power on (not gating) 1: Target domain is power off (during Gating)	R

Note: S-TYPE-3, P-TYPE-2

The PDCTRESWM register controls power gating for ESWM power domain.

For restrictions on this register, not only in this section, see also section 11.5.1. Power Gating Control Using Power Domain Control Register.

PDDE bit (Power control enable)

The PDDE bit controls the power gating of the target domain. The PDDE bit should be set from 0 to 1, after confirmed that the PDCSF = 0 and PDPGSF = 0. On the other hand, the PDDE bit should be set from 1 to 0, after confirmed that the PDCSF = 0 and PDPGSF = 1.

PDCSF flag (Power control status flag)

The PDCSF flag indicates the power control status of the target domain.

PDPGSF flag (Power gating status flag)

The PDPGSF flag indicates whether the target domain is power gated.

11.2.15 PDRAMSCR0 : SRAM Power Domain Standby Control Register 0

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x140

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	RKEEP12	RKEEP11	RKEEP10	RKEEP9	RKEEP8	RKEEP7	RKEEP6	RKEEP5	RKEEP4	RKEEP3	RKEEP2	RKEEP1	RKEEP0
Value after reset:	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
12:0	RKEEP[12:0]	RAM Retention 0: When entering the Software Standby mode, the contents of the target RAM are not kept. 1: When entering the Software Standby mode, the contents of the target RAM are kept.	R/W
14:13	—	These bits are read as 1. The write value should be 1.	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-4, P-TYPE-2

The PDRAMSCR0 register controls the target memory in Software Standby mode.

RKEEPn bit (RAM Retention)

The RKEEPn (n = 0 to 12) bit controls whether to retain the contents of the target memory in software standby mode.

Table 11.5 shows the relationship between each bit and the target memory.

In addition, there are the following constraints regarding the setting of each bit.

RKEEP0 to RKEEP3

0000: When entering the Software Standby mode, the contents of the SRAM0 are not kept.

1111: When entering the Software Standby mode, the contents of the SRAM0 are kept.

Other than the above: Not supported

RKEEP4 to RKEEP7

0000: When entering the Software Standby mode, the contents of the SRAM1 are not kept.

1111: When entering the Software Standby mode, the contents of the SRAM1 are kept.

Other than the above: Not supported

RKEEP8 to RKEEP11

0000: When entering the Software Standby mode, the contents of the SRAM2 are not kept.

1111: When entering the Software Standby mode, the contents of the SRAM2 are kept.

Other than the above: Not supported

Table 11.5 Relationship between each bit and the target memory

PDRAM register bit	Target memory
PDRAM.b0	SRAM0: 0x2200_0000 to 0x2201_FFFF (Secure alias) 0x3200_0000 to 0x3201_FFFF (Non-secure alias) SRAM (CPU1 TCM)*1: 0x2A00_0000 to 0x2A01_FFFF (Secure alias) 0x3A00_0000 to 0x3A01_FFFF (Non-secure alias)
PDRAM.b1	SRAM0: 0x2202_0000 to 0x2203_FFFF (Secure alias) 0x3202_0000 to 0x3203_FFFF (Non-secure alias)
PDRAM.b2	SRAM0: 0x2204_0000 to 0x2205_FFFF (Secure alias) 0x3204_0000 to 0x3205_FFFF (Non-secure alias)
PDRAM.b3	SRAM0: 0x2206_0000 to 0x2207_FFFF (Secure alias) 0x3206_0000 to 0x3207_FFFF (Non-secure alias)
PDRAM.b4	SRAM1: 0x2208_0000 to 0x2209_FFFF (Secure alias) 0x3208_0000 to 0x3209_FFFF (Non-secure alias)
PDRAM.b5	SRAM1: 0x220A_0000 to 0x220B_FFFF (Secure alias) 0x320A_0000 to 0x320B_FFFF (Non-secure alias)
PDRAM.b6	SRAM1: 0x220C_0000 to 0x220D_FFFF (Secure alias) 0x320C_0000 to 0x320D_FFFF (Non-secure alias)
PDRAM.b7	SRAM1: 0x220E_0000 to 0x220F_FFFF (Secure alias) 0x320E_0000 to 0x320F_FFFF (Non-secure alias)
PDRAM.b8	SRAM2: 0x2210_0000 to 0x2211_FFFF (Secure alias) 0x3210_0000 to 0x3211_FFFF (Non-secure alias)
PDRAM.b9	SRAM2: 0x2212_0000 to 0x2213_FFFF (Secure alias) 0x3212_0000 to 0x3213_FFFF (Non-secure alias)
PDRAM.b10	SRAM2: 0x2214_0000 to 0x2215_FFFF (Secure alias) 0x3214_0000 to 0x3215_FFFF (Non-secure alias)
PDRAM.b11	SRAM2: 0x2216_0000 to 0x2217_FFFF (Secure alias) 0x3216_0000 to 0x3217_FFFF (Non-secure alias)
PDRAM.b12	SRAM3: 0x2218_0000 to 0x2219_FFFF (Secure alias) 0x3218_0000 to 0x3219_FFFF (Non-secure alias)

Note 1. SRAM (CPU1 TCM) is supported only for single core product.

11.2.16 PDRAMSCR1 : SRAM Power Domain Standby Control Register 1

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x142

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	RKEEP1	RKEEP0
Value after reset:	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	RKEEP0	RAM Retention 0: When entering the CPU0 Deep Sleep and Software Standby mode, the contents of the target RAM are not kept. 1: When entering the CPU0 Deep Sleep and Software Standby mode, the contents of the target RAM are kept.	R/W
1	RKEEP1*1	RAM Retention 0: When entering the Software Standby mode, the contents of the target RAM are not kept. 1: When entering the Software Standby mode, the contents of the target RAM are kept.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-4, P-TYPE-2
Note 1. RKEEP1 is valid only for a dual-core product.

The PDRAMSCR1 register controls the target memory in CPU0 Deep Sleep and Software Standby mode.

RKEEP0 bit (RAM Retention)

The RKEEP0 bit controls whether to retain the contents of the target memory in CPU0 Deep Sleep and software standby mode.

RKEEP1 bit (RAM Retention)

The RKEEP1 bit controls whether to retain the contents of the target memory in software standby mode.

Table 11.6 shows the relationship between each bit and the target memory.

Table 11.6 Relationship between each bit and the target memory

PDRAM register bit	Target memory
PDRAM.b0	CPU0 TCM
PDRAM.b1*1	CPU1 TCM

Note 1. Available only for a dual-core product.

11.2.17 SBYCR : Standby Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x00C

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

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	OPE	Output Port Enable 0: In Software Standby mode or Deep Software Standby mode, Set the address bus and other bus control signal to the high-impedance state. 1: In Software Standby mode or Deep Software Standby mode, Address bus and other bus control signal retain the output state.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

The SBYCR register controls the bus signal output in Software Standby mode and Deep Software Standby mode.

OPE bit (Output Port Enable)

The OPE bit specifies whether to set to the high-impedance state or to retain the output of the address bus and bus control signals (For SRAM: [A00 to A23, CS0 to CS7, RD, WR0 to WR3, WR, BC0 to BC3 and ALE], and for SDRAM: [A00 to A16, DQM0 to DQM3, SDCS, RAS, CAS, WE and CKE]) in Software Standby mode or Deep Software Standby mode.

11.2.18 SSCR1 : Software Standby Control Register 1

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA98

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	SS2LP[1:0]		—	SS2FR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SS2FR	Software Standby Fast Return 0: When returning from Software Standby mode, fast return function is disabled. 1: When returning from Software Standby mode, fast return function is enabled.	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
3:2	SS2LP[1:0]	Software Standby low power select 0 0: SS2LP_0 (default mode) 0 1: SS2LP_1 (Low power setting 1) Settings other than above are prohibited.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

SS2FR bit (Software Standby Fast Return)

When the SS2FR bit is 1, the recovery time from Software Standby mode is shorter. The setting of this bit is only valid when the SS2LP bit is 01b.

See section 62, Electrical Characteristics for details on the recovery time.

Note: It is recommended to set this bit to 1. Otherwise, it just increases the recovery time.

SS2LP[1:0] bits (Software Standby low power select)

When the SS2LP[1:0] bits are 01b, the power consumption of Software Standby mode is smaller but the recovery time from Software Standby mode is longer. See section 62, Electrical Characteristics for details on the power consumption and the recovery time.

11.2.19 LPSCR : Low Power State Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA90

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

Bit	Symbol	Function	R/W
3:0	LPMD[3:0]	Low power mode setting 0x0: System Active 0x4: Reserved*1 0x5: Software Standby mode 0x8: Deep Software Standby mode 1 0x9: Deep Software Standby mode 2 0xA: Deep Software Standby mode 3 Settings other than above are prohibited.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note 1. Setting prohibited when transitioning to low power mode. Change the setting to a value other than 0x4 before transitioning to low power mode.

The LPSCR register controls the transition destination of low power mode.

LPSCR is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode. For details, see section 6, Resets.

LPMD[3:0] bits (Low power mode setting)

The LPMD[3:0] bits indicate the destination of low power mode. Actual low power mode transition is executed by WFI instruction in CPU.

The transition of Software Standby mode and Deep Software Standby mode does not occur unless the WFI instruction of each CPU is executed.

When the low power mode is canceled by an interrupt, The LPMD bit remains the value before transition. Writing 0 to this bit clears the bit itself.

While MENTRYR.MENTRY bit is 1, the setting to Software Standby and Deep software Standby mode is ineffective. Even if LPMD[3:0] is any of 0x5, 0x8 to 0xA, the MCU is keep to System active state on execution of a WFI instruction. See Table 11.8 for the detail.

While OFS0.IWDTSTPCTL bit is 0 (counting continues), the setting to Deep software Standby 2 and 3 mode is ineffective. Even if LPMD is any of 0x9 or 0xA, the MCU enters Deep Software Standby 1 mode on execution of a WFI instruction.

While IWDTCSTPR.SLCSTP is 0 (counting continues), the setting to Deep software Standby 2 and 3 mode is ineffective. Even if LPMD is any of 0x9 or 0xA, the MCU enters Deep Software Standby 1 mode on execution of a WFI instruction.

See Table 11.8 for the detail.

While the voltage monitor n (n = 1, 2) reset is enabled (PVDnCR0.RI = 1), the setting to Deep software Standby mode 2, 3 is ineffective. Even if LPMD is any of 0x9 or 0xA, the MCU enters Deep Software Standby 1 mode on execution of a WFI instruction. See Table 11.8 for the detail.

When at least PVDnCR0. RE (n = 4, 5) bit is 1 and all PVDnCR0.RI (n = 1, 2) bit are 0 and OFS0.IWDTSTPCTL is 1 and IWDTCSTPR.SLCSTP is 1, the setting to Deep software Standby mode 3 is ineffective. Even if LPMD is 0xA, the MCU enters Deep Software Standby 2 mode on execution of a WFI instruction. See Table 11.8 for the detail.

11.2.20 DPSBYCR : Deep Software Standby Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	IOKEEP	—	—	DCSSMODE[1:0]		—	—
Value after reset:	0	0	0	1	0	1	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
3:2	DCSSMODE[1:0]	DCDC soft start mode Set the wake-up time from the Deep Software Standby mode with DCDC. 0 0: Setting prohibited 0 1: 128 μs 1 0: 256 μs 1 1: 512 μs	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	IOKEEP	I/O Port Retention 0: When the Deep Software Standby mode is canceled, the I/O ports and D/A output are in the reset state 1: When the Deep Software Standby mode is canceled, the I/O ports and D/A output are in the same state as in the Deep Software Standby mode	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

The DPSBYCR register controls the Deep Software Standby mode.

DPSBYCR is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode. For details, see section 6, Resets.

DCSSMODE[1:0] bits (DCDC soft start mode)

The DCSSMODE[1:0] bits specify whether to shorten the time required to recover from the Deep Software Standby mode. For details, see section 62, Electrical Characteristics.

IOKEEP bit (I/O Port Retention)

In Deep Software Standby mode, I/O ports and D/A output keep the same states as in the Software Standby mode. The IOKEEP bit specifies whether to reset the state of the I/O ports and D/A output or not when the Deep Software Standby mode is canceled.

11.2.21 DPSIER0 : Deep Software Standby Interrupt Enable Register 0

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ7E	DIRQ6E	DIRQ5E	DIRQ4E	DIRQ3E	DIRQ2E	DIRQ1E	DIRQ0E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ0E	IRQ0-DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
1	DIRQ1E	IRQ1-DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
2	DIRQ2E	IRQ2-DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
3	DIRQ3E	IRQ3-DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
4	DIRQ4E	IRQ4-DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W

Bit	Symbol	Function	R/W
5	DIRQ5E	IRQ5-DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
6	DIRQ6E	IRQ6-DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W
7	DIRQ7E	IRQ7-DS Pin Enable 0: Cancelling Deep Software Standby mode is disabled 1: Cancelling Deep Software Standby mode is enabled	R/W

Note: S-TYPE-4, P-TYPE-2

DPSIER0 is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode. For details, see section 6, Resets.

After the setting of DPSIER0 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR0 being set to 1. Therefore, DPSIFR0 should be cleared to 0 before entering Deep Software Standby mode.

See Table 11.4 to know the relation between this register and interrupt source for canceling Deep Software standby mode.

11.2.22 DPSIER1 : Deep Software Standby Interrupt Enable Register 1

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA0C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ15E	DIRQ14E	DIRQ13E	DIRQ12E	DIRQ11E	DIRQ10E	DIRQ9E	DIRQ8E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ8E	IRQ8-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
1	DIRQ9E	IRQ9-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
2	DIRQ10E	IRQ10-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
3	DIRQ11E	IRQ11-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
4	DIRQ12E	IRQ12-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
5	DIRQ13E	IRQ13-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
6	DIRQ14E	IRQ14-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
7	DIRQ15E	IRQ15-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W

Note: S-TYPE-4, P-TYPE-2

DPSIER1 is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode. For details, see section 6, Resets.

After the setting of DPSIER1 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR1 being set to 1. Therefore, DPSIFR1 should be cleared to 0 before entering Deep Software Standby mode.

See Table 11.4 to know the relation between this register and interrupt source for canceling Deep Software standby mode.

11.2.23 DPSIER2 : Deep Software Standby Interrupt Enable Register 2

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DNMIE	DRTCAIE	DRTCIIE	DPVD2IE	DPVD1IE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DPVD1IE	PVD1 Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
1	DPVD2IE	PVD2 Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
2	DRTCIIE	RTC Interval interrupt Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
3	DRTCAIE	RTC Alarm interrupt Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
4	DNMIE	NMI Pin Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W*1
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-4, P-TYPE-2

Note 1. 1 can be written only once. Once 1 is written to this bit, subsequent write accesses are disabled.

DPSIER2 is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode. For details, see section 6, Resets.

After the setting of DPSIER2 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR2 being set to 1. Therefore, DPSIFR2 should be cleared to 0 before entering Deep Software Standby mode.

See Table 11.4 to know the relation between this register and interrupt source for canceling Deep Software standby mode.

11.2.24 DPSIER3 : Deep Software Standby Interrupt Enable Register 3

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA14

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DVBATTADIE	DSOSTDIE	DIWDTIE	—	DULPT1IE	DULPT0IE	DUSBHSIE	DUSBFSIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DUSBFSIE	USBFS Suspend/Resume Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
1	DUSBHSIE	USBHS Suspend/Resume Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
2	DULPT0IE	ULPT0 Overflow Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
3	DULPT1IE	ULPT1 Overflow Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	DIWDTIE	IWDT Overflow Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
6	DSOSTDIE	Sub-clock Oscillation stop detection Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
7	DVBATTADIE	VBATT Tamper Detection Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W

Note: S-TYPE-4, P-TYPE-2

DPSIER3 is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode. For details, see section 6, Resets.

After the setting of DPSIER3 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR3 being set to 1. Therefore, DPSIFR3 should be cleared to 0 before entering Deep Software Standby mode.

See Table 11.4 to know the relation between this register and interrupt source for canceling Deep Software standby mode.

11.2.25 DPSIER4 : Deep Software Standby Interrupt Enable Register 4

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xB40

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ23E	DIRQ22E	DIRQ21E	DIRQ20E	DIRQ19E	DIRQ18E	DIRQ17E	DIRQ16E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ16E	IRQ16-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
1	DIRQ17E	IRQ17-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
2	DIRQ18E	IRQ18-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
3	DIRQ19E	IRQ19-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W

Bit	Symbol	Function	R/W
4	DIRQ20E	IRQ20-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
5	DIRQ21E	IRQ21-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
6	DIRQ22E	IRQ22-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
7	DIRQ23E	IRQ23-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W

Note: S-TYPE-4, P-TYPE-2

DPSIER4 is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode. For details, see section 6, Resets.

After the setting of DPSIER4 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR4 being set to 1. Therefore, DPSIFR4 should be cleared to 0 before entering Deep Software Standby mode.

See Table 11.4 to know the relation between this register and interrupt source for canceling Deep Software standby mode.

11.2.26 DPSIER5 : Deep Software Standby Interrupt Enable Register 5

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xB44

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ31E	DIRQ30E	DIRQ29E	DIRQ28E	DIRQ27E	DIRQ26E	DIRQ25E	DIRQ24E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ24E	IRQ24-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
1	DIRQ25E	IRQ25-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
2	DIRQ26E	IRQ26-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
3	DIRQ27E	IRQ27-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
4	DIRQ28E	IRQ28-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
5	DIRQ29E	IRQ29-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
6	DIRQ30E	IRQ30-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
7	DIRQ31E	IRQ31-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W

Note: S-TYPE-4, P-TYPE-2

DPSIER5 is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode. For details, see section 6, Resets.

After the setting of DPSIER5 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR5 being set to 1. Therefore, DPSIFR5 should be cleared to 0 before entering Deep Software Standby mode.

See Table 11.4 to know the relation between this register and interrupt source for canceling Deep Software standby mode.

11.2.27 DPSIFR0 : Deep Software Standby Interrupt Flag Register 0

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA18

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ7F	DIRQ6F	DIRQ5F	DIRQ4F	DIRQ3F	DIRQ2F	DIRQ1F	DIRQ0F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ0F	IRQ0-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
1	DIRQ1F	IRQ1-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
2	DIRQ2F	IRQ2-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
3	DIRQ3F	IRQ3-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
4	DIRQ4F	IRQ4-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
5	DIRQ5F	IRQ5-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
6	DIRQ6F	IRQ6-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
7	DIRQ7F	IRQ7-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1

Note: S-TYPE-4, P-TYPE-2
Note 1. Only 0 can be written to clear flag. The flag must be cleared by writing 0 after 1 is read.

Each flag is set to 1 when a cancel request specified by DPSIEGR0 is generated.

Each flag may be set to 1 when a cancel request is generated in any mode (not even Deep Software Standby mode) or when the setting of DPSIER0 is modified. Therefore, a transition to Deep Software Standby mode should be made after DPSIFR0 is cleared to 0x00.

To clear DPSIFR0 to 0x00 after modifying DPSIER0, wait for at least six PCLKB cycles, read DPSIFR0, and then write 0 to DPSIFR0. Six or more PCLKB cycles can be secured, for example, by reading DPSIER0.

DPSIFR0 is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode. For details, see section 6, Resets.

DIRQnF flag (IRQn-DS Pin Deep Software Standby Cancel Flag) (n = 0 to 7)

The DIRQnF flag indicates that a cancel request by the IRQn-DS pin has been generated.

[Setting condition]

A cancel request by the IRQn-DS pin specified by DPSIEGR0 is generated.

[Clearing condition]

Writing 0 to each flag after 1 is read.

11.2.28 DPSIFR1 : Deep Software Standby Interrupt Flag Register 1

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA1C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ15F	DIRQ14F	DIRQ13F	DIRQ12F	DIRQ11F	DIRQ10F	DIRQ9F	DIRQ8F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ8F	IRQ8-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
1	DIRQ9F	IRQ9-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
2	DIRQ10F	IRQ10-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
3	DIRQ11F	IRQ11-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
4	DIRQ12F	IRQ12-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
5	DIRQ13F	IRQ13-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
6	DIRQ14F	IRQ14-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
7	DIRQ15F	IRQ15-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1

Note: S-TYPE-4, P-TYPE-2
Note 1. Only 0 can be written to clear flag. The flag must be cleared by writing 0 after 1 is read.

Each flag is set to 1 when a cancel request specified by DPSIEGR1 is generated.

Each flag may be set to 1 when a cancel request is generated in any mode (not even Deep Software Standby mode) or when the setting of DPSIER1 is modified. Therefore, a transition to Deep Software Standby mode should be made after DPSIFR1 is cleared to 0x00.

To clear DPSIFR1 to 0x00 after modifying DPSIER1, wait for at least six PCLKB cycles, read DPSIFR1, and then write 0 to DPSIFR1. Six or more PCLKB cycles can be secured, for example, by reading DPSIER1.

DPSIFR1 is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode. For details, see section 6, Resets.

DIRQnF flag (IRQn-DS Pin Deep Software Standby Cancel Flag) (n = 8 to 15)

The DIRQnF flag indicates that a cancel request by the IRQn-DS pin has been generated.

[Setting condition]

A cancel request by the IRQn-DS pin specified by DPSIEGR1 is generated.

[Clearing condition]

Writing 0 to each flag after 1 is read.

11.2.29 DPSIFR2 : Deep Software Standby Interrupt Flag Register 2

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA20

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DNMIF	DRTCAIF	DRTCIIF	DPVD2IF	DPVD1IF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DPVD1IF	PVD1 Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W[*1]
1	DPVD2IF	PVD2 Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W[*1]
2	DRTCIIF	RTC Interval Interrupt Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W[*1]
3	DRTCAIF	RTC Alarm Interrupt Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W[*1]
4	DNMIF	NMI Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W[*1]
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-4, P-TYPE-2

Note 1. Only 0 can be written to clear flag. The flag must be cleared by writing 0 after 1 is read.

Each flag is set to 1 when a cancel request specified by DPSIEGR2 is generated.

Each flag may be set to 1 when a cancel request is generated in any mode (not even Deep Software Standby mode) or when the setting of DPSIER2 is modified. Therefore, a transition to Deep Software Standby mode should be made after DPSIFR2 is cleared to 0x00.

To clear DPSIFR2 to 0x00 after modifying DPSIER2, wait for at least six PCLKB cycles, read DPSIFR2, and then write 0 to DPSIFR2. Six or more PCLKB cycles can be secured, for example, by reading DPSIER2.

DPSIFR2 is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode. For details, see section 6, Resets.

DPVDmIF flag (PVDm Deep Software Standby Cancel Flag) (m = 1 to 2)

The DPVDmIF flag indicates that a cancel request by the voltage monitor m signal has been generated.

[Setting condition]

A cancel request is generated by the voltage monitor m signal that is selected in DPSIEGR2.

[Clearing condition]

Writing 0 to each flag after 1 is read.

DRTCIIF flag (RTC Interval Interrupt Deep Software Standby Cancel Flag)

This flag indicates that a cancel request by the RTC interval interrupt signal has been generated.

[Setting condition]

- A cancel request by the RTC interval interrupt signal is generated.

[Clearing condition]

- Writing 0 to each bit after reading the bit as 1.

DRTCAIF flag (RTC Alarm Interrupt Deep Software Standby Cancel Flag)

This flag indicates that a cancel request by the RTC alarm interrupt signal has been generated.

[Setting condition]

- A cancel request by the RTC alarm interrupt signal is generated.

[Clearing condition]

- Writing 0 to each bit after reading the bit as 1.

DNMIF flag (NMI Pin Deep Software Standby Cancel Flag)

This flag indicates that a cancel request by the NMI pin has been generated.

[Setting condition]

- A cancel request by the NMI pin specified by DPSIEGR2 is generated.

[Clearing condition]

- Writing 0 to each bit after reading the bit as 1.

11.2.30 DPSIFR3 : Deep Software Standby Interrupt Flag Register 3

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA24

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DVBATTADIF	DSOSTDIF	DIWDTIF	—	DULPT1IF	DULPT0IF	DUSBHSIF	DUSBFSIF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DUSBFSIF	USBFS Suspend/Resume Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
1	DUSBHSIF	USBHS Suspend/Resume Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
2	DULPT0IF	ULPT0 Overflow Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
3	DULPT1IF	ULPT1 Overflow Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
4	—	This bit is read as 0. The write value should be 0.	R/W
5	DIWDTIF	IWDT Overflow Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1

Bit	Symbol	Function	R/W
6	DSOSTDIF	Sub-clock Oscillation stop detection Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
7	DVBATTADIF	VBATT Tamper Detection Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1

Note: S-TYPE-4, P-TYPE-2
Note 1. Only 0 can be written to clear flag. The flag must be cleared by writing 0 after 1 is read.

Each flag is set to 1 when the corresponding cancel request is generated.

Each flag may be set to 1 when a cancel request is generated in any mode (not even Deep Software Standby mode) or when the setting of DPSIER3 is modified. Therefore, a transition to Deep Software Standby mode should be made after DPSIFR3 is cleared to 0x00.

To clear DPSIFR3 to 0x00 after modifying DPSIER3, wait for at least six PCLKB cycles, read DPSIFR3, and then write 0 to DPSIFR3. Six or more PCLKB cycles can be secured, for example, by reading DPSIER3.

DPSIFR3 is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode. For details, see section 6, Resets.

DUSBFSIF flag (USBFS Suspend/Resume Deep Software Standby Cancel Flag)

This flag indicates that a cancel request by the USBFS suspend/resume has been generated.

The DUSBFSIF flag is a flag for USBFS.

[Setting condition]

- A cancel request by the USBFS suspend/resume is generated.

[Clearing condition]

- Writing 0 to each bit after reading the bit as 1.

DUSBHSIF flag (USBHS Suspend/Resume Deep Software Standby Cancel Flag)

This flag indicates that a cancel request by the USBHS suspend/resume has been generated.

The DUSBHSIF flag is a flag for USBHS.

[Setting condition]

- A cancel request by the USBHS suspend/resume is generated.

[Clearing condition]

- Writing 0 to each bit after reading the bit as 1.

DULPT0IF flag (ULPT0 Overflow Deep Software Standby Cancel Flag)

This flag indicates that a cancel request by the ULPT0 overflow has been generated.

[Setting condition]

- A cancel request by the ULPT0 is generated.

[Clearing condition]

- Writing 0 to each bit after reading the bit as 1.

DULPT1IF flag (ULPT1 Overflow Deep Software Standby Cancel Flag)

This flag indicates that a cancel request by the ULPT1 overflow has been generated.

[Setting condition]

- A cancel request by the ULPT1 is generated.

[Clearing condition]

- Writing 0 to each bit after reading the bit as 1.

DIWDTIF flag (IWDT Overflow Deep Software Standby Cancel Flag)

This flag indicates that a cancel request by the IWDT overflow has been generated.

[Setting condition]

- A cancel request by the IWDT is generated.

[Clearing condition]

- Writing 0 to each bit after reading the bit as 1.

DSOSTDIF flag (Sub-clock Oscillation stop detection Deep Software Standby Cancel Flag)

This flag indicates that a cancel request by the Sub-clock Oscillation stop detection has been generated.

[Setting condition]

- A cancel request by the SOSTD is generated.

[Clearing condition]

- Writing 0 to each bit after reading the bit as 1.

DVBATTADIF flag (VBATT Tamper Detection Deep Software Standby Cancel Flag)

This flag indicates that a cancel request by the VBATT tamper detection has been generated.

[Setting condition]

- A cancel request by the VBATT tamper detection is generated.

[Clearing condition]

- Writing 0 to each bit after reading the bit as 1.

11.2.31 DPSIFR4 : Deep Software Standby Interrupt Flag Register 4

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xB48

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ23F	DIRQ22F	DIRQ21F	DIRQ20F	DIRQ19F	DIRQ18F	DIRQ17F	DIRQ16F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ16F	IRQ16-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
1	DIRQ17F	IRQ17-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
2	DIRQ18F	IRQ18-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
3	DIRQ19F	IRQ19-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
4	DIRQ20F	IRQ20-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1

Bit	Symbol	Function	R/W
5	DIRQ21F	IRQ21-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W
6	DIRQ22F	IRQ22-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
7	DIRQ23F	IRQ23-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1

Note: S-TYPE-4, P-TYPE-2
Note 1. Only 0 can be written to clear flag. The flag must be cleared by writing 0 after 1 is read.

Each flag is set to 1 when a cancel request specified by DPSIEGR3 is generated.

Each flag may be set to 1 when a cancel request is generated in any mode (not even Deep Software Standby mode) or when the setting of DPSIER4 is modified. Therefore, a transition to Deep Software Standby mode should be made after DPSIFR4 is cleared to 0x00.

To clear DPSIFR4 to 0x00 after modifying DPSIER4, wait for at least six PCLKB cycles, read DPSIFR4, and then write 0 to DPSIFR4. Six or more PCLKB cycles can be secured, for example, by reading DPSIER4.

DPSIFR4 is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode. For details, see section 6, Resets.

DIRQnF flag (IRQn-DS Pin Deep Software Standby Cancel Flag) (n = 16 to 23)

The DIRQnF flag indicates that a cancel request by the IRQn-DS pin has been generated.

[Setting condition]

A cancel request by the IRQn-DS pin specified by DPSIEGR3 is generated.

[Clearing condition]

Writing 0 to each flag after 1 is read.

11.2.32 DPSIFR5 : Deep Software Standby Interrupt Flag Register 5

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xB4C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ31F	DIRQ30F	DIRQ29F	DIRQ28F	DIRQ27F	DIRQ26F	DIRQ25F	DIRQ24F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ24F	IRQ24-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
1	DIRQ25F	IRQ25-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
2	DIRQ26F	IRQ26-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
3	DIRQ27F	IRQ27-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1

Bit	Symbol	Function	R/W
4	DIRQ28F	IRQ28-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
5	DIRQ29F	IRQ29-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
6	DIRQ30F	IRQ30-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1
7	DIRQ31F	IRQ31-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W*1

Note: S-TYPE-4, P-TYPE-2

Note 1. Only 0 can be written to clear flag. The flag must be cleared by writing 0 after 1 is read.

Each flag is set to 1 when a cancel request specified by DPSIEGR4 is generated.

Each flag may be set to 1 when a cancel request is generated in any mode (not even Deep Software Standby mode) or when the setting of DPSIER5 is modified. Therefore, a transition to Deep Software Standby mode should be made after DPSIFR5 is cleared to 0x00.

To clear DPSIFR5 to 0x00 after modifying DPSIER5, wait for at least six PCLKB cycles, read DPSIFR5, and then write 0 to DPSIFR5. Six or more PCLKB cycles can be secured, for example, by reading DPSIER5.

DPSIFR5 is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode. For details, see section 6, Resets.

DIRQnF flag (IRQn-DS Pin Deep Software Standby Cancel Flag) (n = 24 to 31)

The DIRQnF flag indicates that a cancel request by the IRQn-DS pin has been generated.

[Setting condition]

A cancel request by the IRQn-DS pin specified by DPSIEGR4 is generated.

[Clearing condition]

Writing 0 to each flag after 1 is read.

11.2.33 DPSIEGR0 : Deep Software Standby Interrupt Edge Register 0

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA28

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ7 EG	DIRQ6 EG	DIRQ5 EG	DIRQ4 EG	DIRQ3 EG	DIRQ2 EG	DIRQ1 EG	DIRQ0 EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ0EG	IRQ0-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
1	DIRQ1EG	IRQ1-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
2	DIRQ2EG	IRQ2-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W

Bit	Symbol	Function	R/W
3	DIRQ3EG	IRQ3-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
4	DIRQ4EG	IRQ4-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
5	DIRQ5EG	IRQ5-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
6	DIRQ6EG	IRQ6-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
7	DIRQ7EG	IRQ7-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W

Note: S-TYPE-4, P-TYPE-2

DPSIEGR0 is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode. For details, see section 6, Resets.

11.2.34 DPSIEGR1 : Deep Software Standby Interrupt Edge Register 1

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA2C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ15EG	DIRQ14EG	DIRQ13EG	DIRQ12EG	DIRQ11EG	DIRQ10EG	DIRQ9EG	DIRQ8EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ8EG	IRQ8-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
1	DIRQ9EG	IRQ9-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
2	DIRQ10EG	IRQ10-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
3	DIRQ11EG	IRQ11-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
4	DIRQ12EG	IRQ12-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
5	DIRQ13EG	IRQ13-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
6	DIRQ14EG	IRQ14-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
7	DIRQ15EG	IRQ15-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W

Note: S-TYPE-4, P-TYPE-2

DPSIEGR1 is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode. For details, see section 6, Resets.

11.2.35 DPSIEGR2 : Deep Software Standby Interrupt Edge Register 2

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA30

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DNMIEG	—	—	DPVD2EG	DPVD1EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DPVD1EG	PVD1 Edge Select 0: A cancel request is generated when VCC < Vdet1 (fall) is detected 1: A cancel request is generated when VCC ≥ Vdet1 (rise) is detected	R/W
1	DPVD2EG	PVD2 Edge Select 0: A cancel request is generated when VCC < Vdet2 (fall) is detected 1: A cancel request is generated when VCC ≥ Vdet2 (rise) is detected	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	DNMIEG	NMI Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-4, P-TYPE-2

DPSIEGR2 is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode. For details, see section 6, Resets.

11.2.36 DPSIEGR3 : Deep Software Standby Interrupt Edge Register 3

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA34

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ23EG	DIRQ22EG	DIRQ21EG	DIRQ20EG	DIRQ19EG	DIRQ18EG	DIRQ17EG	DIRQ16EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ16EG	IRQ16-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
1	DIRQ17EG	IRQ17-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
2	DIRQ18EG	IRQ18-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
3	DIRQ19EG	IRQ19-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W

Bit	Symbol	Function	R/W
4	DIRQ20EG	IRQ20-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
5	DIRQ21EG	IRQ21-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
6	DIRQ22EG	IRQ22-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
7	DIRQ23EG	IRQ23-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W

Note: S-TYPE-4, P-TYPE-2

DPSIEGR3 is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode. For details, see section 6, Resets.

11.2.37 DPSIEGR4 : Deep Software Standby Interrupt Edge Register 4

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xB50

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ31EG	DIRQ30EG	DIRQ29EG	DIRQ28EG	DIRQ27EG	DIRQ26EG	DIRQ25EG	DIRQ24EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ24EG	IRQ24-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
1	DIRQ25EG	IRQ25-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
2	DIRQ26EG	IRQ26-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
3	DIRQ27EG	IRQ27-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
4	DIRQ28EG	IRQ28-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
5	DIRQ29EG	IRQ29-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
6	DIRQ30EG	IRQ30-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
7	DIRQ31EG	IRQ31-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W

Note: S-TYPE-4, P-TYPE-2

DPSIEGR4 is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode. For details, see section 6, Resets.

11.2.38 SYOCDCR : System Control OCD Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA38

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

Bit	Symbol	Function	R/W
0	—	The read value is undefined. The write value should be 0.	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	DBGEN	Debugger Enable bit 0: Disable on-chip debugger function 1: Enable on-chip debugger function*1	R/W

Note: S-TYPE-5 P-TYPE-2

Note 1. The value of DBGEN can be set by software. However, it is not until the debugger authentication passes that the on-chip debugger function is valid. That is, even if the DBGEN is 1, the on-chip debugger function is invalid unless the debugger authentication passes. For detail, see section 2, CPU.

This register is not controlled by any security attribute register (e.g. LPMSAR, DPFSAR).

SYOCDCR is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode.

DBGEN bit (Debugger Enable bit)

The DBGEN enables the on-chip debugger function.

[Setting condition]

- Power-on reset is generated
- Voltage Monitor 0 reset is generated.
- Core Voltage Monitor reset is generated.
- Writing 1 to the bit.

[Clearing condition]

- Writing 0 to the bit

11.2.39 PLL1LDOCR : PLL1-LDO Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xB04

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

Bit	Symbol	Function	R/W
0	LDOSTP	LDO Stop 0: PLL1-LDO is enabled 1: PLL1-LDO is stopped	R/W

Bit	Symbol	Function	R/W
1	SKEEP	STBY Keep 0: PLL1-LDO is stopped during Software Standby mode. 1: PLL1-LDO state before Software Standby mode is retained during Software Standby mode.	R/W
2	—	The read value is undefined.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

You can control the operation of the LDO dedicated for PLL1 (PLL1-LDO) by using this register.

LDOSTP bit (LDO Stop)

The PLL1-LDO state during Normal mode is controlled by this bit.

After setting 1 to this bit, do not set 0 to this bit for at least 1 µs. This bit can be set 0 to this bit only when 1 µs has elapsed after setting 1 to this bit.

When this bit is changed from 1 to 0, it takes 25 µs for the power supply to stabilize.

Allow the power supply to stabilize before operating the PLL1.

SKEEP bit (STBY Keep)

The PLL1-LDO state during Software Standby mode is controlled by this bit. If the PLL1-LDO operation is stopped during Software Standby mode, the power consumption during Software Standby mode can be reduced.

11.2.40 PLL2LDOCR : PLL2-LDO Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xB08

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

Bit	Symbol	Function	R/W
0	LDOSTP	LDO Stop 0: PLL2-LDO is enabled 1: PLL2-LDO is stopped	R/W
1	SKEEP	STBY Keep 0: PLL2-LDO is stopped during Software Standby mode. 1: PLL2-LDO state before Software Standby mode is retained during Software Standby mode.	R/W
2	—	The read value is undefined.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

You can control the operation of the LDO dedicated for PLL2 (PLL2-LDO) by using this register.

LDOSTP bit (LDO Stop)

The PLL2-LDO state during Normal mode is controlled by this bit.

After setting 1 to this bit, do not set 0 to this bit for at least 1 µs. This bit can be set 0 to this bit only when 1 µs has elapsed after setting 1 to this bit.

When this bit is changed from 1 to 0, it takes 25 µs for the power supply to stabilize.

Allow the power supply to stabilize before operating the PLL2.

SKEEP bit (STBY Keep)

The PLL2-LDO state during Software Standby mode is controlled by this bit. If the PLL2-LDO operation is stopped during Software Standby mode, the power consumption during Software Standby mode can be reduced.

11.2.41 HOCOLDOCR : HOCO-LDO Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xB0C

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

Bit	Symbol	Function	R/W
0	LDOSTP	LDO Stop 0: HOCO-LDO is enabled 1: HOCO-LDO is stopped	R/W
1	SKEEP	STBY Keep 0: HOCO-LDO is stopped during Software Standby mode. 1: HOCO-LDO state before Software Standby mode is retained during Software Standby mode.	R/W
2	—	The read value is undefined.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

You can control the operation of the LDO dedicated for HOCO (HOCO-LDO) by using this register.

LDOSTP bit (LDO Stop)

The HOCO-LDO state during Normal mode is controlled by this bit.

After setting 1 to this bit, do not set 0 to this bit for at least 2 µs. This bit can be set 0 to this bit only when 2 µs has elapsed after setting 1 to this bit.

When this bit is changed from 1 to 0, it takes 5 µs for the power supply to stabilize.

Allow the power supply to stabilize before operating the HOCO.

SKEEP bit (STBY Keep)

The HOCO-LDO state during Software Standby mode is controlled by this bit. If the HOCO-LDO operation is stopped during Software Standby mode, the power consumption during Software Standby mode can be reduced.

11.2.42 VSCR : Voltage Scaling Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x014

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	VSCMTSF	—	VSCM[2:0]		
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
2:0	VSCM[2:0]	Voltage Scaling Control Mode Bit Set the target voltage of VDD. 0 0 1: VSCR_1 0 1 0: VSCR_2 (default) Setting other than above are prohibited.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	VSCMTSF	Voltage Scaling Control Mode Transition Status Flag ● Read 0: Transition completed 1: During transition ● Write The write value should be 0.	R
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

The VSCR register controls the voltage scaling operation.

VSCM[2:0] bits (Voltage Scaling Control Mode Bit)

The VSCM bits specify the target voltage of VDD for voltage scaling operation.

VSCMTSF flag (Voltage Scaling Control Mode Transition Status Flag)

The VSCMTSF flag indicates the voltage scaling transition state when the operating voltage scaling control is executed. When a write access is attempted to change the voltage scaling control mode, the VSCMTSF flag is set to 1. The flag becomes 0 after a transition to the voltage scaling control mode is completed. Make sure that the VSCMTSF flag is 0 (completed operating voltage scaling control mode transition) before the next processing.

11.2.43 SVSCR : SSTBY Voltage Scaling Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA9C

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

Bit	Symbol	Function	R/W
2:0	SVSCM[2:0]	SSTBY Voltage Scaling Control Mode Bit Set the target voltage of VDD in Software Standby mode. 0 0 1: SVSCR_1 0 1 0: SVSCR_2 (default) 0 1 1: SVSCR_3 1 0 0: SVSCR_4 1 0 1: SVSCR_5 Setting other than above are prohibited.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

The SVSCR register controls the voltage scaling operation in Software Standby mode.

SVSCM[2:0] bits (SSTBY Voltage Scaling Control Mode Bit)

The SVSCM bits specify the target voltage of VDD for voltage scaling operation in Software Standby mode.

It is forbidden to set SVSCR_1 and SS2LP[1:0] = 01b and transition to Software Standby mode.

When SVSCR_3, SVSCR_4 or SVSCR_5 is set, the interrupt factor that cancels Software Standby mode is limited to NMI pin or IRQ pin.

11.2.44 CPUDSCR : CPU Deep Sleep Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x100

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PGD1	PGD0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PGD0	Power Gating Disable for CPU0 0: Power Gating Enable during CPU0 Deep Sleep (default) 1: Power Gating Disable during CPU0 Deep Sleep	R/W
1	PGD1	Power Gating Disable for CPU1 0: Power Gating Enable during CPU1 Deep Sleep (default) 1: Power Gating Disable during CPU1 Deep Sleep	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

The CPUDSCR register controls the power gating during CPU deep sleep mode.

Each bit of CPUDSCR should be changed while the target CPU is operation.

When transitioning to Software Standby mode and Deep Software Standby mode, CPUDSCR.PGDn must be set to 0.

PGD0 bit (Power Gating Disable for CPU0)

The PGD0 bit specifies the power gating control of CPU0 power domain during CPU0 deep sleep mode.

PGD1 bit (Power Gating Disable for CPU1)

The PGD1 bit specifies the power gating control of CPU1 power domain during CPU1 deep sleep mode.

11.3 Reducing Power Consumption by Switching Clock Signals

The clock frequency changes when the SCKDIVCR and SCKDIVCR2 registers are set.

For information on module and clock associations, see section 9.2.2. SCKDIVCR : System Clock Division Control Register and section 9.2.3. SCKDIVCR2 : System Clock Division Control Register 2.

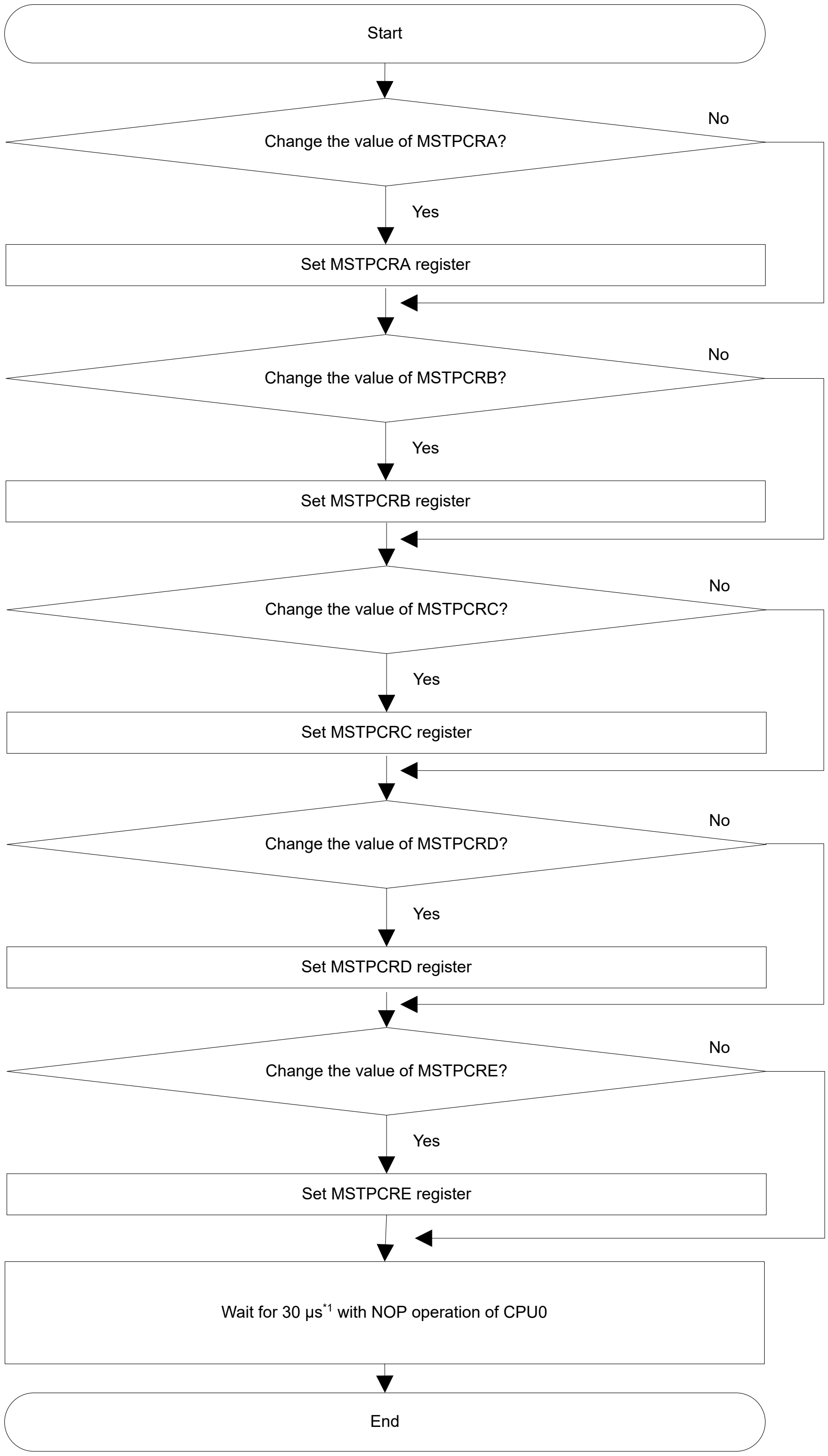
11.4 Module-Stop Function

The module-stop function can be set for each on-chip peripheral module.

When the MSTPmi bit (m = A to E, i = 31 to 0) in MSTPCRA to MSTPCRE is set to 1, the corresponding module stops operating and enters the module-stop state, but the CPU continues to operate independently. Clearing the MSTPmi bit to 0 cancels the module-stop state, allowing the module to restart operating at the end of the bus cycle. The internal states of modules are retained in the module-stop state.

After a reset is canceled, all modules other than the DMAC, DTC and SRAMn (n = 0 to 3) are placed in the module-stop state. Do not access the module while the corresponding MSTPmi bit is 1. Otherwise, the read/write data nor the operation of the module is not guaranteed. Also, do not set 1 to the MSTPmi bit while the corresponding module is being accessed.

If the CPUCLK0 is faster than the ICLK maximum frequency in the electrical characteristics, after the MSTPmi bit is changed, a wait time should be provided according to the flow shown in Figure 11.2. It is recommended to use software to measure the wait time. Be sure to consider the worst-case conditions to ensure that the required wait time elapses. If an interrupt is unavoidably generated during the wait time, retry the measurement after return from the interrupt.



Note 1. 30 µs in DCDC mode, 10 µs in External VDD mode

Figure 11.2 Setting flow of module stop control registers in case of CPUCLK0 is higher than the ICLK maximum frequency in electrical characteristics

11.5 Power Gating Control

The MCU supports the power gating function of the power domain.

User can individually control the power on/off of the divided power domains.

In the power gating state, the domain is powered off and the state is the same as SSTBY in Table 11.3.

11.5.1 Power Gating Control Using Power Domain Control Register

Table 11.7 shows the relationship PDCTR register and the target power domain.

When the PDCTR register is set, power gating control is executed. In addition, when using the power gating function, it should be set MOCOCR.MCSTP to 0 (MOCO is operated) in advance.

Make sure that the power domain control of the CPU and the power domain control by the PDCTRESWM do not conflict.

Table 11.7 Relationship between the PDCTR register and the target power domain

Control register				Target Power Domain name	Implemented IP name
Register name	PDDE	PDCSF	PDPGSF		
PDCTRESWM	Available	Available	Available	ESWM domain	ESWM

Examples of the procedures for power gating control:

1. Power OFF by PDCTRn.PDDE*1
   STEP1 : Confirm that the target domain is not performing the Power gating process
   - Check PDCTRn.PDCSF = 0

   STEP2 : Set the module stop control register to stop.
   - Write MSTPCR* = 1 (*: all register for target domain)

   STEP3 : Set PDCTRn.PDDE bit and start the power control.
   - Write PDCTRn.PDDE = 1

   STEP4 : Confirm that the power gating process is completed and the powered off
   - Check PDCTRn.PDCSF = 0 & PDCTRn.PDPGSF = 1
2. Power ON by PDCTRn.PDDE*1
   STEP1 : Confirm that the target domain is not performing the Power gating process
   - Check PDCTRn.PDCSF = 0

   STEP2 : Set PDCTRn.PDDE bit and start the power control.
   - Write PDCTRn.PDDE = 0

   STEP3 : Confirm that the power gating process is completed and the powered on
   - Check PDCTRn.PDCSF = 0 & PDCTRn.PDPGSF = 0

   STEP4 : Set the module stop control register to operate.
   - Write MSTPCR* = 0 (*: all register for target domain)

Note 1. When PDCTRn.PDCSF bit is “1” (during power gating control processing), PDCTRn register must not be changed.

11.6 Low Power Modes

11.6.1 CPU Sleep Mode and CPU Deep Sleep Mode

See section 2.7. Sleep Modes for Low Power States

11.6.2 Software Standby Mode

11.6.2.1 Entering Software Standby Mode

When WFI instructions of both CPUs are executed while LPSCR.LPMD bit is 0x5 and CPUn.SCR.SLEEPDEEP bit is 1, the MCU enters Software Standby mode depending on LPSCR.LPMD setting. When transitioning to Software Standby mode, CPUDSCR.PGDn must be set to 0. It must be set MOCOCR.MCSTP to 0 (MOCO is operated) before executed the WFI instruction to entry this mode. Take care that the transition is suspended during the configuration update transfer.

In Software Standby mode, the CPU, most of the on-chip peripheral functions and most of the oscillators stop*1. However, the contents of the CPU internal registers and RAM data, the states of on-chip peripheral functions and the I/O ports are retained. Software Standby mode allows significant reduction in power consumption because most of the oscillators stops*1. Table 11.3 shows the status of each on-chip peripheral functions and oscillators. In addition, Software Standby mode allows the internal regulator to transition into a low power state through SSCR1.SS2LP register. This further reduces power consumption. Available resets or interrupts in Software Standby mode make the MCU to return to Normal mode. See Table 11.4 for available interrupt sources for each mode. In case of using an interrupt to return to Normal mode, corresponding IELSRn register must be set before executing a WFI instruction. For the detail, see section 14, Interrupt Controller Unit (ICU).

Note 1. The state of the oscillators depends on the setting of the control register for each oscillator. For details, see section 9, Clock Generation Circuit.

The status of address bus and bus control signals in Software Standby mode can be selected by SBYCR.OPE bit. The SBYCR.OPE bit specifies whether to set to the high-impedance state or to retain the output of the address bus and bus control signals in Software Standby mode.

Clear DMAST.DMST bit and DTCST.DTCST bit to 0 before executing WFI instruction.

Counting by the IWDT stops when the MCU enters Software Standby mode while the IWDT is in auto-start mode and the OFS0.IWDTSTPCTL bit is 1 (IWDT stops in Sleep or Software Standby).

Counting by the IWDT continues when the MCU enters Software Standby mode while the IWDT is in auto-start mode and the OFS0.IWDTSTPCTL bit is 0 (IWDT does not stop in Sleep or Software Standby).

Counting by the IWDT stops when the MCU enters Software Standby mode while the IWDT is in register start mode and the SLCSTP bit in IWDTCSTPR is 1 (IWDT stops in Sleep or Software Standby).

Counting by the IWDT continues when the MCU enters Software Standby mode while the IWDT is in register start mode and the SLCSTP bit in IWDTCSTPR is 0 (IWDT does not stop in Sleep or Software Standby).

WDT stops counting when the MCU enters Software Standby mode because the PCLK stops.

Do not enter Software Standby mode while the MRAM is in programming procedure. In order to enter Software Standby mode, execute a WFI instruction after programming procedure completes.

In External VDD mode, do not enter Software Standby Mode.

Table 11.8 shows the setting of the related control bits and the modes to enter after executing WFI instruction.

Table 11.8 Setting of the bits what affect the mode after executing WFI instruction (1 of 2)

		Setting mode of LPSCR.LPMD bit and CPUn.SCR.SLEEPDEEP (n = 0, 1) bit			
		Software Standby (LPSCR.LPMD = 0x5, CPUn.SCR. SLEEPDEEP = 1)	Deep Software Standby 1 (LPSCR.LPMD = 0x8, CPUn.SCR. SLEEPDEEP = 1)	Deep Software Standby 2 (LPSCR.LPMD = 0x9, CPUn.SCR. SLEEPDEEP = 1)	Deep Software Standby 3 (LPSCR.LPMD = 0xA, CPUn.SCR. SLEEPDEEP = 1)
MENTRYR.MENTRY	0	Software Standby	Deep Software Standby 1	Deep Software Standby 2	Deep Software Standby 3
	1	CPU Deep sleep mode	CPU Deep sleep mode	CPU Deep sleep mode	CPU Deep sleep mode
CPUDSCR.PGD0	0	Software Standby	Deep Software Standby 1	Deep Software Standby 2	Deep Software Standby 3
	1	CPU Deep sleep mode	CPU Deep sleep mode	CPU Deep sleep mode	CPU Deep sleep mode

Table 11.8 Setting of the bits what affect the mode after executing WFI instruction (2 of 2)

		Setting mode of LPSCR.LPMD bit and CPUn.SCR.SLEEPDEEP (n = 0, 1) bit			
		Software Standby (LPSCR.LPMD = 0x5, CPUn.SCR. SLEEPDEEP = 1)	Deep Software Standby 1 (LPSCR.LPMD = 0x8, CPUn.SCR. SLEEPDEEP = 1)	Deep Software Standby 2 (LPSCR.LPMD = 0x9, CPUn.SCR. SLEEPDEEP = 1)	Deep Software Standby 3 (LPSCR.LPMD = 0xA, CPUn.SCR. SLEEPDEEP = 1)
CPUDSCR.PGD1	0	Software Standby	Deep Software Standby 1	Deep Software Standby 2	Deep Software Standby 3
	1	CPU Deep sleep mode	CPU Deep sleep mode	CPU Deep sleep mode	CPU Deep sleep mode
OFS0.IWDTSTPCTL (auto-start mode)	0	Software Standby	Deep Software Standby 1	Deep Software Standby 1	Deep Software Standby 1
	1			Deep Software Standby 2	Deep Software Standby 3
IWDTCSTPR.SLCSTP (register start mode)	0	Software Standby	Deep Software Standby 1	Deep Software Standby 1	Deep Software Standby 1
	1			Deep Software Standby 2	Deep Software Standby 3
OFS2.CVMRDIS	0	Software Standby	Software Standby	Software Standby	Software Standby
	1		Deep Software Standby 1	Deep Software Standby 2	Deep Software Standby 3
PVD1CR0.RI	0	Software Standby	Deep Software Standby 1	Deep Software Standby 2	Deep Software Standby 3
	1			Deep Software Standby 1	Deep Software Standby 1
PVD2CR0.RI	0	Software Standby	Deep Software Standby 1	Deep Software Standby 2	Deep Software Standby 3
	1			Deep Software Standby 1	Deep Software Standby 1
PVD4CR0.RE	0	Software Standby	Deep Software Standby 1	Deep Software Standby 2	Deep Software Standby 3
	1				Deep Software Standby 2*1
PVD5CR0.RE	0	Software Standby	Deep Software Standby 1	Deep Software Standby 2	Deep Software Standby 3
	1				Deep Software Standby 2*1

Note 1. When at least PVDnCR0.RE (n = 4, 5) bit is 1 and all PVDnCR0.RI (n = 1, 2) bit are 0 and OFS0.IWDTSTPCTL is 1 and IWDTCSTPR.SLCSTP is 1, the transition is to Deep Software Standby mode 2 even if LPSCR is set to 0xA.

11.6.2.2 Canceling Software Standby Mode

Software Standby mode is canceled by an available interrupt shown in Table 11.4, RES pin reset, a power-on reset, a voltage monitor reset, a core voltage monitor reset, a reset caused by IWDT underflow or a temperature monitor reset.

The oscillators which were operating before the transition to Software Standby mode restarts. After the oscillation of all these oscillators has been stabilized, the MCU returns from Software Standby mode to Normal mode.

1. Canceling by an interrupt
   When an available interrupt request (for available interrupt, See Table 11.4) is generated, each oscillator which was operating before the transition to Software Standby mode restarts. After all these oscillators have been stabilized, the MCU returns from Software Standby mode to Normal mode and starts interrupt handling. When canceling the software standby mode by interrupt, each power domain returns to the same power state as the state before entering the software standby state.
2. Canceling by a RES pin reset

When RES pin is driven low, the MCU enters the reset state and oscillators, whose default status are operating, starts the oscillation. Make sure to keep RES pin low for the time period specified in section 62, Electrical Characteristics. When RES pin is driven high after the specified time period, the CPU starts the reset exception handling.

3. Canceling by a power-on reset
   Software Standby mode is canceled by a power-on reset and the MCU starts the reset exception handling.
4. Canceling by a voltage monitor reset
   Software Standby mode is canceled by a voltage monitor reset from the voltage detection circuit and the MCU starts the reset exception handling.
5. Canceling by IWDT reset
   Software Standby mode is canceled by an internal reset generated by IWDT underflow and the MCU starts the reset exception handling. However, IWDT stops in Software Standby mode and an internal reset for Canceling Software Standby mode is not generated in the following conditions:
   - OFS0.IWDTSTRT = 0 and OFS0.IWDTSTPCTL = 1 (auto-start mode)
   - IWDTCSTPR.SLCSTP = 1 (register start mode)
6. Canceling by a core voltage monitor reset
   Software Standby mode is canceled by a core voltage monitor reset (CVMR) and the MCU starts the reset exception handling.
7. Canceling by a temperature monitor reset
   Software Standby mode is canceled by a temperature monitor reset and the MCU starts the reset exception handling.

11.6.2.3 Example of Software Standby Mode Application

Figure 11.3 shows an example of entry to Software Standby mode by a falling edge of IRQn pin, and exit from Software Standby mode by a rising edge of IRQn pin.

In this example, an IRQn pin interrupt is accepted with IRQCRi.IRQMD[1:0] (i = 0 to 31) bits of the ICU are 00b (falling edge) in Normal mode, and then IRQCRi.IRQMD[1:0] (i = 0 to 31) bits are set to 01b (rising edge). Then, if MOCO is not operated, it must set MOCOCR.MCSTP to 0 (MOCO is operated). After that, LPSCR.LPMD bit is set to 0x5 and CPUn.SCR.SLEEPDEEP bit is set to 1, and then WFI instruction is executed. Thus, entry to Software Standby mode is completed. After that, exit from Software Standby mode is initiated by a rising edge of IRQn pin.

Setting the ICU is also required to exit Software Standby mode. For details, see section 14, Interrupt Controller Unit (ICU). The oscillation stabilization time in Figure 11.2 is specified in section 62, Electrical Characteristics.

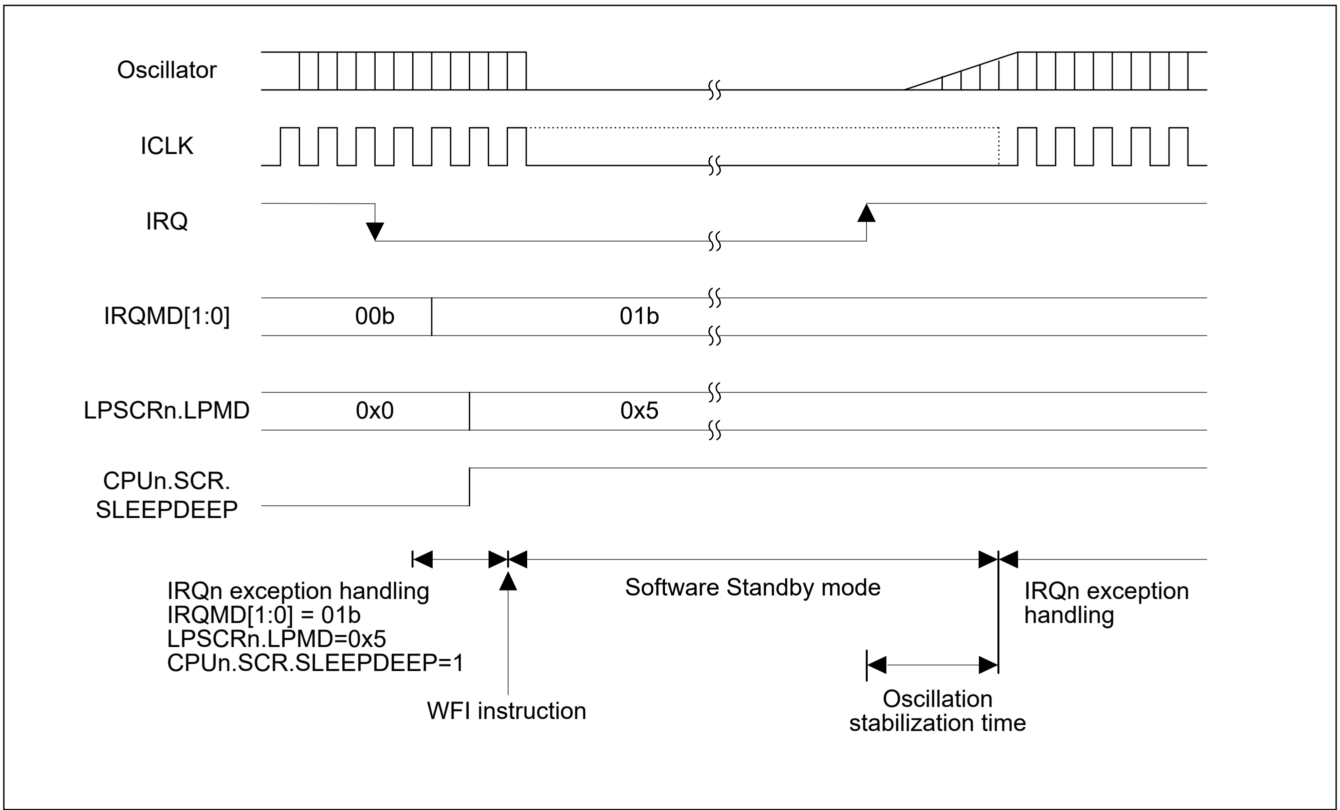


Figure 11.3 Example of Software Standby mode application

11.6.3 Deep Software Standby Mode

11.6.3.1 Transition to Deep Software Standby Mode

When WFI instructions of both CPUs are executed while LPSCR.LPMD bit is 0x8, 0x9, 0xA and CPUn.SCR.SLEEPDEEP bit is 1, the MCU enters Deep Software Standby mode 1 or 2 or 3 depending on LPSCR.LPMD setting. When transitioning to Deep Software Standby mode, CPUDSCR.PGDn must be set to 0. It should be set MOCOCR.MCSTP to 0 (MOCO is operated) before executed the WFI instruction to entry this mode. See Table 11.8 for the setting of the related control bits.

In these modes, the CPU, on-chip peripheral functions, SRAM, and oscillators are almost stopped as shown in Table 11.3. Furthermore, since the internal power supply to these inactive modules is stopped, power consumption is remarkably reduced. The contents of all the registers of the CPU and specific internal peripheral modules become undefined. Table 11.3 shows the status of each on-chip peripheral functions and oscillators.

In Deep Software Standby mode 1, when OFS1.PVDAS = 0 and OFS1.PVDLPSEL = 0, the battery power supply switch is controlled by the voltage monitor 0 which has lower power consumption than VCC voltage drop detection circuit dedicated for VBATT, so power consumption is further reduced at the case that VBATT function is enabled. In Deep Software Standby mode 2, the internal supply of power to the USB resume detecting unit and IWDT is cut off, reducing power consumption. When OFS1.PVDAS = 0 and OFS1.PVDLPSEL = 0, the battery power supply switch is controlled by the voltage monitor 0 which has lower power consumption than VCC voltage drop detection circuit dedicated for VBATT, so power consumption is further reduced at the case that VBATT function is enabled. In Deep Software Standby mode 3, the PVD is also stopped. For details, see section 62, Electrical Characteristics.

Clear DMAST.DMST bit and DTCST.DTCST bit to 0 before executing WFI instruction.

When the MCU enters Deep Software Standby mode 2 or 3 while the IWDT is in auto-start mode and the OFS0.IWDTSTPCTL bit is 1, power supply to the IWDT-dedicated clock and the IWDT is cut off. Counting by the IWDT also stops.

When the IWDT is in auto-start mode and the OFS0.IWDTSTPCTL bit is 0, the MCU enters Deep Software Standby mode 1 instead of Deep Software Standby Mode 2 or 3, regardless of the setting in the LPSCR.LPMD bit.

When the IWDT is in register start mode and the IWDTCSTPR.SLCSTP is 0, the MCU enters Deep Software Standby mode 1 instead of Deep Software Standby Mode 2 or 3, regardless of the setting in the LPSCR.LPMD.

Do not enter Deep Software Standby mode while the MRAM is in programming procedure. In order to enter Deep Software Standby mode, execute a WFI instruction after programming procedure completes.

In External VDD mode, do not enter Deep Software Standby Mode.

When PVDnCR0.RI (n = 1, 2) = 1 (selecting the voltage monitor n reset), the MCU enters Deep Software Standby mode 1 instead of Deep Software Standby mode 2 or 3, regardless of the setting in the LPSCR.LPMD. The I/O port states are same as Software Standby mode.

When at least PVDnCR0. RE (n = 4, 5) bit is 1 and all PVDnCR0. RI (n = 1, 2) bit are 0 and OFS0.IWDTSTPCTL is 1 and IWDTCSTPR.SLCSTP is 1, the MCU enters Deep Software Standby mode 2 instead of Deep Software Standby mode 3, regardless of the setting in the LPSCR.LPMD. The I/O port states are same as Software Standby mode.

When the Core Voltage Detection Circuit is enabled (Core Voltage Monitor disable bit (CVMRDIS) in Option Function Select Register 0 (OFS2) is 0), the MCU enters Software Standby mode instead of Deep Software Standby Mode 1, 2 or 3, regardless of the setting in the LPSCR.LPMD bit.

11.6.3.2 Canceling Deep Software Standby Mode

Deep Software Standby mode is canceled by the interrupt shown in Table 11.4, RES pin reset, a power-on reset, a voltage monitor reset, a core voltage monitor reset, a reset caused by IWDT underflow or a temperature monitor reset.

(1) Canceling by an interrupt

Canceling by interrupts is controlled by DPSIERn (n = 0 to 5) and DPSIFRn (n = 0 to 5). When a Deep Software Standby Canceling interrupt is generated, the corresponding flag in DPSIFRn is set to 1. If the interrupt is enabled in DPSIERn, Deep Software Standby mode is canceled. Rising edge or falling edge can be selected by DPSIEGRn (n = 0 to 4). The interrupts for which an edge can be selected are the NMI, IRQ0-DS to IRQ31-DS, voltage monitor n (n = 1, 2) interrupts. When a Deep Software Standby mode canceling request occurs, the internal power is supplied and MOCO starts oscillating, and then the internal reset (Deep Software Standby reset) is generated for the entire MCU.

The stable MOCO clock is then supplied to the entire MCU and Deep Software Standby reset is canceled, and the MCU starts reset exception handling.

When Deep Software Standby mode is canceled by an external interrupt pin or internal interrupt signal, the RSTSR0.DPSRSTF flag is set to 1.

(2) Canceling by RES pin reset

When the RES pin is driven low, the MCU cancels Deep Software Standby mode and enters the reset state. Make sure to keep RES pin low for the time period specified in section 62, Electrical Characteristics. When RES pin is driven high after the specified time period, the CPU starts the reset exception handling.

(3) Canceling by a power-on reset

Deep Software Standby mode is canceled by a power-on reset and the MCU starts the reset exception handling.

(4) Canceling by a voltage monitor 0 reset

Deep Software Standby mode is canceled by a voltage monitor 0 reset from the voltage detection circuit and the MCU starts the reset exception handling.

(5) Canceling by a voltage monitor n reset (n = 1, 2, 4, 5)

Deep Software Standby mode 1 is canceled by a voltage monitor n reset (n = 1, 2, 4, 5)from the voltage detection circuit and the MCU starts the reset exception handling.

(6) Canceling by a IWDT reset

Deep Software Standby mode 1 is canceled by a IWDT reset (an internal reset generated by IWDT underflow) and the MCU starts the reset exception handling.

However, IWDT stops in Deep Software Standby mode 1 and a IWDT reset for Canceling Deep Software Standby mode 1 is not generated in the following conditions:

- OFS0.IWDTSTRT = 0 and OFS0.IWDTSTPCTL = 1 (auto-start mode)
- IWDTCSTPR.SLCSTP = 1 (register start mode)

(7) Canceling by a temperature monitor reset

Deep Software Standby mode is canceled by a temperature monitor reset and the MCU starts the reset exception handling.

11.6.3.3 Pin States when Deep Software Standby Mode is Canceled

In Deep Software Standby mode, the I/O ports retain the same states as before transition to the Deep Software Standby mode. The MCU is initialized by an internal reset generated when Deep Software Standby mode is canceled. Upon cancellation of Deep Software Standby mode, the reset exception handling starts immediately.

Whether to initialize the I/O ports or to retain the I/O port states as before transition to the Deep Software Standby mode can be selected by the DPSBYCR.IOKEEP bit.

- When the DPSBYCR.IOKEEP bit = 0
  I/O ports are initialized by an internal reset generated when Deep Software Standby mode is canceled.
- When the DPSBYCR.IOKEEP bit = 1

Although the MCU is initialized by an internal reset generated when Deep Software Standby mode is canceled, I/O ports retain their states as before transition to the Deep Software Standby mode regardless of the MCU internal state. At this time, the I/O port states remain unchanged as before transition to the Deep Software Standby mode even if settings of I/O ports or peripheral modules are made. Then, the retained I/O port states are released by clearing the DPSBYCR.IOKEEP bit to 0, and the MCU operates according to the internal state.

The DPSBYCR.IOKEEP bit is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode.

11.6.3.4 Example of Deep Software Standby Mode Application

(1) Entering and exiting Deep Software Standby mode

Figure 11.4 shows an example where a transition to Deep Software Standby mode is made at the falling edge of the IRQn-DS pin, and exiting Deep Software Standby mode is made at the rising edge of the IRQn-DS pin. In this example, an IRQn interrupt is accepted with the IRQCRi.IRQMD[1:0] (i = 0 to 31) bits of the ICU set to 00b (falling edge). Then, after the DPSIEGRy.DIRQnEG (y = 0 or 1 or 4 or 5, n = 0 to 31) bit is set to 1 (rising edge) and the LPSCR.LPMD bit is set to 0x8 or 0x9 or 0xA and CPUn.SCR.SLEEPDEEP bit is set to 1, the WFI instruction is executed. Thus, a transition to Deep Software Standby mode is made. After that, exiting Deep Software Standby mode is made at the rising edge of the IRQ-DS pin.

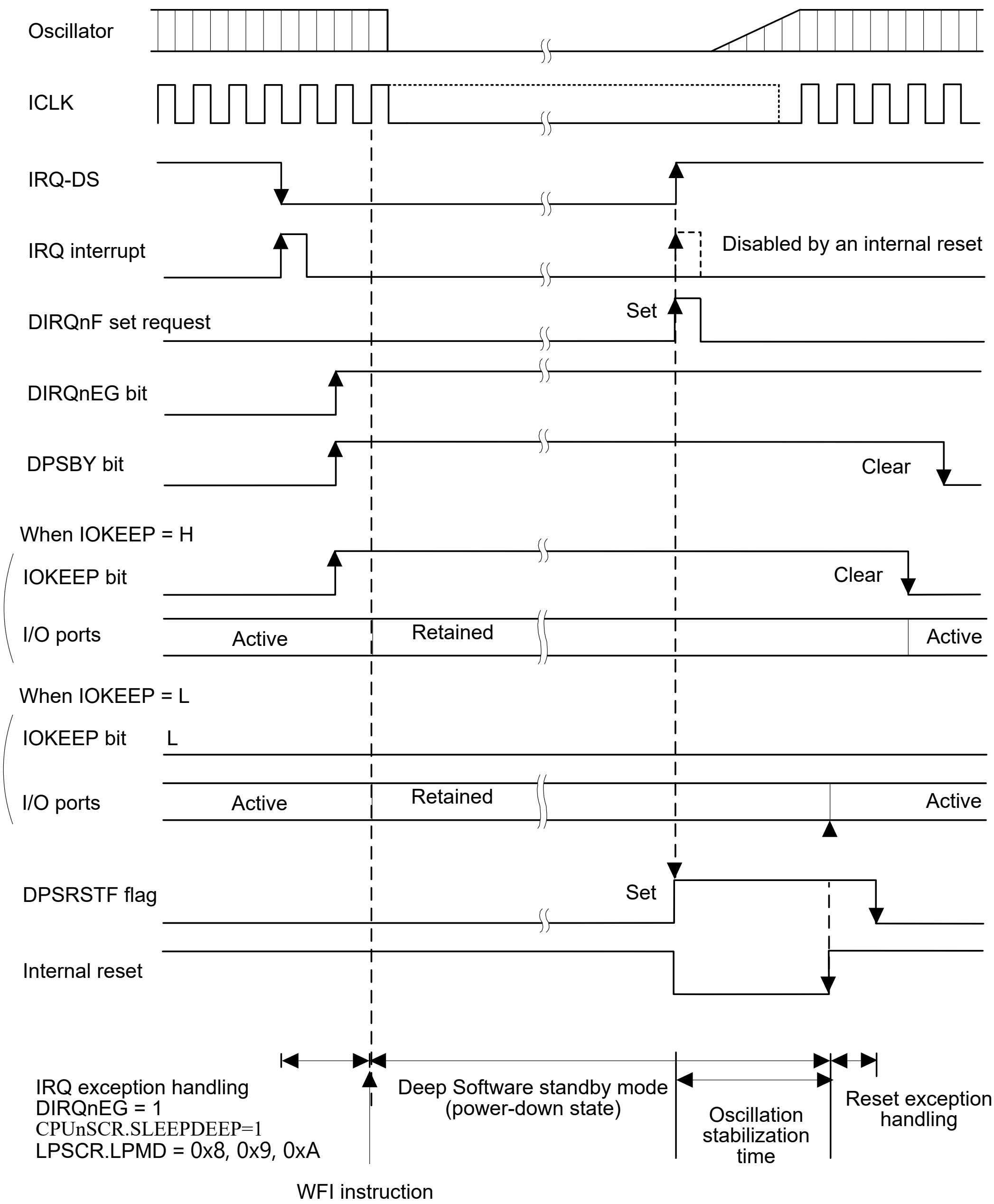


Figure 11.4 Example of Deep Software Standby mode application

11.6.3.5 Usage Flow for Deep Software Standby Mode

Figure 11.5 shows an example of a flowchart to use Deep Software Standby mode.

In this example, the RSTSR0.DPSRSTF flag of the reset function is read after the reset exception handling to determine whether a reset was generated by the RES pin or by the cancellation of Deep Software Standby mode.

In the case of a reset by the RES pin, a transition to Deep Software Standby mode is made after the required register settings have been made.

In the case of a reset by the cancellation of Deep Software Standby mode, the DPSBYCR.IOKEEP bit is cleared to 0 after the I/O port settings have been made.

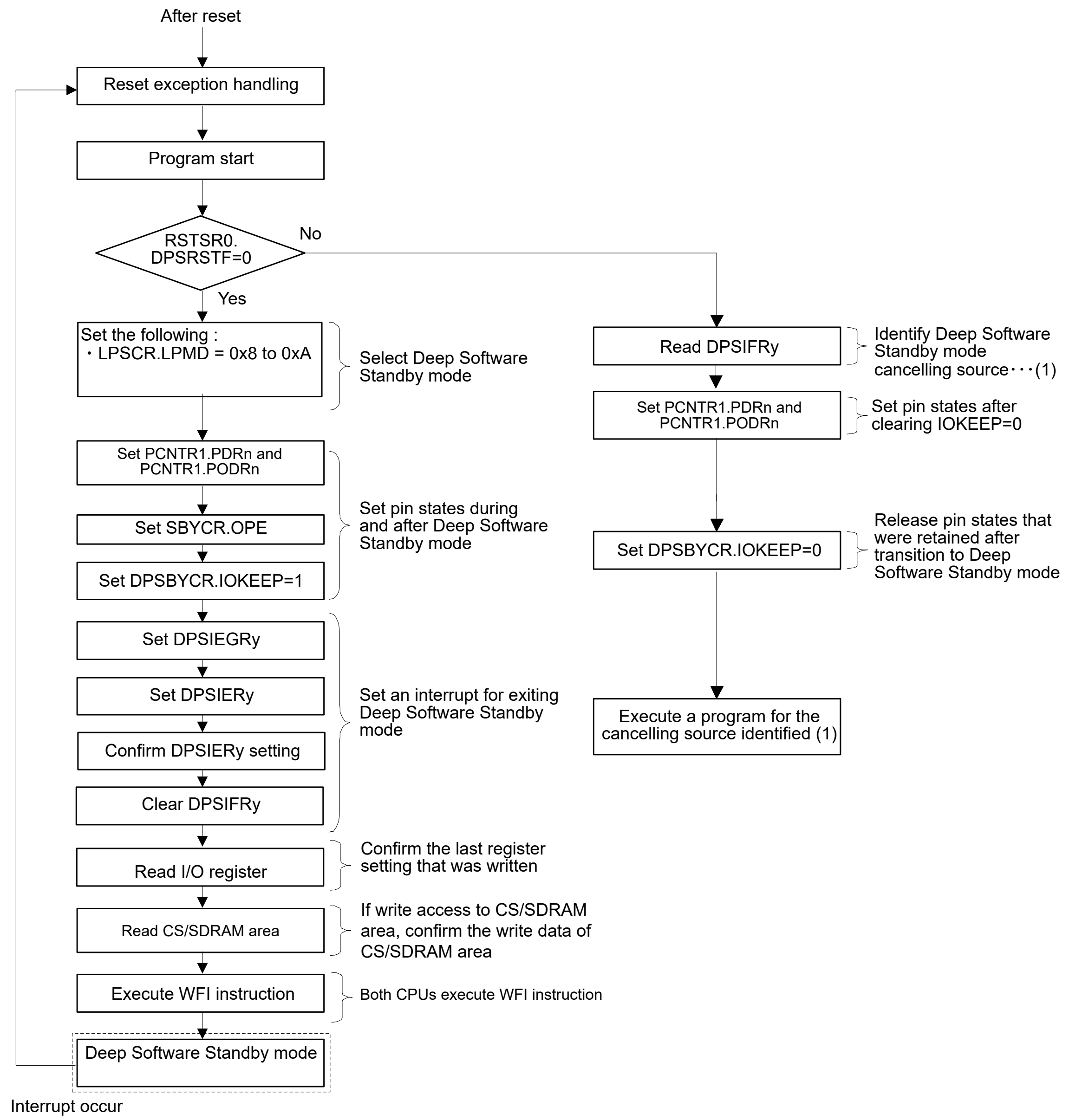


Figure 11.5 Example flowchart to use Deep Software Standby mode

11.7 Voltage Scaling Control

The MCU supports Voltage Scaling (DVFS) control for low power or high performance operation. CM85 TCM and CM85 cache cannot be used during the transition (VSCR.VSCMTSF = 1). Do not access CM85 TCM. Deactivate the CM85 cache or set the cache attribute of the entire address space to Non-cacheable. Figure 11.6 shows the transition of DVFS control.

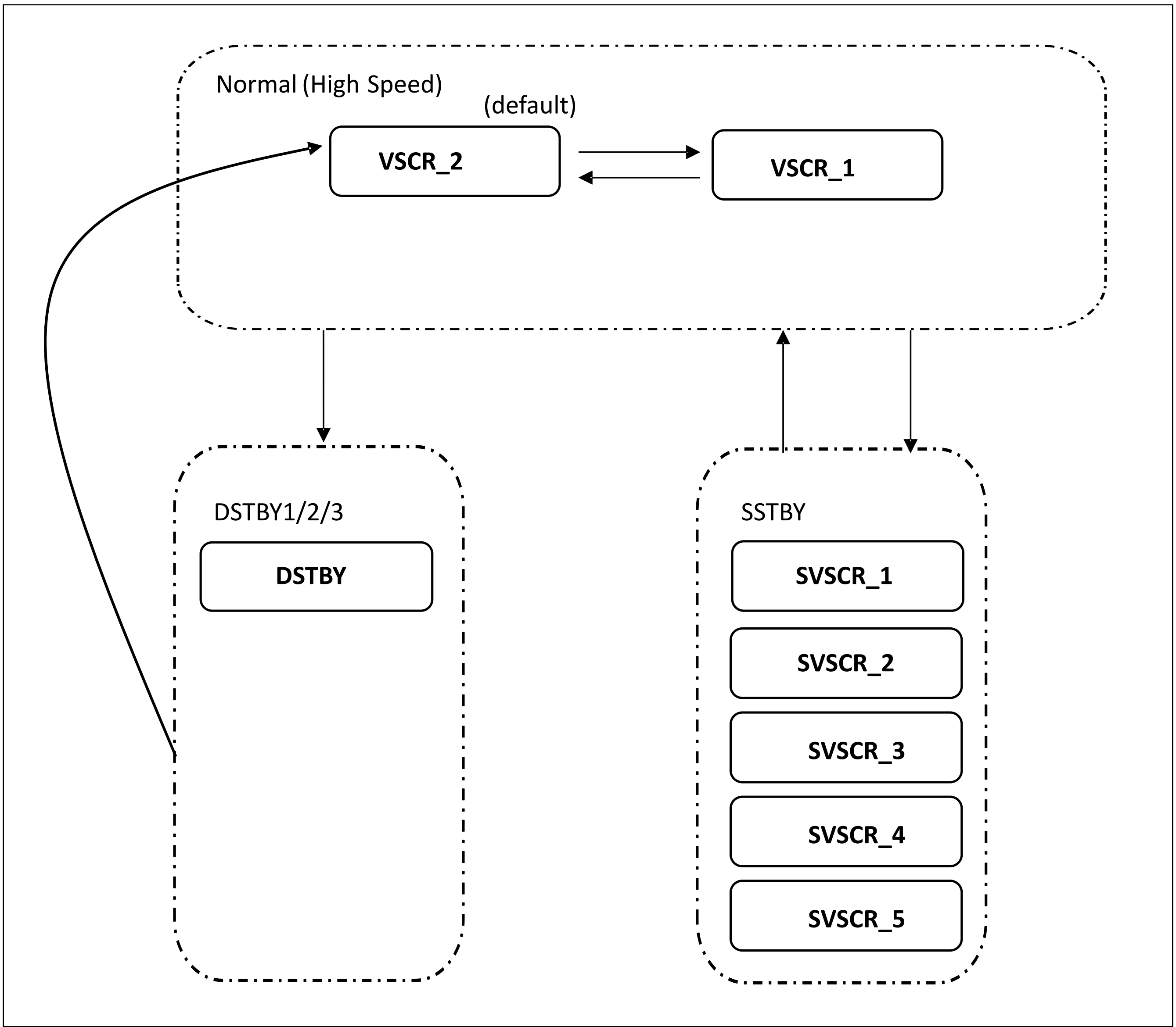


Figure 11.6 Voltage Scaling (DVFS) control for low power or high performance operation

<Normal Mode>

Make sure the operating condition (the frequency range) is always within the specified range before and after switching the VDD in Normal mode. Switch the voltage so that each clock does not exceed the maximum frequency shown in section 62, Electrical Characteristics.

The followings are examples of the procedures for switching the VDD

(1) Switching from VSCR_2 to VSCR_1

1. Set the VSCR.VSCM[2:0] bit to 0x1.
2. Confirm that VSCR.VSCMTSF flag is 0.
3. Set the SCKDIVCR and SCKDIVCR2 to the desired value.

(2) Switching from VSCR_1 to VSCR_2

1. Set the SCKDIVCR and SCKDIVCR2 to the desired value.
2. Set the VSCR.VSCM[2:0] bit to 0x2.
3. Confirm that VSCR.VSCMTSF flag is 0.

(3) In case it is necessary to change the PLL setting when changing the VDD

1. Change the clock source to other than PLL.
2. Stop the PLL.
3. Change the PLL setting.
4. Set the VSCR.VSCM[2:0] bit to the desired value.
5. Confirm that VSCR.VSCMTSF flag is 0.
6. Enable the PLL.
7. Wait the stabilization time for PLL.
8. Change the clock source to PLL.

< Software Standby mode>

- It is possible to transfer to Software Standby mode and to return from Software Standby mode with VSCR set to something other than default at Normal mode.
  In Software Standby mode, SVSCR setting is enabled. SVSCR must be set before transitioning to Software Standby mode.
  VSCR setting is retained during Software Standby mode and after returning from Software Standby mode.
- It is forbidden to set SVSCR_1 and SS2LP[1:0] = 01b and transition to Software Standby mode.

<Deep Software Standby mode 1/2/3>

- It is possible to transfer to Deep Software Standby mode 1/2/3 and to return from Deep Software Standby mode 1/2/3 with VSCR set to something other than default at Normal mode.
  VSCR is initialized when returning from Deep Software Standby mode 1/2/3.

11.8 Usage Notes

11.8.1 Register Access

(1) Do not write to the registers listed below in either of the following conditions:

<Registers>

- All registers with a peripheral name of SYSC.

<Conditions>

- OPCCR.OPCMTSF = 1 (during transition of operating power control mode)
- MENTRYR.MENTRY = 1 (Extra MRAM program mode)
- VSCR.VSCMTSF = 1 (during transition of operating power control mode)
- Time period from executing WFI instructions from both CPUs to returning to Normal mode

(2) Do not write to the registers listed below by DTC or DMAC.

<Registers>

- MSTPCRA
- MSTPCRB
- MSTPCRC
- MSTPCRD
- MSTPCRE

(3) Write access to the registers listed below is invalid when PRCR.PRC1 bit is 0.

<Registers>

- OPCCR
- PGSCR

- PSSTCRn
- PDCTRESWM
- PDRAMSCRn
- SBYCR
- SSCR1
- LPSCR
- DPSBYCR
- DPSIERn
- DPSIFRn
- DPSIEGRn
- PLL1LDOSCR
- PLL2LDOSCR
- HOCOLDOSCR
- LVOCR
- VSCR
- SVSCR
- CPUDSCR
- MWMCR

(4) Write access to the registers listed below is invalid when PRCR.PRC4 bit is 0.

<Registers>

- LPMSAR
- PGCSAR
- DPFSAR
- DPFSAR1
- RSCSAR

11.8.2 I/O Port States

I/O port states in Software Standby mode, Deep Software Standby mode are as same as before entering the modes. Therefore, the supply current is not reduced while output signals are held high.

11.8.3 Module-Stop State of DMAC and DTC

Before writing 1 to MSTPCRA.MSTPA22 or MSTPCRA.MSTPA23, clear the DMAST.DMST bit of the DMAC and the DTCST.DTCST bit of the DTC to 0. For details, see section 17, DMA Controller (DMAC) and section 18, Data Transfer Controller (DTC).

11.8.4 Internal Interrupt Sources

Interrupts do not operate in the module-stop state. If setting module-stop bit while an interrupt request is generated, a CPU interrupt source or a DMAC or DTC startup source cannot be cleared. For this reason, make sure to disable the corresponding interrupts before setting the module-stop bits.

11.8.5 Input Buffer Control by DIRQnE Bit

Setting the DPSIERy.DIRQnE (y = 0 or 1 or 4 or 5, n = 0 to 31) bit to 1 enables the input buffer of the IRQ0-DS to IRQ31-DS pins. Note that, although inputs to these pins are sent to the DPSIFRy.DIRQnF (y = 0 or 1 or 4 or 5, n = 0 to 31) bits, they are not sent to the interrupt controller, peripheral modules, and I/O ports.

11.8.6 Transition to Low Power Modes

Do not enter low power modes (CPU Deep Sleep mode, Software Standby mode, or Deep Software Standby mode) by executing a WFE instruction. It is possible to enter CPU Sleep mode by executing a WFE instruction. In that case, CPU Sleep mode is canceled by the SEV instruction executed by another CPU, or any available interrupt request.

11.8.7 Timing of WFI Instructions

It is possible for the WFI instruction to be executed before I/O register and CS area writes are complete, in which case operation may not be as intended. This can happen if the WFI is placed immediately after a write to a I/O register and CS area. To avoid this problem it is advisable to read back the register and CS area that was written to confirm that the write has completed.

11.8.8 Writing WDT/IWDT Registers by DMAC or DTC

Do not write registers in WDT or IWDT by DMAC or DTC while WDT or IWDT is stopped by entering processor low power mode.

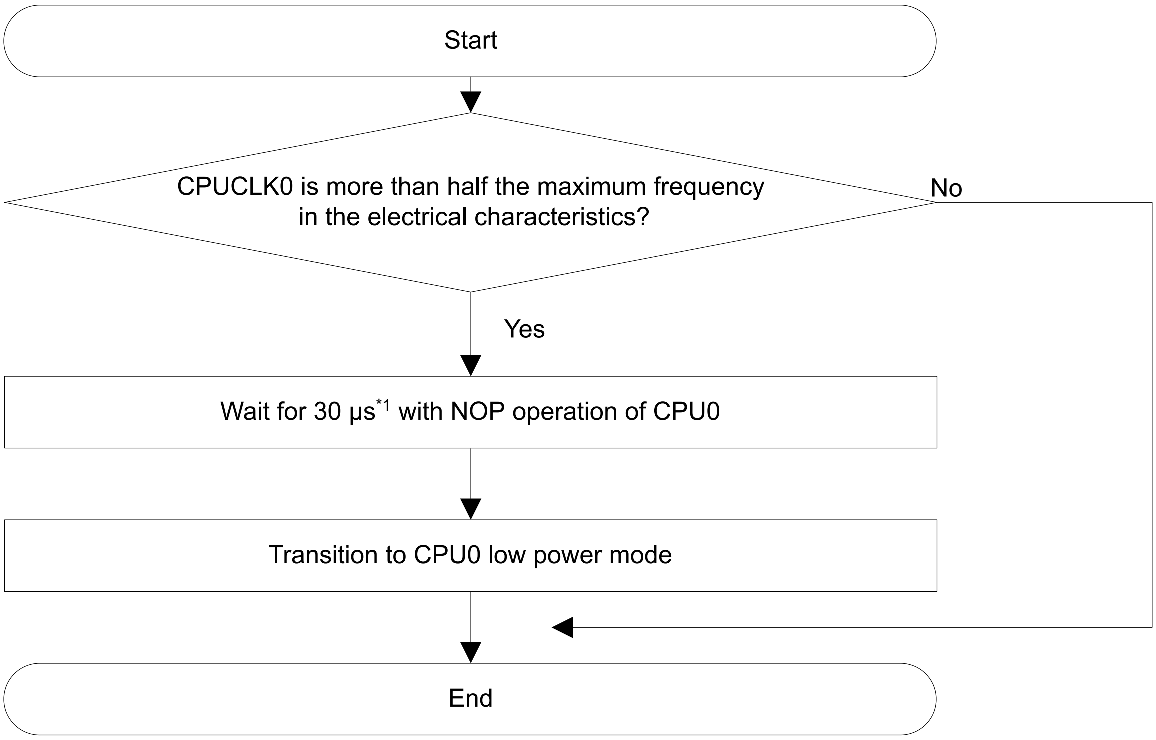
11.8.9 Module Stop Bit Write Timing

It is possible that the access to I/O register may be executed before the corresponding module stop bit write is completed. In this case, the access to I/O register may not proceed as intended. To avoid this issue, before accessing I/O register, read back the module stop bit that was written to confirm that the write completed.

11.8.10 Notes on Transitioning to or Canceling Low Power State

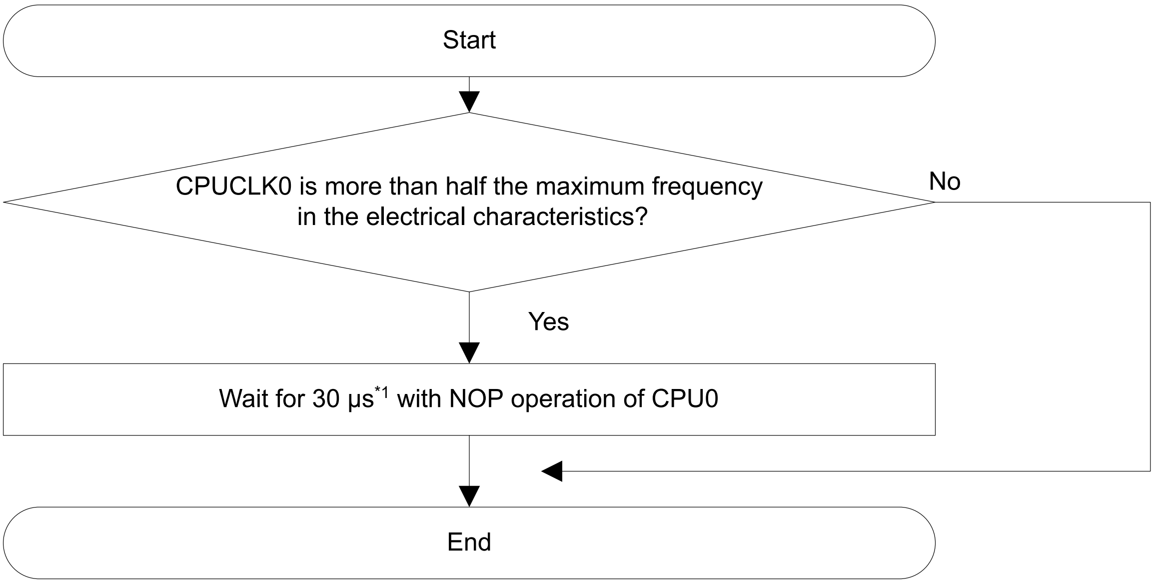
Transitioning to/returning from the low power state requires procedures.

These flows are shown in Figure 11.7 to Figure 11.12. It is recommended to use software to measure the wait time. Be sure to consider the worst-case conditions to ensure that the required wait time elapses. If an interrupt is unavoidably generated during the wait time, retry the measurement after return from the interrupt.



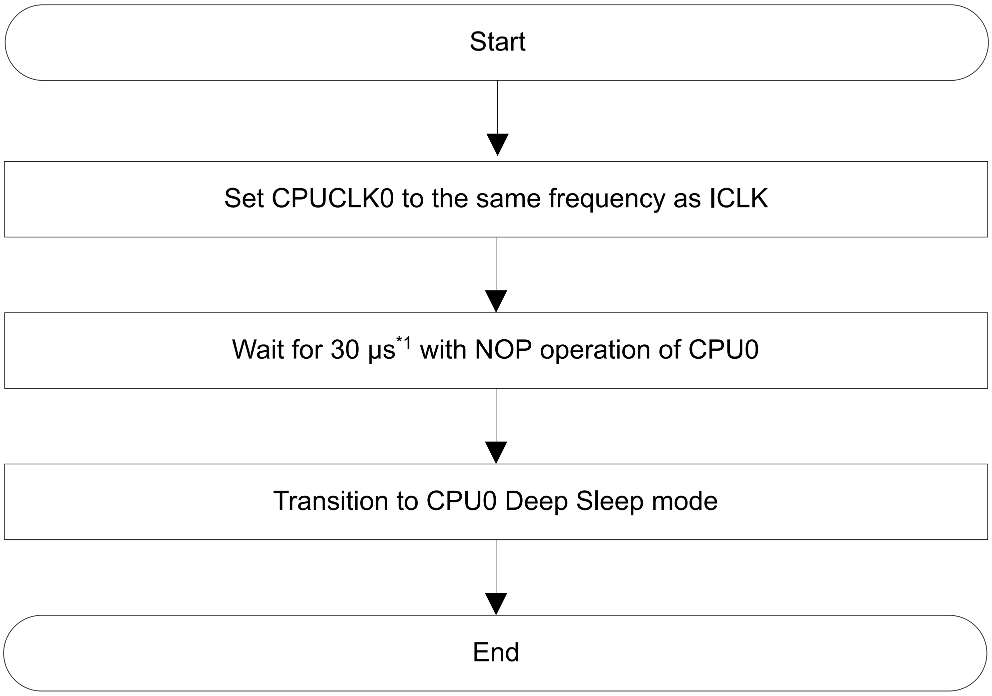
Note 1. 30 µs in DCDC mode, 10 µs in External VDD mode

Figure 11.7 Additional flow for the transition to CPU0 Sleep mode, CPU0 Deep Sleep mode with CPUDSCR.PGD0 bit is 1 according to CPUCLK0 frequency



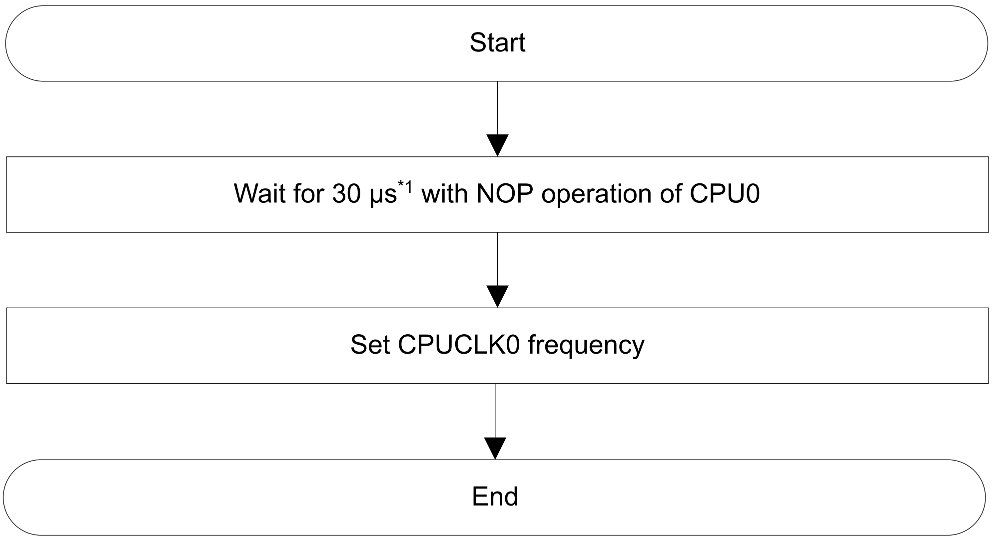
Note 1. 30 µs in DCDC mode, 10 µs in External VDD mode

Figure 11.8 Additional flow for the return from CPU0 Sleep mode, CPU0 Deep Sleep mode with CPUDSCR.PGD0 bit is 1 according to CPUCLK0 frequency



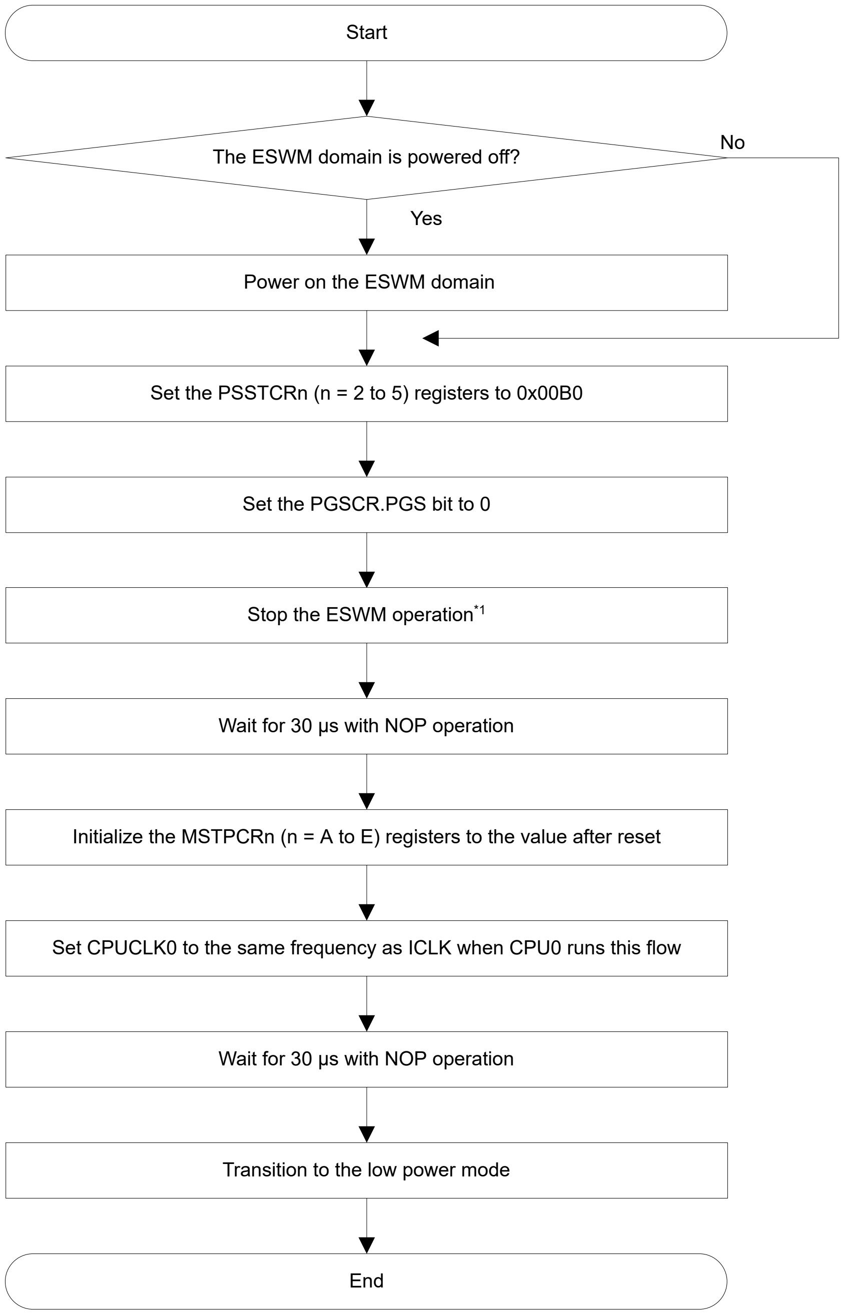
Note 1. 30 µs in DCDC mode, 10 µs in External VDD mode

Figure 11.9 Additional flow for the transition to CPU0 Deep Sleep mode with CPUDSCR.PGD0 bit is 0



Note 1. 30 µs in DCDC mode, 10 µs in External VDD mode

Figure 11.10 Additional flow for the return from CPU0 Deep Sleep mode with CPUDSCR.PGD0 bit is 0 (assuming that CPUCLK0 and ICLK are the same frequency when returning)



Note 1. See the Usage Notes in each chapter for how to stop the operation

Figure 11.11 Additional flow for the transition to Software Standby mode and Deep Software Standby mode 1/2/3

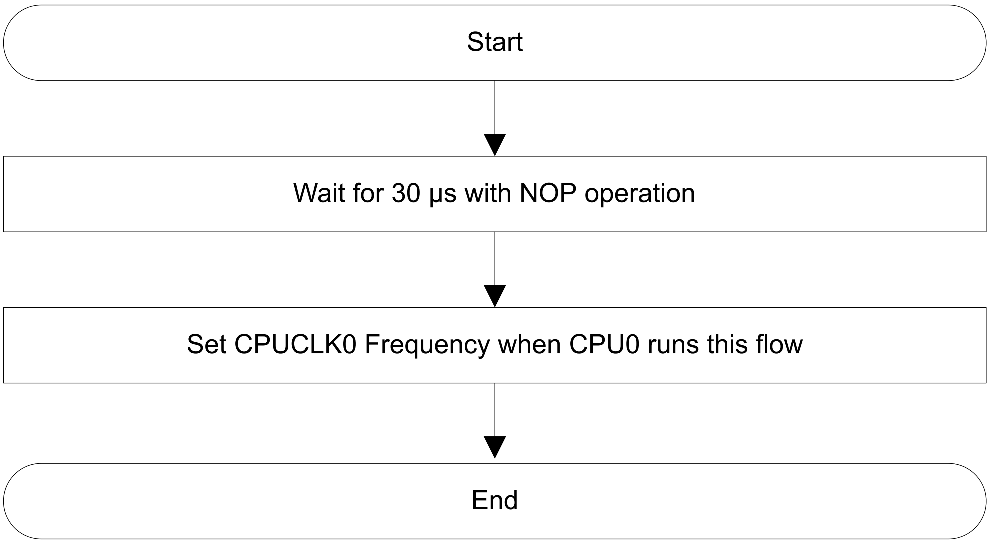


Figure 11.12 Additional flow for the return from Software Standby mode (assuming that CPUCLK0 and ICLK are the same frequency when returning)

11.8.11 Notes about the WFE instruction or Sleep-on-exit function

There are 3 ways to transition to low power modes, WFI instruction, WFE instruction and Sleep-on-exit. When WFE instruction or Sleep-on-exit is used for transition to low power modes, WFI instruction comments written in User’s Manual Hardware is applicable to WFE instruction or Sleep-on-exit.

12. Battery Backup Function

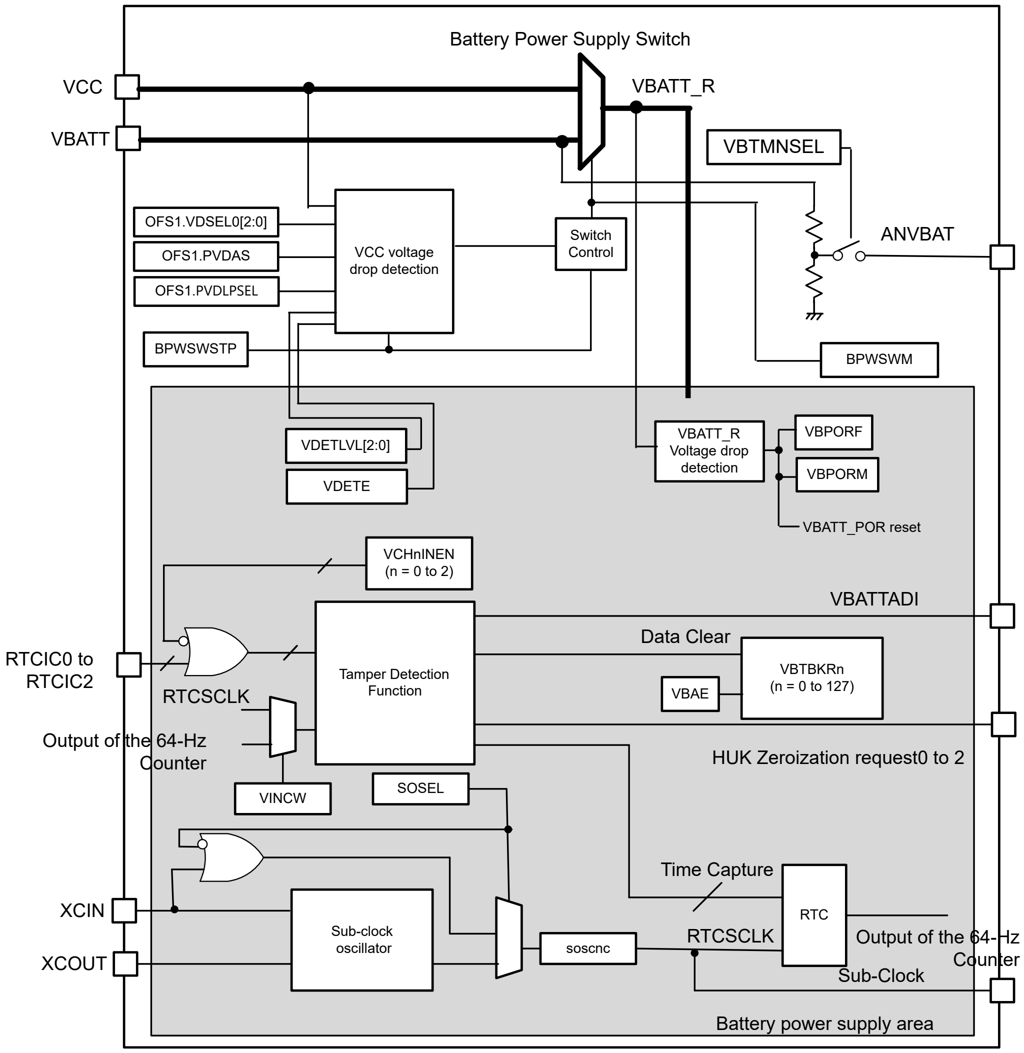
12.1 Overview

The MCU provides a battery backup function that maintains partial battery powering in the event of a power loss. Switching between VCC and VBATT, the battery-powered area includes RTC, SOSC, backup register, tamper detection and VBATT_R voltage drop detection. VBATT_R is the output voltage of the battery power supply switch. This is the power supply of the battery powered area.

During normal operation, the battery-powered area is powered by the main power supply, the VCC pin. When a VCC voltage drop is detected, the power source switches to the dedicated battery backup power pin, the VBATT pin. When the voltage rises again, the power source switches back from VBATT to VCC.

12.1.1 Battery Backup Function Block Diagram

Figure 12.1 shows the configuration of the battery backup function.



VBTMNSEL : VBATT Voltage Monitor Function Select Bit
VCHnINEN : VBATT CHn Input Enable bit (n = 0 to 2)
VDETLVL[2:0] : VDETBATT Level Select
VDETE : Voltage drop detection enable
BPWSWSTP : Battery Power Supply Switch Stop
VBPORF : VBATT_POR flag
BPWSWMON : Battery Power Supply Switch Status Monitor

VBTBKRn : VBATT Backup Register (n = 0 to 127)
VBAE : VBATT backup register access enable bit
OFS1 : Option Function Select Register 1
VDSEL0 : Voltage Detection 0 Level Select
PVDAS : Voltage Detection 0 Circuit Start
PVDLPSEL : PVD0 Select at Deep software standby mode
VINCW :VBATT Input Noise Canceler Width select
soscnc :Noise canceller for SOSC

Note: VCC : Main power supply pin
VBATT: Battery backup power supply pin
XCIN/XCOUT: SOSC input/output
RTCICn (n = 0 to 2): Input port for battery backup function

Figure 12.1 Battery backup function block diagram

12.1.2 Features of Battery Backup Function

The features include:

- Battery power supply switch
- VBATT_R voltage drop detection function
- Backup registers
- Time capture pin detection
- Tamper Detection function
- VBATT voltage monitor function

12.1.3 Battery Power Supply Switch

When the voltage applied to the VCC pin drops, this feature switches the power supply from the VCC pin to the VBATT pin. When the voltage rises, it switches the power supply from the VBATT pin back to the VCC pin.

When VCC is lower than $V_{DETBATT}$, and VCC is higher than VBAT, the injected current connects from the VCC to the VBATT pin through an internal diode. If the power supply battery connected to the VBATT pin cannot support this current injection, for example if the battery is not rechargeable, Renesas strongly recommends that you connect through a low-voltage threshold diode between the power supply battery and the VBATT pin.

It is necessary to enable voltage monitor 0 resets to use the battery backup function. The voltage monitor 0 level must be higher than the VBATT switch level.

12.1.4 VBATT_R Voltage Drop Detection Function

VBATT_R voltage drop detection function supports the battery-powered area. This function monitors the VBATT_R voltage level. VBATT_R is the output voltage of the battery power supply switch. This voltage drop detection causes a VBATT_POR reset and initializes the battery-powered area. See details in each register description. The VBATT status register includes a flag to check for this voltage drop detections.

12.1.5 Backup Registers

The battery-powered area provides 128-byte backup registers. These registers retain data when VBATT is supplied and VCC is powered off. When tampering is detected, data of backup register can be cleared to 0x00.

12.1.6 Time Capture Function

The RTC detects RTCICn (n = 0 to 2) pin input level change. For more information, see section 26, Realtime Clock (RTC).

12.1.7 Tamper Detection Function

The tamper detection function detects the RTCICn (n = 0 to 2) pin input event. The input event is defined as a change of RTCICn (n = 0 to 2) pin input level. The tamper detection flag is set to 1 by the input event. When interrupt is enabled and flag is set to 1, tamper detection interrupt is generated. When backup registers clear is enabled and flag is set to 1, the data of backup registers is cleared. Time Capture Function can select this flag as the source of the time capture trigger.

When Tamper Detection Zeroization Enable is enabled and flag is set to 1, the HUK Zeroization request is outputted.

12.1.8 VBATT Voltage Monitor Function

VBATT voltage monitor function is to monitor the input voltage level to the VBATT pin. The voltage level can be monitored as analog signal. For more information, see section 53, 16-bit A/D Converter (ADC16H).

12.2 Register Descriptions

12.2.1 BBFSAR : Battery Backup Function Security Attribute Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0x3D0

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	NONSEC6	NONSEC5	NONSEC4	NONSEC3	NONSEC2	NONSEC1	NONSEC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NONSEC0	Non-secure Attribute bit 0 Target register: VBATTMNSELR 0: Secure 1: Non-secure	R/W
1	NONSEC1	Non-secure Attribute bit 1 Target register: VBTBER 0: Secure 1: Non-secure	R/W
2	NONSEC2	Non-secure Attribute bit 2 Target register: VBTICTLR, VBTICTLR2, VBTIMONR 0: Secure 1: Non-secure	R/W
3	NONSEC3	Non-secure Attribute bit 3 Target register: VBTBPCR1, VBTBPCR2, VBTBPSR 0: Secure 1: Non-secure	R/W
4	NONSEC4	Non-secure Attribute bit 4 Target register: VBTADSR,VBTADCR1, VBTADCR2 0: Secure 1: Non-secure	R/W
5	NONSEC5	Non-secure Attribute bit 5 Target register: VBTADCR3 0: Secure 1: Non-secure	R/W
6	NONSEC6	Non-secure Attribute bit 6 Target register: VBTNCWCR 0: Secure 1: Non-secure	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1
Note: This register is write-protected by PRCR register.

The BBFSAR register controls the secure attribute of the battery backup function registers.

NONSEC0 bit (Non-secure Attribute bit 0)

This bit controls the security attribute of VBATTMNSELR.

NONSEC1 bit (Non-secure Attribute bit 1)

This bit controls the security attribute of VBTBER.

NONSEC2 bit (Non-secure Attribute bit 2)

This bit controls the security attribute of VBTICTLR, VBTICTLR2, and VBTIMONR.

NONSEC3 bit (Non-secure Attribute bit 3)

This bit controls the security attribute of VBTBPCR1 and VBTADCR2.

NONSEC4 bit (Non-secure Attribute bit 4)

This bit controls the security attribute of VBTADSR, VBTADCR1 and VBTADCR2.

NONSEC5 bit (Non-secure Attribute bit 5)

This bit controls the security attribute of VBTADCR3.

NONSEC6 bit (Non-secure Attribute bit 6)

This bit controls the security attribute of VBTNCWCR.

12.2.2 VBRSABAR : VBATT Backup Register Security Attribute Boundary Address Register

Base address: SYSC = 0x4001_E000

Offset address: 0x3B0

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

Bit	Symbol	Function	R/W
15:0	SABA[15:0]	Boundary address between secure and non-secure	R/W

Note: S-TYPE-6, P-TYPE-2

VBRSABAR specify the boundary address between Secure and non-secure regions of VBATT Backup Register. This register specifies lower 16 bits of VBATT Backup Register address. Secure region is less than SABA. Non-secure region is SABA or higher. The boundary address can set in units of 32 bytes, so the value written from b4 to b0 should be 0. SABA has no effect other than VBATT backup register.

VBATT Backup register is separated as follows.

Secure region : 0x4001_ED00 ≤ Address < 0x4001_0000 + SABA

Non-secure region : 0x4001_0000 + SABA ≤ Address

Note: The initial value specifies an address greater than the end address of VBATT Backup Register, so all area is Secure. If you specify all area of the VBATT Backup Register as Non-secure, specify the top address in the VBATT Backup Register area.

12.2.3 VBRPABARS : VBATT Backup Register Privilege Attribute Boundary Address Register for Secure Region

Base address: SYSC = 0x4001_E000

Offset address: 0x3B4

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

Bit	Symbol	Function	R/W
15:0	PABAS[15:0]	Privilege Attribute Boundary Address for Secure Region Boundary address between Privileged area and any-Privileged area in Secure	R/W

Note: S-TYPE-6, P-TYPE-2

VBRPABARS specify the boundary address between Privileged area and Any-Privileged area in Secure regions of VBATT Backup Register. This register further separates the Secure region of VBATT Backup Register set by VBRSABAR register into Privileged and Any-Privileged areas. This register specifies lower 16 bits of VBATT Backup Register address. Privileged area is less than PABAS. Any-Privileged area is PABAS or higher. The boundary address can set in units of 32byte, so the value written from b4 to b0 should be 0. PABAS has no effect other than VBATT backup register.

VBATT Backup register is separated as follows.

Privileged area in Secure region : 0x4001_ED00 ≤ Address < 0x4001_0000 + PABAS
0x4001_ED00 ≤ Address < 0x4001_0000 + SABA

This area that satisfies both address conditions.

Any-Privileged area in Secure region : 0x4001_0000 + PABAS ≤ Address < 0x4001_0000 + SABA

Note: The initial value specifies before the top address of the VBATT Backup Register, so all the Secure region is Any-Privilege. In this case, the setting of this register is actually ignored.

If you specify Privileged for the entire VBATT Backup Register that you have set as the Secure region, specify VBRPABARS register to the 0xFFE0.

12.2.4 VBRPABARNS : VBATT Backup Register Privilege Attribute Boundary Address Register for Non-secure Region

Base address: SYSC_NS = 0x5001_E000

Offset address: 0x3B8

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

Bit	Symbol	Function	R/W
15:0	PABANS[15:0]	Privilege Attribute Boundary Address for Non-secure Region Boundary address between Privileged area and Any-Privileged area in Non-secure	R/W

Note: S-TYPE-7, P-TYPE-2

VBRPABARNS specify the boundary address between Privileged area and Any-Privileged area in Non-secure regions of VBATT Backup Register. This register further separates the Non-secure region of VBATT Backup Register set by VBRSABAR register into Privileged and Any-Privileged areas. This register specifies lower 16 bits of VBATT Backup Register address. Privileged area is less than PABANS. Any-Privileged area is PABANS or higher. The boundary address can set in units of 32byte, so the value written from b4 to b0 should be 0. PABANS has no effect other than VBATT backup register.

VBATT Backup register is separated as follows.

Privileged area in Non-Secure region : 0x4001_0000 + SABA ≤ Address < 0x4001_0000 + PABANS

Any-Privileged area in Non-Secure region : 0x4001_0000 + PABANS ≤ Address
0x4001_0000 + SABA ≤ Address

This area that satisfies both address conditions.

Note: The initial value specifies before the top address of VBATT Backup Register, so all the Non-secure region is Any-Privilege. In this case, the setting of this register is actually ignored.

If you specify Privileged for the entire VBATT Backup Register that you have set as the Non-secure region, specify VBRPABARNS register to the 0xFFE0.

12.2.5 VBATTMNSELR : Battery Backup Voltage Monitor Function Select Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA84

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

Bit	Symbol	Function	R/W
0	VBTMNSEL	VBATT Voltage Monitor Function Select Bit Select VBATT voltage monitor function 0: Disables VBATT voltage monitor function 1: Enables VBATT voltage monitor function	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

VBATTMNSELR is the register which controls VBATT voltage monitor function.

This register is initialized by all reset sources except VBATT_POR reset.

VBTMNSEL bit (VBATT Voltage Monitor Function Select Bit)

Select VBATT low voltage monitor function. After setting this bit to 1, it is necessary to wait t_{MONWT} for the monitor level to stabilize.

Consumption current increases while VBTMNSEL = 1. So, after monitoring the VBATT voltage level, clear VBTMNSEL to 0 in order to reduce power consumption of VBATT power supply.

For more information on t_{MONWT}, see section 62, Electrical Characteristics.

12.2.6 VBTBER : VBATT Backup Enable Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC40

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

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	VBAE	VBATT backup register access enable bit 0: Disable to access VBTBKR 1: Enable to access VBTBKR	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

VBAE bit is initialized by the power on reset of VCC or after transition to VBATT mode.

VBAE bit (VBATT backup register access enable bit)

You must write 1 to VBAE before accessing VBTBKR. And you must write 0 after finishing all access (Write or Read) to VBTBKR. If you do not write 0 to VBAE, the data of VBTBKR does not be kept at VBATT mode.

Also, while VBAE is set to 0, the value of VBTBKR can be retained even if the VDD power supply and VCC power supply are powered off.

(The value can be retained even if the Core Voltage Monitor reset and the Overcurrent protection reset occurs after setting VBAE to 0.)

To access VBTBKR, wait for at least 500 ns after writing 1 to VBAE, and then access VBTBKR.

Before entering the Deep Software Standby mode, you must write 0 to VBAE.

To enter the deep software standby mode, wait at least 250 ns after writing 0 to VBAE, and then enter the Deep Software Standby mode.

If you don't use VBTBKR, you should change VBAE to 0 in order to reduce power consumption of VBTBKR.

12.2.7 VBTBKRn : VBATT Backup Register (n = 0 to 127)

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xD00 + 0x001 × n

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

Bit	Symbol	Function	R/W
7:0	VBTBKRn[7:0]	VBATT Backup Register The value of this register is retained even in VBATT mode. This register is not initialized by any reset sources.	R/W

Note: S-TYPE-3, P-TYPE-3

VBTBKRn[7:0] bits (VBATT Backup Register)

The value of this register is retained even in VBATT mode.

You can use 32, 16 or 8-bit access instruction when accessing the VBTBKRn register. When accessing, please note that the byte order of the data stored in the VBTBKRn register is little endian.

The data of VBTBKRn register is cleared to 0x00 by VBATT_POR reset or tamper detection function.

12.2.8 VBTICTLR : VBATT Input Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC4C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	VCH2INEN	VCH1INEN	VCH0INEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	VCH0INEN	VBATT CH0 Input Enable 0: RTCIC0 input disable 1: RTCIC0 input enable	R/W
1	VCH1INEN	VBATT CH1 Input Enable 0: RTCIC1 input disable 1: RTCIC1 input enable	R/W
2	VCH2INEN	VBATT CH2 Input Enable 0: RTCIC2 input disable 1: RTCIC2 input enable	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

VBTICTLR is the register that can select RTCICn (n = 0 to 2) pin as input. This register is only initialized by VBATT_POR reset.

VCHnINEN bits (VBATT CHn Input Enable Bits) (n = 0 to 2)

The VCHnINEN bit enables the input direction on the RTCICn. A wait time of 50 µs is required for the operation to stabilize after the RTCICn pin input is enabled.

12.2.9 VBTICTLR2 : VBATT Input Control Register 2

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC4D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	VCH2 EG	VCH1 EG	VCH0 EG	—	VCH2 NCE	VCH1 NCE	VCH0 NCE
Value after reset:	0	1	1	1	0	0	0	0

Bit	Symbol	Function	R/W
0	VCH0NCE	VBATT CH0 Input Noise Canceler Enable 0: RTCIC0 pin input noise canceler disable 1: RTCIC0 pin input noise canceler enable	R/W
1	VCH1NCE	VBATT CH1 Input Noise Canceler Enable 0: RTCIC1 pin input noise canceler disable 1: RTCIC1 pin input noise canceler enable	R/W
2	VCH2NCE	VBATT CH2 Input Noise Canceler Enable 0: RTCIC2 pin input noise canceler disable 1: RTCIC2 pin input noise canceler enable	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	VCH0EG	VBATT CH0 Input Edge Select 0: RTCIC0 pin input event is detected on falling edge 1: RTCIC0 pin input event is detected on rising edge	R/W
5	VCH1EG	VBATT CH1 Input Edge Select 0: RTCIC1 pin input event is detected on falling edge 1: RTCIC1 pin input event is detected on rising edge	R/W
6	VCH2EG	VBATT CH2 Input Edge Select 0: RTCIC2 pin input event is detected on falling edge 1: RTCIC2 pin input event is detected on rising edge	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

VBTICTLR2 is the register that can select RTCICn (n = 0 to 2) pin input mode. This register is only initialized by VBATT_POR reset.

VCHnNCE bits (VBATT CHn Input Noise Canceler Enable) (n = 0 to 2)

The VCHnNCEN bit enables the input noise canceler on the RTCICn (n = 0 to 2) pin input.

VCHnEG bits (VBATT CHn Input Edge Select) (n = 0 to 2)

The VCHnEG bit selects input event detection edge to set VBTADFn flag.

12.2.10 VBTIMONR : VBATT Input Monitor Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC4E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	VCH2 MON	VCH1 MON	VCH0 MON
Value after reset:	0	0	0	0	0	x	x	x

Bit	Symbol	Function	R/W
0	VCH0MON	VBATT CH0 Input monitor 0: RTCIC0 pin input is low level 1: RTCIC0 pin input is high level.	R
1	VCH1MON	VBATT CH1 Input monitor 0: RTCIC1 pin input is low level 1: RTCIC1 pin input is high level	R
2	VCH2MON	VBATT CH2 Input monitor 0: RTCIC2 pin input is low level 1: RTCIC2 pin input is high level.	R
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

VBTIMONR is the register that can monitor RTCICn (n = 0 to 2) pin input level.

VCHnMON bits (VBATT CHn Monitor) (n = 0 to 2)

The VCHnMON bit indicates input level on the RTCICn (n = 0 to 2) pin.

12.2.11 VBTBPCR1 : VBATT Battery Power Supply Control Register 1

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xA88

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

Bit	Symbol	Function	R/W
0	BPWSWSTP	Battery Power Supply Switch Stop 0: Battery power supply switch enable 1: Battery power supply switch stop	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

The VBTBPCR1 register controls battery power supply switch. This register is initialized by all reset sources except Deep Software Standby reset and VBATT_POR reset.

BPWSWSTP bit (Battery Power Supply Switch Stop)

The BPWSWSTP bit can enable switching the backup module supply power source from VCC to VBATT when the voltage applied to the VCC pin drops.

When stop is selected, the battery backup module power supply is always from VCC. Set the VDETE bit to 0 after setting theBPWSWSTP bit to 1. The BPWSWSTP bit must not be set from 1 to 0 while the VDETE bit is 1. If this setting is made, the state of the backup power area cannot be guaranteed.

If the BPWSWSTP bit is changed from 1 to 0, set the value of the VBTBPCR2.VDETLVL before it.

After setting the BPWSWSTP value from 1 to 0, wait for t_{DETWT} then set it to the VBTBPCR2.VDETE bit is 1.

12.2.12 VBTBPCR2 : VBATT Battery Power Supply Control Register 2

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC45

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	VDETE	—	VDETLVL[2:0]		
Value after reset:	0	0	0	0	0	1	1	0

Bit	Symbol	Function	R/W
2:0	VDETLVL[2:0]	$V_{DETBATT}$ Level Select 0 0 0: 2.80 V 0 0 1: 2.53 V 0 1 0: 2.10 V 0 1 1: 1.95 V 1 0 0: 1.85 V 1 0 1: 1.75 V 1 1 0: setting prohibited 1 1 1: setting prohibited	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	VDETE	Voltage drop detection enable 0: VCC Voltage drop detection disable 1: VCC Voltage drop detection enable	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

The VBTBPCR2 register controls battery power supply switch. This register is only initialized by VBATT_POR reset.

VDETLVL[2:0] bits ($V_{DETBATT}$ Level Select)

The VDETLVL[2:0] bits select $V_{DETBATT}$ level. When the voltage of the VCC pin drops below $V_{DETBATT}$, the power supply source switches from the VCC pin to the VBATT pin. The $V_{DETBATT}$ level should be below Voltage monitoring 0 level.

This MCU can reduce power consumption of the Deep Software Standby mode 1 or 2. When the MCU enters the Deep Software Standby mode 1 or 2 with that OFS1.PVDAS = 0 and OFS1.PVDLPSEL = 0, the battery power supply switch is controlled by the voltage monitor 0 which has lower power consumption than VCC voltage drop detection circuit dedicated for VBATT.

The VDETLVL[2:0] bits should change while the VDETE bit is 0. The VCC voltage detection function needs t_{DETWT} wait time for stabilization at the change of $V_{DETBATT}$ level. The VDETE bit should be set to 1 after this wait time.

For details on VDETLVL[2:0] bits, see section 12.3.2. VBATT Battery Power Supply Switch Usage.

VDETE bit (Voltage drop detection enable)

The VDETE bit enables the VCC voltage drop detection function for the battery power supply switch. The initial value of the VDETE bit is 0, at this time, the VCC voltage drop detection function for the battery power supply switch is disabled. If you use the battery power supply switch function, you must select $V_{DETBATT}$ level and enables the VCC voltage drop detection function.

For details on VDETE bit, see section 12.3.2. VBATT Battery Power Supply Switch Usage.

For more information on t_{DETWT}, see section 62, Electrical Characteristics.

12.2.13 VBTBPSR : VBATT Battery Power Supply Status Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC46

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	BPWSWM	VBPORM	—	—	—	VBPORF
Value after reset:	0	0	x	x	0	0	0	x

Bit	Symbol	Function	R/W
0	VBPORF	VBATT_POR Flag 0: VBATT_R voltage drop is not detected 1: VBATT_R voltage drop is detected	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	VBPORM	VBATT_POR Monitor 0: VBATT_R voltage < $V_{PORBATT}$ 1: VBATT_R voltage > $V_{PORBATT}$	R
5	BPWSWM	Battery Power Supply Switch Status Monitor 0: VCC voltage < $V_{DETBATT}$ 1: VCC voltage > $V_{DETBATT}$	R
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

The VBTBPSR register indicate battery power supply status.

VBPORF flag (VBATT_POR Flag)

The VBPORF flag indicates that VBATT_R voltage drop was detected and VBATT_POR reset was asserted.

[Setting condition]

- When VBATT_R voltage drops below V_{PORBAT}.

[Clearing condition]

- When 0 is written.

VBPORM bit (VBATT_POR Monitor)

The VBPORM bit indicates comparison result between VBATT_R and V_{PORBAT}.

BPWSWM bit (Battery Power Supply Switch Status Monitor)

The BPWSWM bit indicates comparison result between VCC and $V_{DETBATT}$.

12.2.14 VBTADSR : VBATT Tamper Detection Status Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC48

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	VBTADF2	VBTADF1	VBTADF0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	VBTADF0	VBATT Tamper Detection flag 0 0: RTCIC0 input edge is not detected 1: RTCIC0 input edge is detected	R/W
1	VBTADF1	VBATT Tamper Detection flag 1 0: RTCIC1 input edge is not detected 1: RTCIC1 input edge is detected	R/W
2	VBTADF2	VBATT Tamper Detection flag 2 0: RTCIC2 input edge is not detected 1: RTCIC2 input edge is detected	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

The VBTADSR register indicate tamper detection function status. This register is only initialized by VBATT_POR reset.

VBTADFn flags (VBATT Tamper Detection flag n) (n = 0, 1, 2)

The VBTADFn (n = 0, 1, 2) flag indicates that RTCICn (n = 0, 1, 2) input edge is detected. The edge type of RTCICn (n = 0, 1, 2) input can be selected by VCHnEG (n = 0, 1, 2) bit.

[Setting condition]

- When RTCICn (n = 0, 1, 2) input edge is detected.

[Clearing conditions]

- When 0 is written after 1 is read from VBTADFn (n = 0, 1, 2) flag.

12.2.15 VBTADCR1 : VBATT Tamper Detection Control Register 1

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC49

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	VBTADCE2	VBTADCE1	VBTADCE0	—	VBTADIE2	VBTADIE1	VBTADIE0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	VBTADIE0	VBATT Tamper Detection Interrupt Enable 0 0: Interrupt by VBTADF0 flag is disable 1: Interrupt by VBTADF0 flag is enable	R/W
1	VBTADIE1	VBATT Tamper Detection Interrupt Enable 1 0: Interrupt by VBTADF1 flag is disable 1: Interrupt by VBTADF1 flag is enable	R/W
2	VBTADIE2	VBATT Tamper Detection Interrupt Enable 2 0: Interrupt by VBTADF2 flag is disable 1: Interrupt by VBTADF2 flag is enable	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	VBTADCE0	VBATT Tamper Detection Backup Register Clear Enable 0 0: Clear Backup Register by VBTADF0 flag is disable 1: Clear Backup Register by VBTADF0 flag is enable	R/W
5	VBTADCE1	VBATT Tamper Detection Backup Register Clear Enable 1 0: Clear Backup Register by VBTADF1 flag is disable 1: Clear Backup Register by VBTADF1 flag is enable	R/W
6	VBTADCE2	VBATT Tamper Detection Backup Register Clear Enable 2 0: Clear Backup Register by VBTADF2 flag is disable 1: Clear Backup Register by VBTADF2 flag is enable	R/W

Bit	Symbol	Function	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

The VBTADCR1 register control tamper detection functions. This register is only initialized by VBATT_POR reset.

VBTADIEn bits (VBATT Tamper Detection Interrupt Enable) (n = 0, 1, 2)

The VBTADIEn (n = 0, 1, 2) bit enables Tamper detection interrupt.

VBTADCEn bits (VBATT Tamper Detection Backup Register Clear Enable) (n = 0, 1, 2)

The VBTADCEn (n = 0, 1, 2) bit enables backup register clearing by tamper detection flag.

12.2.16 VBTADCR2 : VBATT Tamper Detection Control Register 2

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC4A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	VBRTCES2	VBRTCES1	VBRTCES0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	VBRTCES0	VBATT RTC Time Capture Event Source Select 0 0: RTCIC0 1: VBTADF0	R/W
1	VBRTCES1	VBATT RTC Time Capture Event Source Select 1 0: RTCIC1 1: VBTADF1	R/W
2	VBRTCES2	VBATT RTC Time Capture Event Source Select 2 0: RTCIC2 1: VBTADF2	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

The VBTADCR2 register control tamper detection functions. This register is only initialized by VBATT_POR reset.

VBRTCESn bits (VBATT RTC Time Capture Event Source Select) (n = 0, 1, 2)

The VBRTCESn (n = 0, 1, 2) bit selects RTC time capture event source.

When the noise cancellation function of the RTCIC pin input and the Sub-clock oscillation stop detection function are both enabled (VBTICTLR2.VCHnNCEN = 1 and SOSTDCR. SOSTDE = 1), set RTC Time Capture Event Source to VBTADFn (VBRTCESn = 1).

12.2.17 VBTADCR3 : VBATT Tamper Detection Control Register 3

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC54

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	VBTADZE2	VBTADZE1	VBTADZE0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	VBTADZE0	VBATT Tamper Detection Zeroization Enable 0 0: Zeroing HUK request by VBTADF0 flag is disable 1: Zeroing HUK request by VBTADF0 flag is enable	R/W
1	VBTADZE1	VBATT Tamper Detection Zeroization Enable 1 0: Zeroing HUK request by VBTADF1 flag is disable 1: Zeroing HUK request by VBTADF1 flag is enable	R/W
2	VBTADZE2	VBATT Tamper Detection Zeroization Enable 2 0: Zeroing HUK request by VBTADF2 flag is disable 1: Zeroing HUK request by VBTADF2 flag is enable	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

The VBTADCR3 register control tamper detection functions. This register is only initialized by VBATT_POR reset.

VBTADZEn bit (VBATT Tamper Detection Zeroization Enable) (n = 0, 1, 2)

The VBTADZEn (n = 0, 1, 2) bit enables HUK Zeroization requests output by tamper detection flag.

12.2.18 VBTNCWCR : VBATT Noise Canceler Width Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xC50

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

Bit	Symbol	Function	R/W
2:0	VINCW[2:0]	VBATT Input Noise Canceler Width select 0 0 0: 32.768 kHz 0 0 1: 64 Hz 0 1 0: 32 Hz 0 1 1: 16 Hz 1 0 0: 8 Hz 1 0 1: 4 Hz 1 1 0: 2 Hz 1 1 1: 1 Hz	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

This register is only initialized by VBATT_POR reset.

The VBTNCWCR bit selects noise cancellation width of Tamper input.

Denoising is done using the circuit shown in Figure 12.5. The setting is the frequency of the noise sampling clock.

When VBTNCWCR is set to a value other than 3'b000, start the RTC 64 Hz counter.

12.3 Operation

12.3.1 Battery Backup Function

When the voltage on the VCC pin drops, power can be supplied to the VBATT_R backup power area from the VBATT pin. The power supply from the VCC pin is resumed when the voltage on the VCC pin exceeds $V_{DETBATT}$. When a drop of power supply from VCC pin is detected, backup power area enter the VBATT Mode. In the VBATT Mode, power supply is switched to VBATT pin. The power supply from the VCC pin is resumed when the voltage on the VCC pin exceeds $V_{DETBATT}$. This power supply change does not affect the VBATT_R backup power area function.

It is necessary to enable voltage monitor 0 reset to use the battery backup

The VBATT_R backup power area include following functions:

- RTC (including time capture detection, triggered by a change of the time capture pin input level)
- Sub-clock oscillator (including XCIN and XCOUT pins)
- The sub-clock oscillation stopping detection function
- VBATT Backup Register
- Tamper Detection Function
- VBATT Voltage Monitor Function

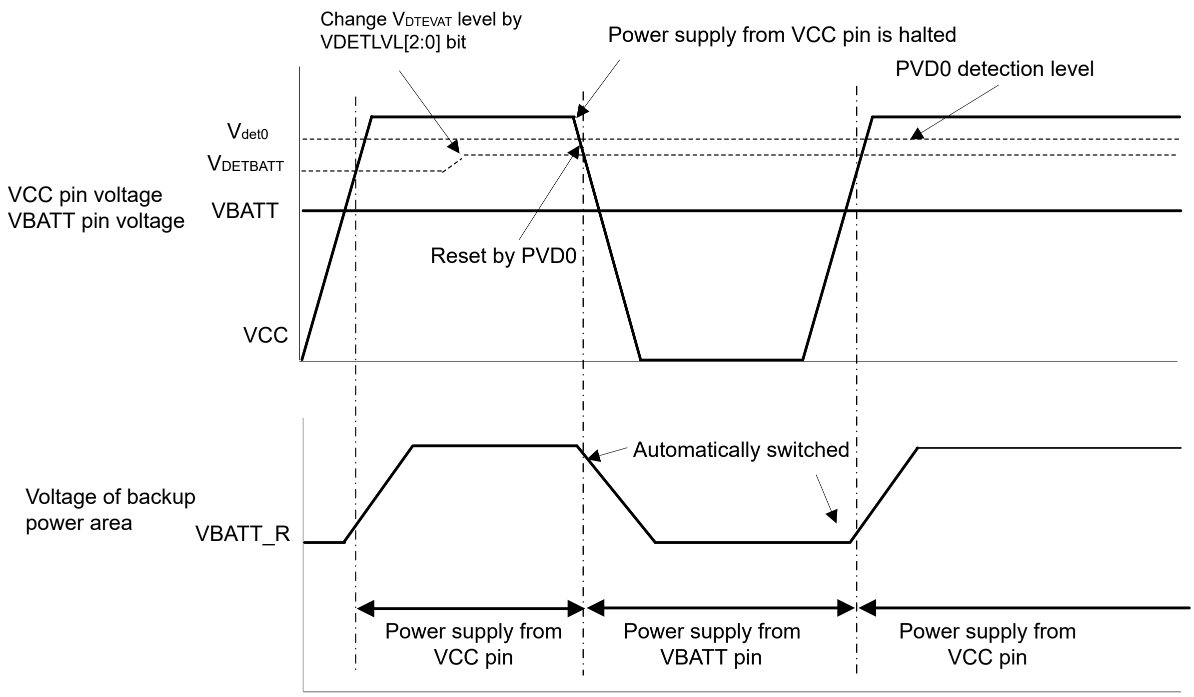
Table 12.1 shows the operating states in VBATT mode.

Table 12.1 Operating States in VBATT Mode

Operating state	VBATT Mode
Transition condition	Detection of VCC voltage drop
Canceling method other than reset	Detection of VCC voltage rise
State after cancellation by an interrupt	—
State after cancellation by a reset	—
Main clock oscillator	Stop
Sub-clock oscillator	Operating or not operating can be selected by SOSCCR.SOSTP bit. The status of the oscillator is same as before entering VBATT mode.
High-speed on-chip oscillator	Stop
Middle-speed on-chip oscillator	Stop
Low-speed on-chip oscillator	Stop
Sub-clock oscillation-stop detection function	Operating or not operating can be selected by SOSTDCR.SOSTDE bit. The status of the function is same as before entering VBATT mode.
PLL	Stop
PLL2	Stop
CPU	Stop (Undefined)
SRAM (ECC RAM included)	Stop (Undefined)
VBATT Backup Register	Stop (Retained or zeroized is selectable when Tamper input was detected.)
MRAM	Stop (Retained)
Realtime clock (RTC)	Selectable when selecting clock which is operating as the count source.
Programmable voltage detection circuit (PVD)	Stop
Power-on reset circuit	Stop
Other Peripheral modules	Stop (Undefined)
I/O ports	RTCICn ports (n = 0 to 2): Operating All ports not specified here: Undefined

Note: Stop (Retained) means that the contents of the internal registers are retained, but the operations are suspended.
Note: Stop (Undefined) means that the contents of the internal registers are undefined and power to the internal circuit is cut off.

Figure 12.2 shows switching sequence of Battery backup function.



Note: For details, see section 62, Electrical Characteristics.

Note: $V_{DETBATT}$ indicates the threshold level of the power supply change between the VCC pin and the VBATT pin.

Figure 12.2 Switching sequence of Battery backup function.

12.3.2 VBATT Battery Power Supply Switch Usage

The battery power supply switch can switch the power supply from the VCC pin to the VBATT pin when the voltage being applied to the VCC pin drops. When the voltage rises, this switch changes the power supply from the VBATT pin to the VCC pin.

The VCC voltage drop detection function for the battery power supply switch in "Figure 12.1" consists of two voltage drop detectors. One is a VCC drop detector for normal operation of the battery power supply switch that operates according to the settings of the VDTETLVL[2:0] bit and the VDETE bit. The other is voltage monitoring 0 to reduce power consumption during the Deep Software Standby mode 1 or 2.

In cold start, VDETE bit is initialized by VBATT_POR reset and the VCC voltage drop detection function for the battery power supply switch is disabled.

To use the battery power supply switch, the proper $V_{DETBATT}$ level must be selected by VDETLVL[2:0] bits and wait t_{DETWT} for stabilization. After wait, the VCC voltage drop detection function is enabled by VDETE bit. The battery power supply switch has a constraint that the $V_{DETBATT}$ level selected by VDETLVL[2:0] bits should be below Voltage monitoring 0 level.

When the MCU transitions to the Deep Software Standby mode 1 or 2 and the low power consumption function of voltage monitor 0 is enabled, the VCC drop detector for normal operation of the battery power supply switch is stopped to reduce power consumption. At this time, the power supply switch is controlled by the voltage monitor 0. This control by voltage monitor 0 continues until it returns from Deep Software Standby mode.

The battery backup function should be used after the voltage monitoring 0 reset is enabled (OFS1.PVDAS bit is 0).

If you don't use battery power supply switch. You must set BPWSWSTP bit to 1 and short the VCC and VBATT pins. When BPWSWSTP bit is 1, switch is stop, and power is always supplied from VCC pin.

12.3.3 VBATT_R Voltage Drop Detection Function Usage

This function enables to detect VBATT_R voltage drop. When VBATT_R voltage is drop and lower than $V_{PDR\ (BATR)}$, VBATT_POR reset is asserted. The VBPORF flag is set to 1 by VBATT_POR reset. If VBPORF flag is 1, RTC counter and register is invalid and Battery Backup Function registers are reset. You must initialize the functions included backup power area.

Figure 12.3 shows VBATT_R voltage drop detection function.

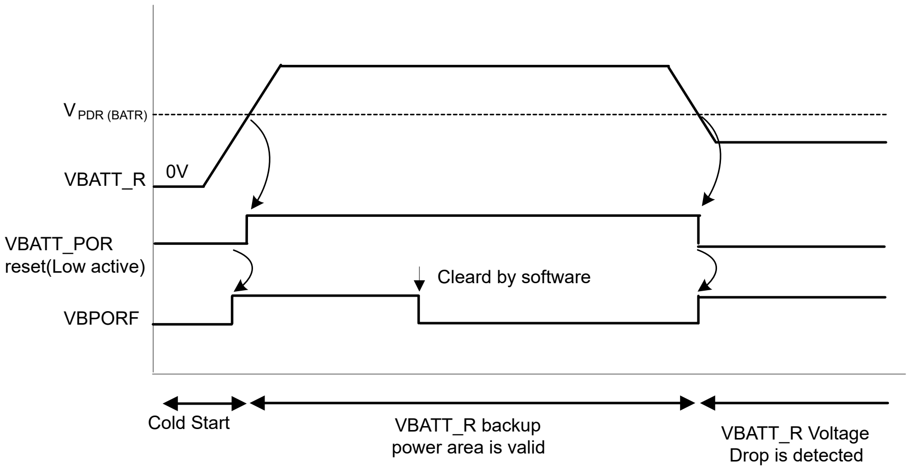


Figure 12.3 VBATT_R voltage drop detection function

12.3.4 VBATT Backup Register Usage

You can use 32, 16 or 8-bit access instruction when accessing the VBATT backup registers.

However, for example, when a 32-bit access instruction is executed, 8-bit read or write operation is executed with 4 consecutive times. When accessing, please note that the byte order of the data stored in the VBTBKR[n] (n = 0 to 127) register is little endian.

The data of VBATT backup register is cleared to 0x00 by VBATT_POR reset or tamper detection function.

The tamper detection function starts clearing operation of the VBATT backup register when the clear function is enabled and the tamper detection flag is set to 1. Do not cancel the clearing operation for 100 ns after starting. To cancel the clearing operation, disable the clear function or clear the tamper detection flag to 0. After canceling the clearing operation, do not access the VBATT backup register for 500 ns.

12.3.5 Tamper Detection Usage

The tamper detection function detects the RTCICn (n = 0 to 2) pin input event. The input event is defined as a change of RTCICn pin input level. The VBTADFn (n = 0 to 2) flag is set to 1 by the input event. When generate interrupt is enabled and the flag is set to 1, tamper detection interrupt is generated. When backup register clear function is enabled and flag is set to 1, the data of VBATT backup register is cleared. When Tamper Detection Zeroization Enable is enabled and flag is set to 1, the HUK Zeroization request is outputted. RTC time capture event source can select RTCICn pin input or the VBTADFn flag. If you need to synchronize time capture with a set of VBTADFn flags to 1, select the VBTADFn flag as the RTC time capture event source is recommended.

Figure 12.4 shows Tamper detection function.

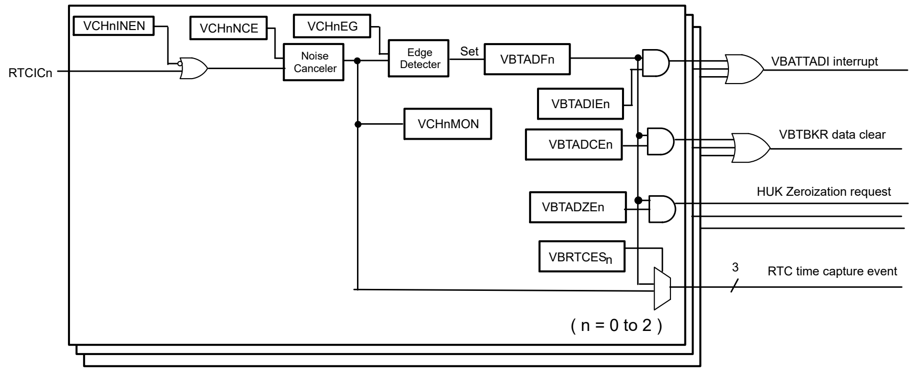


Figure 12.4 Tamper detection function

RTCICn pin inputs is enabled by the VCHnINEN (n = 0 to 2) bit. A wait time of 50us is required for the operation to stabilize after the RTCICn input is enabled.

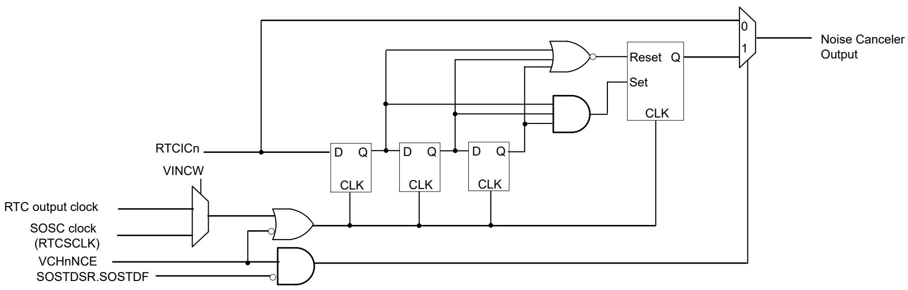
The VCHnNCE (n = 0 to 2) bit enables noise canceler. The noise canceler samples input signals at the Sub-Clock (RTCSCLK) or RTC output frequency and removes the pulses whose length is less than three sampling cycles.

The noise reduction sampling frequency can be selected in the VBTNCWCR.VINCW bit.

Figure 12.5 shows noise canceler circuit block diagram.

When changing the noise cancellation sampling frequency (changing the setting value of VBTNCWCR.VINCW bit), the VCHnNCE bit, VBTADCR1, VBTADCR2, and VBTADCR3 should be disabled.

The recommended initialization procedure is according to the flow example 3. For details, see section 12.3.7.4. Tamper Detection Function Initialization Setting Flow Example.



Note: n = 0 to 2

Figure 12.5 Noise canceler circuit block diagram

After enabling this noise canceler, a stabilization time of 5 clocks is required.

The VCHnEG (n = 0 to 2) bit enables to select Tamper detection edge. The edge detector operates asynchronously. When these control registers are changed, the VBTADFn flag may be set to 1 in a pseudo. The VBTADFn flag needs to be checked and clear to 0 after initialization of control registers. The VCHNnMON (n = 0 to 2) bit enables to monitor current input status. This bit also needs to be checked for inactive level after initialization of control registers.

The VBATTADI interrupt is asserted when 1 or more of 3 channels are flagged and interrupt is enabled.

The VBTBKRm(m = 0 to 127) register data is cleared to 00h when 1 or more of 3 channels are flagged and the VBATT backup register clear function is enabled.

12.3.6 VBATT Voltage Monitor Function Usage

You can monitor the input voltage level of VBATT pin. 1/6 of VBATT pin voltage level can be monitored as analog signal. After setting the VBTMNSEL bit to 1, it is necessary to wait t_{MONWT} for the monitor level to stabilize. For details on t_{MONWT}, see section 62, Electrical Characteristics.

Figure 12.6 shows VBATT voltage monitor function.

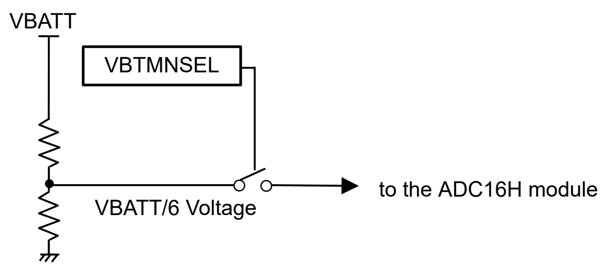


Figure 12.6 VBATT voltage monitor function

Note: For more information, see section 53, 16-bit A/D Converter (ADC16H).

Note: When VBTMNSEL bit is 1, VBAT power consumption is increased. It is recommended that VBTMNSEL bit sets to 1 only monitor timing.

12.3.7 Initial Settings Examples

12.3.7.1 Cold Start and Using the Power Supply Switch Flow Example

When turning on VCC pin and VBATT pin power for the first time, use the following flow to initialize.

1. Check VBPORM bit. If VBPORM flag is 0, wait until it changes to 1.
2. Clear VBPORF flag to 0.
3. Set the VDETLVL[2:0] bit to appropriate value. The $V_{DETBATT}$ level is Selected.
4. Wait t_{DETWT} for the VCC voltage detection function stabilization.
5. Set the VDETE bit to 1. The VCC voltage detection function is enabled.
6. Enable Sub-Clock Oscillator if needed.
7. Set Other Battery Backup Function and RTC registers.

For details on t_{DETWT}, see section 62, Electrical Characteristics.

For details on Sub-Clock Oscillator, see section 9, Clock Generation Circuit.

12.3.7.2 Warm Start and Using the Power Supply Switch Flow Example

The MCU transitions to VBATT mode when the power of the VCC pin is turned off while the power of the VBATT pin is on. In this section, warm start means when the power supply of the VCC pin is turned on in VBATT mode. When the MCU starts processing with a warm start, execute the following flow in order to check the VBATT_R voltage drop before executing other process.

1. Check VBPORM flag. If VBPORM flag is 0, wait until it changes to 1.
2. Check VBPORF flag. If VBPORF flag is 1, the VBATT_R voltage drop is detected. For details on re-initialize the backup power area, see section 12.3.7.1. Cold Start and Using the Power Supply Switch Flow Example. If not, it means that the VBATT_R voltage does not drop and the state of the backup power area is retained. So, you do not need to re-initialize the backup power area.

12.3.7.3 Not Using the Power Supply Switch Flow Example

When not using the power supply switch, VCC pin and VBATT pin should be shorted. In this case, the VBATT_POR reset cannot follow the power on reset of VCC and VBATT_R voltage drop may not be detected. So, initialize the backup power supply area with the following flow.

1. Set the BPWSWSTP bit to 1. The power supply switch is stopped.
2. Check VBPORM flag. If VBPORM flag is 1, wait until it changes to 0.
3. Clear the VDETE bit to 0. The VCC voltage drop detection function is stopped.
4. Clear the VDETLVL[2:0] bit to 110b. The initial value is selected.
5. Check VBPORF flag. When VBPORF flag is 1, clear it to 0.
6. Set the SOSTP bit to 1 regardless of its value. Stop Sub-Clock Oscillator.
7. Initialize the VBTICTLR register and SOMCR.SOSEL bit. Initialization is recommended because these registers are related to the control of the IO port.
8. Initialize Other Battery Backup Function registers if needed. Other Battery Backup Function registers are VBTICTLR2, VBTADSR, VBTADCR1, VBTADCR2, and VBTBKRn.
9. Enable Sub-Clock Oscillator if needed.
10. Set RTC registers.

For details on Sub-Clock Oscillator, see section 9, Clock Generation Circuit.

12.3.7.4 Tamper Detection Function Initialization Setting Flow Example

If using the tamper detection function, it is recommended to initialize according to the following flow example.

1. Set the VCHnINEN to appropriate value.
2. Wait 50us to stabilize after the RTCICn input.
3. Set the VBTNCWCR to appropriate value.
4. Set the VCHnNCE bit and the VCHnEG bit.
5. Wait 5 RTC-clocks to stabilize, if noise canceler is enabled.
6. Checking the VCHnMON bit indicates inactive is recommended. If it is active, it may not be possible to detect the tamper.
7. Initialize the invalid status of VBTADFn flag by clear to 0 after reading dummy read.
8. Set VBTADCR1, VBTADCR2,VBTADCR3 to enable interrupt, backup register clear, HUK Zeroization request output, and RTC time capture event.
9. Enables RTC time capture function if needed.

12.4 Interrupt Sources

The Battery Backup Function has two interrupt sources and are listed in Table 12.2.

Table 12.2 Battery Backup Function interrupt Sources

Symbol	Interrupt source	Interrupt flag	Interrupt conditions
VBATTADI	VBATT Tamper detection	VBTADF0	VBTADF0 = 1, VBTADIE0 = 1
		VBTADF1	VBTADF1 = 1, VBTADIE1 = 1
		VBTADF2	VBTADF2 = 1, VBTADIE2 = 1

12.5 Usage Notes

1. Operation of the sub-clock oscillator and RTC are not guaranteed when the voltage level on VBATT is lower than the guaranteed operation range. Initialize the RTC when the VBATT pin falls below the guaranteed operating voltage and then powers up again.
2. A reset generated while writing to registers described in this section might destroy the register value.

3. When VCC is higher than $V_{DETBATT\_m}$, the VCC pin and VBATT pin are separated. When VCC is lower than $V_{DETBATT}$ and the switch is connected to the VBATT pin, and if the voltage on VBATT drops lower than VCC, current might flow into the VBATT pin through the parasitic diode between the VCC and VBATT pins.
4. During RTC operation using the voltage from the VBATT pin and the I/O ports within the backup, the power supply area can only be used as time capture event input pins for the RTC.

13. Register Write Protection Function

13.1 Overview

The register write protection function protects important registers from being overwritten due to software errors. The registers to be protected are set with the Protect Register (PRCR_S and PRCR_NS).

The two protected registers work on one secure set registers/bits and the other for non-secure set registers/bits. They are collectively mentioned as PRCR.

Table 13.1 lists the association between the PRCR bits and the use of registers to be protected.

The register information to which the PRCR bit is applied is provided in the description of each register.

Table 13.1 Association between PRCR bits and use of registers to be protected

PRCR bit	Register to be protected
PRC0	• Registers related to the clock generation circuit: SCKDIVCR, SCKDIVCR2, SCKSCR, PLLCCR, PLLCR, BCKCR, MOSCCR, HOCOCR, HOCOCR2, MOCOCR, FLLCR1, FLLCR2, CKOCR, OSTDCR, OSTDSR, SOSTDCR, SOSTDSR, PLL2CCR, PLL2CR, PLLCCR2, PLL2CCR2, EBCKOCR, SDCKOCR, SCICKDIVCR, SCICKCR, SPICKDIVCR, SPICKCR, ADCCKDIVCR, ADCCKCR, GPTCKDIVCR, GPTCKCR, BCKADIVCR, ESWCKDIVCR, ESWPCKDIVCR, ETHPCKDIVCR, BCKACR, ESWCKCR, ESWPCKCR, ETHPCKCR, MOCOUTCR, HOCOUTCR, USBCKDIVCR, OCTACKDIVCR, CANFDCKDIVCR, USB60CKDIVCR, I3CCKDIVCR, USBCKCR, OCTACKCR, CANFDCKCR, USB60CKCR, I3CCKCR, MOSCSCR, HOCOSCR, PLLSCR, PLL2SCR, MOCOSCR, MOSCWTCR, LOCOCR, LOCOUTCR, MOMCR, MOMCR2, SOSCCR, SOMCR, SYRACCR, USBCKSELR
PRC1	• Registers related to the low power modes: OPCCR, PGSCR, PSSTCR2, PSSTCR3, PSSTCR4, PSSTCR5, PDCTRESWM, PDRAMSCR0, PDRAMSCR1, SBYCR, SSCR1, LPSCR, DPSBYCR, DPSWCR, DPSIER0-5, DPSIFR0-5, DPSIEGR0-4, LDOSCR, PLL1LDOCR, PLL2LDOCR, HOCOLDOCR, LVOCR, VSCR, SVSCR, MLSMCR, CPUDSCR, MWMCR • Register related to the battery backup function: VBTBER, VBTICTLR, VBTBKRn (n = 0 to 127), VBTBPCR1, VBTBPCR2, VBTBPSR, VBTADSR, VBTADCR1, VBTADCR2, VBTICTLR2, VBTADCR3, VBTNCWCR
PRC3	• Registers related to the PVD: PVD1CMPCR, PVD2CMPCR, PVD4CMPCR, PVD5CMPCR, PVD1CR0, PVD2CR0, PVD4CR0, PVD5CR0, PVD1CR1, PVD1SR, PVD2CR1, PVD2SR, PVD1FCR, PVD2FCR, PVD4FCR, PVD5FCR, PVDLR, VBATTMNSELR
PRC4	• Registers related to the security and privilege setting registers: CGFSAR, RSTSAR, LPMSAR, PVDSAR, BBFSAR, DPFSAR, DPFSAR1, RSCSAR, PGCSAR, VBRSABAR, VBRPABARS, VBRPABARNS, CPUSAR, DEBUGSAR, CACHESAR, TCMSAR, TCMSABARC, TCMSABARS, IPCSAR, IPCPAR, ICUSARx (x = A, B, E to M), TEVTRCR, BUSSARx (x = A, B, C), BUSPARC, MMPUSARx (x = A, B), DMACCHSAR, DMACCHPAR, DMACSAR, DTCSAR, ELCSARx (x = A, B, C), ELCPARx (x = A, B, C), PmSAR (m = 0 to 9, A, B, C, D), PSARx (x = B, C, D, E), MSSAR, PPARBx (x = B, C, D, E), MSPAR, SRAMSABARn (n = 0 to 3), SRAMSAR, SRAMESAR, MSAR
PRC5	• Registers related to the reset control: SYRSTMSK0-2, TEMPRCR, TEMPRLR

13.2 Register Descriptions

The base address of SYSC is 0x4001_E000 (Secure) and 0x5001_E000 (Non-secure).

Table 13.2 lists the registers related to the register write protection function.

Table 13.2 Register list

Register name	Symbol	Value after reset	Address	Access size
Protect Register for Secure Register	PRCR_S	0x0000	0x4001_E3FA	16
Protect Register for Non-secure Register	PRCR_NS	0x0000	0x5001_E3FE	16

13.2.1 PRCR_S : Protect Register for Secure

Base address: SYSC = 0x4001_E000

Offset address: 0x3FA

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PRKEY[7:0]								—	—	PRC5	PRC4	PRC3	—	PRC1	PRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PRC0	Enable writing to the registers related to the clock generation circuit 0: Write disabled 1: Write enabled	R/W
1	PRC1	Enable writing to the registers related to the low power modes, and the battery backup function 0: Write disabled 1: Write enabled	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	PRC3	Enable writing to the registers related to the PVD 0: Write disabled 1: Write enabled	R/W
4	PRC4	Enables writing to the registers related to the security and privilege setting registers 0: Write disabled 1: Write enabled	R/W
5	PRC5	Enables writing to the registers related to the reset control 0: Write disabled 1: Write enabled	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PRKEY[7:0]	PRC Key Code These bits control the write access to the PRCR_S register. To modify the PRCR_S register, write 0xA5 to the upper 8 bits and the target value to the lower 8 bits as a 16-bit unit.	W

Note: S-TYPE6, P-TYPE2

PRCR_S is used to protect registers that are always secure or configured as Secure.

PRC0 bit (Enable writing to the registers related to the clock generation circuit)

The PRC0 bit protects writing to the registers related to the clock generation circuit. Setting this bit to 1 enables writing to the corresponding registers. Setting this bit to 0 disables writing to the corresponding registers.

PRC1 bit (Enable writing to the registers related to the low power modes, and the battery backup function)

The PRC1 bit protects writing to the registers related to the low power modes and the battery backup function. Setting this bit to 1 enables writing to the corresponding registers. Setting this bit to 0 disables writing to the corresponding registers.

PRC3 bit (Enable writing to the registers related to the PVD)

The PRC3 bit protects writing to the registers related to the PVD. Setting this bit to 1 enables writing to the corresponding registers. Setting this bit to 0 disables writing to the corresponding registers.

PRC4 bit (Enables writing to the registers related to the security and privilege setting registers)

The PRC4 bit protects writing to the registers related to the security and privilege setting registers. Setting this bit to 1 enables writing to the corresponding registers. Setting this bit to 0 disables writing to the corresponding registers.

The register controlled by PRC4 may not reflect the PRC4 change when PRCR_S and its controlled registers are continuously written access. Avoid continuous write access or read the PRCR_S after PRC4 change, and then write access the PRC4-controlled register.

PRC5 bit (Enables writing to the registers related to the reset control)

The PRC5 bit protects writing to the registers related to the reset control. Setting this bit to 1 enables writing to the corresponding registers. Setting this bit to 0 disables writing to the corresponding registers.

PRKEY[7:0]bits (PRC Key Code)

The PRKEY[7:0] bits control permission and prohibition of writing to the PRCR_S register. To write PRCi bits of PRCR_S register, write 0xA5 to the PRKEY[7:0]. In case of writing other than 0xA5 to PRKEY[7:0], PRCi bits do not be changed even if writing to the PRCR_S register.

13.2.2 PRCR_NS : Protect Register for Non-Secure

Base address: SYSC_NS = 0x5001_E000

Offset address: 0x3FE

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PRKEY[7:0]								—	—	—	PRC4	PRC3	—	PRC1	PRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PRC0	Enable writing to the registers related to the clock generation circuit 0: Write disabled 1: Write enabled	R/W
1	PRC1	Enable writing to the registers related to the low power modes, and the battery backup function 0: Write disabled 1: Write enabled	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	PRC3	Enable writing to the registers related to the PVD 0: Write disabled 1: Write enabled	R/W
4	PRC4	Enables writing to the registers related to the privilege setting registers 0: Write disabled 1: Write enabled	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PRKEY[7:0]	PRC Key Code These bits control the write access to the PRCR_NS register. To modify the PRCR_NS register, write 0xA5 to the upper 8 bits and the target value to the lower 8 bits as a 16-bit unit.	W

Note: S-TYPE7, P-TYPE2

PRCR_NS is used to protect registers that are configured as Non-secure.

PRC0 bit (Enable writing to the registers related to the clock generation circuit)

The PRC0 bit protects writing to the registers related to the clock generation circuit. Setting this bit to 1 enables writing to the corresponding registers. Setting this bit to 0 disables writing to the corresponding registers.

PRC1 bit (Enable writing to the registers related to the low power modes, and the battery backup function)

The PRC1 bit protects writing to the registers related to the low power modes and the battery backup function. Setting this bit to 1 enables writing to the corresponding registers. Setting this bit to 0 disables writing to the corresponding registers.

PRC3 bit (Enable writing to the registers related to the PVD)

The PRC3 bit protects writing to the registers related to the PVD. Setting this bit to 1 enables writing to the corresponding registers. Setting this bit to 0 disables writing to the corresponding registers.

PRC4 bit (Enables writing to the registers related to the privilege setting registers)

The PRC4 bit protects writing to the registers related to the privilege setting registers. Setting this bit to 1 enables writing to the corresponding registers. Setting this bit to 0 disables writing to the corresponding registers.

The register controlled by PRC4 may not reflect the PRC4 change when PRCR_NS and its controlled registers are continuously written access. Avoid continuous write access or read the PRCR_NS after PRC4 change, and then write access the PRC4-controlled register.

PRKEY[7:0]bits (PRC Key Code)

The PRKEY[7:0] bits control permission and prohibition of writing to the PRCR_NS register. To write PRCi bits of PRCR_NS register, write 0xA5 to the PRKEY[7:0]. In case of writing other than 0xA5 to PRKEY[7:0], PRCi bits do not be changed even if writing to the PRCR_NS register.

14. Interrupt Controller Unit (ICU)

14.1 Overview

The Interrupt Controller Unit (ICU) controls which event signals are linked to the Nested Vector Interrupt Controller (NVIC), DMA Controller (DMAC) module and the Data Transfer Controller (DTC) modules. The ICU also controls non-maskable interrupts.

Table 14.1 lists the ICU specifications, Figure 14.1 shows a block diagram, and Table 14.2 lists the I/O pins.

Table 14.1 ICU specifications (1 of 2)

Parameter		Description
Maskable interrupts	Peripheral function interrupts	Interrupts from peripheral modules Number of sources: 497 (a factor is chosen with an event list number 33 to 1023)
	External pin interrupts	● Interrupt detection on low level*4, falling edge, rising edge, rising and falling edges. One of these detection methods can be set for each source ● Digital filter function supported ● 32 sources, with interrupts from IRQi (i = 0 to 31) pins.
	CPU mutual interrupt	● IPC CPU mutual interrupt 0 (IPC_IRQ0) ● IPC CPU mutual interrupt 1 (IPC_IRQ1)
	Interrupt requests to CPU (NVIC)	96 interrupt requests are output to NVIC.
	DMAC control	● The DMAC can be activated using interrupt sources*1 ● The target interrupt source can be selected individually for every DMAC channels.
	DTC control	● The DTC can be activated using interrupt sources*1 ● The method for selecting an interrupt source is the same as that of the interrupt request to NVIC.
Non-maskable interrupts*2	NMI pin interrupt	● Interrupt from the NMI pin ● Interrupt detection on falling edge or rising edge ● Digital filter function supported
	Oscillation stop detection interrupt*3	Interrupt on detection of main oscillation is stopped
	Sub Oscillation stop detection interrupt*3	Interrupt on detection of sub oscillation is stopped
	WDT underflow/refresh error*3	Interrupt on an underflow of the down-counter or occurrence of a refresh error (CPU0, CPU1)
	IWDT underflow/refresh error*3	Interrupt on an underflow of the down-counter or occurrence of a refresh error
	Voltage-monitor 1 interrupt*3	Voltage monitor interrupt of the voltage detection circuit 1 (PVD_PVD1)
	Voltage-monitor 2 interrupt*3	Voltage monitor interrupt of the voltage detection circuit 2 (PVD_PVD2)
	Common memory error interrupt	Common memory error means SRAM ECC error.
	Local memory error interrupt	Local memory error interrupt includes Cache and TCM ECC Error. (CPU1 only)
	Bus error Interrupt	Bus error interrupt includes MPU and TZF error
	Lockup error interrupt	Lockup error interrupt (CPU0, CPU1)
	CPU mutual Interrupt	IPC NMI CPU mutual interrupt (CPU0, CPU1)
	FPU Exception interrupt	FPU Exception interrupt (CPU0, CPU1)
	MRAM read error interrupt	● MRAM read error interrupt for MRC ● MRAM read error interrupt for MRE
Security	Secure	Some registers have Security Attribution.
	Privilege	Each register of the ICU can only be accessed with Privilege access.

Table 14.1 ICU specifications (2 of 2)

Parameter	Description
Return from low power mode	• CPU Sleep mode: Return is initiated by non-maskable interrupts or any other interrupt source. • CPU Deep Sleep mode: Return is initiated by non-maskable interrupts. Interrupt can be selected as WUPEN0, WUPEN1, DSLPWUPIRQENj register (j = 0 to 2). See section 14.2.4. ICUSARF : Interrupt Controller Unit Security Attribution Register F, section 14.2.9. ICUSARK : Interrupt Controller Unit Security Attribution Register K, section 14.2.10. ICUSARL : Interrupt Controller Unit Security Attribution Register L, section 14.2.11. TEVTRCR : Trusted Event Route Control Register, section 14.9.2. Return from CPU Deep Sleep Mode. • Software Standby mode: Return is initiated by non-maskable interrupts. Interrupt can be selected as WUPEN0, WUPEN1 register. See section 14.2.4. ICUSARF : Interrupt Controller Unit Security Attribution Register F, section 14.2.9. ICUSARK : Interrupt Controller Unit Security Attribution Register K, section 14.2.10. ICUSARL : Interrupt Controller Unit Security Attribution Register L, section 14.9.2. Return from CPU Deep Sleep Mode.

Note 1. For the DMAC and DTC activation sources, see Table 14.4.
Note 2. Non-maskable interrupts can be enabled only once after a reset release.
Note 3. These non-maskable interrupts can also be used as interrupts in general. When used as maskable interrupts, do not change the value of the NMIER register from the reset state. To enable voltage monitor 1 to 2 interrupts, set the PVD1CR1.PVD1IRQSEL and PVD2CR1.PVD2IRQSEL bits to 1.
Note 4. Low level: interrupt detection is not canceled if you do not clear it after a detection.

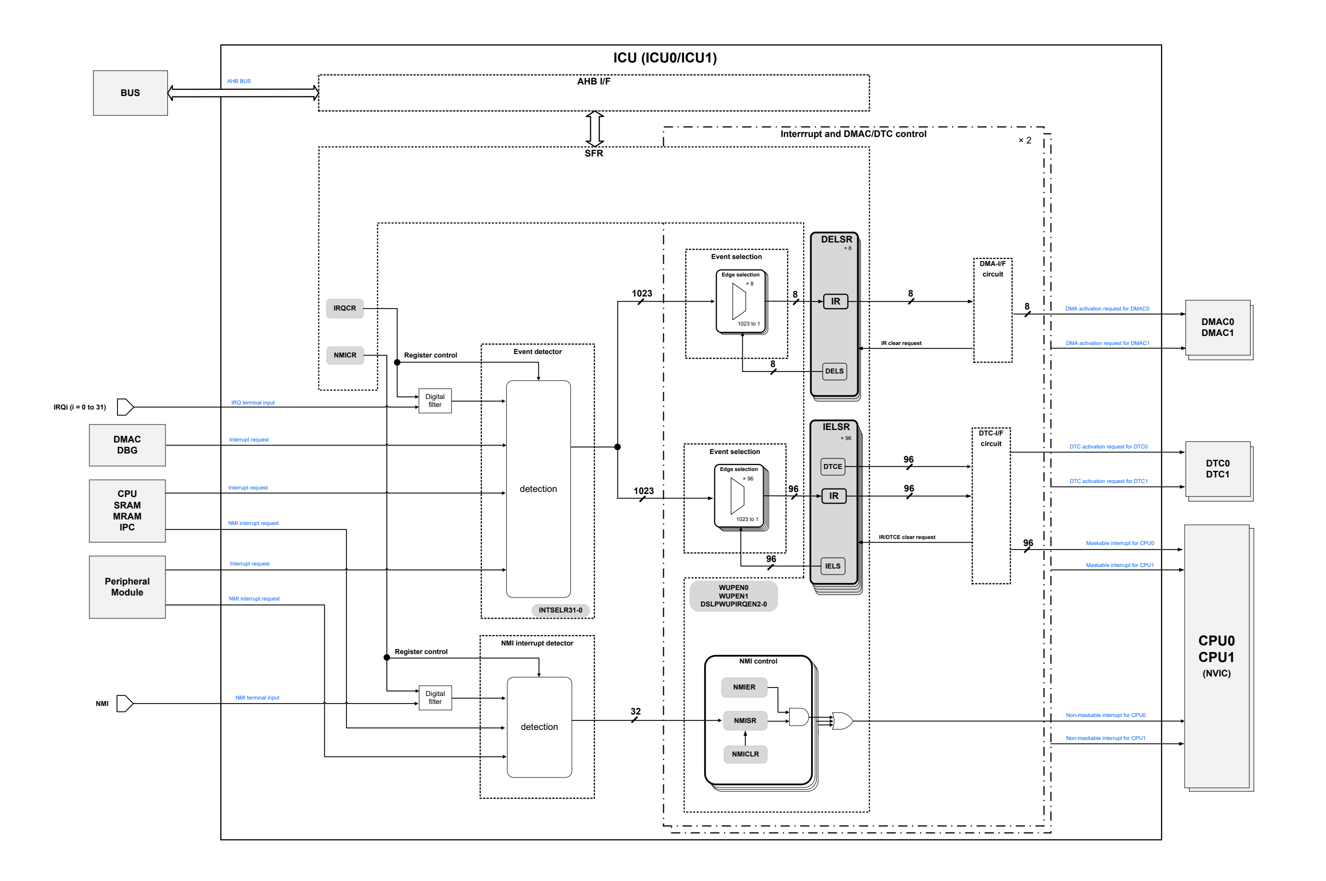


Figure 14.1 ICU block diagram

NMISR: Non-Maskable Interrupt Status Register

NMIER: Non-Maskable Interrupt Enable Register

NMICLR: Non-Maskable Interrupt Status Clear Register

NMICR: NMI Pin Interrupt Control Register

IRQCRi: IRQi Control Register (i = 0 to 31)

WUPEN0/1: Wake Up Interrupt Enable Register0/1

IELSRn: ICU Event Link Setting Register (n = 0 to 95)

DELSRm: DMAC Event Link Setting Register (m = 0 to 7)

IR: Interrupt Status Flag (IELSRn.IR, DELSRm.IR)

DTCE: DTC Activation Enable (IELSRn.DTCE)

DSLPWUPIRQENj: Deep Sleep Wake Up IRQ Enable Register (j = 0 to 2)

INTSELRp: Interrupt request select Register (p = 0 to 31)

Table 14.2 ICU I/O pins

Pin name	I/O	Description
NMI	Input	Non-maskable interrupt request pin
IRQi (i = 0 to IRQ31)	Input	External interrupt request pins

14.2 Register Descriptions

This section does not describe registers internal to Arm® NVIC.

For information on these registers, see the NVIC chapter of Arm® Processor Technical Reference Manual [1], [3].

ICU_BASE (base address of ICU) is 0x4000_C000(Secure) and 0x5000_C000(Non-secure)

COMMON_ICU_BASE (base address of COMMON_ICU) is 0x4000_6000(Secure) and 0x5000_6000(Non-secure)

CPSCU_BASE (base address of CPSCU) is 0x4000_8000(secure) and 0x5000_8000(non-secure).

The ICU0 and ICU1 registers have the same base address. ICU0 is dedicated to CPU0 and can only be accessed from CPU0. ICU1 is dedicated to CPU1 and can only be accessed from CPU1.

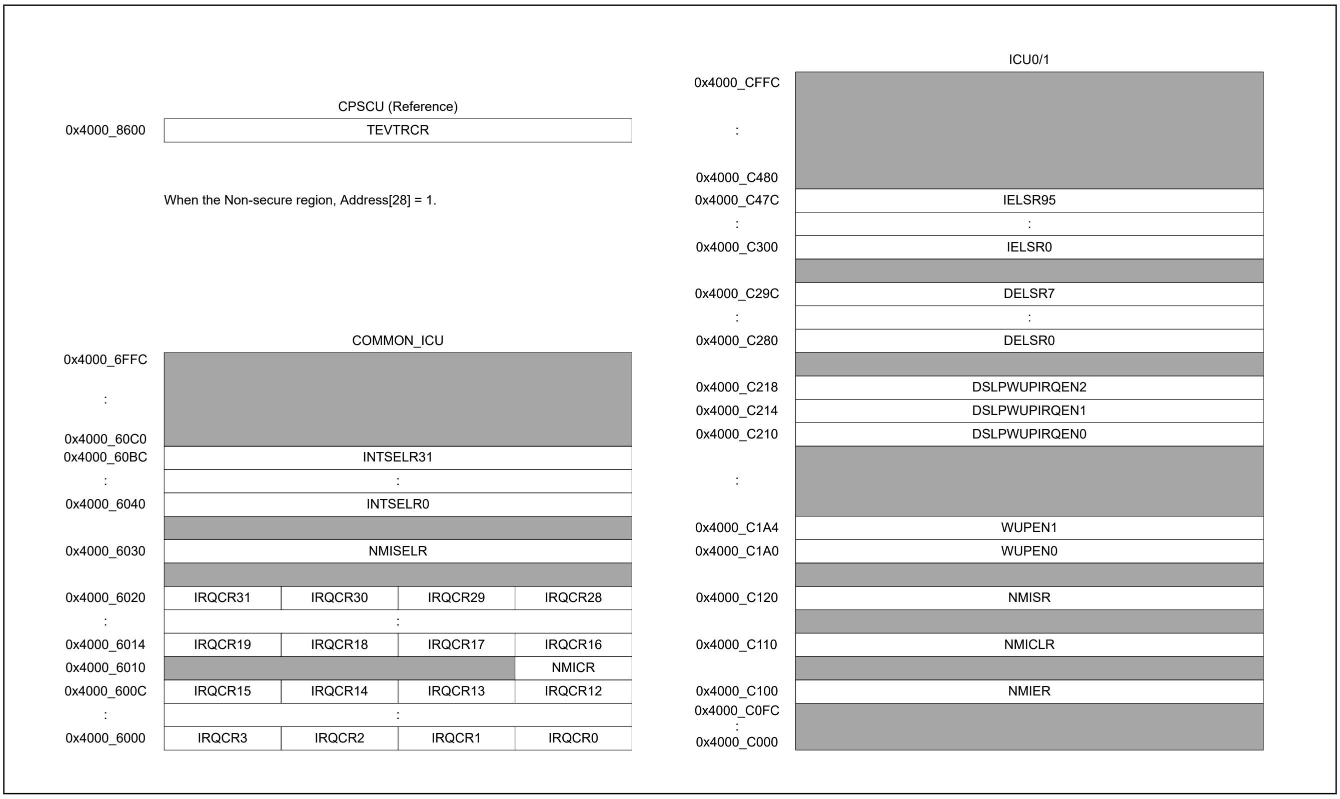


Figure 14.2 ICU Register Address Map

14.2.1 ICUSARA : Interrupt Controller Unit Security Attribution Register A

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x40

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SAIRQCR31	SAIRQCR30	SAIRQCR29	SAIRQCR28	SAIRQCR27	SAIRQCR26	SAIRQCR25	SAIRQCR24	SAIRQCR23	SAIRQCR22	SAIRQCR21	SAIRQCR20	SAIRQCR19	SAIRQCR18	SAIRQCR17	SAIRQCR16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SAIRQCR15	SAIRQCR14	SAIRQCR13	SAIRQCR12	SAIRQCR11	SAIRQCR10	SAIRQCR9	SAIRQCR8	SAIRQCR7	SAIRQCR6	SAIRQCR5	SAIRQCR4	SAIRQCR3	SAIRQCR2	SAIRQCR1	SAIRQCR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SAIRQCR31 to SAIRQCR0	Security attributes of registers for the IRQCR, WUPEN0, WUPEN1 registers 0: Secure 1: Non-Secure	R/W

Note: S-TYPE1, P-TYPE1
Note: This register is write-protected by PRCR_S.PRC4.

SAIRQCRn (n = 0 to 31)

Security attributes of registers for IRQCRn, WUPEN0, WUPEN1 registers. The target registers are as follows:

- ICU.IRQCR0 to ICU.IRQCR31 registers
- ICU0.WUPEN0[15:0]
- ICU0.WUPEN1[31:16]
- ICU1.WUPEN0[15:0]
- ICU1.WUPEN1[31:16]

14.2.2 ICUSARB : Interrupt Controller Unit Security Attribution Register B

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x44

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SANMI1	SANMI0	SANMI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SANMI	Security attributes of the NMICR register 0: Secure 1: Non-Secure	R/W
1	SANMI0	Security attributes of the ICU0.NMISR, ICU0.NMIER, ICU0.NMICLR registers 0: Secure 1: Non-Secure	R/W

Bit	Symbol	Function	R/W
2	SANMI1	Security attributes of the ICU1.NMISR, ICU1.NMIER, ICU1.NMICLR registers 0: Secure 1: Non-Secure	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE1, P-TYPE1
Note: This register is write-protected by PRCR_S.PRC4.

SANMI bit (Security attributes of the NMICR register)

Security attributes of registers for non-maskable interrupt. The target register is as follows:

- NMICR

SANMI0 bit (Security attributes of the ICU0.NMISR, ICU0.NMIER, ICU0.NMICLR registers)

Security attributes of registers for non-maskable interrupt. The target registers are as follows:

- ICU0. NMIER
- ICU0. NMICLR
- ICU0. NMISR

SANMI1 bit (Security attributes of the ICU1.NMISR, ICU1.NMIER, ICU1.NMICLR registers)

Security attributes of registers for non-maskable interrupt. The target registers are as follows:

- ICU1. NMIER
- ICU1. NMICLR
- ICU1. NMISR

The AIRCR.BFHFNMINS (bit 13) value in Application Interrupt and Reset Control Register of ARM CPU should be the same as the value of security attribution. The initial values of AIRCR.BFHFNMINS and the SANMI bit are different.

AIRCR.BFHFNMINS is secure and SANMI is non-secure. Polarity has the same meaning. Program these to match.

Note: Only one of Secure and Non-Secure can set security attribution for non-maskable interrupt related registers. If you programmed the Secure attribute as secure, it will always jump to the Secure interrupt handler. If you want to release any of the non-maskable interrupt sources to the non-secure user, prepare a function for executing a nonsecure program from the interrupt handler for Secure.

14.2.3 ICUSARE : Interrupt Controller Unit Security Attribution Register E

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x50

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SAIIC0WUP	SAAGT1CBWUP	SAAGT1CAWUP	SAAGT1UDWUP	SAUSBFS0WUP	SAUSBHSWUP	SARTCPRDWUP	SARTCALMWUP	—	—	—	SAVBATTWUP	SAPVD2WUP	SAPVD1WUP	—	SAIWDTWUP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
16	SAIWDTWUP	Security attributes of registers for WUPEN0.b16 0: Secure 1: Non-Secure	R/W
17	—	This bit is read as 0. The write value should be 0.	R/W
18	SAPVD1WUP	Security attributes of registers for WUPEN0.b18 0: Secure 1: Non-Secure	R/W
19	SAPVD2WUP	Security attributes of registers for WUPEN0.b19 0: Secure 1: Non-Secure	R/W
20	SAVBATTWUP	Security attributes of registers for WUPEN0.b20 0: Secure 1: Non-Secure	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
24	SARTCALMWUP	Security attributes of registers for WUPEN0.b24 0: Secure 1: Non-Secure	R/W
25	SARTCPRDWUP	Security attributes of registers for WUPEN0.b25 0: Secure 1: Non-Secure	R/W
26	SAUSBHSWUP	Security attributes of registers for WUPEN0.b26 0: Secure 1: Non-Secure	R/W
27	SAUSBFS0WUP	Security attributes of registers for WUPEN0.b27 0: Secure 1: Non-Secure	R/W
28	SAAGT1UDWUP	Security attributes of registers for WUPEN0.b28 0: Secure 1: Non-Secure	R/W
29	SAAGT1CAWUP	Security attributes of registers for WUPEN0.b29 0: Secure 1: Non-Secure	R/W
30	SAAGT1CBWUP	Security attributes of registers for WUPEN0.b30 0: Secure 1: Non-Secure	R/W
31	SAIIC0WUP	Security attributes of registers for WUPEN0.b31 0: Secure 1: Non-Secure	R/W

Note: S-TYPE1, P-TYPE1

Note: This register is write-protected by PRCR_S.PRC4.

14.2.4 ICUSARF : Interrupt Controller Unit Security Attribution Register F

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x54

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SAPDMWUP	SAULP1BWUP	SAULP1AWUP	SAULP1UWUP	SAI3CWUP	SAULP0BWUP	SAULP0AWUP	SAULP0UWUP	SASOSCWUP	—	—	—	SACOMPHS0WUP	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	SACOMPHS0WUP	Security attributes of registers for WUPEN1.b3 0: Secure 1: Non-Secure	R/W
6:4	—	These bits are read as 0. The write value should be 0.	R/W
7	SASOSCWUP	Security attributes of registers for WUPEN1.b7 0: Secure 1: Non-Secure	R/W
8	SAULP0UWUP	Security attributes of registers for WUPEN1.b8 0: Secure 1: Non-Secure	R/W
9	SAULP0AWUP	Security attributes of registers for WUPEN1.b9 0: Secure 1: Non-Secure	R/W
10	SAULP0BWUP	Security attributes of registers for WUPEN1.b10 0: Secure 1: Non-Secure	R/W
11	SAI3CWUP	Security attributes of registers for WUPEN1.b11 0: Secure 1: Non-Secure	R/W
12	SAULP1UWUP	Security attributes of registers for WUPEN1.b12 0: Secure 1: Non-Secure	R/W
13	SAULP1AWUP	Security attributes of registers for WUPEN1.b13 0: Secure 1: Non-Secure	R/W
14	SAULP1BWUP	Security attributes of registers for WUPEN1.b14 0: Secure 1: Non-Secure	R/W
15	SAPDMWUP	Security attributes of registers for WUPEN1.b15 0: Secure 1: Non-Secure	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE1, P-TYPE1
Note: This register is write-protected by PRCR_S.PRC4.

14.2.5 ICUSARG : Interrupt Controller Unit Security Attribution Register G

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x70

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SAIELSR31	SAIELSR30	SAIELSR29	SAIELSR28	SAIELSR27	SAIELSR26	SAIELSR25	SAIELSR24	SAIELSR23	SAIELSR22	SAIELSR21	SAIELSR20	SAIELSR19	SAIELSR18	SAIELSR17	SAIELSR16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SAIELSR15	SAIELSR14	SAIELSR13	SAIELSR12	SAIELSR11	SAIELSR10	SAIELSR9	SAIELSR8	SAIELSR7	SAIELSR6	SAIELSR5	SAIELSR4	SAIELSR3	SAIELSR2	SAIELSR1	SAIELSR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SAIELSR31 to SAIELSR0	Security attributes of registers for ICU0 event link setting0 0: Secure 1: Non-Secure	R/W

Note: S-TYPE1, P-TYPE1
Note: This register is write-protected by PRCR_S.PRC4.

SAIELSRn (n = 0 to 31)

Security attributes of registers for ICU0 event link setting0. The target registers are as follows:

- ICU0.IELSR0 to IELSR31
- ICU0.DSLPWUPIRQEN0

The Secure Attribute managed within the Arm® CPU NVIC must match the security attribution of ICU0.IELSEn (n = 0 to 31). NVIC internal registers are in NVIC_ITNS0[31:0]. Polarity has the same meaning. Program these to match.

14.2.6 ICUSARH : Interrupt Controller Unit Security Attribution Register H

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x74

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SAIELSR63	SAIELSR62	SAIELSR61	SAIELSR60	SAIELSR59	SAIELSR58	SAIELSR57	SAIELSR56	SAIELSR55	SAIELSR54	SAIELSR53	SAIELSR52	SAIELSR51	SAIELSR50	SAIELSR49	SAIELSR48
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SAIELSR47	SAIELSR46	SAIELSR45	SAIELSR44	SAIELSR43	SAIELSR42	SAIELSR41	SAIELSR40	SAIELSR39	SAIELSR38	SAIELSR37	SAIELSR36	SAIELSR35	SAIELSR34	SAIELSR33	SAIELSR32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SAIELSR32 to SAIELSR63	Security attributes of registers for ICU0event link setting1. 0: Secure 1: Non-Secure	R/W

Note: S-TYPE1, P-TYPE1
Note: This register is write-protected by PRCR_S.PRC4.

SAIELSRn (n = 32 to 63)

Security attributes of registers for ICU0 event link setting1. The target registers are as follows:

- ICU0.IELSR32 to IELSR63

- ICU0.DSLPWUPIRQEN1

The Secure Attribute managed within the Arm® CPU NVIC must match the security attribution of ICU0.IELSRn (n = 32 to 63). NVIC internal registers are in NVIC_ITNS1[31:0]. Polarity has the same meaning. Program these to match.

14.2.7 ICUSARI : Interrupt Controller Unit Security Attribution Register I

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x78

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SAIELSR95	SAIELSR94	SAIELSR93	SAIELSR92	SAIELSR91	SAIELSR90	SAIELSR89	SAIELSR88	SAIELSR87	SAIELSR86	SAIELSR85	SAIELSR84	SAIELSR83	SAIELSR82	SAIELSR81	SAIELSR80
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SAIELSR79	SAIELSR78	SAIELSR77	SAIELSR76	SAIELSR75	SAIELSR74	SAIELSR73	SAIELSR72	SAIELSR71	SAIELSR70	SAIELSR69	SAIELSR68	SAIELSR67	SAIELSR66	SAIELSR65	SAIELSR64
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SAIELSR64 to SAIELSR95	Security attributes of registers for ICU0 event link setting2. 0: Secure 1: Non-Secure	R/W

Note: S-TYPE1, P-TYPE1
Note: This register is write-protected by PRCR_S.PRC4.

SAIELSRn (n = 64 to 95)

Security attributes of registers for ICU0 event link setting2. The target registers are as follows:

- ICU0.IELSR64 to IELSR95
- ICU0.DSLPWUPIRQEN2

The Secure Attribute managed within the Arm® CPU NVIC must match the security attribution of ICU0.IELSRn (n = 64 to 95). NVIC internal registers are in NVIC_ITNS2[31:0]. Polarity has the same meaning. Program these to match.

14.2.8 ICUSARJ : Interrupt Controller Unit Security Attribution Register J

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x7C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SAIELSR31	SAIELSR30	SAIELSR29	SAIELSR28	SAIELSR27	SAIELSR26	SAIELSR25	SAIELSR24	SAIELSR23	SAIELSR22	SAIELSR21	SAIELSR20	SAIELSR19	SAIELSR18	SAIELSR17	SAIELSR16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SAIELSR15	SAIELSR14	SAIELSR13	SAIELSR12	SAIELSR11	SAIELSR10	SAIELSR9	SAIELSR8	SAIELSR7	SAIELSR6	SAIELSR5	SAIELSR4	SAIELSR3	SAIELSR2	SAIELSR1	SAIELSR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SAIELSR0 to SAIELSR31	Security attributes of registers for ICU1 event link setting0. 0: Secure 1: Non-Secure	R/W

Note: S-TYPE1, P-TYPE1

Note: This register is write-protected by PRCR_S.PRC4.

SAIELSRn (n = 0 to 31)

Security attributes of registers for ICU1 event link setting0. The target registers are as follows:

- ICU1.IELSR0 to IELSR31
- ICU1.DSLPWUPIRQEN0

The Secure Attribute managed within the Arm® CPU NVIC must match the security attribution of ICU1.IELSRn (n = 0 to 31). NVIC internal registers are in NVIC_ITNS0[31:0]. Polarity has the same meaning. Program these to match.

14.2.9 ICUSARK : Interrupt Controller Unit Security Attribution Register K

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x80

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SAIELSR63	SAIELSR62	SAIELSR61	SAIELSR60	SAIELSR59	SAIELSR58	SAIELSR57	SAIELSR56	SAIELSR55	SAIELSR54	SAIELSR53	SAIELSR52	SAIELSR51	SAIELSR50	SAIELSR49	SAIELSR48
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SAIELSR47	SAIELSR46	SAIELSR45	SAIELSR44	SAIELSR43	SAIELSR42	SAIELSR41	SAIELSR40	SAIELSR39	SAIELSR38	SAIELSR37	SAIELSR36	SAIELSR35	SAIELSR34	SAIELSR33	SAIELSR32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SAIELSR32 to SAIELSR63	Security attributes of registers for ICU1 event link setting1 0: Secure 1: Non-Secure	R/W

Note: S-TYPE1, P-TYPE1
Note: This register is write-protected by PRCR_S.PRC4.

SAIELSR (n = 32 to 63)

Security attributes of registers for ICU1 event link setting1. The target registers are as follows:

- ICU1.IELSR32 to IELSR63
- ICU1.DSLPWUPIRQEN1

The Secure Attribute managed within the Arm® CPU NVIC must match the security attribution of ICU1.IELSRn (n = 32 to 63). NVIC internal registers are in NVIC_ITNS1[31:0]. Polarity has the same meaning. Program these to match.

14.2.10 ICUSARL : Interrupt Controller Unit Security Attribution Register L

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x84

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SAIELSR95	SAIELSR94	SAIELSR93	SAIELSR92	SAIELSR91	SAIELSR90	SAIELSR89	SAIELSR88	SAIELSR87	SAIELSR86	SAIELSR85	SAIELSR84	SAIELSR83	SAIELSR82	SAIELSR81	SAIELSR80
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SAIELSR79	SAIELSR78	SAIELSR77	SAIELSR76	SAIELSR75	SAIELSR74	SAIELSR73	SAIELSR72	SAIELSR71	SAIELSR70	SAIELSR69	SAIELSR68	SAIELSR67	SAIELSR66	SAIELSR65	SAIELSR64
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SAIELSR64 to SAIELSR95	Security attributes of registers for ICU1 event link setting2. 0: Secure 1: Non-Secure	R/W

Note: S-TYPE1, P-TYPE1
Note: This register is write-protected by PRCR_S.PRC4.

SAIELSR (n = 64 to 95)

Security attributes of registers for ICU1 event link setting2. The target registers are as follows:

- ICU1.IELSR64 to IELSR95
- ICU1.DSLPWUPIRQEN2

The Secure Attribute managed within the ARM CPU NVIC must match the security attribution of ICU1.IELSRn (n = 64 to 95). NVIC internal registers are in NVIC_ITNS2[31:0]. Polarity has the same meaning. Program these to match.

14.2.11 TEVTRCR : Trusted Event Route Control Register

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x600

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	TEVTEICU1	TEVTEICU0	TEVTE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TEVTE	Trusted Event Route Control Register for ELC 0: Disable 1: Enable	R/W
1	TEVTEICU0	Trusted Event Route Control Register for ICU0 0: Disable 1: Enable	R/W
2	TEVTEICU1	Trusted Event Route Control Register for ICU1 0: Disable 1: Enable	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE1, P-TYPE1
Note: This register is write-protected by PRCR_S.PRC4

TEVTE bit (Trusted Event Route Control Register for ELC)

When TEVTE = 1, all ELS [9:0] bits of ELC.ELSRn (n = 0 to18) are allowed secure access write. Non-secure access write is protected.

- ELSRn.IELS[9:0] (n = 0 to 18)

TEVTEICU0 bit (Trusted Event Route Control Register for ICU0)

When TEVTEICU0 = 1, the IELS [9:0] bits of ICU0.IELSRn (n = 0 to 95), all DELS [9:0] bits of ICU0.DELSRm (m = 0 to 7) are allowed secure access write. Non-secure access write is protected. Additionally, when TEVTEICU0 = 1, if the Security Attribution of the target register (IELS [31:16] bits of ICU0.IELSRn (n = 0 to 95)) is non-secure, then secure access is not allowed. At this time, the upper level [31:16] cannot be read and write, but the response is OK, and no error occurs.

- ICU0.IELSRn.IELS[9:0] (n = 0 to 95)
- ICU0.DELSRm.DELS[9:0] (m = 0 to 7)

TEVTEICU1 bit (Trusted Event Route Control Register for ICU1)

When TEVTEICU1 = 1, the IELS [9:0] bits of ICU1.IELSRn (n = 0 to 95), the all DELS [9:0] bits of ICU1.DELSRm (m = 0 to 7) are allowed secure access write. Non-secure access write is protected. Additionally, when TEVTEICU1 = 1, if the Security Attribution of the target register (IELS [31:16] bits of ICU1.IELSRn (n = 0 to 95)) is non-secure, then secure access is not allowed. At this time, the upper level [31:16] cannot be read and write, but the response is OK, and no error occurs.

- ICU1.IELSRn.IELS[9:0] (n = 0 to 95)
- ICU1.DELSRm.DELS[9:0] (m = 0 to 7)

14.2.12 IRQCRi : IRQ Control Register (i = 0 to 31)

Base address: ICU_COMMON = 0x4000_6000
ICU_COMMON_NS = 0x5000_6000

Offset address: 0x000 + 0x01 × i (i = 0 to 15)
0x004 + 0x01 × i (i = 16 to 31)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	FLTEN	—	FCLKSEL[1:0]		—	—	IRQMD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	IRQMD[1:0]	IRQi Detection Sense Select 0 0: Falling edge 0 1: Rising edge 1 0: Rising and falling edges 1 1: Low level	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
5:4	FCLKSEL[1:0]	IRQi Digital Filter Sampling Clock Select 0 0: PCLKB 0 1: PCLKB/8 1 0: PCLKB/32 1 1: PCLKB/64	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	FLTEN	IRQi Digital Filter Enable 0: Digital filter is disabled 1: Digital filter is enabled.	R/W

Note: S-TYPE3, P-TYPE2

IRQCRi register changes must satisfy the following conditions:

1. For a CPU interrupt or DTC trigger:
   Change the IRQCRi register value before setting the target IELSRn register (n = 0 to 95).
   The register value should be changed only when the value of the target IELSRn register is 0x0000.
2. For a DMAC trigger:
   Change the IRQCRi register value before setting the target DELSRm register (m = 0 to 7).
   The register value should be changed only when the value of the target DELSRm register is 0x0000.
3. For a wakeup enable signal:
   Change the IRQCRi register setting before setting the target WUPEN0.IRQWUPEN[n] (n = 0 to 15) or WUPEN1.IRQWUPEN[m] (m = 31 to 16). You can only change the register values when the target WUPEN0.IRQWUPEN[n] or WUPEN1.IRQWUPEN[m] is 0.

IRQMD[1:0] bits (IRQi Detection Sense Select)

The IRQMD[1:0] bits set the detection sensing method for the IRQi external pin interrupt sources. For setting method when using external pin interrupt, see section 14.5.7. External Pin Interrupts.

FCLKSEL[1:0] bits (IRQi Digital Filter Sampling Clock Select)

The FCLKSEL[1:0] bits select the digital filter sampling clock for the IRQi external pin interrupt request pins, selectable to:

- PCLKB (every cycle)
- PCLKB/8 (once every 8 cycles)
- PCLKB/32 (once every 32 cycles)
- PCLKB/64 (once every 64 cycles)

For details of the digital filter, see section 14.5.6. Digital Filter.

FLTEN bit (IRQi Digital Filter Enable)

The FLTEN bit enables the digital filter used for the IRQi external pin interrupt sources. The digital filter is enabled when the IRQCRi.FLTEN bit is 1 and disabled when the IRQCRi.FLTEN bit is 0. The IRQi pin level is sampled at the clock cycle specified in the IRQCRi.FCLKSEL[1:0] bits. When the sampled level matches three times, the output level from the digital filter changes. For details of the digital filter, see section 14.5.6. Digital Filter.

14.2.13 NMISR : Non-Maskable Interrupt Status Register

Base address: ICU = 0x4000_C000
ICU_NS = 0x5000_C000

Offset address: 0x120

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	LUST	LMST	CMST	BUSST	—	—	—	—	NMIST	OSTST	SOSTST	—	PVD2ST	PVD1ST	WDTST	IWDTST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IWDTST	IWDT Underflow/Refresh Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
1	WDTST	WDT Underflow/Refresh Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
2	PVD1ST	Voltage Monitor 1 Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
3	PVD2ST	Voltage Monitor 2 Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
4	—	These bits are read as 0.	R
5	SOSTST	Sub Oscillation Stop Detection Interrupt Status Flag 0: Interrupt not requested for sub oscillation stop 1: Interrupt requested for sub oscillation stop	R
6	OSTST	Main Clock Oscillation Stop Detection Interrupt Status Flag 0: Interrupt not requested for main clock oscillation stop 1: Interrupt requested for main clock oscillation stop	R

Bit	Symbol	Function	R/W
7	NMIST	NMI Pin Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
11:8	—	These bits are read as 0.	R
12	BUSST	Bus Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
13	CMST	Common Memory Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
14	LMST	Local Memory Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested.	R
15	LUST	LockUp Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
16	FPUEXCST	FPU Exception Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
17	MRCRDST	MRAM MRC read Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
18	MRERDST	MRAM MRE read Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
19	—	This bit is read as 0.	R
20	IPCST	IPC NMI CPU mutual Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
31:21	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE2

The NMISR register monitors the status of non-maskable interrupt sources. Writes to the NMISR register are ignored.

NMISR is set with a signal that is valid as a non-maskable interrupt supplied to CPU0/CPU1 by the setting judgment of NMIER. For details of the judgment by the NMIER setting, see section 14.8.2. Trusted Interrupt Management.

Before the end of the non-maskable interrupt handler, check that all of the bits in this register are set to 0 to confirm that no other NMI requests are generated during handler processing.

IWDTST flag (IWDT Underflow/Refresh Error Interrupt Status Flag)

The IWDTST flag indicates an IWDT underflow/refresh error interrupt request. It is read-only and cleared by the NMICLR.IWDTCLR bit. The IWDT interrupt is only valid for primary CPU. See section 14.8.2. Trusted Interrupt Management.

[Setting condition]

When the IWDT underflow/refresh error interrupt is generated and this interrupt source is enabled.

[Clearing condition]

When 1 is written to the NMICLR.IWDTCLR bit.

WDTST flag (WDT Underflow/Refresh Error Interrupt Status Flag)

The WDTST flag indicates a WDT underflow/refresh error interrupt request. It is read-only and cleared by the NMICLR.WDTCLR bit. Accepts WDT interrupt for same CPUs. See section 14.8.2. Trusted Interrupt Management.

[Setting condition]

When the WDT underflow/refresh error interrupt is generated.

[Clearing condition]

When 1 is written to the NMICLR.WDTCLR bit.

PVD1ST flag (Voltage Monitor 1 Interrupt Status Flag)

The PVD1ST flag indicates a request for voltage monitor 1 interrupt. It is read-only and cleared by the NMICLR.PVD1CLR bit. The voltage monitor 1 interrupt is valid only when bit 2 of the NMIER register is set to "1" and supply to the target CPU is selected. See section 14.8.2. Trusted Interrupt Management.

[Setting condition]

When the voltage monitor 1 interrupt is generated, and this interrupt source is enabled.

[Clearing condition]

When 1 is written to the NMICLR.PVD1CLR bit.

PVD2ST flag (Voltage Monitor 2 Interrupt Status Flag)

The PVD2ST flag indicates a request for voltage monitor 2 interrupt. It is read-only and cleared by the NMICLR.PVD2CLR bit. The voltage monitor 2 interrupt is valid only when bit 3 of the NMIER register is set to "1" and supply to the target CPU is selected. See section 14.8.2. Trusted Interrupt Management.

[Setting condition]

When the voltage monitor 2 interrupt is generated, and this interrupt source is enabled.

[Clearing condition]

When 1 is written to the NMICLR.PVD2CLR bit.

SOSTST flag (Sub Oscillation Stop Detection Interrupt Status Flag)

The SOSTST flag indicates a sub oscillation stop detection interrupt request. It is read-only and cleared by the NMICLR.SOSTCLR bit. The sub oscillation stop detection interrupt is valid only when bit 5 of the NMIER register is set to "1" and supply to the target CPU is selected. See section 14.8.2. Trusted Interrupt Management.

[Setting condition]

When the sub oscillation stops detection, interrupt is generated.

[Clearing condition]

When 1 is written to the NMICLR.SOSTCLR bit.

OSTST flag (Main Clock Oscillation Stop Detection Interrupt Status Flag)

The OSTST flag indicates an oscillation stop detection interrupt request. It is read-only and cleared by the NMICLR.OSTCLR bit. The oscillation stop detection interrupt is valid only when bit 6 of the NMIER register is set to "1" and supply to the target CPU is selected. See section 14.8.2. Trusted Interrupt Management.

[Setting condition]

When the oscillation stops detection, interrupt is generated.

[Clearing condition]

When 1 is written to the NMICLR.OSTCLR bit.

NMIST flag (NMI Pin Interrupt Status Flag)

The NMIST flag indicates a NMI pin interrupt request. It is read-only and cleared by the NMICLR.NMICLR bit. The NMI pin interrupt is valid only when bit 7 of the NMIER register is set to "1" and supply to the target CPU is selected. See section 14.8.2. Trusted Interrupt Management.

[Setting condition]

When an edge specified by the NMICR.NMIMD bit is input to the NMI pin.

[Clearing condition]

When 1 is written to the NMICLR.NMICLR bit.

BUSST flag (Bus Error Interrupt Status Flag)

The BUSST flag indicates a request for bus error interrupt as a NMI trigger. It is read-only and cleared by the NMICLR.BUSCLR bit. Bus error interrupt includes MPU and TZF error. The bus error interrupt is valid only when bit 12 of the NMIER register is set to "1" and supply to the target CPU is selected. See section 14.8.2. Trusted Interrupt Management.

[Setting condition]

When the bus error detection interrupt is generated.

[Clearing condition]

When 1 is written to the NMICLR.BUSCLR bit.

Be sure to clear the error status of the request source before clearing. If CPU does not clear the error status of the request source, the NMI status is set again even if this status is cleared, and an NMI request is issued to the CPU. If the CPU returns from the NMI handler, the CPU jumps back to the NMI handler.

In the case of level detection, use the following steps to clear the Status flag.

(1)Negate the level of an input factor.

(2)Perform a peripheral read access once and make sure the level interrupt is cleared.

(3)Clear the status flag by NMICLR.BUSCLR.

CMST flag (Common Memory Error Interrupt Status Flag)

The CMST flag indicates common memory error. It is read-only and cleared by the NMICLR.CMCLR bit.

Common memory error is SRAM ECC error. The common memory error interrupt is valid only when bit 13 of the NMIER register is set to "1" and supply to the target CPU is selected. See section 14.8.2. Trusted Interrupt Management.

[Setting condition]

When the common memory error detection interrupt is generated.

[Clearing condition]

When 1 is written to the NMICLR.CMCLR bit.

Be sure to clear the error status of the request source before clearing. If CPU does not clear the error status of the request source, the NMI status is set again even if this status is cleared, and an NMI request is issued to the CPU. If the CPU returns from the NMI handler, the CPU jumps back to the NMI handler.

In the case of level detection, use the following steps to clear the Status flag.

(1)Negate the level of an input factor.

(2)Perform a peripheral read access once and make sure the level interrupt is cleared.

(3)Clear the status flag by NMICLR.CMCLR.

LMST flag (Local Memory Error Interrupt Status Flag)

The LMST flag indicates local memory error. It is read-only and cleared by the NMICLR.LMCLR bit.

Local memory error includes CPU1 cache and TCM ECC error. The local memory error interrupt is valid only when bit 14 of the NMIER register is set to "1" and supply to the target CPU is selected. See section 14.8.2. Trusted Interrupt Management.

[Setting condition]

When the local memory error detection interrupt is generated.

[Clearing condition]

When 1 is written to the NMICLR.LMCLR bit.

Be sure to clear the error status of the request source before clearing. If CPU does not clear the error status of the request source, the NMI status will be set again even if this status is cleared, and an NMI request will be issued to the CPU. If the CPU returns from the NMI handler, the CPU will jump back to the NMI handler.

In the case of level detection, clear of the Status flag should follow the steps below.

(1)Negate the level of an input factor.

(2)Perform a peripheral read access once and make sure the level interrupt is cleared.

(3)Clear the status flag by NMICLR.LUSTCLR.

LUST flag (LockUp Error Interrupt Status Flag)

The LUST flag indicates la lockup. It is read-only and cleared by the NMICLR. LUCLR bit. Accepts Lockup error interrupt for same CPUs. See section 14.8.2. Trusted Interrupt Management.

[Setting condition]

When the local memory error detection interrupt is generated.

[Clearing condition]

When 1 is written to the NMICLR.LMCLR bit.

Be sure to clear the error status of the request source before clearing. If CPU does not clear the error status of the request source, the NMI status will be set again even if this status is cleared, and an NMI request is issued to the CPU. If the CPU returns from the NMI handler, the CPU will jump back to the NMI handler.

In the case of level detection, use the following steps to clear the Status flag.

(1)Negate the level of an input factor.

(2)Perform a peripheral read access once and make sure the level interrupt is cleared.

(3)Clear the status flag by NMICLR.LUCLR.

FPUEXCST flag (FPU Exception Interrupt Status Flag)

The FPUEXCST flag indicates FPU Exception interrupt. It is read-only and cleared by the NMICLR.FPUEXCCLR bit. Accepts FPU Exception interrupt for same CPU. See section 14.8.2. Trusted Interrupt Management.

[Setting condition]

When the FPU Exception detection interrupt is generated

[Clearing condition]

When 1 is written to the NMICLR.FPUEXCCLR bit

Be sure to clear the error status of the request source before clearing. If CPU does not clear the error status of the request source, the NMI status will be set again even if this status is cleared, and an NMI request will be issued to the CPU. If the CPU returns from the NMI handler, the CPU will jump back to the NMI handler.

In the case of level detection, clear of the Status flag should follow the steps below.

(1)Negate the level of an input factor.

(2)Perform a peripheral read access once and make sure the level interrupt is cleared.

(3)Clear the status flag by NMICLR.FPUEXCCLR.

MRCRDST flag (MRAM MRC read Error Interrupt Status Flag)

The MRCRDST flag indicates MRAM MRC read error interrupt. It is read-only and cleared by the NMICLR.MRCRDCLR bit. The MRAM MRC read error interrupt is valid only when bit 17 of the NMIER register is set to "1" and supply to the target CPU is selected. See section 14.8.2. Trusted Interrupt Management.

[Setting condition]

When the MRAM MRC read error detection interrupt is generated

[Clearing condition]

When 1 is written to the NMICLR.MRCRDSCLR bit

Be sure to clear the error status of the request source before clearing. If CPU does not clear the error status of the request source, the NMI status will be set again even if this status is cleared, and an NMI request will be issued to the CPU. If the CPU returns from the NMI handler, the CPU will jump back to the NMI handler.

In the case of level detection, clear of the Status flag should follow the steps below.

(1)Negate the level of an input factor.

(2)Perform a peripheral read access once and make sure the level interrupt is cleared.

(3)Clear the status flag by NMICLR.MRCRDCLR.

MRERDST flag (MRAM MRE read Error Interrupt Status Flag)

The MRERDST flag indicates MRAM MRE read error interrupt. It is read-only and cleared by the NMICLR.MRERDCLR bit. The MRAM MRE read error interrupt is valid only when bit 18 of the NMIER register is set to "1" and supply to the target CPU is selected. See section 14.8.2. Trusted Interrupt Management.

[Setting condition]

When the MRAM MRE read error detection interrupt is generated

[Clearing condition]

When 1 is written to the NMICLR.MRERDCLR bit

Be sure to clear the error status of the request source before clearing. If CPU does not clear the error status of the request source, the NMI status will be set again even if this status is cleared, and an NMI request will be issued to the CPU. If the CPU returns from the NMI handler, the CPU will jump back to the NMI handler.

In the case of level detection, clear of the Status flag should follow the steps below.

(1)Negate the level of an input factor.

(2)Perform a peripheral read access once and make sure the level interrupt is cleared.

(3)Clear the status flag by NMICLR.MRERDCLR.

IPCST flag (IPC NMI CPU mutual Interrupt Status Flag)

The IPCST flag indicates a request for IPC NMI CPU mutual interrupt as a NMI trigger. It is read-only and cleared by the NMICLR.IPCCLR bit. Accepts IPC interrupt for other CPUs. See section 14.8.2. Trusted Interrupt Management.

[Setting condition]

When the IPC NMI CPU mutual interrupt is generated.

[Clearing condition]

When 1 is written to the NMICLR.IPCCLR bit.

Be sure to clear the error status of the request source before clearing. If CPU does not clear the error status of the request source, the NMI status is set again even if this status is cleared, and an NMI request is issued to the CPU. If the CPU returns from the NMI handler, the CPU jumps back to the NMI handler.

In the case of level detection, use the following steps to clear the Status flag.

(1)Negate the level of an input factor.

(2)Perform a peripheral read access once and make sure the level interrupt is cleared.

(3)Clear the status flag by NMICLR.IPCCLR.

14.2.14 NMIER : Non-Maskable Interrupt Enable Register

Base address: ICU = 0x4000_C000
ICU_NS = 0x5000_C000

Offset address: 0x100

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	LUEN	LMEN	CMEN	BUSEN	—	—	—	—	NMIEN	OSTEN	SOSTEN	—	PVD2EN	PVD1EN	WDTEN	IWDTEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IWDTEN	IWDT Underflow/Refresh Error Interrupt Enable 0: Disabled 1: Enabled	R/W*1 *2
1	WDTEN	WDT Underflow/Refresh Error Interrupt Enable 0: Disabled 1: Enabled	R/W*1 *2
2	PVD1EN	Voltage monitor 1 Interrupt Enable 0: Disabled 1: Enabled	R/W*1 *2
3	PVD2EN	Voltage monitor 2 Interrupt Enable 0: Disabled 1: Enabled	R/W*1 *2
4	—	This bit is read as 0. The write value should be 0.	R/W
5	SOSTEN	Sub Oscillation Stop Detection Interrupt Enable 0: Disabled 1: Enabled	R/W
6	OSTEN	Oscillation Stop Detection Interrupt Enable 0: Disabled 1: Enabled	R/W*1 *2
7	NMIEN	NMI Pin Interrupt Enable 0: Disabled 1: Enabled	R/W*1
11:8	—	These bits are read as 0. The write value should be 0.	R/W
12	BUSEN	Bus Error Interrupt Enable 0: Disabled 1: Enabled	R/W*1
13	CMEN	Common Memory Error Interrupt Enable 0: Disabled 1: Enabled	R/W*1
14	LMEN	Local Memory Error Interrupt Enable 0: Disabled 1: Enabled	R/W
15	LUEN	LockUp Error Interrupt Enable 0: Disabled 1: Enabled	R/W*1
16	FPUEXCEN	FPU Exception Interrupt Enable 0: Disabled 1: Enabled	R/W
17	MRCRDEN	MRAM MRC read Error Interrupt Enable 0: Disabled 1: Enabled	R/W
18	MRERDEN	MRAM MRE read Error Interrupt Enable 0: Disabled 1: Enabled	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
20	IPCEN	IPC NMI CPU mutual Interrupt Enable 0: Disabled 1: Enabled	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE2
Note 1. You can write 1 to this bit only once after reset. Subsequent write accesses are invalid. Writing 0 to this bit is invalid.
Note 2. Do not write 1 to this bit when the source is used as an interrupt signal.

For the NMIER bit, set only one of CPU0 and CPU1 to "1". Setting "1" to the NMIER bit of both CPU0 and CPU1 is prohibited.

IWDTEN bit (IWDT Underflow/Refresh Error Interrupt Enable)

The IWDTEN bit enables IWDT underflow/refresh error interrupt as a NMI trigger. This bit is the enable bit to control the use of the IWDT underflow/refresh error interrupt as a Deep Sleep/Software Standby return factor.

WDTEN bit (WDT Underflow/Refresh Error Interrupt Enable)

The WDTEN bit enables WDT underflow/refresh error interrupt as a NMI trigger. This bit is the enable bit to control the use of the WDT underflow/refresh error interrupt as a Deep Sleep return factor.

PVD1EN bit (Voltage monitor 1 Interrupt Enable)

The PVD1EN bit enables voltage monitor 1 interrupt as a NMI trigger.

PVD2EN bit (Voltage monitor 2 Interrupt Enable)

The PVD2EN bit enables voltage monitor 2 interrupt as a NMI trigger. This bit is the enable bit to control the use of the voltage monitor 2 interrupt as a Deep Sleep/Software Standby return factor.

SOSTEN bit (Sub Oscillation Stop Detection Interrupt Enable)

The OSTEN bit enables main clock oscillation stop detection interrupt as a NMI trigger. This bit is the enable bit to control the use of the sub oscillation stop detection interrupt as a Deep Sleep/Software Standby return factor.

OSTEN bit (Oscillation Stop Detection Interrupt Enable)

The OSTEN bit enables main clock oscillation stop detection interrupt as a NMI trigger. This bit is the enable bit to control the use of the main oscillation stop detection interrupt as a Deep Sleep/Software Standby return factor.

NMIEN bit (NMI Pin Interrupt Enable)

The NMIEN bit enables NMI pin interrupt as a NMI trigger. This bit is the enable bit to control the use of the NMI pin interrupt as a Deep Sleep/Software Standby return factor.

BUSEN bit (Bus Error Interrupt Enable)

The BUSEN bit enables bus error interrupt as a NMI trigger. This bit is the enable bit to control the use of the bus error interrupt as a Deep Sleep return factor. Bus error includes MPU and TZF errors.

CMEN bit (Common Memory Error Interrupt Enable)

The CMEN bit enables common memory error interrupt as a NMI trigger. This bit is the enable bit to control the use of the bus error interrupt as a Deep Sleep return factor. Common memory error is SRAM ECC error.

LMEN bit (Local Memory Error Interrupt Enable)

LMEN bit enables local memory error interrupt as a NMI trigger. This bit is the enable bit to control the use of the local memory error interrupt as a Deep Sleep return factor. Local memory error interrupt includes Cache and TCM ECC Error.

LUEN bit (LockUp Error Interrupt Enable)

LUEN bit enables LockUp error interrupt as a NMI trigger.

FPUEXCEN bit (FPU Exception Interrupt Enable)

FPUEXCEN bit enables FPU Exception Interrupt as a NMI trigger.

MRCRDEN bit (MRAM MRC read Error Interrupt Enable)

MRCRDEN bit enables MRAM MRC read error Interrupt as a NMI trigger. This bit is the enable bit to control the use of the MRAM MRC read error interrupt as a Deep Sleep return factor.

MRERDEN bit (MRAM MRE read Error Interrupt Enable)

MRERDEN bit enables MRAM MRE read error Interrupt as a NMI trigger. This bit is the enable bit to control the use of the MRAM MRE read error interrupt as a Deep Sleep return factor.

IPCEN bit (IPC NMI CPU mutual Interrupt Enable)

IPCEN bit enables IPC NMI CPU mutual interrupt as a NMI trigger. This bit is the enable bit to control the use of the IPC NMI CPU mutual interrupt as a Deep Sleep return factor.

14.2.15 NMICLR : Non-Maskable Interrupt Status Clear Register

Base address: ICU = 0x4000_C000
ICU_NS = 0x5000_C000

Offset address: 0x110

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	LUCLR	LMCLR	CMCLR	BUSCLR	—	—	—	—	NMICLR	OSTCLR	SOSTCLR	—	PVD2CLR	PVD1CLR	WDTCLR	IWDTCLR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IWDTCLR	IWDT Underflow/Refresh Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.IWDTST flag	R/W*1
1	WDTCLR	WDT Underflow/Refresh Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.WDTST flag	R/W*1
2	PVD1CLR	Voltage Monitor 1 Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.PVD1ST flag	R/W*1
3	PVD2CLR	Voltage Monitor 2 Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.PVD2ST flag.	R/W*1
4	—	This bit is read as 0. The write value should be 0.	R/W
5	SOSTCLR	Oscillation Stop Detection Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.OSTST flag	R/W*1
6	OSTCLR	Oscillation Stop Detection Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.OSTST flag	R/W*1
7	NMICLR	NMI Pin Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.NMIST flag	R/W*1
11:8	—	These bits are read as 0. The write value should be 0.	R/W
12	BUSCLR	Bus Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.BUSST flag	R/W*1
13	CMCLR	Common Memory Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.CMST flag	R/W*1
14	LMCLR	Local Memory Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.LMST flag	R/W*1
15	LUCLR	LockUp Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.LUST flag	R/W*1
16	FPUEXCCLR	FPU Exception Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.FPUEXCST flag	R/W*1

Bit	Symbol	Function	R/W
17	MRCRDCLR	MRAM MRC read Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.MRCRDST flag	R/W*1
18	MRERDCLR	MRAM MRE read Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.MRERDST flag	R/W*1
19	—	This bit is read as 0. The write value should be 0.	R/W
20	IPCCLR	IPC NMI CPU mutual Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.IPCST flag	R/W*1
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE2
Note: There may be a difference in processing speed between the CPU and ICU, and the CPU may exit the interrupt handler before clearing the NMISR. Then the CPU will accidentally jump to the NMI handler again. To avoid this, be sure to read NMISR before exiting the NMI handler and make sure that NMISR is cleared before exiting the NMI handler.
Note 1. Only write 1 to this bit.

IWDTCLR bit (IWDT Underflow/Refresh Error Interrupt Status Flag Clear)

Writing 1 to the IWDTCLR bit clears the NMISR.IWDTST flag. This bit is read as 0.

WDTCLR bit (WDT Underflow/Refresh Error Interrupt Status Flag Clear)

Writing 1 to the WDTCLR bit clears the NMISR.WDTST flag. This bit is read as 0.

PVD1CLR bit (Voltage Monitor 1 Interrupt Status Flag Clear)

Writing 1 to the PVD1CLR bit clears the NMISR.PVD1ST flag. This bit is read as 0.

PVD2CLR bit (Voltage Monitor 2 Interrupt Status Flag Clear)

Writing 1 to the PVD2CLR bit clears the NMISR.PVD2ST flag. This bit is read as 0.

SOSTCLR bit (Oscillation Stop Detection Interrupt Status Flag Clear)

Writing 1 to the SOSTCLR bit clears the NMISR.SOSTST flag. This bit is read as 0.

OSTCLR bit (Oscillation Stop Detection Interrupt Status Flag Clear)

Writing 1 to the OSTCLR bit clears the NMISR.OSTST flag. This bit is read as 0.

NMICLR bit (NMI Pin Interrupt Status Flag Clear)

Writing 1 to the NMICLR bit clears the NMISR.NMIST flag. This bit is read as 0.

BUSCLR bit (Bus Error Interrupt Status Flag Clear)

Writing 1 to the BUSCLR bit clears the NMISR.BUSST flag. This bit is read as 0.

CMCLR bit (Common Memory Error Interrupt Status Flag Clear)

Writing 1 to the CMCLR bit clears the NMISR.CMST flag. This bit is read as 0.

LMCLR bits (Local Memory Error Interrupt Status Flag Clear)

Writing 1 to the LMCLR bit clears the NMISR.LMST flag. This bit is read as 0.

LUCLR bit (LockUp Error Interrupt Status Flag Clear)

Writing 1 to the LUCLR bit clears the NMISR.LUST flag. This bit is read as 0.

FPUEXCCLR bits (FPU Exception Interrupt Status Flag Clear)

Writing 1 to the FPUEXCCLR bit clears the NMISR.FPUEXCST flag. This bit is read as 0.

MRCRDCLR bits (MRAM MRC read Error Interrupt Status Flag Clear)

Writing 1 to the MRCRDCLR bit clears the NMISR.MRCRDST flag. This bit is read as 0.

MRERDCLR bits (MRAM MRE read Error Interrupt Status Flag Clear)

Writing 1 to the MRERDCLR bit clears the NMISR.MRERDST flag. This bit is read as 0

IPCCLR bits (IPC NMI CPU mutual Interrupt Status Flag Clear)

Writing 1 to the IPCCLR bit clears the NMISR.IPCST flag. This bit is read as 0.

14.2.16 NMICR : NMI Pin Interrupt Control Register

Base address: ICU_COMMON = 0x4000_6000
ICU_COMMON_NS = 0x5000_6000

Offset address: 0x10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	NFLTEN	—	NFCLKSEL[1:0]		—	—	—	NMIMD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NMIMD	NMI Detection Set 0: Falling edge 1: Rising edge	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
5:4	NFCLKSEL[1:0]	NMI Digital Filter Sampling Clock Select 0 0: PCLKB 0 1: PCLKB/8 1 0: PCLKB/32 1 1: PCLKB/64	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	NFLTEN	NMI Digital Filter Enable 0: Disabled 1: Enabled	R/W

Note: S-TYPE3, P-TYPE2

Change the NMICR register settings before enabling NMI pin interrupts, that is, before setting the NMIER.NMIEN bit to 1.

NMIMD bit (NMI Detection Set)

The NMIMD bit selects the detection sensing method for the NMI pin interrupts.

NFCLKSEL[1:0] bits (NMI Digital Filter Sampling Clock Select)

The NFCLKSEL[1:0] bits select the digital filter sampling clock for the NMI pin interrupts, selectable to:

- PCLKB (every cycle)
- PCLKB/8 (once every 8 cycles)
- PCLKB/32 (once every 32 cycles)
- PCLKB/64 (once every 64 cycles)

For details of the digital filter, see section 14.5.6. Digital Filter.

NFLTEN bit (NMI Digital Filter Enable)

The NFLTEN bit enables the digital filter used for NMI pin interrupts. The filter is enabled when NFLTEN bit is 1, and disabled when the NFLTEN bit is 0. The NMI pin level is sampled at the sampling clock cycle specified in NMIFLTC.NFCLKSEL[1:0]. When the sampled level matches three times, the output level from the digital filter changes. For details of the digital filter, see section 14.5.6. Digital Filter.

14.2.17 IELSRn : Interrupt Controller Unit Event Link Setting Register n (n = 0 to 95)

Base address: ICU = 0x4000_C000
ICU_NS = 0x5000_C000

Offset address: 0x300 + 0x4 × n

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

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

Bit	Symbol	Function	R/W
9:0	IELS[9:0]	ICU Event Link Select 0x00: Disable interrupts to the associated NVIC or DTC module Others: Event signal number to be linked. For details, see section 14.3.2. Event Number.	R/W*1
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	IR	Interrupt Status Flag 0: No interrupt request generated. 1: An interrupt request is generated.	R/W*2
23:17	—	These bits are read as 0. The write value should be 0.	R/W
24	DTCE	DTC Activation Enable 0: DTC activation is disabled. 1: DTC activation is enabled.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note 1. The [15:0] bits in this register can only be accessed in halfword or word access. Byte access is ignored.
Note 2. Writing 1 to the IR flag is prohibited.

This register has different secure access permission depending on the setting of Trusted Event Route Control Register (TEVTRCR).

If the security attribution is configured as secure,

- Secure access is allowed
- Non-secure write access is ignored and non-secure read access is read as 0. TrustZone access error is generated.

If the security attribution is configured as non-secure and the trusted event route is disabled:

- Secure write access is ignored and secure read access is read as 0. TrustZone access error is generated.
- Non-secure access is allowed.

If the security attribution is configured as non-secure and the trusted event route is enabled:

- Secure access to IELS bit is allowed
- Non-secure write access to IELS bit is ignored and non-secure read access is allowed
- Secure write access to other bits is ignored and secure read access to other bits is read as 0
- Non-secure access to other bits is allowed
- TrustZone access error is not generated

The IELSRn register selects the interrupt source used by the NVIC. For details, see Table 14.3. IELSRn, where n = 0 to 95, corresponds to the NVIC interrupt input source numbers 0 to 95.

IELS[9:0] bits (ICU Event Link Select)

The IELS[9:0] bits link an event signal to the associated NVIC or DTC module. For details, see section 14.3. Vector Table. All IELS[9:0] bits must be written at the same time.

[Setting condition]

When value is written to these bits.

[Clearing conditions]

When 0 is written to these bits.

When DTC transfer is stopped by the access error occurs. See section 18, Data Transfer Controller (DTC) unit chapter.

IR flag (Interrupt Status Flag)

The IR status flag indicates an individual interrupt request from the event specified in IELS[9:0].

[Setting condition]

When an interrupt request is received from the associated peripheral module or IRQi pin.

[Clearing condition]

- The IR flag is cleared to 0 by writing 0.
- In the case of DTC.DISEL = 0. At the time other than the final transfer end in DTC transfer during DTCE = 1, IR flag repeat set and cleared by Hardware.
- In the case of DTC.DISEL = 1. For DTC transfers during DTCE = 1, the hardware does not clear the IR flag. Must be cleared by the CPU like No.1.

When DTC transfer except last transfer is completed (DTCE bit is changed from 1 to 0).

In the case of level detection, clear of the IR flag should follow the steps below.

1. Negate the level of an input factor.
2. Run the peripheral read access once and wait for 2 clock cycles of the associated peripheral module clock.
3. Clear the IR flag by writing 0.

Note: During DTCE = 1 writing 0 to IR register is prohibited.

Note: There may be a difference in processing speed between the CPU and ICU, and the CPU may exit the interrupt handler before clearing the IR. Then the CPU will accidentally jump to the interrupt handler again. To avoid this, be sure to read this register before exiting the interrupt handler and make sure that IR is cleared before exiting the interrupt handler.

DTCE bit (DTC Activation Enable)

When the DTCE bit is set to 1, the associated event is selected as the source for DTC activation.

[Setting condition]

- When 1 is written to the DTCE bit.

[Clearing condition]

- When the specified number of transfers is complete. For chain transfers, when the specified number of transfers for the last chain transfer is complete.
- When 0 is written to the DTCE bit.

Note: The secure attribution managed within the Arm CPU NVIC must match the SA (Security Attribution) of IELSRn (n = 0 to 95).

Note: Error during DTC transfer

If an error response occurs during DTC transfer, the DTC notifies the ICU that an error has occurred. ICU clears all bits of the target IELSRn (n = 0 to 95). IELSRn (n = 0 to 95) that is not the target is not cleared.

14.2.18 DELSRm : DMAC Event Link Setting Register m (m = 0 to 7)

Base address: ICU = 0x4000_C000
ICU_NS = 0x5000_C000

Offset address: 0x280 + 0x4 × m

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

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

Bit	Symbol	Function	R/W
9:0	DELS[9:0]	DMAC Event Link Select 0x00: Disable interrupts to the associated DMAC module Others: Event signal number to be linked. For details, see section 14.3.2. Event Number.	R/W*1
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	IR	DMAC Activation Request Status flag 0: No DMAC activation request occurred 1: DMAC activation request occurred	R/W*2
31:17	—	These bits are read as 0. The write value should be 0.	R/W*1

Note: S-TYPE-3, P-TYPE-2
Note 1. This register must be accessed by halfword or word.
Note 2. Writing 1 to the IR flag is prohibited.

This register has different secure access permission depending on the setting of Trusted Event Route Control Register (TEVTRCR).

If the security attribution is configured as Secure,

- Secure access is allowed.
- Non-secure write access is ignored and Non-secure read access is read as 0, TrustZone access error is generated.

If the security attribution is configured as Non-secure and the trusted event route is disabled,

- Secure write access is ignored and Secure read access is read as 0, TrustZone access error is generated.
- Non-secure access is allowed.

If the security attribution is configured as Non-secure and the trusted event route is enabled,

- Secure access to DELS bit is allowed.
- Non-secure write access to DELS bit is ignored and Non-secure read access to DELS bit is allowed.
- Secure write access to other bits is ignored and Secure read access to other bits is read as 0.
- Non-secure access to other bits is allowed.
- TrustZone access error is not generated.

DELS[9:0] bits (DMAC Event Link Select)

The DELS[9:0] bits link an event signal to the associated DMAC module. All DELS[9:0] bits must be written to simultaneously. Do not set the same event number in multiple DELSRm registers.

[Setting condition]

When value is written to these bits.

[Clearing conditions]

When 0 is written to these bits.

When DMA transfer is stopped by the access error occurs. See section 17, DMA Controller (DMAC) unit chapter.

IR flag (DMAC Activation Request Status flag)

This flag is a status flag for a DMAC activation request.

This flag is associated with the DELS[9:0] bits of this register.

[Setting condition]

When a DMAC activation request occurs from the associated peripheral module or IRQi pin.

[Clearing conditions]

When 0 is written to the flag.

When the DMA transfer is started after a DMAC activation request occurs.

When DMA transfer is stopped by the access error occurs. See section 17, DMA Controller (DMAC).

Note: IR flag is automatically cleared after completion of DMA transfer, so do not write "0" except in case of emergency, such as when an abort occurs. DMA transfer operation when "0" is written during DMA transfer cannot be guaranteed.

14.2.19 WUPEN0 : Wake Up Interrupt Enable Register 0

Base address: ICU = 0x4000_C000
ICU_NS = 0x5000_C000

Offset address: 0x1A0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RIIC0 WUPE N	AGT1 CBWU PEN	AGT1 CAWU PEN	AGT1 UDWU PEN	USBF S0WU PEN	USBH SWUP EN	RTCP RDWU PEN	RTCA LMWU PEN	—	—	—	VBATT WUPE N	PVD2 WUPE N	PVD1 WUPE N	—	IWDT WUPE N
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
15:0	IRQWUPEN0 to IRQWUPEN15	Interrupt Software Standby Mode Returns Enable bits from IRQ15 to IRQ0 0: Software Standby Mode returns by IRQn interrupt is disabled. 1: Software Standby Mode returns by IRQn interrupt is enabled*1	R/W
16	IWDTWUPEN	IWDT Interrupt Software Standby Mode Returns Enable bit 0: Software Standby Mode returns by IWDT interrupt is disabled 1: Software Standby Mode returns by IWDT interrupt is enabled	R/W
17	—	This bit is read as 0. The write value should be 0.	R/W
18	PVD1WUPEN	PVD1 Interrupt Software Standby Mode Returns Enable bit 0: Software Standby Mode returns by PVD1 interrupt is disabled 1: Software Standby Mode returns by PVD1 interrupt is enabled	R/W
19	PVD2WUPEN	PVD2 Interrupt Software Standby Mode Returns Enable bit 0: Software Standby Mode returns by PVD2 interrupt is disabled 1: Software Standby Mode returns by PVD2 interrupt is enabled	R/W
20	VBATTWUPEN	VBATT Monitor Interrupt Software Standby Mode Returns Enable bit 0: Software Standby Mode returns by VBATT monitor interrupt is disabled 1: Software Standby Mode returns by VBATT monitor interrupt is enabled	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
24	RTCALMWUPEN	RTC Alarm Interrupt Software Standby Mode Returns Enable bit 0: Software Standby Mode returns by RTC alarm interrupt is disabled 1: Software Standby Mode returns by RTC alarm interrupt is enabled	R/W

Bit	Symbol	Function	R/W
25	RTCPRDWUPEN	RTC Period Interrupt Software Standby Mode Returns Enable bit 0: Software Standby Mode returns by RTC period interrupt is disabled 1: Software Standby Mode returns by RTC period interrupt is enabled	R/W
26	USBHSWUPEN	USBHS Interrupt Software Standby Mode Returns Enable bit 0: Software Standby Mode returns by USBHS interrupt is disabled 1: Software Standby Mode returns by USBHS interrupt is enabled	R/W
27	USBFS0WUPEN	USBFS Interrupt Software Standby Mode Returns Enable bit 0: Software Standby Mode returns by USBFS interrupt is disabled 1: Software Standby Mode returns by USBFS interrupt is enabled	R/W
28	AGT1UDWUPEN	AGT1 Underflow Interrupt Software Standby Mode Returns Enable bit 0: Software Standby Mode returns by AGT1 underflow interrupt is disabled 1: Software Standby Mode returns by AGT1 underflow interrupt is enabled	R/W
29	AGT1CAWUPEN	AGT1 Compare Match A Interrupt Software Standby Mode Returns Enable bit 0: Software Standby Mode returns by AGT1 compare match A interrupt is disabled 1: Software Standby Mode returns by AGT1 compare match A interrupt is enabled	R/W
30	AGT1CBWUPEN	AGT1 Compare Match B Interrupt Software Standby Mode Returns Enable bit 0: Software Standby Mode returns by AGT1 compare match B interrupt is disabled 1: Software Standby Mode returns by AGT1 compare match B interrupt is enabled	R/W
31	RIIC0WUPEN	RIIC0 Address Match Interrupt Software Standby Mode Returns Enable bit 0: Software Standby Mode returns by IIC0 address match A interrupt is disabled 1: Software Standby Mode returns by IIC0 address match A interrupt is enabled	R/W

Note: S-TYPE4, P-TYPE2

Note: It is necessary to set to 1 the DSLPWUPIRQENj register bit corresponding to the IELSRn register that selects the factor that enabled the bit of this register.

Note 1. The security attribution of this register is set for each wakeup event. To avoid the occurrence of a security hole, the target event of a wakeup and the security attribution added to this bit must match.

IRQWUPEN15 bit to IRQWUPEN0 bit(Interrupt Software Standby Mode Returns Enable bits from IRQ15 to IRQ0)

The IRQWUPENn (n = 0 to 15) is the enable bit to control the use of the IRQn pin as a Software Standby mode return factor.

IWDTWUPEN bit (IWDT Interrupt Software Standby Mode Returns Enable bit)

The IWDTWUPEN is the enable bit to control the use of the IWDT interrupt as a Software Standby mode return factor.

PVD1WUPEN bit (PVD1 Interrupt Software Standby Mode Returns Enable bit)

The PVD1WUPEN is the enable bit to control the use of the PVD1 interrupt as a Software Standby mode return factor.

PVD2WUPEN bit (PVD2 Interrupt Software Standby Mode Returns Enable bit)

The PVD2WUPEN is the enable bit to control the use of the PVD2 interrupt as a Software Standby mode return factor.

VBATTWUPEN bits (VBATT Monitor Interrupt Software Standby Mode Returns Enable bit)

The VBATTWUPEN is the enable bit to control the use of the VBATT monitor interrupt as a Software Standby mode return factor.

RTCALMWUPEN bits (RTC Alarm Interrupt Software Standby Mode Returns Enable bit)

The RTCALMWUPEN is the enable bit to control the use of the RTC alarm interrupt as a Software Standby mode return factor.

RTCPRDWUPEN bits (RTC Period Interrupt Software Standby Mode Returns Enable bit)

The RTCPRDWUPEN is the enable bit to control the use of the RTC period interrupt as a Software Standby mode return factor.

USBHSWUPEN bits (USBHS Interrupt Software Standby Mode Returns Enable bit)

The USBHSWUPEN is the enable bit to control the use of the USBHS interrupt as a Software Standby mode return factor.

USBFS0WUPEN bits (USBFS Interrupt Software Standby Mode Returns Enable bit)

The USBFS0WUPEN is the enable bit to control the use of the USBFS interrupt as a Software Standby mode return factor.

AGT1UDWUPEN bits (AGT1 Underflow Interrupt Software Standby Mode Returns Enable bit)

The AGT1UDWUPEN is the enable bit to control the use of the AGT1 underflow interrupt as a Software Standby mode return factor.

AGT1CAWUPEN bits (AGT1 Compare Match A Interrupt Software Standby Mode Returns Enable bit)

The AGT1CAWUPEN is the enable bit to control the use of the AGT1 compare match A interrupt as a Software Standby mode return factor.

AGT1CBWUPEN bits (AGT1 Compare Match B Interrupt Software Standby Mode Returns Enable bit)

The AGT1CBWUPEN is the enable bit to control the use of the AGT1 compare match B interrupt as a Software Standby mode return factor.

RIIC0WUPEN bits (RIIC0 Address Match Interrupt Software Standby Mode Returns Enable bit)

The RIIC0WUPEN is the enable bit to control the use of the RIIC0 interrupt as a Software Standby mode return factor.

14.2.20 WUPEN1 : Wake Up Interrupt Enable Register 1

Base address: ICU = 0x4000_C000
ICU_NS = 0x5000_C000

Offset address: 0x1A4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	IRQWUPEN31	IRQWUPEN30	IRQWUPEN29	IRQWUPEN28	IRQWUPEN27	IRQWUPEN26	IRQWUPEN25	IRQWUPEN24	IRQWUPEN23	IRQWUPEN22	IRQWUPEN21	IRQWUPEN20	IRQWUPEN19	IRQWUPEN18	IRQWUPEN17	IRQWUPEN16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PDMWUPEN	ULP1BWUPEN	ULP1AWUPEN	ULP1UWUPEN	I3CWUPEN	ULP0BWUPEN	ULP0AWUPEN	ULP0UWUPEN	SOSCWUPEN	—	—	—	COMPHS0WUPEN	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	COMPHS0WUPEN	Comparator-HS0 Interrupt Software Standby Mode returns Enable bit 0: Software Standby returns by Comparator-HS0 interrupt is disabled. 1: Software Standby returns by Comparator-HS0 interrupt is enabled.	R/W
6:4	—	These bits are read as 0. The write value should be 0.	R/W
7	SOSCWUPEN	Sub Oscillation Stop Detection Interrupt Software Standby Returns Enable bit 0: Software Standby returns by Sub Oscillation Stop Detection Interrupt is disabled 1: Software Standby returns by Sub Oscillation Stop Detection Interrupt is enabled	R/W
8	ULP0UWUPEN	ULPT0 Underflow Interrupt Software Standby Mode returns Enable bit 0: Software Standby returns by ULPT0 interrupt is disabled. 1: Software Standby returns by ULPT0 Underflow interrupt is enabled.	R/W
9	ULP0AWUPEN	ULPT0 Compare Match A Interrupt Software Standby Mode returns Enable bit 0: Software Standby returns by ULPT0 Compare match A interrupt is disabled. 1: Software Standby returns by ULPT0 Compare match A interrupt is enabled.	R/W
10	ULP0BWUPEN	ULPT0 Compare Match B Interrupt Software Standby Mode returns Enable bit 0: Software Standby returns by ULPT0 Compare match B interrupt is disabled. 1: Software Standby returns by ULPT0 Compare match B interrupt is enabled.	R/W
11	I3CWUPEN	I3C Wakeup Condition Detection Interrupt Software Standby Mode returns Enable bit 0: Software Standby returns by I3C wake up interrupt is disabled. 1: Software Standby returns by I3C wake up interrupt is enabled.	R/W

Bit	Symbol	Function	R/W
12	ULP1UWUPEN	ULPT1 Underflow Interrupt Software Standby Mode returns Enable bit 0: Software Standby returns by ULPT1 Underflow interrupt is disabled. 1: Software Standby returns by ULPT1 Underflow interrupt is enabled	R/W
13	ULP1AWUPEN	ULPT1 Compare Match A Interrupt Software Standby Mode returns Enable bit 0: Software Standby returns by ULPT1 Compare match A interrupt is disabled. 1: Software Standby returns by ULPT1 Compare match A interrupt is enabled.	R/W
14	ULP1BWUPEN	ULPT1 Compare Match B Interrupt Software Standby Mode returns Enable bit 0: Software Standby returns by ULPT1 Compare match B interrupt is disabled. 1: Software Standby returns by ULPT1 Compare match B interrupt is enabled.	R/W
15	PDMWUPEN	PDMIF Sound Detection Interrupt Software Standby Returns Enable bit 0: Software Standby returns by PDMIF Sound Detection interrupt is disabled 1: Software Standby returns by PDMIF Sound Detection interrupt is enabled	R/W
31:16	IRQWUPEN16 to IRQWUPEN31	Interrupt Software Standby Returns Enable bits IRQ31 to IRQ16 0: Software Standby returns by IRQn interrupt is disabled 1: Software Standby returns by IRQn interrupt is enabled	R/W

Note: S-TYPE4, P-TYPE2

Note: It is necessary to set to 1 the DSLPWUPIRQENj register bit corresponding to the IELSRn register that selects the factor that enabled the bit of this register.

Note: The security attribution of this register is set for each wakeup event. To avoid the occurrence of a security hole, the target event of a wakeup and the security attribution added to this bit must match.

COMPHS0WUPEN bit (Comparator-HS0 Interrupt Software Standby Mode returns Enable bit)

This bit is the enable bit to control the use of the Comparator-HS0 interrupt as a Software Standby return factor.

SOSCWUPEN bit (Sub Oscillation Stop Detection Interrupt Software Standby Returns Enable bit)

This bit is the enable bit to control the use of the Sub Oscillation Stop Detection Interrupt as a Software Standby return factor.

ULP0UWUPEN bit (ULPT0 Underflow Interrupt Software Standby Mode returns Enable bit)

This bit is the enable bit to control the use of the ULPT0 Underflow interrupt as a Software Standby return factor.

ULP0AWUPEN bit (ULPT0 Compare Match A Interrupt Software Standby Mode returns Enable bit)

This bit is the enable bit to control the use of the ULPT0 Compare match A interrupt as a Software Standby return factor.

ULP0BWUPEN bit (ULPT0 Compare Match B Interrupt Software Standby Mode returns Enable bit)

This bit is the enable bit to control the use of the ULPT0 Compare match B interrupt as a Software Standby return factor

I3CWUPEN bit (I3C Wakeup Condition Detection Interrupt Software Standby Mode returns Enable bit)

This bit is the enable bit to control the use of the I3C wake up condition detection interrupt as a Software Standby return factor.

ULP1UWUPEN bit (ULPT1 Underflow Interrupt Software Standby Mode returns Enable bit)

This bit is the enable bit to control the use of the ULPT1 Underflow interrupt as a Software Standby return factor.

ULP1AWUPEN bit (ULPT1 Compare Match A Interrupt Software Standby Mode returns Enable bit)

This bit is the enable bit to control the use of the ULPT1 Compare match A interrupt as a Software Standby return factor.

ULP1BWUPEN bit (ULPT1 Compare Match B Interrupt Software Standby Mode returns Enable bit)

This bit is the enable bit to control the use of the ULPT1 Compare match B interrupt as a Software Standby return factor.

PDMWUPEN bit (PDMIF Sound Detection Interrupt Software Standby Returns Enable bit)

This bit is the enable bit to control the use of the PDMIF Sound Detection interrupt as a Software Standby return factor.

IRQWUPEN31 bit to IRQWUPEN16 bit(Interrupt Software Standby Returns Enable bits IRQ31 to IRQ16)

The IRQWUPEN31 is the enable bit to control the use of the IRQ31 pin as a Software Standby mode return factor.

The IRQWUPEN16 is the enable bit to control the use of the IRQ16 pin as a Software Standby mode return factor.

Bit position and IRQ terminal number correspond

14.2.21 DSLPWUPIRQENj : Deep Sleep Wake Up IRQ Enable Register (j = 0 to 2)

Base address: ICU = 0x4000_C000
ICU_NS = 0x5000_C000

Offset address: 0x210 + 0x4 × j

Bit position:	31 … 0
Bit field:	IRQn (n = 32 × j + 31) to IRQn (n = 32 × j + 0)
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	IRQn	IRQ Deep Sleep Returns Enable bit n. (n = 32 × j + i (i = 0 to 31)) 0: Deep Sleep returns by IRQ event is disabled. 1: Deep Sleep returns by IRQ event is enabled.	R/W

Note: S-TYPE4, P-TYPE2

IRQ bits

This bit is the enable bit to control the use of the IRQ as a Deep Sleep return factor.

The IRQn (n = 32 × j + i (i = 0 to 31)) is selected in the IELSRn.IELS[9:0] bit (n = 0 to 95).

14.2.22 INTSELRp : Interrupt Request Select Register (p = 0 to 31)

Base address: ICU_COMMON = 0x4000_6000
ICU_COMMON_NS = 0x5000_6000

Offset address: 0x40 + 0x4 × p

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	IS32p +31	IS32p +30	IS32p +29	IS32p +28	IS32p +27	IS32p +26	IS32p +25	IS32p +24	IS32p +23	IS32p +22	IS32p +21	IS32p +20	IS32p +19	IS32p +18	IS32p +17	IS32p +16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	IS32p +15	IS32p +14	IS32p +13	IS32p +12	IS32p +11	IS32p +10	IS32p +9	IS32p +8	IS32p +7	IS32p +6	IS32p +5	IS32p +4	IS32p +3	IS32p +2	IS32p +1	IS32p +0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	IS32p+31 to IS32p+0	This bit sets whether the interrupt requests and DTC requests are on the CPU0 side or the CPU1 side. 0: Selected as CPU0 factor 1: Selected as CPU1 factor	R/W[*1]

Note: S-TYPE-6, P-TYPE-2

Note 1. Bits in the following conditions are fixed to 0 because they are not assigned.

- INTSELR0.IS0 (bit0: event No.0) with no factor assigned.
- Invalid event number with no factor assigned.
  For more information, see Column M (Event Func. Assign) and Column Y (CPU Int. Maskable) in the Event List.
- Event number (event No.88,89,91,92,101) unique to CPU0/CPU1.
  (DBG CTI1-0 interrupts, IPC CPU mutual interrupt 0, IPC CPU mutual interrupt 1, FPU Exception.)

IS32p + (31 to 0) bits

For example, when IS1 = 0, the cause of Event No.1 is an interrupt for CPU0. Also, when IS2 = 1, the cause of Event No.2 is an interrupt for CPU1.

The relationship between the register and the event number is shown below.

- IS32p+31 to IS32p+0

p = 0 : selected event No.31 to No.0

p = 1 : selected event No.63 to No.32

p = 2 : selected event No.95 to No.64

p = 3 : selected event No.127 to No.96

p = 4 : selected event No.159 to No.128

p = 5 : selected event No.191 to No.160

⋮

p = 14 : selected event No.479 to No.448

p = 15 : selected event No. 511 to No.480

⋮

p = 30: selected event No.991 to No.960

p = 31 : selected event No.1023 to No.992

14.3 Vector Table

The ICU detects maskable and non-maskable interrupts. Interrupt priorities are set up in the Arm NVIC. For information about these registers, see section 14.11. Reference.

14.3.1 Interrupt Vector Table

Table 14.3 describes the interrupt vector table. The interrupt vector addresses conform to the NVIC specifications.

Table 14.3 Interrupt vector table (1 of 4)

Exception number	IRQ number	Vector offset	Source	Description
0	—	0x000	Arm	Initial stack pointer
1	—	0x004	Arm	Initial program counter (reset vector)
2	—	0x008	Arm	Non-Maskable Interrupt (NMI)
3	—	0x00C	Arm	HardFault
4	—	0x010	Arm	MemManage Fault
5	—	0x014	Arm	BusFault
6	—	0x018	Arm	UsageFault
7	—	0x01C	Arm	SecureFault
8	—	0x020	Arm	Reserved
9	—	0x024	Arm	Reserved
10	—	0x028	Arm	Reserved
11	—	0x02C	Arm	Supervisor Call (SVCall)
12	—	0x030	Arm	DebugMonitor
13	—	0x034	Arm	Reserved
14	—	0x038	Arm	Pendable request for system service (PendableSrvReq)
15	—	0x03C	Arm	System Tick Timer (SysTick)
16	0	0x040	ICU.IELSR0	Event selected in the ICU.IELSR0 register
17	1	0x044	ICU.IELSR1	Event selected in the ICU.IELSR1 register
18	2	0x048	ICU.IELSR2	Event selected in the ICU.IELSR2 register
19	3	0x04C	ICU.IELSR3	Event selected in the ICU.IELSR3 register
20	4	0x050	ICU.IELSR4	Event selected in the ICU.IELSR4 register
21	5	0x054	ICU.IELSR5	Event selected in the ICU.IELSR5 register

Table 14.3 Interrupt vector table (2 of 4)

Exception number	IRQ number	Vector offset	Source	Description
22	6	0x058	ICU.IELSR6	Event selected in the ICU.IELSR6 register
23	7	0x05C	ICU.IELSR7	Event selected in the ICU.IELSR7 register
24	8	0x060	ICU.IELSR8	Event selected in the ICU.IELSR8 register
25	9	0x064	ICU.IELSR9	Event selected in the ICU.IELSR9 register
26	10	0x068	ICU.IELSR10	Event selected in the ICU.IELSR10 register
27	11	0x06C	ICU.IELSR11	Event selected in the ICU.IELSR11 register
28	12	0x070	ICU.IELSR12	Event selected in the ICU.IELSR12 register
29	13	0x074	ICU.IELSR13	Event selected in the ICU.IELSR13 register
30	14	0x078	ICU.IELSR14	Event selected in the ICU.IELSR14 register
31	15	0x07C	ICU.IELSR15	Event selected in the ICU.IELSR15 register
32	16	0x080	ICU.IELSR16	Event selected in the ICU.IELSR16 register
33	17	0x084	ICU.IELSR17	Event selected in the ICU.IELSR17 register
34	18	0x088	ICU.IELSR18	Event selected in the ICU.IELSR18 register
35	19	0x08C	ICU.IELSR19	Event selected in the ICU.IELSR19 register
36	20	0x090	ICU.IELSR20	Event selected in the ICU.IELSR20 register
37	21	0x094	ICU.IELSR21	Event selected in the ICU.IELSR21 register
38	22	0x098	ICU.IELSR22	Event selected in the ICU.IELSR22 register
39	23	0x09C	ICU.IELSR23	Event selected in the ICU.IELSR23 register
40	24	0x0A0	ICU.IELSR24	Event selected in the ICU.IELSR24 register
41	25	0x0A4	ICU.IELSR25	Event selected in the ICU.IELSR25 register
42	26	0x0A8	ICU.IELSR26	Event selected in the ICU.IELSR26 register
43	27	0x0AC	ICU.IELSR27	Event selected in the ICU.IELSR27 register
44	28	0x0B0	ICU.IELSR28	Event selected in the ICU.IELSR28 register
45	29	0x0B4	ICU.IELSR29	Event selected in the ICU.IELSR29 register
46	30	0x0B8	ICU.IELSR30	Event selected in the ICU.IELSR30 register
47	31	0x0BC	ICU.IELSR31	Event selected in the ICU.IELSR31 register
48	32	0x0C0	ICU.IELSR32	Event selected in the ICU.IELSR32 register
49	33	0x0C4	ICU.IELSR33	Event selected in the ICU.IELSR33 register
50	34	0x0C8	ICU.IELSR34	Event selected in the ICU.IELSR34 register
51	35	0x0CC	ICU.IELSR35	Event selected in the ICU.IELSR35 register
52	36	0x0D0	ICU.IELSR36	Event selected in the ICU.IELSR36 register
53	37	0x0D4	ICU.IELSR37	Event selected in the ICU.IELSR37 register
54	38	0x0D8	ICU.IELSR38	Event selected in the ICU.IELSR38 register
55	39	0x0DC	ICU.IELSR39	Event selected in the ICU.IELSR39 register
56	40	0x0E0	ICU.IELSR40	Event selected in the ICU.IELSR40 register
57	41	0x0E4	ICU.IELSR41	Event selected in the ICU.IELSR41 register
58	42	0x0E8	ICU.IELSR42	Event selected in the ICU.IELSR42 register
59	43	0x0EC	ICU.IELSR43	Event selected in the ICU.IELSR43 register
60	44	0x0F0	ICU.IELSR44	Event selected in the ICU.IELSR44 register
61	45	0x0F4	ICU.IELSR45	Event selected in the ICU.IELSR45 register
62	46	0x0F8	ICU.IELSR46	Event selected in the ICU.IELSR46 register

Table 14.3 Interrupt vector table (3 of 4)

Exception number	IRQ number	Vector offset	Source	Description
63	47	0x0FC	ICU.IELSR47	Event selected in the ICU.IELSR47 register
64	48	0x100	ICU.IELSR48	Event selected in the ICU.IELSR48 register
65	49	0x104	ICU.IELSR49	Event selected in the ICU.IELSR49 register
66	50	0x108	ICU.IELSR50	Event selected in the ICU.IELSR50 register
67	51	0x10C	ICU.IELSR51	Event selected in the ICU.IELSR51 register
68	52	0x110	ICU.IELSR52	Event selected in the ICU.IELSR52 register
69	53	0x114	ICU.IELSR53	Event selected in the ICU.IELSR53 register
70	54	0x118	ICU.IELSR54	Event selected in the ICU.IELSR54 register
71	55	0x11C	ICU.IELSR55	Event selected in the ICU.IELSR55 register
72	56	0x120	ICU.IELSR56	Event selected in the ICU.IELSR56 register
73	57	0x124	ICU.IELSR57	Event selected in the ICU.IELSR57 register
74	58	0x128	ICU.IELSR58	Event selected in the ICU.IELSR58 register
75	59	0x12C	ICU.IELSR59	Event selected in the ICU.IELSR59 register
76	60	0x130	ICU.IELSR60	Event selected in the ICU.IELSR60 register
77	61	0x134	ICU.IELSR61	Event selected in the ICU.IELSR61 register
78	62	0x138	ICU.IELSR62	Event selected in the ICU.IELSR62 register
79	63	0x13C	ICU.IELSR63	Event selected in the ICU.IELSR63 register
80	64	0x140	ICU.IELSR64	Event selected in the ICU.IELSR64 register
81	65	0x144	ICU.IELSR65	Event selected in the ICU.IELSR65 register
82	66	0x148	ICU.IELSR66	Event selected in the ICU.IELSR66 register
83	67	0x14C	ICU.IELSR67	Event selected in the ICU.IELSR67 register
84	68	0x150	ICU.IELSR68	Event selected in the ICU.IELSR68 register
85	69	0x154	ICU.IELSR69	Event selected in the ICU.IELSR69 register
86	70	0x158	ICU.IELSR70	Event selected in the ICU.IELSR70 register
87	71	0x15C	ICU.IELSR71	Event selected in the ICU.IELSR71 register
88	72	0x160	ICU.IELSR72	Event selected in the ICU.IELSR72 register
89	73	0x164	ICU.IELSR73	Event selected in the ICU.IELSR73 register
90	74	0x168	ICU.IELSR74	Event selected in the ICU.IELSR74 register
91	75	0x16C	ICU.IELSR75	Event selected in the ICU.IELSR75 register
92	76	0x170	ICU.IELSR76	Event selected in the ICU.IELSR76 register
93	77	0x174	ICU.IELSR77	Event selected in the ICU.IELSR77 register
94	78	0x178	ICU.IELSR78	Event selected in the ICU.IELSR78 register
95	79	0x17C	ICU.IELSR79	Event selected in the ICU.IELSR79 register
96	80	0x180	ICU.IELSR80	Event selected in the ICU.IELSR80 register
97	81	0x184	ICU.IELSR81	Event selected in the ICU.IELSR81 register
98	82	0x188	ICU.IELSR82	Event selected in the ICU.IELSR82 register
99	83	0x18C	ICU.IELSR83	Event selected in the ICU.IELSR83 register
100	84	0x190	ICU.IELSR84	Event selected in the ICU.IELSR84 register
101	85	0x194	ICU.IELSR85	Event selected in the ICU.IELSR85 register
102	86	0x198	ICU.IELSR86	Event selected in the ICU.IELSR86 register
103	87	0x19C	ICU.IELSR87	Event selected in the ICU.IELSR87 register

Table 14.3 Interrupt vector table (4 of 4)

Exception number	IRQ number	Vector offset	Source	Description
104	88	0x1A0	ICU.IELSR88	Event selected in the ICU.IELSR88 register
105	89	0x1A4	ICU.IELSR89	Event selected in the ICU.IELSR89 register
106	90	0x1A8	ICU.IELSR90	Event selected in the ICU.IELSR90 register
107	91	0x1AC	ICU.IELSR91	Event selected in the ICU.IELSR91 register
108	92	0x1B0	ICU.IELSR92	Event selected in the ICU.IELSR92 register
109	93	0x1B4	ICU.IELSR93	Event selected in the ICU.IELSR93 register
110	94	0x1B8	ICU.IELSR94	Event selected in the ICU.IELSR94 register
111	95	0x1BC	ICU.IELSR95	Event selected in the ICU.IELSR95 register

14.3.2 Event Number

The following table lists heading details for Table 14.5, which describes each event number.

Table 14.4 Heading details for Event List

Heading	Description
Interrupt request source	Name of the source generating the interrupt request
Name	Name of the interrupt
Connect to NVIC	" ✓" indicates the interrupt can be used as a CPU interrupt
Invoke DTC	" ✓" indicates the interrupt can be used to request DTC activation
Invoke DMAC	"✓" indicates the interrupt can be used to request DMAC activation
Canceling CPU Deep Sleep	"✓" indicates the interrupt can be used to request a return from CPU Deep Sleep mode
Canceling Software Standby	" ✓" indicates the interrupt can be used to request a return from Software Standby mode
Canceling Deep Software Standby	"✓" indicates the interrupt can be used to request a return from Deep Software Standby mode

Table 14.5 Event List (1 of 13)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x001	Port	PORT_IRQ0	✓	✓	✓	✓	✓	✓
0x002		PORT_IRQ1	✓	✓	✓	✓	✓	✓
0x003		PORT_IRQ2	✓	✓	✓	✓	✓	✓
0x004		PORT_IRQ3	✓	✓	✓	✓	✓	✓
0x005		PORT_IRQ4	✓	✓	✓	✓	✓	✓
0x006		PORT_IRQ5	✓	✓	✓	✓	✓	✓
0x007		PORT_IRQ6	✓	✓	✓	✓	✓	✓
0x008		PORT_IRQ7	✓	✓	✓	✓	✓	✓
0x009		PORT_IRQ8	✓	✓	✓	✓	✓	✓
0x00A		PORT_IRQ9	✓	✓	✓	✓	✓	✓
0x00B		PORT_IRQ10	✓	✓	✓	✓	✓	✓
0x00C		PORT_IRQ11	✓	✓	✓	✓	✓	✓
0x00D		PORT_IRQ12	✓	✓	✓	✓	✓	✓
0x00E		PORT_IRQ13	✓	✓	✓	✓	✓	✓
0x00F		PORT_IRQ14	✓	✓	✓	✓	✓	✓
0x010		PORT_IRQ15	✓	✓	✓	✓	✓	✓
0x011		PORT_IRQ16	✓	✓	✓	✓	✓	✓
0x012		PORT_IRQ17	✓	✓	✓	✓	✓	✓
0x013		PORT_IRQ18	✓	✓	✓	✓	✓	✓
0x014		PORT_IRQ19	✓	✓	✓	✓	✓	✓
0x015		PORT_IRQ20	✓	✓	✓	✓	✓	✓
0x016		PORT_IRQ21	✓	✓	✓	✓	✓	✓
0x017		PORT_IRQ22	✓	✓	✓	✓	✓	✓
0x018		PORT_IRQ23	✓	✓	✓	✓	✓	✓
0x019		PORT_IRQ24	✓	✓	✓	✓	✓	✓
0x01A		PORT_IRQ25	✓	✓	✓	✓	✓	✓
0x01B		PORT_IRQ26	✓	✓	✓	✓	✓	✓
0x01C		PORT_IRQ27	✓	✓	✓	✓	✓	✓
0x01D		PORT_IRQ28	✓	✓	✓	✓	✓	✓
0x01E		PORT_IRQ29	✓	✓	✓	✓	✓	✓
0x01F		PORT_IRQ30	✓	✓	✓	✓	✓	✓
0x020		PORT_IRQ31	✓	✓	✓	✓	✓	✓
0x040	DMAC00	DMAC00_INT	✓	✓	—	✓	—	—
0x041	DMAC01	DMAC01_INT	✓	✓	—	✓	—	—
0x042	DMAC02	DMAC02_INT	✓	✓	—	✓	—	—
0x043	DMAC03	DMAC03_INT	✓	✓	—	✓	—	—
0x044	DMAC04	DMAC04_INT	✓	✓	—	✓	—	—
0x045	DMAC05	DMAC05_INT	✓	✓	—	✓	—	—
0x046	DMAC06	DMAC06_INT	✓	✓	—	✓	—	—
0x047	DMAC07	DMAC07_INT	✓	✓	—	✓	—	—

Table 14.5 Event List (2 of 13)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x048	DMAC10	DMAC10_INT	✓	✓	—	✓	—	—
0x049	DMAC11	DMAC11_INT	✓	✓	—	✓	—	—
0x04A	DMAC12	DMAC12_INT	✓	✓	—	✓	—	—
0x04B	DMAC13	DMAC13_INT	✓	✓	—	✓	—	—
0x04C	DMAC14	DMAC14_INT	✓	✓	—	✓	—	—
0x04D	DMAC15	DMAC15_INT	✓	✓	—	✓	—	—
0x04E	DMAC16	DMAC16_INT	✓	✓	—	✓	—	—
0x04F	DMAC17	DMAC17_INT	✓	✓	—	✓	—	—
0x056	DMAC/DTC	DMA0_TRANSERR	✓	—	—	✓	—	—
0x057	DMAC/DTC	DMA1_TRANSERR	✓	—	—	✓	—	—
0x058	DBG	DBG_CTIIRQ0	✓	—	—	—	—	—
0x059		DBG_CTIIRQ1	✓	—	—	—	—	—
0x05A		DBG_JBRXI	✓	—	—	✓	—	—
0x05B	IPC	IPC_IRQ0	✓	—	—	✓	—	—
0x05C		IPC_IRQ1	✓	—	—	✓	—	—
0x05F	Local Memory	LM0_ERR	✓	—	—	✓	—	—
0x060		LM1_ERR	✓	—	—	✓	—	—
0x061	CPU	CPU0_LOCKUP	✓	—	—	✓	—	—
0x062		CPU1_LOCKUP	✓	—	—	✓	—	—
0x063	BUS	BUS_ERR	✓	—	—	✓	—	—
0x064	CM	CM_ERR	✓	—	—	✓	—	—
0x065	FPU	FPU_EXC	✓	—	—	—	—	—
0x068	MRAM	MRAM_MRCRD	✓	—	—	✓	—	—
0x069		MRAM_MRERD	✓	—	—	✓	—	—
0x06B		MRAM_MRCPR	✓	—	—	✓	—	—
0x06C		MRAM_MREPR	✓	—	—	✓	—	—
0x06D		MRAM_ENDOFPE	✓	—	—	✓	—	—
0x070	PVD1	PVD_PVD1	✓	—	—	✓	✓	✓*4
0x071	PVD2	PVD_PVD2	✓	—	—	✓	✓	✓*4
0x075	BBF	VBATT_TADI	✓	—	—	✓	✓	✓
0x076	MOSC	MOSC_STOP	✓	—	—	✓	—	—
0x077	SOSC	SOSC_STOP	✓	—	—	✓	✓	✓
0x080	ULPT0	ULPT0_ULPTI	✓	✓	✓	✓	✓	✓*5
0x081		ULPT0_ULPTCMAI	✓	✓	✓	✓	✓	—
0x082		ULPT0_ULPTCMBI	✓	✓	✓	✓	✓	—
0x083	ULPT1	ULPT1_ULPTI	✓	✓	✓	✓	✓	✓*5
0x084		ULPT1_ULPTCMAI	✓	✓	✓	✓	✓	—
0x085		ULPT1_ULPTCMBI	✓	✓	✓	✓	✓	—

Table 14.5 Event List (3 of 13)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x086	AGT0	AGT0_AGTI	✓	✓	✓	✓	—	—
0x087		AGT0_AGTCMAI	✓	✓	✓	✓	—	—
0x088		AGT0_AGTCMBI	✓	✓	✓	✓	—	—
0x089	AGT1	AGT1_AGTI	✓	✓	✓	✓	✓	—
0x08A		AGT1_AGTCMAI	✓	✓	✓	✓	✓	—
0x08B		AGT1_AGTCMBI	✓	✓	✓	✓	✓	—
0x092	IWDT	IWDT_NMIUNDF	✓	—	—	✓	✓	✓ *5
0x093	WDT0	WDT0_NMIUNDF	✓	—	—	✓	—	—
0x094	WDT1	WDT1_NMIUNDF	✓	—	—	✓	—	—
0x095	RTC	RTC_ALM	✓	—	—	✓	✓	✓
0x096		RTC_PRD	✓	—	—	✓	✓	✓
0x097		RTC_CUP	✓	—	—	✓	—	—
0x098	USBFS	USBFS_D0FIFO	✓	✓	✓	✓	—	—
0x099		USBFS_D1FIFO	✓	✓	✓	✓	—	—
0x09A		USBFS_USBI	✓	—	—	✓	—	—
0x09B		USBFS_USBR	✓	—	—	✓	✓	✓ *5
0x09C	IIC0	IIC0_RXI	✓	✓	✓	✓	—	—
0x09D		IIC0_TXI	✓	✓	✓	✓	—	—
0x09E		IIC0_TEI	✓	—	—	✓	—	—
0x09F		IIC0_EEI	✓	—	—	✓	—	—
0x0A0		IIC0_WUI	✓	—	—	✓	✓	—
0x0A1	IIC1	IIC1_RXI	✓	✓	✓	✓	—	—
0x0A2		IIC1_TXI	✓	✓	✓	✓	—	—
0x0A3		IIC1_TEI	✓	—	—	✓	—	—
0x0A4		IIC1_EEI	✓	—	—	✓	—	—
0x0A6	IIC2	IIC2_RXI	✓	✓	✓	✓	—	—
0x0A7		IIC2_TXI	✓	✓	✓	✓	—	—
0x0A8		IIC2_TEI	✓	—	—	✓	—	—
0x0A9		IIC2_EEI	✓	—	—	✓	—	—
0x0AB	SDHI/ MMC0	SDHI_MMC0_ACCS	✓	—	—	✓	—	—
0x0AC		SDHI_MMC0_SDIO	✓	—	—	✓	—	—
0x0AD		SDHI_MMC0_CARD	✓	—	—	✓	—	—
0x0AE		SDHI_MMC0_ODMSDBREQ	—	✓	✓	✓	—	—
0x0AF	SDHI/ MMC1	SDHI_MMC1_ACCS	✓	—	—	✓	—	—
0x0B0		SDHI_MMC1_SDIO	✓	—	—	✓	—	—
0x0B1		SDHI_MMC1_CARD	✓	—	—	✓	—	—
0x0B2		SDHI_MMC1_ODMSDBREQ	—	✓	✓	✓	—	—
0x0B3	SSI0	SSI0_SSITXI	✓	✓	✓	✓	—	—
0x0B4		SSI0_SSIRXI	✓	✓	✓	✓	—	—
0x0B6		SSI0_SSIF	✓	—	—	✓	—	—

Table 14.5 Event List (4 of 13)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x0B9	SSI1	SSI1_SSIRT	✓	✓	✓	✓	—	—
0x0BA		SSI1_SSIF	✓	—	—	✓	—	—
0x0BB	OSPI0	OSPI0_ERR	✓	—	—	✓	—	—
0x0BC		OSPI0_CMP	✓	—	—	✓	—	—
0x0BD	OSPI1	OSPI1_ERR	✓	—	—	✓	—	—
0x0BE		OSPI1_CMP	✓	—	—	✓	—	—
0x0BF	PDMIF	PDM_DAT0	✓	✓	✓	✓	—	—
0x0C0		PDM_DAT1	✓	✓	✓	✓	—	—
0x0C1		PDM_DAT2	✓	✓	✓	✓	—	—
0x0C2		PDM_SDET	✓	—	—	✓	✓	—
0x0C3		PDM_ERR0	✓	—	—	✓	—	—
0x0C4		PDM_ERR1	✓	—	—	✓	—	—
0x0C5		PDM_ERR2	✓	—	—	✓	—	—
0x0C6	ACMPHS	ACMP_HS0	✓	—	—	✓*1	✓*1	—
0x0C7		ACMP_HS1	✓	—	—	✓	—	—
0x0C8		ACMP_HS2	✓	—	—	✓	—	—
0x0C9		ACMP_HS3	✓	—	—	✓	—	—
0x0CC	ELC	ELC_SWEVT0	✓*3	✓	—	✓	—	—
0x0CD		ELC_SWEVT1	✓*3	✓	—	✓	—	—
0x0CE		ELC_SWEVT2	✓*3	✓	—	✓	—	—
0x0CF		ELC_SWEVT3	✓*3	✓	—	✓	—	—
0x0D0	PORT	IOPORT_GROUP1	✓	✓*2	✓*2	✓	—	—
0x0D1		IOPORT_GROUP2	✓	✓*2	✓*2	✓	—	—
0x0D2		IOPORT_GROUP3	✓	✓*2	✓*2	✓	—	—
0x0D3		IOPORT_GROUP4	✓	✓*2	✓*2	✓	—	—
0x0D4	CAC	CAC_FEERI	✓	—	—	✓	—	—
0x0D5		CAC_MENDI	✓	—	—	✓	—	—
0x0D6		CAC_OVFI	✓	—	—	✓	—	—
0x0D7	POEG	POEG_GROUPA	✓	—	—	✓	—	—
0x0D8		POEG_GROUPB	✓	—	—	✓	—	—
0x0D9		POEG_GROUPC	✓	—	—	✓	—	—
0x0DA		POEG_GROUPD	✓	—	—	✓	—	—
0x180	GPT	GPT_UVWEDGE	✓	—	—	✓	—	—

Table 14.5 Event List (5 of 13)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x181	GPT0	GPT0_CCMPA	✓	✓	✓	✓	—	—
0x182		GPT0_CCMPB	✓	✓	✓	✓	—	—
0x183		GPT0_CMPC	✓	✓	✓	✓	—	—
0x184		GPT0_CMPD	✓	✓	✓	✓	—	—
0x185		GPT0_CMPE	✓	✓	✓	✓	—	—
0x186		GPT0_CMPF	✓	✓	✓	✓	—	—
0x187		GPT0_OVF	✓	✓	✓	✓	—	—
0x188		GPT0_UDF	✓	✓	✓	✓	—	—
0x189		GPT0_PC	✓	✓	✓	✓	—	—
0x18A	GPT1	GPT1_CCMPA	✓	✓	✓	✓	—	—
0x18B		GPT1_CCMPB	✓	✓	✓	✓	—	—
0x18C		GPT1_CMPC	✓	✓	✓	✓	—	—
0x18D		GPT1_CMPD	✓	✓	✓	✓	—	—
0x18E		GPT1_CMPE	✓	✓	✓	✓	—	—
0x18F		GPT1_CMPF	✓	✓	✓	✓	—	—
0x190		GPT1_OVF	✓	✓	✓	✓	—	—
0x191		GPT1_UDF	✓	✓	✓	✓	—	—
0x192		GPT1_PC	✓	✓	✓	✓	—	—
0x193	GPT2	GPT2_CCMPA	✓	✓	✓	✓	—	—
0x194		GPT2_CCMPB	✓	✓	✓	✓	—	—
0x195		GPT2_CMPC	✓	✓	✓	✓	—	—
0x196		GPT2_CMPD	✓	✓	✓	✓	—	—
0x197		GPT2_CMPE	✓	✓	✓	✓	—	—
0x198		GPT2_CMPF	✓	✓	✓	✓	—	—
0x199		GPT2_OVF	✓	✓	✓	✓	—	—
0x19A		GPT2_UDF	✓	✓	✓	✓	—	—
0x19B		GPT2_PC	✓	✓	✓	✓	—	—
0x19C	GPT3	GPT3_CCMPA	✓	✓	✓	✓	—	—
0x19D		GPT3_CCMPB	✓	✓	✓	✓	—	—
0x19E		GPT3_CMPC	✓	✓	✓	✓	—	—
0x19F		GPT3_CMPD	✓	✓	✓	✓	—	—
0x1A0		GPT3_CMPE	✓	✓	✓	✓	—	—
0x1A1		GPT3_CMPF	✓	✓	✓	✓	—	—
0x1A2		GPT3_OVF	✓	✓	✓	✓	—	—
0x1A3		GPT3_UDF	✓	✓	✓	✓	—	—
0x1A4		GPT3_PC	✓	✓	✓	✓	—	—

Table 14.5 Event List (6 of 13)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x1A5	GPT4	GPT4_CCMPA	✓	✓	✓	✓	—	—
0x1A6		GPT4_CCMPB	✓	✓	✓	✓	—	—
0x1A7		GPT4_CMPC	✓	✓	✓	✓	—	—
0x1A8		GPT4_CMPD	✓	✓	✓	✓	—	—
0x1A9		GPT4_CMPE	✓	✓	✓	✓	—	—
0x1AA		GPT4_CMPF	✓	✓	✓	✓	—	—
0x1AB		GPT4_OVF	✓	✓	✓	✓	—	—
0x1AC		GPT4_UDF	✓	✓	✓	✓	—	—
0x1AE	GPT5	GPT5_CCMPA	✓	✓	✓	✓	—	—
0x1AF		GPT5_CCMPB	✓	✓	✓	✓	—	—
0x1B0		GPT5_CMPC	✓	✓	✓	✓	—	—
0x1B1		GPT5_CMPD	✓	✓	✓	✓	—	—
0x1B2		GPT5_CMPE	✓	✓	✓	✓	—	—
0x1B3		GPT5_CMPF	✓	✓	✓	✓	—	—
0x1B4		GPT5_OVF	✓	✓	✓	✓	—	—
0x1B5		GPT5_UDF	✓	✓	✓	✓	—	—
0x1B7	GPT6	GPT6_CCMPA	✓	✓	✓	✓	—	—
0x1B8		GPT6_CCMPB	✓	✓	✓	✓	—	—
0x1B9		GPT6_CMPC	✓	✓	✓	✓	—	—
0x1BA		GPT6_CMPD	✓	✓	✓	✓	—	—
0x1BB		GPT6_CMPE	✓	✓	✓	✓	—	—
0x1BC		GPT6_CMPF	✓	✓	✓	✓	—	—
0x1BD		GPT6_OVF	✓	✓	✓	✓	—	—
0x1BE		GPT6_UDF	✓	✓	✓	✓	—	—
0x1C0	GPT7	GPT7_CCMPA	✓	✓	✓	✓	—	—
0x1C1		GPT7_CCMPB	✓	✓	✓	✓	—	—
0x1C2		GPT7_CMPC	✓	✓	✓	✓	—	—
0x1C3		GPT7_CMPD	✓	✓	✓	✓	—	—
0x1C4		GPT7_CMPE	✓	✓	✓	✓	—	—
0x1C5		GPT7_CMPF	✓	✓	✓	✓	—	—
0x1C6		GPT7_OVF	✓	✓	✓	✓	—	—
0x1C7		GPT7_UDF	✓	✓	✓	✓	—	—
0x1C9	GPT8	GPT8_CCMPA	✓	✓	✓	✓	—	—
0x1CA		GPT8_CCMPB	✓	✓	✓	✓	—	—
0x1CB		GPT8_CMPC	✓	✓	✓	✓	—	—
0x1CC		GPT8_CMPD	✓	✓	✓	✓	—	—
0x1CD		GPT8_CMPE	✓	✓	✓	✓	—	—
0x1CE		GPT8_CMPF	✓	✓	✓	✓	—	—
0x1CF		GPT8_OVF	✓	✓	✓	✓	—	—
0x1D0		GPT8_UDF	✓	✓	✓	✓	—	—

Table 14.5 Event List (7 of 13)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x1D2	GPT9	GPT9_CCMPA	✓	✓	✓	✓	—	—
0x1D3		GPT9_CCMPB	✓	✓	✓	✓	—	—
0x1D4		GPT9_CMPC	✓	✓	✓	✓	—	—
0x1D5		GPT9_CMPD	✓	✓	✓	✓	—	—
0x1D6		GPT9_CMPE	✓	✓	✓	✓	—	—
0x1D7		GPT9_CMPF	✓	✓	✓	✓	—	—
0x1D8		GPT9_OVF	✓	✓	✓	✓	—	—
0x1D9		GPT9_UDF	✓	✓	✓	✓	—	—
0x1DB	GPT10	GPT10_CCMPA	✓	✓	✓	✓	—	—
0x1DC		GPT10_CCMPB	✓	✓	✓	✓	—	—
0x1DD		GPT10_CMPC	✓	✓	✓	✓	—	—
0x1DE		GPT10_CMPD	✓	✓	✓	✓	—	—
0x1DF		GPT10_CMPE	✓	✓	✓	✓	—	—
0x1E0		GPT10_CMPF	✓	✓	✓	✓	—	—
0x1E1		GPT10_OVF	✓	✓	✓	✓	—	—
0x1E2		GPT10_UDF	✓	✓	✓	✓	—	—
0x1E3		GPT10_PC	✓	✓	✓	✓	—	—
0x1E4	GPT11	GPT11_CCMPA	✓	✓	✓	✓	—	—
0x1E5		GPT11_CCMPB	✓	✓	✓	✓	—	—
0x1E6		GPT11_CMPC	✓	✓	✓	✓	—	—
0x1E7		GPT11_CMPD	✓	✓	✓	✓	—	—
0x1E8		GPT11_CMPE	✓	✓	✓	✓	—	—
0x1E9		GPT11_CMPF	✓	✓	✓	✓	—	—
0x1EA		GPT11_OVF	✓	✓	✓	✓	—	—
0x1EB		GPT11_UDF	✓	✓	✓	✓	—	—
0x1EC		GPT11_PC	✓	✓	✓	✓	—	—
0x1ED	GPT12	GPT12_CCMPA	✓	✓	✓	✓	—	—
0x1EE		GPT12_CCMPB	✓	✓	✓	✓	—	—
0x1EF		GPT12_CMPC	✓	✓	✓	✓	—	—
0x1F0		GPT12_CMPD	✓	✓	✓	✓	—	—
0x1F1		GPT12_CMPE	✓	✓	✓	✓	—	—
0x1F2		GPT12_CMPF	✓	✓	✓	✓	—	—
0x1F3		GPT12_OVF	✓	✓	✓	✓	—	—
0x1F4		GPT12_UDF	✓	✓	✓	✓	—	—
0x1F5		GPT12_PC	✓	✓	✓	✓	—	—

Table 14.5 Event List (8 of 13)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x1F6	GPT13	GPT13_CCMPA	✓	✓	✓	✓	—	—
0x1F7		GPT13_CCMPB	✓	✓	✓	✓	—	—
0x1F8		GPT13_CMPC	✓	✓	✓	✓	—	—
0x1F9		GPT13_CMPD	✓	✓	✓	✓	—	—
0x1FA		GPT13_CMPE	✓	✓	✓	✓	—	—
0x1FB		GPT13_CMPF	✓	✓	✓	✓	—	—
0x1FC		GPT13_OVF	✓	✓	✓	✓	—	—
0x1FD		GPT13_UDF	✓	✓	✓	✓	—	—
0x1FE		GPT13_PC	✓	✓	✓	✓	—	—
0x29A	ESWM	ETHER_FWEI	✓	—	—	✓	—	—
0x29B		ETHER_CAEI	✓	—	—	✓	—	—
0x29C		ETHER_GWEI	✓	—	—	✓	—	—
0x29D		ETHER_EAEI0	✓	—	—	✓	—	—
0x29E		ETHER_EAEI1	✓	—	—	✓	—	—
0x29F		ETHER_PTPSI0	✓	—	—	✓	—	—
0x2A0		ETHER_PTPSI1	✓	—	—	✓	—	—
0x2A1		ETHER_FWSI	✓	—	—	✓	—	—
0x2A3		ETHER_CAMI	✓	—	—	✓	—	—
0x2A4		ETHER_EASI0	✓	—	—	✓	—	—
0x2A5		ETHER_EASI1	✓	—	—	✓	—	—
0x2A6		ETHER_GWDI0	✓	—	—	✓	—	—
0x2A7		ETHER_GWDI1	✓	—	—	✓	—	—
0x2A8		ETHER_GWDI2	✓	—	—	✓	—	—
0x2A9		ETHER_GWDI3	✓	—	—	✓	—	—
0x2AA		ETHER_GWDI4	✓	—	—	✓	—	—
0x2AB		ETHER_GWDI5	✓	—	—	✓	—	—
0x2AC		ETHER_GWDI6	✓	—	—	✓	—	—
0x2AD		ETHER_GWDI7	✓	—	—	✓	—	—
0x2AE		ETHER_TSDI0	✓	—	—	✓	—	—
0x2AF		ETHER_TSDI1	✓	—	—	✓	—	—
0x2B0		ETHER_MDIO0	✓	—	—	✓	—	—
0x2B1		ETHER_MDIO1	✓	—	—	✓	—	—
0x2B2		ETHER_RMPI0	✓	—	—	✓	—	—
0x2B3		ETHER_RMPI1	✓	—	—	✓	—	—
0x2B4		GPTP_PTPOUT0	✓	✓	✓	✓	—	—
0x2B5		GPTP_PTPOUT1	✓	✓	✓	✓	—	—
0x2B6		GPTP_PTPOUT2	✓	✓	✓	✓	—	—
0x2B7		GPTP_PTPOUT3	✓	✓	✓	✓	—	—
0x2B8		GPTP_MATCH0	✓	✓	✓	✓	—	—
0x2B9		GPTP_MATCH1	✓	✓	✓	✓	—	—

Table 14.5 Event List (9 of 13)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x2C1	USBHS	USBHS_D0FIFO	✓	✓	✓	✓	—	—
0x2C2		USBHS_D1FIFO	✓	✓	✓	✓	—	—
0x2C3		USBHS_USBIR	✓	—	—	✓	✓	✓*5
0x2C4	SCI0	SCI0_RXI	✓	✓	✓	✓	—	—
0x2C5		SCI0_TXI	✓	✓	✓	✓	—	—
0x2C6		SCI0_TEI	✓	—	—	✓	—	—
0x2C7		SCI0_ERI	✓	—	—	✓	—	—
0x2C8		SCI0_AED	✓	—	—	✓	—	—
0x2C9		SCI0_BFD	✓	—	—	✓	—	—
0x2CA		SCI0_AM	✓	—	—	✓	—	—
0x2CB	SCI1	SCI1_RXI	✓	✓	✓	✓	—	—
0x2CC		SCI1_TXI	✓	✓	✓	✓	—	—
0x2CD		SCI1_TEI	✓	—	—	✓	—	—
0x2CE		SCI1_ERI	✓	—	—	✓	—	—
0x2CF		SCI1_AED	✓	—	—	✓	—	—
0x2D0		SCI1_BFD	✓	—	—	✓	—	—
0x2D1		SCI1_AM	✓	—	—	✓	—	—
0x2D2	SCI2	SCI2_RXI	✓	✓	✓	✓	—	—
0x2D3		SCI2_TXI	✓	✓	✓	✓	—	—
0x2D4		SCI2_TEI	✓	—	—	✓	—	—
0x2D5		SCI2_ERI	✓	—	—	✓	—	—
0x2D6		SCI2_AED	✓	—	—	✓	—	—
0x2D7		SCI2_BFD	✓	—	—	✓	—	—
0x2D8		SCI2_AM	✓	—	—	✓	—	—
0x2D9	SCI3	SCI3_RXI	✓	✓	✓	✓	—	—
0x2DA		SCI3_TXI	✓	✓	✓	✓	—	—
0x2DB		SCI3_TEI	✓	—	—	✓	—	—
0x2DC		SCI3_ERI	✓	—	—	✓	—	—
0x2DD		SCI3_AED	✓	—	—	✓	—	—
0x2DE		SCI3_BFD	✓	—	—	✓	—	—
0x2DF		SCI3_AM	✓	—	—	✓	—	—
0x2E0	SCI4	SCI4_RXI	✓	✓	✓	✓	—	—
0x2E1		SCI4_TXI	✓	✓	✓	✓	—	—
0x2E2		SCI4_TEI	✓	—	—	✓	—	—
0x2E3		SCI4_ERI	✓	—	—	✓	—	—
0x2E4		SCI4_AED	✓	—	—	✓	—	—
0x2E5		SCI4_BFD	✓	—	—	✓	—	—
0x2E6		SCI4_AM	✓	—	—	✓	—	—

Table 14.5 Event List (10 of 13)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x2E7	SCI5	SCI5_RXI	✓	✓	✓	✓	—	—
0x2E8		SCI5_TXI	✓	✓	✓	✓	—	—
0x2E9		SCI5_TEI	✓	—	—	✓	—	—
0x2EA		SCI5_ERI	✓	—	—	✓	—	—
0x2EB		SCI5_AED	✓	—	—	✓	—	—
0x2EC		SCI5_BFD	✓	—	—	✓	—	—
0x2ED		SCI5_AM	✓	—	—	✓	—	—
0x2EE	SCI6	SCI6_RXI	✓	✓	✓	✓	—	—
0x2EF		SCI6_TXI	✓	✓	✓	✓	—	—
0x2F0		SCI6_TEI	✓	—	—	✓	—	—
0x2F1		SCI6_ERI	✓	—	—	✓	—	—
0x2F2		SCI6_AED	✓	—	—	✓	—	—
0x2F3		SCI6_BFD	✓	—	—	✓	—	—
0x2F4		SCI6_AM	✓	—	—	✓	—	—
0x2F5	SCI7	SCI7_RXI	✓	✓	✓	✓	—	—
0x2F6		SCI7_TXI	✓	✓	✓	✓	—	—
0x2F7		SCI7_TEI	✓	—	—	✓	—	—
0x2F8		SCI7_ERI	✓	—	—	✓	—	—
0x2F9		SCI7_AED	✓	—	—	✓	—	—
0x2FA		SCI7_BFD	✓	—	—	✓	—	—
0x2FB		SCI7_AM	✓	—	—	✓	—	—
0x2FC	SCI8	SCI8_RXI	✓	✓	✓	✓	—	—
0x2FD		SCI8_TXI	✓	✓	✓	✓	—	—
0x2FE		SCI8_TEI	✓	—	—	✓	—	—
0x2FF		SCI8_ERI	✓	—	—	✓	—	—
0x300		SCI8_AED	✓	—	—	✓	—	—
0x301		SCI8_BFD	✓	—	—	✓	—	—
0x302		SCI8_AM	✓	—	—	✓	—	—
0x303	SCI9	SCI9_RXI	✓	✓	✓	✓	—	—
0x304		SCI9_TXI	✓	✓	✓	✓	—	—
0x305		SCI9_TEI	✓	—	—	✓	—	—
0x306		SCI9_ERI	✓	—	—	✓	—	—
0x307		SCI9_AED	✓	—	—	✓	—	—
0x308		SCI9_BFD	✓	—	—	✓	—	—
0x309		SCI9_AM	✓	—	—	✓	—	—

Table 14.5 Event List (11 of 13)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x30A	SCI10	SCI10_RXI	✓	✓	✓	✓	—	—
0x30B		SCI10_TXI	✓	✓	✓	✓	—	—
0x30C		SCI10_TEI	✓	—	—	✓	—	—
0x30D		SCI10_ERI	✓	—	—	✓	—	—
0x30E		SCI10_AED	✓	—	—	✓	—	—
0x30F		SCI10_BFD	✓	—	—	✓	—	—
0x310		SCI10_AM	✓	—	—	✓	—	—
0x311	SCI11	SCI11_RXI	✓	✓	✓	✓	—	—
0x312		SCI11_TXI	✓	✓	✓	✓	—	—
0x313		SCI11_TEI	✓	—	—	✓	—	—
0x314		SCI11_ERI	✓	—	—	✓	—	—
0x315		SCI11_AED	✓	—	—	✓	—	—
0x316		SCI11_BFD	✓	—	—	✓	—	—
0x317		SCI11_AM	✓	—	—	✓	—	—
0x318	SPI0	SPI0_SPRI	✓	✓	✓	✓	—	—
0x319		SPI0_SPTI	✓	✓	✓	✓	—	—
0x31A		SPI0_SPII	✓	—	—	✓	—	—
0x31B		SPI0_SPEI	✓	—	—	✓	—	—
0x31C		SPI0_SPCEND	✓	—	—	✓	—	—
0x31D	SPI1	SPI1_SPRI	✓	✓	✓	✓	—	—
0x31E		SPI1_SPTI	✓	✓	✓	✓	—	—
0x31F		SPI1_SPII	✓	—	—	✓	—	—
0x320		SPI1_SPEI	✓	—	—	✓	—	—
0x321		SPI1_SPCEND	✓	—	—	✓	—	—

Table 14.5 Event List (12 of 13)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x322	CANFD	CAN_RXF	✓	—	—	✓	—	—
0x323		CAN_GLERR	✓	—	—	✓	—	—
0x324		CAN0_RF_DMAREQ0	✓	✓	✓	✓	—	—
0x325		CAN0_RF_DMAREQ1	✓	✓	✓	✓	—	—
0x328		CAN1_RF_DMAREQ0	✓	✓	✓	✓	—	—
0x329		CAN1_RF_DMAREQ1	✓	✓	✓	✓	—	—
0x32C		CAN0_TX	✓	—	—	✓	—	—
0x32D		CAN0_CHERR	✓	—	—	✓	—	—
0x32E		CAN0_COMFRX	✓	—	—	✓	—	—
0x32F		CAN0_CF_DMAREQ	✓	✓	✓	✓	—	—
0x330		CAN0_RXMB	✓	—	—	✓	—	—
0x331		CAN1_TX	✓	—	—	✓	—	—
0x332		CAN1_CHERR	✓	—	—	✓	—	—
0x333		CAN1_COMFRX	✓	—	—	✓	—	—
0x334		CAN1_CF_DMAREQ	✓	✓	✓	✓	—	—
0x335		CAN1_RXMB	✓	—	—	✓	—	—
0x338		CAN0_MRAM_ERI	✓	—	—	✓	—	—
0x339		CAN1_MRAM_ERI	✓	—	—	✓	—	—
0x33A	I3C	I3C_RESP	✓	✓	✓	✓	—	—
0x33B		I3C_CMD	✓	✓	✓	✓	—	—
0x33C		I3C_IBI	✓	✓	✓	✓	—	—
0x33D		I3C_RX	✓	✓	✓	✓	—	—
0x33E		I3C_TX	✓	✓	✓	✓	—	—
0x33F		I3C_RCV	✓	✓	✓	✓	—	—
0x340		I3C_HRESP	✓	✓	✓	✓	—	—
0x341		I3C_HCMD	✓	✓	✓	✓	—	—
0x342		I3C_HRX	✓	✓	✓	✓	—	—
0x343		I3C_HTX	✓	✓	✓	✓	—	—
0x344		I3C_TEND	✓	—	—	✓	—	—
0x345		I3C_EEI	✓	—	—	✓	—	—
0x346		I3C_STEV	✓	—	—	✓	—	—
0x347		I3C_MREFOVF	✓	—	—	✓	—	—
0x348		I3C_MREFCPT	✓	—	—	✓	—	—
0x349		I3C_AMEV	✓	—	—	✓	—	—
0x34A		I3C_WU	✓	—	—	✓	✓	—

Table 14.5 Event List (13 of 13)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x34B	ADC16H	ADC_LIMCLPI	✓	—	—	✓	—	—
0x34C		ADC_FIFOOVF	✓	—	—	✓	—	—
0x34D		ADC_ADI0	✓	✓	✓	✓	—	—
0x34E		ADC_ADI1	✓	✓	✓	✓	—	—
0x34F		ADC_ADI2	✓	✓	✓	✓	—	—
0x350		ADC_ADI3	✓	✓	✓	✓	—	—
0x351		ADC_ADI4	✓	✓	✓	✓	—	—
0x352		ADC_ADI5	✓	✓	✓	✓	—	—
0x353		ADC_ADI6	✓	✓	✓	✓	—	—
0x354		ADC_ADI7	✓	✓	✓	✓	—	—
0x355		ADC_ADI8	✓	✓	✓	✓	—	—
0x356		ADC_FIFOREQ0	✓	✓	✓	✓	—	—
0x357		ADC_FIFOREQ1	✓	✓	✓	✓	—	—
0x358		ADC_FIFOREQ2	✓	✓	✓	✓	—	—
0x359		ADC_FIFOREQ3	✓	✓	✓	✓	—	—
0x35A		ADC_FIFOREQ4	✓	✓	✓	✓	—	—
0x35B		ADC_FIFOREQ5	✓	✓	✓	✓	—	—
0x35C		ADC_FIFOREQ6	✓	✓	✓	✓	—	—
0x35D		ADC_FIFOREQ7	✓	✓	✓	✓	—	—
0x35E		ADC_FIFOREQ8	✓	✓	✓	✓	—	—
0x35F		ADC_CMPI0	✓	—	—	✓	—	—
0x360		ADC_CMPI1	✓	—	—	✓	—	—
0x361		ADC_CCMPM0	✓	✓	✓	✓	—	—
0x362		ADC_CCMPUM0	✓	✓	✓	✓	—	—
0x363		ADC_ERR0	✓	—	—	✓	—	—
0x364		ADC_RESOVF0	✓	—	—	✓	—	—
0x366		ADC_CALEND0	✓	—	—	✓	—	—
0x367		ADC_CMPI2	✓	—	—	✓	—	—
0x368		ADC_CMPI3	✓	—	—	✓	—	—
0x369		ADC_CCMPM1	✓	✓	✓	✓	—	—
0x36A		ADC_CCMPUM1	✓	✓	✓	✓	—	—
0x36B		ADC_ERR1	✓	—	—	✓	—	—
0x36C		ADC_RESOVF1	✓	—	—	✓	—	—
0x36E		ADC_CALEND1	✓	—	—	✓	—	—
0x36F	DOC	DOC_DOPCI	✓	—	—	✓	—	—
0x371	RSIP-E50D	RSIP_TADI	✓	—	—	✓	—	—
0x397	CEU	CEU_CEUI	✓	—	—	✓	—	—

Note 1. Only supported when CMPCTL0.CSTEN = 1.
Note 2. Only the first edge detection is valid.
Note 3. Support only interrupt after DTC transfer.
Note 4. Only supported in Deep Software Standby mode 1 and Deep Software Standby mode 2.

Note 5. Only supported in Deep Software Standby mode 1.

14.4 Maskable Interrupt Operation

The ICU performs the following functions:

- Detecting interrupts
- Enabling and disabling interrupts
- Selecting interrupt request destinations such as CPU interrupt, DTC activation.

14.4.1 Interrupt Detection Selection

The interrupt controller unit selects an event source input from a peripheral function interrupt or an external terminal interrupt with IELSRn.IELS [9:0] (n = 0 to 95).

The accepted interrupt source sets the IELSRn. IR (n = 0 to 95) to 1 and sends an interrupt request to the NVIC.

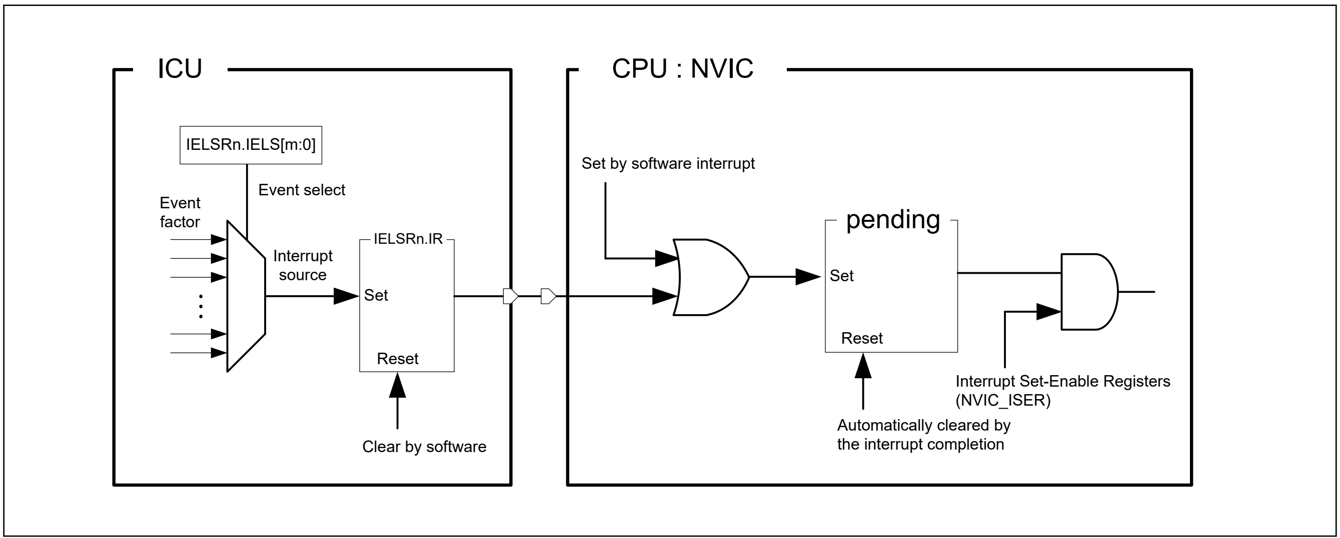


Figure 14.3 Interrupt path of the ICU and CPU : NVIC

14.4.2 Detecting Interrupts

External pin interrupt requests are detected by either the edge or level (falling edge, rising edge, rising and falling edges, or low level) of the interrupt signal. Set the IRQCRi.IRQMD[1:0] (i = 0 to 31) bits to select detection mode for the IRQi (i = 0 to 31) pins. For interrupt sources associated with peripheral modules, see section 14.3.2. Event Number.

14.5 Maskable Interrupt Setting Procedure

14.5.1 Operations During an Interrupt

- When an interrupt is generated: except for software interrupts:
  The IELSRn.IR (n = 0 to 95) flag and Interrupt Set Pending Register (NVIC_ISPRn) are set.
- When an interrupt is generated: software interrupts:
  Interrupt Set Pending Register (NVIC_ISPRn) are set.
- When an interrupt is complete:
  Clear the IELSRn.IR (n = 0 to 95) flag by software. The Interrupt Set Pending Register (NVIC_ISPRn) is automatically cleared.

Note: There may be a difference in processing speed between the CPU and ICU, and the CPU may exit the interrupt handler before clearing the IR. Then the CPU will accidentally jump to the interrupt handler again. To avoid this, be sure to read IELSRn (n = 0 to 95) register before exiting the interrupt handler and make sure that IR is cleared before exiting the interrupt handler.

14.5.2 Enabling Interrupt Requests

The procedure for enabling an interrupt request is as follows:

1. Set the Interrupt Set-Enable register (NVIC_ISER).
2. Set the IELSRn.IELS[9:0] (n = 0 to 95) bits as the interrupt source.
3. Specify the operation settings for the event source, such as Deep Sleep/Software Standby mode cancellation (WUPEN0, WUPEN1, DSLPWUPIRQENj (j = 0 to 2) register setting).

14.5.3 Disabling Interrupt Requests

The procedure to disable the interrupt request is as follows:

1. Disable the operation settings for the event source, such as Deep Sleep/Software Standby mode cancellation (WUPEN0, WUPEN1, DSLPWUPIRQENj (j = 0 to 2) register setting).
2. Clear the interrupt source setting (IELSRn.IELS[9:0] (n = 0 to 95) = 0x00).
3. Clear the interrupt status flag (IELSRn.IR (n = 0 to 95) = 0).
4. Clear the interrupt Clear-Enable register (NVIC_ICER) and interrupt Clear-Pending register (NVIC_ICPR).

14.5.4 Polling for interrupts

The procedure for polling for interrupt requests is as follows:

1. Set the Interrupt Clear-Enable register (NVIC_ICER).
2. Set the IELSRn.IELS[9:0] (n = 0 to 95) bits as the interrupt source.
3. Poll the interrupt Set-Pending register (NVIC_ISPR).

14.5.5 Selecting Interrupt Request Destinations

The interrupt output destination, CPU and DTC can be independently selected for each interrupt source.

The available destinations are fixed for each interrupt, as described in section 14.3.2. Event Number.

Use an interrupt request destination setting that is indicated by a “✓” in the event list (see section 14.3.2. Event Number).

If the CPU or DTC in one IELSRn (n = 0 to 95) register is selected, setting the same interrupt source in any other IELSRn (n = 0 to 95) register is prohibited.

Setting the same interrupt source for IELSRn (n = 0 to 95) and DELSRm is prohibited.

If the DMAC or DTC is selected as the destination for requests from an IRQi (i = 0 to 31) pin, you must set the IRQCRi.IRQMD[1:0] (i = 0 to 31) bits for that interrupt to select edge detection.

14.5.5.1 CPU Interrupt Request

When IELSRn.DTCE (n = 0 to 95) = 0, the event specified in the IELSRn (n = 0 to 95) register is output to the NVIC.

14.5.5.2 DTC Activation

When IELSRn.DTCE (n = 0 to 95) = 1, the event specified in the IELSRn register is output to the DTC. Use the following procedure:

1. Set the IELSRn.IELS[9:0] (n = 0 to 95) bits to the target event and set the IELSRn.DTCE (n = 0 to 95) bit to 1.
2. Set the DTC Module Start bit (DTCST.DTCST) to 1.

An interrupt is generated by the same factor after DTC transmission is complete.

Table 14.6 shows operation when the DTC is the interrupt request destination.

Table 14.6 Operation at DTC activation

Interrupt request destination	DISEL*1	Remaining transfer operations	Operation per request	IR*2	DTCE after transfer
DTC*3	1	≠ 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	IELSRn.DTCE = 1
		= 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	IELSRn.DTCE = 0 (IELSRn.DTCE bit is automatically cleared
	0	≠ 0	DTC transfer	Cleared at the start of DTC data transfer after reading DTC transfer data	IELSRn.DTCE = 1
		= 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	IELSRn.DTCE = 0 (IELSRn.DTCE bit is automatically cleared)

Note 1. DTC.MRB.DISEL bit controls the interrupt generates timing from DTC to CPU.
Note 2. When the IELSRn.IR (n = 0 to 95) flag is 1, an interrupt request (DTC activation request) that occurs again is ignored.
Note 3. For chain transfers, DTC transfer continues until the last chain transfer ends. The DISEL bit state and the remaining transfer count determine whether a CPU interrupt occurs, the IELSRn.IR (n = 0 to 95) flag clear timing, and the interrupt request destination after transfer. See Table 18.2 in section 18, Data Transfer Controller (DTC).

Note: Error during DTC transfer

If an error response occurs during DTC transfer, the DTC notifies the ICU that an error has occurred. ICU clears all bits of the target IELSRn (n = 0 to 95). IELSRn (n = 0 to 95) that is not the target is not cleared.

14.5.5.3 DMAC Activation

Events specified in the DELSRm (m= 0 to 7) registers are output to the DMAC.

To set the interrupt source for DMAC, use the following procedure:

1. Set the DELSRm.DELS[9:0] (m= 0 to 7) bits to the event to activate the DMAC.
2. When using interrupts to CPU, set the IELSRn.IELS (n = 0 to 95) bit to factor of DMAC interrupt and IELSRn.DTCE (n = 0 to 95) bit to 0.
3. Set the activation source for the target DMAC channel (DMACm.DMTMD.DCTG[1:0] (m= 0 to 7)) to 01b (interrupt module detection).
4. Set the DMAC transfer enable bit for the target DMAC channel (DMACm.DMCNT.DTE (m= 0 to 7)) to 1.
5. Set the DMAC operation enable bit (DMAST.DMST) to 1.

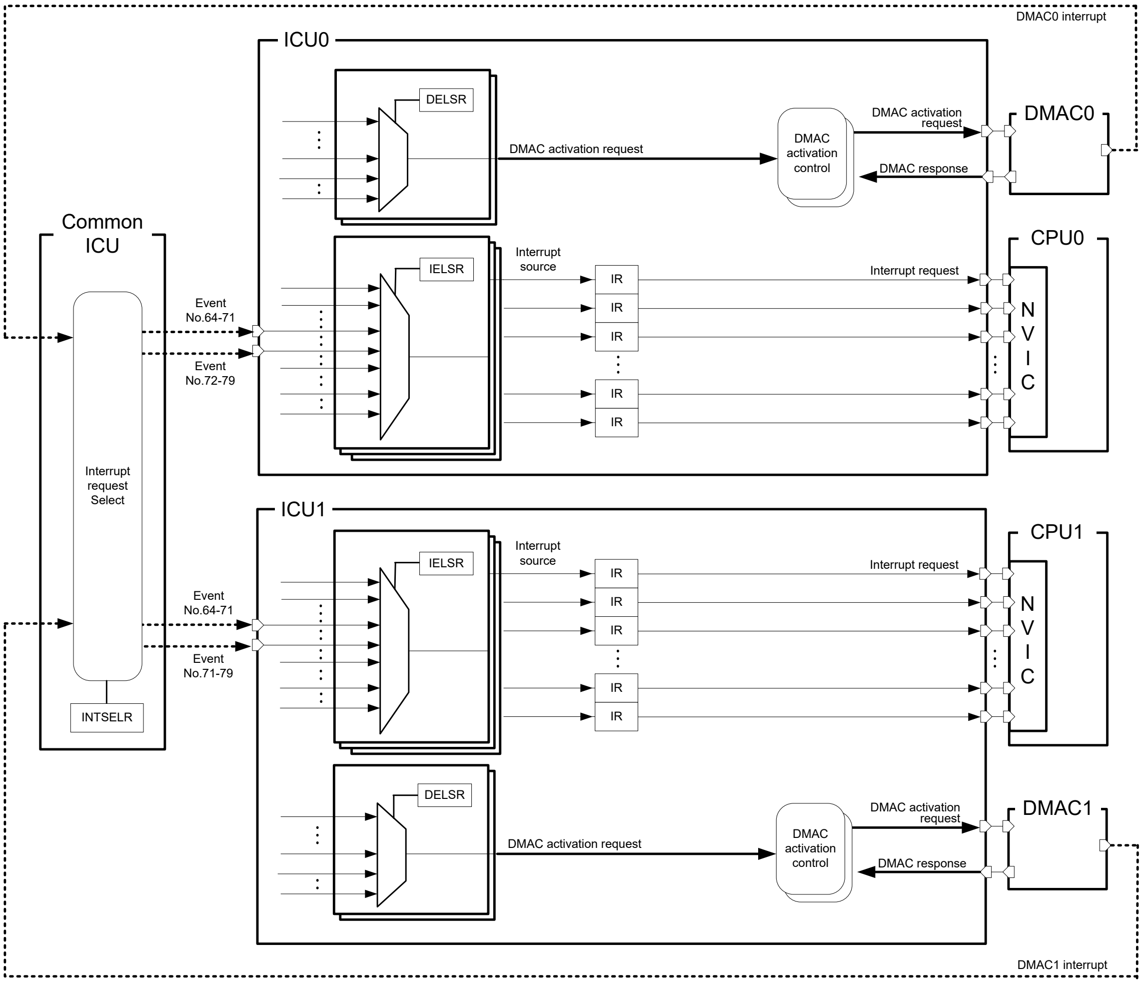


Figure 14.4 Course of the DMAC request trigger and interrupt

Note: If an error response occurs during DMAC transfer, the DMAC notifies the ICU that an error has occurred.
The ICU clears all bits of the target channel of DELSRm (m = 0 to 7). DELSRm (m = 0 to 7) that is not the target channel is not cleared.

14.5.6 Digital Filter

A digital filter function is provided for the external interrupt request pins IRQi, (i = 0 to 31) and the NMI pin interrupt. It samples input signals on the filter PCLKB sampling clock, and removes any signal with a pulse width less than 3 sampling cycles.

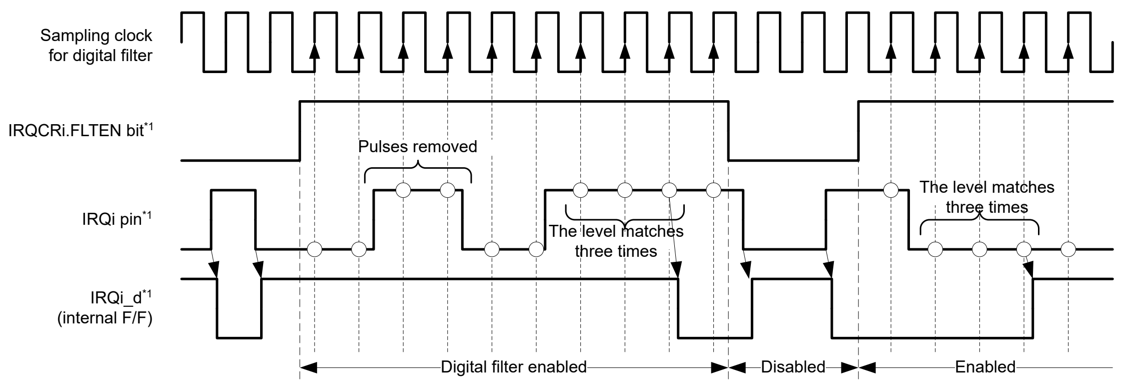
To use the digital filter for an IRQi pin:

1. Set the sampling clock cycle to PCLKB, PCLKB/8, PCLKB/32, or PCLKB/64 in the IRQCRi.FCLKSEL[1:0] bits (i = 0 to 31).
2. Set the IRQCRi.FLTEN bit (i = 0 to 31) to 1 (digital filter enabled).

To use the digital filter for an NMI pin:

1. Set the sampling clock cycle to PCLKB, PCLKB/8, PCLKB/32, or PCLKB/64 in the NMICR.NFCLKSEL[1:0] bits.
2. Set the NMICR.NFLTEN bit to 1 (digital filter enabled).

Figure 14.5 shows an example of digital filter operation.



Operation example with IRQCRi.IRQMD[1:0] = 11b (low-level detection)

Note 1. i = 0 to 15

Figure 14.5 Digital filter operation example

During Software Standby mode, the digital filter is forcibly turned off by Hardware. After returning to normal mode from Software Standby mode, follow the values of IRQCRi.FLTEN (i = 0 to 31) and NMICR.NFLTEN. Once the filter is disabled, the event information sampled up to that point will be lost.

14.5.7 External Pin Interrupts

To use external pin interrupts:Configure I/O ports settings.

1. Clear the IRQCRi.FLTEN bit (i = 0 to 31) to 0 (digital filter disabled).
2. Specify or confirm the I/O port settings.
3. Set the IRQMD[1:0] the FCLKSEL[1:0] bits, and the FLTEN bit of the IRQCRi register (i = 0 to 31) register.
4. Select the IRQ pin as follows:
   - If the IRQ pin is to be used for CPU interrupt requests, set the IELSRn.IELS[9:0] (n = 0 to 95) bits and the IELSRn.DTCE (n = 0 to 95) bit to 0.
   - If the IRQ pin is to be used for DTC activation, set the IELSRn.IELS[9:0] (n = 0 to 95) bits and the IELSRn.DTCE (n = 0 to 95) bit to 1.
   - If the IRQ pin is to be used for DMAC activation, set the DMAC. DELSRm.DELS[9:0] (m = 0 to 7) bits.

14.6 Non-Maskable Interrupt Operation

The following sources can trigger a non-maskable interrupt:

- NMI pin interrupt
- Oscillation stop detection interrupt
- Sub Oscillation stop detection interrupt
- WDT underflow/refresh error interrupt (CPU0, CPU1)
- IWDT underflow/refresh error interrupt
- Voltage monitor 1 interrupt
- Voltage monitor 2 interrupt
- Common memory error interrupt

- Local memory error interrupt (CPU1 only)
- Bus error interrupt
- Lockup error interrupt (CPU0, CPU1)
- IPC NMI CPU mutual interrupt (CPU0, CPU1)
- FPU Exception interrupt (CPU0, CPU1)
- MRAM MRC/MRE read error interrupt

Non-maskable interrupts can only be used with the CPU, not to activate the DTC. Non-maskable interrupts take precedence over all other interrupts. The non-maskable interrupt states can be verified in the Non-Maskable Interrupt Status Register (NMISR). Confirm that all bits in the NMISR are 0 before returning from the NMI handler.

There may be a difference in processing speed between the CPU and ICU, and the CPU may exit the interrupt handler before clearing the NMISR. Then the CPU will accidentally jump to the NMI handler again. To avoid this, be sure to read NMISR before exiting the NMI handler and make sure that NMISR is cleared before exiting the NMI handler.

Non-maskable interrupts are disabled by default. To use non-maskable interrupts:

1. Clear the NMICR.NFLTEN bit to 0 (digital filter disabled).
2. Set the NMIMD bit, NFCLKSEL[1:0] bits, and NFLTEN bit of NMICR register.
3. Write 1 to the NMICLR.NMICLR bit to clear the NMISR.NMIST flag to 0.
4. Enable the non-maskable interrupt by writing 1 to the associated bit in the Non-Maskable Interrupt Enable Register (NMIER).

After 1 is written to the NMIER register, subsequent write access to the NMIEN bit in NMIER is ignored. An NMI cannot be disabled when enabled, except by a reset.

14.6.1 Correspondence to TrustZone-M by NMI

The NMI security is set by AIRCR.BFHFNMINS.

Although there is only one NMI per CPU, multiple factors can be set. This section describes the procedure for mixing Secure and Non-secure programs of NMI. When doing so, the NMI-related registers of the ICU are set to Secure.

Register related to Non-Maskable Interrupt :

- NMISR
- NMIER
- NMICLR
- NMICR

Figure 14.6 shows the flow.

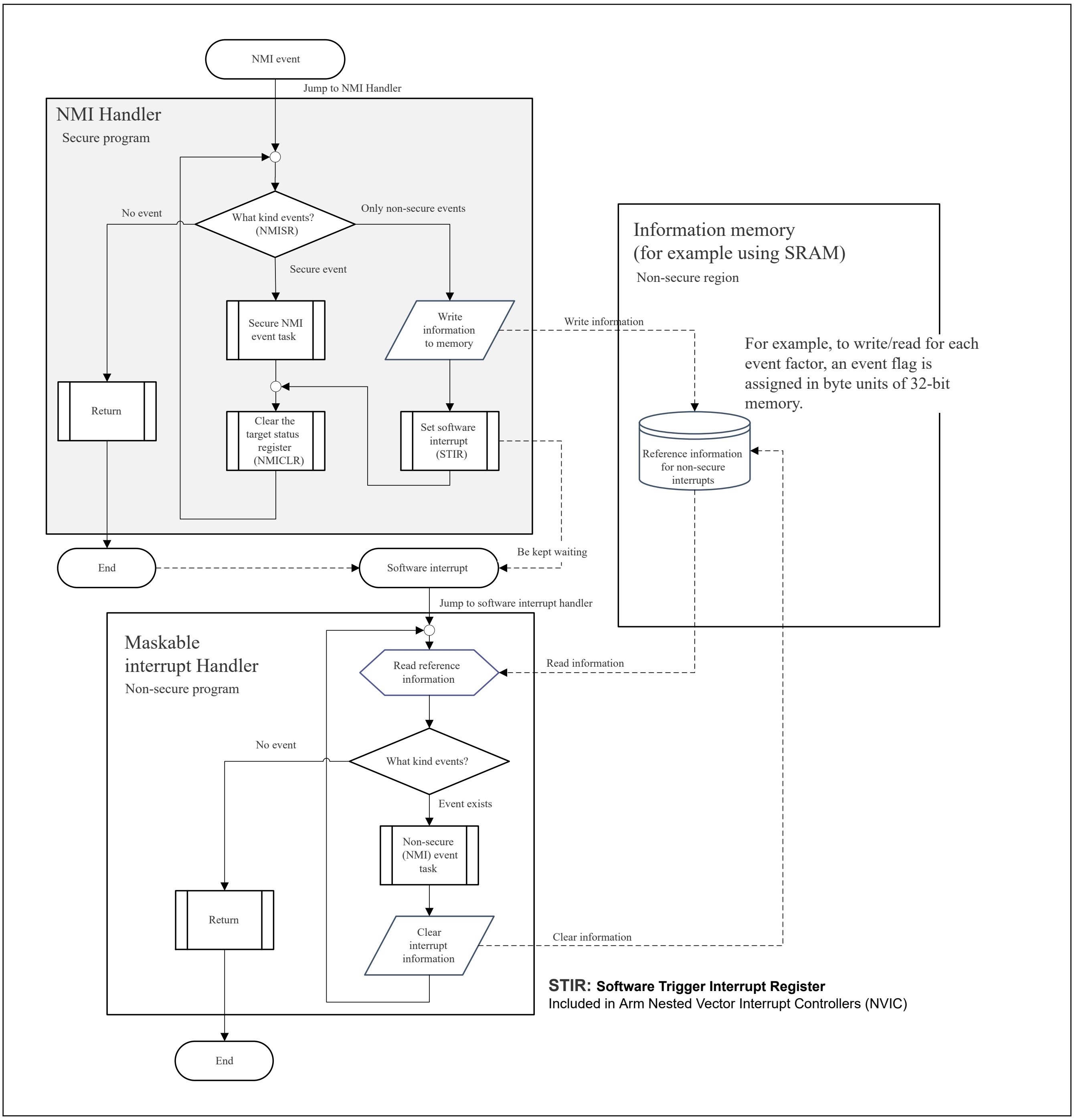


Figure 14.6 Correspondence to TrustZone-M by NMI

14.7 Inter-Processor Communication

The inter-processor communication has an interrupt function from CPU0 to CPU1 and an interrupt function from CPU1 to CPU0.

The following can be used as mutual interrupts between CPUs.

- IPC NMI CPU mutual interrupt (NMI bit20)
- IPC CPU mutual interrupt 0 (Event No.91)
- IPC CPU mutual interrupt 1 (Event No.92)

14.8 Security

14.8.1 Security Related to CPU Interrupt Inputs

Non-maskable interrupt

Arm® CPU NMI Secure is modified by AIRCR.BFHFNMINS and is managed by the software developer who manages Secure.

The following registers must follow the secure attribution set in AIRCR.BFHFNMINS.

- ICU.NMISR
- ICU.NMIER
- ICU.NMICLR

Therefore, the secure attribute set by ICUSARB must match the secure attribute set by AIRCR.BFHFNMINS.

Maskable interrupts

The secure of maskable interrupts are set in the Arm CPU NVIC internal registers (NVIC_ITNS0 to NVIC_ITNS15).

Which maskable interrupts are secured is controlled by the software developer who manages secure.

The following registers must follow the secure attribution set in NVIC_ITNS0 to NVIC_ITNS15.

- ICU.IELSRn (n = 0 to 95)

Therefore, the secure attribute set by ICUSARG, ICUSARH, and ICUSARI, ICUSARJ, ICUSARK, and ICUSARL must match the secure attribute set by NVIC_ITNS0 to NVIC_ITNS15.

14.8.2 Trusted Interrupt Management

The secure-interrupt should not be visible to non-secure-operation. Because there is a risk that vital information will be leaked to the attacker, the on-chip interrupt network can route any interrupt to secure or non-secure world. However, secure interrupt routing must be suppressed so that it can only be configured from a secure world.

When set TEVTRCR.TEVTEICU0/TEVTEICU1 = 1, secure program manages the selection of interrupt factors. It protects non-secure programs from using and monitoring secure interrupt factors without permission. Specifically, when TEVTRCR.TEVTEICU0/TEVTEICU1 = 1, the write permission of IELSR.IELS [9:0] is limited to the secure attribute, and the write by the non-secure attribute is ignored. TEVTRCR.TEVTEICU0/TEVTEICU1 protects only IELS [9:0], and the security attributes of IR and DTCE follow the settings of ICUSARG, ICUSARH, ICUSARI, ICUSARJ, ICUSARK, and ICUSARL. If TEVTRCR.TEVTEICU0/TEVTEICU1 = 1, it is necessary to set all IELS including non-secure interrupts in the secure program, or prepare a secure API to set IELS in response to the request from the non-secure program.

- Non-maskable interrupt
The non-maskable interrupt factor that can be used by CPU0 and CPU1 is determined by NMIER register.
The judgment conditions and priorities are as follows.
Condition 1: If the target bit of the NMIER register is set to only one of CPU0 and CPU1, it is supplied to the set CPU.
Condition 2: When both CPU0 and CPU1 set the target bit of the NMIER register, it is judged by the priority according to the primary/secondary of the CPU and the security level at the time of writing the NMIER register.
Priority (1) If the NMIER register on the primary CPU side is set with a secure write, the primary CPU has priority.
Priority (2) If the NMIER register on the secondary CPU side is set with a secure write, the secondary CPU has priority.
Priority (3) If the NMIER register is set with a non-secure write for both CPUs, the primary CPU has priority.
Example 1: When ICU0.NMIER.NMIEN=0, ICU1.NMIER.NMIEN= 1, supply to CPU1.
Example 2: When CPU0 is the primary CPU.
  - When both CPU0/CPU1 set the NMIER register with a secure write, it is supplied to CPU0.
  - When NMIER of CPU0 is set by non-secure write and NMIER of CPU1 is set by secure write, it is supplied to CPU1.
  - When the NMIER register is set with a non-secure write for both CPU0/CPU1, it is supplied to CPU0.

  The exception conditions are shown below.
Exception 1) Fixed to the primary CPU side, not selectable on the secondary CPU side.
  - bit0. IWDT underflow/refresh error interrupt

Exception 2) There is no distribution control between CPU0 and CPU1 due to individual CPU factors.

- bit1. WDT Underflow/Refresh Error Interrupt (Accepts for same CPUs).
- bit15. Lockup error interrupt (Accepts for same CPUs).
- bit16. FPU Exception interrupt (Accepts for same CPUs).
- bit20. IPC NMI CPU mutual interrupt (Controlled by IPC).

- Maskable interrupts
  The maskable interrupt factor that can be used by CPU0 and CPU1 is determined by INTSELRp register. (p = 0 to 31).
  The bit number of the INTSELRp register corresponds to the number of EventList.
  Setting the INTSELRp register to 0 will allow it to be used by CPU0, and setting 1 will allow it to be used by CPU1.
  However, the following factors can be used regardless of the setting value of the INTSELRp register.
  - Event No.88 DBG CTI0 interrupt (Accepts for same CPUs).
  - Event No.89 DBG CTI1 interrupt (Accepts for same CPUs).
  - Event No.91 IPC CPU mutual interrupt 0 (Controlled by IPC).
  - Event No.92 IPC CPU mutual interrupt 1 (Controlled by IPC).
  - Event No.101 FPU Exception interrupt (Accepts for same CPUs).

14.8.3 Trusted IELSR Setting Procedure

14.8.3.1 Case Where Secure Program Sets all IELSRs

Case where all settings are performed in the secure initial sequence.

Initial IELSR setting procedure after system reset release

1. Secure program sets interrupt request select (ICU.INTSELRp)
2. Secure program sets TEVTEICU0/TEVTEICU1 = 1
3. Secure program sets security attributes of all interrupts (NVIC_ITNS, Security Attribution of ICU.IELSRn.IELS)
4. Secure program selects all interrupt source (ICU.IELSRn.IELS)
5. Secure program allows secure interrupts (NVIC_ISER for secure interrupt)
6. Secure program jumps to non-secure program
7. Non-secure program allows non-secure interrupts (NVIC_ISER for non-secure interrupt)

14.8.3.2 Case Where Non-secure Program Sets IELSRn by Secure API

Case of setting upon receiving a request from a non-secure program.

Initial IELSR setting procedure after system reset release

1. Secure program sets interrupt request select (ICU.INTSELRp)
2. Secure program sets TEVTEICU0/TEVTEICU1 = 1
3. Secure program sets security attributes of all interrupts (NVIC_ITNS, Security Attribution of ICU.IELSRn.IELS)
4. Secure program selects secure interrupt source (ICU.IELSRn.IELS)
5. Secure program enables secure interrupt (NVIC_ISER for secure interrupt)
6. Secure program jumps to non-secure program
7. Non-secure program calls secure program (API call)
8. Secure program selects the cause of non-secure interrupt (ICU.IELSRn.IELS)
9. Secure program returns to non-secure program
10. Non-secure program allows non-secure interrupt (NVIC_ISER for non-secure interrupt)

14.8.3.3 IELSR Release Procedure When TEVTEICU0/TEVTEICU1 = 1

< When clearing interrupt settings of secure attribute by the secure program >

1. Clear the interrupt source setting (IELSRn.IELS = 0x00)
2. Clear the interrupt status flag (ICU.IELSRn.IR = 0)
3. Clear the Interrupt Clear Enable register (NVIC_ICPR)

<When clearing interrupt settings of non-secure attribute by the secure program >

1. Clear the interrupt source setting (IELSRn.IELS = 0x00)
2. Clear the interrupt status flag by using the non-secure alias address (IELSRn.IR = 0)
3. Clear the Interrupt Clear Enable register (NVIC_ICPR)

Secure can also rewrite NVIV_ICPRn_NS.

< When clearing interrupt settings of secure attribute by the non-secure program >

It cannot be canceled.

< When clearing interrupt settings of non-secure attribute by non-secure program>

1. Non-secure program calls secure program (API call)
2. Clear the interrupt source setting (IELSRn.IELS = 0x00)
3. Return from secure program to non-secure program
4. Clear the interrupt status flag (IELSRn.IR = 0)
5. Clear the Interrupt Clear Enable register (NVIC_ICPR)

14.9 Return from Low Power Modes

Table 14.5 lists the interrupt sources that can be used to exit Sleep, Deep Sleep or Software Standby mode. For more information, see section 11, Low Power Mode.

For recovery from Deep Software Standby mode, see chapter “Deep Software Standby Mode”.

Note: The return factor from low power mode must have a priority that the CPU can accept.

14.9.1 Return from CPU Sleep Mode

To return from CPU Sleep mode in response to an interrupt:

non-maskable interrupt

Use the NMIER register to enable the target interrupt request.

maskable interrupt

1. Select the CPU as the interrupt request destination.
2. Enable the interrupt in the NVIC.

14.9.2 Return from CPU Deep Sleep Mode

The MCU returns from CPU Deep Sleep mode using a non-maskable interrupt or a maskable interrupt. For the non-maskable interrupt factors exiting the Deep Sleep mode, see section 14.2.14. NMIER : Non-Maskable Interrupt Enable Register. For the maskable interrupt factors exiting the Deep Sleep mode, see section section 14.2.22. INTSELRp : Interrupt Request Select Register (p = 0 to 31).

To return from CPU Deep Sleep mode in response to an interrupt:

Non-maskable interrupt

Use the NMIER register to enable the target interrupt request.

Maskable interrupt

1. Select the interrupt source that enables return from CPU Deep Sleep mode.

2. Use the DSLPWUPIRQENj (j = 0 to 2) register to enable the target interrupt request.
3. Select the CPU as the interrupt request destination.
4. Enable the interrupt in the NVIC.

Interrupt requests through the IRQi (i = 0 to 31) pins that do not satisfy these conditions are not detected while the clock is stopped in CPU Deep Sleep mode. Similarly, it cannot detect a request for a non-maskable interrupt from a request source whose clock is stopped in CPU Deep Sleep mode. Therefore, those events cannot be used as return factors. For example, the CPU cannot return from software standby mode because the following NMI events do not occur.

- Lockup error interrupt (same CPU)
- FPU Exception interrupt (same CPU)

14.9.3 Return from Software Standby Mode

MCU goes to the Software Standby mode via CPU deep sleep state.

The MCU returns from Software Standby mode using a non-maskable interrupt or a maskable interrupt. For the non-maskable interrupt factors exiting the Software Standby mode, see section section 14.2.14. NMIER : Non-Maskable Interrupt Enable Register. For the maskable interrupt factors exiting the Software Standby mode, see section section 14.2.19. WUPEN0 : Wake Up Interrupt Enable Register 0, section 14.2.20. WUPEN1 : Wake Up Interrupt Enable Register 1, and section 14.2.22. INTSELRp : Interrupt Request Select Register (p = 0 to 31).

The maskable interrupt sources to exit the software standby mode are specified by the bits in WUPEN0 and WUPEN1. Additionally, the corresponding bits in DSLPWUPIRQENj (j = 0 to 2) need to be set to “1”.

However, if the mode is switched to S/W standby mode while the return factor in CPU Deep sleep mode is detected, the re-turn factor from deep sleep may cause the S/W standby mode to return to normal mode.

No event occurs from the function that stops in Software Standby mode. Therefore, those events cannot be used as return factors. For example, the CPU cannot return from Software Standby mode because the following NMI events do not occur.

- WDT underflow/refresh error (CPU0, CPU1)
- Common memory error interrupt
- Local memory error interrupt (CPU1 only)
- Bus error interrupt
- Lockup error interrupt (CPU0, CPU1)
- IPC NMI CPU mutual interrupt (CPU0, CPU1)
- FPU Exception interrupt (CPU0, CPU1)
- MRAM MRC/MRE read error interrupt

14.10 Using the WFI Instruction with Non-Maskable Interrupts

Whenever a WFI instruction is executed, confirm that all status flags in the NMISR register are 0.

14.11 Reference

[1]ARM Limited., ARM® Cortex®-M33 Processor Technical Reference Manual (ARM 100230)

[2]ARM Limited., TrustZone® technology for ARM®v8-M Architecture (ARM 100690_0100_00_en)

[3]ARM Limited., ARM® Cortex®-M85 Processor Technical Reference Manual (ARM 101924_0000_02_en)

15. Buses

15.1 Overview

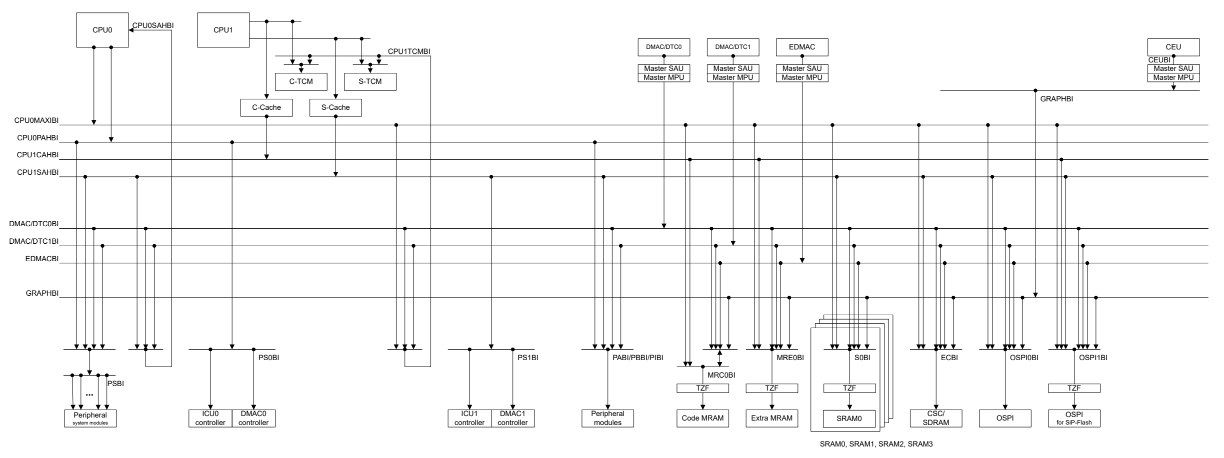
EDMAC in this chapter means the GWCA function of ESWM. Table 15.1 lists the bus masters and bus slaves, Figure 15.1 shows the system bus configuration, and Table 15.3 lists the addresses assigned for each bus.

Table 15.1 Bus specifications

Bus master/ slave name	Sync clock	Description
Classification: Bus masters		
CPU0MAXIBI (Arm® Cortex®-M85)	CPUCLK0	Connected to the CPU0 Master-AXI (M-AXI) Interface
CPU0PAHBI (Arm® Cortex®-M85)	CPUCLK0	Connected to the CPU0 Peripheral AHB (P-AHB) Interface
CPU1CAHBI (Arm® Cortex®-M33)	CPUCLK1	Connected to the CPU1 Instruction Cache for instructions and operands
CPU1SAHBI (Arm® Cortex®-M33)	CPUCLK1	Connected to the CPU1 Data Cache for system
DMAC/DTC0BI	ICLK	Connected to the DMAC/DTC0 Interface
DMAC/DTC1BI	ICLK	Connected to the DMAC/DTC1 Interface
EDMACBI	ICLK	Connected to the EDMAC Interface
CEUBI	PCLKA	Connected to the CEU Interface (Write Only)
Classification: Bus slaves		
MRC0BI	MRICLK	Connected to Code MRAM
MRE0BI	MRPCLK	Connected to Extra MRAM (Read Only)
CPU0SAHBI	ICLK	Connected to CPU0 S-AHB
CPU1TCMBI	CPUCLK1	Connected to CPU1 S-TCM and CPU1 C-TCM
S0BI	ICLK	Connected to SRAM0
S1BI	ICLK	Connected to SRAM1
S2BI	ICLK	Connected to SRAM2
S3BI	ICLK	Connected to SRAM3
ECBI	BCLK	Connected to the external devices (External Memory Interface)
OSPI0BI	ICLK	Connected to the OSPI (via DOTF) (External Memory Interface)
OSPI1BI	ICLK	Connected to the OSPI for SIP-Flash (External Memory Interface)
PBBI	PCLKB	Connected to peripheral modules synchronizes with PCLKB
PABI	PCLKA	Connected to peripheral modules synchronizes with PCLKA
PIBI	ICLK	Connected to peripheral modules synchronizes with ICLK
PS0BI	ICLK	Connected to ICU0 controller Connected to DMAC/DTC0 controller
PS1BI	ICLK	Connected to ICU1 controller Connected to DMAC/DTC1 controller
PSBI	CPUCLK1	Connected to peripheral system module. (CPU1-cache controller)
	ICLK	Connected to peripheral system modules. (MPU, CSC/SDRAM, SRAM, Debug component, Bus controller, common ICU controller, CPU controller, debug controller, IPC controller, and security attribution controller)
	MRPCLK	Connected to peripheral system module. (MRAM controller)
	PCLKB	Connected to peripheral system module. (system controller)

Note: CPU0MAXIBI: CPU0 M-AXI Bus Interface

CPU0PAHBI: CPU0 P-AHB Bus Interface
CPU1CAHBI: CPU1 C-AHB Bus Interface
CPU1SAHBI: CPU1 S-AHB Bus Interface
DMAC/DTC0BI: DMAC/DTC0 Bus Interface
DMAC/DTC1BI: DMAC/DTC1 Bus Interface
EDMACBI: EDMAC Bus Interface
CEUBI: CEU Bus Interface
MRC0BI: Code MRAM Bus Interface
MRE0BI: Extra MRAM Bus Interface
CPU0SAHBI: CPU0 S-AHB Interface (Bus Interface from Bus Masters except CPU0 to CPU0 ITCM and CPU0 DTCM)
CPU1TCMBI: Bus Interface from Bus Masters except CPU1 to CPU1 S-TCM and CPU1 C-TCM
S0BI: SRAM0 Bus Interface
S1BI: SRAM1 Bus Interface
S2BI: SRAM2 Bus Interface
S3BI: SRAM3 Bus Interface
ECBI: External memory CSC and SDRAM Bus Interface
OSPI0BI: External memory OSPI Bus Interface
OSPI1BI: External memory OSPI for SIP-Flash Bus Interface
PBBI: Peripheral synchronizes with PCLKB Bus Interface
PABI: Peripheral synchronizes with PCLKA Bus Interface
PIBI: Peripheral synchronizes with ICLK Bus Interface
PS0BI: ICU0, DMAC/DTC0 Controller Bus Interface
PS1BI: ICU1, DMAC/DTC1 Controller Bus Interface
PSBI: Peripheral System Bus Interface



Note: “TZF” is Slave TrustZone Filter.
Note: GRAPHBI is the bus interface that CEUBI is connected.

Figure 15.1 System bus connection

Table 15.2 System bus access path

		Master							
	Name	CPU0MAXI BI	CPU0PAHB_	CPU1CAHB_	CPU1SAHB_	DMAC/ DTC0BI	DMAC/ DTC1BI	EDMACBI	CEUBI (W-only)
Slave	MRC0BI (Read)	T	F	T	F	T	T	T	F
	MRC0BI (Write)	T	F	T	F	T	T	F	F
	MRE0BI (Read)	T	F	T	F	T	T	T	F
	CPU0SAHBI	F	F	F	T	T	T	F	F
	CPU1TCMBI	T	F	F	F	T	T	F	F
	S0BI	T	F	F	T	T	T	T	T
	S1BI	T	F	F	T	T	T	T	T
	S2BI	T	F	F	T	T	T	T	T
	S3BI	T	F	F	T	T	T	T	T
	ECBI	T	F	F	T	T	T	T	T
	OSPI0BI	T	F	F	T	T	T	T	T
	OSPI1BI	T	F	T	T	T	T	T	T
	PBBI	F	T	F	T	T	T	F	F
	PABI	F	T	F	T	T	T	F	F
	PIBI	F	T	F	T	T	T	F	F
	PS0BI	F	T	F	F	F	F	F	F
	PS1BI	F	F	F	T	F	F	F	F
	PSBI	F	T	F	T	T	T	F	F

T: The master can access slaves

F: The master cannot access slaves

Table 15.3 Addresses assigned for each slave (1 of 3)

Region	Address	Bus Interface	Area
Code	0x0200_0000 to 0x0207_FFFF (Non-secure callable or Secure) 0x1200_0000 to 0x1207_FFFF (Non-secure)	MRC0BI	Code MRAM
	0x0208_0000 to 0x020F_FFFF (Non-secure callable or Secure) 0x1208_0000 to 0x120F_FFFF (Non-secure)	MRC0BI[*2]	Code MRAM
	0x02A0_0000 to 0x02CF_FFFF (Non-secure callable or Secure) 0x02E0_0000 to 0x02FF_FFFF (Non-secure callable or Secure) 0x12A0_0000 to 0x12CF_FFFF (Non-secure) 0x12E0_0000 to 0x12FF_FFFF (Non-secure)	MRE0BI	Extra MRAM
	0x0800_0000 to 0x0FFF_FFFF (Non-secure callable or Secure) 0x1800_0000 to 0x1FFF_FFFF (Non-secure)	OSPI1BI	OSPI for SIP-Flash

Table 15.3 Addresses assigned for each slave (2 of 3)

Region	Address	Bus Interface	Area
Data	0x2200_0000 to 0x2207_FFFF (Non-secure callable or Secure) 0x221A_0000 to 0x221A_FFFF(Non-secure callable or Secure) 0x3200_0000 to 0x3207_FFFF (Non-secure) 0x321A_0000 to 0x321A_FFFF(Non-secure)	S0BI	SRAM0
	0x2208_0000 to 0x220F_FFFF (Non-secure callable or Secure) 0x221B_0000 to 0x221B_FFFF(Non-secure callable or Secure) 0x3208_0000 to 0x320F_FFFF (Non-secure) 0x321B_0000 to 0x321B_FFFF(Non-secure)	S1BI	SRAM1
	0x2210_0000 to 0x2217_FFFF (Non-secure callable or Secure) 0x221C_0000 to 0x221C_FFFF (Non-secure callable or Secure) 0x3210_0000 to 0x3217_FFFF (Non-secure) 0x321C_0000 to 0x321C_FFFF (Non-secure)	S2BI	SRAM2
	0x2218_0000 to 0x2219_FFFF (Non-secure callable or Secure) 0x221D_0000 to 0x221D_3FFF (Non-secure callable or Secure) 0x3218_0000 to 0x3219_FFFF (Non-secure) 0x321D_0000 to 0x321D_3FFF (Non-secure)	S3BI	SRAM3
	0x2800_0000 to 0x2803_FFFF (Non-secure callable or Secure) 0x3800_0000 to 0x3803_FFFF (Non-secure)	CPU0SAHBI	CPU0 ITCM/DTCM
	0x2A00_0000 to 0x2A01_FFFF (Non-secure callable or Secure) 0x3A00_0000 to 0x3A01_FFFF (Non-secure)	CPU1TCMBI	CPU1 CTCM/STCM
Peripheral	0x4000_0000 to 0x4000_1FFF (Secure) 0x5000_0000 to 0x5000_1FFF (Non-secure)	PSBI	MPU controller
	0x4000_2000 to 0x4000_2FFF (Secure) 0x5000_2000 to 0x5000_2FFF (Non-secure)		SRAM controller
	0x4000_3000 to 0x4000_3FFF (Secure) 0x5000_3000 to 0x5000_3FFF (Non-secure)		CSC/SDRAM controller
	0x4000_4000 to 0x4000_4FFF (Secure) 0x5000_4000 to 0x5000_4FFF (Non-secure)		BUS controller
	0x4000_6000 to 0x4000_6FFF (Secure) 0x5000_6000 to 0x5000_6FFF (Non-secure)		Common ICU controller
	0x4000_8000 to 0x4000_8FFF (Secure) 0x5000_8000 to 0x5000_8FFF (Non-secure)		Security attribution controller
	0x4000_A000 to 0x4000_AFFF (Secure) 0x5000_A000 to 0x5000_AFFF (Non-secure) 0x4000_C000 to 0x4000_CFFF (Secure) 0x5000_C000 to 0x5000_CFFF (Non-secure)	PS0BI PS1BI	DMAC/DTC controller ICU controller
	0x4000_F000 to 0x4000_FFFF (Secure) 0x5000_F000 to 0x5000_FFFF (Non-secure)	PSBI	CPU controller
	0x4001_0000 to 0x4001_AFFF (Secure) 0x5001_0000 to 0x5001_AFFF (Non-secure)		Debug component
	0x4001_B000 to 0x4001_BFFF (Secure) 0x5001_B000 to 0x5001_BFFF (Non-secure)		Debug controller
	0x4001_C000 to 0x4001_CFFF (Secure) 0x5001_C000 to 0x5001_CFFF (Non-secure)		CPU1 cache controller
	0x4001_E000 to 0x4001_EFFF (Secure) 0x5001_E000 to 0x5001_EFFF (Non-secure)		System controller
	0x4002_0000 to 0x4002_0FFF (Secure) 0x5002_0000 to 0x5002_0FFF (Non-secure)		IPC controller
	0x4012_0000 to 0x4013_FFFF (Secure) 0x5012_0000 to 0x5013_FFFF (Non-secure)		MRAM controller

Table 15.3 Addresses assigned for each slave (3 of 3)

Region	Address	Bus Interface	Area
	0x4020_0000 to 0x402F_FFFF (Secure) 0x5020_0000 to 0x502F_FFFF (Non-secure)	PBBI	Peripheral synchronizes with PCLKB
	0x4030_0000 to 0x403F_FFFF (Secure) 0x5030_0000 to 0x503F_FFFF (Non-secure)	PABI	Peripheral synchronizes with PCLKA
	0x4040_0000 to 0x404F_FFFF (Secure) 0x5040_0000 to 0x504F_FFFF (Non-secure)	PIBI	Peripheral synchronizes with ICLK
External RAM	0x6000_0000 to 0x6FFF_FFFF	ECBI	CSC/SDRAM (external bus)
	0x7000_0000 to 0x7FFF_FFFF	OSPI1BI*1	OSPI (via DOTF)
	0x8000_0000 to 0x9FFF_FFFF	OSPI0BI	OSPI (via DOTF)

Note 1. Products without SIP-FLASH can be accessed, but products with SIP-FLASH cannot.
Note 2. MRAM size 1 MB product is accessible, but MRAM size 512 KB product is not.

15.2 Description of Buses

15.2.1 Arbitration

For arbitration between masters in each slave, fixed-priority and round-robin methods can be selected. For details, see section 15.3.23. BUSSABT0<slave> : Bus Slave Arbitration Control Register 0 and section 15.3.24. BUSSABT1MRC0BI : Bus Slave Arbitration Control Register 1.

15.2.2 External Bus

The external bus controller arbitrates requests for bus access on the external address space from the CPU M-AXI bus, DMAC/DTC bus, EDMAC bus and CEU bus. See section 15.2.1. Arbitration for the priority and arbitration method of each master to the External bus.

The bus system provides an external space for the OSPI (OSPI0BI,OSPI1BI). See section 44, Octal Serial Peripheral Interface (OSPI).

Table 15.4 lists the external bus specifications and Table 15.5 lists the I/O pins.

The external bus controller transfers between the AXI master and the external bus over the FIFO.

In External address space access, the internal bus is released by capturing the bus access request with the FIFO before the access to the external bus is complete. This allows the internal bus to start the next bus access. However, if the FIFO is full and the next access is an external address space access, it will wait for the previous external bus operation to complete.

Figure 15.2 shows an example of operation when the FIFO function is in use. When this function is in use, if the next operation is peripheral module access after external device access, peripheral module access is executed in parallel without waiting for the completion of external device access.

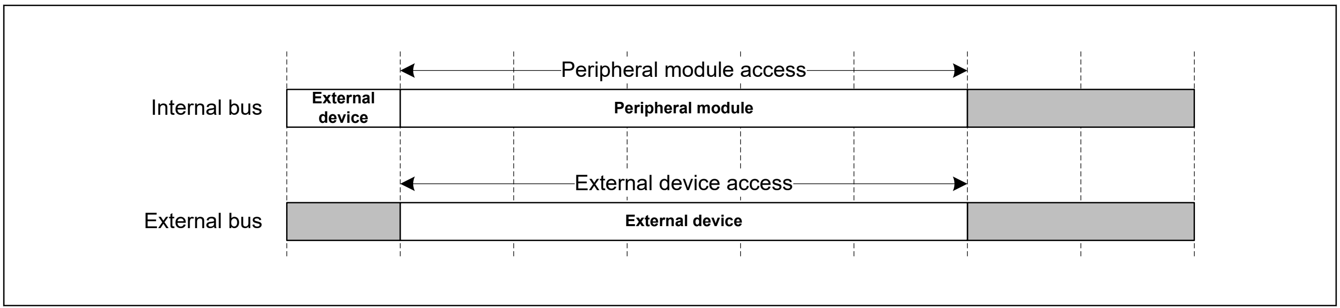


Figure 15.2 Example operation when the FIFO function is in use

Table 15.4 External bus specifications

Parameter	Specifications

External address space	• The external address space is divided into 8 CS areas (CS0 to CS7) and the SDRAM area (SDCS) for management.
	• Chip select signals can be output for each area.
	• The bus width can be set for each area.
	  – Separate bus: Selectable to 8-bit, 16-bit or 32-bit bus space
	  – Address/data multiplexed bus: Selectable to 8-bit or 16-bit bus space
	• Endian mode can be specified for each area.
	Big endian mode has the following restrictions. See section 15.2.5.1. Endianness Constraint for details.

CS area controller	• Recovery cycles can be inserted:
	  – Read recovery: Up to 15 cycles
	  – Write recovery: Up to 15 cycles
	• Cycle wait function: Wait for up to 31 cycles (for page access, up to 7 cycles)
	• Use wait control to set up:
	  – Assertion and negation timing of chip select signals (CS0 to CS7)
	  – Assertion timing of the read signal (RD) and write signals (WR0/WR and WR1)
	  – Timing of data output starts and ends
	• Write access modes:
	  – Single-write strobe mode and byte strobe mode
	Separate bus or address/data multiplexed bus can be set for each area.

SDRAM area controller	• Multiplexed output of row address and column address (8, 9, 10, or 11 bits) • Self-refresh and auto-refresh selectable • CAS latency can be specified from 1 to 3 cycles.
FIFO	FIFO number : 16 for Read Data and Write Data
Frequency	• The CS area controller (CSC) operates in synchronization with the external bus clock (BCLK)*1 • The frequency of the EBCLK pin output is the same as BCLK by default. Half of the BCLK cycles can be supplied by setting the EBCLK Pin Output Select bit, BCKCR.BCLKDIV, in the External Bus Clock Control Register. For more information, see section 9, Clock Generation Circuit. • The SDRAM area controller (SDRAMC) operates in synchronization with the SDRAM clock (SDCLK).

Note 1. BCLK and SDCLK must operate at the same frequency when the SDRAM is in use.

Table 15.5 External bus I/O pins (1 of 2)

Pin name	I/O	Related functions	Description
EBCLK, SDCLK*1	Output	CSC, SDRAMC	Clock output pin
A23 to A00*2	Output	CSC, SDRAMC	Address output pins
D31 to D00 DQ31 to DQ00	I/O	CSC, SDRAMC	D31 to D00 are CSC data input/output pins. DQ31 to DQ00 are SDRAMC data input/output pins. • D31 to D00, DQ31 to DQ00 pins are enabled when the 32-bit bus space is specified. • D15 to D00, DQ15 to DQ00 pins are enabled when the 16-bit bus space is specified. • D07 to D00, DQ07 to DQ00 pins are enabled when the 8-bit bus space is specified.
BC0	Output	CSC	• Strobe signal that indicates (when low) that D07 to D00 are valid during access to an external address space in single-write strobe mode, active-low. • When an 8-bit bus space is specified, this output pin is always held low regardless of the write access mode.
BC1	Output	CSC	• Strobe signal that indicates (when low) that D15 to D08 are valid during access to an external address space in single-write strobe mode, active-low. • This pin is not used when the 8-bit bus space is specified.
BC2	Output	CSC	• Strobe signal that indicates (when low) that D23 to D16 are valid during access to an external address space in single-write strobe mode, active-low. • This pin is not used when the 8- or 16-bit bus space is specified.
BC3	Output	CSC	• Strobe signal that indicates (when low) that D31 to D24 are valid during access to an external address space in single-write strobe mode, active-low. • This pin is not used when the 8- or 16-bit bus space is specified.
CS0*3	Output	CSC	Chip select signal for area 0 (CS0), active-low
CS1*3	Output	CSC	Chip select signal for area 1 (CS1), active-low

Table 15.5 External bus I/O pins (2 of 2)

Pin name	I/O	Related functions	Description
CS2*3	Output	CSC	Chip select signal for area 2 (CS2), active-low
CS3*3	Output	CSC	Chip select signal for area 3 (CS3), active-low
CS4	Output	CSC	Chip select signal for area 4 (CS4), active-low
CS5	Output	CSC	Chip select signal for area 5 (CS5), active-low
CS6	Output	CSC	Chip select signal for area 6 (CS6), active-low
CS7	Output	CSC	Chip select signal for area 7 (CS7), active-low
RD	Output	CSC	Strobe signal that indicates that a read from an external address space (CS0 to CS7) is in progress, active-low.
WR0/WR*4	Output	CSC	● WR0 signal is a strobe signal that indicates that a write to an external address space is in progress in byte strobe mode, and D07 to D00 are valid, active-low. ● WR signal is a strobe signal that indicates that a write to an external address space is in progress in single-write strobe mode, active-low. ● When an 8-bit bus space is specified, this output pin is held low during a write access regardless of the write access mode.
WR1	Output	CSC	● Strobe signal that indicates that D15 to D08 are valid during a write to an external address space in byte strobe mode, active-low. ● This signal is invalid in single-write strobe mode. ● This pin is not used when the 8-bit bus space is specified.
WR2	Output	CSC	● Strobe signal that indicates that D23 to D16 are valid during a write to an external address space in byte strobe mode, active-low. ● This signal is invalid in single-write strobe mode. ● This pin is not used when the 8- or 16-bit bus space is specified.
WR3	Output	CSC	● Strobe signal that indicates that D31 to D24 are valid during a write to an external address space in byte strobe mode, active-low. ● This signal is invalid in single-write strobe mode. ● This pin is not used when the 8- or 16-bit bus space is specified.
ALE	Output	CSC	Address latch signal when address/data multiplexed bus is selected.
WAIT	Input	CSC	Wait request signal used when accessing the external address space (CS0 to CS7), active-low
CKE	Output	SDRAMC	Clock enable signal
SDCS	Output	SDRAMC	Chip select signal, active-low
RAS	Output	SDRAMC	Row address strobe signal, active-low
CAS	Output	SDRAMC	Column address strobe signal, active-low
WE	Output	SDRAMC	Write enable signal, active-low
DQM0	Output	SDRAMC	I/O data mask enable signal for DQ07 to DQ00
DQM1	Output	SDRAMC	I/O data mask enable signal for DQ15 to DQ08
DQM2	Output	SDRAMC	I/O data mask enable signal for DQ23 to DQ16
DQM3	Output	SDRAMC	I/O data mask enable signal for DQ31 to DQ24

Note 1. The EBCLK and the SDCLK pin functions are shared by the CS area controller (CSC) and the SDRAM area controller (SDRAMC). When using the CSC and the SDRAMC simultaneously, the SDCLK pin function is valid.

Note 2. The A23 to A00 pin functions are shared by the CSC and the SDRAMC.
When using the CSC only:
The A00 and BC0 pin functions share the same pin, and either becomes valid according to the area, with the function being A00 in byte strobe mode and BC0 in single-write strobe mode. Setting the 8-bit external bus width is prohibited in single-write strobe mode.
When using the SDRAMC only:
The A15 to A00 pin functions are valid.
The A00 and DQM1 pin functions share the same pin, and either pin function becomes valid according to the external bus width. When selecting 8-bit bus width, the pin function is A00. When selecting 16-bit bus width, the pin function is DQM1.
When using the CSC and the SDRAMC simultaneously:
The A23 to A16 pin functions are valid for CSC. The A15 to A00 pin functions are shared by the CSC and the SDRAMC. In the SDRAMC functions, the A00 and the DQM1 pin functions work as described.
In the CSC functions, the A00 and the BC0 pin function works as described.

Note 3. The CS0 to CS3 (CSC) and SDRAMC pin functions share the same pin. When using the CSC and the SDRAMC simultaneously, the CS0 to CS3 pin functions are invalid.
Note 4. The WR0 signal and WR signal are identical. The WR0 signal is referred to as WR in single-write strobe mode.

15.2.3 Parallel Operation

Parallel operation is possible when different bus-master modules are requesting access to different slave interface. For example, if the CPU is fetching an instruction from Code Flash and an operand from MRAM, the DMAC can handle transfers between a peripheral module and the external memory interface at the same time.

Figure 15.3 shows an example of parallel operations. In this example, the CPU uses CPUMAXIBI to access MRC0BI. Furthermore, the DMAC/DTC simultaneously accesses the peripheral bus or the external bus during access to MRC0BI by the CPU.

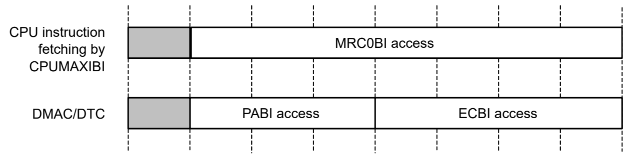


Figure 15.3 Example of parallel operations

15.2.4 Bus Settings

Set up the external bus with the following registers :

- Mode settings :
  CSn Mode Register (CSnMOD)
  CSn Wait Control Register 1 (CSnWCR1)
  CSn Wait Control Register 2 (CSnWCR2)
  CSn Control Register (CSnCR)
  CSn Recovery Cycle Setting Register (CSnREC)
  CS Recovery Cycle Insertion Enable Register (CSRECEN)
- I/O port assignments :
  PmnPFS.PMR = 1 and PmnPFS.PSEL[4:0] = 0x0B
- Frequency of the external bus clock (BCLK) and SDRAM clock (SDCLK) :
  SCKDIVCR register.

See section 20, I/O Ports, for information on PmnPFS and section Clock Generation Function for information on SCKDIVCR.

15.2.5 Restrictions

15.2.5.1 Endianness Constraint

Memory space must be little-endian to execute code of the Arm Cortex-M85 and Arm Cortex-M33core.

Big endian mode has the following restrictions.

- Only CPU0, CPU1, DMAC/DTC0 and DMAC/DTC1 can access to big endian area.
- The Arm Cortex-M85 core and Arm Cortex-M33 core is fixed at little-endian order, so instruction code can only be allocated to external spaces with little-endian specified. If an area is specified as big-endian, no instruction code can be allocated to it.
- No unaligned access
- Accesses with different sizes to the same address are prohibited.
- 64-bit data access is not possible in the case of areas with big-endian setting.
- The memory type must be 'device memory'. (Sparse-write is prohibited)

- For changing the memory type, see reference [1].
- CPU0 (Cortex-M85 core) cannot access to big-endian area when using M-profile Vector Extension (MVE).

15.2.5.2 Bufferable Write Access

In the following bufferable write access, the write response comes from the midpoint of the bus, not from the endpoint of the write access.

1. CPU0 M-AXI accesses CPU1TCMBI.
2. CPU0 P-AHB accesses any area.
3. CPU1 C-AHB accesses any area.
4. CPU1 S-AHB accesses any area.
5. DMA accesses PABI, PBBI, PIBI, PSBI, CPU0SAHBI and CPU1TCMBI.

If an access violation occurs due to bufferable write access, an error response is not returned to the bus master, but an Interrupt request or Reset request is generated depending on the setting of the BUSOAD register. For more information, see section 15.7.2. Operations When a Bus Error Occurs.

15.2.5.3 Notes on Writing Completion Timing

For the following accesses, a write response is returned to the bus master before the write is completed, regardless of the bufferable attribute.

When writing to MRC0BI, the write response is returned from the write buffer for MRAM, not from the MRAM itself, regardless of the bufferable/non-bufferable setting.

In the write access to ECBI, the write response comes from the CSC/SDRAMC controller when the write access start s to CS/SDRAM regardless of whether it is bufferable or non-bufferable setting.

15.3 Register Descriptions

15.3.1 BUSSARA : Bus Security Attribution Register A

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x100

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BUSSA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BUSSA0	Bus Security Attribution A0 0: Secure 1: Non-secure	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

BUSSA0 bit (Bus Security Attribution A0)

Security attributes for arbitration control register. The target registers are as follows:

- BUSSABT0<slave>

<slave> = MRE0BI, S0BI, S1BI, S2BI, S3BI, ECBI, OSPI0BI, OSPI1BI, CPU0SAHBI, CPU1TCMBI, PBBI, PABI, PIBI, PSBI

- BUSSABT1MRC0BI

15.3.2 BUSSARB : Bus Security Attribution Register B

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x104

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BUSSB0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BUSSB0	Bus Security Attribution B0 0: Secure 1: Non-secure	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

BUSSB0 bit (Bus Security Attribution B0)

Security attributes of bus error related registers. The target registers are as follows:

- BUSERRCLR<Master Name>
- MBWERRCLR
- BUSOAD
- BUSOADPT
- BUSIRQEN<Master Name>

<Master Name>: CPU0/CPU1/DMAC0/DMAC1/EDMAC/CEU

15.3.3 BUSSARC : Bus Security Attribution Register C

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x110

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BUSSC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BUSSC0	Bus Security Attribution C0 0: Secure 1: Non-secure	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

BUSSC0 bit (Bus Security Attribution C0)

Security attributes of registers for bus (SDRAMC/CSC) Control. The target registers are as follows:

- BUS.CSnCR (n = 0 to 7)
- BUS.CSnREC (n = 0 to 7)
- BUS.CSRECEN
- BUS.CSnMOD (n = 0 to 7)
- BUS.CSnWCR1 (n = 0 to 7)
- BUS.CSnWCR2 (n = 0 to 7)
- BUS.SDCCR
- BUS.SDCMOD
- BUS.SDAMOD
- BUS.SDSELF
- BUS.SDRFCR
- BUS.SDRFEN
- BUS.SDICR
- BUS.SDIR
- BUS.SDADR
- BUS.SDTR
- BUS.SDMOD
- BUS.SDSR
- CPSCU.BUSPARC

15.3.4 BUSPARC : Bus Privileged Attribution Register C

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x114

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BUSPA0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	BUSPA0	External bus controller privilege attribution 0: Privileged 1: Unprivileged	R/W

Bit	Symbol	Function	R/W
31:1	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-2, P-TYPE-1

BUSPA0 bit (External bus controller privilege attribution)

Privileged attributes of registers for Bus (SDRAMC/CSC) Control. The target registers are as follows:

- BUS.CSnCR (n = 0 to 7)
- BUS.CSnREC (n = 0 to 7)
- BUS.CSRECEN
- BUS.CSnMOD (n = 0 to 7)
- BUS.CSnWCR1 (n = 0 to 7)
- BUS.CSnWCR2 (n = 0 to 7)
- BUS.SDCCR
- BUS.SDCMOD
- BUS.SDAMOD
- BUS.SDSELF
- BUS.SDRFCR
- BUS.SDRFEN
- BUS.SDICR
- BUS.SDIR
- BUS.SDADR
- BUS.SDTR
- BUS.SDMOD
- BUS.SDSR

15.3.5 CSnCR : CSn Control Register (n = 0 to 7)

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x802 + 0x10 × n (n = 0 to 7)

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

Bit	Symbol	Function	R/W
0	EXENB	Operation Enable 0: Disable operation 1: Enable operation	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
5:4	BSIZE[1:0]	External Bus Width Select 0 0: 16-bit bus space 0 1: 32-bit bus space 1 0: 8-bit bus space Others: Setting prohibited	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
8	EMODE	Endian Mode 0: Little endian 1: Big endian	R/W
11:9	—	These bits are read as 0. The write value should be 0.	R/W
12	MPXEN	Address/Data Multiplexed I/O Interface Select 0: Separate bus interface is selected for area n. 1: Address/data multiplexed I/O interface is selected for area n.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. When n = 0, the value after reset is 1.

The BSIZE, EMODE, and MPXEN bits are not rewritable when the EXENB bit is set to 1. It is possible to rewrite at the same time as the EXENB bit.

Do not attempt to write the CSnCR register while the external bus is being accessed.

EXENB bit (Operation Enable)

The EXENB bit enables operation of the associated CS area. On MCU reset, operation is enabled (EXENB = 1) only for area 0. Operation in other areas is disabled (EXENB = 0). Attempts to access disabled areas have no effect.

When the CSC and SDRAMC are in use at the same time, EBCLK and SDCLK must operate at the same frequency.

BSIZE[1:0] bits (External Bus Width Select)

The BSIZE[1:0] bits specify the data bus width for the associated area.

When the address/data multiplexed I/O interface is set with the MPXEN bit, it is prohibited to set it in 32-bit bus space, and the operation is not guaranteed when it is set.

EMODE bit (Endian Mode)

The EMODE bit specifies the endianness for the associated area. See section 15.2.5.1. Endianness Constraint for details.

MPXEN bit (Address/Data Multiplexed I/O Interface Select)

The MPXEN bit specifies separate bus interface or address/data multiplexed I/O interface of each area.

15.3.6 CSnREC : CSn Recovery Cycle Register (n = 0 to 7)

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x80A + 0x10 × n

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

Bit	Symbol	Function	R/W
3:0	RRCV[3:0]	Read Recovery 0 0 0 0: No recovery cycle is inserted. 0 0 0 1: 1 recovery cycle is inserted. 0 0 1 0: 2 recovery cycles are inserted. 0 0 1 1: 3 recovery cycles are inserted. 0 1 0 0: 4 recovery cycles are inserted. 0 1 0 1: 5 recovery cycles are inserted. 0 1 1 0: 6 recovery cycles are inserted. 0 1 1 1: 7 recovery cycles are inserted. 1 0 0 0: 8 recovery cycles are inserted. 1 0 0 1: 9 recovery cycles are inserted. 1 0 1 0: 10 recovery cycles are inserted. 1 0 1 1: 11 recovery cycles are inserted. 1 1 0 0: 12 recovery cycles are inserted. 1 1 0 1: 13 recovery cycles are inserted. 1 1 1 0: 14 recovery cycles are inserted. 1 1 1 1: 15 recovery cycles are inserted.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
11:8	WRCV[3:0]	Write Recovery 0 0 0 0: No recovery cycle is inserted. 0 0 0 1: 1 recovery cycle is inserted. 0 0 1 0: 2 recovery cycles are inserted. 0 0 1 1: 3 recovery cycles are inserted. 0 1 0 0: 4 recovery cycles are inserted. 0 1 0 1: 5 recovery cycles are inserted. 0 1 1 0: 6 recovery cycles are inserted. 0 1 1 1: 7 recovery cycles are inserted. 1 0 0 0: 8 recovery cycles are inserted. 1 0 0 1: 9 recovery cycles are inserted. 1 0 1 0: 10 recovery cycles are inserted. 1 0 1 1: 11 recovery cycles are inserted. 1 1 0 0: 12 recovery cycles are inserted. 1 1 0 1: 13 recovery cycles are inserted. 1 1 1 0: 14 recovery cycles are inserted. 1 1 1 1: 15 recovery cycles are inserted.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

When the CSnCR.EXENB bit is set to 1, if a write access is made, the write is ignored.

Do not attempt to write to the CSnREC register while the external bus is being accessed.

When the preceding bus access is from a separate bus, CSnREC is valid when the recovery cycle insertion is enabled in the Separate Bus Recovery Cycle Insertion Enable bit (RCVENi (i = 0 to 7)) in CSRECEN. When the preceding bus access is an address/data multiplexed bus access, CSnREC is valid when the recovery cycle insertion is enabled with the Multiplexed Bus Recovery Cycle Insertion Enable bit (RCVENMj (j = 0 to 7)) in CSRECEN. For more information, see section 15.5.4. Insertion of Recovery Cycles.

RRCV[3:0] bits (Read Recovery)

The RRCV[3:0] bits specify the number of recovery cycles inserted after a read access on the external bus for CSn (n = 0 to 7). When recovery cycle insertion is enabled and a value other than 0000 is set, 1 to 15 recovery cycles are inserted when:

- After a read access to the external bus, a read access is made to the external bus in the same area.
- After a read access to the external bus, a read access is made to the external bus in a different area.
- After a read access to the external bus, a write access is made to the external bus in the same area.
- After a read access to the external bus, a write access is made to the external bus in a different area.

WRCV[3:0] bits (Write Recovery)

The WRCV[3:0] bits specify the number of recovery cycles inserted after a write access on the external bus for CSn (n = 0 to 7). When recovery cycle insertion is enabled and a value other than 0000 is set, 1 to 15 recovery cycles are inserted when:

- After a write access to the external bus, a read access is made to the external bus in the same area.
- After a write access to the external bus, a read access is made to the external bus in a different area.
- After a write access to the external bus, a write access is made to the external bus in the same area.
- After a write access to the external bus, a write access is made to the external bus in a different area.

15.3.7 CSRECEN : CS Recovery Cycle Insertion Enable Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x880

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RCVENM7	RCVENM6	RCVENM5	RCVENM4	RCVENM3	RCVENM2	RCVENM1	RCVENM0	RCVEN7	RCVEN6	RCVEN5	RCVEN4	RCVEN3	RCVEN2	RCVEN1	RCVEN0
Value after reset:	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	0

Bit	Symbol	Function	R/W
0	RCVEN0	Separate Bus Recovery Cycle Insertion Enable 0 0: Disabled 1: Enabled	R/W
1	RCVEN1	Separate Bus Recovery Cycle Insertion Enable 1 0: Disabled 1: Enabled	R/W
2	RCVEN2	Separate Bus Recovery Cycle Insertion Enable 2 0: Disabled 1: Enabled	R/W
3	RCVEN3	Separate Bus Recovery Cycle Insertion Enable 3 0: Disabled 1: Enabled	R/W
4	RCVEN4	Separate Bus Recovery Cycle Insertion Enable 4 0: Disabled 1: Enabled	R/W
5	RCVEN5	Separate Bus Recovery Cycle Insertion Enable 5 0: Disabled 1: Enabled	R/W
6	RCVEN6	Separate Bus Recovery Cycle Insertion Enable 6 0: Disabled 1: Enabled	R/W
7	RCVEN7	Separate Bus Recovery Cycle Insertion Enable 7 0: Disabled 1: Enabled	R/W
8	RCVENM0	Multiplexed Bus Recovery Cycle Insertion Enable 0 0: Disabled 1: Enabled	R/W
9	RCVENM1	Multiplexed Bus Recovery Cycle Insertion Enable 1 0: Disabled 1: Enabled	R/W
10	RCVENM2	Multiplexed Bus Recovery Cycle Insertion Enable 2 0: Disabled 1: Enabled	R/W
11	RCVENM3	Multiplexed Bus Recovery Cycle Insertion Enable 3 0: Disabled 1: Enabled	R/W
12	RCVENM4	Multiplexed Bus Recovery Cycle Insertion Enable 4 0: Disabled 1: Enabled	R/W

Bit	Symbol	Function	R/W
13	RCVENM5	Multiplexed Bus Recovery Cycle Insertion Enable 5 0: Disabled 1: Enabled	R/W
14	RCVENM6	Multiplexed Bus Recovery Cycle Insertion Enable 6 0: Disabled 1: Enabled	R/W
15	RCVENM7	Multiplexed Bus Recovery Cycle Insertion Enable 7 0: Disabled 1: Enabled	R/W

Note: S-TYPE-3, P-TYPE-3

Writes are ignored if the CSnCR.EXENB bit is set to 1. Write CSRECEN only when all CSnCR.EXENB bits are “0”.

Do not attempt to write the CSRECEN register while the external bus is being accessed. For more information on insertion recovery cycles, see section 15.5.4. Insertion of Recovery Cycles.

RCVENi Bit (Separate Bus Recovery Cycle Insertion Enable i) (i = 0 to 7)

This bit enables the insertion of read or write recovery cycles when, after a read or write access on the external bus, a read or write access is made on the external bus to the same or different area.

RCVENMj Bit (Multiplexed Bus Recovery Cycle Insertion Enable j) (j = 0 to 7)

This bit enables the insertion of read or write recovery cycles when, after a read or write access on the external bus, a read or write access is made on the external bus to the same or different area.

Table 15.6 Access type associations with the RCVENn bits

Access type	External Address Space	Insertion of recovery cycles	Associated bits (Separate/ Multiplexed)
Read access after read access	Same area	Recovery cycles specified in the RRCV[3:0] bits are inserted for the priority access area	RCVEN0/RCVENM0
	Different area	Recovery cycles specified in the RRCV[3:0] bits are inserted for the priority access area	RCVEN1/RCVENM1
Write access after read access	Same area	Recovery cycles specified in the RRCV[3:0] bits are inserted for the priority access area	RCVEN2/RCVENM2
	Different area	Recovery cycles specified in the RRCV[3:0] bits are inserted for the priority access area	RCVEN3/RCVENM3
Read access after write access	Same area	Recovery cycles specified in the WRCV[3:0] bits are inserted for the priority access area	RCVEN4/RCVENM4
	Different area	Recovery cycles specified in the WRCV[3:0] bits are inserted for the priority access area	RCVEN5/RCVENM5
Write access after write access	Same area	Recovery cycles specified in the WRCV[3:0] bits are inserted for the priority access area	RCVEN6/RCVENM6
	Different area	Recovery cycles specified in the WRCV[3:0] bits are inserted for the priority access area	RCVEN7/RCVENM7

15.3.8 CSnMOD : CSn Mode Register (n = 0 to 7)

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x002 + 0x10 × n

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

Bit	Symbol	Function	R/W
0	WRMOD	Write Access Mode Select 0: Byte strobe mode 1: Single-write strobe mode	R/W
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	EWENB	External Wait Enable 0: Disabled 1: Enabled	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	PRENB	Page Read Access Enable 0: Disabled 1: Enabled	R/W
9	PWENB	Page Write Access Enable 0: Disabled 1: Enabled	R/W
14:10	—	These bits are read as 0. The write value should be 0.	R/W
15	PRMOD	Page Read Access Mode Select 0: Normal access compatible mode 1: External data read continuous assertion mode	R/W

Note: S-TYPE-3, P-TYPE-3

When the CSnCR.EXENB bit is set to 1, if a write access is made, the write is ignored.

Do not write to the CSnMOD register while access to the CSn area is in progress.

WRMOD bit (Write Access Mode Select)

The WRMOD bit selects the write access operating mode. Writing 0 selects byte strobe mode, in which data writes are controlled by the WRn signals (n = 0 to 3) associated with the respective byte positions. Writing 1 selects single-write strobe mode, in which data writes are controlled by the BCn (n = 0 to 3) and WR signals associated with the respective byte positions.

Note: Setting the external bus width to 8 bits is prohibited in single-write strobe mode.

Table 15.7 Control signals for write access mode

Mode	Pin name							
Write access mode	**WR3**	**WR2**	**WR1**	**WR0/WR**	**BC3**	**BC2**	**BC1**	**BC0**
Byte strobe mode	✓	✓	✓	✓ (WR0)	—	—	—	—
Single write strobe mode	—	—	—	✓ (WR)	✓	✓	✓	✓

✓ : Enabled, — : Disabled

EWENB bit (External Wait Enable)

The EWENB bit enables external waits. Writing 0 disables the WAIT signal. Writing 1 selects external wait and allows the WAIT signal to control the number of waits per cycle. In this state, wait cycles are inserted when the WAIT signal is low.

PRENB bit (Page Read Access Enable)

The PRENB bit enables page read accesses.

Note: When the address/data multiplexed I/O interface is selected with the CSnCR.MPXEN bit, PRENB should not be set to enable page read accesses. Page read accesses are not supported in the address/data multiplexed I/O interface.

PWENB bit (Page Write Access Enable)

The PWENB bit enables page write accesses.

Note: When the address/data multiplexed I/O interface is selected with the CSnCR.MPXEN bit, PWENB should not be set to enable page write accesses. Page write accesses are not supported in the address/data multiplexed I/O interface.

PRMOD bit (Page Read Access Mode Select)

The PRMOD bit selects the operating mode for page read accesses. Writing 0 selects normal access compatible mode, in which the RD signal is negated and an RD assert wait is inserted each time a unit of data is read. When there is no RD assert wait, the RD signal is negated only in the final transfer of the external bus access.

Writing 1 selects external data read continuous assertion mode, in which an RD assert wait is inserted and the RD signal is continuously asserted during the wait.

15.3.9 CSnWCR1 : CSn Wait Control Register 1 (n = 0 to 7)

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x004 + 0x10 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	CSRWAIT[4:0]					—	—	—	CSWWAIT[4:0]				
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

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

Bit	Symbol	Function	R/W
2:0	CSPWWAIT[2:0]	Page Write Cycle Wait Select*1 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
10:8	CSPRWAIT[2:0]	Page Read Cycle Wait Select*2 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
20:16	CSWWAIT[4:0]	Normal Write Cycle Wait Select 0 0 0 0 0: No wait is inserted. 0 0 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 0 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 0 0 1 1: Wait with a length of 3 clock cycles is inserted. ⋮ 1 1 1 0 1: Wait with a length of 29 clock cycles is inserted. 1 1 1 1 0: Wait with a length of 30 clock cycles is inserted. 1 1 1 1 1: Wait with a length of 31 clock cycles is inserted.	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
28:24	CSRWAIT[4:0]	Normal Read Cycle Wait Select 0 0 0 0 0: No wait is inserted. 0 0 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 0 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 0 0 1 1: Wait with a length of 3 clock cycles is inserted. ⋮ 1 1 1 0 1: Wait with a length of 29 clock cycles is inserted. 1 1 1 1 0: Wait with a length of 30 clock cycles is inserted. 1 1 1 1 1: Wait with a length of 31 clock cycles is inserted.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The CSPWWAIT[2:0] value is only valid when the CSnMOD.PWENB bit is set to 1.
Note 2. The CSPRWAIT[2:0] value is only valid when the CSnMOD.PRENB bit is set to 1.

When the CSnCR.EXENB bit is set to 1, if a write access is made, the write is ignored.

Do not attempt to write to the CSnWCR1 register while the external bus is being accessed. Set each of these bits within a range of the restrictions described in (1) Constraints on using separate bus interface or (2) Constraints on using address/data multiplexed bus interface, according to the bus interface used.

CSPWWAIT[2:0] bits (Page Write Cycle Wait Select)

The CSPWWAIT[2:0] bits specify the number of wait cycles to be inserted into the second and subsequent accesses during a page write cycle. The setting is enabled when the CSnMOD.PWENB bit is set to 1.

Note: The settings must satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value, and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.

CSPRWAIT[2:0] bits (Page Read Cycle Wait Select)

The CSPRWAIT[2:0] bits specify the number of wait cycles to be inserted into the second and subsequent accesses during a page read cycle. The setting is enabled when the CSnMOD.PRENB bit is set to 1.

Note: The settings must satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSPRWAIT[2:0] value.

CSWWAIT[4:0] bits (Normal Write Cycle Wait Select)

The CSWWAIT[4:0] bits specify the number of wait cycles to be inserted into the first access during a normal write cycle or page write cycle.

Note: The settings must satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value, and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

CSRWAIT[4:0] bits (Normal Read Cycle Wait Select)

The CSRWAIT[4:0] bits specify the number of wait cycles to be inserted into the first access during a normal read cycle or page read cycle.

Note: The settings must satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.

15.3.10 CSnWCR2 : CSn Wait Control Register 2 (n = 0 to 7)

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x008 + 0x10 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	CSON[2:0]			—	WDON[2:0]			—	WRON[2:0]			—	RDON[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	CSROFF[2:0]	Read Access CS Extension Cycle Select 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	CSWOFF[2:0]	Write Access CS Extension Cycle Select 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
10:8	WDOFF[2:0]	Write Data Output Extension Cycle Select 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
13:12	AWAIT[1:0]	Address Cycle Wait Select 0 0: No wait is inserted. 0 1: Wait with a length of 1 clock cycle is inserted. 1 0: Wait with a length of 2 clock cycles is inserted. 1 1: Wait with a length of 3 clock cycles is inserted.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
18:16	RDON[2:0]	RD Assert Wait Select 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
22:20	WRON[2:0]	WR Assert Wait Select 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
26:24	WDON[2:0]	Write Data Output Wait Select 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
30:28	CSON[2:0]	CS Assert Wait Select 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles is inserted. 0 1 1: Wait with a length of 3 clock cycles is inserted. 1 0 0: Wait with a length of 4 clock cycles is inserted. 1 0 1: Wait with a length of 5 clock cycles is inserted. 1 1 0: Wait with a length of 6 clock cycles is inserted. 1 1 1: Wait with a length of 7 clock cycles is inserted.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

When the CSnCR.EXENB bit is set to 1, if a write access is made, the write is ignored.

Do not attempt to write to the CSnWCR2 register while the external bus is being accessed. Set each of these bits within a range of the restrictions described in (1) Constraints on using separate bus interface or (2) Constraints on using address/data multiplexed bus interface, according to the bus interface used.

CSROFF[2:0] bits (Read Access CS Extension Cycle Select)

The CSROFF[2:0] bits specify the number of wait cycles to be inserted during the period from the end of a wait cycle (RD signal negated) until the CSn signal (n = 0 to 7) is negated in read access mode.

CSWOFF[2:0] bits (Write Access CS Extension Cycle Select)

The CSWOFF[2:0] bits specify the number of wait cycles to be inserted during the period from the end of a wait cycle (WRn signal (n = 0 to 3) negated) until the CSn signal (n = 0 to 7) is negated in write access mode.

Note: The settings must satisfy CSnWCR2.WDOFF[2:0] value ≤ CSnWCR2.CSWOFF[2:0] value.

WDOFF[2:0] bits (Write Data Output Extension Cycle Select)

The WDOFF[2:0] bits specify the number of wait cycles to be inserted during the period from the end of a wait cycle (WRn signal (n = 0 to 3) negated) until the write-data output is complete in write access mode.

Note: The settings must satisfy CSnWCR2.WDOFF[2:0] value ≤ CSnWCR2.CSWOFF[2:0] value.

AWAIT[1:0] bits (Address Cycle Wait Select)

AWAIT[1:0] bits specify the number of wait cycles to be inserted into an address output cycle with the address/data multiplexed I/O interface.

Note: CSnWCR2.CSON[2:0] value ≤ CSnWCR2.AWAIT[1:0] value.

For read access, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.

For write access, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value and CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

RDON[2:0] bits (RD Assert Wait Select)

RDON[2:0] bits specify the number of wait cycles to be inserted before the RD signal is asserted.

Note: For normal read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.

For page read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSPRWAIT[2:0] value.

When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.

WRON[2:0] bits (WR Assert Wait Select)

The WRON[2:0] bits specify the number of wait cycles to be inserted before the WRn signal (n = 0 to 3) is asserted.

Note: For normal write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value, and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

For page write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value, and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.

When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

WDON[2:0] bits (Write Data Output Wait Select)

The WDON[2:0] bits specify the number of wait cycles to be inserted before the write data is output.

Note: For normal write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

For page write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.

When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

CSON[2:0] bits (CS Assert Wait Select)

The CSON[2:0] bits specify the number of wait cycles to be inserted before the CSn signal (n = 0 to 7) is asserted.

Note: For normal read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.

For page read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSPRWAIT[2:0] value.

For normal write access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

For page write access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.

When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.AWAIT[1:0] value.

15.3.11 SDCCR : SDC Control Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0xC00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	BSIZE[1:0]		—	—	—	EXENB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EXENB	Operation Enable 0: Disable 1: Enable	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
5:4	BSIZE[1:0]	SDRAM Bus Width Select 0 0: 16-bit bus space 0 1: 32-bit bus space 1 0: 8-bit bus space Others: Setting prohibited	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The BSIZE bits is not rewritable when the EXENB bit is set to 1. It is possible to rewrite at the same time as the EXENB bit.

EXENB bit (Operation Enable)

The EXENB bit enables the operation of the SDRAM address space. On reset, operation is disabled (EXENB = 0). Attempts to access disabled areas have no effect.

When CSC and SDRAMC are in use at the same time, EBCLK and SDCLK must operate at the same frequency.

BSIZE[1:0] bits (SDRAM Bus Width Select)

The BSIZE[1:0] bits specify the data bus width for the associated area.

15.3.12 SDCMOD : SDC Mode Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0xC01

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

Bit	Symbol	Function	R/W
0	EMODE	Endian Mode 0: Little endian 1: Big endian	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

When the SDCCR.EXENB bit is set to 1, if a write access is made, the write is ignored.

Writing to this register is possible only once after release from reset. The second and subsequent writes to this register are prohibited. SDRAM access operation is not guaranteed if more than one write access is attempted.

EMODE bit (Endian Mode)

The EMODE bit specifies the endianness for the SDRAM address space. See section 15.2.5.1. Endianness Constraint for details.

15.3.13 SDAMOD : SDRAM Access Mode Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0xC02

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

Bit	Symbol	Function	R/W
0	BE	Continuous Access Enable 0: Continuous access is disabled 1: Continuous access is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

When the SDCCR.EXENB bit is set to 1, if a write access is made, the write is ignored.

Set SDAMOD while the conditions listed in Table 15.31 are satisfied. The operation is not guaranteed if this register is set while these conditions are not satisfied.

BE bit (Continuous Access Enable)

This bit enables or disables continuous access to the SDRAM access space.

15.3.14 SDSELF : SDRAM Self-Refresh Control Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0xC10

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

Bit	Symbol	Function	R/W
0	SFEN	SDRAM Self-Refresh Enable 0: Disable 1: Enable	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Set the SDSELF register only when the conditions in Table 15.31 are satisfied. Otherwise, operation of SDRAM I/F function is not guaranteed.

SFEN bit (SDRAM Self-Refresh Enable)

The SFEN bit controls the self-refresh operation. Setting this bit to 1 initiates an auto-refresh cycle, after which self-refresh begins. Clearing this bit to 0 ends the self-refresh, and auto-refresh resumes. When the bit is set to 1, the write value takes effect when the self-refresh operation starts. When it is cleared to 0, if the self-refresh operation has already finished, the write value takes effect before the auto-refresh starts. If the self-refresh operation is in progress, the write value takes effect after the self-refresh is finished and before the auto-refresh is started.

15.3.15 SDRFCR : SDRAM Refresh Control Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0xC14

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

Bit	Symbol	Function	R/W
11:0	RFC[11:0]	Auto-Refresh Request Interval Setting 0 0 0 0 0 0 0 0 0 0 0 0: Setting prohibited 0 0 0 0 0 0 0 0 0 0 0 1: 2 cycles 0 0 0 0 0 0 0 0 0 0 1 0: 3 cycles ⋮ 1 1 1 1 1 1 1 1 1 1 1 1: 4096 cycles	R/W
15:12	REFW[3:0]	Auto-Refresh Cycle/Self-Refresh Clearing Cycle Count Setting 0 0 0 0: 1 cycle 0 0 0 1: 2 cycles 0 0 1 0: 3 cycles 0 0 1 1: 4 cycles 0 1 0 0: 5 cycles 0 1 0 1: 6 cycles 0 1 1 0: 7 cycles 0 1 1 1: 8 cycles 1 0 0 0: 9 cycles 1 0 0 1: 10 cycles 1 0 1 0: 11 cycles 1 0 1 1: 12 cycles 1 1 0 0: 13 cycles 1 1 0 1: 14 cycles 1 1 1 0: 15 cycles 1 1 1 1: 16 cycles	R/W

Note: S-TYPE-3, P-TYPE-3

RFC[11:0] bits (Auto-Refresh Request Interval Setting)

The RFC[11:0] bits specify the auto-refresh request interval. They can be written to at any time, regardless of the state of the Auto-Refresh Operation Enable bit (RFEN) in SDRFEN. If auto-refresh is enabled, the write value takes effect after the end of the auto-refresh cycles. The refresh counter uses SDCLK.

REFW[3:0] bits (Auto-Refresh Cycle/Self-Refresh Clearing Cycle Count Setting)

The REFW[3:0] bits specify the number of auto-refresh cycles and the number of self-refresh clearing cycles. They can be written to at any time, regardless of the state of the Auto-Refresh Operation Enable bit (RFEN) in SDRFEN. If an auto-refresh cycle is in progress, the value written to the bits while auto-refresh is enabled takes effect after the cycle completes.

Note: Auto-refresh requests are not accepted while the SDRAM is being accessed. This means they must sometimes wait until the access completes for the auto-refresh interval to be extended. Set the RFC[11:0] bits to an auto-refresh request interval value that meets the specifications of the SDRAM being used. Additionally, make sure to set the auto-refresh request interval to a duration longer than the auto-refresh cycle. The auto-refresh interval cannot be automatically adjusted when the frequency is changed during operation. In this case, perform a self-refresh operation and set the auto-refresh interval to an appropriate value for the frequency again.

15.3.15.1 Auto-refresh Request Interval and RFC Set Value

The SDRAMC (SDRAM area controller) includes a 12-bit refresh counter that generates auto-refresh requests at fixed intervals. Use the following equation to calculate the set value for the RFC[11:0] bits from the auto-refresh request interval:

RFC = (Auto-refresh request interval / SDCLK cycle) - 1

Note: Auto-refresh requests are not accepted while the SDRAM is being accessed. They must wait until the access completes. However, the counter value is updated regardless of whether or not the request was accepted. If two or more auto-refresh requests are generated while the SDRAM is being accessed, the second and subsequent requests are ignored.

15.3.16 SDRFEN : SDRAM Auto-Refresh Control Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0xC16

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

Bit	Symbol	Function	R/W
0	RFEN	Auto-Refresh Operation Enable 0: Disable 1: Enable	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

RFEN bit (Auto-Refresh Operation Enable)

The RFEN bit enables auto-refresh operation. When auto-refresh is required, set the RFEN bit to 1 before SDRAM access.

If the RFEN bit is set to 0 when auto-refresh is enabled, the RFEN bit becomes 0 after the auto-refresh cycle ends, and the auto-refresh operation stops. The interval at which refresh requests are generated is determined by the value in the Auto-Refresh Request Interval Setting bits (RFC[11:0]) in the SDRAM Refresh Control Register (SDRFCR).

15.3.17 SDICR : SDRAM Initialization Sequence Control Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0xC20

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

Bit	Symbol	Function	R/W
0	INIRQ	Initialization Sequence Start 0: Invalid 1: Start initialization sequence	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Writing to this register is possible only once after release from reset. Operation is not guaranteed if more than one write access is attempted.

INIRQ bit (Initialization Sequence Start)

Setting the INIRQ bit to 1 starts the SDRAM initialization sequence and automatically sets the Initialization Status bit (INIST) in the SDRAM Status Register (SDSR) to 1. The INIST bit clears automatically after the initialization sequence ends. The value written to the INIRQ bit is not retained.

Initialize the SDRAM according to the procedure in section 15.6.11.1. SDRAMC Access Procedure.

Note: Set the INIRQ bit to start the SDRAM initialization sequence only when the conditions in Table 15.31 are satisfied. Otherwise, operation is not guaranteed.

15.3.18 SDIR : SDRAM Initialization Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0xC24

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

Bit	Symbol	Function	R/W
3:0	ARFI[3:0]	Initialization Auto-Refresh Interval 0 0 0 0: 3 cycles 0 0 0 1: 4 cycles 0 0 1 0: 5 cycles 0 0 1 1: 6 cycles 0 1 0 0: 7 cycles 0 1 0 1: 8 cycles 0 1 1 0: 9 cycles 0 1 1 1: 10 cycles 1 0 0 0: 11 cycles 1 0 0 1: 12 cycles 1 0 1 0: 13 cycles 1 0 1 1: 14 cycles 1 1 0 0: 15 cycles 1 1 0 1: 16 cycles 1 1 1 0: 17 cycles 1 1 1 1: 18 cycles	R/W
7:4	ARFC[3:0]	Initialization Auto-Refresh Count 0 0 0 0: Setting prohibited 0 0 0 1: 1 time 0 0 1 0: 2 times 0 0 1 1: 3 times 0 1 0 0: 4 times 0 1 0 1: 5 times 0 1 1 0: 6 times 0 1 1 1: 7 times 1 0 0 0: 8 times 1 0 0 1: 9 times 1 0 1 0: 10 times 1 0 1 1: 11 times 1 1 0 0: 12 times 1 1 0 1: 13 times 1 1 1 0: 14 times 1 1 1 1: 15 times	R/W

Bit	Symbol	Function	R/W
10:8	PRC[2:0]	Initialization Precharge Cycle Count 0 0 0: 3 cycles 0 0 1: 4 cycles 0 1 0: 5 cycles 0 1 1: 6 cycles 1 0 0: 7 cycles 1 0 1: 8 cycles 1 1 0: 9 cycles 1 1 1: 10 cycles	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Writing to this register is possible only once after release from reset. Operation is not guaranteed if more than one write access is attempted.

ARFI[3:0] bits (Initialization Auto-Refresh Interval)

The ARFI[3:0] bits specify the interval at which the auto-refresh commands are issued in the SDRAM initialization sequence.

ARFC[3:0] bits (Initialization Auto-Refresh Count)

The ARFC[3:0] bits specify the number of times auto-refresh is to be performed in the SDRAM initialization sequence.

PRC[2:0] bits (Initialization Precharge Cycle Count)

The PRC[2:0] bits specify the number of precharged cycles in the SDRAM initialization sequence.

Note: Implement settings that satisfy the specifications of the connected SDRAM before starting the initialization sequence.

15.3.19 SDADR : SDRAM Address Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0xC40

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

Bit	Symbol	Function	R/W
1:0	MXC[1:0]	Address Multiplex Select 0 0: 8-bit shift 0 1: 9-bit shift 1 0: 10-bit shift 1 1: 11-bit shift	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

When the SDCCR.EXENB bit is set to 1, if a write access is made, the write is ignored.

Set SDADR only when the conditions in Table 15.31 are satisfied. Otherwise, operation is not guaranteed.

MXC[1:0] bits (Address Multiplex Select)

The MXC[1:0] bits select the size of the shift towards the lower half of the row address in row address/column address multiplexing. For details, see Table 15.36.

15.3.20 SDTR : SDRAM Timing Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0xC44

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

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

Bit	Symbol	Function	R/W
2:0	CL[2:0]	SDRAMC Column Latency 0 0 0: Setting prohibited 0 0 1: 1 cycle 0 1 0: 2 cycles 0 1 1: 3 cycles Others: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	WR	Write Recovery Interval 0: 1 cycle 1: 2 cycles	R/W
11:9	RP[2:0]	Row Precharge Interval 0 0 0: 1 cycle 0 0 1: 2 cycles 0 1 0: 3 cycles 0 1 1: 4 cycles 1 0 0: 5 cycles 1 0 1: 6 cycles 1 1 0: 7 cycles 1 1 1: 8 cycles	R/W
13:12	RCD[1:0]	Row Column Latency 0 0: 1 cycle 0 1: 2 cycles 1 0: 3 cycles 1 1: 4 cycles	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
18:16	RAI[2:0]	Row Active Interval 0 0 0: 1 cycle 0 0 1: 2 cycles 0 1 0: 3 cycles 0 1 1: 4 cycles 1 0 0: 5 cycles 1 0 1: 6 cycles 1 1 0: 7 cycles 1 1 1: Setting prohibited	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

When the SDCCR.EXENB bit is set to 1, if a write access is made, the write is ignored.

The SDTR register specifies the timing for read and write accesses to the SDRAM. For more information, see section 15.6.11.3. Timing Register Settings and Access Timing.

Set the SDTR register only when the conditions in Table 15.31 are satisfied. Otherwise, operation is not guaranteed.

Writing to this register is possible only once after release from reset. Operation is not guaranteed if more than one write access is attempted.

CL[2:0] bits (SDRAMC Column Latency)

The CL[2:0] bits specify the column latency of the SDRAM controller. This setting only affects the latency setting on the SDRAM controller side. To specify the column latency for externally connected SDRAM, use the SDRAM mode register (SDMOD).

WR bit (Write Recovery Interval)

The WR bit specifies the interval that must elapse between the SDRAM write command (WRIT) and deactivation (PRE).

RP[2:0] bits (Row Precharge Interval)

The RP[2:0] bits specify the minimum number of cycles that must elapse between the SDRAM deactivation command (PRE) and the next valid command.

RCD[1:0] bits (Row Column Latency)

The RCD[1:0] bits set the SDRAM row-column latency.

RAI[2:0] bits (Row Active Interval)

The RAI[2:0] bits specify the minimum interval that must elapse between the SDRAM row activation command (ACTV) and deactivation (PRE). The value specified in these bits must be less than or equal to the sum of the row column latency (RCD[1:0]) and column latency (CL[2:0]) settings.

15.3.21 SDMOD : SDRAM Mode Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0xC48

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

Bit	Symbol	Function	R/W
14:0	MR[14:0]	Mode Register Setting Writing to these bits triggers a mode register set command	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The SDMOD register specifies the value to be written to the SDRAM mode register. Writing to SDMOD causes a mode register set command to be issued automatically to the SDRAM. Set SDMOD only when the conditions in Table 15.31 are satisfied. Otherwise, operation is not guaranteed.

Writing to this register is possible only once after release from reset. Operation is not guaranteed if more than one write access is attempted.

MR[14:0] bits (Mode Register Setting)

Writing to the MR[14:0] bits causes a mode register set command to be issued to the SDRAM, and the setting in the MR[14:0] bits is output to the lower bits of the address. For more information, see section 15.6.9. Setting the Mode Register.

Write the mode register according to the procedure in section 15.6.11.1. SDRAMC Access Procedure.

Note: 1. Set a burst length of 1 for the SDRAM. Operation is not guaranteed with any other burst length setting.
2. The SDRAM column latency must match the setting in the SDRAMC Column Latency setting (CL[2:0]) in the SDRAM Timing Register (SDTR). Operation is not guaranteed if the latency settings do not agree.
3. Make sure the SRFST, INIST, and MRSST status bits in the SDRAM Status Register (SDSR) are all 0.

15.3.22 SDSR : SDRAM Status Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0xC50

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

Bit	Symbol	Function	R/W
0	MRSST	Mode Register Setting Status 0: Mode register setting not in progress 1: Mode register setting in progress	R
2:1	—	These bits are read as 0.	R
3	INIST	Initialization Status 0: Initialization sequence not in progress 1: Initialization sequence in progress	R
4	SRFST	Self-Refresh Transition/Recovery Status 0: Transition/recovery not in progress 1: Transition/recovery in progress	R
7:5	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

MRSST bit (Mode Register Setting Status)

When set to 1, the MRSST bit indicates that SDRAM mode register setting is in progress.

INIST bit (Initialization Status)

When set to 1, the INIST bit indicates that the SDRAM initialization sequence is in progress.

SRFST bit (Self-Refresh Transition/Recovery Status)

When set to 1, the SRFST bit indicates that a transition to or recovery from a self-refresh operation is in progress for the SDRAM. The in progress interval begins when the bits in Table 15.8 are written to and lasts until the associated commands are issued.

Note: Execution of a self-refresh, initialization sequence, or mode register setting can only be performed when all the status bits are 0. Do not rewrite the registers and bits in Table 15.8 when any of the SRFST, INIST, or MRSST status bits is set to 1.

Table 15.8 Registers and bits requiring status bit checking

Function	Register	Bits
Self-refresh	SDSELF	SFEN
Initialization sequence	SDICR	INIRQ
Mode register setting	SDMOD	MR[14:0]

15.3.23 BUSSABT0<slave> : Bus Slave Arbitration Control Register 0

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x1208 (BUSSABT0MRE0BI), 0x1220 (BUSSABT0S0BI), 0x1228 (BUSSABT0S1BI), 0x1230 (BUSSABT0S2BI), 0x1238 (BUSSABT0S3BI), 0x1250 (BUSSABT0ECBI), 0x1258 (BUSSABT0OSPI0BI), 0x1260 (BUSSABT0OSPI1BI), 0x1268 (BUSSABT0CPU0SAHBI), 0x1270 (BUSSABT0CPU1TCMBI), 0x1278 (BUSSABT0PBBI), 0x1280 (BUSSABT0PABI), 0x1288 (BUSSABT0PIBI), 0x1290 (BUSSABT0PSBI)

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

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

Bit	Symbol	Function	R/W
0	ARBS	Arbitration Select for <slave> 0: Fixed priority 1: Round-robin	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

This register specifies the arbitration method for <slave> buses.

(<slave> = MRE0BI, S0BI, S1BI, S2BI, S3BI, ECBI, OSPI0BI, OSPI1BI, CPU0SAHBI, CPU1TCMBI, PBBI, PABI, PIBI, PSBI)

ARBS bit (Arbitration Select for <slave>)

The ARBS bit specifies the priority between bus masters connected to <slave>. For fixed priority, see Table 15.9. For round-robin, see Table 15.10.

Table 15.9 Bus priorities with fixed-priority arbitration (ARBS = 0)

Bus Slave Arbitration Control Register	Slave interface	Priority order*1
BUSSABT0MRE0BI	MRE0BI	GRAPH > EDMAC > DMAC/DTC0 > DMAC/DTC1 > CPU0MAXI > CPU1CAHB
BUSSABT0SnBI (n = 0 to 3) BUSSABT0ECBI BUSSABT0OSPI0BI	S0BI, S1BI, S2BI, S3BI ECBI OSPI0BI	GRAPH > EDMAC > DMAC/DTC0 > DMAC/DTC1 > CPU0MAXI > CPU1SAHB
BUSSABT0OSPI1BI	OSPI1BI	GRAPH > EDMAC > DMAC/DTC0 > DMAC/DTC1 > CPU0MAXI > CPU1SAHB > CPU1CAHB
BUSSABT0CPU0SAHBI	CPU0SAHBI	DMAC/DTC0 (R > W) > DMAC/DTC1 (R >W) > CPU1SAHB
BUSSABT0CPU1TCMBI	CPU1TCMBI	(DMAC/DTC0 (R > W) >DMAC/DTC1 (R > W)) > CPU0MAXI
BUSSABT0PBBI BUSSABT0PABI BUSSABT0PIBI BUSSABT0PSBI	PBBI PABI PIBI PSBI	DMAC/DTC0 (R > W) > DMAC/DTC1 (R > W) > CPU0PAHB > CPU1SAHB

Note 1. “>” represents left side has high priority.

Table 15.10 Bus priorities with round-robin arbitration (ARBS = 1) (1 of 2)

Bus Slave Arbitration Control Register	Slave interface	Priority order*1
BUSSABT0MRE0BI	MRE0BI	GRAPH ↔ EDMAC ↔ DMAC/DTC0 ↔ DMAC/DTC1 ↔ CPU0MAXI ↔ CPU1CAHB

Table 15.10 Bus priorities with round-robin arbitration (ARBS = 1) (2 of 2)

Bus Slave Arbitration Control Register	Slave interface	Priority order[*1]
BUSSABT0SnBI (n = 0 to 3) BUSSABT0ECBI BUSSABT0OSPI0BI	S0BI, S1BI, S2BI, S3BI ECBI OSPI0BI	GRAPH ↔ EDMAC ↔ DMAC/DTC0 ↔ DMAC/DTC1 ↔ CPU0MAXI ↔ CPU1SAHB
BUSSABT0OSPI1BI	OSPI1BI	GRAPH ↔ EDMAC ↔ DMAC/DTC0 ↔ DMAC/DTC1 ↔ CPU0MAXI ↔ CPU1SAHB ↔ CPU1CAHB
BUSSABT0CPU0SAHBI	CPU0SAHBI	DMAC/DTC0 ↔ DMAC/DTC1 ↔ CPU1SAHB
BUSSABT0CPU1TCMBI	CPU1TCMBI	(DMAC/DTC0 ↔ DMAC/DTC1) ↔ CPU0MAXI
BUSSABT0PBBI BUSSABT0PABI BUSSABT0PIBI BUSSABT0PSBI	PBBI PABI PIBI PSBI	DMAC/DTC0 ↔ DMAC/DTC1 ↔ CPU0PAHB ↔ CPU1SAHB

Note 1. "↔" represents round-robin.

15.3.24 BUSSABT1MRC0BI : Bus Slave Arbitration Control Register 1

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x1200

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

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

Bit	Symbol	Function	R/W
1:0	ARBS[1:0]	Arbitration Select 0 0: Fixed priority 0 1: Setting prohibited 1 0: Combination of round-robin and fixed priority 1 1: Round-robin	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

This register specifies the arbitration method for MRC0BI.

ARBS[1:0] bits (Arbitration Select)

The ARBS bits specify the priority between bus masters connected to Code MRAM. For the priority of each setting, see Table 15.11, Table 15.12 and Table 15.13.

Table 15.11 Bus priorities with fixed-priority arbitration (ARBS[1:0] = 00b)

Bus Slave Arbitration Control Register	Slave interface	Priority order[*1]
BUSSABT1MRC0BI	MRC0BI	EDMAC > DMAC/DTC0 > DMAC/DTC1 > CPU0MAXI > CPU1CAHB

Note 1. ">" represents left side has high priority.

Table 15.12 Bus priorities with combination of round-robin and fixed priority arbitration (ARBS[1:0] = 10b)

Bus Slave Arbitration Control Register	Slave interface	Priority order[*1]
BUSSABT1MRC0BI	MRC0BI	(EDMAC ↔ DMAC/DTC0 ↔ DMAC/DTC1) > CPU0MAXI > CPU1CAHB

Note 1. "↔" represents round-robin. ">" represents left side has high priority.

Table 15.13 Bus priorities with round-robin arbitration (ARBS[1:0] = 11b)

Bus Slave Arbitration Control Register	Slave interface	Priority order[*1]
BUSSABT1MRC0BI	MRC0BI	(EDMAC ↔ DMAC/DTC0 ↔ DMAC/DTC1) ↔ CPU0MAXI ↔ CPU1CAHB

Note 1. "↔" represents round-robin.

15.3.25 BUSERRSTAT<Master Name> : BUS Error Status Register

(Master Name: CPU0, CPU1, DMAC0, DMAC1, EDMAC, CEU)

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x1A00 (BUSERRSTATCPU0), 0x1A10 (BUSERRSTATCPU1), 0x1A20 (BUSERRSTATDMAC0), 0x1A30 (BUSERRSTATDMAC1), 0x1A50 (BUSERRSTATEDMAC), 0x1AA0 (BUSERRSTATCEU)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	MSERRSTAT	ILERRSTAT	MMERRSTAT	—	—	SLERRSTAT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SLERRSTAT	Slave Bus Error Status 0: No error occurred 1: Error occurred	R
2:1	—	These bits are read as 0.	R
3	MMERRSTAT	Master MPU Error Status 0: No error occurred 1: Error occurred	R
4	ILERRSTAT	Illegal Address Access Error Status 0: No error occurred 1: Error occurred	R
5	MSERRSTAT	Master Security Attribution Unit Error Status 0: No error occurred 1: Error occurred	R
7:6	—	These bits are read as 0.	R

Note: S-TYPE-5, P-TYPE-2
Note: This register is initialized after returning from Software Standby mode while the debugger is connected.

This register is cleared by reset other than Bus error reset, Common memory reset, and CPU local memory reset.

When the bus error occurred to plural at the same time, the STAT bit becomes effective only 1 by the following order of priority. The left side is high priority.

Master Security attribution unit Error > Slave bus Error, Slave TrustZone filter Error > Master-MPU Error> Illegal Address Access Error

After one of MSERRSTAT, ILERRSTAT, MMERRSTAT, and SLERRSTAT are set once, MSERRSTAT, ILERRSTAT, MMERRSTAT, and SLERRSTAT are not renewed.

SLERRSTAT bit (Slave Bus Error Status)

When slave error or TrustZone filter error occurs by bus, BUSERRSTAT.SLERRSTAT<Master Name>.SLERRSTAT is set to 1.

There is clearing conditions.

- Set BUSERRCLR<Master Name>.SLERRCLR to 1.

Slave error is an error that occurs on a slave such as Time-out.

The SLERRSTAT bit is not set when debugger violate.

For detail of slave error that occurs by bus, see section 15.7. Bus Error Monitoring Section.

MMERRSTAT bit (Master MPU Error Status)

When master mpu error occurs by bus, BUSERRSTAT<Master Name>.MMERRSTAT is set to 1.

There is clearing conditions.

- Set BUSERRCLR<Master Name>.MMERRCLR to 1.

For detail of master mpu error that occurs by bus, see section 16, Memory Protection Unit (MPU).

ILERRSTAT bit (Illegal Address Access Error Status)

When illegal address access error occurs by bus and write access to MRE0BI occurs, BUSERRSTAT<Master Name>.ILERRSTAT is set to 1.

There is clearing conditions.

- Set BUSERRCLR<Master Name>.ILLERRCLR to 1.

The ILERRSTAT bit is not set when debugger violate.

For detail of illegal address access error that occurs by bus, see section 15.7. Bus Error Monitoring Section.

MSERRSTAT bit (Master Security Attribution Unit Error Status)

When MSAU error occurs by bus, BUSERRSTAT<Master Name>.MSERRSTAT is set to 1.

There is clearing conditions.

- Set BUSERRCLR<Master Name>.MSERRCLR to 1.

For detail of MSAU error that occurs by bus, see section 15.7. Bus Error Monitoring Section.

Table 15.14 Associations between bus types and registers

n	Bus master type	Bus Error Status Register (0x1A00 + 0x10 × n)	Bus Error Clear Register (0x1A04 + 0x10 × n)	Bus Error IRQ Enable Register (0x1A08 + 0x10 × n)
0	CPU0	BUSERRSTATCPU0	BUSERRCLRCPU0	BUSIRQENCPU0
1	CPU1	BUSERRSTATCPU1	BUSERRCLRCPU1	BUSIRQENCPU1
2	DMAC/DTC0	BUSERRSTATDMAC0	BUSERRCLRDMAC0	BUSIRQENDMAC0
3	DMAC/DTC1	BUSERRSTATDMAC1	BUSERRCLRDMAC1	BUSIRQENDMAC1
5	EDMAC	BUSERRSTATEDMAC	BUSERRCLREDMAC	BUSIRQENEDMAC
10	CEU	BUSERRSTATCEU	BUSERRCLRCEU	BUSIRQENCEU

15.3.26 BUSERRCLR<Master Name> : BUS Error Clear Register

(Master Name: CPU0, CPU1, DMAC0, DMAC1, EDMAC, CEU)

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x1A04 (BUSERRCLRCPU0), 0x1A14 (BUSERRCLRCPU1), 0x1A24 (BUSERRCLRDMAC0), 0x1A34 (BUSERRCLRDMAC1), 0x1A54 (BUSERRCLREDMAC), 0x1AA4 (BUSERRCLRCEU)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	MSERRCLR	ILERRCLR	MMERRCLR	—	—	SLERRCLR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SLERRCLR	Slave Bus Error Clear 0: No effect 1: Clear the BUSERRSTAT<Master Name>.SLERRSTAT flag	W
2:1	—	The write value should be 0.	W
3	MMERRCLR	Master MPU Error Clear 0: No effect 1: Clear the BUSERRSTAT<Master Name>.MMERRSTAT flag	W
4	ILERRCLR	Illegal Address Access Error Clear 0: No effect 1: Clear the BUSERRSTAT<Master Name>.ILERRSTAT flag	W
5	MSERRCLR	Master Security Attribution Unit Error Clear 0: No effect 1: Clear the BUSERRSTAT<Master Name>.MSERRSTAT flag	W
7:6	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-2

When writing 1 to BUSERRCLR<Master Name>, stop the bus access that causes an error in the corresponding bus master.

15.3.27 BUSOAD : BUS Operation After Detection Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x1000

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

Bit	Symbol	Function	R/W
0	ILERROAD	Illegal address access error operation after detection All bus master 0: Interrupt 1: Reset	R/W
1	SLERROAD	Slave bus error operation after detection All bus master 0: Interrupt 1: Reset	R/W
2	BWERROAD	Bufferable write error operation after detection All bus master 0: Interrupt 1: Reset	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: When an error occurs in speculative access issued by CPU0 and CPU1, interrupt or a system reset request may occur depending on the setting values of the OAD bit and BUSIRQEN<Master Name>.EN. In that case, it can be avoided by using Arm-MPU. See section 15.8. References[3] for the detail.

ILERROAD bit (Illegal address access error operation after detection)

The OAD bit specify operation when the illegal address access error is detected. When OAD bit is 0, error response is returned and if BUSIRQEN<Master Name>.EN = 1, and interrupt request is generated. When OAD bit is 1, error response is returned and reset request is generated.

SLERROAD bit (Slave bus error operation after detection)

The OAD bit specify operation when the slave bus error is detected. When OAD bit is 0, error response is returned and if BUSIRQEN<Master Name>.EN = 1, and interrupt request is generated. When OAD bit is 1, error response is returned and reset request is generated.

BWERROAD bit (Bufferable write error operation after detection)

The OAD bit specify operation when the bufferable write error is detected.

When OAD bit is 0, interrupt request is generated.

When OAD bit is 1, reset request is generated.

Interrupts due to bus errors (including bufferable errors) can also be used as NMI. See the section 14, Interrupt Controller Unit (ICU) NMIER register specifications for details.

15.3.28 BUSOADPT : BUS Operation After Detection Protect Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x1004

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

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: BUSOAD register writing is possible. 1: BUSOAD register writing is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key code These bits enable or disable writes to the PROTECT bit.	W

Note: S-TYPE-3, P-TYPE-2

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writes to the associated registers to be protected.

BUSOADPT.PROTECT controls the following registers:

- BUSOAD

When the PROTECT bit is set simultaneously, write 0xA5 to the KEY[7:0] bits using half word access.

KEY[7:0] bits (Key code)

The KEY[7:0] bits enable or disable writes to the PROTECT bit. When writing to the PROTECT bit simultaneously, write 0xA5 to the KEY[7:0] bits. When other values are written to the KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.

15.3.29 MBWERRSTAT : Master Bufferable Write Error Status Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x1B00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBWERR19	—	MBWERR17	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	MBWERR9	MBWERR8	—	—	—	—	—	—	MBWERR1	MBWERR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBWERR0	Master Bufferable Write Error 0: No bufferable write error in Master #0 1: Bufferable write error occurs in Master #0	R
1	MBWERR1	Master Bufferable Write Error 0: No bufferable write error in Master #1 1: Bufferable write error occurs in Master #1	R
7:2	—	These bits are read as 0.	R
8	MBWERR8	Master Bufferable Write Error 0: No bufferable write error in Master #8 1: Bufferable write error occurs in Master #8	R
9	MBWERR9	Master Bufferable Write Error 0: No bufferable write error in Master #9 1: Bufferable write error occurs in Master #9	R
16:10	—	These bits are read as 0.	R
17	MBWERR17	Master Bufferable Write Error 0: No bufferable write error in Master #17 1: Bufferable write error occurs in Master #17	R
18	—	This bit is read as 0.	R
19	MBWERR19	Master Bufferable Write Error 0: No bufferable write error in Master #19 1: Bufferable write error occurs in Master #19	R
31:20	—	These bits are read as 0.	R

Note: S-TYPE-5, P-TYPE-2
Note: This register is initialized after returning from Software Standby mode while the debugger is connected.
Note: This register is cleared by reset other than Bus error reset, Common memory reset, and CPU local memory reset.

MBWERRn bit (Master Bufferable Write Error n)

When bufferable write access error occurs, MBWERRSTAT. MBWERRn is set to 1.

There are clearing conditions.

- Set MBWERRCLR. MBWECLRn to 1

If an error is detected on the slave side, the error flag of the master with the access path is set at the same time in addition to the flag of the master that issued the access.

Table 15.15 lists the registers associated with each bus master type.

Table 15.15 Associations between bus master types and bits

Bus master type	Master bufferable write error status bit	Master bufferable write error clear bit
CPU0	MBWERR0	MBWECLR0
CPU1	MBWERR1	MBWECLR1
DMAC/DTC0	MBWERR8	MBWECLR8
DMAC/DTC1	MBWERR9	MBWECLR9
EDMAC	MBWERR17	MBWECLR17
CEU	MBWERR19	MBWECLR19

15.3.30 MBWERRCLR : Master Bufferable Write Error Clear Register

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x1B08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBWECLR19	—	MBWECLR17	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	MBWECLR9	MBWECLR8	—	—	—	—	—	—	MBWECLR1	MBWECLR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBWECLR0	Master Bufferable Write Error Clear 0: No effect 1: Clear the MBWERRSTAT.MBWERR0 flag	W
1	MBWECLR1	Master Bufferable Write Error Clear 0: No effect 1: Clear the MBWERRSTAT.MBWERR1 flag	W
7:2	—	The write value should be 0.	W
8	MBWECLR8	Master Bufferable Write Error Clear 0: No effect 1: Clear the MBWERRSTAT.MBWERR8 flag	W
9	MBWECLR9	Master Bufferable Write Error Clear 0: No effect 1: Clear the MBWERRSTAT.MBWERR9 flag	W
16:10	—	These bits are read as 0.	W
17	MBWECLR17	Master Bufferable Write Error Clear 0: No effect 1: Clear the MBWERRSTAT.MBWERR17 flag	W
18	—	The write value should be 0.	W
19	MBWECLR19	Master Bufferable Write Error Clear 0: No effect 1: Clear the MBWERRSTAT.MBWERR19 flag	W
31:20	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-2

15.3.31 BUSIRQEN<Master Name> : BUS Error Interrupt Enable Register

(Master Name: CPU0, CPU1, DMAC0, DMAC1, EDMAC, CEU)

Base address: BUS = 0x4000_3000
BUS_NS = 0x5000_3000

Offset address: 0x1A08 (BUSIRQENCPU0), 0x1A18 (BUSIRQENCPU1), 0x1A28 (BUSIRQENDMAC0), 0x1A38 (BUSIRQENDMAC1), 0x1A58 (BUSIRQENEDMAC), 0x1AA8 (BUSIRQENCEU)

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

Bit	Symbol	Function	R/W
0	EN	Interrupt enable after bus error detected Bus interrupt request permission setting to ICU when a bus error (except Master Bufferable Write Error) occurs. 0: Disable 1: Enable	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

When writing 1 to BUSIRQEN<Master Name>, stop the bus access that causes an error in the corresponding bus master.

EN bit (Interrupt enable after bus error detected)

To output a bus error interrupt request to the ICU due to an error, select the interrupt in the OAD setting corresponding to each error, and enable the interrupt request with this bit. The setting of this bit does not depend on the output a bus error interrupt request due to Master Bufferable Write Error.

15.4 Endian and Data Alignment

The external bus has a data alignment function to control which byte of the data bus (D31 to D24, D23 to D16, D15 to D08, D07 to D00) is used when accessing the external address space (the CS and SDRAM areas). Alignment is based on the bus specifications of the area to be accessed (8-bit, 16-bit, or 32-bit bus space), the data size, and the endian order.

15.4.1 Data Alignment Control for the CS Area

15.4.1.1 32-Bit Bus Space

When a 32-bit width is selected for a bus space by the BSIZE[1:0] bits in CSnCR, the address buses A23 to A2 are enabled to output address signals in units of 32 bits, and the address buses A1 and A0 are disabled (always output low).

When byte strobe mode is selected (the WRMOD bit = 0 in CSnMOD), the WR0 to WR3 pins are enabled. The BC0 to BC3 pins are not used.

When single-write strobe mode is selected (the WRMOD bit = 1 in CSnMOD), only the WR0 pin is enabled and always outputs the low level during write access, regardless of the data size. The WR1 to WR3 pins are invalid (always output high). The valid byte position is indicated by the BC0 to BC3 pins.

In 32-bit bus space, the valid positions of data external to the MCU and of control signals differ according to whether the endian is big or little.

Table 15.16 Data alignment (little-endian) in 32-bit bus space

Data size	Access address	Number of access	Bus cycle	Unit of data	Address	WR3/BC3	WR2/BC2	WR1/BC1	WR0/BC0
						RD			
						Data bus			
						D31 D24	D23 D16	D15 D8	D7 D0
8 bits	4n	One	First	8 bits	4n				7 0
	4n + 1	One			4n			7 0	
	4n + 2	One	First	8 bits	4n		7 0		
	4n + 3	One			4n	7 0			
16 bits	4n	One	First	16 bits	4n			15 8	7 0
	4n + 2	One	First	16 bits	4n	15 8	7 0		
32 bits	4n	One	First	32 bits	4n	31 24	23 16	15 8	7 0
64 bits	4n	Two	First	32 bits	4n	31 24	23 16	15 8	7 0
			Second	32 bits	4n + 4	63 56	55 48	47 40	39 32

Table 15.17 Data alignment (big-endian) in 32-bit bus space

Data size	Access address	Number of access	Bus cycle	Unit of data	Address	WR3/BC3	WR2/BC2	WR1/BC1	WR0/BC0
						RD			
						Data bus			
						D31 D24	D23 D16	D15 D8	D7 D0
8 bits	4n	One	First	8 bits	4n	7 0			
	4n + 1	One			4n		7 0		
	4n + 2	One	First	8 bits	4n			7 0	
	4n + 3	One			4n				7 0
16 bits	4n	One	First	16 bits	4n	15 8	7 0		
	4n + 2	One	First	16 bits	4n			15 8	7 0
32 bits	4n	One	First	32 bits	4n	31 24	23 16	15 8	7 0

15.4.1.2 16-Bit Bus Space

When a 16-bit bus space is selected in the BSIZE[1:0] bits in CSnCR, address buses A23 to A01 are enabled to output address signals in 16-bit units, and the address bus A00 is disabled (always outputs low).

When byte strobe mode is selected (WRMOD = 0 in CSnMOD), the WR0 and WR1 pins are enabled, and the WR2 and WR3 pins are disabled (fixed high). The BC0 to BC3 pins are not used.

When single-write strobe mode is selected (WRMOD = 1 in CSnMOD), only the WR0 pin is enabled, and it always outputs low during write access, regardless of the data size. The WR1 to WR3 pins are invalid (always outputs high). The valid byte position is indicated by the BC0 and BC1 pins.

The valid positions of control signals and data external to the MCU differ depending on the endian order. See Table 15.18 and Table 15.19.

Page access can occur for accesses to data in 32-bit units. Page access can only occur when an access does not extend over a 32-bit boundary and causes no change in the BC0 and BC1 signals. The conditions in which page access occurs are indicated by the letter (p) in Table 15.18 and Table 15.19.

Table 15.18 Data alignment (little-endian) in 16-bit bus space

							WR1/BC1	WR0/BC0
							RD	
							Data bus	
Data size	**Accessed address**	**Number of accesses**	**Bus cycle**	**Unit of data**	**Address**		**D15 D08**	**D07 D00**
8 bits	4n	One	First	8 bits	4n			7 0
	4n + 1	One	First	8 bits	4n		7 0	
	4n + 2	One	First	8 bits	4n + 2			7 0
	4n + 3	One	First	8 bits	4n + 2		7 0	
16 bits	4n	One	First	16 bits	4n		15 8	7 0
	4n+2	One	First	16 bits	4n + 2		15 8	7 0
32 bits	4n	Two	First	16 bits	4n		15 8	7 0
			Second	16 bits	4n + 2	(p)	31 24	23 16
64 bits	4n	Four	First	16 bits	4n		15 8	7 0
			Second	16 bits	4n + 2	(p)	31 24	23 16
			Third	16 bits	4n + 4		47 40	39 32
			Forth	16 bits	4n + 6	(p)	63 56	55 48

(p):Page access (only when page access is enabled in the PRENB and PWENB bits in CSnMOD)

Table 15.19 Data alignment (big-endian) in 16-bit bus space

Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address		WR1/BC1 RD Data bus D15 … D08	WR0/BC0 RD Data bus D07 … D00
8 bits	4n	One	First	8 bits	4n		7 … 0	
	4n + 1	One	First	8 bits	4n			7 … 0
	4n + 2	One	First	8 bits	4n + 2		7 … 0	
	4n + 3	One	First	8 bits	4n + 2			7 … 0
16 bits	4n	One	First	16 bits	4n		15 … 8	7 … 0
	4n+2	One	First	16 bits	4n + 2		15 … 8	7 … 0
32 bits	4n	Two	First	16 bits	4n		31 … 24	23 … 16
			Second	16 bits	4n + 2	(p)	15 … 8	7 … 0

(p):Page access (only when page access is enabled in the PRENB and PWENB bits in CSnMOD)

15.4.1.3 8-Bit Bus Space

When an 8-bit bus space is selected by the BSIZE[1:0] bits in CSnCR, the address buses A23 to A0 are enabled to output address signals in byte units.

In 8-bit bus space, only the WR0# pin is valid regardless of write access mode, and always outputs the low level during write access. The WR1# to WR3# pins and the BC0# to BC3# pins are not used.

Page access can occur in access to data in 16-, 32- or 64-bit units. Specifically, page access can occur when an access does not spread over a 32-bit boundary. The conditions in which page access occurs are indicated by the letter (p) in Table 15.20 and Table 15.21.

In 8-bit bus space, the valid positions of data external to the MCU are D7 to D0 and WR0# is used as the control signal, regardless of the endian mode.

Table 15.20 Data alignment (little-endian) in 8-bit bus space

Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address		Data bus (RD) WR1/BC1 D15 – D08	Data bus (RD) WR0/BC0 D07 – D00
8 bits	4n	One	First	8 bits	4n			7 0
	4n + 1	One	First	8 bits	4n + 1			7 0
	4n + 2	One	First	8 bits	4n + 2			7 0
	4n + 3	One	First	8 bits	4n + 3			7 0
16 bits	4n	Two	First	8 bits	4n			7 0
			Second	8 bits	4n + 1	(p)		15 8
	4n + 2	Two	First	8 bits	4n + 2			7 0
			Second	8 bits	4n + 3	(p)		15 8
32 bits	4n	Four	First	8 bits	4n			7 0
			Second	8 bits	4n + 1	(p)		15 8
			Third	8 bits	4n + 2	(p)		23 16
			Fourth	8 bits	4n + 3	(p)		31 24
64 bits	4n	Eight	First	8 bits	4n			7 0
			Second	8 bits	4n + 1	(p)		15 8
			Third	8 bits	4n + 2	(p)		23 16
			Fourth	8 bits	4n + 3	(p)		31 24
			Fifth	8 bits	4n + 4			39 32
			Sixth	8 bits	4n + 5	(p)		47 40
			Seventh	8 bits	4n + 6	(p)		55 48
			Eighth	8 bits	4n + 7	(p)		63 56

(p):Page access (only when page access is enabled in the PRENB and PWENB bits in CSnMOD)

Table 15.21 Data alignment (big-endian) in 8-bit bus space

Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address		Data bus (RD) WR1/BC1 D15 D08	Data bus (RD) WR0/BC0 D07 D00
8 bits	4n	One	First	8 bits	4n			7 0
	4n + 1	One	First	8 bits	4n + 1			7 0
	4n + 2	One	First	8 bits	4n + 2			7 0
	4n + 3	One	First	8 bits	4n + 3			7 0
16 bits	4n	Two	First	8 bits	4n			15 8
			Second	8 bits	4n + 1	(p)		7 0
	4n + 2	Two	First	8 bits	4n + 2			15 8
			Second	8 bits	4n + 3	(p)		7 0
32 bits	4n	Four	First	8 bits	4n			31 24
			Second	8 bits	4n + 1	(p)		23 16
			Third	8 bits	4n + 2	(p)		15 8
			Fourth	8 bits	4n + 3	(p)		7 0

(p):Page access (only when page access is enabled in the PRENB and PWENB bits in CSnMOD)

15.4.2 Data Alignment Control for the SDRAM Area

15.4.2.1 32-Bit Bus Space

When a 32-bit width is selected for a bus space by the BSIZE[1:0] bits in SDCCR, the address buses A26 to A02 are enabled to output address signals in units of 32 bits, and the address buses A01 and A00 are disabled (always output low). The valid byte position is indicated by DQM0 to DQM3 signals.

The external data is accessed using the D31 to D24, D23 to D16, D15 to D8, and D7 to D0 pins. Data can be accessed in either 8-bit, 16-bit, or 32-bit units at a time.

In 32-bit bus space, the valid positions of data external to the chip and of SDRAM control signals (DQM0 to DQM3) differ according to whether the endian is big or little. Table 15.22 and Table 15.23 show data alignment control when the endian is little and big, respectively.

When the SDAMOD.BE bit is set to 1, consecutive access may occur under the following conditions.

- When access to the same row address is accessed consecutively.
- Access with a data size of 64 bits that does not cross 64-bit address boundaries.

The conditions in which consecutive access occurs are indicated by the letter "(r1)" in Table 15.22 and Table 15.23.

Table 15.22 Data alignment (little-endian) in 32-bit bus space

Data size	Access address	Number of access	Bus cycle	Unit of data	Address	DQM3	DQM2	DQM1	DQM0
						WE			
						Data bus			
						D31 D24	D23 D16	D15 D8	D7 D0
8 bits	4n	One	First	8 bits	4n (r1)				7 0
	4n + 1	One			4n (r1)			7 0	
	4n + 2	One	First	8 bits	4n (r1)		7 0		
	4n + 3	One			4n (r1)	7 0			
16 bits	4n	One	First	16 bits	4n (r1)			15 8	7 0
	4n + 2	One	First	16 bits	4n (r1)	15 8	7 0		
32 bits	4n	One	First	32 bits	4n (r1)	31 24	23 16	15 8	7 0
64 bits	4n	Two	First	32 bits	4n (r1)	31 24	23 16	15 8	7 0
			Second	32 bits	4n + 4 (r1)	63 56	55 48	47 40	39 32

Table 15.23 Data alignment (big-endian) in 32-bit bus space (1 of 2)

Data size	Access address	Number of access	Bus cycle	Unit of data	Address	DQM3	DQM2	DQM1	DQM0
						WE			
						Data bus			
						D31 D24	D23 D16	D15 D8	D7 D0
8 bits	4n	One	First	8 bits	4n (r1)	7 0			
	4n + 1	One			4n (r1)		7 0		
	4n + 2	One	First	8 bits	4n (r1)			7 0	
	4n + 3	One			4n (r1)				7 0

Table 15.23 Data alignment (big-endian) in 32-bit bus space (2 of 2)

Data size	Access address	Number of access	Bus cycle	Unit of data	Address	DQM3	DQM2	DQM1	DQM0
						WE			
						Data bus			
						D31 … D24	D23 … D16	D15 … D8	D7 … D0
16 bits	4n	One	First	16 bits	4n (r1)	15 … 8	7 … 0		
	4n + 2	One	First	16 bits	4n (r1)			15 … 8	7 … 0
32 bits	4n	One	First	32 bits	4n (r1)	31 … 24	23 … 16	15 … 8	7 … 0

15.4.2.2 16-Bit Bus Space

When a 16-bit bus space is selected in the BSIZE[1:0] bits in SDCCR, address buses A26 to A01 are enabled to output address signals in 16-bit units, and the address bus A00 is disabled (always outputs low). The valid byte position is indicated by the DQM0 and DQM1 signals.

External data is accessed using the DQ15 to DQ08 and DQ07 to DQ00 pins and DQM0 and DQM1 control signals. Data can be accessed in either 8-bit or 16-bit units at a time.

When the SDAMOD.BE bit is set to 1, consecutive access may occur under the following conditions.

- When access to the same row address is accessed consecutively.
- Access with a data size of 32 bits
- Access with a data size of 64 bits that does not cross 64-bit address boundaries.
- Access with a data size of 64 bits that does not cross 32-bit address boundaries.
- The conditions in which consecutive access occurs are indicated by the letter "(r1)" in Table 15.24 and Table 15.25.

The valid positions of control signals and data external to the MCU differ depending on the endian order. See Table 15.24 and Table 15.25.

Table 15.24 Data alignment in 16-bit bus space with little-endian order for SDRAM area (1 of 2)

Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address		DQM1	DQM0
							WE	
							Data bus	
							DQ15 … DQ08	DQ07 … DQ00
8 bits	4n	One	First	8 bits	4n	(r1)		7 … 0
	4n + 1	One	First	8 bits	4n	(r1)	7 … 0	
	4n + 2	One	First	8 bits	4n + 2	(r1)		7 … 0
	4n + 3	One	First	8 bits	4n + 2	(r1)	7 … 0	

Table 15.24 Data alignment in 16-bit bus space with little-endian order for SDRAM area (2 of 2)

Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address		DQM1		DQM0	
							WE			
							Data bus			
							DQ15	DQ08	DQ07	DQ00
16 bits	4n	One	First	16 bits	4n	(r1)	15	8	7	0
	4n+2	One	First	16 bits	4n + 2	(r1)	15	8	7	0
32 bits	4n	Two	First	16 bits	4n	(r1)	15	8	7	0
			Second	16 bits	4n + 2	(r1)	31	24	23	16
64 bits	4n	Four	First	16 bits	4n	(r1)	15	8	7	0
			Second	16 bits	4n + 2	(r1)	31	24	23	16
			Third	16 bits	4n + 4	(r1)	47	40	39	32
			Fourth	16 bits	4n + 6	(r1)	63	56	55	48

Table 15.25 Data alignment in 16-bit bus space with big-endian order for SDRAM area

Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address		DQM1		DQM0	
							WE			
							Data bus			
							DQ15	DQ08	DQ07	DQ00
8 bits	4n	One	First	8 bits	4n	(r1)	7	0		
	4n + 1	One	First	8 bits	4n	(r1)			7	0
	4n + 2	One	First	8 bits	4n + 2	(r1)	7	0		
	4n + 3	One	First	8 bits	4n + 2	(r1)			7	0
16 bits	4n	One	First	16 bits	4n	(r1)	15	8	7	0
	4n+2	One	First	16 bits	4n + 2	(r1)	15	8	7	0
32 bits	4n	Two	First	16 bits	4n	(r1)	31	24	23	16
			Second	16 bits	4n + 2	(r1)	15	8	7	0

15.4.2.3 8-Bit Bus Space

When an 8-bit width is selected in the BSIZE[1:0] bits in SDCCR, address buses A26 to A00 are enabled to output address signals in 8-bit units.

External data is accessed using the DQ07 to DQ00 pins and DQM0 control signal. Eight-bit data is accessed in single 8-bit access, 16-bit data with two 8-bit accesses, and 32-bit data with four 8-bit accesses.

When the SDAMOD.BE bit is set to 1, consecutive access may occur under the following conditions.

- When access to the same row address is accessed consecutively.
- Access with a data size of 16 / 32bit.
- Access with a data size of 64 bits that does not cross 64-bit address boundaries.
- Access with a data size of 64 bits that does not cross 32-bit address boundaries.

Table 15.26 and Table 15.27 show the case where continuous access occurs (r1).

The valid positions of control signals and data external to the MCU differ depending on the endian order. See Table 15.26 and Table 15.27.

Table 15.26 Data alignment in 8-bit bus space with little-endian order for SDRAM area (1 of 2)

							DQM1		DQM0	
							WE			
							Data bus			
Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address		DQ15	DQ08	DQ07	DQ00
8 bits	4n	One	First	8 bits	4n	(r1)			7	0
	4n + 1	One	First	8 bits	4n + 1	(r1)			7	0
	4n + 2	One	First	8 bits	4n + 2	(r1)			7	0
	4n + 3	One	First	8 bits	4n + 3	(r1)			7	0
16 bits	4n	Two	First	8 bits	4n	(r1)			7	0
			Second	8 bits	4n + 1	(r1)			15	8
	4n + 2	Two	First	8 bits	4n + 2	(r1)			7	0
			Second	8 bits	4n + 3	(r1)			15	8
32 bits	4n	Four	First	8 bits	4n	(r1)			7	0
			Second	8 bits	4n + 1	(r1)			15	8
			Third	8 bits	4n + 2	(r1)			23	16
			Fourth	8 bits	4n + 3	(r1)			31	24

Table 15.26 Data alignment in 8-bit bus space with little-endian order for SDRAM area (2 of 2)

Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address		DQM1 WE Data bus DQ15	DQM1 WE Data bus DQ08	DQM0 WE Data bus DQ07	DQM0 WE Data bus DQ00
64 bits	4n	Eight	First	8 bits	4n	(r1)			7	0
			Second	8 bits	4n + 1	(r1)			15	8
			Third	8 bits	4n + 2	(r1)			23	16
			Fourth	8 bits	4n + 3	(r1)			31	24
			Fifth	8 bits	4n + 4	(r1)			39	32
			Sixth	8 bits	4n + 5	(r1)			47	40
			Seventh	8 bits	4n + 6	(r1)			55	48
			Eighth	8 bits	4n + 7	(r1)			63	56

Table 15.27 Data alignment in 8-bit bus space with big-endian order for SDRAM area (1 of 2)

Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address		DQM1 WE Data bus DQ15	DQM1 WE Data bus DQ08	DQM0 WE Data bus DQ07	DQM0 WE Data bus DQ00
8 bits	4n	One	First	8 bits	4n	(r1)			7	0
	4n + 1	One	First	8 bits	4n + 1	(r1)			7	0
	4n + 2	One	First	8 bits	4n + 2	(r1)			7	0
	4n + 3	One	First	8 bits	4n + 3	(r1)			7	0
16 bits	4n	Two	First	8 bits	4n	(r1)			15	8
			Second	8 bits	4n + 1	(r1)			7	0
	4n + 2	Two	First	8 bits	4n + 2	(r1)			15	8
			Second	8 bits	4n + 3	(r1)			7	0

Table 15.27 Data alignment in 8-bit bus space with big-endian order for SDRAM area (2 of 2)

Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address		DQM1: DQ15 – DQ08 (WE, Data bus)	DQM0: DQ07 – DQ00 (WE, Data bus)
32 bits	4n	Four	First	8 bits	4n	(r1)		31 – 24
			Second	8 bits	4n + 1	(r1)		23 – 16
			Third	8 bits	4n + 2	(r1)		15 – 8
			Fourth	8 bits	4n + 3	(r1)		7 – 0

15.5 Operation of CS Area Controller

15.5.1 Separate Bus

This section describes the periods shown in the timing diagrams. The CS area controller (CSC) operates in synchronization with the external bus clock, BCLK. Operation cycles, such as wait cycles, specified in the CSC register, are counted on BCLK. In the following description, the frequencies of BCLK and EBCLK pin output are the same, unless otherwise noted. Access through the external bus starts at the same point as the output of a rising edge on the EBCLK pin. However, if the external bus clock, BCLK, and the output on the EBCLK pin are at different frequencies, the wait settings can cause the start of access for the second and subsequent rounds to coincide with the falling edge of the output on the EBCLK pin. See Figure 15.11 to Figure 15.15. If recovery cycles are inserted for bus access, the setting for the number of recovery cycles can also cause the start of access for the second and subsequent rounds to coincide with the falling edge of the output on the EBCLK pin. See Figure 15.33.

Tw1 to Twn (Clock Cycles of Wait for a Normal Read Cycle or Normal Write Cycle)

The period from Tw1 to Twn is the number of clock cycles from the start of access through the external bus clock to 1 cycle before the strobe signal is valid. The number of cycles is selectable from 0 to 31. Within this period, the timing of CSn, RD, and WRn assertion (driving the signals low) is determined by the respective wait settings. The wait periods are controlled by the following bits in the CSn Wait Control Register 2 (CSnWCR2).

- CS Assert Wait Select bits (CSON)
- RD Assert Wait Select bits (RDON)
- WR Assert Wait Select bits (WRON)
- Write-Data Output Wait Select bits (WDON)

The number of clock cycles for each of these wait periods is selectable as a value from 0 to 7, counted from the start of external bus access. The selectable number of cycles is also within the overall number of clock cycles required for waiting to read or write.

Tend (Clock Cycle where the Strobe Signal is Valid)

Tend is the next clock cycle after completion of the wait period for a normal cycle of read or write, or for a cycle of page read or page write. If the wait select bit for these cycles is 0, bus access starts on the clock cycle where the strobe signal is valid. The RD and WRn signals are negated in the next clock cycle. For a read access, the clock cycle where the strobe signal is valid is where the data to be read is sampled. If an external wait is enabled, the wait signal is sampled on the cycle where the strobe signal is valid. The bus cycle is extended if the wait signal is low. The bus cycle completes in the next clock cycle if the wait signal is high. Tend indicates the cycle where sampling of the wait signal starts.

After the first cycle where the strobe signal is valid during page access, second and subsequent page access operations (see Tpw1 to Tpwn (Page-Read Cycle Wait or Page-Write Cycle Wait)) start in the next cycle, except during write access with a setting other than 0 for write-data output extension clock cycles (see Tdw1 to Tdwn (Write-Data Output Extension Clock

Cycles)). If the setting for the RD or WR assertion wait is any value other than 0, the RD and WRn signals are negated in the next clock cycle. If the setting is 0, assertion continues. Additionally, the CSn signal continues to be asserted rather than negated.

Tn1 to Tnm (Clock Cycles of CS Extension)

For normal access, Tn1 to Tnm represent the clock cycles of the period following the cycle where the strobe signal is valid (Tend) up to negation of the CSn signal. For read or write access, the negation timing can be controlled by the read-access CS Extension Cycle Select bits (CSROFF) and the write-access CS Extension Cycle Select bits (CSWOFF) in the CSn Wait Control Register 2 (CSnWCR2). The number of cycles is counted from the cycle following the cycle where the strobe signal is valid.

For page access, Tn1 to Tnm represent the clock cycles of the period following the last cycle where the strobe signal is valid up to negation of the CSn signal.

For write access, setting the Write Data Output Extension Cycle Select bits (WDOFF) controls extension of the period where the address and output data is valid.

Tdw1 to Tdwn (Write-Data Output Extension Clock Cycles)

For write access, if the wait setting for the write-data output extension is any value other than 0, the specified clock cycles are inserted from the cycle following the cycle where the strobe signal is valid (Tend).

For normal access, this period is inserted within the clock cycle period for CS extension (Tn1 to Tnm).

For page access, this period is inserted within the clock cycle period where the strobe signal is valid and subsequent page accesses, or within the clock cycle period for the CS extension (Tn1 to Tnm). Valid address and data output are extended over this period, and the WRn signal is negated.

Tpw1 to Tpwn (Page-Read Cycle Wait or Page-Write Cycle Wait)

For the second and subsequent bus cycles during page access, the values for a page-read cycle wait or page-write cycle wait are used instead of the settings for a normal read or write cycle wait. The settings in the WR Assert Wait Select bits become enabled in the same way as for the first access. The RD assertion control operation depends on the page read access mode setting (the PRMOD bit in CSnMOD) as follows:

- CSnMOD.PRMOD = 0: A wait for RD assertion is inserted in the same way as for the first access, and the RD signal is negated
- CSnMOD.PRMOD = 1: Although a wait for RD assertion is inserted in the same way as for normal-access compatibility mode, the RD signal continues to be asserted over this period.

Tr1 to Trn (Recovery Cycles)

Recovery cycles can be inserted from the point where a bus cycle is complete (CSn signal negation). The number of recovery cycles can be controlled by setting the Read Recovery (RRCV) or Write Recovery (WRCV) bits in the CSn Recovery Cycle Register (CSnREC). Both numbers of recovery cycles are counted from the end of a bus cycle (CSn negation) and can be selected from 0 to 15 cycles. For more information, see section 15.5.4. Insertion of Recovery Cycles.

(1) Normal Access

When the PRENB and PWENB bits in CSnMOD are set to 0 to disable page read and page write access, all bus accesses take the form of normal read and write operations. Even when these bits are set to 1 to enable page read and page write access, bus access other than page access takes the form of normal read and write operations. Figure 15.4 to Figure 15.6 show the normal access operations.

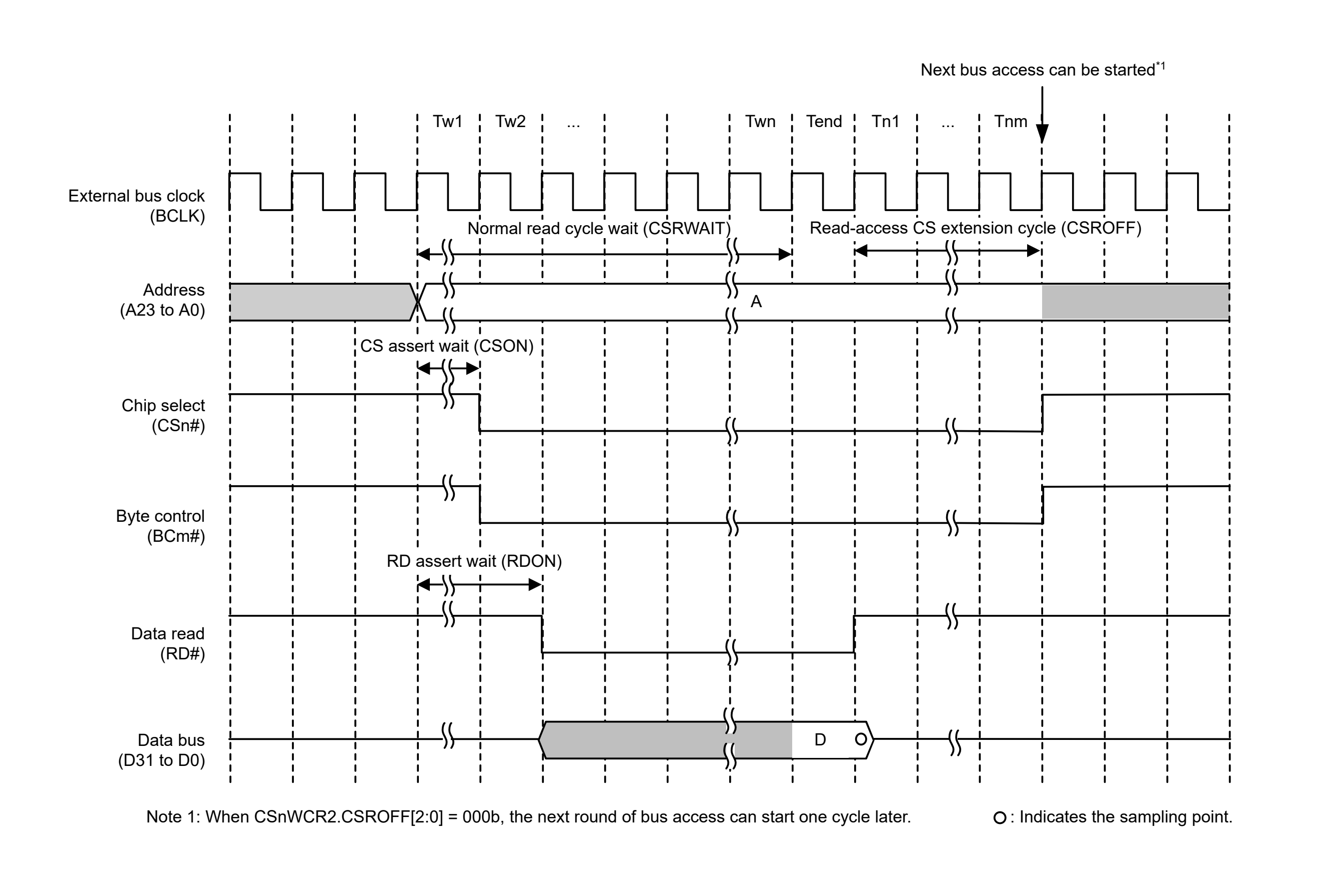


Figure 15.4 Bus timing for normal read operation (n = 0 to 7, m = 0 to 3)

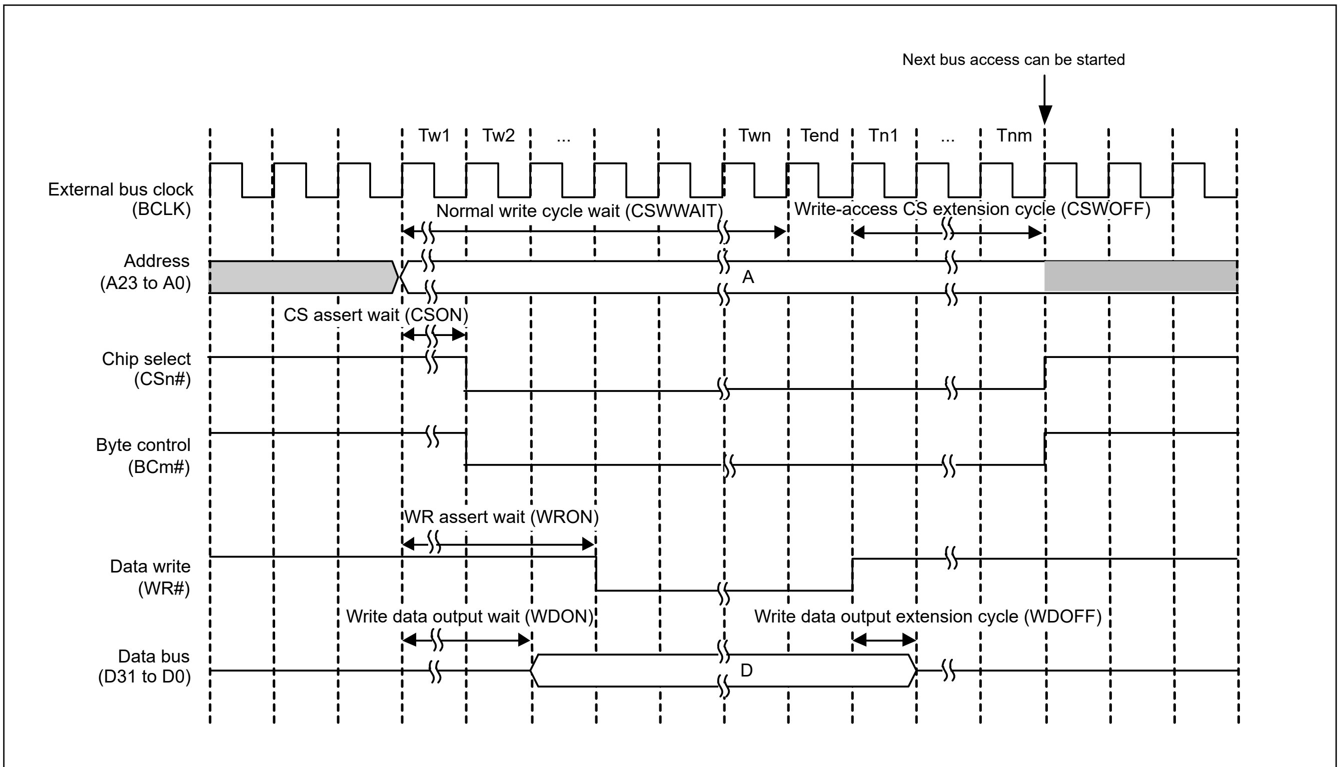


Figure 15.5 Bus timing for normal write operation in single-write strobe mode (n = 0 to 7, m = 0 to 3)

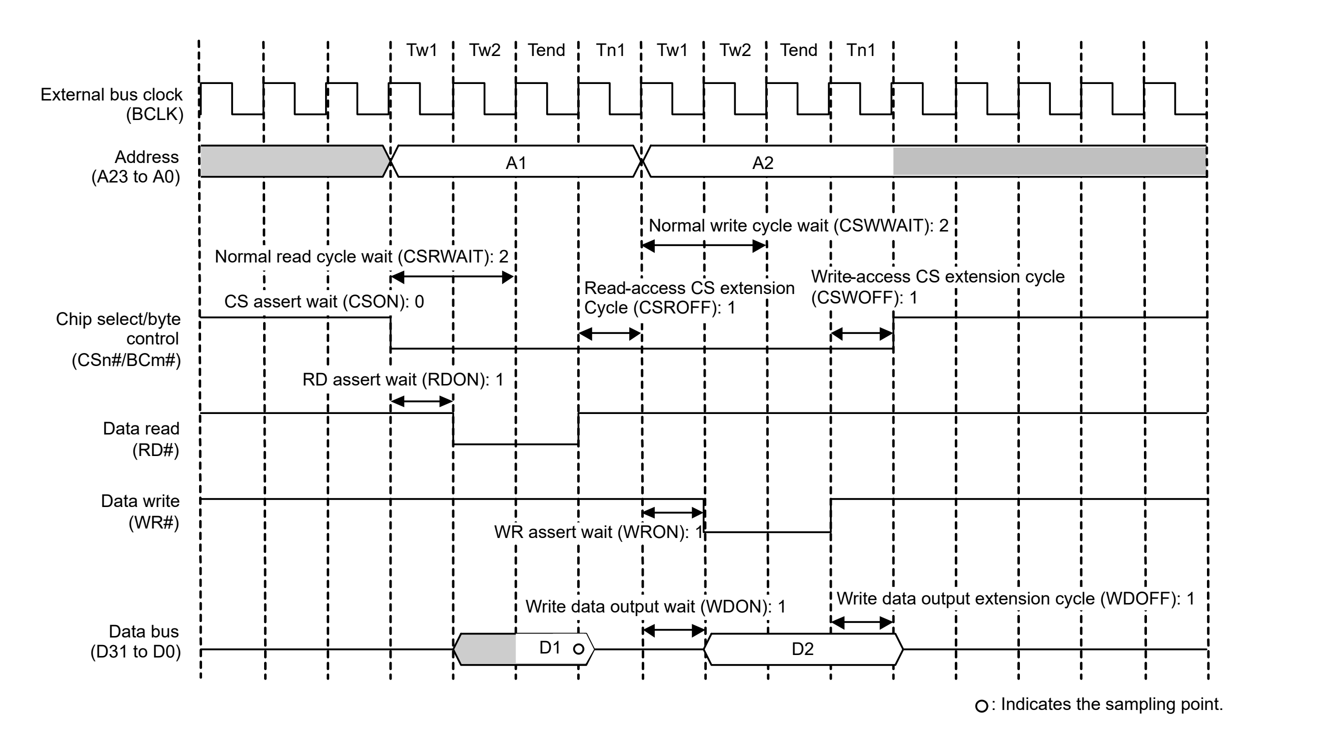


Figure 15.6 Example of normal access operation for read and write (n = 0 to 7, m = 0 to 3)

When two or more rounds of external bus access are required in response to a single request for transfer from a bus master, normal access operations are repeated. See Tw1 to Twn (Clock Cycles of Wait for a Normal Read Cycle or Normal Write Cycle) to Tdw1 to Tdwn (Write-Data Output Extension Clock Cycles). Figure 15.7 and Figure 15.8 show examples of operations when two rounds of bus access are generated in response to a single transfer request. If the recovery cycle insertion condition is satisfied, recovery cycles Tr1 to Trn (Recovery Cycles) are also inserted in the second and subsequent external bus accesses. See Figure 15.31.

The values in the wait control registers shown in the figures are example settings. In your application, set the registers appropriately for the specifications of connected devices.

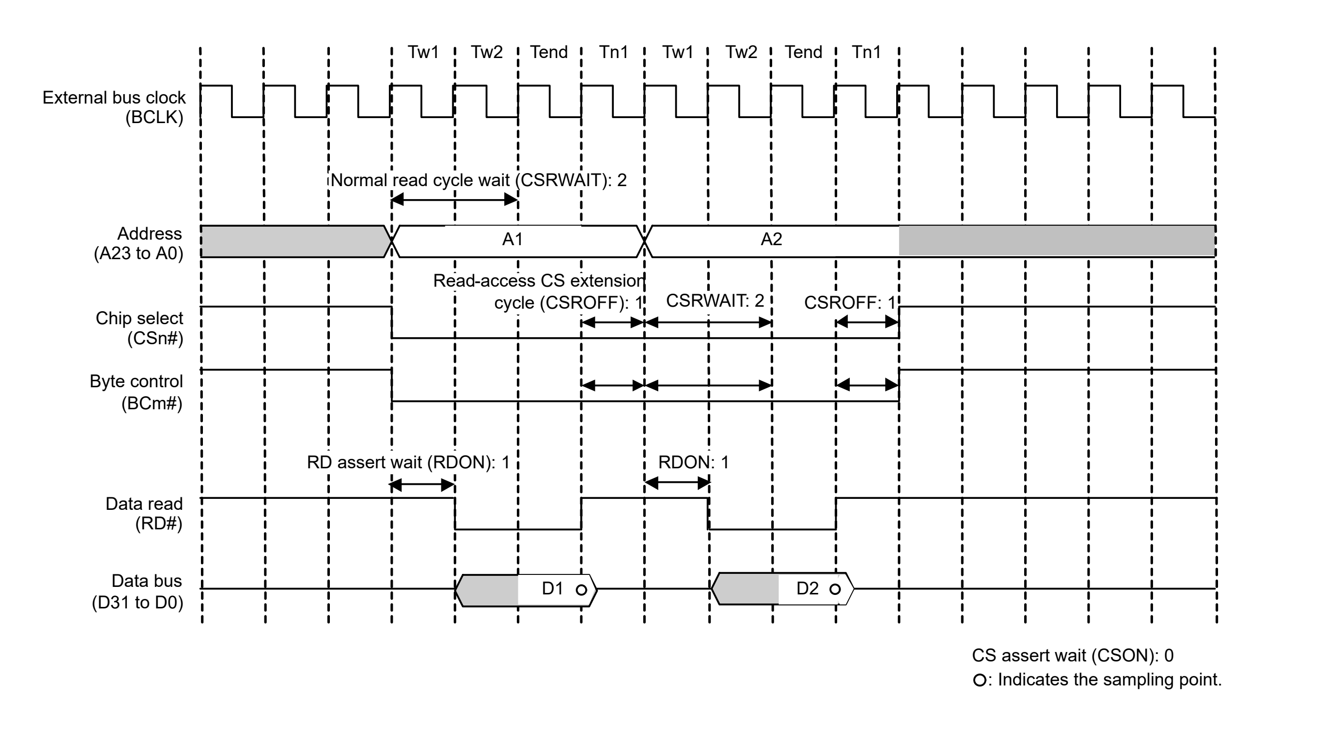


Figure 15.7 Example of normal read operation when two rounds of bus access are generated in response to a single transfer request (n = 0 to 7, m = 0 to 3)

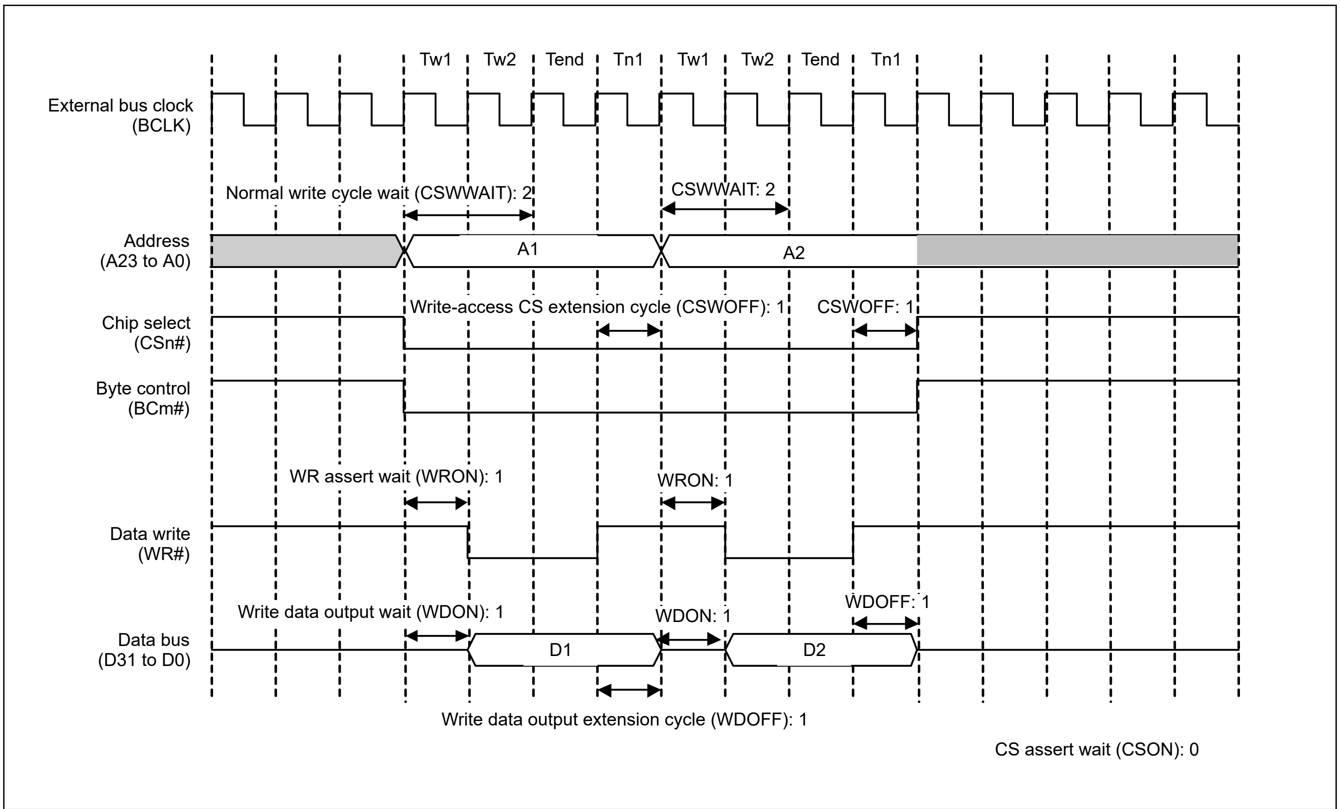


Figure 15.8 Example of normal write operation when two rounds of bus access are generated in response to a single transfer request in single-write strobe mode (n = 0 to 7, m = 0 to 3)

Figure 15.9 and Figure 15.10 show examples of normal read and write accesses to a 32-bit bus space in 16 bits.

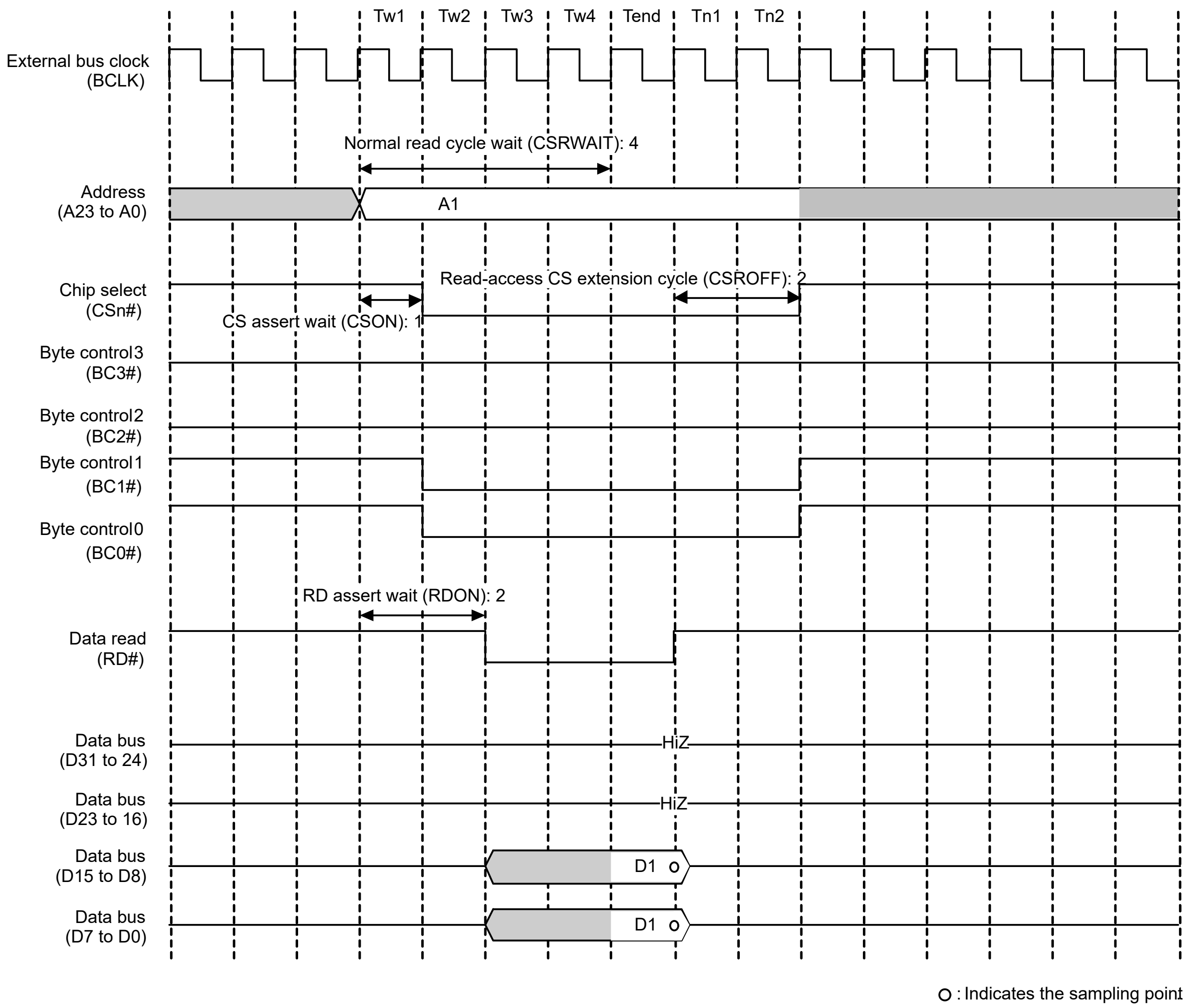


Figure 15.9 Example of normal read operation when 32-bit bus space is accessed in 16 bits (n = 0 to 7)

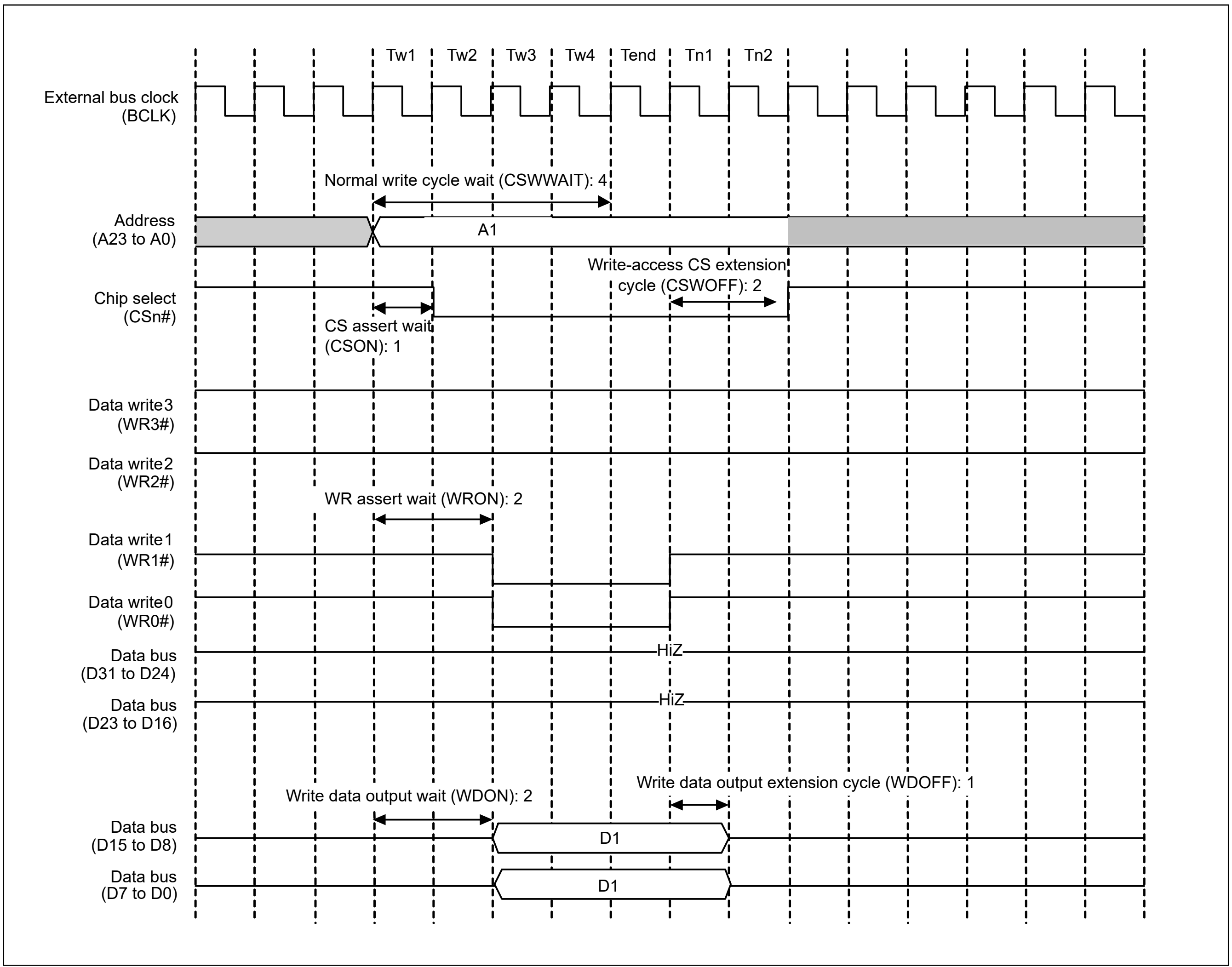


Figure 15.10 Example of normal write operation when 32-bit bus space is accessed in 16 bits, in byte strobe mode (n = 0 to 7)

Figure 15.11 to Figure 15.15 show examples of normal accesses made with the 1/2 BCLK selected with the EBCLK pin output select bit.

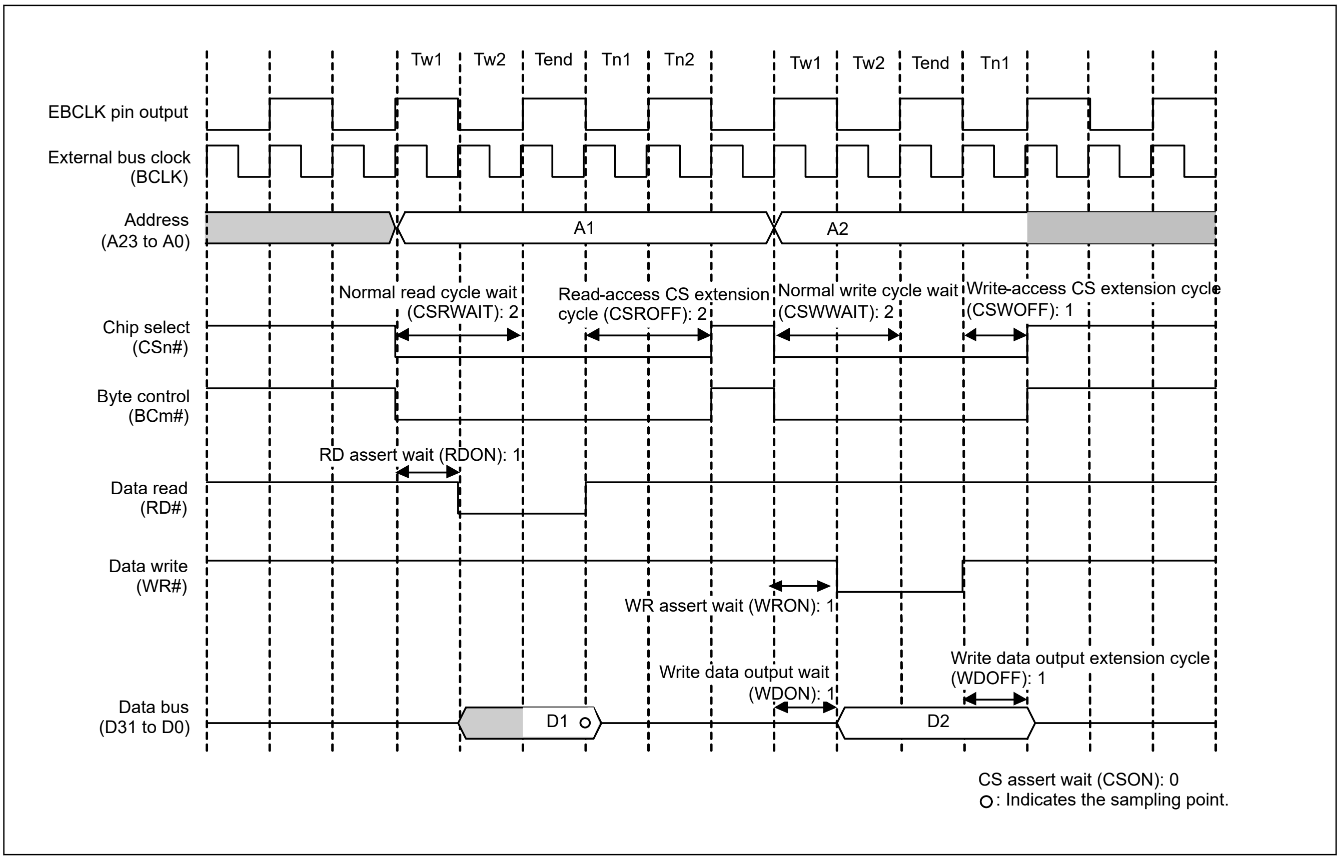


Figure 15.11 Example of normal access when BCLK/2 is selected in the EBCLK Pin Output Select bit (n = 0 to 7, m = 0 to 3)

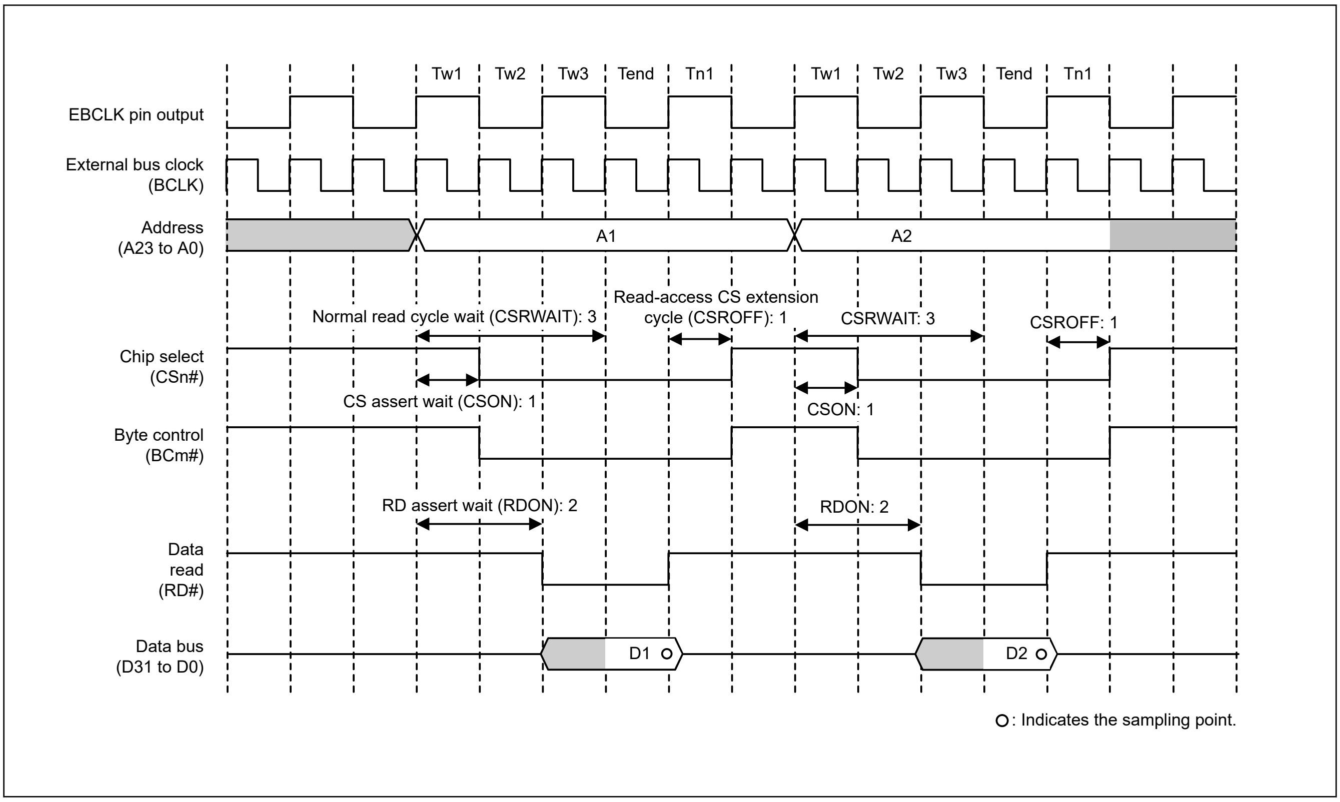


Figure 15.12 Example of normal read operation when BCLK/2 is selected in the EBCLK Pin Output Select bit (n = 0 to 7, m = 0 to 3)

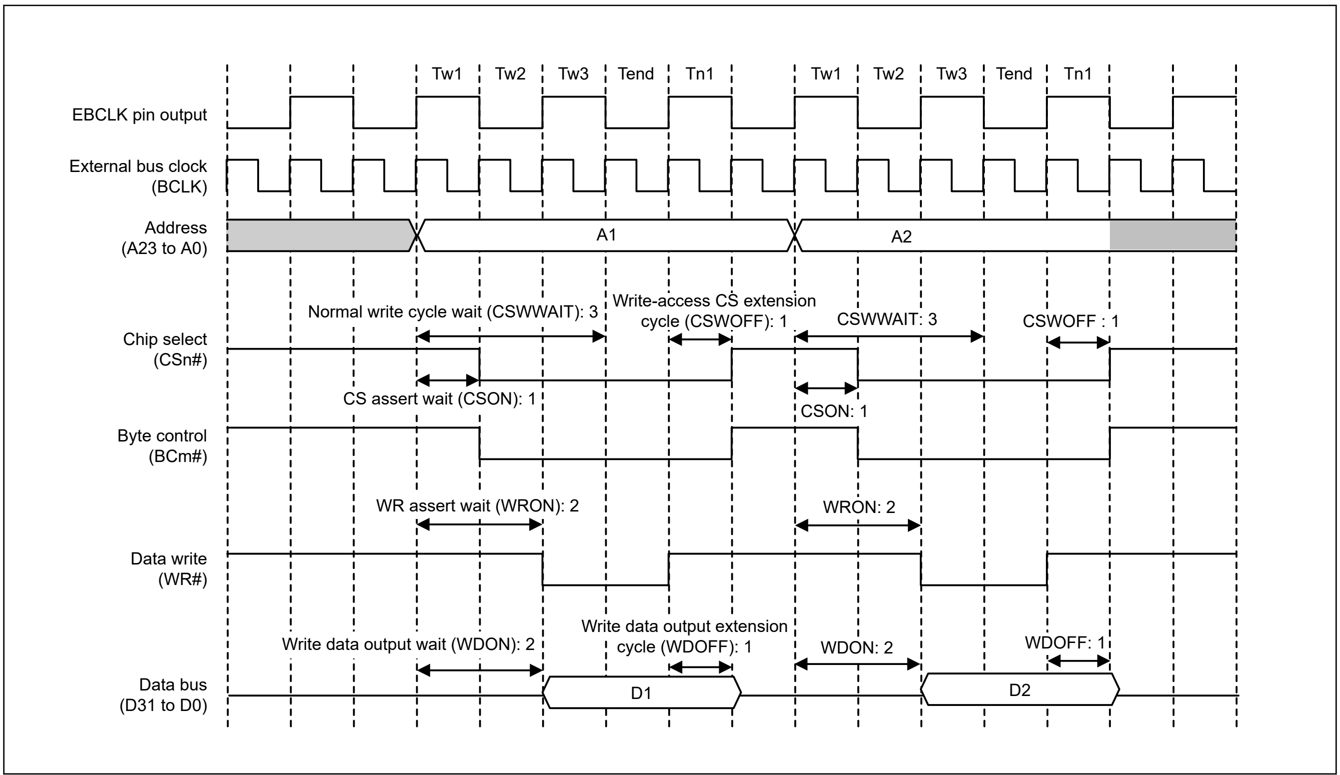


Figure 15.13 Example of normal write operation when BCLK/2 is selected in the EBCLK Pin Output Select bit (n = 0 to 7, m = 0 to 3)

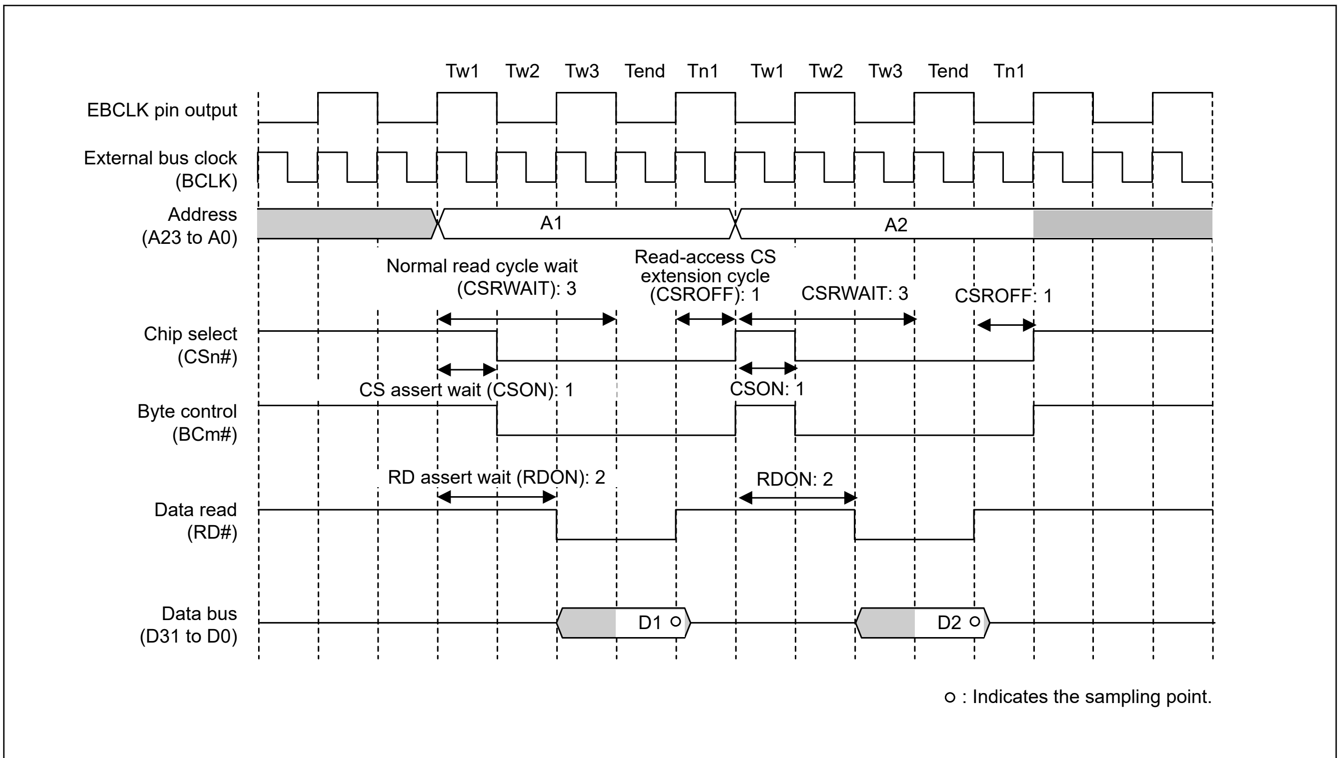


Figure 15.14 Example of normal read operation when BCLK/2 is selected in the EBCLK Pin Output Select bit and two rounds of bus access are generated in response to a single transfer request (n = 0 to 7, m = 0 to 3)

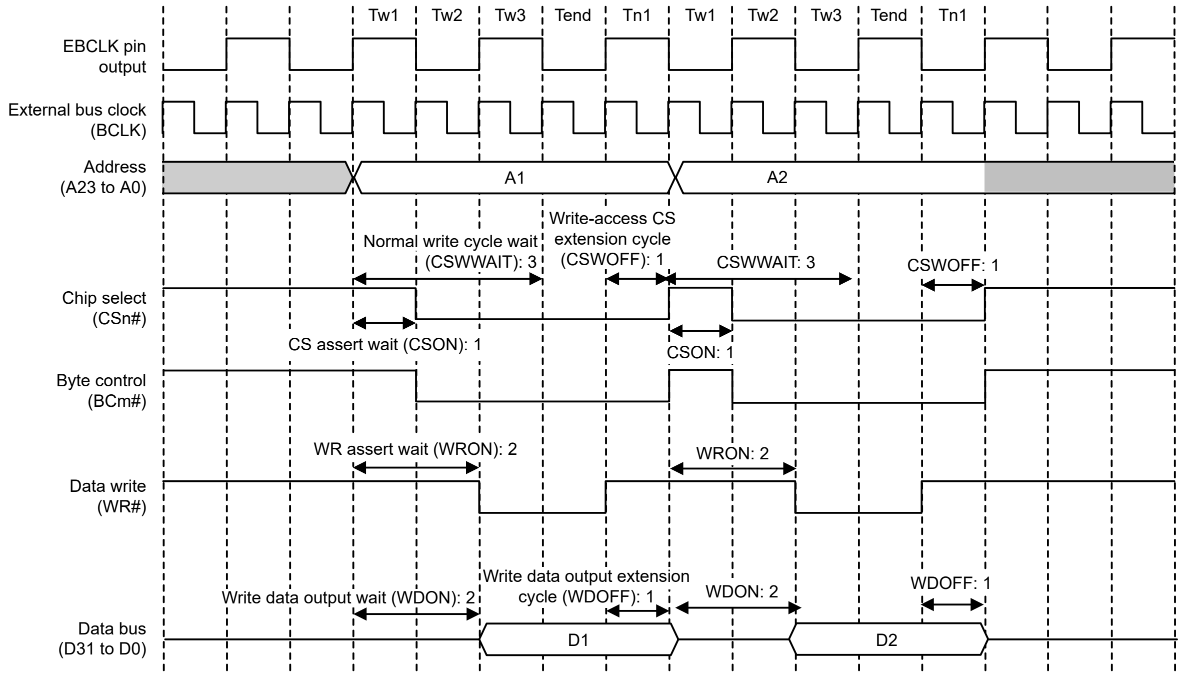


Figure 15.15 Example of normal write operation when BCLK/2 is selected in the EBCLK Pin Output Select bit and two rounds of bus access are generated in response to a single transfer request (n = 0 to 7, m = 0 to 3)

(2) Page Access

When the PRENB and PWENB bits in CSnMOD are set to 1 to enable page read and page write access, the bus access for page access operations becomes page read and write. Page access can only occur when two or more rounds of external bus access are required for a single transfer request from the bus master. However, normal access is made when split accesses are not aligned or access extends across the 32-bit boundary. See Table 15.18 to Table 15.21 for the conditions under which page access occurs.

Figure 15.16 and Figure 15.17 show examples of page access operations.

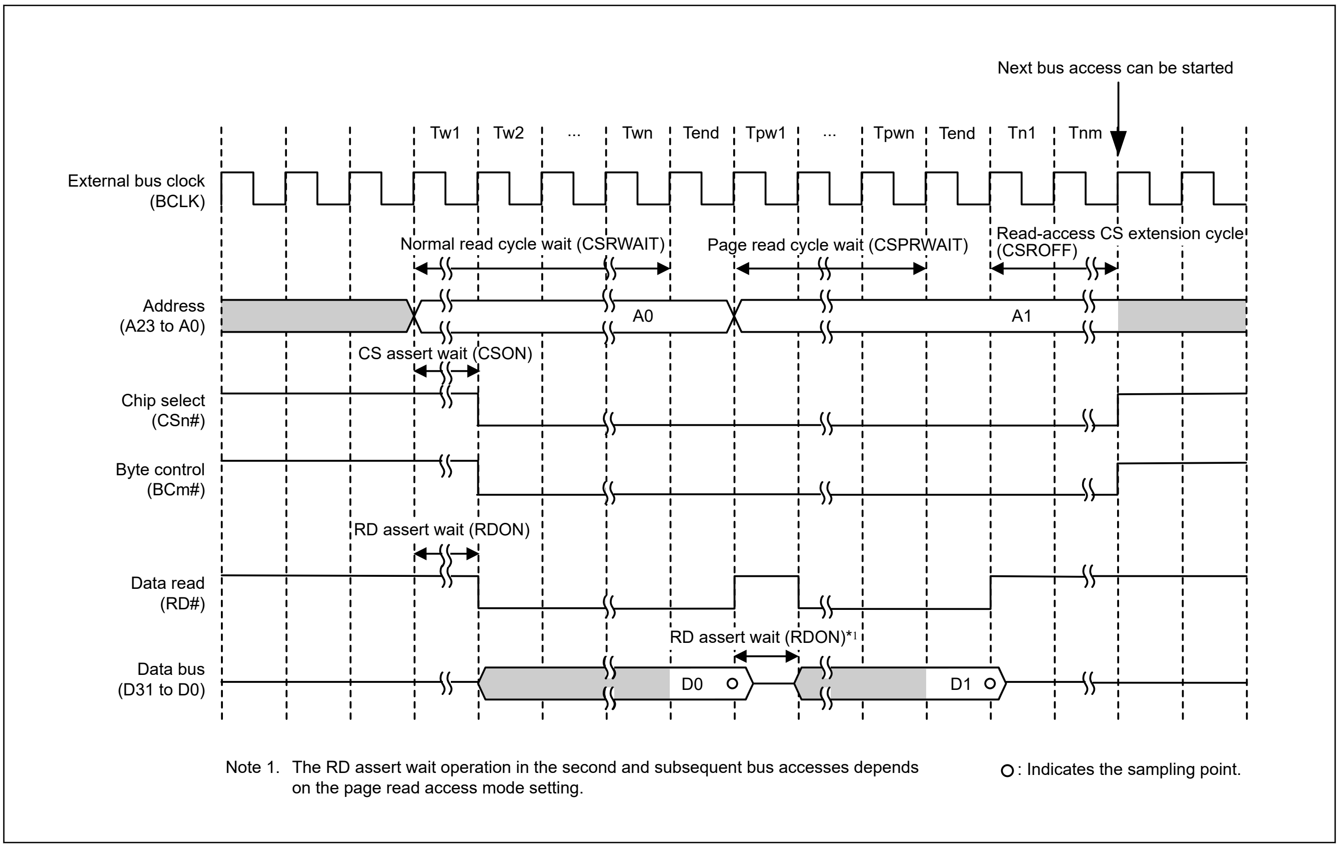


Figure 15.16 Page read access timing (n = 0 to 7, m = 0 to 3)

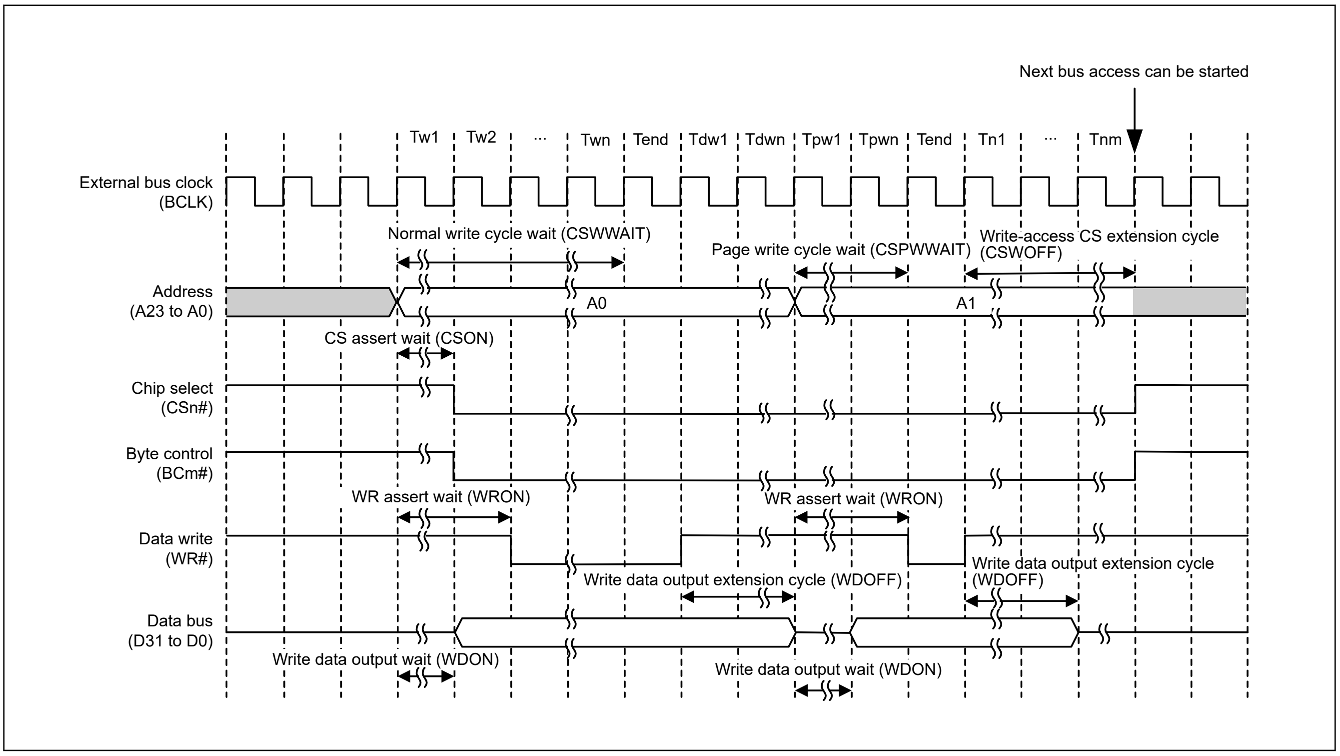


Figure 15.17 Page write access timing (n = 0 to 7, m = 0 to 3)

Figure 15.18 and Figure 15.19 show examples of operations for access to a 16-bit bus space in 32 bits. The values of the wait control registers shown in the figures are example settings. In your application, set the registers appropriately for the specifications of connected devices.

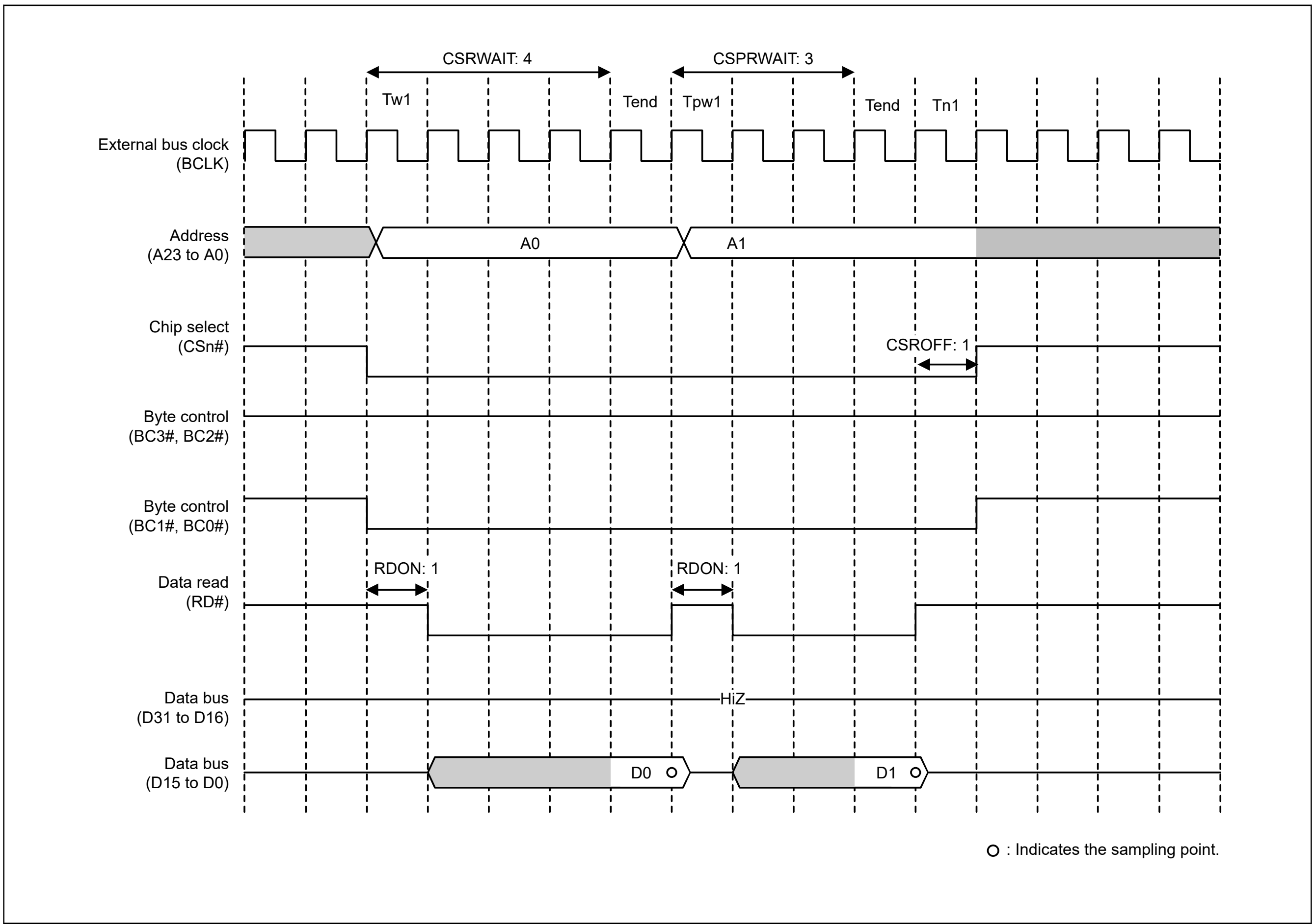


Figure 15.18 Example page read access operation when 16-bit bus space is accessed in 32 bits (n = 0 to 7)

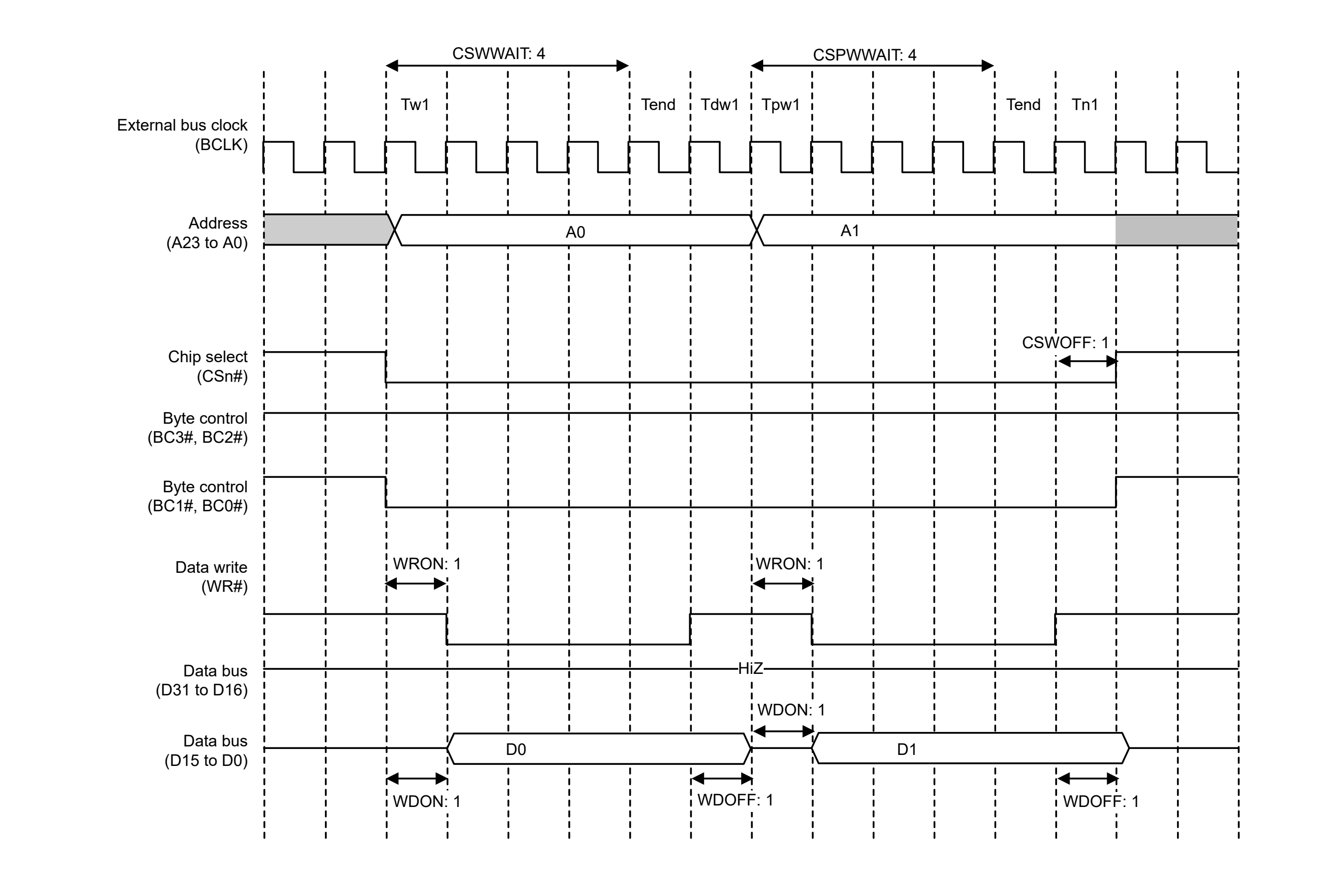


Figure 15.19 Example page write access operation when 16-bit bus space is accessed in 32 bits in single-write strobe mode (n = 0 to 7)

Figure 15.20 and Figure 15.21 show examples of page access operations when BCLK/2 is selected as the frequency division in the EBCLK Pin Output Select bit.

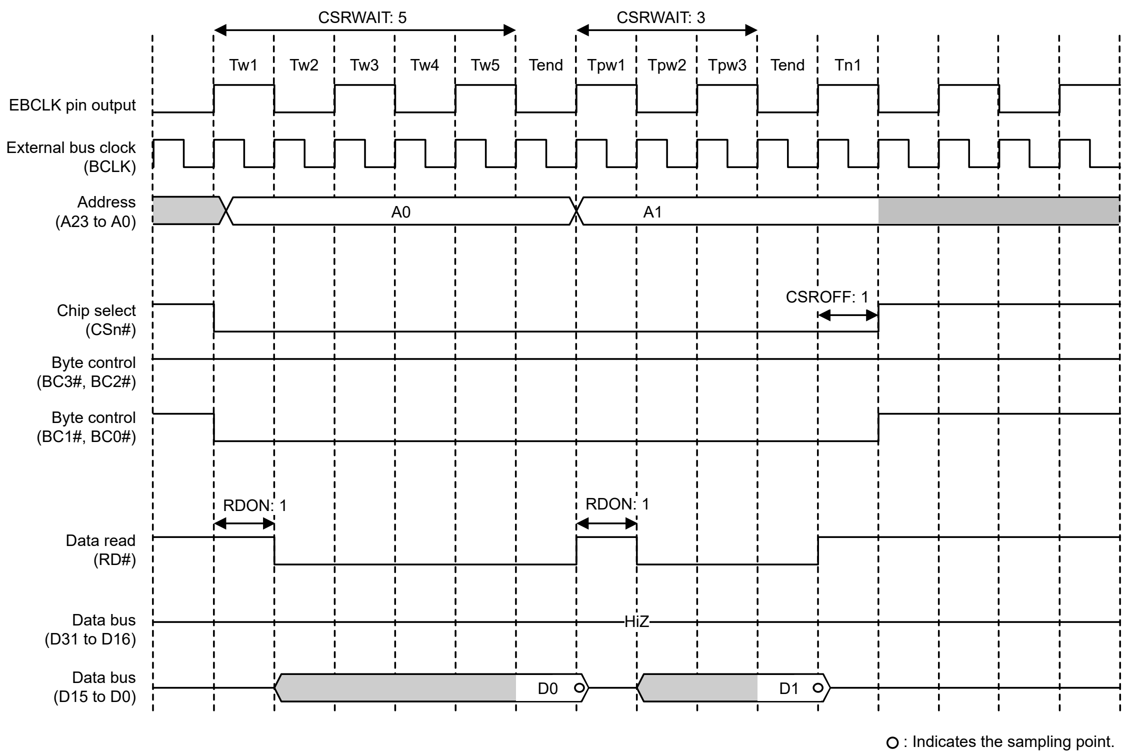


Figure 15.20 Example page read access operation when BCLK/2 is selected in the EBCLK Pin Output Select bit and two rounds of bus access are generated in response to a single transfer request (n = 0 to 7)

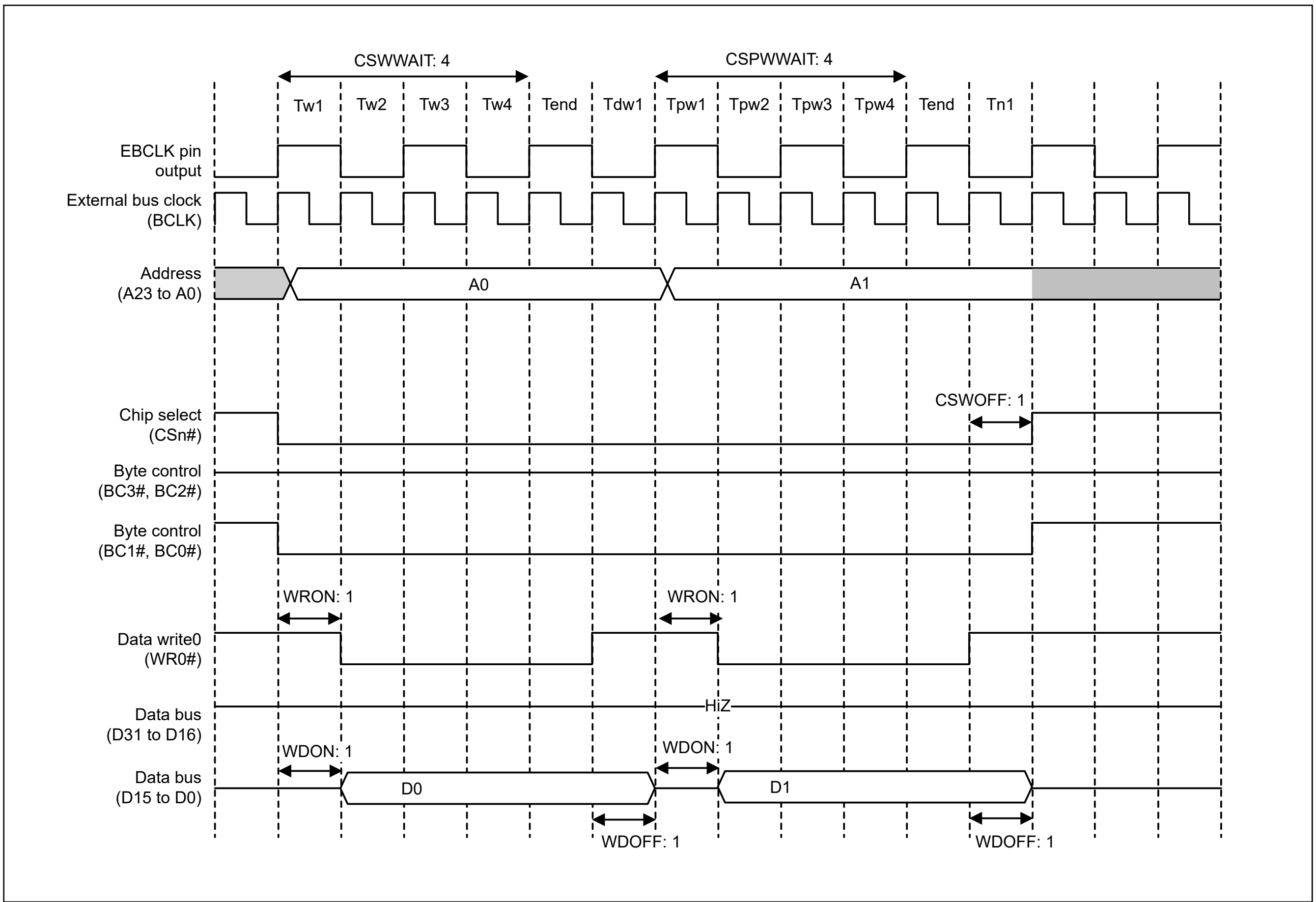


Figure 15.21 Example page write access operation when BCLK/2 is selected in the EBCLK Pin Output Select bit and two rounds of bus access are generated in response to a single transfer request, in single-write strobe mode (n = 0 to 7)

15.5.2 Address/Data Multiplexed Bus

When the address/data Multiplexed I/O Interface Select bit (MPXEN) in CSnCR is set to 1, addresses and data can be multiplexing input/output to/from the D15 to D00 pins in the corresponding area. Using this function enables direct connection of this MCU to peripherals of the MCU requiring address/data multiplexing. When 8-bit width is selected with the BSIZE[1:0] bits in CSnCR, D7 to D00 are multiplexed with A07 to A00. When 16-bit width is selected, D15 to D00 are multiplexed with A15 to A00. In the address/data multiplexed I/O space, accesses are controlled with the ALE, RD, WRn, and BCn signals.

Byte strobe mode or single-write strobe mode is selectable in the same way as for a separate bus. However, with regard to the BCn signals within the address cycle, the byte-control signal is output for the data being read or written.

During the address/data multiplexed I/O space access, after the number of wait cycles specified by the Address Cycle Wait Select bits (AWAIT[1:0]) in CSnWCR2 is inserted in the address output cycle, data access is performed.

Ta1 to Tan (Address Cycle Wait)

The period Ta1 to Tan is valid only when the address/data multiplexed I/O space is specified. This period is made up of the number of clock cycles between the start of external bus access and 1 cycle before the address latch (ALE) signal is negated. The number of cycles is selectable within the range from 0 to 3. Addresses are output until the next cycle of ALE signal negation (address cycle). The timing of ALE signal is the same as that of CS assertion. After the address cycle, a data cycle is started. CSnWCR1 and CSnWCR2 should be set so that an address cycle and a data cycle do not overlap.

Page access to the address/data multiplexed I/O space is invalid. When the PRENB or PWENB bit in CSnMOD is set to 1 to enable page-read or page-write access, these settings are ignored and normal read or write operation is performed.

Figure 15.22 to Figure 15.24 show examples of operations with the address/data multiplexed I/O interface

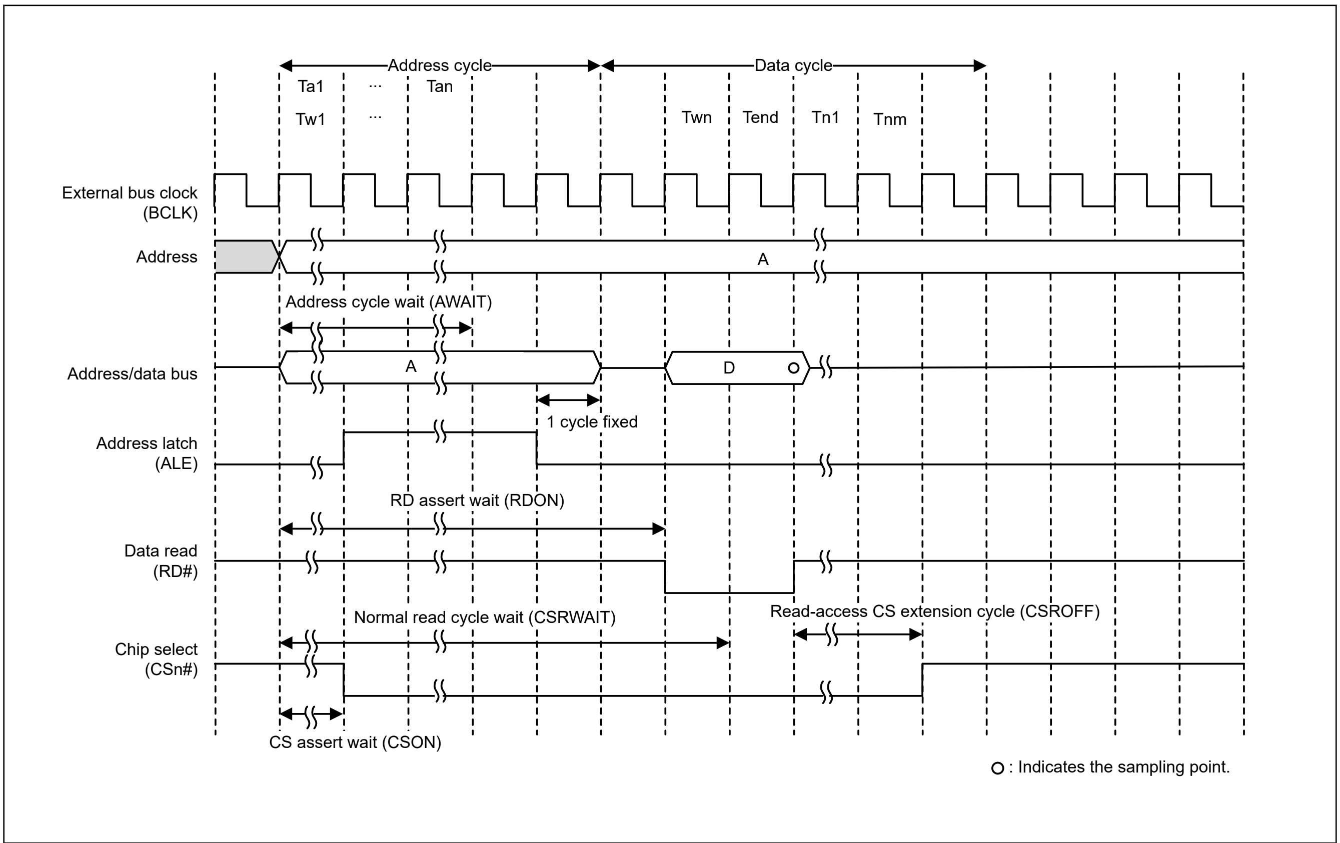


Figure 15.22 Example of read access operation with address/data multiplexed I/O interface (n = 0 to 7)

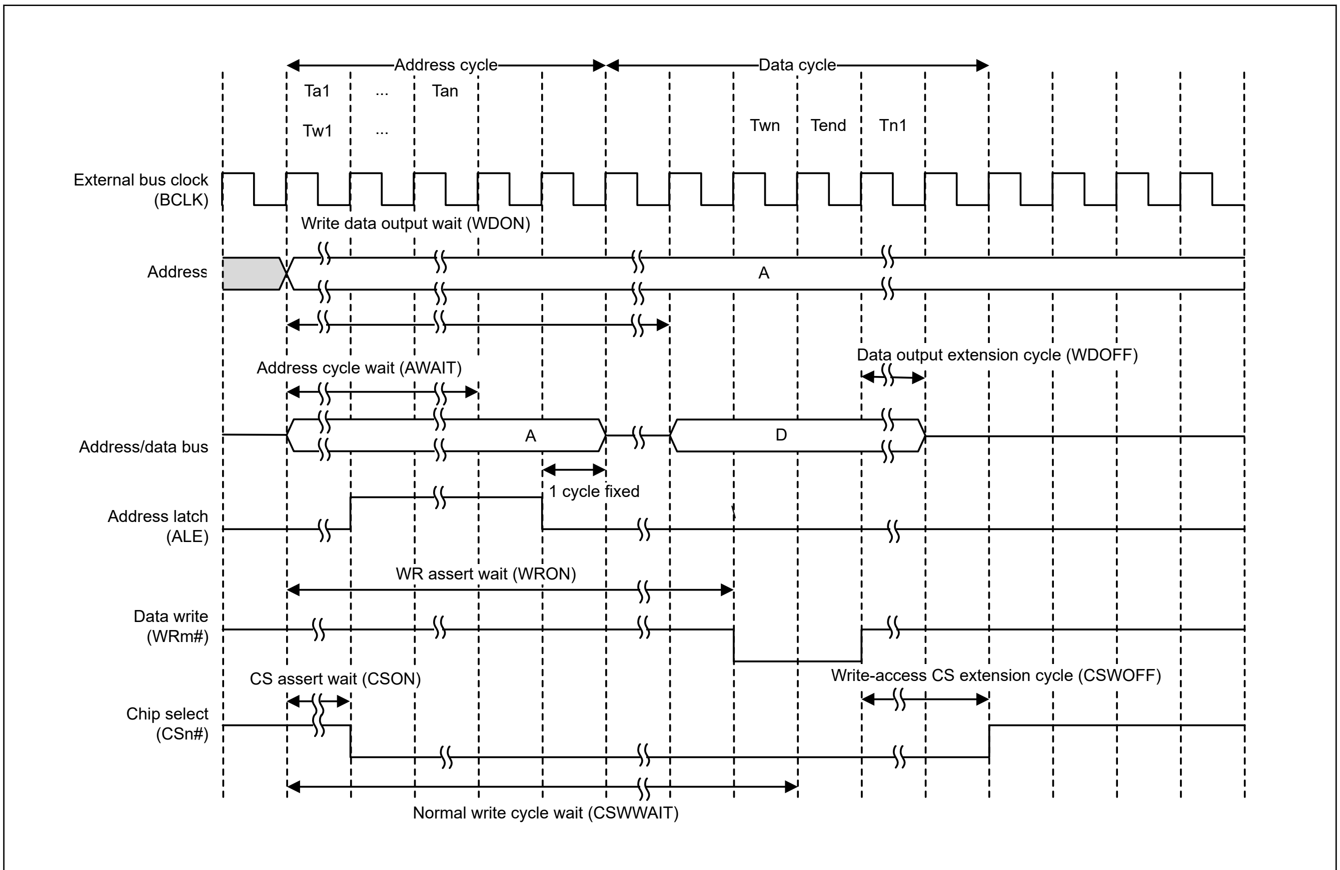


Figure 15.23 Example of write access operation with address/data multiplexed I/O interface (n = 0 to 7, m = 0, 1)

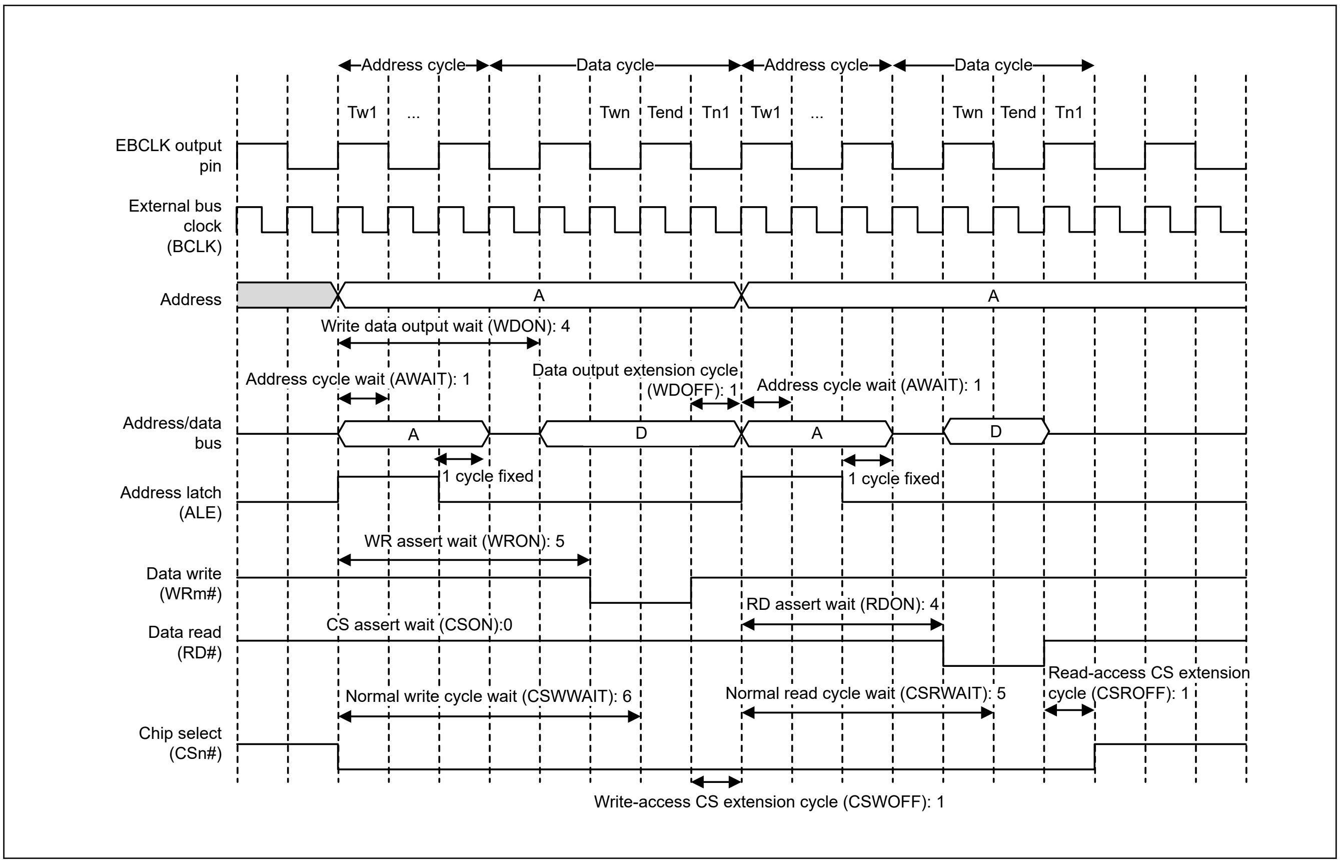


Figure 15.24 Example of bus timing with address/data multiplexed I/O interface (n = 0 to 7, m = 0, 1)

15.5.3 External Wait Function

Wait cycles can be extended by the WAIT signal beyond the length of the normal access cycle wait specified in the CSRWAIT[4:0] and CSWWAIT[4:0] bits in CSnWCR1, and the page access cycle wait specified in the CSPRWAIT[2:0] and CSPWWAIT[2:0] bits in CSnWCR1.

When external wait is enabled (EWENB = 1 in CSnMOD), wait cycles are inserted while the WAIT signal is held low. When external wait is disabled (EWENB = 0 in CSnMOD), the WAIT signal has no effect. All wait cycles specified in CSnWCR1 are inserted independently of the WAIT signal.

(1) Normal Access

Sampling of the WAIT signal begins on completion of the wait cycle (Tend) specified in CSnWCR1. The bus cycle is extended while the WAIT signal is held low. The wait cycle ends (Tend) at the next cycle after the WAIT signal goes high.

(2) Page Access

The first access operation is the same as the normal access operation. Sampling of the WAIT signal begins on completion of the wait cycle (Tend) specified in the CSnWCR1 register. The bus cycle is extended while the WAIT signal is held low. The wait cycle (Tend) ends at the next cycle after the WAIT signal goes high.

For the second and subsequent accesses, sampling of the WAIT signal begins on completion of the page access wait cycle (Tend). The page access wait cycle is extended while the WAIT signal is held low, and ends (Tend) at the next cycle after the WAIT signal goes high.

Figure 15.25 to Figure 15.28 show examples of external wait insertion timing with the separate bus interface.

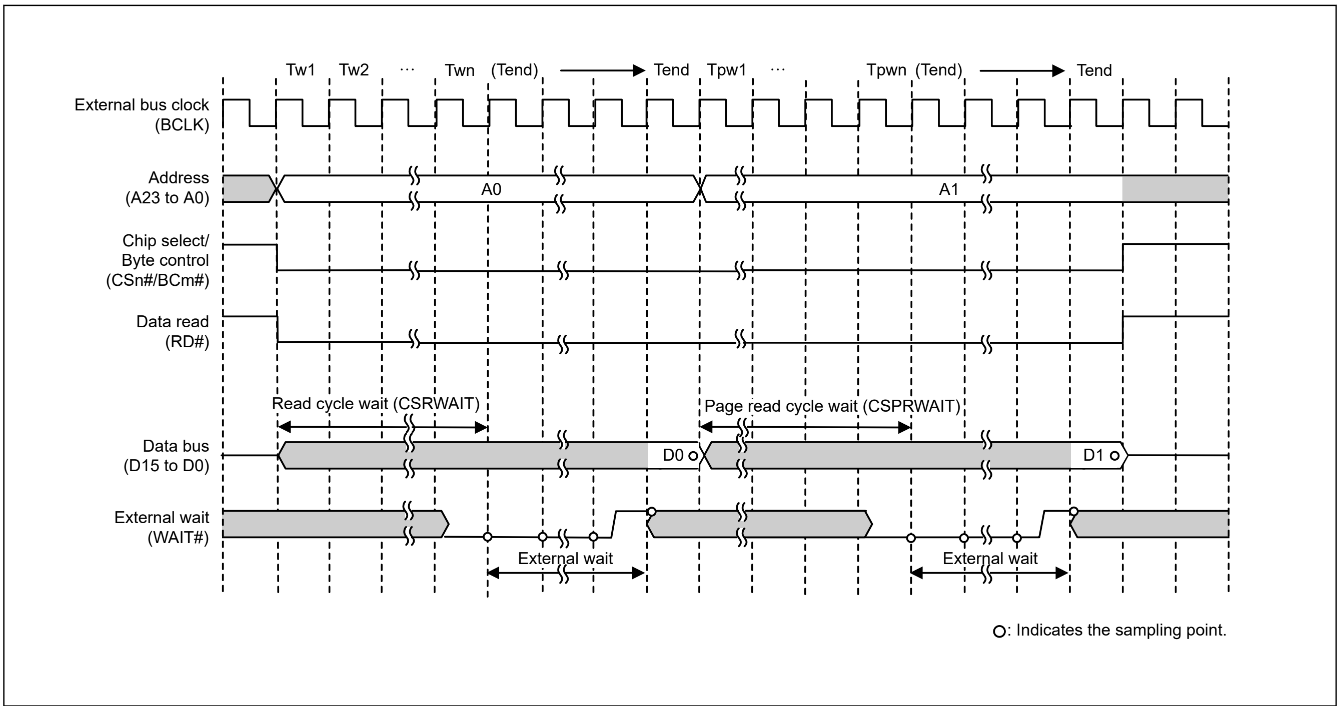


Figure 15.25 Example external wait timing for page read access to 16-bit bus space (when 1/1 BCLK is selected with the BCLK Pin Output Select bit) (n = 0 to 7, m = 0, 1)

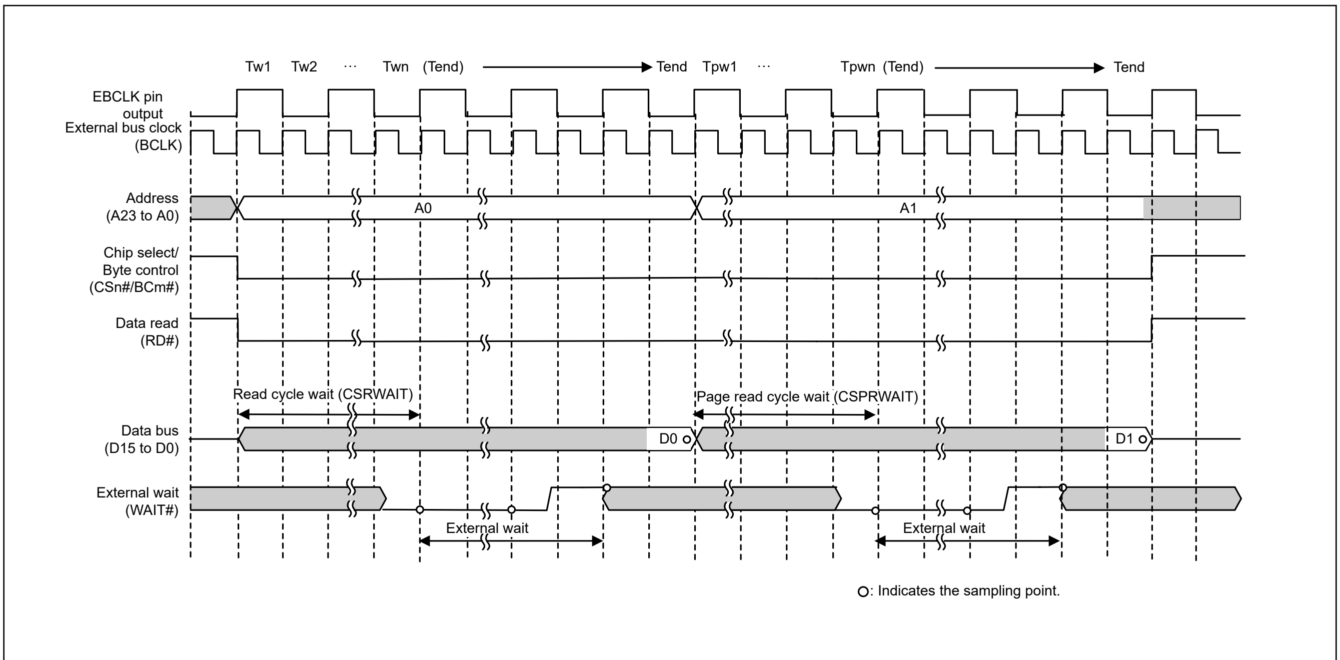


Figure 15.26 Example external wait timing for page read access to 16-bit bus space (when 1/2 BCLK is Selected with the EBCLK Pin Output Select bit) (n = 0 to 7, m = 0, 1)

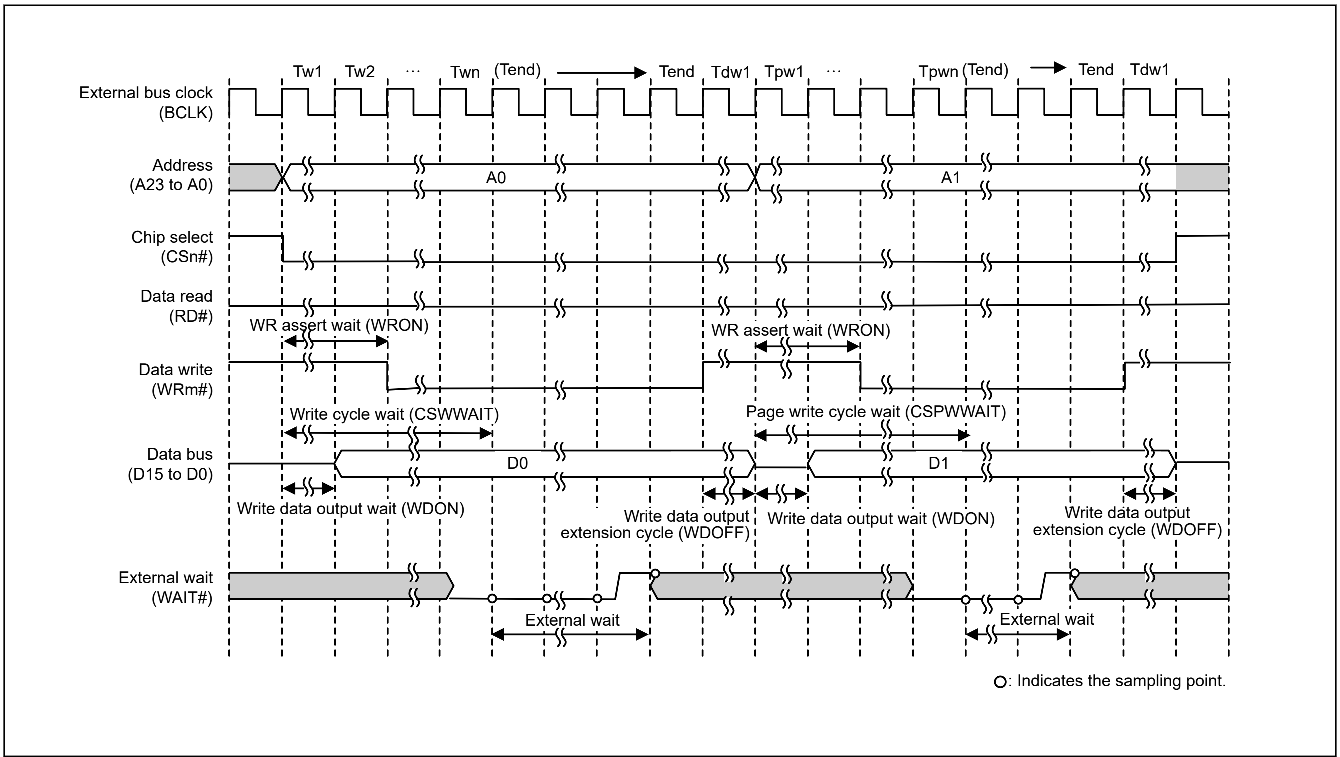


Figure 15.27 Example external wait timing for page write access to 16-bit bus space in byte strobe mode (when 1/1 BCLK is selected with the BCLK Pin Output Select bit) (n = 0 to 7, m = 0, 1)

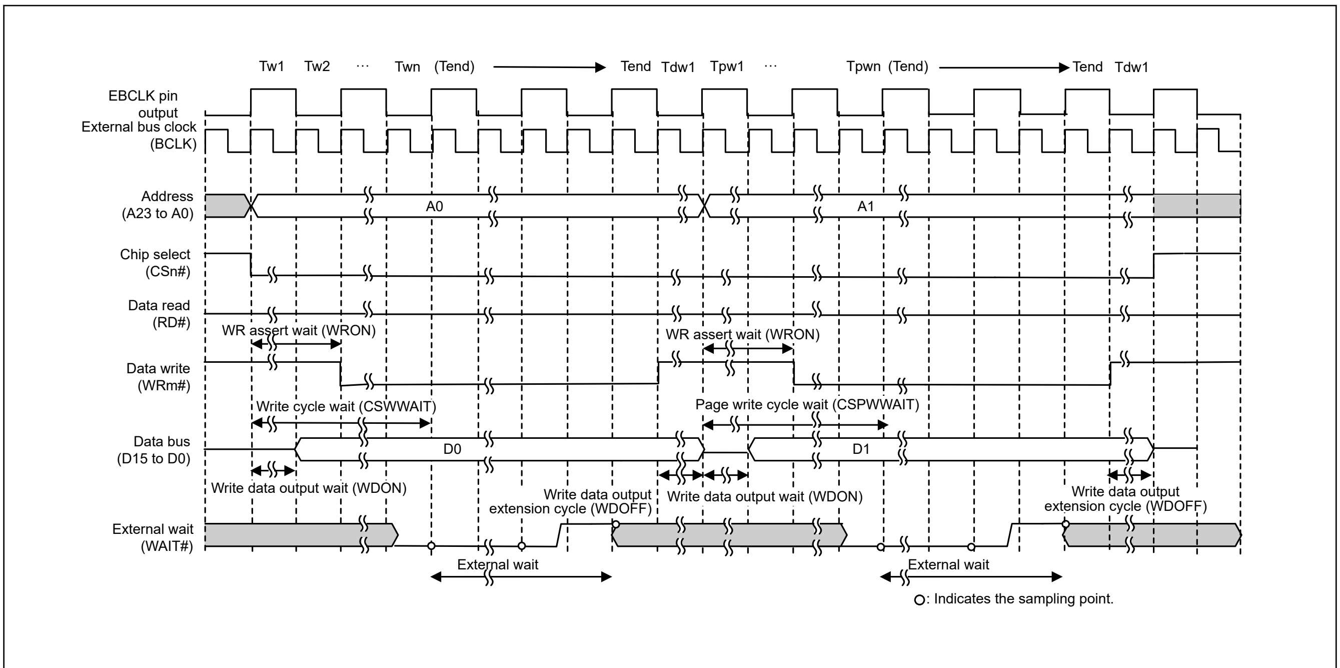


Figure 15.28 Example external wait timing for page write access to 16-bit bus space in byte strobe mode (when 1/2 BCLK is selected with the EBCLK Pin Output Select bit) (n = 0 to 7, m = 0, 1)

(3) Address/Data Multiplexed I/O Interface

In a data cycle with the address/data multiplexed I/O interface, programmed waits and pin waits using the WAIT pin can be inserted in the same way as that with the separate bus interface.

Address cycles are not affected by the wait control settings. Figure 15.29 shows an example of external wait insertion timing with the address/data multiplexed I/O interface.

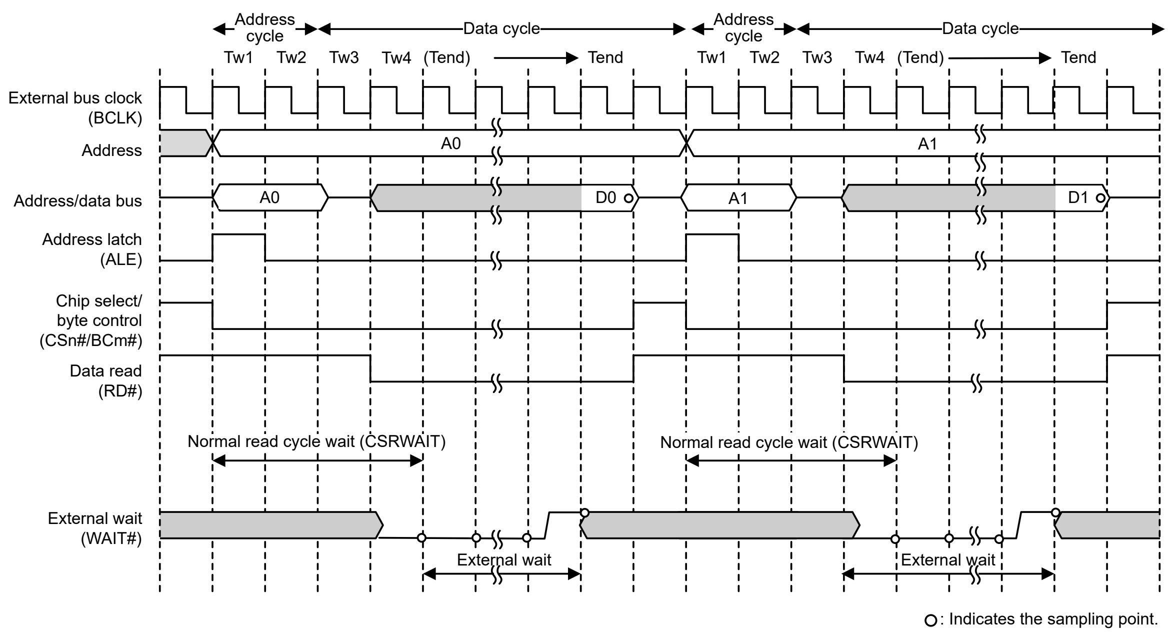


Figure 15.29 Example external wait insertion timing with address/data multiplexed I/O interface (n = 0 to 7, m = 0, 1)

15.5.4 Insertion of Recovery Cycles

Recovery cycles can be inserted between consecutive rounds of external bus access by setting the Recovery Cycle Insertion Enable bit in CSRECEN to 1.

The number of recovery cycles to be inserted after read cycles and write cycles can be independently set for each area using CSnREC. When the preceding bus cycle is a write access, the number of write recovery cycles must be set with the WRCV[3:0] bits for the associated area. When the preceding bus cycle is a read access, the number of read recovery cycles must be set with the RRCV[3:0] bits for the associated area. For example, when a CS1 read access occurs after a CS0 read access, the number of recovery cycles to be inserted between them is set in the RRCV[3:0] bits in CS0REC.

Recovery cycle insertion can be enabled or disabled with RCVENi (i = 0 to 7) in CSRECEN when the preceding bus access is a separate bus access, and with RCVENMj (j = 0 to 7) when the preceding bus access is an address/data multiplexed bus access.

Recovery cycles can be inserted on any of the following conditions:

- After a read access to the external bus, a read access is made to the external bus in the same area.
- After a read access to the external bus, a read access is made to the external bus in a different area.
- After a read access to the external bus, a write access is made to the external bus in the same area.
- After a read access to the external bus, a write access is made to the external bus in a different area.
- After a write access to the external bus, a read access is made to the external bus in the same area.
- After a write access to the external bus, a read access is made to the external bus in a different area.
- After a write access to the external bus, a write access is made to the external bus in the same area.
- After a write access to the external bus, a write access is made to the external bus in a different area.

The recovery cycle starts at the end of the preceding bus cycle, for example when the CSn signal (n = 0 to 7) is negated. A high-level period of the CSn signal is inserted for the specified recovery cycle period starting from this point.

In the fastest case, the CSn signal for the next round of bus access is asserted immediately after the end of the recovery cycles. Even if the next request for access to an external address space is generated during the recovery period, the next access over the external bus starts immediately after the end of the recovery cycles.

When two or more external bus access cycles are required for a single transfer request from a bus master, and the recovery cycle insertion condition is satisfied, recovery cycles are also inserted between these bus access cycles. However, when page read access is enabled (CSnMOD.PRENB = 1) or page write access is enabled (CSnMOD.PWENB = 1), recovery cycles are not inserted except after the last bus access cycle of the transfer, even if the recovery cycle insertion condition is satisfied. See Figure 15.33.

Similarly, during normal access with page access enabled, recovery cycles are not inserted between bus access cycles but only after the last bus access cycle of the transfer. With the address/data multiplexed I/O interface, when the recovery cycle insertion condition is satisfied, recovery cycles are inserted between bus access cycles regardless of the page access enable setting.

Figure 15.30 to Figure 15.33 show examples of recovery cycle insertion with the separate bus interface.

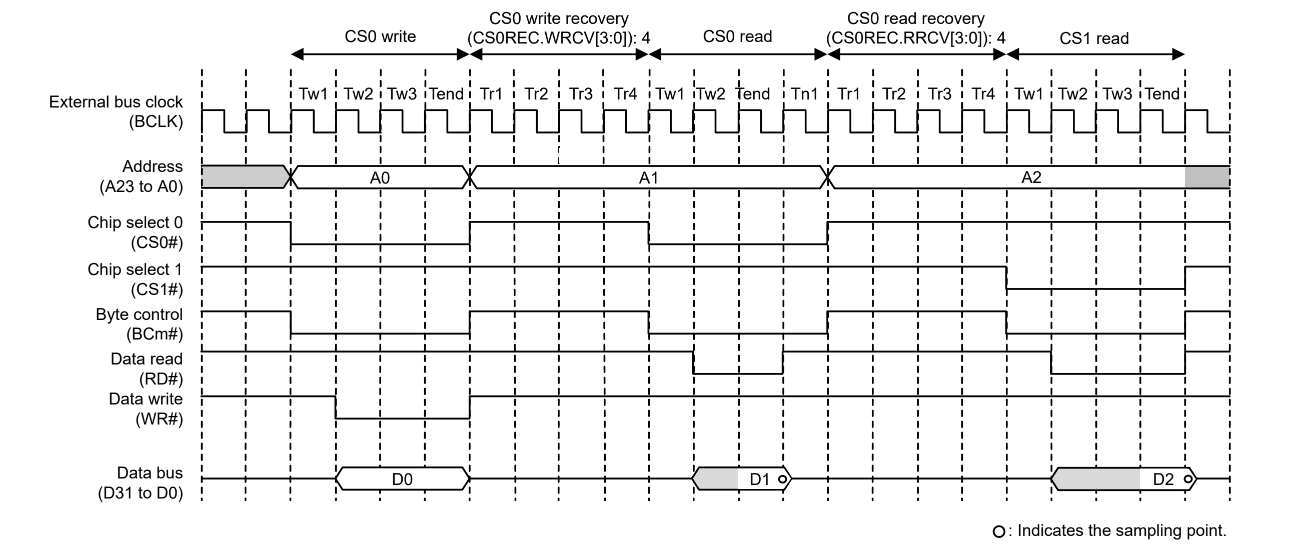


Figure 15.30 Example recovery cycle insertion with separate bus interface (m = 0 to 3)

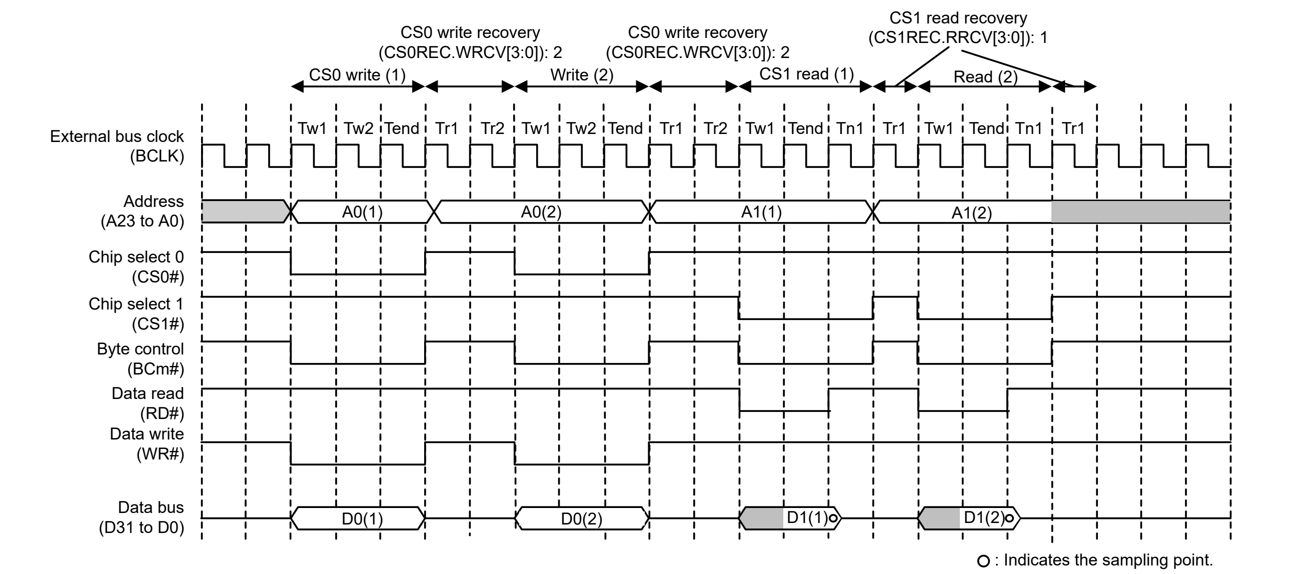


Figure 15.31 Example recovery cycle insertion when bus access is split, with separate bus interface and normal access (m = 0 to 3)

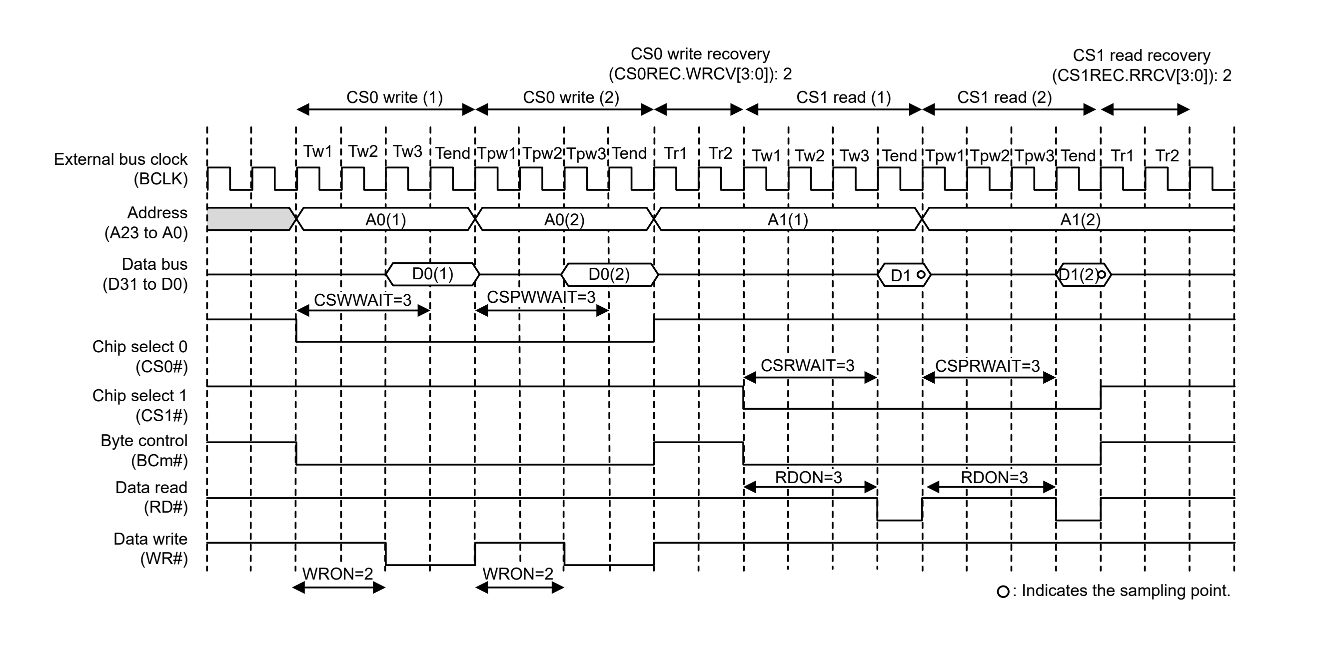


Figure 15.32 Example recovery cycle insertion when bus access is split, with separate bus interface and page access (m = 0 to 3)

Figure 15.33 shows an example operation when BCLK/2 is selected as the frequency division in the EBCLK Pin Output Select bit.

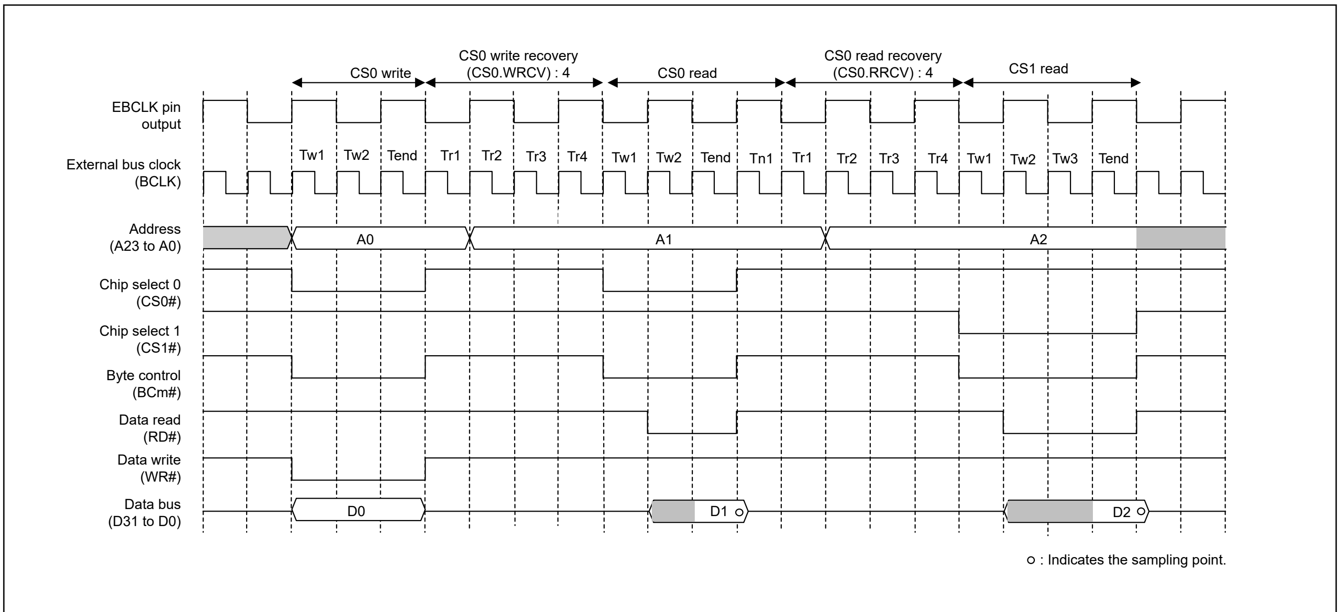


Figure 15.33 Example operation for recovery cycles when BCLK/2 is selected in the EBCLK Pin Output Select bit, with normal access through a separate bus interface (m = 0 to 3)

With the address/data multiplexed I/O interface, recovery cycles are inserted in the same way as that with the separate bus interface. Figure 15.34 and Figure 15.35 show examples of recovery cycle insertion with the address/data multiplexed I/O interface.

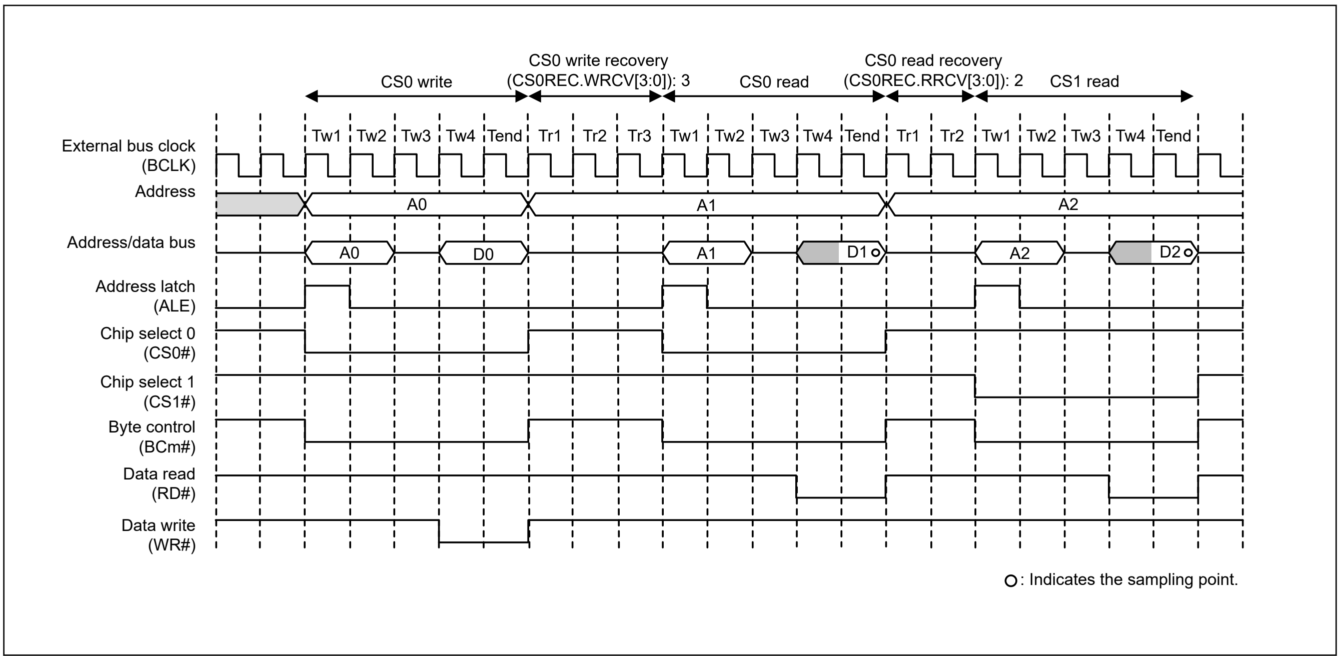


Figure 15.34 Example of recovery cycle insertion with address/data multiplexed I/O interface (m = 0, 1)

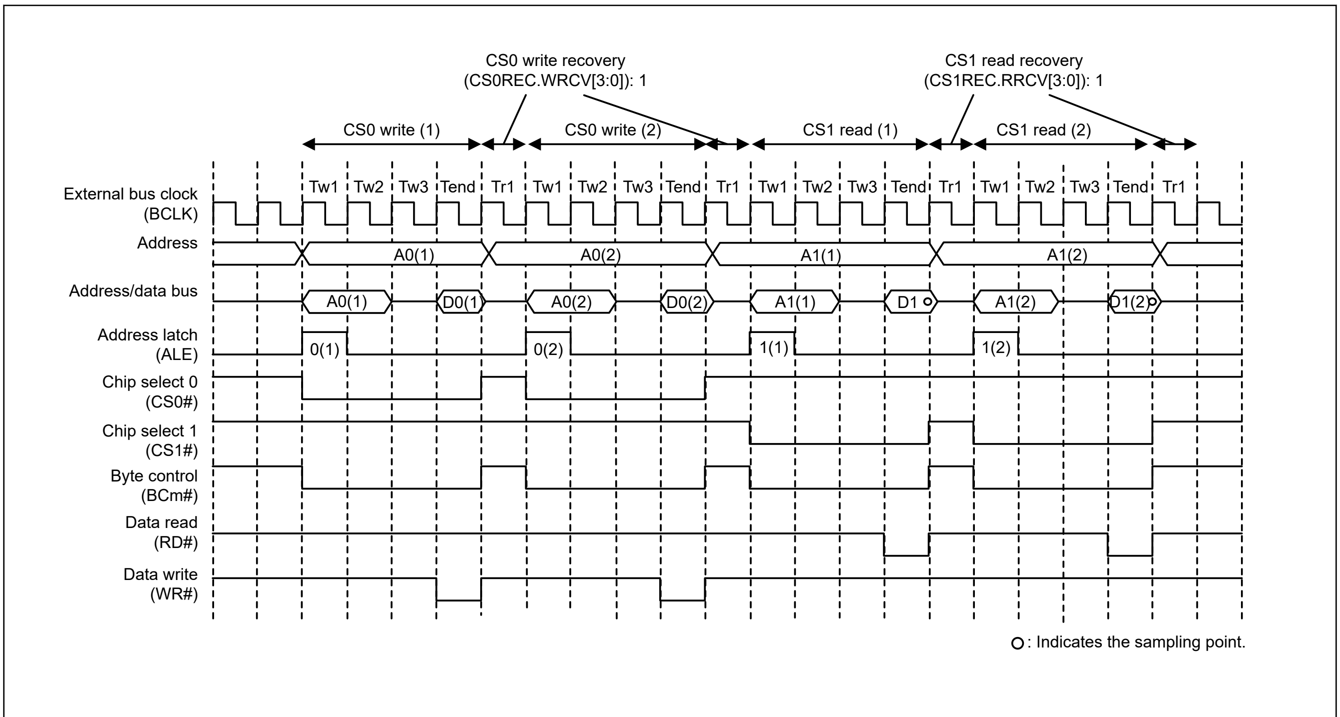


Figure 15.35 Example of recovery cycle insertion when a bus access is split with address/data multiplexed I/O interface (m = 0, 1)

15.5.5 No Access State

When no external address space is accessed, CSn, BCn, WRn, and RDn signals are high, ALE signal is low, and D31 to D00 are in the high-impedance state.

15.5.6 Limitations

(1) Constraints on using separate bus interface

Table 15.28 lists the constraints that apply to bits in the CSn Wait Control Register 1 (CSnWCR1) and CSn Wait Control Register 2 (CSnWCR2) when normal and page accesses occur.

Even if the Page Read Access Enable bit or Page Write Access Enable bit in the CSn Mode Register is set to enable (CSnMOD.PRENB = 1 or CSnMOD.PWENB = 1), the first page access or access that does not fall within the scope of a page access is a normal access operation. Because of this, constraints on normal access must be satisfied.

Table 15.28 Constraints on normal access and page access

Limitations at the time of normal access		Limitations at the time of page access	
Reading	**Writing**	**Reading**	**Writing**
CSON[2:0] ≤ CSRWAIT RDON[2:0] ≤ CSRWAIT CSON[2:0] ≤ RDON	1 ≤ WDON[2:0] CSON[2:0] ≤ CSWWAIT WRON[2:0] ≤ CSWWAIT WDON[2:0] ≤CSWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON	CSON[2:0] ≤ CSPRWAIT RDON[2:0] ≤ CSPRWAIT CSON[2:0] ≤ RDON	1 ≤ WDON[2:0] CSON[2:0] ≤ CSPWWAIT WRON[2:0] ≤ CSPWWAIT WDON[2:0] ≤ CSPWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON

Note: When 2 or more external bus access cycles are required for a single transfer request from a bus master, and the recovery cycle insertion condition is satisfied, with page read access enabled (CSnMOD.PRENB = 1) or page write access enabled (CSnMOD.PWENB = 1), recovery cycles are not inserted between bus access cycles and are inserted only after the last bus access cycle of the transfer.

(2) Constraints on using address/data multiplexed bus interface

In the address/data multiplexed I/O space, page accesses are invalid. If a page access setting is specified, the setting is ignored and the normal read or write operation is performed. When the address/data multiplexed I/O interface is set, the BSIZE[1:0] bits in CSnCR should not be set to the 32-bit bus space. If set, the operation cannot be guaranteed.

Table 15.29 Constraints at the time of normal access

Reading	Writing
CSON[2:0] ≤ CSRWAIT RDON[2:0] ≤ CSRWAIT CSON[2:0] ≤ RDON AWAIT[1:0] + 2 ≤ RDON CSON[2:0] ≤ AWAIT	CSON[2:0] ≤ CSWWAIT WRON[2:0] ≤ CSWWAIT WDON[2:0] ≤ CSWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON AWAIT[1:0] + 2 ≤ WRON AWAIT[1:0] + 2 ≤ WDON CSON[2:0] ≤ AWAIT

(3) Constraint on pin multiplexing between the A00 and BC0 functions

Setting the single-write strobe mode is prohibited in the 8-bit bus space.

(4) Constraints when BCLK/2 is selected in the EBCLK Pin Output Select bit

When 1/2 cycle of BCLK is selected in the EBCLK Pin Output Select bit, the external bus access cycle starts on the rising edge of the EBCLK pin output. However, when 2 or more external bus access cycles are generated for a single transfer request from a bus master, the second or subsequent external bus access cycle can start on the falling edge of the EBCLK pin output, depending on the wait cycle settings. Set the registers appropriately for the specifications of connected devices.

(5) Instruction code constraints

You must fix the instruction code to little-endian order.

15.6 SDRAM Area Controller Operation

This section describes how the SDRAM area controller (SDRAMC) is enabled and the SDRAM bus width is set, followed by a description of the SDRAMC operations, including read, write, auto-refresh, self-refresh, initialization sequence, and mode register settings.

15.6.1 Enabling/Disabling SDRAM Access and Setting the SDRAM Bus Width

SDRAM access can be enabled or disabled using the SDC Control Register (SDCCR). The SDRAM bus width can also be set using SDCCR. The refresh operation is available even when the operation of the SDRAM address space is disabled, as long as self-refresh or auto-refresh is enabled.

15.6.2 No Access State

When no external address space is accessed, the SDCS, WE, RAS and CAS signals are high.

15.6.3 Insertion of Recovery Cycles

When access to the SDRAM area follows access to the CS area, data recovery cycles are inserted for the CS area controller (CSC). If the number of recovery cycles for the CSC is 0, the ACT command for the next SDRAM access is issued immediately after negation of CSn signal at the earliest. If the number of recovery cycles are not 0, the ACT command is issued 2 cycles after the specified recovery cycle period elapsed after negation of CSn signal at the earliest. Because no data conflicts can occur during access to the SDRAM area, there is no need to set data recovery cycles for the SDRAM (fixed to 0 cycle).

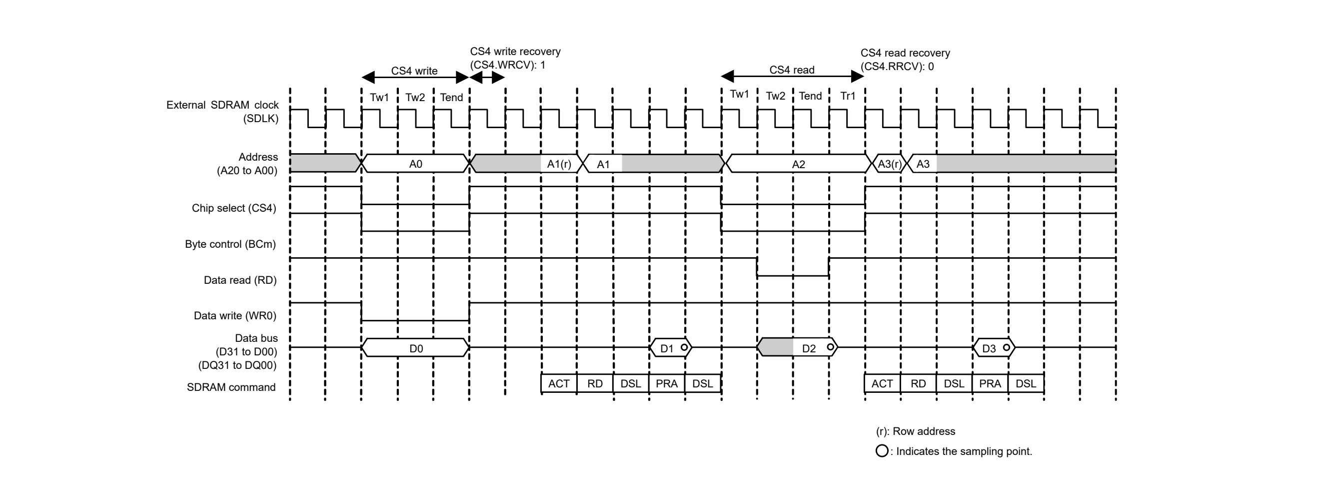


Figure 15.36 Example of recovery timing for SDRAM access

15.6.4 SDRAM Commands

To control the SDRAM, the SDRAMC issues a command for each bus cycle. Commands are defined by a combination of the SDCS, RAS, CAS, WE, CKE, and other signals. Table 15.30 lists the commands issued by the SDRAMC.

Table 15.30 SDRAMC commands

Name	Abbreviation	Command	SDCS	RAS	CAS	WE	CKE		BA1	BA0
							n-1	n		
DESL	DSL	Device deselect	H	x	x	x	H	x	x	x
ACTV	ACT	Bank active	L	L	H	H	H	x	V	V
READ	RD	Read	L	H	L	H	H	x	V	V
WRIT	WRI	Write	L	H	L	L	H	x	V	V
PRE	PRE	Precharge	L	L	H	L	H	x	V	V
PALL	PRA	All bank precharge	L	L	H	L	H	x	x	x
REF	RFA	Auto-refresh	L	L	L	H	H	x	x	x
MRS	MRS	Mode register set	L	L	L	L	H	x	L	L
SELF	RFS	Self-refresh entry	L	L	L	H	H	L	x	x
SELFX	RFX	Self-refresh end	H	x	x	x	L	H	x	x

Note: H = High level, L = Low level, V = Valid, x = Don't care.
n = Command issue cycle, n – 1 = 1 cycle before the command is issued.

15.6.5 Conditions for Setting the SDRAMC Registers

The SDRAMC registers must only be modified when all the conditions shown in Table 15.31 are satisfied.

Table 15.31 Conditions for register modification

Function or operation	Registers	Conditions
Self-refresh	SDSELF*1	• SDRAM access is disabled (SDCCR.EXENB = 0*2) • Auto-refresh operation is enabled (SDRFEN.RFEN = 1)
Auto-refresh	SDRFCR	Self-refresh operation is disabled (SDSELF.SFEN = 0)
	SDRFEN	• SDRAM access is disabled (SDCCR.EXENB = 0*2) • Self-refresh operation is disabled (SDSELF.SFEN = 0)
Initialization sequence	SDIR*1	SDICR is not set yet, and the same conditions as for SDICR modification are satisfied
	SDICR*1	• Auto-refresh operation is disabled (SDRFEN.RFEN = 0) • Self-refresh operation is disabled (SDSELF.SFEN = 0)
Address register	SDADR	• SDRAM access is disabled (SDCCR.EXENB = 0*2) • Auto-refresh operation is disabled (SDRFEN.RFEN = 0) • Self-refresh operation is disabled (SDSELF.SFEN = 0)
Timing register	SDTR	• Self-refresh operation is in progress (SDSELF.SFEN = 1) or • SDRAM access is disabled (SDCCR.EXENB = 0*2) • Auto-refresh operation is disabled (SDRFEN.RFEN = 0) • Self-refresh operation is disabled (SDSELF.SFEN = 0)
Mode register	SDMOD*1	• SDRAM access is disabled (SDCCR.EXENB = 0*2) • Self-refresh operation is disabled (SDSELF.SFEN = 0)
Access Mode register	SDAMOD	• SDRAM access is disabled (SDCCR.EXENB = 0*2) • Auto-refresh operation is disabled (SDRFEN.RFEN = 0) • Self-refresh operation is disabled (SDSELF.SFEN = 0)

Note 1. Before modifying this register, confirm that all the status bits in SDSR are 0.
Note 2. After writing 0 to the EXENB bit, confirm that it is cleared to 0.

15.6.6 Self-Refresh

Transition to or recovery from self-refresh mode is controlled with the SDRAM Self-Refresh Control Register (SDSELF). Immediately before the transition to self-refresh mode, an auto-refresh operation is performed. In self-refresh mode, the CKE signal is low. Immediately after recovery from self-refresh mode, the auto-refresh cycle starts.

Figure 15.37 and Figure 15.38 show timing examples of the transition to and recovery from self-refresh mode.

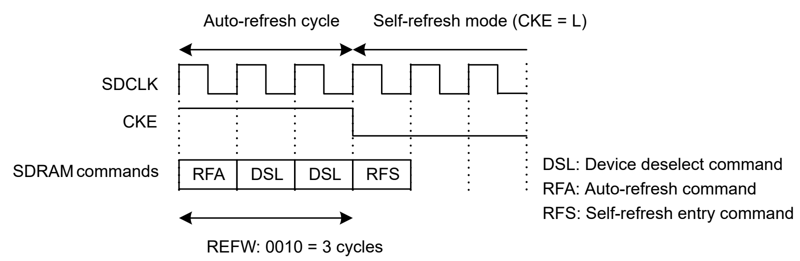


Figure 15.37 Example timing for transition to self-refresh mode when SDRFCR.REFW[3:0] = 0010 (3 cycles)

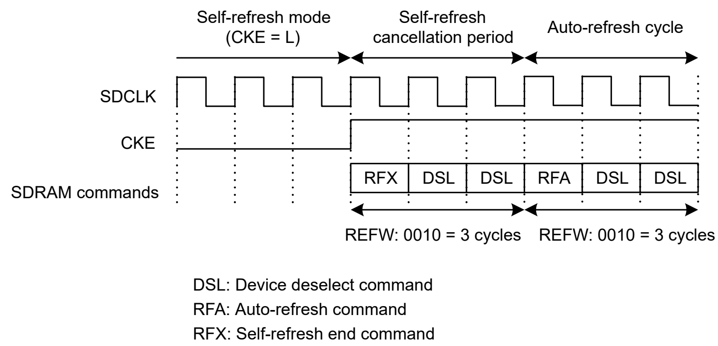


Figure 15.38 Example timing for recovery from self-refresh mode

(1) Self-refresh in Software Standby mode

When invoking self-refresh in Software Standby mode, first follow the procedure shown in section 15.6.11.2. Procedure for Transitioning to and Recovering from Self-refresh Mode. Next, set up the transition to Software Standby mode. In this mode, set the Output Port Enable bit (OPE) in the Standby Control Register (SBYCR) to 1 to hold the output state of the address bus and bus control signals.

After canceling Software Standby mode, follow the procedure shown in section 15.6.11.2. Procedure for Transitioning to and Recovering from Self-refresh Mode. For details on invoking and canceling Software Standby mode, see section 11, Low Power Mode.

(2) Self-refresh in Deep Software Standby mode

Deep Software Standby mode is invoked from within Software Standby mode. On this transition, the pin states remain unchanged. Therefore, invoking of self-refresh in Deep Software Standby mode can be handled the same as for Software Standby mode with one additional setting. You must also set the I/O Port Keep bit (IOKEEP) in the Deep Software Standby mode Control Register (DPSBYCR) to 1.

Because the SDRAMC is reset internally when Deep Software Standby mode is canceled, the SDRAM control registers must be set again. After canceling Software Standby mode, follow the procedure in this section to cancel self-refresh.

Figure 15.39 shows self-refresh timing in Deep Software Standby mode. For details on invoking and canceling Deep Software Standby mode, see section 11, Low Power Mode.

To cancel self-refresh mode:

1. Set DPSBYCR.IOKEEP to 1 to keep the CKE signal output low in Deep Software Standby mode.
2. Start the clock supply to the SDRAMC.
3. Set the SDRAM control registers (SDCMOD, SDADR, and SDTR) again. These registers were initialized by an internal reset on entering Deep Software Standby mode.
4. Enable an auto-refresh operation by setting SDRFEN.RFEN to 1.
5. Check that all the status bits in SDSR are cleared to 0 and set SDSELF.SFEN to 1 to select self-refresh mode again.
6. Modify the port settings for the SDRAM interface.
7. Set SDCKOCR.SDCKOEN to 1 to start the clock supply to the SDRAM with the SDCLK pin.
8. Check that all the status bits in SDSR are cleared to 0 and set SDSELF.SFEN to 0 to cancel self-refresh mode.

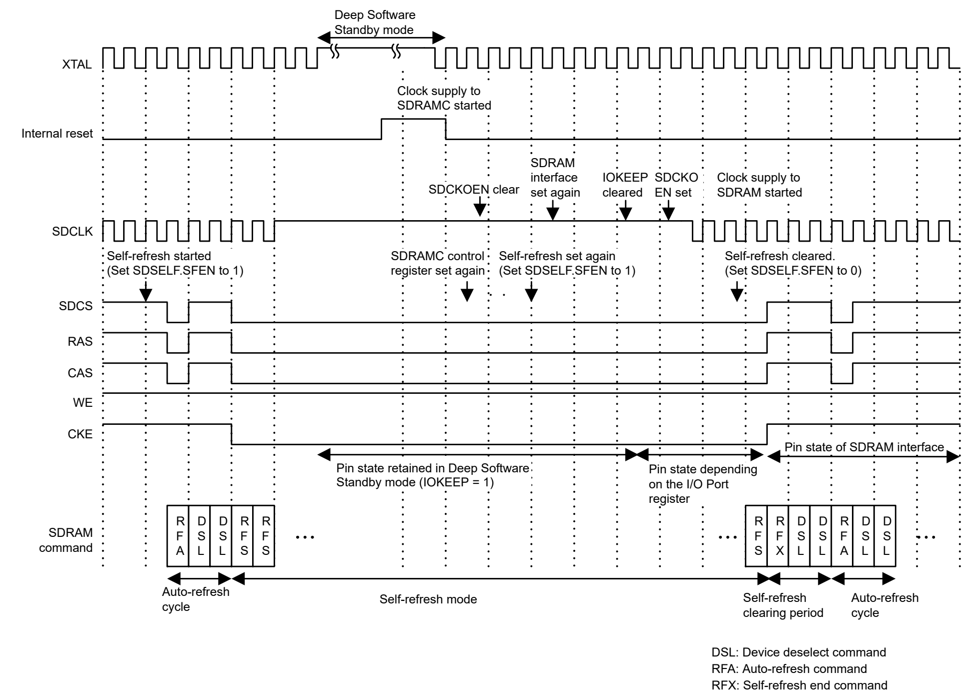


Figure 15.39 Example timing for self-refresh cycle in Deep Software Standby mode

15.6.7 Auto-Refresh

The auto-refresh cycle can be started by setting the Auto-Refresh Operation Enable bit (RFEN) in the SDRAM Auto-Refresh Control Register (SDRFEN) to 1. After the cycle starts, refresh requests are generated at fixed intervals determined by the refresh counter. However, because refresh requests are not accepted during read or write access, the auto-refresh cycle might be suspended.

If an SDRAM access and a refresh request are generated at the same time, the refresh request takes precedence. A CS area access and a refresh request can be made at the same if the SDCS, RAS, CAS, WE, and CKE signals, which are required for issuing the refresh command, are exclusively provided for SDRAM access.

The refresh counter is halted during a self-refresh operation. After recovery from the self-refresh mode, the auto-refresh cycle starts and the counter value is reset, resuming the counter operation.

Figure 15.40 shows a timing example of an auto-refresh cycle.

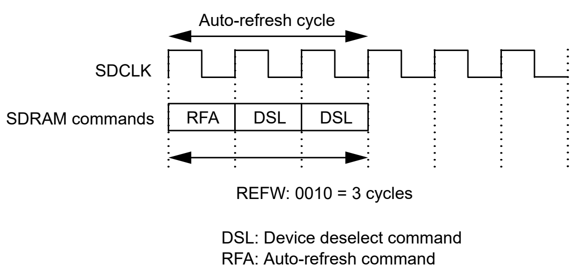


Figure 15.40 Example timing for auto-refresh cycle (1)

Figure 15.41 and Figure 15.42 show examples of operation when an auto-refresh request is generated during single access and continuous access, respectively.

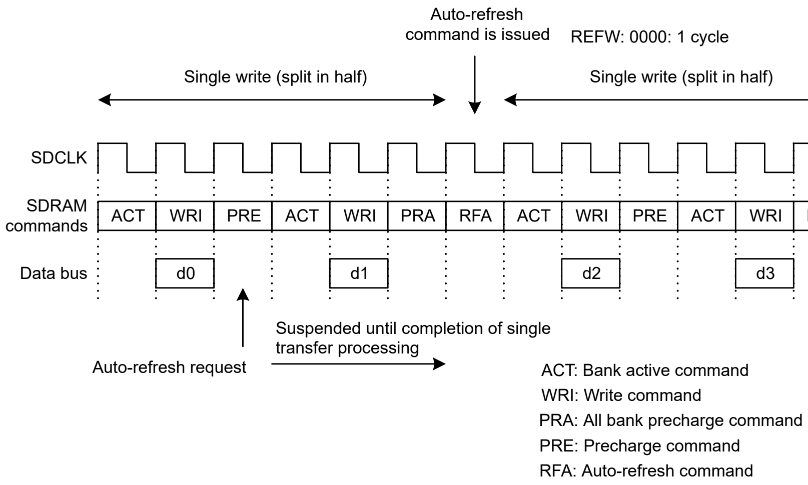


Figure 15.41 Example timing for auto-refresh cycle (2), when the auto-refresh request is made during single access

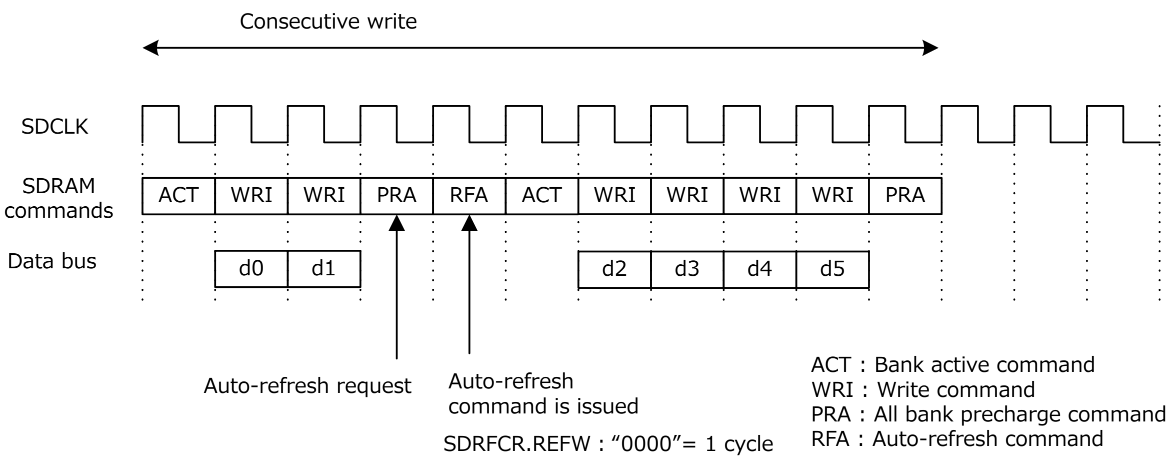


Figure 15.42 Example timing for auto-refresh Cycle (3) when auto-refresh request is made during continuous access

15.6.8 Initialization Sequencer

The SDRAMC has a sequencer to issue SDRAM initialization commands. After a reset, the initialization sequencer must be activated without fail. Operation is not guaranteed if the SDRAM is not initialized.

The SDRAM initialization sequencer issues an all-bank precharge command followed by auto-refresh commands n times, where n = 1 to 15. The SDRAM initialization sequence timing can be set using the SDRAM Initialization Register (SDIR). The SDRAM initialization sequence can be activated using the SDRAM Initialization Sequence Control Register (SDICR). These registers must be set only when the conditions listed in Table 15.31 are satisfied.

Figure 15.43 shows a timing example of the SDRAM initialization sequence. When the ARFC[3:0] bits in SDIR are set so that auto-refresh operation is performed two or more times, auto-refresh cycles are repeated in the initialization sequence accordingly.

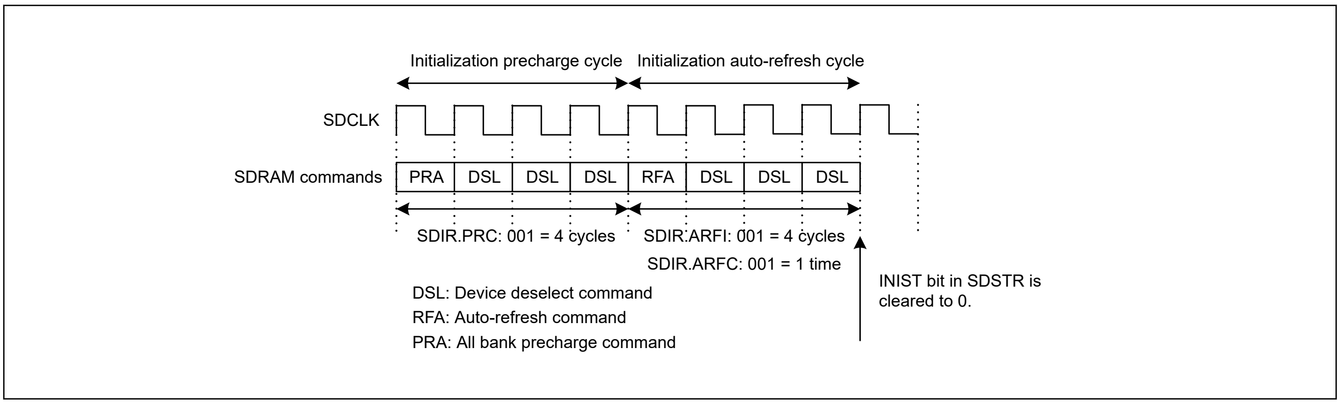


Figure 15.43 Example timing for SDRAM initialization sequence

15.6.9 Setting the Mode Register

Setting the SDRAM Mode Register (SDMOD) allows the mode register set command to be issued to the SDRAM and the value set in the MR[14:0] bits in SDMOD to be output to the lower bits of the address, specifically to A14 to A00 for 8-bit bus width, A15 to A01 for 16-bit bus width or A16 to A02 for 32-bit bus width. Before setting the mode register, set the SDRAM Bus Width Select bits in the SDC Control Register (SDCCR.BSIZE[1:0]) to determine the data bus width of the SDRAM.

Figure 15.44 shows the mode register setting timing.

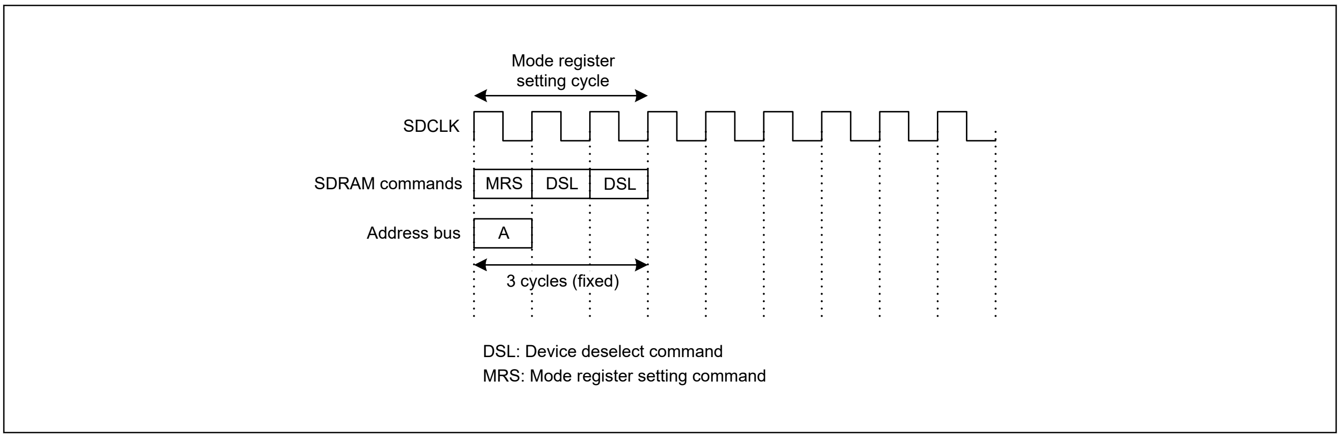


Figure 15.44 Mode register setting timing

15.6.10 Read/Write Access

There are two types of SDRAM read / write access:

- Single access
  The row address is output each time the data is accessed to access the data.
- Consecutive access
  When the same row address is accessed consecutively, after the row address is output, the data can be accessed at high speed simply by changing the column address.

When the continuous access enable bit (SDAMOD.BE) of the SDRAM access mode register is 1, consecutive access operation of SDRAM is enabled.

If the transfer from the bus master has a burst length of 2 or more and the transfer data size is less than or equal to the external bus width, consecutive access will occur.

(1) Single access

Figure 15.45 shows timing examples for single read and single write. The access timing changes depending on the setting of the SDRAM timing register (SDTR). For details, see section 15.6.11.3. Timing Register Settings and Access Timing.

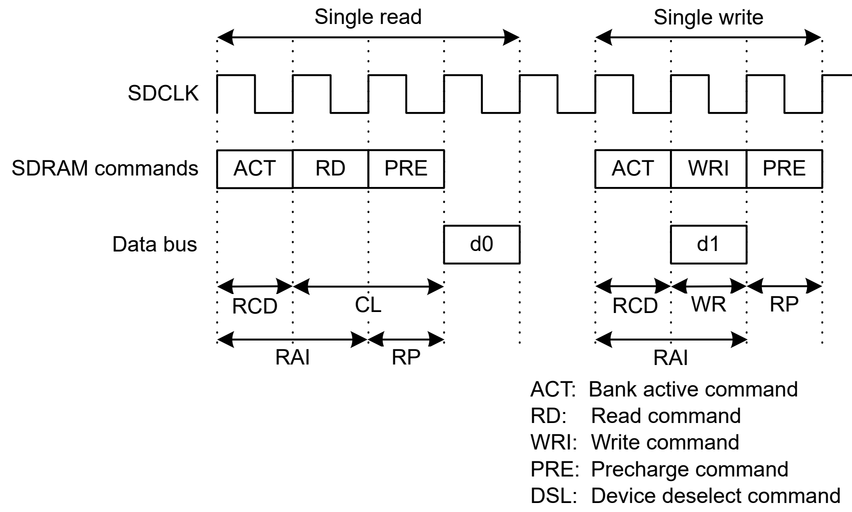


Figure 15.45 Example timing for single read/write

(2) Consecutive access

Figure 15.46 shows an example of the timing when performing consecutive read and continuous write of 4 data.

In addition, if the row address of SDRAM changes during transfer, the row is automatically deactivated and activated.

The access timing changes depending on the setting of the SDRAM timing register (SDTR). For details, see section 15.6.11.3. Timing Register Settings and Access Timing.

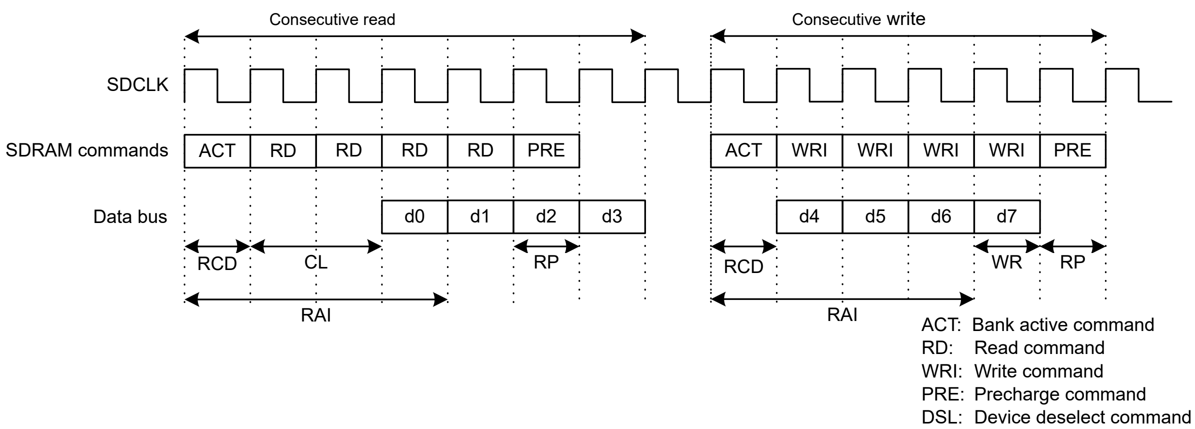


Figure 15.46 Example timing for Consecutive read/write

Figure 15.47 shows an example of the timing when the row address changes during consecutive writing.

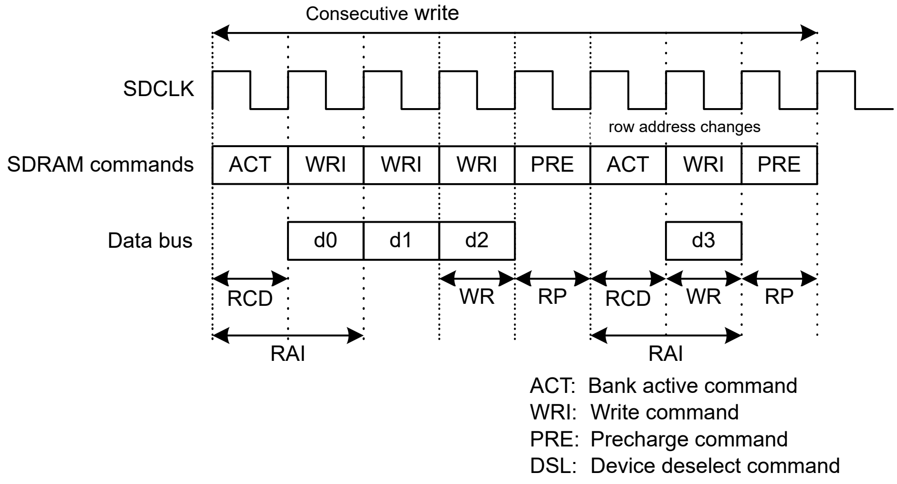


Figure 15.47 Example timing for Consecutive write when row address changes

Note: When consecutive burst write accesses are issued from bus masters to SDRAM area, a redundant write access may occurs if the following conditions are all satisfied, the last write access to SDRAM in the preceding burst access is repeated with the same address and the same write data.

- The preceding burst write and the following burst write have the same transfer size, and the transfer size is 8, 16, or 32 bits.
- The address of the subsequent burst write is the last address of the preceding burst write plus the transfer size.
- The burst length of the preceding burst write is 2 or more and that of the subsequent burst write is 1.

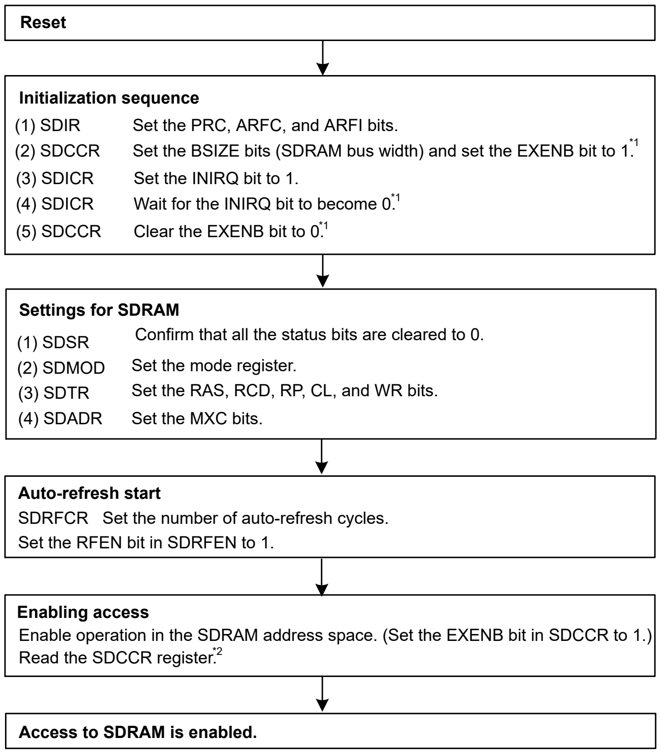
15.6.11 SDRAMC Setting Examples

This section describes the following:

- SDRAMC setting procedure
- Timing register setting examples
- Procedure for transitioning to and recovering from self-refresh mode.

15.6.11.1 SDRAMC Access Procedure

Figure 15.48 shows the SDRAMC setting procedure.



Note 1. If the SDRAM bus width is 16-bit, these two steps can be skipped.

Note 2. When setting with CPU0 (Cortex-M85 core), use a barrier instruction to prevent the SDCCR read access and external bus access from being reversed.

Figure 15.48 SDRAMC setting procedure

15.6.11.2 Procedure for Transitioning to and Recovering from Self-refresh Mode

Figure 15.49 shows the procedure for transitioning to and recovering from self-refresh mode.

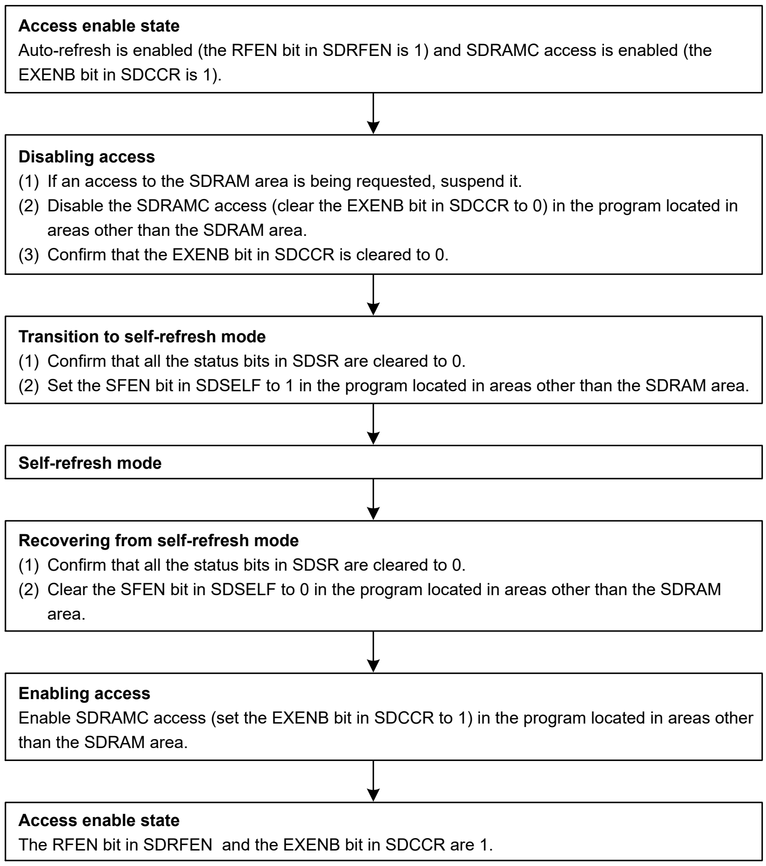


Figure 15.49 Procedure for transitioning to and recovering from self-refresh mode

Note: Self-refresh mode cannot be invoked during SDRAM access. SDRAM access must be disabled during both transition to and recovery from self-refresh mode. Follow the programming instructions shown in Figure 15.50.

Before transitioning to self-refresh mode, disable access to the SDRAM area.

During transition to self-refresh mode, self-refresh operation, and recovery from self-refresh mode, do not allow any operand access or instruction fetch, including prefetch to the SDRAM area, to be generated.

Figure 15.50 shows the procedure for transitioning to and recovering from self-refresh mode in Deep Software Standby mode.

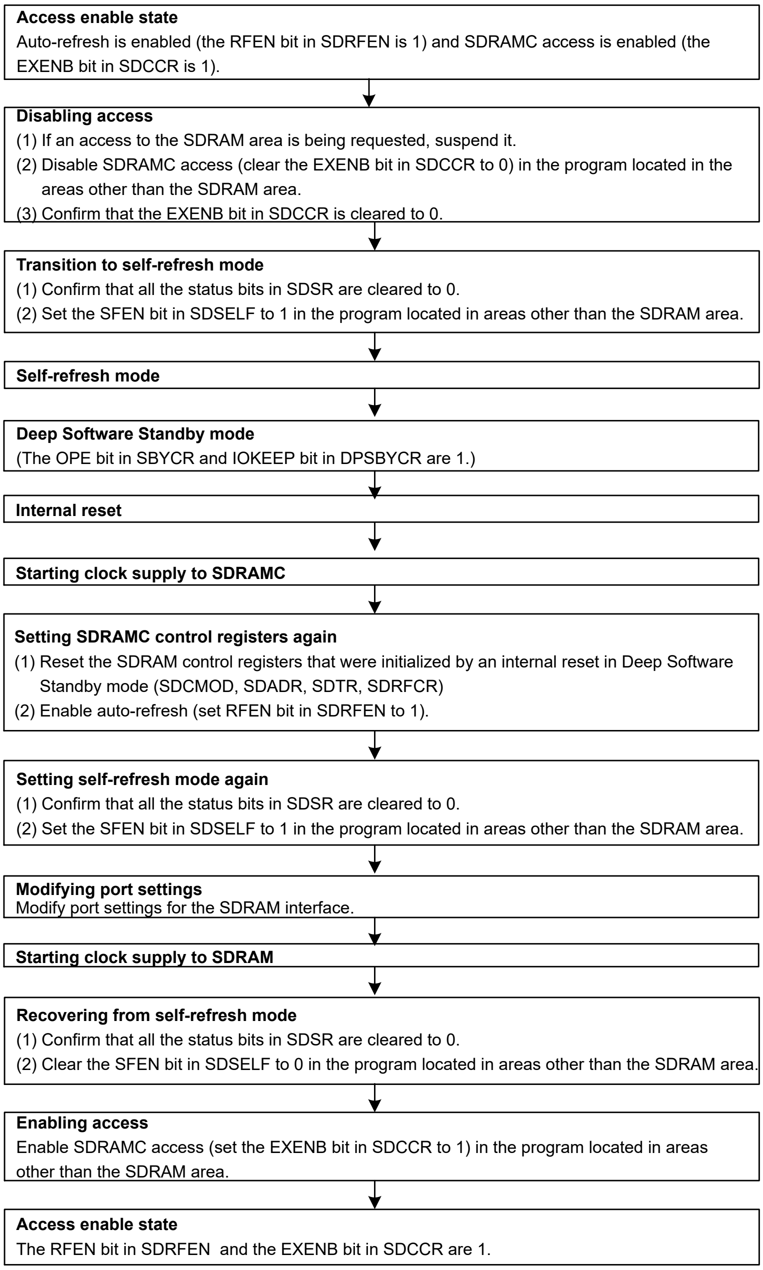


Figure 15.50 Procedure for transitioning to and recovering from self-refresh mode in Deep Software Standby mode

15.6.11.3 Timing Register Settings and Access Timing

This section describes the relationship between read and write timing and the settings in the SDRAM Timing Register (SDTR).

(1) Single read timing examples

Figure 15.51 to Figure 15.55 show the relationship between single read timing and the SDTR register settings. Table 15.32 shows the correspondence between the figures and the SDTR register settings.

During read access, the next bus access is enabled at the earliest 2 cycles after the read data becomes valid. However, if two or more accesses occur for one transfer request, the next bus access is enabled at the earliest 1 cycle after the read data becomes valid, as shown in Figure 15.55.

Table 15.32 Correspondence between timing figures and STDR register settings for single read timing

Figure number	RAI[2:0] settings	Number of cycles	RCD[1:0] settings	Number of cycles	RP[2:0] settings	Number of cycles	CL[2:0] settings	Number of cycles
Figure 15.51	010	3	00	1	001	2	010	2
Figure 15.52	000	1	01	2	001	2	010	2
Figure 15.53	000	1	01	2	001	2	011	3
Figure 15.54 Figure 15.55	010	3	00	1	000	1	010	2

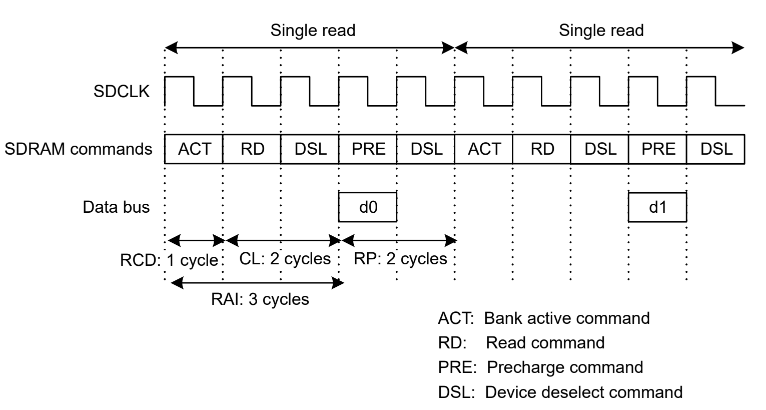


Figure 15.51 Example timing for single read (1)

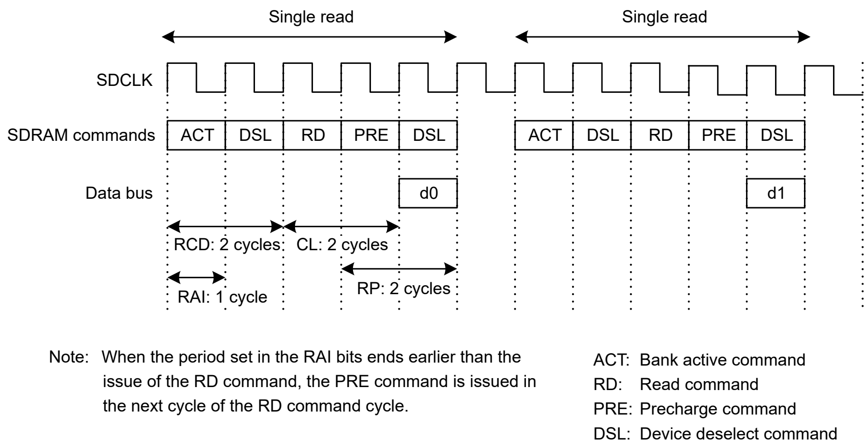


Figure 15.52 Example timing for single read (2)

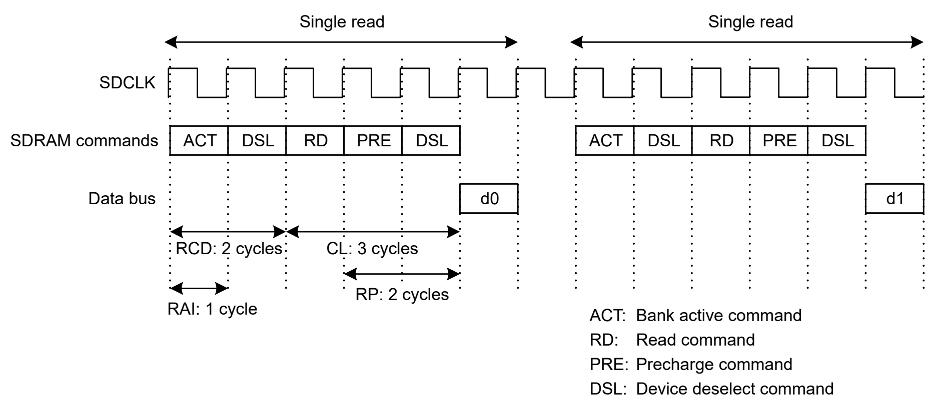


Figure 15.53 Example timing for single read (3)

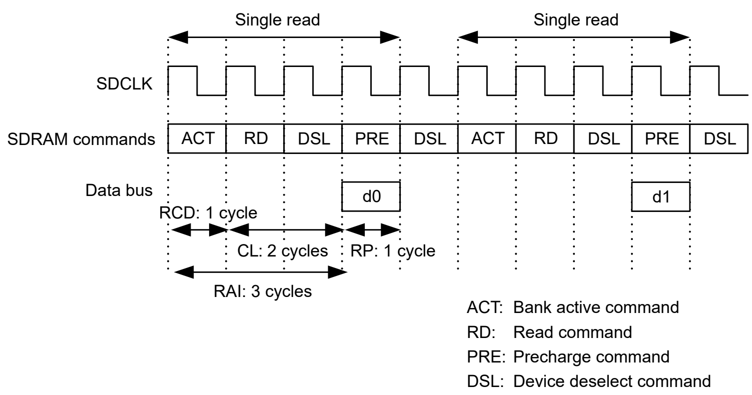


Figure 15.54 Example timing for single read (4)

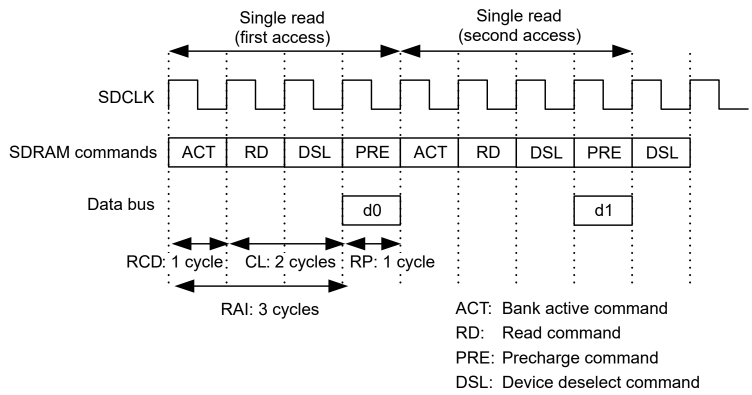


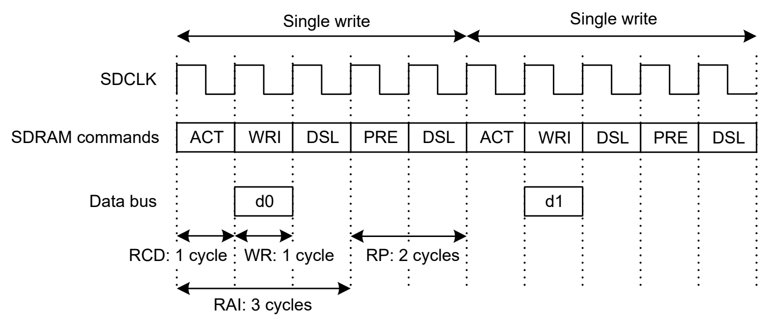
Figure 15.55 Example timing for single read (5), when two bus accesses occur for one transfer request

(2) Single write timing examples

Figure 15.56 to Figure 15.60 show the relationship between the single write timing and the SDTR register settings. Table 15.33 shows the association between the figures and the SDTR register settings. During write access, the next bus access is enabled at the earliest 2 cycles after a precharge command (PRE) is issued. However, if two or more accesses occur for one transfer request, the next bus access is enabled at the earliest 1 cycle after the PRE is issued, as shown in Figure 15.60.

Table 15.33 Association between timing figures and STDR register settings for single write timing

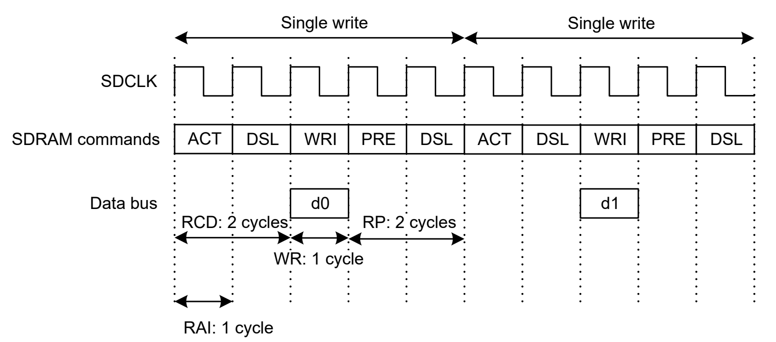
Figure number	RAI[2:0] settings	Number of cycles	RCD[1:0] settings	Number of cycles	RP[2:0] settings	Number of cycles	WR settings	Number of cycles
Figure 15.56	010	3	00	1	001	2	0	1
Figure 15.57	000	1	01	2	001	2	0	1
Figure 15.58	000	1	01	2	001	2	1	2
Figure 15.59 Figure 15.60	010	3	00	0	000	2	0	1



Note: When the period specified in the RAI bits is longer than the period specified in WR bits, starting from the issue of the WRI command, the RAI setting is applied.

ACT: Bank active command
WRI: Write command
PRE: Precharge command
DSL: Device deselect command

Figure 15.56 Example timing for single write (1)



Note: When the period specified in the RAI bits is shorter than the period specified in WR bits, starting from the issue of the WRI command, the WR setting is applied.

ACT: Bank active command
WRI: Write command
PRE: Precharge command
DSL: Device deselect command

Figure 15.57 Example timing for single write (2)

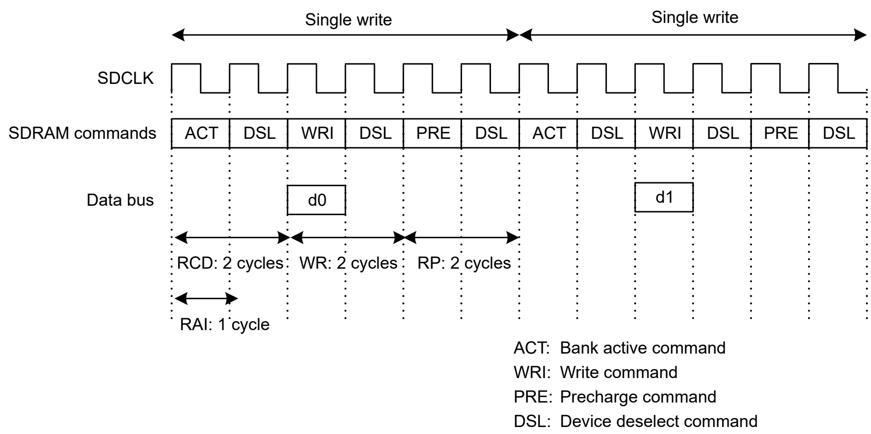


Figure 15.58 Example timing for single write (3)

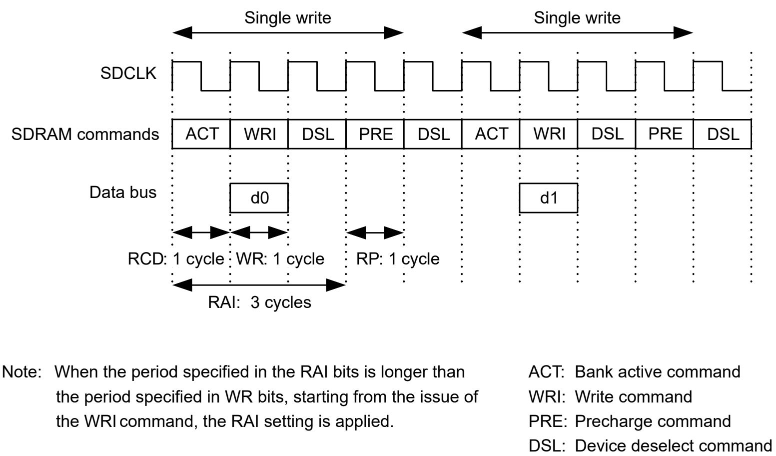


Figure 15.59 Example timing for single write (4)

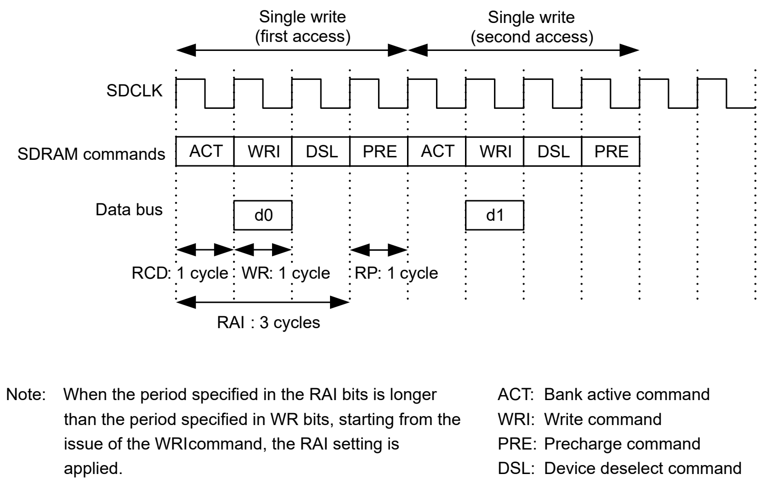


Figure 15.60 Example timing for single write (5), when two bus accesses occur for one transfer request

(3) Consecutive Read Timing Examples

Figure 15.61 to Figure 15.63 show the relationship between the consecutive read timing for four data and the SDTR register settings. Table 15.34 shows the correspondence between the figures and the SDTR register settings.

Table 15.34 Correspondence between Target Figures and STDR Register Settings (Consecutive Read Timing)

Figure No.	RAI[2:0] Settings	Number of Cycle	RCD[1:0] Settings	Number of Cycle	RP[2:0] Settings	Number of Cycle	CL[2:0] Settings	Number of Cycle
Figure 15.61	010	3	00	1	001	2	010	2
Figure 15.62	000	1	01	2	001	2	010	2
Figure 15.63	000	1	01	2	001	2	011	3

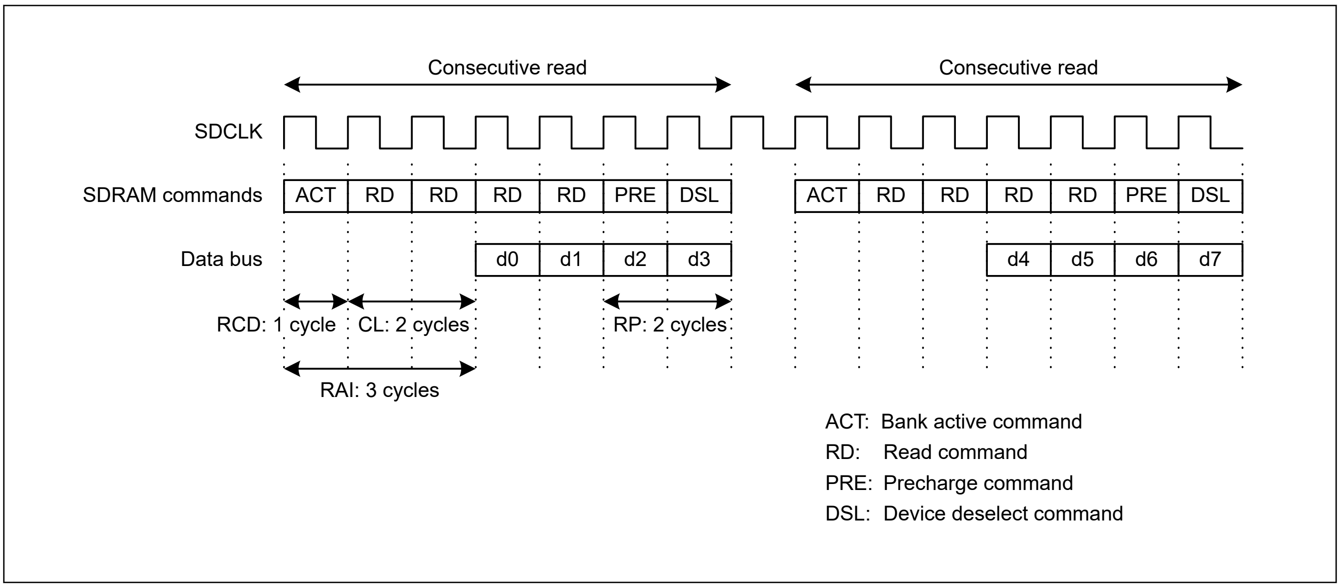


Figure 15.61 Timing example of consecutive read (1)

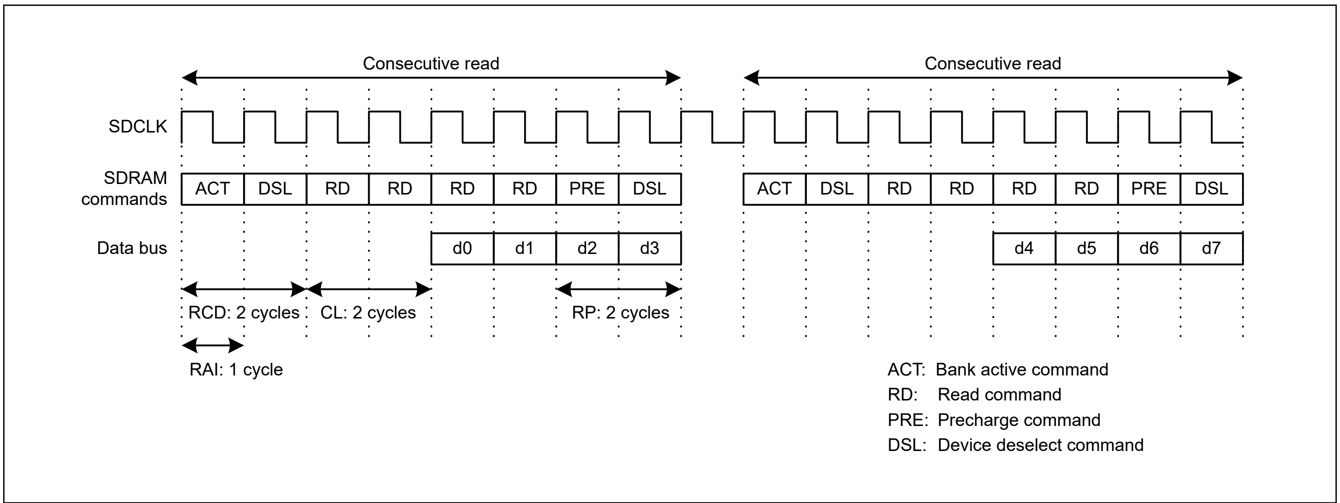


Figure 15.62 Timing example of consecutive read (2)

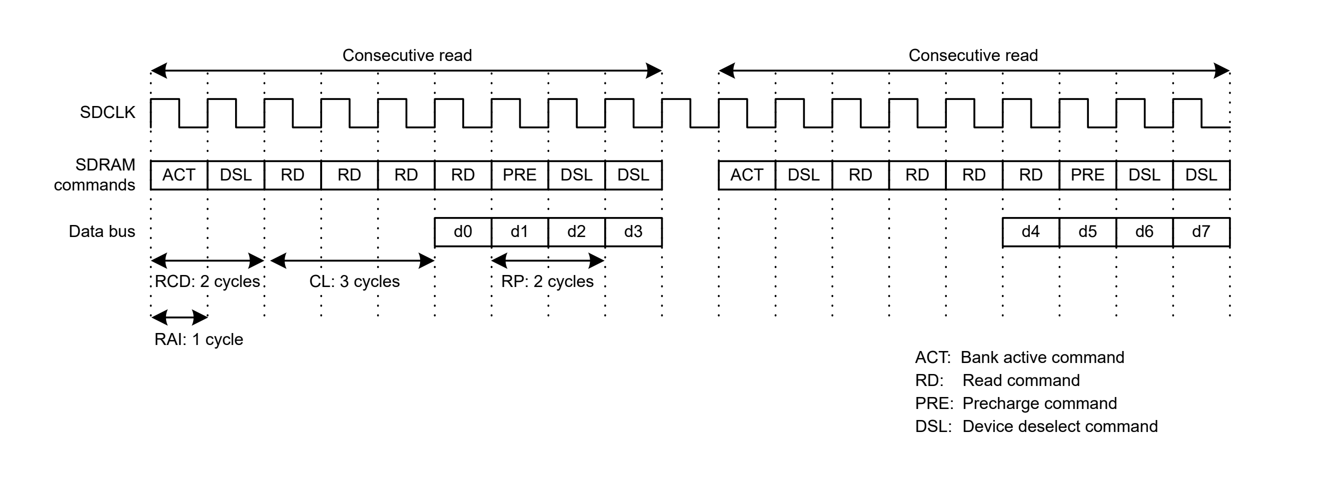


Figure 15.63 Timing example of consecutive read (3)

(4) Consecutive Write Timing Examples

Figure 15.64 to Figure 15.66 show the relationship between the consecutive write timing for four data and the SDTR register settings. Table 15.35 shows the correspondence between the figures and the SDTR register settings.

Table 15.35 Correspondence between Target Figures and STDR Register Settings (Consecutive Write Timing)

Figure No.	RAI[2:0] Settings	Number of Cycles	RCD[1:0] Settings	Number of Cycles	RP[2:0] Settings	Number of Cycles	WR Settings	Number of Cycles
Figure 15.64	010	3	00	1	001	2	0	1
Figure 15.65	000	1	01	2	001	2	0	1
Figure 15.66	000	1	01	2	001	2	1	2

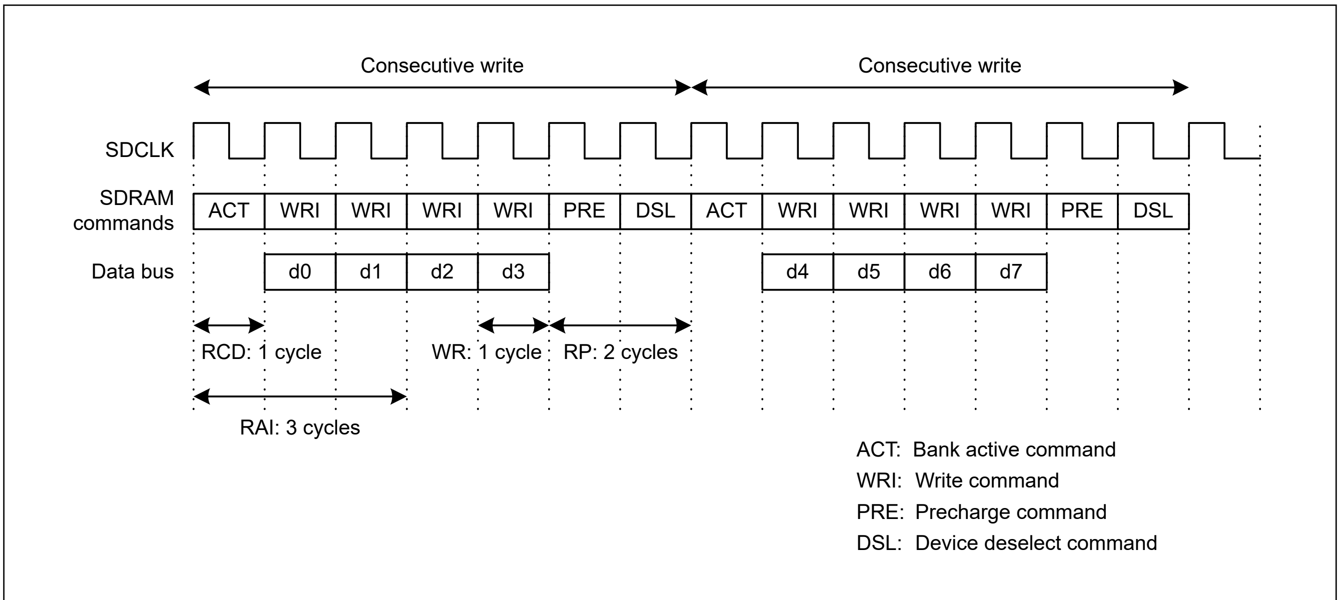


Figure 15.64 Timing example of consecutive write (1)

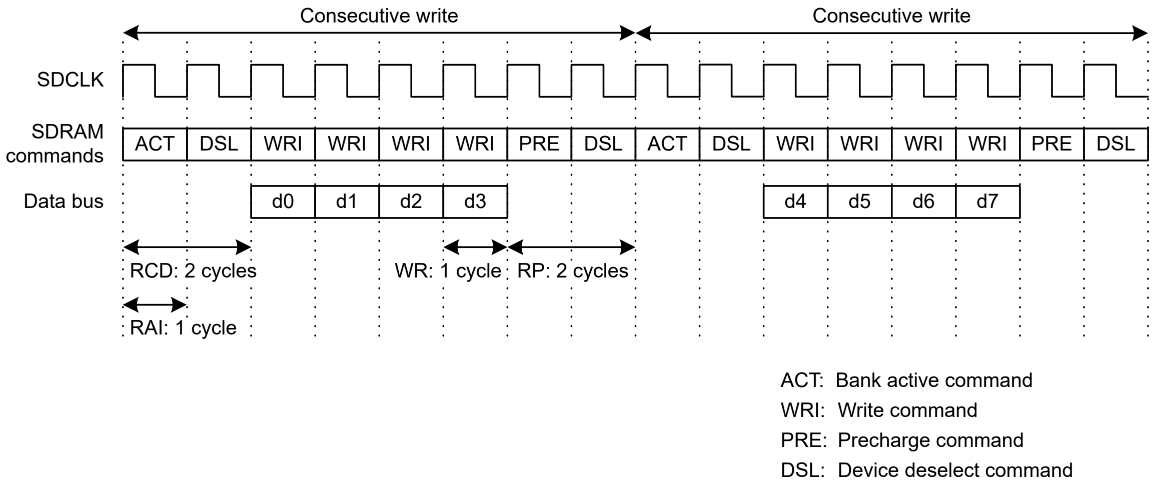


Figure 15.65 Timing example of consecutive write (2)

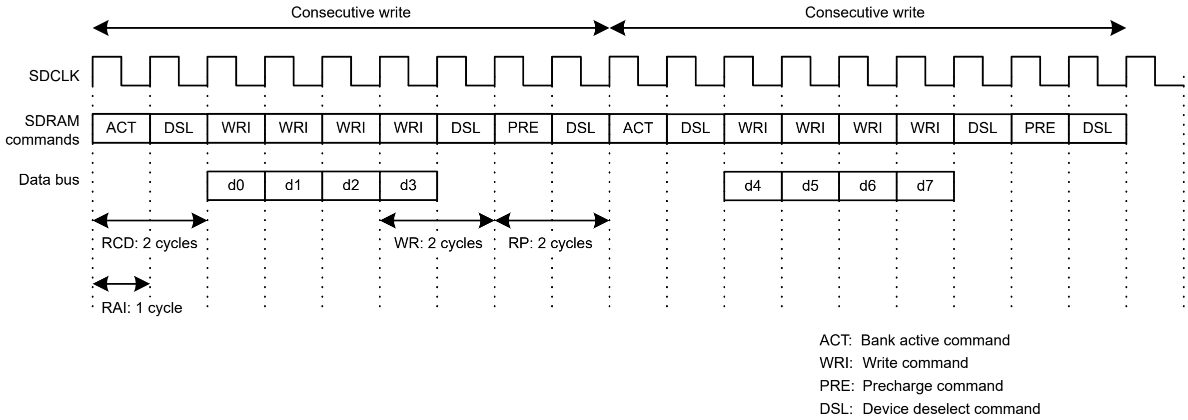


Figure 15.66 Timing example of consecutive write (3)

15.6.12 Address Multiplexing

In the SDRAM space, row and column addresses are multiplexed. The size of the shift in a row address must be specified in the Address Multiplex Select bits (SDADR.MXC[1:0]) in the SDRAM Address Register (SDADR). Additionally, in the SDRAM space, the address precharge-select command (Precharge-sel) is output to the upper bits of column addresses. Table 15.36 shows the relationship between the SDADR.MXC[1:0] settings and the shift amount.

Table 15.36 Address multiplexing (1 of 2)

MXC [1:0]	Shift amount	Data bus width	Address	Address pins external to the MCU																
				A16	A15	A14	A13	A12	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
00	8 bits	8 bits	Row	A24	A23	A22	A21	A20	A19	A18[*1]	A17	A16	A15	A14	A13	A12	A11	A10	A09	A08
			Column	A24	A23	A22	A21	A20	A19	P	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
		16 bits	Row	A24	A23	A22	A21	A20	A19[*1]	A18	A17	A16	A15	A14	A13	A12	A11	A10	A09	A08
			Column	A24	A23	A22	A21	A20	P	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
		32 bits	Row	A24	A23	A22	A21	A20[*1]	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A09	A08
			Column	A24	A23	A22	A21	P	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00

Table 15.36 Address multiplexing (2 of 2)

MXC [1:0]	Shift amount	Data bus width	Address	Address pins external to the MCU																
				A16	A15	A14	A13	A12	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
01	9 bits	8 bits	Row	A25	A24	A23	A22	A21	A20	A19[*1]	A18	A17	A16	A15	A14	A13	A12	A11	A10	A09
			Column	A25	A24	A23	A22	A21	A20	P	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
		16 bits	Row	A25	A24	A23	A22	A21	A20[*1]	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A09
			Column	A25	A24	A23	A22	A21	P	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
		32 bits	Row	A25	A24	A23	A22	A21[*1]	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A09
			Column	A25	A24	A23	A22	P	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
10	10 bits	8 bits	Row	A26	A25	A24	A23	A22	A21	A20[*1]	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10
			Column	A26	A25	A24	A23	A22	A21	P	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
		16 bits	Row	A26	A25	A24	A23	A22	A21[*1]	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10
			Column	A26	A25	A24	A23	A22	P	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
		32 bits	Row	A26	A25	A24	A23	A22[*1]	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10
			Column	A26	A25	A24	A23	P	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
11	11 bits	8 bits	Row	—	A26	A25	A24	A23	A22	A21[*1]	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11
			Column	—	A26	A25	A24	A23	A10	P	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
		16 bits	Row	—	A26	A25	A24	A23	A22[*1]	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11
			Column	—	A26	A25	A24	A11	P	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
		32 bits	Row	—	A26	A25	A24	A23[*1]	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11
			Column	—	A26	A25	A24	P	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00

Note: P: Precharge-select command (Precharge-sel) is output.

Note 1. When the PALL command is issued, Precharge-sel = 1 (high) is output. When the Active command is issued, the associated address is output.

15.6.13 Example SDRAM Connections

15.6.13.1 32-Bit Bus Space

Table 15.37 shows an example for connecting to two 512-Mb SDRAMs with 13-bit row address, 10-bit column address and 16-bit bus.

Table 15.37 Example of 512-Mb SDRAM × 2 connections to a 16-bit bus (1 of 2)

This MCU (shift amount: 10 bits)					SDRAM (512 Mb, with 16-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
1	RAS			→	RAS#			
2	CAS			→	CAS#			
3	WE			→	WE#			
4	CKE			→	CKE			
5	SDCS			→	CS#			
6	SDCLK			→	CLK			
7	DQM1			→	UDQM			
8	DQM0			→	LDQM			
9	A16	A26	A26	→	BA1 (A14)	BA1	—	—
10	A15	A25	A25	→	BA0 (A13)	BA0	—	—
11	A14	A24	A24	→	A12		A12	—

Table 15.37 Example of 512-Mb SDRAM × 2 connections to a 16-bit bus (2 of 2)

This MCU (shift amount: 10 bits)					SDRAM (512 Mb, with 16-bit bus)			
No.	**Pin name**	**Row**	**Column**		**Pin name**	**BA/AP**	**Row**	**Column**
12	A13	A23	A23	→	A11		A11	—
13	A12	A22	P	→	A10	AP	A10	—
14	A11	A21	A11	→	A9		A9	A9
15	A10	A20	A10	→	A8		A8	A8
16	A09	A19	A9	→	A7		A7	A7
17	A08	A18	A8	→	A6		A6	A6
18	A07	A17	A7	→	A5		A5	A5
19	A06	A16	A6	→	A4		A4	A4
20	A05	A15	A5	→	A3		A3	A3
21	A04	A14	A4	→	A2		A2	A2
22	A03	A13	A3	→	A1		A1	A1
23	A02	A12	A2	→	A0		A0	A0
24	DQ15 to DQ08			↔	DQ[15:8]			
25	DQ07 to DQ00			↔	DQ[7:0]			
This MCU (shift amount 10 bits)					SDRAM (512 Mb, with 16-bit bus)			
No.*1	Pin name	Row	Column		Pin name	BA/AP	Row	Column
(1)	RAS			→	RAS#			
(2)	CAS			→	CAS#			
(3)	WE			→	WE#			
(4)	CKE			→	CKE			
(5)	SDCS			→	CS#			
(6)	SDCLK			→	CLK			
26	DQM3			→	UDQM			
27	DQM2			→	LDQM			
(9)	A16	A26	A26	→	BA1 (A14)	BA1	—	—
(10)	A15	A25	A25	→	BA0 (A13)	BA0	—	—
(11)	A14	A24	A24	→	A12		A12	—
(12)	A13	A23	A23	→	A11		A11	—
(13)	A12	A22	P	→	A10	AP	A10	—
(14)	A11	A21	A11	→	A9		A9	A9
(15)	A10	A20	A10	→	A8		A8	A8
(16)	A09	A19	A9	→	A7		A7	A7
(17)	A08	A18	A8	→	A6		A6	A6
(18)	A07	A17	A7	→	A5		A5	A5
(19)	A06	A16	A6	→	A4		A4	A4
(20)	A05	A15	A5	→	A3		A3	A3
(21)	A04	A14	A4	→	A2		A2	A2
(22)	A03	A13	A3	→	A1		A1	A1
(23)	A02	A12	A2	→	A0		A0	A0
28	DQ31 to DQ24			↔	DQ[15:8]			
29	DQ23 to DQ16			↔	DQ[7:0]			

Note 1. The number (No.) in parentheses indicates that the pin branches out to two SDRAM pins.

Table 15.38 shows an example for connecting to a 256-Mb SDRAM with 12-bit row address, 9-bit column address and 32-bit bus.

Table 15.38 Example of 256-Mb SDRAM × 1 connection to a 32-bit bus

This MCU (shift amount: 9 bits)					SDRAM (256 Mb, with 32-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
1	RAS			→	RAS#			
2	CAS			→	CAS#			
3	WE			→	WE#			
4	CKE			→	CKE			
5	SDCS			→	CS#			
6	SDCLK			→	CLK			
7	DQM3			→	DQM3			
8	DQM2			→	DQM2			
9	DQM1			→	DQM1			
10	DQM0			→	DQM0			
11	A15	A24	A24	→	BA1 (A13)	BA1	—	—
12	A14	A23	A23	→	BA0 (A12)	BA0	—	—
13	A13	A22	A22	→	A11		A11	—
14	A12	A21	P	→	A10	AP	A10	—
15	A11	A20	A11	→	A9		A9	—
16	A10	A19	A10	→	A8		A8	A8
17	A09	A18	A9	→	A7		A7	A7
18	A08	A17	A8	→	A6		A6	A6
19	A07	A16	A7	→	A5		A5	A5
20	A06	A15	A6	→	A4		A4	A4
21	A05	A14	A5	→	A3		A3	A3
22	A04	A13	A4	→	A2		A2	A2
23	A03	A12	A3	→	A1		A1	A1
24	A02	A11	A2	→	A0		A0	A0
25	DQ31 to DQ24			↔	DQ[31:24]			
26	DQ23 to DQ16			↔	DQ[23:16]			
27	DQ15 to DQ08			↔	DQ[15:8]			
28	DQ07 to DQ00			↔	DQ[7:0]			

Table 15.39 shows an example for connecting to two 128-Mb SDRAMs with 12-bit row address, 9-bit column address and 16-bit bus.

Table 15.39 Example of 128-Mb SDRAM × 2 connections to a 16-bit bus (1 of 3)

This MCU (shift amount: 9 bits)					SDRAM (128 Mb, with 16-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
1	RAS			→	RAS#			
2	CAS			→	CAS#			
3	WE			→	WE#			
4	CKE			→	CKE			

Table 15.39 Example of 128-Mb SDRAM × 2 connections to a 16-bit bus (2 of 3)

This MCU (shift amount: 9 bits)					**SDRAM (128 Mb, with 16-bit bus)**			
No.	**Pin name**	**Row**	**Column**		**Pin name**	**BA/AP**	**Row**	**Column**
5	SDCS			→	CS#			
6	SDCLK			→	CLK			
7	DQM1			→	UDQM			
8	DQM0			→	LDQM			
9	A15	A24	A24	→	BA1 (A13)	BA1	—	—
10	A14	A23	A23	→	BA0 (A12)	BA0	—	—
11	A13	A22	A22	→	A11		A11	—
12	A12	A21	P	→	A10	AP	A10	—
13	A11	A20	A11	→	A9		A9	—
14	A10	A19	A10	→	A8		A8	A8
15	A09	A18	A9	→	A7		A7	A7
16	A08	A17	A8	→	A6		A6	A6
17	A07	A16	A7	→	A5		A5	A5
18	A06	A15	A6	→	A4		A4	A4
19	A05	A14	A5	→	A3		A3	A3
20	A04	A13	A4	→	A2		A2	A2
21	A03	A12	A3	→	A1		A1	A1
22	A02	A11	A2	→	A0		A0	A0
23	DQ15 to DQ08			↔	DQ[15:8]			
24	DQ07 to DQ00			↔	DQ[7:0]			
This MCU (shift amount 9 bits)					SDRAM (128 Mb, with 16-bit bus)			
No.*1	Pin name	Row	Column		Pin name	BA/AP	Row	Column
(1)	RAS			→	RAS#			
(2)	CAS			→	CAS#			
(3)	WE			→	WE#			
(4)	CKE			→	CKE			
(5)	SDCS			→	CS#			
(6)	SDCLK			→	CLK			
25	DQM3			→	UDQM			
26	DQM2			→	LDQM			
(9)	A15	A24	A24	→	BA1 (A13)	BA1	—	—
(10)	A14	A23	A23	→	BA0 (A12)	BA0	—	—
(11)	A13	A22	A22	→	A11		A11	—
(12)	A12	A21	P	→	A10	AP	A10	—
(13)	A11	A20	A11	→	A9		A9	—
(14)	A10	A19	A10	→	A8		A8	A8
(15)	A09	A18	A9	→	A7		A7	A7
(16)	A08	A17	A8	→	A6		A6	A6
(17)	A07	A16	A7	→	A5		A5	A5
(18)	A06	A15	A6	→	A4		A4	A4
(19)	A05	A14	A5	→	A3		A3	A3

Table 15.39 Example of 128-Mb SDRAM × 2 connections to a 16-bit bus (3 of 3)

This MCU (shift amount: 9 bits)					SDRAM (128 Mb, with 16-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
(20)	A04	A13	A4	→	A2		A2	A2
(21)	A03	A12	A3	→	A1		A1	A1
(22)	A02	A11	A2	→	A0		A0	A0
27	DQ31 to DQ24			↔	DQ[15:8]			
28	DQ23 to DQ16			↔	DQ[7:0]			

Note 1. The number (No.) in parentheses indicates that the pin branches out to two SDRAM pins.

15.6.13.2 16-bit Bus Space

Table 15.40 shows an example connection to two 512-Mb SDRAMs with a 13-bit row address, 11-bit column address, and 8-bit bus.

Table 15.40 Example of 512-Mb SDRAM × 2 connections to an 8-bit bus (1 of 2)

This MCU (shift amount: 11 bits)					SDRAM (512 Mb, with 8-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
1	RAS			→	RAS#			
2	CAS			→	CAS#			
3	WE			→	WE#			
4	CKE			→	CKE			
5	SDCS			→	CS#			
6	SDCLK			→	CLK			
7	DQM0			→	LDQM			
8	A15	A26	A26	→	BA1 (A14)	BA1	—	—
9	A14	A25	A25	→	BA0 (A13)	BA0	—	—
10	A13	A24	A24	→	A12		A12	—
11	A12	A23	A11	→	A11		A11	A11
12	A11	A22	P	→	A10	AP	A10	—
13	A10	A21	A10	→	A9		A9	A9
14	A09	A20	A9	→	A8		A8	A8
15	A08	A19	A8	→	A7		A7	A7
16	A07	A18	A7	→	A6		A6	A6
17	A06	A17	A6	→	A5		A5	A5
18	A05	A16	A5	→	A4		A4	A4
19	A04	A15	A4	→	A3		A3	A3
20	A03	A14	A3	→	A2		A2	A2
21	A02	A13	A2	→	A1		A1	A1
22	A01	A12	A1	→	A0		A0	A0
23	DQ07 to DQ00			↔	DQ[7:0]			
This MCU (shift amount 11 bits)					SDRAM (512 Mb, with 8-bit bus)			
No.*1	Pin name	Row	Column		Pin name	BA/AP	Row	Column
(1)	RAS			→	RAS#			
(2)	CAS			→	CAS#			
(3)	WE			→	WE#			

Table 15.40 Example of 512-Mb SDRAM × 2 connections to an 8-bit bus (2 of 2)

This MCU (shift amount: 11 bits)					SDRAM (512 Mb, with 8-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
(4)	CKE			→	CKE			
(5)	SDCS			→	CS#			
(6)	SDCLK			→	CLK			
24	DQM1			→	LDQM			
(8)	A15	A26	A26	→	BA1 (A14)	BA1	—	—
(9)	A14	A25	A25	→	BA0 (A13)	BA0	—	—
(10)	A13	A24	A24	→	A12		A12	—
(11)	A12	A23	A11	→	A11		A11	A11
(12)	A11	A22	P	→	A10	AP	A10	—
(13)	A10	A21	A10	→	A9		A9	A9
(14)	A09	A20	A9	→	A8		A8	A8
(15)	A08	A19	A8	→	A7		A7	A7
(16)	A07	A18	A7	→	A6		A6	A6
(17)	A06	A17	A6	→	A5		A5	A5
(18)	A05	A16	A5	→	A4		A4	A4
(19)	A04	A15	A4	→	A3		A3	A3
(20)	A03	A14	A3	→	A2		A2	A2
(21)	A02	A13	A2	→	A1		A1	A1
(22)	A01	A12	A1	→	A0		A0	A0
25	DQ15 to DQ08			↔	DQ[7:0]			

Note 1. The number (No.) in parentheses indicates that the pin branches out to two SDRAM pins.

Table 15.41 shows an example connection to a 512-Mb SDRAMs with a 13-bit row address, 10-bit column address, and 16-bit bus.

Table 15.41 Example of 512-Mb SDRAM × 1 connection to a 16-bit bus (1 of 2)

This MCU (shift amount: 10 bits)					SDRAM (512 Mb, with 16-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
1	RAS			→	RAS#			
2	CAS			→	CAS#			
3	WE			→	WE#			
4	CKE			→	CKE			
5	SDCS			→	CS#			
6	SDCLK			→	CLK			
7	DQM1			→	UDQM			
8	DQM0			→	LDQM			
9	A15	A25	A25	→	BA1 (A14)	BA1	—	—
10	A14	A24	A24	→	BA0 (A13)	BA0	—	—
11	A13	A23	A23	→	A12		A12	—
12	A12	A22	A22	→	A11		A11	—
13	A11	A21	P	→	A10	AP	A10	—
14	A10	A20	A10	→	A9		A9	A9

Table 15.41 Example of 512-Mb SDRAM × 1 connection to a 16-bit bus (2 of 2)

This MCU (shift amount: 10 bits)					SDRAM (512 Mb, with 16-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
15	A09	A19	A9	→	A8		A8	A8
16	A08	A18	A8	→	A7		A7	A7
17	A07	A17	A7	→	A6		A6	A6
18	A06	A16	A6	→	A5		A5	A5
19	A05	A15	A5	→	A4		A4	A4
20	A04	A14	A4	→	A3		A3	A3
21	A03	A13	A3	→	A2		A2	A2
22	A02	A12	A2	→	A1		A1	A1
23	A01	A11	A1	→	A0		A0	A0
24	DQ15 to DQ08			↔	DQ[15:8]			
25	DQ07 to DQ00			↔	DQ[7:0]			

Table 15.42 shows an example connection to a 256-Mb SDRAMs with a 13-bit row address, 9-bit column address, and 16-bit bus.

Table 15.42 Example of 256-Mb SDRAM × 1 connection to a 16-bit bus (1 of 2)

This MCU (shift amount: 9 bits)					SDRAM (256 Mb, with 16-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
1	RAS			→	RAS#			
2	CAS			→	CAS#			
3	WE			→	WE#			
4	CKE			→	CKE			
5	SDCS			→	CS#			
6	SDCLK			→	CLK			
7	DQM1			→	UDQM			
8	DQM0			→	LDQM			
9	A15	A24	A24	→	BA1 (A14)	BA1	—	—
10	A14	A23	A23	→	BA0 (A13)	BA0	—	—
11	A13	A22	A22	→	A12		A12	—
12	A12	A21	A21	→	A11		A11	—
13	A11	A20	P	→	A10	AP	A10	—
14	A10	A19	A10	→	A9		A9	—
15	A09	A18	A9	→	A8		A8	A8
16	A08	A17	A8	→	A7		A7	A7
17	A07	A16	A7	→	A6		A6	A6
18	A06	A15	A6	→	A5		A5	A5
19	A05	A14	A5	→	A4		A4	A4
20	A04	A13	A4	→	A3		A3	A3
21	A03	A12	A3	→	A2		A2	A2
22	A02	A11	A2	→	A1		A1	A1
23	A01	A10	A1	→	A0		A0	A0
24	DQ15 to DQ08			↔	DQ[15:8]			

Table 15.42 Example of 256-Mb SDRAM × 1 connection to a 16-bit bus (2 of 2)

This MCU (shift amount: 9 bits)					SDRAM (256 Mb, with 16-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
25	DQ07 to DQ00			↔	DQ[7:0]			

15.6.14 Constraints

(1) Prohibition of access across areas of the address space

Access across multiple areas of the address space is prohibited with one access, and operation in that case is not guaranteed. Make sure that one word, longword access does not span two areas across the boundaries of each area of the address space.

(2) Low Power condition

In Software Standby and Deep Software Standby modes, auto-refresh operation is not available because the clock supply to the SDRAMC is stopped. To retain the data in the SDRAM when the SDRAM is externally connected, use the self-refresh function. For the procedure for transitioning to and recovering from self-refresh mode, see section 15.6.6. Self-Refresh.

(3) Setting the SDRAM Timing Register

Set the RAI[2:0] bits in the SDRAM Timing Register (SDTR) to a value less than or equal to the sum of the row column latency (SDTR.RCD[1:0]) and column latency (SDTR.CL[2:0]) settings. Operation is not guaranteed if this condition is not satisfied.

(4) Instruction code constraint

You must fix the instruction code to little-endian order.

See section 15.2.5.1. Endianness Constraint for details.

(5) Consecutive access mode

When SDAMOD.BE is "1" (continuous access enable), setting SDTR.CL = "1" is prohibited, and the operation is not guaranteed when SDTR.CL = "1" is set.

15.7 Bus Error Monitoring Section

This monitoring system monitors each individual area, and whenever it detects an error, it returns the error to the requesting master IP.

15.7.1 Bus Error Types

The following types of errors can occur on each bus:

- Master Security Attribution Unit error
- Bus master MPU error.
- Illegal address access.
- Slave TrustZone Filter error
- Bus error transmitted from each slave IP.

Table 15.46 lists the address ranges where access leads to illegal address access errors. The reserved area in the slave does not trigger an illegal address access error. For more information on the bus master MPU, see section 16, Memory Protection Unit (MPU).

15.7.2 Operations When a Bus Error Occurs

15.7.2.1 Operations When a Bufferable Write Error Occurs

If an access violation occurs due to the bufferable write access an error response may not be returned to the bus master.

When a bufferable write access error that does not return an error response is detected, access is blocked and 1 is set in the bit corresponding to the error occurrence location of MBWERRSTAT. Furthermore, the bus error interrupt request or reset request is generated according to the setting of the BUSOAD.BWERROAD. Since MBWERRSTAT is held until reset other than Bus and Memory Error Reset or cleared by MBWERRCLR, they can be confirmed in the bus error interrupt handler.

A bufferable write access error can occur in the following cases.

1. CPU0 M-AXI accesses CPU1TCMBI and slave bus error or a slave TrustZone Filter error occurs.
2. CPU0 P-AHB accesses illegal address area.
3. CPU0 P-AHB accesses accessible area and slave bus error or a slave TrustZone Filter error occurs.
4. CPU1 C-AHB accesses illegal address area.
5. CPU1 C-AHB accesses accessible area and slave bus error or a slave TrustZone Filter error occurs.
6. CPU1 S-AHB accesses illegal address area.
7. CPU1 S-AHB accesses accessible area and slave bus error or a slave TrustZone Filter error occurs.
8. DMA accesses PABI, PBBI, PIBI, PSBI, CPU0SAHBI and CPU1TCMBI and slave bus error or a slave TrustZone Filter error occurs.

Table 15.43 Bus error notice method for Bufferable write access

OAD Setting	DAP	CPU0, CPU1	Other
BUSOAD.BWERROAD = 0	Interrupt	Interrupt	Interrupt
BUSOAD.BWERROAD = 1	Reset	Reset	Reset

The bus error interrupt request BUS_ERR can be assigned as Non-maskable interrupt request. See the ICU.NMIER register specification for details.

15.7.2.2 Operations When an Error except Bufferable Write Error Occurs

When a bus error occurs for an access other than the bufferable write, error access is blocked, and error operations determine by OAD setting and the bus master are performed. Table 15.44 shows OAD setting corresponding to each error. Table 15.45 shows how to notify when a bus error occurs.

- When OAD setting is interrupt.
  Error response is returned to the bus master.
  If BUSIRQEN<Master Name>.EN=1, the bus error interrupt BUS_ERR is generated.
  The bus error interrupt request BUS_ERR can also be used as Non-maskable interrupt request. See the ICU.NMIER register specifications for details.
- When OAD setting is Reset.
  Error response is returned to the bus master and reset request is generated.

Table 15.44 OAD setting corresponding to each error

Error type	OAD bit
MSAU Error	MSAOAD.OAD
MPU Error	MMPUOAD.OAD
Illegal Address Access Error	BUSOAD.ILERROAD
Slave Bus Error	BUSOAD.SLERROAD
Slave TrustZone Filter error	BUSOAD.SLERROAD

Table 15.45 Bus error notice method for Non-bufferable write access

OAD setting		DAP*1	CPUn (n = 0, 1)	DMAC/DTCn (n = 0, 1)	EDMAC	CEU
Interrupt	Bus	Error response only	BUS_ERR	BUS_ERR	BUS_ERR	BUS_ERR
	Bus Master		BusFault	DMAn_TRANSERR	ETHER_GWEI	—
Reset	Bus/Bus Master	Error response only	Reset	Reset	Reset	Reset

Note: DMAC/DTC0, DMAC/DTC1 and EDMAC can issue interrupt when it receives error response. This interrupt is output to ICU and transmitted from ICU to CPU.

Note 1. Only when the bus master is DAP, only an error response is returned regardless of the BUSOAD.OAD setting, and Interrupt and Reset requests are not generated.

Figure 15.67 shows the OAD setting flow after reset. During this register setting, stop the Bus Master.

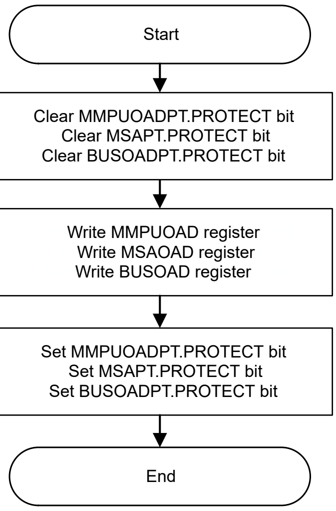


Figure 15.67 OAD setting flow after reset

For details of MMPUOADPT register and MMPUOAD register, see section 16, Memory Protection Unit (MPU).

For details of MSAPT register and MSAOAD register, see section 51, Security Features.

(1) Master Security Attribution Unit Error

The bus master of other than CPU and DAP has MSAU to block access to secure alias address from non-secure masters. The CPU does not have MSAU since it has IDAU and SAU. When MSAU Error is detected, an Error response is returned to the Master. At the same time, set 1 to MSERRSTAT bit of BUSERRSTAT<Master Name>. Furthermore, the bus error interrupt request or reset request is generated according to the setting of the MSAOAD.OAD and BUSIRQEN<Master Name>.EN. Since BUSERRSTAT<Master Name> is held until reset other than Bus error reset, Common memory reset, and CPU local memory reset Error Reset or cleared by BUSERRCLR<Master Name>, they can be confirmed in the bus error interrupt handler or after reset. The bus error interrupt is generated only at the first MSAU Error after reset or clearing BUSERRSTAT<Master Name>.MSERRSTAT bit by BUSERRCLR<Master Name>.

(2) Bus Master MPU Error

The bus master of other than CPU and DAP has Master-MPU for access control of the set address area. The CPU does not have Master-MPU since it has ARM MPU. When Master-MPU Error is detected, an Error response is returned to the Master. At the same time, set 1 to MMERRSTAT bit of BUSERRSTAT<Master Name>. Furthermore, the bus error interrupt request or reset request is generated according to the setting of the MMPUOAD.OAD and BUSIRQEN<Master Name>.EN. Since BUSERRSTAT<Master Name> is held until reset other than Bus error reset, Common memory reset, and CPU local memory reset Error Reset or cleared by BUSERRCLR<Master Name>, they can be confirmed in the bus error interrupt handler or after reset. The bus error interrupt is generated only at the first Master-MPU Error after reset or clearing BUSERRSTAT<Master Name>.MMERRSTAT bit by BUSERRCLR<Master Name>.

(3) Illegal Access Error

According to section 15.7.3. Conditions Leading to Illegal Address Access Errors, the illegal address access error occurs. When illegal address access error is detected, an Error response is returned to the Master. When a master other than DAP caused the illegal address access error, set 1 to ILERRSTAT bit of BUSERRSTAT<Master Name>. Furthermore, the bus error interrupt request or reset request is generated according to the setting of the BUSOAD.ILERROAD and BUSIRQEN<Master Name>.EN. Since BUSERRSTAT<Master Name> is held until reset other than Bus error reset, Common memory reset, and CPU local memory reset Error Reset or cleared by BUSERRCLR<Master Name>, they can be confirmed in the BusFault Handler or the bus error interrupt handler. The bus error interrupt is generated only at the first Illegal Error after reset or clearing BUSERRSTAT<Master Name>.ILERRSTAT bit by BUSERRCLR<Master Name>.

(4) Slave TrustZone Filter Error

When Slave TrustZone Filter Error is detected, an Error response is returned to the Master. When a master other than DAP caused Slave TrustZone Filter error, set 1 to SLERRSTAT bit of BUSERRSTAT<Master Name>. Furthermore, the bus error interrupt request or reset request is generated according to the setting of the BUSOAD.SLERROAD and BUSIRQEN<Master Name>.EN. Since BUSERRSTAT<Master Name> is held until reset other than Bus error reset, Common memory reset, and CPU local memory reset or cleared by BUSERRCLR<Master Name>, they can be confirmed in the bus error interrupt handler or after reset. The bus error interrupt is generated only at the first Slave TrustZone Filter Error after reset or clearing BUSERRSTAT<Master Name>.SLERRSTAT bit by BUSERRCLR<Master Name>.

(5) Slave Bus Error

Slave Bus Error occurs in the slave. When Slave Bus Error is detected, an Error response is returned to the Master. When a master other than DAP caused Slave Bus error, set 1 to SLERRSTAT bit of BUSERRSTAT<Master Name>. Furthermore, the bus error interrupt request or reset request is generated according to the setting of the BUSOAD.SLERROAD and BUSIRQEN<Master Name>.EN. Since BUSERRSTAT<Master Name> is held until reset other than Bus and Memory Error Reset or cleared by BUSERRCLR<Master Name>, they can be confirmed in the BusFault Handler or bus error interrupt handler. The bus error interrupt is generated only at the first Slave-Bus Error after reset or clearing BUSERRSTAT<Master Name>.SLERRSTAT bit by BUSERRCLR<Master Name>.

15.7.3 Conditions Leading to Illegal Address Access Errors

Illegal address access error by bus access occurs in the address space shown in Table 15.46.

Table 15.46 Conditions leading to illegal address access errors (1 of 4)

Address			MASTER bus							
							DMAC/DTC			GRAPHIC
Start	End	SLAVE bus	CPU0 MAXI	CPU0 PAHB	CPU1 CAHB	CPU1 SAHB	0	1	EDMAC	CEU
0x0000_0000	0x01FF_FFFF	Reserved	E	N/A	E	N/A	E	E	E	E
0x0200_0000	0x0207_FFFF	MRC0BI (Read)	—	N/A	—	N/A	—	—	—	N/A
		MRC0BI (Write)	—	N/A	—	N/A	—	—	E	E
0x0208_0000	0x020B_FFFF	MRC0BI (Read)	—/E*1	N/A	—/E*1	N/A	—/E*1	—/E*1	—/E*1	N/A
		MRC0BI (Write)	—/E*1	N/A	—/E*1	N/A	—/E*1	—/E*1	E	E
0x020C_0000	0x020_FFFFF	MRC0BI (Read)	—/E*2	N/A	—/E*2	N/A	—/E*2	—/E*2	—/E*2	N/A
		MRC0BI (Write)	—/E*2	N/A	—/E*2	N/A	—/E*2	—/E*2	E	E
0x0210_0000	0x027F_FFFF	Reserved	E	N/A	E	N/A	E	E	E	E
0x0280_0000	0x0280_1FFF	Reserved	E	N/A	E	N/A	E	E	E	E
0x0280_2000	0x029F_FFFF	Reserved	E	N/A	E	N/A	E	E	E	E
0x02A0_0000	0x02CF_FFFF	MRE0BI	—	N/A	—	N/A	—	—	—	E
0x02D0_0000	0x02DF_FFFF	Reserved	E	N/A	E	N/A	E	E	E	E
0x02E0_0000	0x02FF_FFFF	MRE0BI	—	N/A	—	N/A	—	—	—	E
0x0300_0000	0x07FF_FFFF	Reserved	E	N/A	E	N/A	E	E	E	E
0x0800_0000	0x0FFF_FFFF	OSPI1BI	—	N/A	—	N/A	—	—	—	—
0x1000_0000	0x11FF_FFFF	Reserved	E	N/A	E	N/A	E	E	E	E
0x1200_0000	0x1207_FFFF	MRC0BI (Read)	—	N/A	—	N/A	—	—	—	N/A
		MRC0BI (Write)	—	N/A	—	N/A	—	—	E	E
0x1208_0000	0x120B_FFFF	MRC0BI (Read)	—/E*1	N/A	—/E*1	N/A	—/E*1	—/E*1	—/E*1	N/A
		MRC0BI (Write)	—/E*1	N/A	—/E*1	N/A	—/E*1	—/E*1	E	E
0x120C_0000	0x120F_FFFF	MRC0BI (Read)	—/E*2	N/A	—/E*2	N/A	—/E*2	—/E*2	—/E*2	N/A
		MRC0BI (Write)	—/E*2	N/A	—/E*2	N/A	—/E*2	—/E*2	E	E
0x1210_0000	0x127F_FFFF	Reserved	E	N/A	E	N/A	E	E	E	E

Table 15.46 Conditions leading to illegal address access errors (2 of 4)

Address		SLAVE bus	MASTER bus							
							DMAC/DTC			GRAPHIC
Start	End		CPU0 MAXI	CPU0 PAHB	CPU1 CAHB	CPU1 SAHB	0	1	EDMAC	CEU
0x1280_0000	0x1280_1FFF	Reserved	E	N/A	E	N/A	E	E	E	E
0x1280_2000	0x129F_FFFF	Reserved	E	N/A	E	N/A	E	E	E	E
0x12A0_0000	0x12CF_FFFF	MRE0BI	—	N/A	—	N/A	—	—	—	E
0x12D0_0000	0x12DF_FFFF	Reserved	E	N/A	E	N/A	E	E	E	E
0x12E0_0000	0x12FF_FFFF	MRE0BI	—	N/A	—	N/A	—	—	—	E
0x1300_0000	0x17FF_FFFF	Reserved	E	N/A	E	N/A	E	E	E	E
0x1800_0000	0x1FFF_FFFF	OSPI1BI	—	N/A	—	N/A	—	—	—	—
0x2000_0000	0x21FF_FFFF	Reserved	E	N/A	N/A	E	E	E	E	E
0x2200_0000	0x2207_FFFF	S0BI	—	N/A	N/A	—	—	—	—	—
0x2208_0000	0x220F_FFFF	S1BI	—	N/A	N/A	—	—	—	—	—
0x2210_0000	0x2217_FFFF	S2BI	—	N/A	N/A	—	—	—	—	—
0x2218_0000	0x2219_FFFF	S3BI	—	N/A	N/A	—	—	—	—	—
0x221A_0000	0x221A_FFFF	S0BI	—	N/A	N/A	—	—	—	—	—
0x221B_0000	0x221B_FFFF	S1BI	—	N/A	N/A	—	—	—	—	—
0x221C_0000	0x221C_FFFF	S2BI	—	N/A	N/A	—	—	—	—	—
0x221D_0000	0x221D_3FFF	S3BI	—	N/A	N/A	—	—	—	—	—
0x221D_4000	0x27FF_FFFF	Reserved	E	N/A	N/A	E	E	E	E	E
0x2800_0000	0x2803_FFFF	CPU0SAHBI	E	N/A	N/A	—	—	—	E	E
0x2804_0000	0x29FF_FFFF	Reserved	E	N/A	N/A	E	E	E	E	E
0x2A00_0000	0x2A01_FFFF	CPU1TCMBI	—	N/A	N/A	E	—	—	E	E
0x2A02_0000	0x2FFF_FFFF	Reserved	E	N/A	N/A	E	E	E	E	E
0x3000_0000	0x31FF_FFFF	Reserved	E	N/A	N/A	E	E	E	E	E
0x3200_0000	0x3207_FFFF	S0BI	—	N/A	N/A	—	—	—	—	—
0x3208_0000	0x320F_FFFF	S1BI	—	N/A	N/A	—	—	—	—	—
0x3210_0000	0x3217_FFFF	S2BI	—	N/A	N/A	—	—	—	—	—
0x3218_0000	0x3219_FFFF	S3BI	—	N/A	N/A	—	—	—	—	—
0x321A_0000	0x321A_FFFF	S0BI	—	N/A	N/A	—	—	—	—	—
0x321B_0000	0x321B_FFFF	S1BI	—	N/A	N/A	—	—	—	—	—
0x321C_0000	0x321C_FFFF	S2BI	—	N/A	N/A	—	—	—	—	—
0x321D_0000	0x321D_3FFF	S3BI	—	N/A	N/A	—	—	—	—	—
0x321D_4000	0x37FF_FFFF	Reserved	E	N/A	N/A	E	E	E	E	E
0x3800_0000	0x3803_FFFF	CPU0SAHBI	E	N/A	N/A	—	—	—	E	E
0x3804_0000	0x39FF_FFFF	Reserved	E	N/A	N/A	E	E	E	E	E
0x3A00_0000	0x3A01_FFFF	CPU1TCMBI	—	N/A	N/A	E	—	—	E	E
0x3A02_0000	0x3FFF_FFFF	Reserved	E	N/A	N/A	E	E	E	E	E
0x4000_0000	0x4000_1FFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x4000_2000	0x4000_2FFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x4000_3000	0x4000_3FFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x4000_4000	0x4000_4FFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x4000_5000	0x4000_5FFF	Reserved	N/A	E	N/A	E	E	E	E	E
0x4000_6000	0x4000_6FFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x4000_7000	0x4000_7FFF	Reserved	N/A	E	N/A	E	E	E	E	E
0x4000_8000	0x4000_8FFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x4000_9000	0x4000_9FFF	Reserved	N/A	E	N/A	E	E	E	E	E

Table 15.46 Conditions leading to illegal address access errors (3 of 4)

Address			MASTER bus							
							DMAC/DTC			GRAPHIC
Start	End	SLAVE bus	CPU0 MAXI	CPU0 PAHB	CPU1 CAHB	CPU1 SAHB	0	1	EDMAC	CEU
0x4000_A000	0x4000_AFFF	PS0BI	N/A	—	N/A	N/A	E	E	E	E
		PS1BI	N/A	N/A	N/A	—	E	E	E	E
0x4000_B000	0x4000_BFFF	Reserved	N/A	E	N/A	E	E	E	E	E
0x4000_C000	0x4000_CFFF	PS0BI	N/A	—	N/A	N/A	E	E	E	E
		PS1BI	N/A	N/A	N/A	—	E	E	E	E
0x4000_D000	0x4000_EFFF	Reserved	N/A	E	N/A	E	E	E	E	E
0x4000_F000	0x4000_FFFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x4001_0000	0x4001_AFFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x4001_B000	0x4001_BFFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x4001_C000	0x4001_CFFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x4001_D000	0x4001_DFFF	Reserved	N/A	E	N/A	E	E	E	E	E
0x4001_E000	0x4001_EFFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x4001_F000	0x4001_FFFF	Reserved	N/A	E	N/A	E	E	E	E	E
0x4002_0000	0x4002_0FFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x4002_1000	0x4011_FFFF	Reserved	N/A	E	N/A	E	E	E	E	E
0x4012_0000	0x4013_FFFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x4014_0000	0x4014_0FFF	Reserved	N/A	—	N/A	—	—	—	E	E
0x4014_1000	0x401F_FFFF	Reserved	N/A	E	N/A	E	E	E	E	E
0x4020_0000	0x402F_FFFF	PBBI	N/A	—	N/A	—	—	—	E	E
0x4030_0000	0x403F_FFFF	PABI	N/A	—	N/A	—	—	—	E	E
0x4040_0000	0x404F_FFFF	PIBI	N/A	—	N/A	—	—	—	E	E
0x4050_0000	0x4FFF_FFFF	Reserved	N/A	E	N/A	E	E	E	E	E
0x5000_0000	0x5000_1FFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x5000_2000	0x5000_2FFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x5000_3000	0x5000_3FFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x5000_4000	0x5000_4FFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x5000_5000	0x5000_5FFF	Reserved	N/A	E	N/A	E	E	E	E	E
0x5000_6000	0x5000_6FFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x5000_7000	0x5000_7FFF	Reserved	N/A	E	N/A	E	E	E	E	E
0x5000_8000	0x5000_8FFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x5000_9000	0x5000_9FFF	Reserved	N/A	E	N/A	E	E	E	E	E
0x5000_A000	0x5000_AFFF	PS0BI	N/A	—	N/A	E	E	E	E	E
		PS1BI	N/A	E	N/A	—	E	E	E	E
0x5000_B000	0x5000_BFFF	Reserved	N/A	E	N/A	E	E	E	E	E
0x5000_C000	0x5000_CFFF	PS0BI	N/A	—	N/A	E	E	E	E	E
		PS1BI	N/A	E	N/A	—	E	E	E	E
0x5000_D000	0x5000_EFFF	Reserved	N/A	E	N/A	E	E	E	E	E
0x5000_F000	0x5000_FFFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x5001_0000	0x5001_AFFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x5001_B000	0x5001_BFFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x5001_C000	0x5001_CFFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x5001_D000	0x5001_DFFF	Reserved	N/A	E	N/A	E	E	E	E	E
0x5001_E000	0x5001_EFFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x5001_F000	0x5001_FFFF	Reserved	N/A	E	N/A	E	E	E	E	E

Table 15.46 Conditions leading to illegal address access errors (4 of 4)

Address		SLAVE bus	MASTER bus							
							DMAC/DTC			GRAPHIC
Start	End		CPU0 MAXI	CPU0 PAHB	CPU1 CAHB	CPU1 SAHB	0	1	EDMAC	CEU
0x5002_0000	0x5002_0FFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x5002_1000	0x5011_FFFF	Reserved	N/A	E	N/A	E	E	E	E	E
0x5012_0000	0x5013_FFFF	PSBI	N/A	—	N/A	—	—	—	E	E
0x5014_0000	0x5014_0FFF	Reserved	N/A	—	N/A	—	—	—	E	E
0x5014_1000	0x501F_FFFF	Reserved	N/A	E	N/A	E	E	E	E	E
0x5020_0000	0x502F_FFFF	PBBI	N/A	—	N/A	—	—	—	E	E
0x5030_0000	0x503F_FFFF	PABI	N/A	—	N/A	—	—	—	E	E
0x5040_0000	0x504F_FFFF	PIBI	N/A	—	N/A	—	—	—	E	E
0x5050_0000	0x5FFF_FFFF	Reserved	N/A	E	N/A	E	E	E	E	E
0x6000_0000	0x6FFF_FFFF	ECBI	—	N/A	N/A	—	—	—	—	—
0x7000_0000	0x7FFF_FFFF	OSPI1BI	—/E*3	N/A	N/A	—/E*3	—/E*3	—/E*3	—/E*3	—/E*3
0x8000_0000	0x9FFF_FFFF	OSPI0BI	—	N/A	N/A	—	—	—	—	—
0xA000_0000	0xFFFF_FFFF	Reserved	E	N/A	N/A	E	E	E	E	E

Note: "E" : A bus error is occurred.
"N/A" : Transfer does not occur.
"—" : A bus error has not occurred. Even if there is reserved area, a bus error has not occurred.
Do not access the reserved area in Code MRAM, Extra MRAM, SRAM0, SRAM1, SRAM2, and SRAM3. If accessed, a slave TrustZone Filter error may occur.

Note 1. The product with Code MRAM size of 512 KB is "E", others is "—".
Note 2. The product with Code MRAM size of 512 KB or 768 KB is "E", others is "—".
Note 3. The product without SIP-FLASH is "—", and the product with SIP-FLASH is "E".

15.7.4 Timeout

As for some peripheral module, a timeout error can occur in the module-stop function.

When there is no response from the slave for a certain time period, a timeout error is detected.

Timeout error is returned to the request master IP by Error response of the AHB-Lite protocol.

15.8 References

1. ARM Limited, *Arm v8-M Architecture Reference Manual*(ARM DDI0553B.g)
2. ARM Limited, *Arm® Cortex®-M85 Processor Integration and Implementation Manual Revision:r0p2* (ARM 101925_0002_05_en)
3. ARM Limited, *Arm® Cortex®-M85 Processor Technical Reference Manual Revision:r0p2* (ARM 101924_0002_05_en)
4. ARM Limited, *Arm® Cortex®-M85 Processor User Guide Reference Material Revision:r0p2* (ARM 101927_0002_05_en)
5. ARM Limited, *Arm® Cortex®-M33 Processor Integration and Implementation Manual Revision:r0p2* (ARM 100323_0002_00_en)
6. ARM Limited, *Arm® Cortex®-M33Processor Technical Reference Manual Revision:r0p4* (ARM 100230_0004_00_en)
7. ARM Limited, *Arm® Cortex®-M33Processor User Guide Reference Material Revision:r0p2* (ARM 100234_0002_00_en)
8. ARM Limited, *ARM AMBA 5 AHB Protocol Specification AHB5, AHB-Lite* (ARM IHI 0033B.b)
9. ARM Limited, *ARM AMBA AXI and ACE Protocol Specification AXI3, AXI4, and AXI4-Lite*, ACE and ACE-Lite (ARM IHI 0022H.c)
10. ARM Limited, *ARM AMBA APB Protocol Specification Version: 2.0* (ARM IHI 0024D)

16. Memory Protection Unit (MPU)

16.1 Overview

EDMAC in this chapter refers to the GWCA function of ESWM.

All bus masters have Memory Protection Units (MPUs).

Table 16.1 lists the MPU specifications, and Table 16.2 shows the behavior on detection of each MPU error.

Table 16.1 MPU specifications

Classification	Module/Function	Specifications
Memory protection	Arm MPU	Memory protection function for the CPU: ● CPU0: Secure MPU 8 regions and Non-secure MPU 8 regions ● CPU1: Secure MPU 8 regions and Non-secure MPU 8 regions
	Bus master MPU	Memory protection function for each bus master except for the CPU: ● DMAC0 (DMAC/DTC0): 8 regions ● DMAC1 (DMAC/DTC1): 8 regions ● EDMAC (Ether-DMAC): 5 regions ● CEU: 2 regions

Table 16.2 Behavior on MPU error detection

MPU type	Access permissions setting	Boundary address setting minimum unit	Error response for the MPU error notification	Bus access at error detection	Hold the information of error access
Arm MPU	Read access Write access Execution	32 bytes	Supported*1	● Incorrectly write access ● Incorrectly read access	Hold in CPU
Bus master MPU	Read access Write access Privileged access (DMAC/DTC only)	DMAC0/1: 32 bytes EDMAC: 32 bytes CEU: 4 KB	Supported	● Write access ignored ● Read access is read as 0	Hold

Note 1. A privileged DAP request through the unprivileged debug extension mechanism is demoted to an unprivileged access and is subject to MPU checks. Both privileged and unprivileged requests are subject to MPU checks.

For information on error access for the Arm MPU, see section 16.4. References. For information on error access for other MPUs, see section 16.3.1. Register Descriptions and section 15.7. Bus Error Monitoring Section.

16.2 Arm MPU

The Arm MPU provides full support for:

- CPU0, 1: 8 secure regions and 8 non-secure regions
- Access permissions
- Export of memory attributes to the system

Arm MPU mismatches and permission violations invoke the programmable-priority MemManage fault (Hard Fault) handler. For details, see section 16.4. References.

16.3 Bus Master MPU

The bus master MPU monitors the addresses accessed by the bus masters in the entire address space (0x0000_0000 to 0xFFFF_FFFF). The access control information can be set up to:

- 8 regions in DMAC0
- 8 regions in DMAC1
- 5 regions in EDMAC
- 2 regions in CEU

The bus master MPU monitors access to each region based on these settings.

If accesses violate the access permissions that are configured in the bus master MPU or crossing the boundaries of a valid MPU region with burst access, the bus master MPU returns an error response. For information on error access, see section 16.3.1. Register Descriptions and section 15.7.2. Operations When a Bus Error Occurs.

Table 16.3 lists the specifications of the bus master MPU.

Table 16.3 Bus master MPU specifications

Parameter	Description
Master groups	(Group name: Corresponding bus master) ● DMAC0: DMAC/DTC0 ● DMAC1: DMAC/DTC1 ● EDMAC: Ether-DMAC ● CEU: CEU
Protected regions	0x0000_0000 to 0xFFFF_FFFF
Number of regions	● DMAC0: 8 regions ● DMAC1: 8 regions ● EDMAC: 5 regions ● CEU: 2 regions
Address specification for individual regions	● Region start and end addresses are configurable
Enable or disable setting for memory protection in individual regions	● Settings enabled or disabled for the associated region
Access-control settings for individual regions	● Permission to read and write ● Permission to unprivileged access
Operation on error detection	● Reset or error response
Register protection	● Register can be protected from illegal writes

16.3.1 Register Descriptions

Register writes processing should be performed after stopping the corresponding master group bus access.

SA (Secure Attribution) is set in each register and the SA is used to determine whether the target register is a secure register or a non-secure register.

16.3.1.1 MMPUSARA : Master Memory Protection Unit Security Attribution Register A

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x130

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

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

Bit	Symbol	Function	R/W
7:0	MMPUASA7 to MMPUASA0	MMPUAn Security Attribution (n = 0 to 7) 0: Secure 1: Non-secure	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
23:16	MMPUASA23 to MMPUASA16	MMPUAn Security Attribution (n = 16 to 23) 0: Secure 1: Non-secure	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1
P-TYPE-1

MMPUASA bits (MMPUAn Security Attribution (n = 0 to 7))

These bits specify security attributes of the Bus Master MPU Region Setting register for DMAC0. The target registers are as follows:

- MMPUSDMAC00n
- MMPUEDMAC00n
- MMPUACDMAC00n

MMPUASA bits (MMPUAn Security Attribution (n = 16 to 23))

These bits specify security attributes of the Bus Master MPU Region Setting register for DMAC1. The target registers are as follows:

- MMPUSDMAC10y (y = n − 16)
- MMPUEDMAC10y (y = n − 16)
- MMPUACDMAC10y (y = n − 16)

16.3.1.2 MMPUSARB : Master Memory Protection Unit Security Attribution Register B

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x134

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	MMPU BSA8	—	—	—	—	—	—	MMPU BSA1	MMPU BSA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MMPUBSA0	MMPUB0 Security Attribution 0: Secure 1: Non-secure	R/W
1	MMPUBSA1	MMPUB1 Security Attribution 0: Secure 1: Non-secure	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	MMPUBSA8	MMPUB8 Security Attribution 0: Secure 1: Non-secure	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1
P-TYPE-1

MMPUBSA0 bit (MMPUB0 Security Attribution)

This bit specifies the security attributes of the DMAC0 MPU Enable Setting register. The target registers are as follows:

- MMPUENDMAC0
- MMPUENPTDMAC0

MMPUBSA1 bit (MMPUB1 Security Attribution)

This bit specifies the security attributes of the DMAC1 MPU Enable Setting register. The target registers are as follows:

- MMPUENDMAC1
- MMPUENPTDMAC1

MMPUBSA8 bit (MMPUB8 Security Attribution)

This bit specifies the security attributes of the MPU Operation After Detection Setting register. The target registers are as follows:

- MMPUOAD
- MMPUOADPT

16.3.1.3 MMPUSDMACmn : MMPU Start Address Register for DMAC (m = 0, 1, n = 00 to 07)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0204 + 0x0010 × n (DMAC0n)
0x0404 + 0x0010 × n (DMAC1n)

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

Bit	Symbol	Function	R/W
31:0	n/a	Region start address register Address where the region starts, for use in region determination. The starting address of the MPU area must be set in the range of 0x0000_0000 to 0xFFFF_FFE0. Writing is ignored for bit 0 to bit 4. These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3
P-TYPE-2

The MMPUSDMACmn (m = 0, 1, n = 00 to 07) register specifies the start address where the region starts.

This register requires word access. Byte access and halfword access are prohibited. When byte access and halfword access are executed, operation is not guaranteed.

Regions set by MMPUSDMACmn (m = 0, 1, n = 00 to 07), MMPUEDMACmn (m = 0, 1, n = 00 to 07) and MMPUACDMACmn (m = 0, 1, n = 00 to 07) registers, can be set for a secure access or a non-secure access with the MMPUSARA register. If the corresponding MMPUSARA.MMPUASAn (n = 0 to 7, 16 to 23) bit is set to 1, it is only possible to use that region with a non-secure access. On the other hand, if the corresponding MMPUSARA.MMPUASAn (n = 0 to 7, 16 to 23) bit is set to 0, it is only possible to use that region with a secure access.

16.3.1.4 MMPUSEDMACn : MMPU Start Address Register for EDMAC (n = 0 to 4)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0604 + 0x010 × n

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

Bit	Symbol	Function	R/W
31:0	n/a	Region start address register for EDMAC Address where the region starts, for use in region determination. The starting address of the MPU area must be set in the range of 0x0000_0000 to 0xFFFF_FFE0. Writing is ignored for bit 0 to bit 4. These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3
P-TYPE-2

The MMPUSEDMACn (n = 0 to 4) register specifies the start address where the region starts.

This register requires word access. Byte access and halfword access are prohibited. When byte access and halfword access are executed, operation is not guaranteed.

16.3.1.5 MMPUSCEUn : MMPU Start Address Register for CEU (n = 0, 1)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0E04+ 0x010 × n

Bit position: 31 … 0

Bit field:

Value after reset: x x x x x x x x x x x x x x x x x x x x 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	Region start address register for CEU Address where the region starts, for use in region determination. The starting address of the MPU area must be set in the range of 0x0000_0000 to 0xFFFF_F000. Writing is ignored for bit 0 to bit 11. These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3
P-TYPE-2

The MMPUSCEUn (n = 0, 1) register specifies the start address where the region starts.

This register requires word access. Byte access and halfword access are prohibited. When byte access and halfword access are executed, operation is not guaranteed.

16.3.1.6 MMPUEDMACmn : MMPU End Address Register for DMAC (m = 0, 1, n = 00 to 07)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0208 + 0x0010 × n (DMAC0n)
0x0408 + 0x0010 × n (DMAC1n)

Bit position: 31 … 0

Bit field:

Value after reset: x x x x x x x x x x x x x x x x x x x x x x x x x x x 1 1 1 1 1

Bit	Symbol	Function	R/W
31:0	n/a	Region end address register Address where the region end, for use in region determination. The ending address of the MPU area must be set in the range of 0x0000_001F to 0xFFFF_FFFF. Writing is ignored for bit 0 to bit 4. These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-3
P-TYPE-2

The MMPUEDMACmn (m = 0, 1, n = 00 to 07) register specifies the end address where the region ends.

This register requires word access. Byte access and halfword access are prohibited. When byte access and halfword access are executed, operation is not guaranteed.

16.3.1.7 MMPUEEDMACn : MMPU End Address Register for EDMAC (n = 0 to 4)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0608 + 0x0010 × n

Bit position: 31 … 0

Bit field:

Value after reset: x x x x x x x x x x x x x x x x x x x x x x x x x x x 1 1 1 1 1

Bit	Symbol	Function	R/W
31:0	n/a	Region end address register for EDMAC Address where the region ends, for use in region determination. The ending address of the MPU area must be set in the range of 0x0000_001F to 0xFFFF_FFFF. Writing is ignored for bit 0 to bit 4. These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-3
P-TYPE-2

The MMPUEEDMACn (n = 0 to 4) register specifies the end address where the region ends.

This register requires word access. Byte access and halfword access are prohibited. When byte access and halfword access are executed, operation is not guaranteed.

16.3.1.8 MMPUECEUn : MMPU End Address Register for CEU (n = 0, 1)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0E08 + 0x010 × n

Bit position: 31 … 0

Bit field:

Value after reset: x x x x x x x x x x x x x x x x x x x x 1 1 1 1 1 1 1 1 1 1 1 1

Bit	Symbol	Function	R/W
31:0	n/a	Region end address register for CEU Address where the region ends, for use in region determination. The ending address of the MPU area must be set in the range of 0x0000_0FFF to 0xFFFF_FFFF. Writing is ignored for bit 0 to bit 11. These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-3
P-TYPE-2

The MMPUECEUn (n = 0, 1) register specifies the end address where the region ends.

This register requires word access. Byte access and halfword access are prohibited. When byte access and halfword access are executed, operation is not guaranteed.

16.3.1.9 MMPUACDMACmn : MMPU Access Control Register for DMAC (m = 0, 1, n = 00 to 07)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0200 + 0x0010 × n (DMAC0n)
0x0400 + 0x0010 × n (DMAC1n)

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

Bit	Symbol	Function	R/W
0	ENABLE	Region Enable 0: DMAC region n unit is disabled 1: DMAC region n unit is enabled	R/W
1	RP	Read protection 0: Read permission 1: Read protection	R/W
2	WP	Write protection 0: Write permission 1: Write protection	R/W
3	PP	Privilege protection 0: Unprivileged access permission 1: Unprivileged access protection	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3
P-TYPE-2

ENABLE bit (Region Enable)

The ENABLE bit enables or disables the DMACm region n (m = 0, 1, n = 00 to 07) unit.

When the ENABLE bit is set to 1, the RP bit, the WP bit and the PP bit are enabled for permit or protect access to the region that is set in MMPUSDMACmn (m = 0, 1, n = 00 to 07) and MMPUEDMACmn (m = 0, 1, n = 00 to 07).

When the ENABLE bit is set to 0, the access to DMACm region n (m = 0, 1, n = 00 to 07) is not specified.

RP bit (Read protection)

The RP bit enables or disables read protection for DMACm region n (m = 0, 1, n = 00 to 07).

When the ENABLE bit is set to 1, the RP bit is available.

WP bit (Write protection)

The WP bit enables or disables write protection for DMACm region n (m = 0, 1, n = 00 to 07).

When the ENABLE bit is set to 1, the WP bit is available.

PP bit (Privilege protection)

The PP bit enables or disables unprivileged access protection for DMACm region n (m = 0, 1, n = 00 to 07).

When the ENABLE bit is set to 1, the PP bit is available.

16.3.1.10 MMPUACEDMACn : MMPU Access Control Register for EDMAC (n = 0 to 4)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0600 + 0x0010 × n

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

Bit	Symbol	Function	R/W
0	ENABLE	Region Enable 0: EDMAC region n unit is disabled. 1: EDMAC region n unit is enabled.	R/W
1	RP	Read protection 0: Read permission 1: Read protection	R/W
2	WP	Write protection 0: Write permission 1: Write protection	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3
P-TYPE-2

ENABLE bit (Region Enable)

The ENABLE bit enables or disables the EDMAC region n (n = 0 to 4) unit.

When the ENABLE bit is set to 1, the RP bit and the WP bit are enabled for permit or protect access to the region that is set in MMPUSEDMACn (n = 0 to 4) and MMPUEEDMACn (n = 0 to 4).

When the ENABLE bit is set to 0, the access to EDMAC region n (n = 0 to 4) is not specified.

RP bit (Read protection)

The RP bit enables or disables read protection for EDMAC region n (n = 0 to 4).

When the ENABLE bit is set to 1, the RP bit is available.

WP bit (Write protection)

The WP bit enables or disables write protection for EDMAC region n (n = 0 to 4).

When the ENABLE bit is set to 1, the WP bit is available.

16.3.1.11 MMPUACCEUn : MMPU Access Control Register for CEU (n = 0, 1)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0E00 + 0x010 × n

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

Bit	Symbol	Function	R/W
0	ENABLE	Region Enable 0: CEU region n unit is disabled 1: CEU region n unit is enabled	R/W

Bit	Symbol	Function	R/W
1	RP	Read protection 0: Read permission 1: Read protection	R/W
2	WP	Write protection 0: Write permission 1: Write protection	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3
P-TYPE-2

ENABLE bit (Region Enable)

The ENABLE bit enables or disables the CEU region n (n = 0, 1) unit.

When the ENABLE bit is set to 1, the RP bit and the WP bit are enabled for permit or protect access to the region that is set in MMPUSCEUn (n = 0, 1) and MMPUECEUn (n = 0, 1).

When the ENABLE bit is set to 0, the access to CEU region n (n = 0, 1) is not specified.

RP bit (Read protection)

The RP bit enables or disables read protection for CEU region n (n = 0, 1).

When the ENABLE bit is set to 1, the RP bit is available.

WP bit (Write protection)

The WP bit enables or disables write protection for CEU region n (n = 0, 1).

When the ENABLE bit is set to 1, the WP bit is available.

16.3.1.12 MMPUENDMACm : MMPU Enable Register for DMAC (m = 0, 1)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0100 (DMAC0n)
0x0300 (DMAC1n)

Bit position:	15							8								0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ENABLE	Bus master MPU of DMAC Enable 0: Bus master MPU of DMAC is disabled 1: Bus master MPU of DMAC is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the ENABLE bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access as byte-write access is prohibited.
When byte-write access is executed, operation is not guaranteed.

ENABLE bit (Bus master MPU of DMAC Enable)

The ENABLE bit enables or disables the bus master MPU function for DMACm (m = 0, 1).

When the ENABLE bit is set to 1, MMPUACDMACmn (m = 0, 1, n = 00 to 07) is available.

When the ENABLE bit is set to 0, MMPUACDMACmn (m = 0, 1, n = 00 to 07) is unavailable including permission for all regions.

The bus master MPU function of each master group sets the ENABLE bit.

When the ENABLE bit is set, write 0xA5 to the KEY[7:0] bits at the same time.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits are used to enable or disable writing to the ENABLE bit.

When writing the ENABLE bit, write 0xA5 to the KEY[7:0] bits at the same time.

When a value other than 0xA5 is written to the KEY[7:0] bits, the ENABLE bit is not updated. The KEY[7:0] bits are always read as 0x00.

16.3.1.13 MMPUENEDMAC : MMPU Enable Register for EDMAC

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0500

Bit position:	15							8								0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ENABLE	Bus master MPU of EDMAC Enable 0: Bus master MPU of EDMAC is disabled 1: Bus master MPU of EDMAC is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the ENABLE bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access as byte-write access is prohibited.
When byte-write access is executed, operation is not guaranteed.

ENABLE bit (Bus master MPU of EDMAC Enable)

The ENABLE bit enables or disables the bus master MPU function of each master group.

When the ENABLE bit is set to 1, MMPUACEDMACn (n = 0 to 3) is available.

When the ENABLE bit is set to 0, MMPUACEDMACn (n = 0 to 3) is unavailable, and permission for all regions.

The bus master MPU function of each master group sets the ENABLE bit.

When the ENABLE bit is set, write 0xA5 to the KEY[7:0] bits at the same time.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits are used to enable or disable writing to the ENABLE bit.

When writing the ENABLE bit, write 0xA5 to the KEY[7:0] bits at the same time.

When a value other than 0xA5 is written to the KEY[7:0] bits, the ENABLE bit is not updated. The KEY[7:0] bits are always read as 0x00.

16.3.1.14 MMPUENCEU : MMPU Enable Register for CEU

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0D00

Bit position:	15							8								0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ENABLE	Bus master MPU of CEU Enable 0: Bus master MPU of CEU is disabled 1: Bus master MPU of CEU is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the ENABLE bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access as byte-write access is prohibited.
When byte-write access is executed, operation is not guaranteed.

ENABLE bit (Bus master MPU of CEU Enable)

The ENABLE bit enables or disables the bus master MPU function of each master group.

When the ENABLE bit is set to 1, MMPUACCEUn (n = 0, 1) is available.

When the ENABLE bit is set to 0, MMPUACCEUn (n = 0, 1) is unavailable, and permission for all regions.

The bus master MPU function of each master group sets the ENABLE bit.

When the ENABLE bit is set, write 0xA5 to the KEY[7:0] bits at the same time.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits are used to enable or disable writing to the ENABLE bit.

When writing the ENABLE bit, write 0xA5 to the KEY[7:0] bits at the same time.

When a value other than 0xA5 is written to the KEY[7:0] bits, the ENABLE bit is not updated. The KEY[7:0] bits are always read as 0x00.

16.3.1.15 MMPUENPTDMACm : MMPU Enable Protect Register for DMAC m (m = 0, 1)

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0104 (DMAC0n)
0x0304 (DMAC1n)

Bit position:	15							8								0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: MMPUENDMACm register write is possible. 1: MMPUENDMACm register write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the PROTECT bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access as byte-write access is prohibited.
When byte-write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit controls protection of MMPUENDMACm (m = 0, 1) register.

When the PROTECT bit is set, write 0xA5 to the KEY[7:0] bits at the same time.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits are used to enable or disable writing to the PROTECT bit.

When writing the PROTECT bit, write 0xA5 to the KEY[7:0] bits at the same time.

When a value other than 0xA5 is written to the KEY[7:0] bits, the ENABLE bit is not updated. The KEY[7:0] bits are always read as 0x00.

16.3.1.16 MMPUENPTEDMAC : MMPU Enable Protect Register for EDMAC

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0504

Bit position:	15							8								0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: MMPUENEDMAC register write is possible. 1: MMPUENEDMAC register write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the PROTECT bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access as byte-write access is prohibited.
When byte-write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit controls protection of MMPUENEDMAC register.

When the PROTECT bit is set, write 0xA5 to the KEY[7:0] bits at the same time.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits are used to enable or disable writing to the PROTECT bit.

When writing the PROTECT bit, write 0xA5 to the KEY[7:0] bits at the same time.

When a value other than 0xA5 is written to the KEY[7:0] bits, the ENABLE bit is not updated. The KEY[7:0] bits are always read as 0x00.

16.3.1.17 MMPUENPTCEU : MMPU Enable Protect Register for CEU

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0D04

Bit position:	15							8								0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: MMPUENCEU register write is possible. 1: MMPUENCEU register write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the PROTECT bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access as byte-write access is prohibited.
When byte-write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit controls protection of MMPUENCEU register.

When the PROTECT bit is set, write 0xA5 to the KEY[7:0] bits at the same time.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits are used to enable or disable writing to the PROTECT bit.

When writing the PROTECT bit, write 0xA5 to the KEY[7:0] bits at the same time.

When a value other than 0xA5 is written to the KEY[7:0] bits, the ENABLE bit is not updated. The KEY[7:0] bits are always read as 0x00.

16.3.1.18 MMPURPTDMACm : MMPU Regions Protect Register for DMAC Non-secure m (m = 0, 1)

Base address: RMPU_NS = 0x5000_0000

Offset address: 0x0108 (DMAC0n)
0x0308 (DMAC1n)

Bit position:	15							8								0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: Bus master MPU register for DMAC write is possible. 1: Bus master MPU register for DMAC write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the PROTECT bit.	W

Note: S-TYPE-7
P-TYPE-2

Note: It is necessary to write by halfword access as byte-write access is prohibited.
When byte-write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writing to the associated registers to be protected.

MMPURPTDMACm.PROTECT (m =0, 1) controls the following registers:

- MMPUSDMACmn (m =0, 1, n = 00 to 07) of non-secure
- MMPUEDMACmn (m =0, 1, n = 00 to 07) of non-secure
- MMPUACDMACmn (m =0, 1, n = 00 to 07) of non-secure

When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits. When a value other than 0xA5 is written to the KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.

16.3.1.19 MMPURPTDMAC_SECm : MMPU Regions Protect Register for DMAC Secure m (m = 0, 1)

Base address: RMPU = 0x4000_0000

Offset address: 0x010C (DMAC0n)
0x030C (DMAC1n)

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

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: Bus master MPU register for DMAC Secure write is possible. 1: Bus master MPU register for DMAC Secure write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the PROTECT bit.	W

Note: S-TYPE-6
P-TYPE-2

Note: It is necessary to write by halfword access as byte-write access is prohibited.
When byte-write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writing to the associated registers to be protected.

MMPURPTDMAC_SECm.PROTECT (m = 0, 1) controls the following registers:

- MMPUSDMACmn (m = 0, 1, n = 00 to 07) of secure
- MMPUEDMACmn (m = 0, 1, n = 00 to 07) of secure
- MMPUACDMACmn (m = 0, 1, n = 00 to 07) of secure

When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits. When a value other than 0xA5 is written to the KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.

16.3.1.20 MMPURPTEDMAC : MMPU Regions Protect Register for EDMAC

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0508

Bit position:	15							8								0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: Bus master MPU register for EDMAC write is possible. 1: Bus master MPU register for EDMAC write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the PROTECT bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access as byte-write access is prohibited.
When byte-write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writing to the associated registers to be protected.

MMPURPTEDMAC.PROTECT controls the following registers:

- MMPUSEDMACn (n = 0 to 4)
- MMPUEEDMACn (n = 0 to 4)
- MMPUACEDMACn (n = 0 to 4)

When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits. When a value other than 0xA5 is written to the KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.

16.3.1.21 MMPURPTCEU : MMPU Regions Protect Register for CEU

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0D08

Bit position:	15							8								0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: Bus master MPU register for CEU write is possible. 1: Bus master MPU register for CEU write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the PROTECT bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access as byte-write access is prohibited.
When byte-write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writing to the associated registers to be protected.

MMPURPTCEU.PROTECT controls the following registers:

- MMPUSCEUn (n = 0, 1)
- MMPUECEUn (n = 0, 1)
- MMPUACCEUn (n = 0, 1)

When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits. When a value other than 0xA5 is written to the KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.

16.3.1.22 MMPUOAD : MMPU Operation After Detection Register

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0000

Bit position:	15							8								0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	OAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OAD	Operation after detection 0: IRQ 1: Reset	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the OAD bit.	R/W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access as byte-write access is prohibited.
When byte-write access is executed, operation is not guaranteed.

OAD bit (Operation after detection)

The OAD bit specifies operation when the access violation is detected.

When OAD bit is 0, error response is returned to the bus master, and if BUSIRQEN<Master Name>.EN = 1, IRQ request is generated. For details, see section 15.3.31. BUSIRQEN<Master Name> : BUS Error Interrupt Enable Register . Interrupts due to bus errors can also be used as NMI. For details, see the specifications of the ICU.NMIER register.

When OAD bit is 1, reset request is generated.

When writing to the OAD bit, write 0xA5 simultaneously to the KEY[7:0] bits using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the OAD bit. When writing to the OAD bit, write 0xA5 simultaneously to the KEY[7:0] bits. When other values are written, the OAD bit is not updated. The KEY[7:0] bits always read as 0x00.

16.3.1.23 MMPUOADPT : MMPU Operation After Detection Protect Register

Base address: RMPU = 0x4000_0000
RMPU_NS = 0x5000_0000

Offset address: 0x0004

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

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: MMPUOAD register writing is possible. 1: MMPUOAD register writing is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits enable or disable writing to the PROTECT bit.	W

Note: S-TYPE-3
P-TYPE-2

Note: It is necessary to write by halfword access as byte-write access is prohibited.
When byte-write access is executed, operation is not guaranteed.

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writing to the associated registers to be protected.

MMPUOAD.PROTECT controls the following registers:

- MMPUOAD

When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits. When other values are written to the KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.

16.3.2 Operation

16.3.2.1 Memory Protection

The bus master MPU monitors memory access using control settings made individually for the access control regions. If accesses violate the access permissions that are configured in bus master MPU, the bus master MPU generates a memory protection error.

Bus master MPU has master groups of DMAC0, DMAC1, EDMAC and CEU. The memory protection function checks the address of the bus for a unified master group, and blocks illegal access of the master group to the protected region by bus master MPU.

The region setting registers of the bus master MPU for DMAC can be set for secure master and non-secure master using the MMPUSARA register. The MPU region setting with MMPUSARA.MMPUASAn = 0 applies only to access from the secure master, and the MPU region setting with MMPUSARA.MMPUASAn = 1 applies only to access from the non-secure master. Figure 16.1 shows an example of access authorization for the secure master and non-secure master when both secure and non-secure MPU settings are configured.

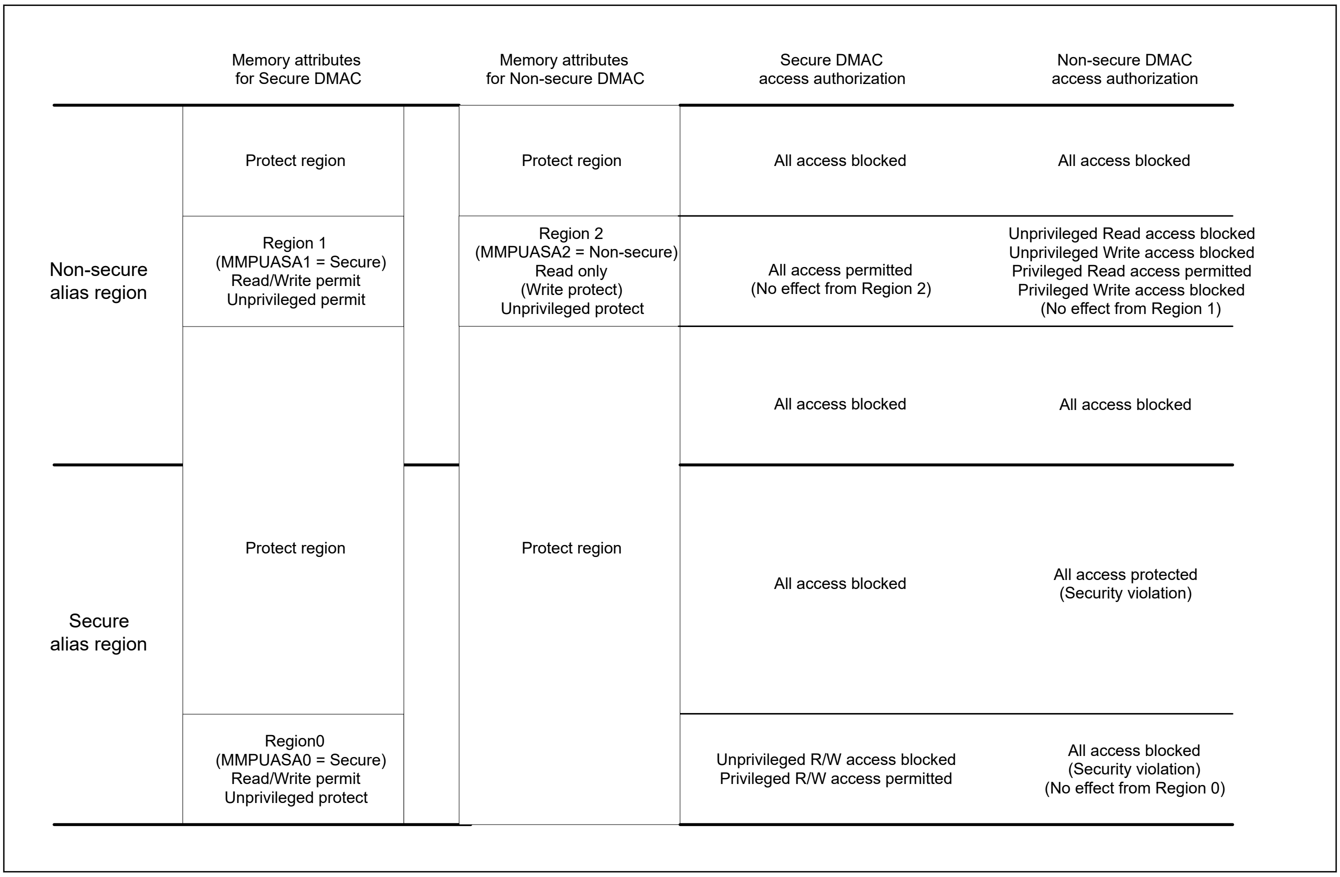


Figure 16.1 Example of access authority when both secure and non-secure regions of MPUs are set

Bus master MPU permit of all regions after reset. If MMPUENXXXX.ENABLE is 1 and there is no MPU region setting for the corresponding bus master, all regions are protected. (XXXX = Master Group name)

Figure 16.2 shows the access permission or protection by the overlapping bus master MPU regions.

Access control for the overlapping regions is as follows:

- The region is handled as a protected region when output of one or more region units is a protected region
- The region is handled as a protected region when output of all region units is outside of the regions
- Other cases are handled as permitted regions.

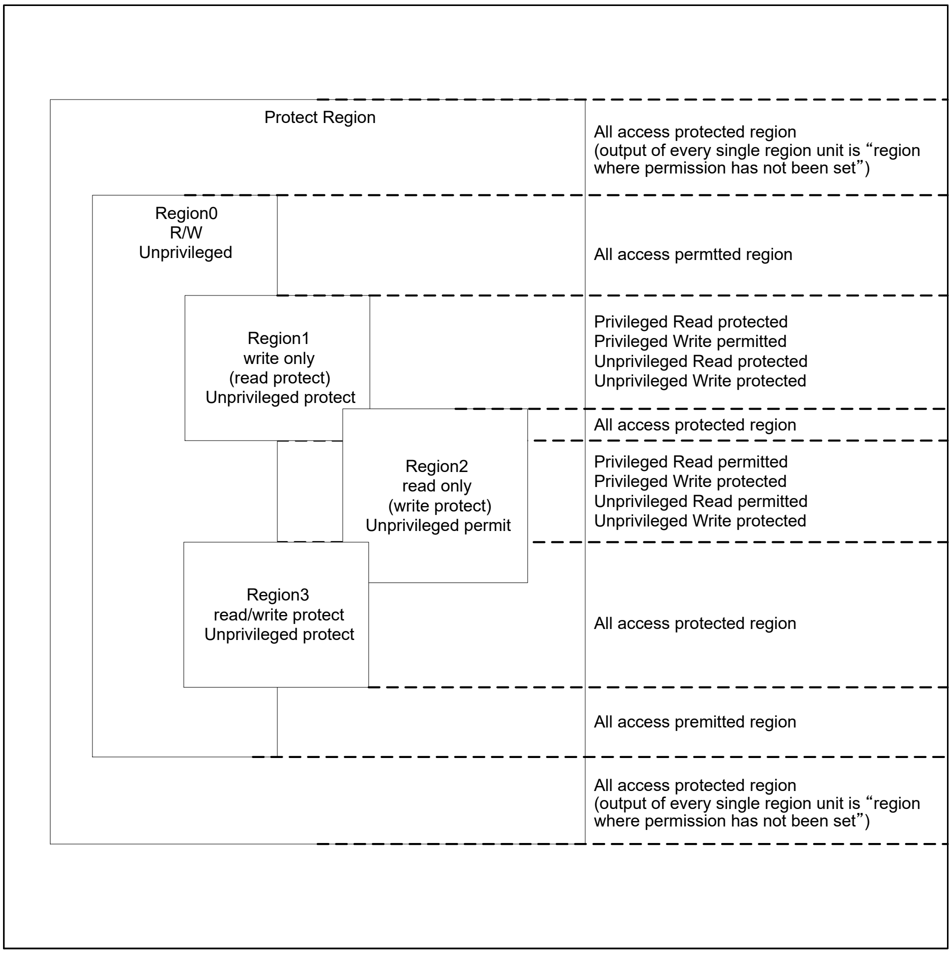


Figure 16.2 Access permission or protection by overlap of the bus master MPU regions

Figure 16.3 shows the register setting flow after reset. During this register setting, stop the bus master except the CPU.

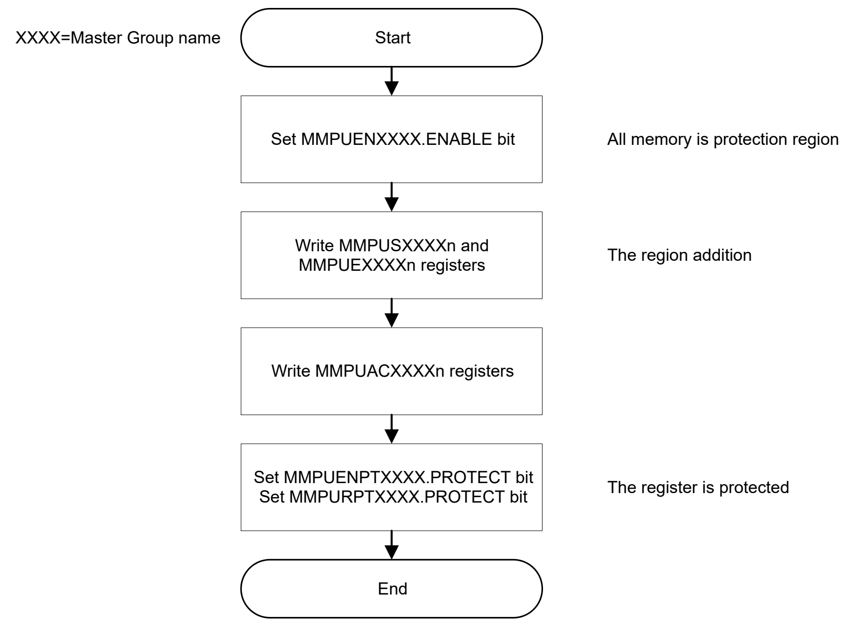


Figure 16.3 Register setting flow after reset

Figure 16.4 shows the register setting flow for adding regions. During this register setting, stop the master except the CPU.

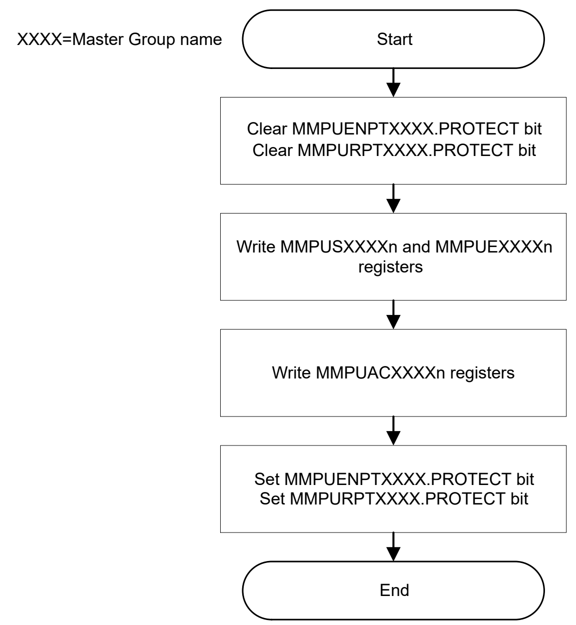


Figure 16.4 Register setting flow for region addition

16.3.2.2 Protecting the Registers

Registers related to the bus master MPU can be protected with the PROTECT bit as shown in Table 16.4.

Table 16.4 PROTECT bit and protect target registers (1 of 2)

PROTECT bit	Protect target registers
MMPUENPTDMACm.PROTECT	MMPUENDMACm (m = 0, 1)

Table 16.4 PROTECT bit and protect target registers (2 of 2)

PROTECT bit	Protect target registers
MMPUENPTEDMAC.PROTECT	MMPUENEDMAC
MMPUENPTCEU.PROTECT	MMPUENCEU
MMPURPTDMACm.PROTECT	The following registers set to Non-secure by MMPUSARA.MMPUASAn (n = 0 to 7, 16 to 23). MMPUSDMACmk (m = 0, 1, k = 00 to 07) MMPUEDMACmk (m = 0, 1, k = 00 to 07) MMPUACDMACmk (m = 0, 1, k = 00 to 07)
MMPURPTDMACm_SEC.PROTECT	The following registers set to Secure by MMPUSARA.MMPUASAn (n = 0 to 7, 16 to 23). MMPUSDMACmk (m = 0, 1, k = 00 to 07) MMPUEDMACmk (m = 0, 1, k = 00 to 07) MMPUACDMACmk (m = 0, 1, k = 00 to 07)
MMPURPTEDMAC.PROTECT	MMPUSEDMACn (n = 0 to 4) MMPUEEDMACn (n = 0 to 4) MMPUACEDMACn (n = 0 to 4)
MMPURPTCEU.PROTECT	MMPUSCEUn (n = 0, 1) MMPUECEUn (n = 0, 1) MMPUACCEUn (n = 0, 1)
MMPUOADPT.PROTECT	MMPUOAD.OAD

16.3.2.3 Memory Protection Error

If access to a protected region is detected, the bus master MPU generates an error. Set the MMPUOAD bit to select whether the error is reported as an IRQ or a reset.

For details of bus master MPU error, see section 15, Buses.

The IRQ status is indicated in ICU.IELSRn.IR. For details, see section 14, Interrupt Controller Unit (ICU). The reset status is indicated in SYSC.RSTSR1.BUSRF. For details, see section 6, Resets. Interrupts due to bus errors can also be used as NMI. For details of the specifications, see section 14.2.14. NMIER : Non-Maskable Interrupt Enable Register.

16.4 References

1. *ARM® v8-M Architecture Reference Manual* (ARM DDI0553B.g)
2. *ARM® Cortex®-M85 Processor Technical Reference Manual* (ARM 101924_0002_05_en)
3. *ARM® Cortex®-M85 Processor User Guide Reference Material* (ARM 101927_0002_05_en)
4. *ARM® Cortex®-M85 Processor Integration and Implementation Manual* (ARM 101925_0002_05_en)
5. *ARM® Cortex®-M33 Processor Technical Reference Manual* (100230_0004_00_en)
6. *ARM® Cortex®-M33 Processor User Guide Reference Material* (ARM 100234_0002_00_en)
7. *ARM® Cortex®-M33 Processor Integration and Implementation Manual* (ARM 100323_0002_00_en)

17. DMA Controller (DMAC)

17.1 Overview

The 8-channel direct memory access controller (DMAC) that can transfer data without intervention from the CPU. When a DMA transfer request is generated, the DMAC transfers data stored at the transfer source address to the transfer destination address.

Table 17.1 lists the DMAC specifications, and Figure 17.1 shows a block diagram of the DMAC.

Table 17.1 DMAC specifications (1 of 2)

Item		DMAC0 (for CPU0)	DMAC1 (for CPU1)
Number of channels		8 channels (DMAC0n (n = 0 to 7))	8 channels (DMAC1n (n = 0 to 7))
Transfer space		4 GB (0x00000000 to 0xFFFFFFFF excluding reserved areas)	
Maximum transfer volume		64 M data (Maximum number of transfers in block transfer mode: 1,024 data × 65,536 blocks)	
DMAC activation source		Selectable for each channel: ● Software trigger ● Interrupt requests from peripheral modules or trigger input from external interrupt input pins.*1	
Channel priority		Selectable from fixed priority or round-robin ● channel 0 > channel 1 > channel 2 > channel 3 … > channel 7 > DTC (channel 0: Highest) ● Round-robin	
Transfer data	Single data	Bit length: 8, 16, 32, 64 bits	
	Block size	Number of data: 1 to 1,024	
Transfer mode	Normal transfer mode	● One data transfer by one DMA transfer request ● Free running function (setting in which total number of data transfers is not specified) settable	
	Repeat transfer mode	● One data transfer by one DMA transfer request ● Program returns to the transfer start address on completion of the repeat size of data transfer specified for the transfer source or destination. ● Maximum settable repeat size: 1,024 ● Selectable free running function	
	Repeat-block transfer mode	● One block data transfer by one DMA transfer request ● Maximum settable block size: 1,024 ● Block transfer can be repeated ● Maximum settable repeat size: 64K ● Selectable free running function	
	Block transfer mode	● One block data transfer by one DMA transfer request ● Maximum settable block size: 1,024 data ● Selectable free running function	
Selective functions	Extended repeat area function	● Function in which data can be transferred by repeating the address values in the specified range with the upper bit values in the transfer address register fixed ● Area of 2 bytes to 128 MB separately settable as extended repeat area for transfer source and destination	
Processing on DMAC transfer error		● When DMAC transfer error occurs, the channel on which the error occurred is stopped. ● DMAC Event link setting register (DELSRn (n =0 to 7)) in the ICU is cleared. Clearing targets are selectable for each channel or all channels.	
Interrupt DMACmn_INT (m = 0, 1, n = 0 to 7)	Transfer end interrupt	Generated on completion of transferring data volume specified by the transfer counter.	
	Transfer escape end interrupt	● Generated when the repeat size of data transfer is completed. ● Generated when the source address extended repeat area overflows. ● Generated when the destination address extended repeat area overflows.	
Interrupt DMAm_TRANSERR (m = 0, 1)	Error response detection interrupt	● Generated when the DMAC transfer error occurs	

Table 17.1 DMAC specifications (2 of 2)

Item	DMAC0 (for CPU0)	DMAC1 (for CPU1)
Event link activation (DMACmn_INT (m = 0, 1, n = 0 to 7))	Generated after each data transfer (for block transfer, after each block is transferred).	
Module-stop function	Module-stop state can be set to reduce power consumption.	
TrustZone Filter	Security and Privilege can be set for each channel.	

Note: Security attribution Register of DMAC channel is described in ICU.ICUSARC

Note 1. For details on DMAC activation sources, see Table 14.5 in section 14, Interrupt Controller Unit (ICU).

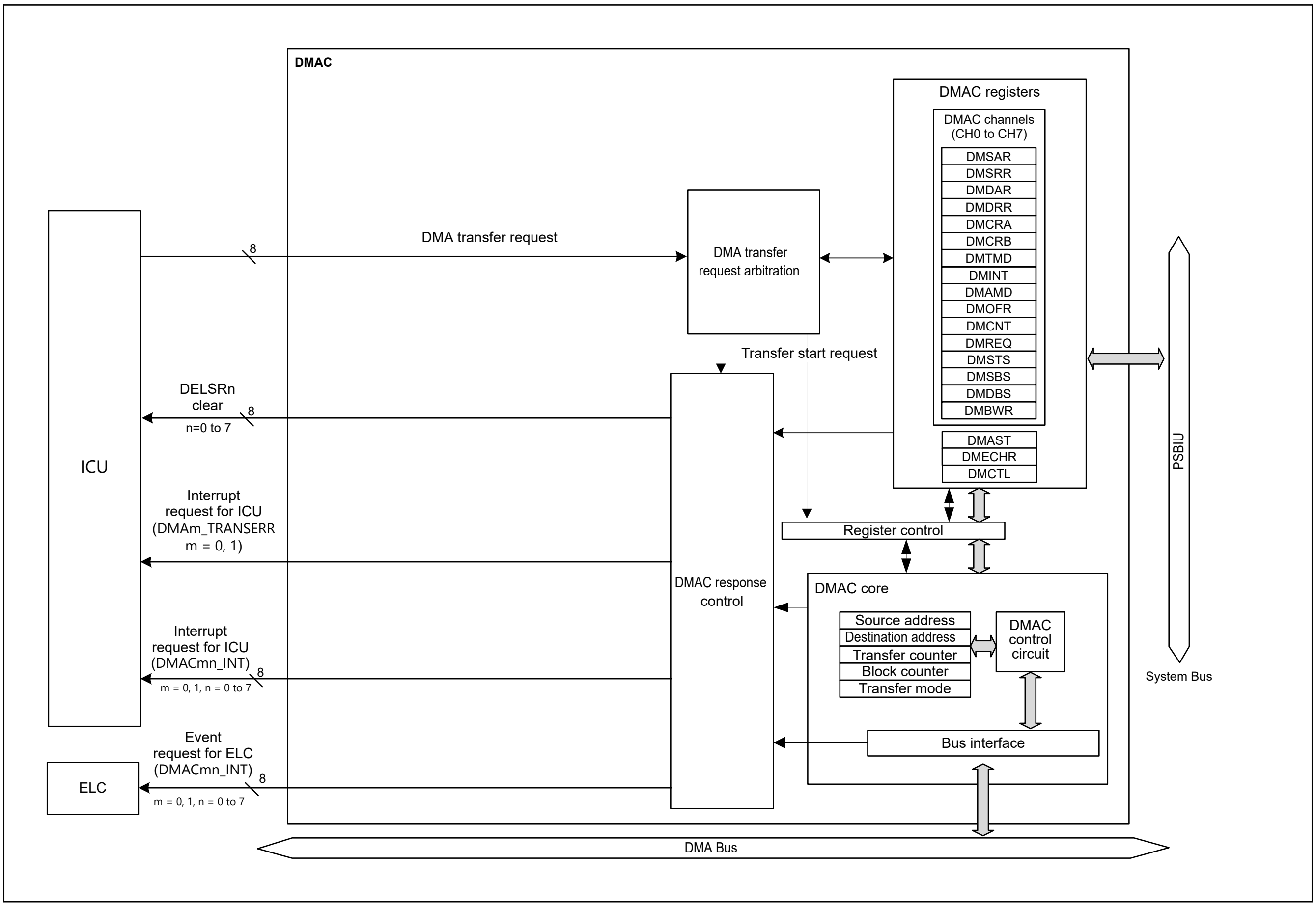


Figure 17.1 DMAC block diagram

17.2 Register Descriptions

The DMAC0 and DMAC1 registers have the same base address. DMAC0 is dedicated to CPU0 and can only be accessed from CPU0. DMAC1 is dedicated to CPU1 and can only be accessed from CPU1.

17.2.1 DMACCHSAR : DMA Channel Security Attribution Register

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x1A0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	SADMAC107	SADMAC106	SADMAC105	SADMAC104	SADMAC103	SADMAC102	SADMAC101	SADMAC100
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	SADMAC007	SADMAC006	SADMAC005	SADMAC004	SADMAC003	SADMAC002	SADMAC001	SADMAC000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	SADMAC007 to SADMAC000	Security attributes of output and registers for DMAC0 channel 0: Secure 1: Non-secure	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
23:16	SADMAC107 to SADMAC100	Security attributes of output and registers for DMAC1 channel 0: Secure 1: Non-secure	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

SADMAC00n (n = 0 to 7)

Security attributes registers for DMAC0 channel.

This bit determines the security attribute of the output as the master for each DMAC0 channel.

The controlled channel of DMAC0 registers are shown below. For details, see section 17.3.13. Channel Security.

- CPSCU.DMACCHPAR.PADMAC00n
- ICU0.DELSR channel n
- DMAC0.DMSAR channel n
- DMAC0.DMSRR channel n
- DMAC0.DMDAR channel n
- DMAC0.DMDRR channel n
- DMAC0.DMCRA channel n
- DMAC0.DMCRB channel n
- DMAC0.DMTMD channel n
- DMAC0.DMINT channel n
- DMAC0.DMAMD channel n
- DMAC0.DMOFR channel n
- DMAC0.DMCNT channel n
- DMAC0.DMREQ channel n
- DMAC0.DMSTS channel n
- DMAC0.DMSBS channel n
- DMAC0.DMDBS channel n
- DMAC0.DMBWR channel n

SADMAC10n (n = 0 to 7)

Security attributes registers for DMAC1 channel.

This bit determines the security attribute of the output as the master for each DMAC1 channel.

The controlled channel of DMAC1 registers are shown below. For details, see section 17.3.13. Channel Security.

- CPSCU.DMACCHPAR.PADMAC10n
- ICU1.DELSR channel n
- DMAC1.DMSAR channel n
- DMAC1.DMSRR channel n
- DMAC1.DMDAR channel n
- DMAC1.DMDRR channel n
- DMAC1.DMCRA channel n
- DMAC1.DMCRB channel n
- DMAC1.DMTMD channel n
- DMAC1.DMINT channel n
- DMAC1.DMAMD channel n
- DMAC1.DMOFR channel n
- DMAC1.DMCNT channel n
- DMAC1.DMREQ channel n
- DMAC1.DMSTS channel n
- DMAC1.DMSBS channel n
- DMAC1.DMDBS channel n
- DMAC1.DMBWR channel n

17.2.2 DMACCHPAR : DMA Channel Privilege Attribution Register

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x1F0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	PADMAC107	PADMAC106	PADMAC105	PADMAC104	PADMAC103	PADMAC102	PADMAC101	PADMAC100
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	PADMAC007	PADMAC006	PADMAC005	PADMAC004	PADMAC003	PADMAC002	PADMAC001	PADMAC000
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	PADMAC007 to PADMAC000	Privilege attributes of outputs and registers for DMAC0 channel 0: Privileged. 1: Unprivileged.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
23:16	PADMAC107 to PADMAC100	Privilege attributes of outputs and registers for DMAC1 channel 0: Privileged. 1: Unprivileged.	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-2, P-TYPE-1

PADMAC00n (n = 0 to 7)

Privilege attributes of output and registers for DMAC0 channel.

This bit determines the Privilege attribute of the output as the master for each DMAC0 channel. It also determines the privileged attributes of DMAC0 registers for each channel. For details, see section 17.3.14. Channel Privilege.

The controlled the channel of DMAC0 registers are shown below.

- DMAC0.DMSAR channel n
- DMAC0.DMSRR channel n
- DMAC0.DMDAR channel n
- DMAC0.DMDRR channel n
- DMAC0.DMCRA channel n
- DMAC0.DMCRB channel n
- DMAC0.DMTMD channel n
- DMAC0.DMINT channel n
- DMAC0.DMAMD channel n
- DMAC0.DMOFR channel n
- DMAC0.DMCNT channel n
- DMAC0.DMREQ channel n
- DMAC0.DMSTS channel n
- DMAC0.DMSBS channel n
- DMAC0.DMDBS channel n
- DMAC0.DMBWR channel n

PADMAC10n (n = 0 to 7)

Privilege attributes of output and registers for DMAC1 channel.

This bit determines the Privilege attribute of the output as the master for each DMAC1 channel. It also determines the privileged attributes of DMAC1 registers for each channel. For details, see section 17.3.14. Channel Privilege.

The controlled the channel of DMAC1 registers are shown below.

- DMAC1.DMSAR channel n
- DMAC1.DMSRR channel n
- DMAC1.DMDAR channel n
- DMAC1.DMDRR channel n
- DMAC1.DMCRA channel n
- DMAC1.DMCRB channel n
- DMAC1.DMTMD channel n
- DMAC1.DMINT channel n
- DMAC1.DMAMD channel n
- DMAC1.DMOFR channel n
- DMAC1.DMCNT channel n
- DMAC1.DMREQ channel n
- DMAC1.DMSTS channel n
- DMAC1.DMSBS channel n
- DMAC1.DMDBS channel n
- DMAC1.DMBWR channel n

17.2.3 DMACSAR : DMAC Controller Security Attribution Register

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x34

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMASTSA1
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMASTSA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DMASTSA0	DMAC0 DMAST Security Attribution 0: Secure 1: Non-secure	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W
16	DMASTSA1	DMAC1 DMAST Security Attribution 0: Secure 1: Non-secure	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

For DMAC, security attribution is set for each channel. However, this register only sets the DMAST register security attribution.

DMASTSA0 bit (DMAC0 DMAST Security Attribution)

This bit specifies security attribution of the registers for DMAC0.DMAST and DMAC0.DMCTL. Do not write to the DMASTSA0 bit while DMAC0 transfer is enabled or a bus master is writing to the DMAC0 registers.

DMASTSA1 bit (DMAC1 DMAST Security Attribution)

This bit specifies security attribution of the registers for DMAC1.DMAST and DMAC1.DMCTL. Do not write to the DMASTSA1 bit while DMAC1 transfer is enabled or a bus master is writing to the DMAC1 registers.

17.2.4 DMSAR : DMA Source Address Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x00

Bit position:	31–0
Bit field:	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	Specifies the transfer source start address Setting range is 0x00000000 to 0xFFFFFFFF (4 GB).	R/W

Note: S-TYPE-3, P-TYPE-3

Set DMSAR while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer is disabled (DMCNT.DTE = 0).

Note: Address alignment in this register must match the Transfer Data Size value selected in the DMTMD.SZ bit.

17.2.5 DMSRR : DMA Source Reload Address Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x20

Bit position:	31																														0
Bit field:																															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Specifies the transfer source reload address 0x0000 0000 to 0xFFFF FFFF (4 GB)	R/W

Note: S-TYPE-3, P-TYPE-3

Set DMSRR while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer is disabled (DMCNT.DTE = 0).

DMSRR is used to store the start address of the buffer size set in DMSBS during repeat-blcok transfer mode. In repeat-block transfer mode, DMSAR reloads value of DMSRR after specified transfer finished.

In normal transfer mode, repeat transfer mode and block transfer mode, DMSRR is not used. The setting is invalid.

Note: Address alignment in this register must match the Transfer Data Size value selected in the DMTMD.SZ bit.

17.2.6 DMDAR : DMA Destination Address Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x04

Bit position:	31																														0
Bit field:																															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Specifies the transfer destination start address Setting range is 0x0000_0000 to 0xFFFF_FFFF (4 GB).	R/W

Note: S-TYPE-3, P-TYPE-3

Set DMDAR while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer is disabled (DMCNT.DTE = 0).

Note: Address alignment in this register must match the Transfer Data Size value selected in the DMTMD.SZ bits.

17.2.7 DMDRR : DMA Destination Reload Address Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x24

Bit position:	31																														0
Bit field:																															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Specifies the transfer destination reload address Setting range is 0x0000_0000 to 0xFFFF_FFFF (4 GB).	R/W

Note: S-TYPE-3, P-TYPE-3

Set DMDRR while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer is disabled (DMCNT.DTE = 0).

DMDRR is used to store the start address of the buffer size set in DMDBS during repeat-blcok transfer mode. In repeat-block transfer mode, DMDAR reloads the value of DMDRR after the specified transfer is finished.

In normal transfer mode, repeat transfer mode and block transfer mode, DMDRR is not used. The setting is invalid.

Note: Address alignment in this register must match the Transfer Data Size value selected in the DMTMD.SZ bits.

17.2.8 DMCRA : DMA Transfer Count Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	DMCRAH[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
15:0	DMCRAL[15:0]	Lower bits of transfer count Specifies the number of transfer operations.	R/W
25:16	DMCRAH[9:0]	Upper bits of transfer count Specifies the number of transfer operations.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Set DMCRA while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer is disabled (DMCNT.DTE = 0).

Set the same value for DMCRAH and DMCRAL in repeat transfer mode, block transfer mode, and repeat-block transfer mode. Bits 15 to 10 are fixed to 0 in repeat transfer mode, block transfer mode, and repeat-block transfer mode.

(1) Normal Transfer Mode (DMTMD.MD[1:0] = 00b)

DMCRAL functions as a 16-bit transfer counter.

The number of transfer operations is one when the setting is 0x0001, and 65,535 when it is 0xFFFF. The value is decremented by one each time data is transferred.

When the setting is 0x0000, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running function).

Free running function is not selected by DMTMD.TKP bit in normal transfer mode.

DMCRAH is not used in normal transfer mode. Write 0x0000 to DMCRAH.

(2) Repeat Transfer Mode (DMTMD.MD[1:0] = 01b)

DMCRAH specifies the repeat size and DMCRAL functions as a 10-bit transfer counter.

The number of transfer operations is one when the setting is 0x001, 1023 when it is 0x3FF, and 1024 when it is 0x000. In repeat transfer mode, a value in the range of 0x000 to 0x3FF (1 to 1024) can be set for DMCRAH and DMCRAL.

Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.

The value in DMCRAL is decremented by one each time data is transferred until it reaches 0x000, at which the value in DMCRAH is loaded into DMCRAL.

(3) Block Transfer Mode (DMTMD.MD[1:0] = 10b)

DMCRAH specifies the block size and DMCRAL functions as a 10-bit block size counter.

The block size is one when the setting is 0x001, 1023 when it is 0x3FF, and 1024 when it is 0x000. In block transfer mode, a value in the range of 0x000 to 0x3FF can be set for DMCRAH and DMCRAL.

Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.

The value in DMCRAL is decremented by one each time data is transferred until it reaches 0x000, at which the value in DMCRAH is loaded into DMCRAL.

(4) Repeat-Block Transfer Mode (DMTMD.MD[1:0] = 11b)

DMCRAH specifies the block size and DMCRAL functions as a 10-bit block size counter.

The block size is one when the setting is 0x001, 1023 when it is 0x3FF, and 1024 when it is 0x000. In repeat-block transfer mode, a value in the range of 0x000 to 0x3FF can be set for DMCRAH and DMCRAL.

Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.

The value in DMCRAL is decremented by one each time data is transferred until it reaches 0x000, at which the value in DMCRAH is loaded into DMCRAL.

17.2.9 DMCRB : DMA Block Transfer Count Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DMCRBH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
15:0	DMCRBL[15:0]	Functions as a number of block, repeat or repeat-block transfer counter. 0x0001 to 0xFFFF (1 to 65535) 0x0000 (65536)	R/W
31:16	DMCRBH[15:0]	Specifies the number of block, repeat or repeat-block transfer operations. 0x0001 to 0xFFFF (1 to 65535) 0x0000 (65536)	R/W

Note: S-TYPE-3, P-TYPE-3

Set DMCRB while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer is disabled (DMCNT.DTE = 0).

Set the same value for DMCRBH and DMCRBL in repeat transfer mode, block transfer mode and repeat-block transfer mode.

DMCRBH specifies the number of block, repeat and repeat-block transfer operations, and DMCRBL functions as a 16-bit the number of block counter in block, repeat, and repeat-block transfer mode, respectively.

The number of transfer operations is one when the setting is 0x0001, 65535 when it is 0xFFFF, and 65536 when it is 0x0000.

In repeat transfer mode, the value is decremented by one when the final data of one repeat size is transferred.

In block transfer mode and repeat-block transfer mode, the value is decremented by one when the final data of one block size is transferred.

In normal transfer mode, DMCRB is not used. The setting is invalid.

When DMTMD.TKP is 1 and the final data of one repeat size or one block size is transferred, DMCRBL reloads the value of DMCRBH automatically.

17.2.10 DMTMD : DMA Transfer Mode Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x10

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

Bit	Symbol	Function	R/W
1:0	DCTG[1:0]	Transfer Request Source Select 0 0: Software request 0 1: Hardware request*1 1 0: Setting prohibited 1 1: Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	SZ[1:0]	Transfer Data Size Select 0 0: 8 bits 0 1: 16 bits 1 0: 32 bits 1 1: 64 bits	R/W
10	TKP	Transfer Keeping 0: Transfer is stopped by completion of specified total number of transfer operations. 1: Transfer is not stopped by completion of specified total number of transfer operations (free-running).	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
13:12	DTS[1:0]	Repeat Area Select 0 0: The destination is specified as the repeat area or block area. 0 1: The source is specified as the repeat area or block area. 1 0: The repeat area or block area is not specified. 1 1: Setting prohibited.	R/W
15:14	MD[1:0]	Transfer Mode Select 0 0: Normal transfer 0 1: Repeat transfer 1 0: Block transfer 1 1: Repeat-block transfer	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. To select the DMAC activation source, use the ICU.DELSRn registers. For details on DMAC activation sources, see Table 14.5 in section 14, Interrupt Controller Unit (ICU).

Set DMTMD while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer is disabled (DMCNT.DTE = 0).

DTS[1:0] bits (Repeat Area Select)

DTS[1:0] select either the source or destination as the repeat area in repeat or block transfer mode. In normal or repeat-block transfer mode, setting these bits is invalid.

TKP bit (Transfer Keeping)

TKP selects either stopping transfer or keeping transfer by completion of specified total number of transfer operations in repeat, block or repeat-block transfer mode. In normal transfer mode, setting this bit is invalid.

17.2.11 DMINT : DMA Interrupt Setting Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x13

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DTIE	ESIE	RPTIE	SARIE	DARIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DARIE	Destination Address Extended Repeat Area Overflow Interrupt Enable 0: Disables an interrupt request for an extended repeat area overflow on the destination address. 1: Enables an interrupt request for an extended repeat area overflow on the destination address.	R/W
1	SARIE	Source Address Extended Repeat Area Overflow Interrupt Enable 0: Disables an interrupt request for an extended repeat area overflow on the source address. 1: Enables an interrupt request for an extended repeat area overflow on the source address.	R/W
2	RPTIE	Repeat Size End Interrupt Enable 0: Disables the repeat size end interrupt request. 1: Enables the repeat size end interrupt request.	R/W
3	ESIE	Transfer Escape End Interrupt Enable 0: Disables the transfer escape end interrupt request. 1: Enables the transfer escape end interrupt request.	R/W
4	DTIE	Transfer End Interrupt Enable 0: Disables the transfer end interrupt request. 1: Enables the transfer end interrupt request.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Set DMINT while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer is disabled (DMCNT.DTE = 0).

DARIE bit (Destination Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the destination address occurs while DARIE bit is set to 1, the DMCNT.DTE bit is cleared to 0. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that an interrupt by an extended repeat area overflow on the destination address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 in the DMCNT.DTE bit of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the destination address, this bit is ignored.

Do not set this bit to 1 in repeat-block transfer mode.

SARIE bit (Source Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the source address occurs while SARIE bit is set to 1, the DMCNT.DTE bit is cleared to 0. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that an interrupt by an extended repeat area overflow on the source address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 in the DMCNT.DTE bit of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the source address, this bit is ignored.

Do not set this bit to 1 in repeat-block transfer mode.

RPTIE bit (Repeat Size End Interrupt Enable)

When RPTIE bit is set to 1 in repeat transfer mode, the DMCNT.DTE bit is cleared to 0 after completion of a 1-repeat size data transfer. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DMTMD.DTS[1:0] bits are 10b (= repeat area or block area is not specified).

When this bit is set to 1 in block transfer mode, the DMCNT.DTE bit is cleared to 0 after completion of a 1-block data transfer in the same way as repeat transfer mode. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DMTMD.DTS[1:0] bits are 10b (= repeat area or block area is not specified).

Do not set this bit to 1 in repeat-block transfer mode.

ESIE bit (Transfer Escape End Interrupt Enable)

ESIE bit enables or disables the transfer escape end interrupt requests (repeat size end interrupt request and extended repeat area overflow interrupt request) that are generated during DMA transfer.

The transfer escape end interrupt is generated when the DMSTS.ESIF flag is set to 1 with this bit set to 1. The transfer escape end interrupt is cleared by clearing this bit or the DMSTS.ESIF flag to 0.

DTIE bit (Transfer End Interrupt Enable)

DTIE bit enables or disables the transfer end interrupt request to be generated on completion of a specified number of data transfers.

The transfer end interrupt is generated when the DMSTS.DTIF flag is set to 1 with this bit set to 1. The transfer end interrupt is cleared by clearing this bit or the DMSTS.DTIF flag to 0.

17.2.12 DMAMD : DMA Address Mode Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x14

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SM[1:0]		SADR	SARA[4:0]					DM[1:0]		DADR	DARA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	DARA[4:0]	Destination Address Extended Repeat Area Specifies the extended repeat area on the destination address. For details on the settings, see Table 17.2.	R/W
5	DADR	Destination Address Update Select After Reload 0: Only reloading. 1: Add index after reloading.	R/W
7:6	DM[1:0]	Destination Address Update Mode 0 0: Destination address is fixed. 0 1: Offset addition. 1 0: Destination address is incremented. 1 1: Destination address is decremented.	R/W
12:8	SARA[4:0]	Source Address Extended Repeat Area Specifies the extended repeat area on the source address. For details on the settings, see Table 17.2.	R/W
13	SADR	Source Address Update Select After Reload 0: Only reloading. 1: Add index after reloading.	R/W

Bit	Symbol	Function	R/W
15:14	SM[1:0]	Source Address Update Mode 0 0: Source address is fixed. 0 1: Offset addition. 1 0: Source address is incremented. 1 1: Source address is decremented.	R/W

Note: S-TYPE-3, P-TYPE-3

Set DMAMD while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer is disabled (DMCNT.DTE = 0).

DARA[4:0] bits (Destination Address Extended Repeat Area)

DARA[4:0] bits specify the extended repeat area on the destination address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 MB.

When the lower address overflows the extended repeat area by address increment, the start address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the end address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer destination, do not specify the extended repeat area on the destination address. When repeat transfer or block transfer is selected, and when DMTMD.DTS[1:0] = 00b (the transfer destination is specified as the repeat area or block area), write 00000b in the DARA[4:0] bits.

In repeat-block transfer mode, write 00000b in the DARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the DMINT.DARIE bit set to 1. Table 17.2 lists the settings and the corresponding extended repeat areas.

DADR bits (Destination Address Update Select After Reload)

In repeat-block transfer mode, this bit specifies the behavior of DMDAR after reloading DMDRR.

When this bit is set to 1, an index value ((DMDBSH-DMDBSL) × DataSize) is added to DMDAR after reloading DMDRR.

When this bit is set to 0, DMDAR only reloads DMDRR. This behavior is described in Table 17.13.

In normal, repeat or block transfer mode, this bit is ignored.

DM[1:0] bits (Destination Address Update Mode)

DM[1:0] bits select the mode of updating the destination address.

When increment is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, 10b, or 11b, the destination address is incremented by 1, 2, 4, or 8 respectively.

When decrement is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, 10b, or 11b, the destination address is decremented by 1, 2, 4, or 8 respectively.

When offset addition is selected, the offset specified by the DMOFR register is added to the address.

SARA[4:0] bits (Source Address Extended Repeat Area)

SARA[4:0] bits specify the extended repeat area on the source address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 MB.

When the lower address overflows the extended repeat area by address increment, the start address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the end address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer source, do not specify the extended repeat area on the source address. When repeat transfer or block transfer is selected, and when DMTMD.DTS[1:0] = 01b (the transfer source is specified as the repeat area or block area), write 00000b in the SARA[4:0] bits.

In repeat-block transfer mode, write 00000b in the SARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the DMINT.SARIE bit set to 1. Table 17.2 lists the settings and the corresponding extended repeat areas.

SADR bits (Source Address Update Select After Reload)

In repeat-block transfer mode, this bit specifies the behavior of DMSAR after reloading DMSRR.

When this bit is set to 1, an index value ((DMSBSH-DMSBSL) × DataSize) is added to DMSAR after reloading DMSRR.

When this bit is set to 0, DMSAR only reloads DMSRR. This behavior is described in Table 17.12.

In normal, repeat or block transfer mode, this bit is ignored.

SM[1:0] bits (Source Address Update Mode)

SM[1:0] bits select the mode of updating the source address.

When increment is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, 10b, or 11b, the source address is incremented by 1, 2, 4, or 8 respectively.

When decrement is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, 10b, or 11b, the source address is decremented by 1, 2, 4, or 8 respectively.

When offset addition is selected, the offset specified by the DMOFR register is added to the address.

Table 17.2 SARA[4:0] or DARA[4:0] settings and corresponding repeat areas

SARA[4:0] or DARA[4:0] settings	Extended repeat area
00000b	Not specified
00001b	2 bytes specified as extended repeat area by the lower 1 bit of the address
00010b	4 bytes specified as extended repeat area by the lower 2 bits of the address
00011b	8 bytes specified as extended repeat area by the lower 3 bits of the address
00100b	16 bytes specified as extended repeat area by the lower 4 bits of the address
00101b	32 bytes specified as extended repeat area by the lower 5 bits of the address
00110b	64 bytes specified as extended repeat area by the lower 6 bits of the address
00111b	128 bytes specified as extended repeat area by the lower 7 bits of the address
01000b	256 bytes specified as extended repeat area by the lower 8 bits of the address
01001b	512 bytes specified as extended repeat area by the lower 9 bits of the address
01010b	1 KB specified as extended repeat area by the lower 10 bits of the address
01011b	2 KB specified as extended repeat area by the lower 11 bits of the address
01100b	4 KB specified as extended repeat area by the lower 12 bits of the address
01101b	8 KB specified as extended repeat area by the lower 13 bits of the address
01110b	16 KB specified as extended repeat area by the lower 14 bits of the address
01111b	32 KB specified as extended repeat area by the lower 15 bits of the address
10000b	64 KB specified as extended repeat area by the lower 16 bits of the address
10001b	128 KB specified as extended repeat area by the lower 17 bits of the address
10010b	256 KB specified as extended repeat area by the lower 18 bits of the address
10011b	512 KB specified as extended repeat area by the lower 19 bits of the address
10100b	1 MB specified as extended repeat area by the lower 20 bits of the address
10101b	2 MB specified as extended repeat area by the lower 21 bits of the address
10110b	4 MB specified as extended repeat area by the lower 22 bits of the address
10111b	8 MB specified as extended repeat area by the lower 23 bits of the address
11000b	16 MB specified as extended repeat area by the lower 24 bits of the address
11001b	32 MB specified as extended repeat area by the lower 25 bits of the address
11010b	64 MB specified as extended repeat area by the lower 26 bits of the address
11011b	128 MB specified as extended repeat area by the lower 27 bits of the address
11100b to 11111b	Setting prohibited.

17.2.13 DMOFR : DMA Offset Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x18

Bit position: 31 ... 0

Bit field: (31:0)

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	Specifies the offset when offset addition is selected as the address update mode for transfer source or destination. 0x00000000 to 0x00FFFFFF (0 bytes to (16 M – 1) bytes) 0xFF000000 to 0xFFFFFFFF (–16 MB to –1 byte)	R/W

Note: S-TYPE-3, P-TYPE-3

Set DMOFR while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer is disabled (DMCNT.DTE = 0).

Write to this register while the DMAC operation is stopped or DMA transfer is disabled (not during data transfer).

Setting bits 31 to 25 is invalid; a value of bit 24 is extended to bits 31 to 25. Reading DMOFR returns the extended value.

In repeat-block transfer mode, the offset is not specified by DMOFR when offset addition is selected, write 0 to DMOFR.

17.2.14 DMCNT : DMA Transfer Enable Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x1C

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

Bit	Symbol	Function	R/W
0	DTE	DMA Transfer Enable 0: Disables DMA transfer. 1: Enables DMA transfer.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

DTE bit (DMA Transfer Enable)

When the DMAST.DMST bit is set to 1 (DMAC activation is enabled) and this bit is set to 1 (DMA transfer is enabled), DMA transfer can be started for the corresponding channel.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When 0 is written to this bit.
- When the specified total volume of data transfer is completed.
- When DMA transfer is stopped by the repeat size end interrupt.
- When DMA transfer is stopped by the extended repeat area overflow interrupt.
- When DMA transfer is stopped by the access error occurs. See section 17.5. Processing on DMA Transfer Error.

If the DTE of the corresponding channel is set to 0 during DMA transfer, no new transfer requests will be accepted after the transfer ends.

17.2.15 DMREQ : DMA Software Start Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x1D

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

Bit	Symbol	Function	R/W
0	SWREQ	DMA Software Start 0: DMA transfer is not requested. 1: DMA transfer is requested.	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	CLRS	DMA Software Start Bit Auto Clear Select 0: SWREQ bit is cleared after DMA transfer is started by software. 1: SWREQ bit is not cleared after DMA transfer is started by software.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

SWREQ bit (DMA Software Start)

When 1 is written to SWREQ bit, a DMA transfer request is generated. After DMA transfer is started in response to the request, this bit is cleared to 0 if the CLRS bit is set to 0. This bit is not cleared to 0 while the CLRS bit is set to 1. In this case, a DMA transfer request can be issued again after completion of a transfer.

Note that, however, setting this bit is valid and DMA transfer by software is enabled only when the DMTMD.DCTG[1:0] bits are set to 00b (DMAC activation source is software).

Setting this bit is invalid when the DMTMD.DCTG[1:0] bits are set to a value other than 00b.

To start DMA transfer by software with the CLRS bit being 0, ensure that the SWREQ bit is 0, and then write 1 to the SWREQ bit.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When a DMA transfer request by software is accepted and DMA transfer is started while the CLRS bit is set to 0 (the SWREQ bit is cleared after DMA transfer is started by software).
- When 0 is written to this bit.

When DMA transfer is stopped by the access error occurs. See section 17.5. Processing on DMA Transfer Error

CLRS bit (DMA Software Start Bit Auto Clear Select)

CLRS bit specifies whether to clear the SWREQ bit to 0 after DMA transfer is started in response to the DMA transfer request generated by setting the SWREQ bit to 1. With this bit set to 0, the SWREQ bit is cleared to 0 after DMA transfer is started. With this bit set to 1, the SWREQ bit is not cleared to 0. In this case, a DMA transfer request can be issued again after completion of a transfer.

17.2.16 DMSTS : DMA Status Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x1E

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

Bit	Symbol	Function	R/W
0	ESIF	Transfer Escape End Interrupt Flag 0: A transfer escape end interrupt has not been generated. 1: A transfer escape end interrupt has been generated.	R/W*1
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	DTIF	Transfer End Interrupt Flag 0: A transfer end interrupt has not been generated. 1: A transfer end interrupt has been generated.	R/W*1
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	ACT	DMAC Active Flag 0: DMAC is in the idle state. 1: DMAC is operating.	R

Note: S-TYPE-3, P-TYPE-3

Note 1. Only 0 can be written to clear the flag.

ESIF flag (Transfer Escape End Interrupt Flag)

This flag indicates that the transfer escape end interrupt has been generated.

[Setting conditions]

- When 1-repeat size data transfer is completed in repeat transfer mode with the DMINT.RPTIE bit set to 1.
- When 1-block data transfer is completed in block transfer mode with the DMINT.RPTIE bit set to 1.
- When an extended repeat area overflow on the source address occurs while the DMINT.SARIE bit is set to 1 and the DMAMD.SARA[4:0] bits are set to a value other than 00000b (extended repeat area is specified on the transfer source address).
- When an extended repeat area overflow on the destination address occurs while the DMINT.DARIE bit is set to 1 and the DMAMD.DARA[4:0] bits are set to a value other than 00000b (extended repeat area is specified on the transfer destination address).

[Clearing conditions]

- When 0 is written to this bit.
- When 1 is written to the DMCNT.DTE bit.

DTIF flag (Transfer End Interrupt Flag)

This flag indicates that the transfer end interrupt has been generated.

[Setting conditions]

- When the specified number of unit-transfers are completed in normal transfer mode (the value of DMCRAL becoming 0 on completion of transfer).
- When the specified number of repeat transfer operations are completed in repeat transfer mode (the value of DMCRBL becoming 0 on completion of transfer with DMTMD.TKP = 0 or the value of DMCRBL reloading DMCRBH with DMTMD.TKP = 1).
- When the specified number of blocks have been transferred in block transfer mode and repeat-block transfer mode (the value of DMCRBL becoming 0 on completion of transfer with DMTMD.TKP = 0 or the value of DMCRBL reloading DMCRBH with DMTMD.TKP = 1).

[Clearing conditions]

- When 0 is written to this bit.
- When 1 is written to the DMCNT.DTE bit.

ACT flag (DMAC Active Flag)

This flag indicates whether the DMAC is in the idle or active state.

[Setting condition]

- When the DMAC starts data transfer operation.

[Clearing condition]

- When data transfer in response to one transfer request is completed.

17.2.17 DMSBS : DMA Source Buffer Size Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x28

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DMSBSH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
15:0	DMSBSL[15:0]	Functions as data transfer counter in repeat-block transfer mode See Table 17.3 for available settings.	R/W
31:16	DMSBSH[15:0]	Specifies the repeat-area size in repeat-block transfer mode See Table 17.3 for available settings.	R/W

Note: S-TYPE-3, P-TYPE-3

Set DMSBS while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer is disabled (DMCNT.DTE = 0).

Set the same value for DMSBSH and DMSBSL in repeat-block transfer mode. Write 0x00000000 to DMSBS in normal, repeat and block transfer mode.

DMSBSH specifies buffer size and DMSBSL functions as a 16-bit buffer size counter in repeat-block transfer mode. In repeat-block transfer mode, source repeat area is specified by DMSBSH.

When address update mode is incremented address or decremented address, this register means the numbers of data of whole buffer. When address update mode is offset addition, this register means the numbers of data of an individual buffer. In offset addition, setting DMSBSH and DMSBSL to 0x0000 is prohibited. When final data of one buffer size is transferred, DMSBSL reloads value of DMSBSH. When address update mode is fixed address, this register is ignored. Table 17.3 shows the setting values of DMA Source Buffer Size Register corresponding to Transfer Data Size in Source Address Update Mode.

Table 17.3 Available setting for DMSBS register in repeat-block transfer mode (1 of 2)

Source address update mode (DMAMD.SM)	Transfer data size (DMTMD.SZ)	Available setting for DMSBSH and DMSBSL bits
Source address is fixed (SM = 00b)	Don't care	0x0000 (DMSBS is not used)

Table 17.3 Available setting for DMSBS register in repeat-block transfer mode (2 of 2)

Source address update mode (DMAMD.SM)	Transfer data size (DMTMD.SZ)	Available setting for DMSBSH and DMSBSL bits
Offset addition (SM = 01b)	8 bits (SZ = 00b)	0x0001 to 0xFFFF (1 to 65535)
	16 bits (SZ = 01b)	0x0001 to 0x7FFF (1 to 32767)
	32 bits (SZ = 10b)	0x0001 to 0x3FFF (1 to 16383)
	64 bits (SZ = 11b)	0x0001 to 0x1FFF (1 to 8191)
Source address is incremented or decremented (SM = 1xb)	Don't care	0x0000 (infinite) 0x0001 to 0xFFFF (1 to 65535)

In normal, repeat and block transfer mode, DMSBS is not used. The setting is invalid.

17.2.18 DMDBS : DMA Destination Buffer Size Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x2C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DMDBSH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
15:0	DMDBSL[15:0]	Functions as data transfer counter in repeat-block transfer mode. See Table 17.4 for available settings.	R/W
31:16	DMDBSH[15:0]	Specifies the repeat-area size in repeat-block transfer mode. See Table 17.4 for available settings.	R/W

Note: S-TYPE-3, P-TYPE-3

Set DMDBS while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer is disabled (DMCNT.DTE = 0).

Set the same value for DMDBSH and DMDBSL in repeat-block transfer mode. Write 0x00000000 to DMDBS in normal, repeat and block transfer mode.

DMDBSH specifies buffer size and DMDBSL functions as a 16-bit buffer size counter in repeat-block transfer mode. In repeat-block transfer mode, destination repeat area is specified by DMDBSH.

When address update mode is incremented address or decremented address, this register means the numbers of data of whole buffer. When address update mode is offset addition, this register means the numbers of data of an individual buffer. In offset addition, setting DMDBSH and DMDBSL to 0x0000 is prohibited. When final data of one buffer size is transferred, DMDBSL reloads value of DMDBSH. When address update mode is fixed address, this register is ignored. Table 17.4 shows the setting values of Destination Buffer Size Register corresponding to Transfer Data Size in Destination Address Update Mode.

Table 17.4 Available setting for DMDBS register in repeat-block transfer mode (1 of 2)

Destination address update mode (DMAMD.DM)	Transfer data size (DMTMD.SZ)	Available setting for DMDBSH and DMDBSL bits
Destination address is fixed (DM = 00b)	Don't care	0x0000 (DMDBS is not used)

Table 17.4 Available setting for DMDBS register in repeat-block transfer mode (2 of 2)

Destination address update mode (DMAMD.DM)	Transfer data size (DMTMD.SZ)	Available setting for DMDBSH and DMDBSL bits
Offset addition (DM = 01b)	8 bits (SZ = 00b)	0x0001 to 0xFFFF (1 to 65535)
	16 bits (SZ = 01b)	0x0001 to 0x7FFF (1 to 32767)
	32 bits (SZ = 10b)	0x0001 to 0x3FFF (1 to 16383)
	64 bits (SZ = 11b)	0x0001 to 0x1FFF (1 to 8191)
Destination address is incremented or decremented (DM = 1xb)	Don't care	0x0000 (infinite) 0x0001 to 0xFFFF (1 to 65535)

In normal, repeat and block transfer mode, DMDBS is not used. The setting is invalid.

17.2.19 DMBWR : DMA Bufferable Write Enable Register

Base address: DMAC0n = 0x4000_A000 + 0x0040 × n (n = 0 to 7)
DMAC0n_NS = 0x5000_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x30

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

Bit	Symbol	Function	R/W
0	BWE	Bufferable Write Enable 0: Disables Bufferable Write 1: Enables Bufferable Write	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Set DMBWR while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer is disabled (DMCNT.DTE = 0).

BWE bit (Bufferable Write Enable)

BWE bit indicates bufferable write is either enabled or disabled.

If this bit is 1, even if the write access as DMAC is completed, the actual slave write may not have ended.

When this bit is 1, it requests an early response by writing write data to the temporary buffer if possible to the slave. Therefore, even if write access as DMAC is completed, the actual slave write may not be finished. If an error occurs during write access to the target slave, the error may not be detected and the transfer may not be automatically stopped. In such cases, the error response detection interrupt DMAm_TRANSERR (m = 0, 1) will not occur.

See section 15.7.2. Operations When a Bus Error Occurs for slave groups that support bufferable writes and slave groups for which error response detection interrupts cannot be generated.

Some individual slave modules also support bufferable writes, so Refer to the respective module chapters.

[Setting condition]

- When 1 is written to this bit

[Clearing conditions]

- When 0 is written to this bit

17.2.20 DMAST : DMA Module Activation Register

Base address: DMA0 = 0x4000_A800
DMA0_NS = 0x5000_A800

Offset address: 0x00

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

Bit	Symbol	Function	R/W
0	DMST	DMAC Operation Enable 0: DMAC activation is disabled. 1: DMAC activation is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE2

DMST bit (DMAC Operation Enable)

Setting the DMAST.DMST to 1 enables DMAC activation for all channels. When the DMST bit is set to 1 (DMAC activation is enabled), and 1 is written to the DMCNT.DTE bit (DMA transfer is enabled) for multiple channels, all associated channels can be placed in the transfer request ready state at the same time.

When the DMST bit clears to 0 during DMA transfer, DMA transfer is suspended after the current data transfer associated with a single transfer request completes. To resume DMA transfer, set the DMST bit to 1 again.

[Setting condition]

- When 1 is written to this bit.

[Clearing condition]

- When 0 is written to this bit.

17.2.21 DMECHR : DMAC Error Channel Register

Base address: DMA0 = 0x4000_A800
DMA0_NS = 0x5000_A800

Offset address: 0x40

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

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

Bit	Symbol	Function	R/W
3:0	DMECH[3:0]	DMAC Error Channel Indicates the channel number causing the error 0000: Error occurred on channel 0 0001: Error occurred on channel 1 0010: Error occurred on channel 2 ⋮ ⋮ 0111: Error occurred on channel 7	R
7:4	—	These bits are read as 0. The write value should be 0.	R

Bit	Symbol	Function	R/W
8	DMECHSAM	DMAC Error Channel Security Attribution Monitor Indicates the security attribution of a channel causing the error 0: secure channel 1: non-secure channel	R
15:9	—	These bits are read as 0. The write value should be 0.	R
16	DMESTA	DMAC Error Status 0: No DMA transfer error occurred 1: DMA transfer error occurred	R/W *1
31:17	—	These bits are read as 0. The write value should be 0.	R

Note: P-TYPE-2
Note 1. Writing to DMESTA depends on the value of DMECHSAM.

When reading this register, it can be accessed from both secure and non-secure access.

When writing this register, it depends on DMECHR.DMESA.

- When DMESA = 1, it can be accessed from secure and non-secure access.
- When DMESA = 0, it can be accessed from secure. An error is returned when write access is performed in a non-secure access.

This register is cleared by a reset caused by a transfer error. Please select interrupt DMAm_TRANSERR (m = 0, 1) in BUS.OADCFG.OAD when you want to debug the program.

DMECH[3:0] bits (DMAC Error Channel)

When a transfer error due to DMA transfer occurs, it stores the channel of DMAC that was violated.

When reset was selected in MPU.MMPUOAD.OAD and TZF.TZFOAD.OAD, Since this register is also reset. Please select NMI when you want to debug the program.

[Set condition]

- When the DMAC transfer error occurs and DMESTA = 0.

[Clearing condition]

- When 1 is written to DMESTA.

DMECHSAM bit (DMAC Error Channel Security Attribution Monitor)

When a transfer error due to DMA transfer occurs, it indicates the security attribution of the violating DMAC channel.

When reset was selected in MPU.MMPUOAD.OAD and TZF.TZFOAD.OAD, Since this register is also reset. Please select NMI when you want to debug the program.

[Set condition]

- When the DMAC transfer error occurs and DMESTA = 0.

[Clearing condition]

- When 1 is written to DMESTA.

DMESTA bit (DMAC Error Status)

Indicates whether or not a DMA transfer error occurred.

DMECH[3:0], DMECHSAM, DMESTA are cleared by writing 1 to DMESTA. Writing 0 to DMESTA is ignored.

When reset was selected in MPU.MMPUOAD.OAD and TZF.TZFOAD.OAD, Since this register is also reset. Please select NMI when you want to debug the program.

[Set condition]

- When the DMAC transfer error occurs.

[Clearing condition]

- When 1 is written to DMESTA.

Note: When DMECHSAM = 1, it can be cleared in the secure state and non-secure state. DMECHSAM = 0, it cannot be cleared in the non-secure state.

17.2.22 DMCTL : DMAC Control Register

Base address: DMA0 = 0x4000_A800
DMA0_NS = 0x5000_A800

Offset address: 0x10

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

Bit	Symbol	Function	R/W
0	PR	Priority Control Select 0: Fixed priority mode 1: Round-robin mode	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	ERCH	Clear Channel Select 0: Target channel only 1: All channels	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Set DMCTL while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer is disabled (DMCNT.DTE = 0).

PR bit (Priority Control Select)

When DMCTL.PR = 0, arbitration is performed in the fixed priority mode of channel 0 > channel 1 > channel 2 > channel 3 …> channel 7 > DTC (channel 0: highest priority) and DMA transfer is started.

When DMCTL.PR = 1, arbitration is performed in the round-robin mode and DMA transfer is started.

ERCH bit (Clear Channel Select)

When DMCTL.ERCH = 0, the ICU clears all bits of the target channel of DELSRn (n = 0 to 7) by the error response during transfer. DELSRn (n = 0 to 7) other than the target channel is not cleared.

When DMCTL.ERCH = 1, the ICU clears all bits of all channels of DELSRn (n = 0 to 7) by the error response during transfer.

17.3 Operation

17.3.1 Transfer Mode

17.3.1.1 Normal Transfer Mode

In normal transfer mode, one data is transferred by one transfer request. A maximum of 65535 can be set as the number of transfer operations using the DMCRAL register. When these bits are set to 0x0000, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free-running function). Setting DMCRB register is invalid in normal transfer mode. Except in free-running function, a transfer end interrupt request can be generated after completion of the specified number of transfer operations.

Table 17.5 summarizes the register update operation in normal transfer mode, and Figure 17.2 shows the operation in normal transfer mode.

Table 17.5 Register update operation in normal transfer mode

Register	Function	Update operation after completion of a transfer by one transfer request
DMSAR	Transfer source address	Increment/decrement/fixed/offset addition
DMDAR	Transfer destination address	Increment/decrement/fixed/offset addition
DMCRAL	Transfer count	Decremented by one/not updated (in free running function)
DMCRAH	—	Not updated (Not used in normal transfer mode)
DMCRB	—	Not updated (Not used in normal transfer mode)

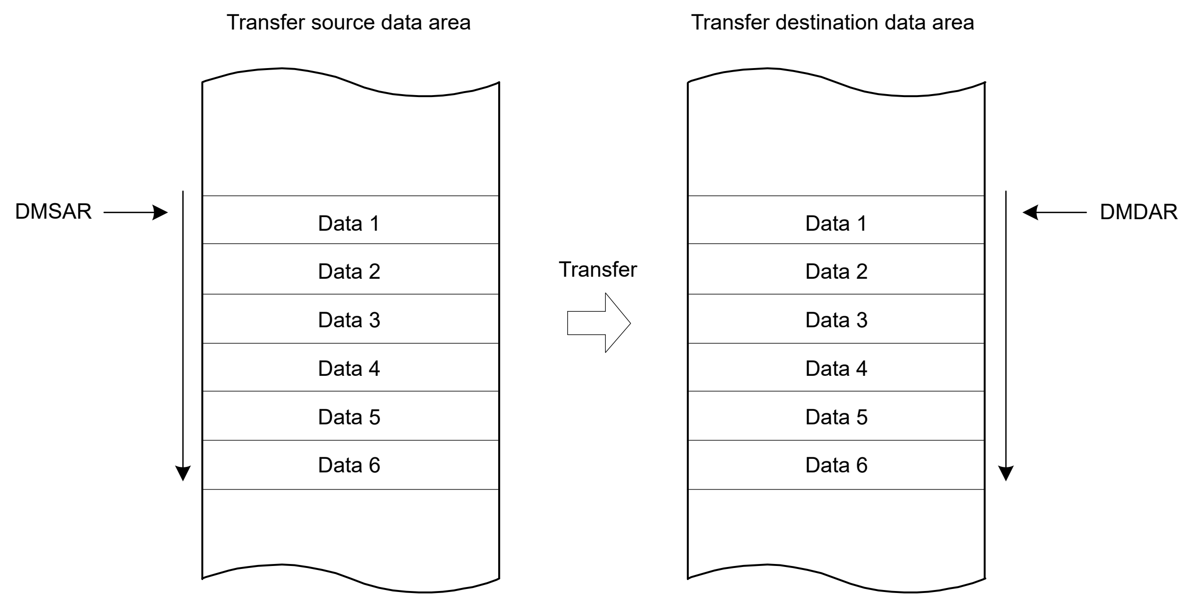


Figure 17.2 Operation in normal transfer mode

17.3.1.2 Repeat Transfer Mode

In repeat transfer mode, one data is transferred by one transfer request.

A maximum of 1K data can be set as a total repeat transfer size using DMCRA register.

A maximum of 64K can be set as the number of repeat transfer operations using DMCRB register; therefore, a maximum of 64M data (1K data × 64K counts of repeat transfer operations) can be set as a total data transfer size.

Either the transfer source or transfer destination can be specified as a repeat area. When transfer of the repeat size data is completed, the address of the specified repeat area (DMSAR or DMDAR) returns to the transfer start address. When data of the specified repeat size has all been transferred in repeat transfer mode, DMA transfer can be stopped, and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DMCNT.DTE bit in the repeat size end interrupt handling.

A transfer end interrupt request can be generated after completion of the specified number of repeat transfer operations.

Table 17.6 summarizes the register update operation in repeat transfer mode, and Figure 17.3 shows the operation in repeat transfer mode.

Table 17.6 Register update operation in repeat transfer mode

Register	Function	Update operation after completion of a transfer by one transfer request	
		When DMCRAL register is not 1	When DMCRAL register is 1 (Transfer of the last data in repeat size)
DMSAR	Transfer source address	Increment/decrement/fixed/offset addition	• DMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition • DMTMD.DTS[1:0] = 01b Initial value of DMSAR • DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition
DMDAR	Transfer destination address	Increment/decrement/fixed/offset addition	• DMTMD.DTS[1:0] = 00b Initial value of DMDAR • DMTMD.DTS[1:0] = 01b Increment/decrement/fixed/offset addition • DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition
DMCRAH	Repeat size	Not updated	Not updated
DMCRAL	Transfer count	Decremented by one	DMCRAH
DMCRBH	Number of repeat transfer operations	Not updated	Not updated
DMCRBL	Count of repeat transfer operations	Not updated	Decremented by one

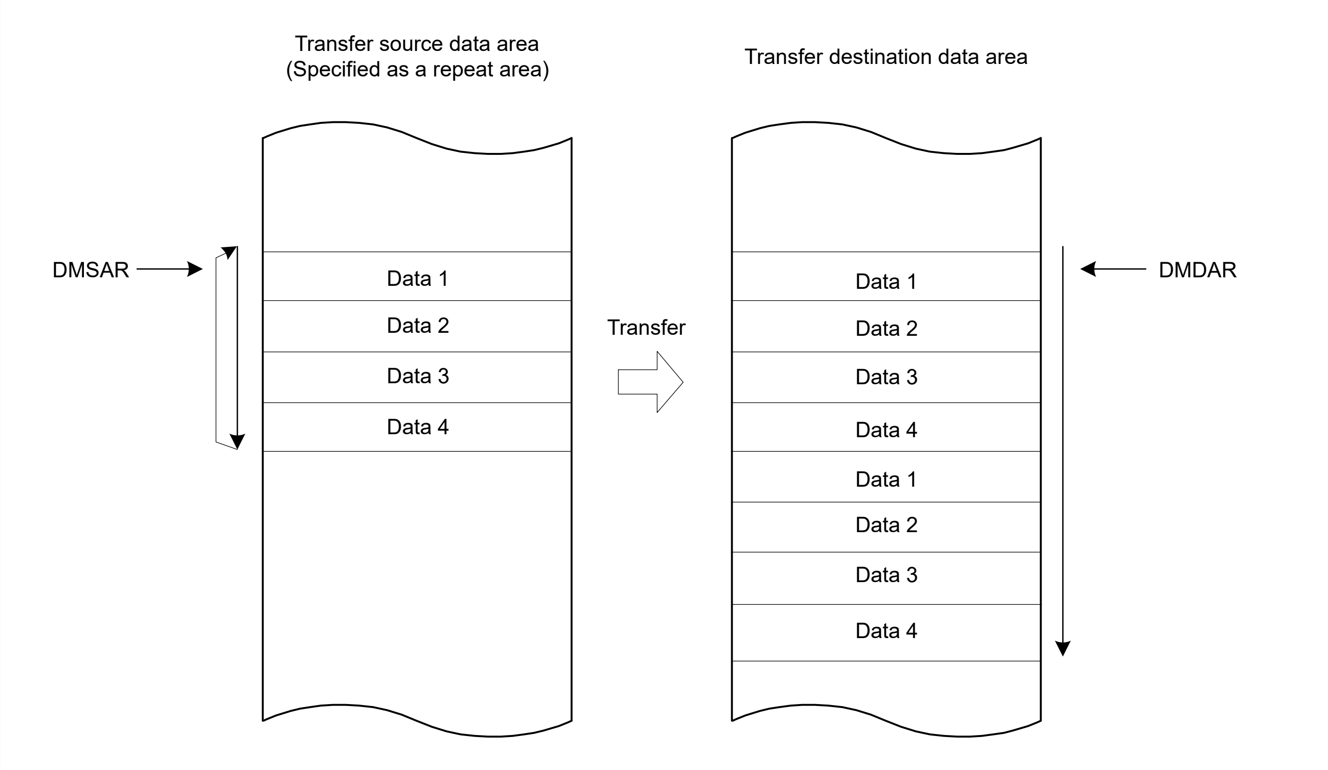


Figure 17.3 Operation in repeat transfer mode

17.3.1.3 Block Transfer Mode

In block transfer mode, a single block data is transferred by one transfer request.

A maximum of 1K data can be set as a total block transfer size using DMCRA register.

A maximum of 64K can be set as the number of block transfer operations using DMCRB register; therefore, a maximum of 64M data (1K data × 64K counts of block transfer operations) can be set as a total data transfer size.

Either the transfer source or transfer destination can be specified as a block area. When transfer of a single block data is completed, the address of the specified block area (DMSAR or DMDAR) returns to the transfer start address. When a single block data has all been transferred in block transfer mode, DMA transfer can be stopped, and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DMCNT.DTE bit in the repeat size end interrupt handling.

Transfer end interrupt request can be generated after completion of the specified number of block transfer operations.

Table 17.7 summarizes the register update operation in block transfer mode, and Figure 17.4 shows the operation in block transfer mode.

Table 17.7 Register update operation in block transfer mode

Register	Function	Update operation after completion of single-block transfer by one transfer request
DMSAR	Transfer source address	• DMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition • DMTMD.DTS[1:0] = 01b Initial value of DMSAR • DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition
DMDAR	Transfer destination address	• DMTMD.DTS[1:0] = 00b Initial value of DMDAR • DMTMD.DTS[1:0] = 01b Increment/decrement/fixed/offset addition • DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition
DMCRAH	Block size	Not updated
DMCRAL	Transfer count	DMCRAH
DMCRBH	Number of block transfer operations	Not updated
DMCRBL	Count of block transfer operations	Decremented by one

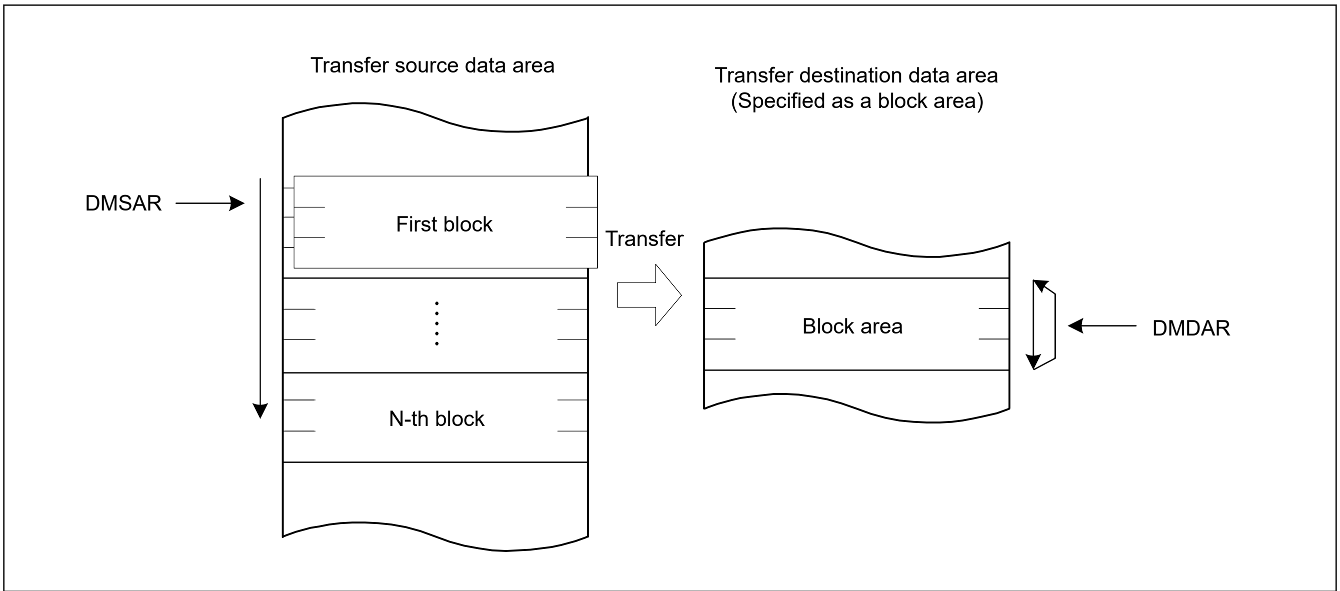


Figure 17.4 Operation in block transfer mode

17.3.1.4 Repeat-Block Transfer Mode

Repeat-block transfer is the operation mode with the following functions added to the block transfer function.

Repeat function: Added function (ring buffer) to repeat specified address area.

Offset function: Multiple areas with offset can be specified within one block transfer.

The repeat function and the offset function can be used for both the transfer source and the transfer destination of repeat-block transfer.

Figure 17.5 shows an example of adding a repeat function to the transfer destination.

Figure 17.6 shows repeat-block transfer with an offset to the transfer destination.

In repeat-block transfer mode, a single block data is transferred by one transfer request.

A maximum of 1K data can be set as a total block transfer size using DMCRA of the DMACn.

A maximum of 64K can be set as the number of block transfer operations using DMCRB of the DMACn; therefore, a maximum of 64M data (1K data × 64K counts of block transfer operations) can be set as a total data transfer size.

A transfer end interrupt request can be generated after completion of the specified number of repeat-block transfer operations.

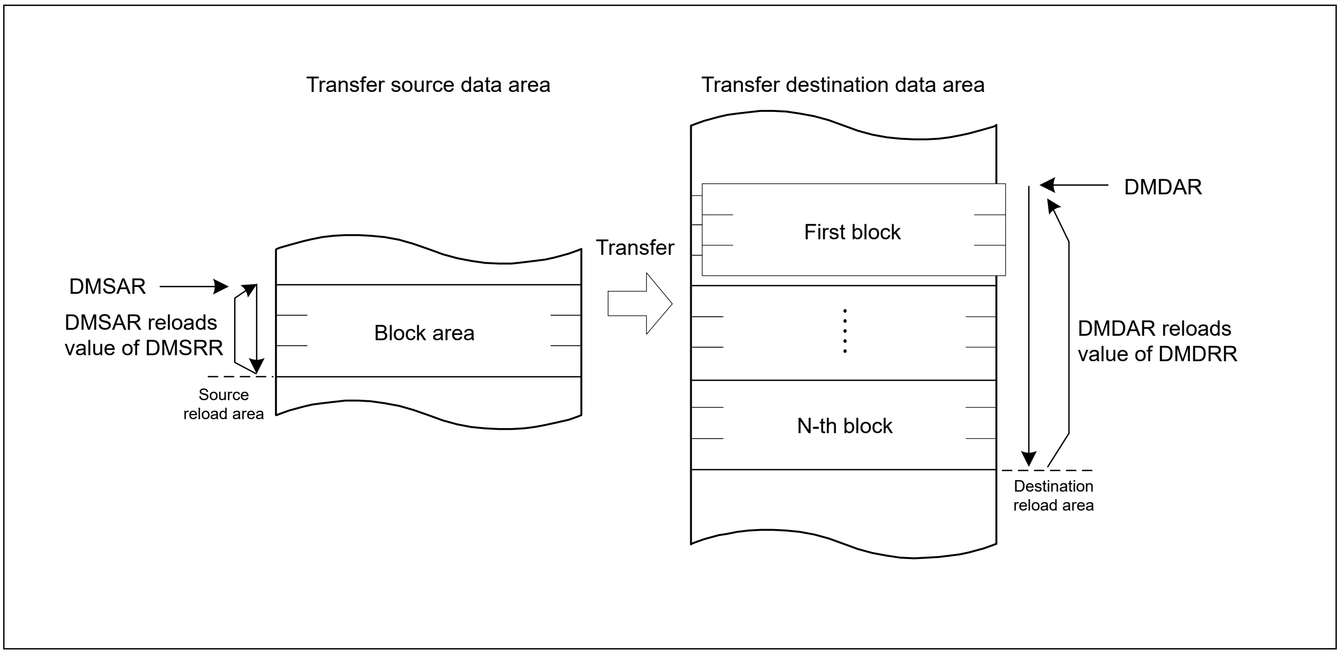


Figure 17.5 Operation in repeat-block transfer mode

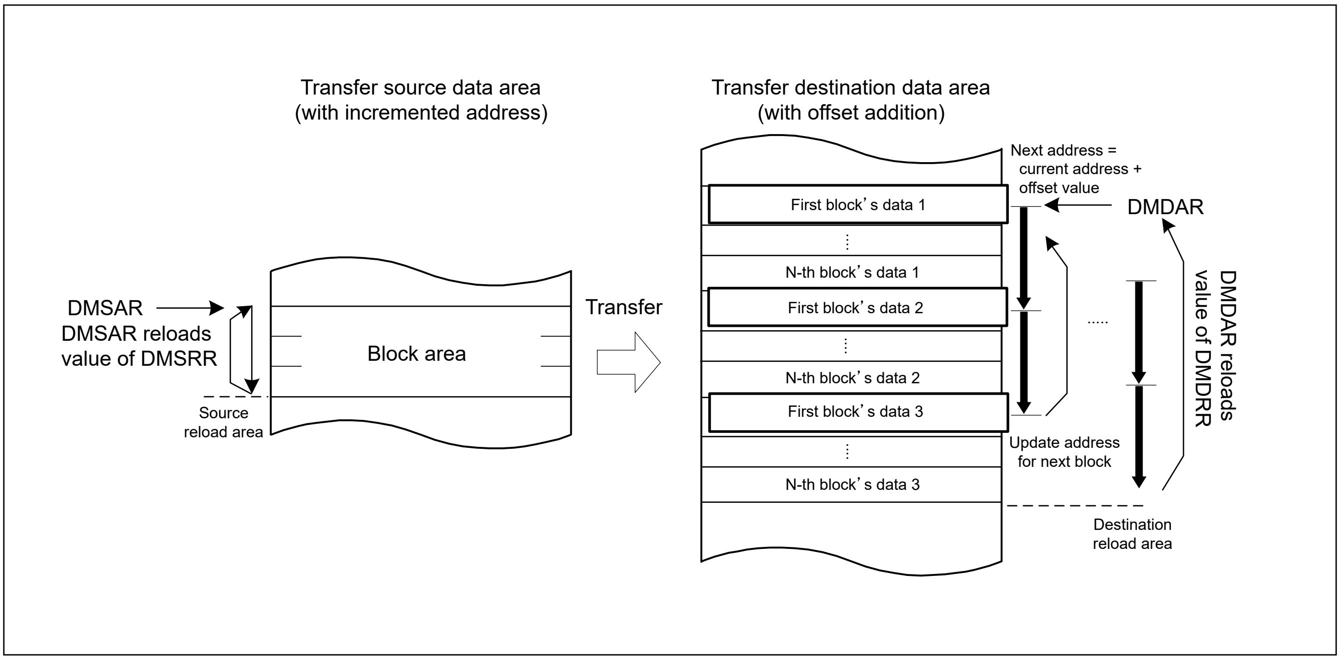


Figure 17.6 Operation in repeat-block transfer mode with offset addition

Table 17.8 to Table 17.13 summarize the register update operations in repeat-block transfer mode.

For more information about address update function in repeat-block transfer mode, see section 17.3.5. Address Update Function in Repeat-Block Transfer Mode.

Table 17.8 Register update operation associated with source area in repeat-block transfer mode (fixed address DMAMD.SM[1:0] = 00b)

Register	Function	Update operation after single data is transferred		
		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMSRR	Transfer source reload address	Not updated	Not updated	Not updated
DMSAR	Transfer source address	Not updated	Not updated	Not updated
DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated
DMCRAL[15:0]	Block size count	Decremented by 1	DMCRAH[9:0]	DMCRAH[9:0]
DMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated
DMCRBL[15:0]	Count of block transfer operations when DMTMD.TKP = 0	Not updated	Decremented by 1	0
	Count of block transfer operations when DMTMD.TKP = 1			DMCRBH[15:0]

Table 17.9 Register update operation associated with destination area in repeat-block transfer mode (fixed address DMAMD.DM[1:0] = 00b)

Register	Function	Update operation after single data is transferred		
		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMDRR	Transfer destination reload address	Not updated	Not updated	Not updated
DMDAR	Transfer destination address	Not updated	Not updated	Not updated
DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated
DMCRAL[15:0]	Block size count	Decremented by 1	DMCRAH[9:0]	DMCRAH[9:0]
DMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated
DMCRBL[15:0]	Count of block transfer operations when DMTMD.TKP = 0	Not updated	Decremented by 1	0
	Count of block transfer operations when DMTMD.TKP = 1			DMCRBH[15:0]

Table 17.10 Register update operation associated with source area in repeat-block transfer mode (incremented or decremented address DMAMD.SM[1:0] = 10b or 11b)

Register	Function	Update operation after single data is transferred					
		DMSBSL[15:0] is not 1			DMSBSL[15:0] is 1		
		DMCRAL [15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)		DMCRAL [15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1		DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMSRR	Transfer source reload address	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMSAR	Transfer source address when DMTMD.SM [1:0] = 10b	Incremented by Data Size			DMSRR		
	Transfer source address when DMTMD.SM [1:0] = 11b	Decremented by Data Size			DMSRR		
DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRAL[15:0]	Block size count	Decremented by 1	DMCRAH[9:0]	DMCRAH[9:0]	Decremented by 1	DMCRAH[9:0]	DMCRAH[9:0]
DMSBSH[15:0]	Source buffer size (Repeat-size)	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMSBSL[15:0]	Count of transfer data in source buffer	Decremented by 1	Decremented by 1	Decremented by 1	DMSBSH	DMSBSH	DMSBSH
DMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRBL[15:0]	Count of block transfer operations when DMTMD.TKP = 0	Not updated	Decremented by 1	0	Not updated	Decremented by 1	0
	Count of block transfer operations when DMTMD.TKP = 1			DMCRBH[15:0]			DMCRBH[15:0]

Table 17.11 Register update operation associated with destination area in repeat-block transfer mode (incremented or decremented address DMAMD.DM[1:0] = 10b or 11b)

Register	Function	Update operation after single data is transferred					
		DMDBSL[15:0] is not 1			DMDBSL[15:0] is 1		
		DMCRAL [15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)		DMCRAL [15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1		DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMDRR	Transfer destination reload address	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMDAR	Transfer destination address when DMTMD.DM [1:0] = 10b	Incremented by Data Size			DMDRR		
	Transfer destination address when DMTMD.DM [1:0] = 11b	Decremented by Data Size			DMDRR		
DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRAL[15:0]	Block size count	Decremented by 1	DMCRAH[9:0]	DMCRAH[9:0]	Decremented by 1	DMCRAH[9:0]	DMCRAH[9:0]
DMDBSH[15:0]	Destination buffer size (Repeat-size)	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMDBSL[15:0]	Count of transfer data in destination buffer	Decremented by 1	Decremented by 1	Decremented by 1	DMDBSH	DMDBSH	DMDBSH
DMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRBL[15:0]	Count of block transfer operations when DMTMD.TKP = 0	Not updated	Decremented by 1	0	Not updated	Decremented by 1	0
	Count of block transfer operations when DMTMD.TKP = 1			DMCRBH[15:0]			DMCRBH[15:0]

Table 17.12 Register update operation associated with source area in repeat-block transfer mode (offset addition DMAMD.SM[1:0] = 01b) (1 of 2)

Register	Function	DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)			
			DMSBSL[15:0] is not 1		DMSBSL[15:0] is 1	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1	DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMSRR	Transfer source reload address	Not updated	Not updated	Not updated	Not updated	Not updated

Table 17.12 Register update operation associated with source area in repeat-block transfer mode (offset addition DMAMD.SM[1:0] = 01b) (2 of 2)

Register	Function	DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)			
			DMSBSL[15:0] is not 1		DMSBSL[15:0] is 1	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1	DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMSAR	Transfer source address when DMAMD.SADR = 0	Offset addition by DMSBSH	DMSRR		DMSRR	
	Transfer source address when DMAMD.SADR = 1		DMSRR + (DMS-BSH - DMSBSL) × DataSize			
DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRAL[15:0]	Block size count	Decremented by 1	DMCRAH[9:0]	DMCRAH[9:0]	DMCRAH[9:0]	DMCRAH[9:0]
DMSBSH[15:0]	Source buffer size (Repeat-size)	Not updated	Not updated	Not updated	Not updated	Not updated
DMSBSL[15:0]	Count of transfer data in source buffer	Not updated	Decremented by 1	Decremented by 1	DMSBSH	DMSBSH
DMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRBL[15:0]	Count of block transfer operations when DMTMD.TKP = 0	Not updated	Decremented by 1	0	Decremented by 1	0
	Count of block transfer operations when DMTMD.TKP = 1			DMCRBH[15:0]		DMCRBH[15:0]

Table 17.13 Register update operation associated with destination area in repeat-block transfer mode (offset addition DMAMD.DM[1:0] = 01b) (1 of 2)

Register	Function	DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)			
			DMDBSL[15:0] is not 1		DMDBSL[15:0] is 1	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1	DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMDRR	Transfer destination reload address	Not updated	Not updated	Not updated	Not updated	Not updated
DMDAR	Transfer destination address when DMAMD.DADR = 0	Offset addition by DMDBSH	DMDRR		DMDRR	
	Transfer destination address when DMAMD.DADR = 1		DMDRR + (DMDBSH - DMDBSL) × DataSize			
DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRAL[15:0]	Block size count	Decremented by 1	DMCRAH[9:0]	DMCRAH[9:0]	DMCRAH[9:0]	DMCRAH[9:0]

Table 17.13 Register update operation associated with destination area in repeat-block transfer mode (offset addition DMAMD.DM[1:0] = 01b) (2 of 2)

<table>
<tr><th rowspan="3">Register</th><th rowspan="3">Function</th><th rowspan="3">DMCRAL[15:0] is not 1</th><th colspan="4">DMCRAL[15:0] is 1 (single block is transferred)</th></tr>
<tr><th colspan="2">DMDBSL[15:0] is not 1</th><th colspan="2">DMDBSL[15:0] is 1</th></tr>
<tr><th>DMCRBL[15:0] is not 1</th><th>DMCRBL[15:0] is 1</th><th>DMCRBL[15:0] is not 1</th><th>DMCRBL[15:0] is 1</th></tr>
<tr><td>DMDBSH[15:0]</td><td>Destination buffer size
(Repeat-size)</td><td>Not updated</td><td>Not updated</td><td>Not updated</td><td>Not updated</td><td>Not updated</td></tr>
<tr><td>DMDBSL[15:0]</td><td>Count of transfer data in destination buffer</td><td>Not updated</td><td>Decremented by 1</td><td>Decremented by 1</td><td>DMDBSH</td><td>DMDBSH</td></tr>
<tr><td>DMCRBH[15:0]</td><td>Number of block transfer operations</td><td>Not updated</td><td>Not updated</td><td>Not updated</td><td>Not updated</td><td>Not updated</td></tr>
<tr><td rowspan="2">DMCRBL[15:0]</td><td>Count of block transfer operations when DMTMD.TKP = 0</td><td rowspan="2">Not updated</td><td rowspan="2">Decremented by 1</td><td>0</td><td rowspan="2">Decremented by 1</td><td>0</td></tr>
<tr><td>Count of block transfer operations when DMTMD.TKP = 1</td><td>DMCRBH[15:0]</td><td>DMCRBH[15:0]</td></tr>
</table>

17.3.2 Extended Repeat Area Function

The DMAC supports a function to specify the extended repeat areas on the transfer source and destination addresses. With the extended repeat areas set, the address registers repeatedly indicate the addresses of the specified extended repeat areas.

The extended repeat areas can be specified separately to the transfer source address register (DMSAR) and transfer destination address register (DMDAR).

The extended repeat area on the source address is specified by the DMAMD.SARA[4:0] bits. The extended repeat area on the destination address is specified by the DMAMD.DARA[4:0] bits. The size can be specified separately for the source and destination sides.

However, the area (of transfer source or transfer destination) which is specified as the repeat area or block area should not be specified as the extended repeat area.

When the address register value reaches the end address of the extended repeat area and the extended repeat area overflows, DMA transfer is stopped and an interrupt by an extended repeat area overflow can be requested. When an overflow occurs in the extended repeat area on the transfer source while the DMINT.SARIE bit is set to 1, the DMSTS.ESIF flag is set to 1 and the DMCNT.DTE bit is cleared to 0 to stop DMA transfer. At this time, if the DMINT.ESIE bit is set to 1, an interrupt by an extended repeat area overflow is requested. When the DMINT.DARIE bit is set to 1, the destination address register becomes a target to apply the function. DMA transfer can be resumed by writing 1 to the DMCNT.DTE bit in the interrupt handling.

Figure 17.7 shows an example of the extended repeat area operation.

Eight bytes are specified as an extended repeat area by the lower three bits of DMSAR (DMAMD.SARA[4:0] bits = 00011b). The data size is eight bits (DMTMD.SZ[1:0] = 00b).

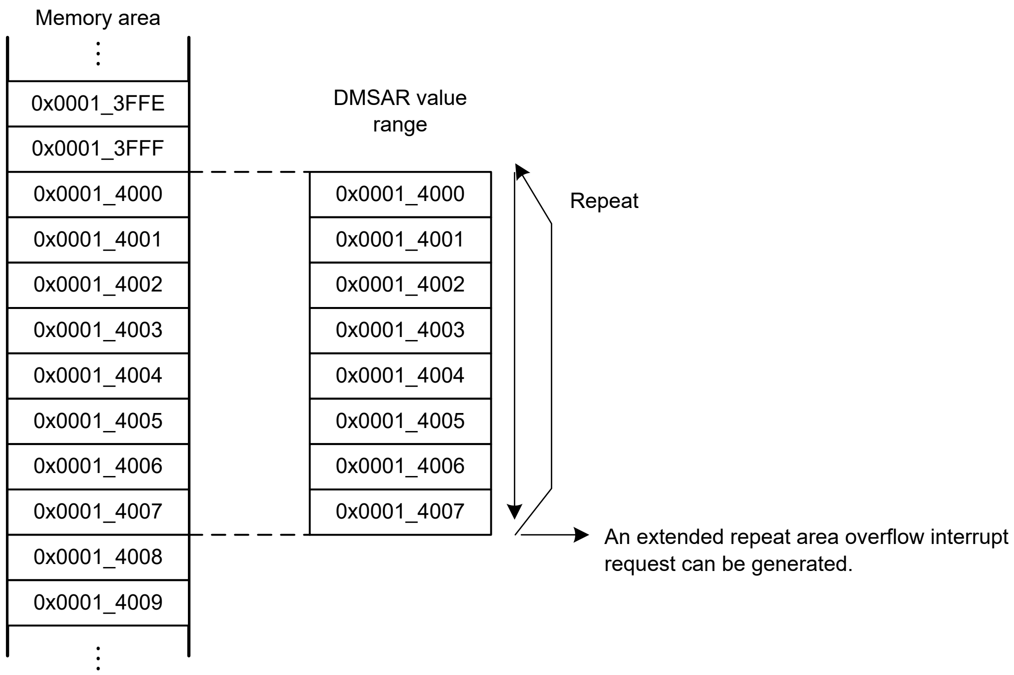


Figure 17.7 Example of extended repeat area operation

When an interrupt by an extended repeat area overflow is used in block transfer mode, the following should be taken into consideration.

When a transfer is stopped by an interrupt by an extended repeat area overflow, the address register must be set so that the block size is a power of 2 or the block size boundary is aligned with the extended repeat area boundary. When an overflow on the extended repeat area occurs during a transfer of one block, the interrupt by the overflow is suspended until transfer of the block is completed, and the transfer overruns.

Figure 17.8 shows an example when the extended repeat area function is used in block transfer mode.

Eight bytes are specified as an extended repeat area by the lower three bits of DMSAR (DMAMD.SARA[4:0] bits = 00011b), block transfer mode with block size 5 is set (DMCRA = 0x0005_0005), and transfer source address is not specified as a block area. Data size is eight bits (DMTMD.SZ[1:0] bits = 00b).

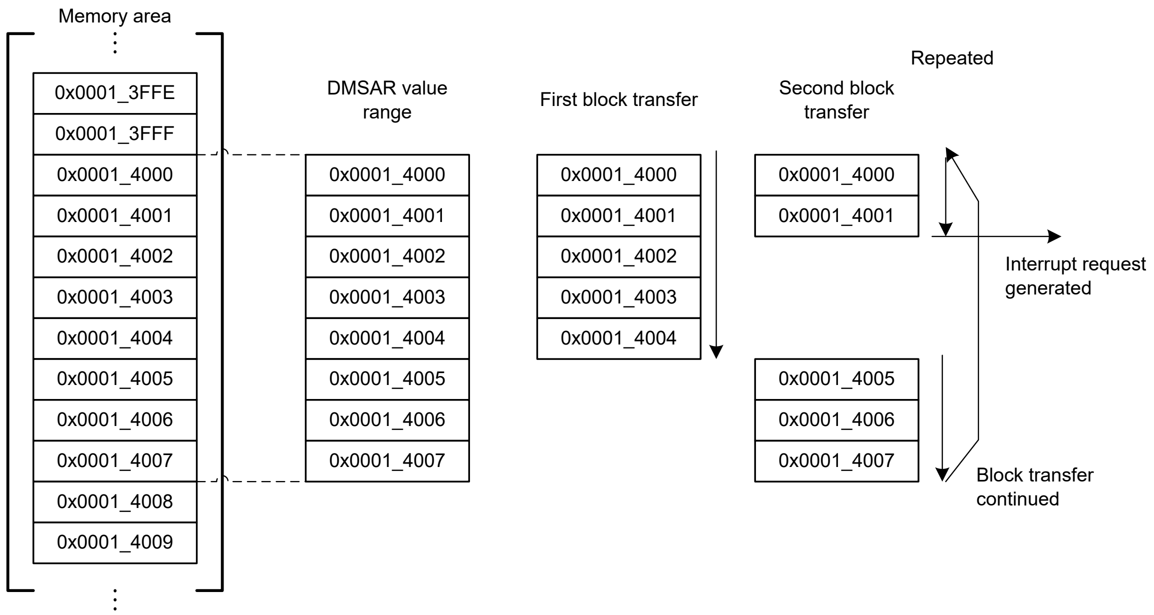


Figure 17.8 Example of extended repeat area function in block transfer mode

17.3.3 Free-running Function

The DMAC supports free-running function. This function allows to transfer repeatedly without reconfiguring in interrupt handler.

17.3.3.1 In Normal Transfer Mode

In normal transfer mode, when DMCRA.DMCRAL bits are set to 0x0000, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped.

For more information, see section 17.3.1.1. Normal Transfer Mode.

17.3.3.2 In Other Transfer Modes

In repeat, block and repeat-block transfer mode, the DMAC supports free-running function using the DMTMD.TKP bit. If the DMTMD.TKP bit is to be set to 1, the transfer is not stopped by completion of specified total number of transfer operations and reloads DMCRBH repeatedly.

Figure 17.9 show an example of block transfer operation without free-running function.

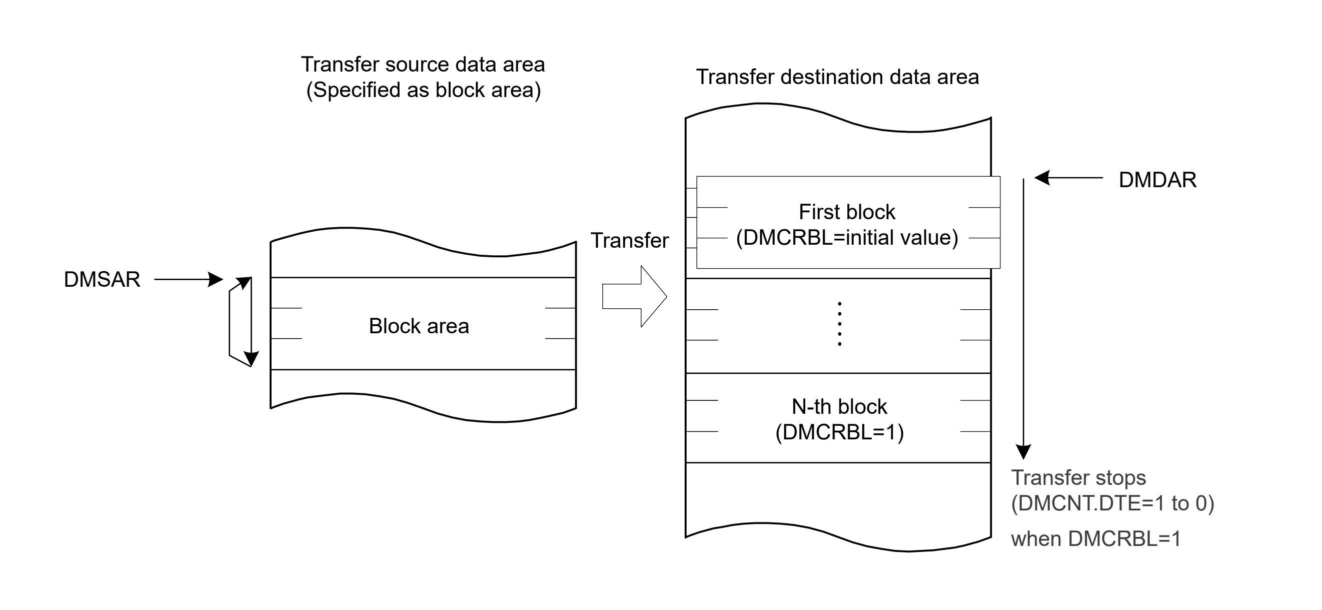


Figure 17.9 Operation in block transfer mode when DMTMD.TKP bit is set to 0

Figure 17.10 show an example of block transfer operation with free-running function.

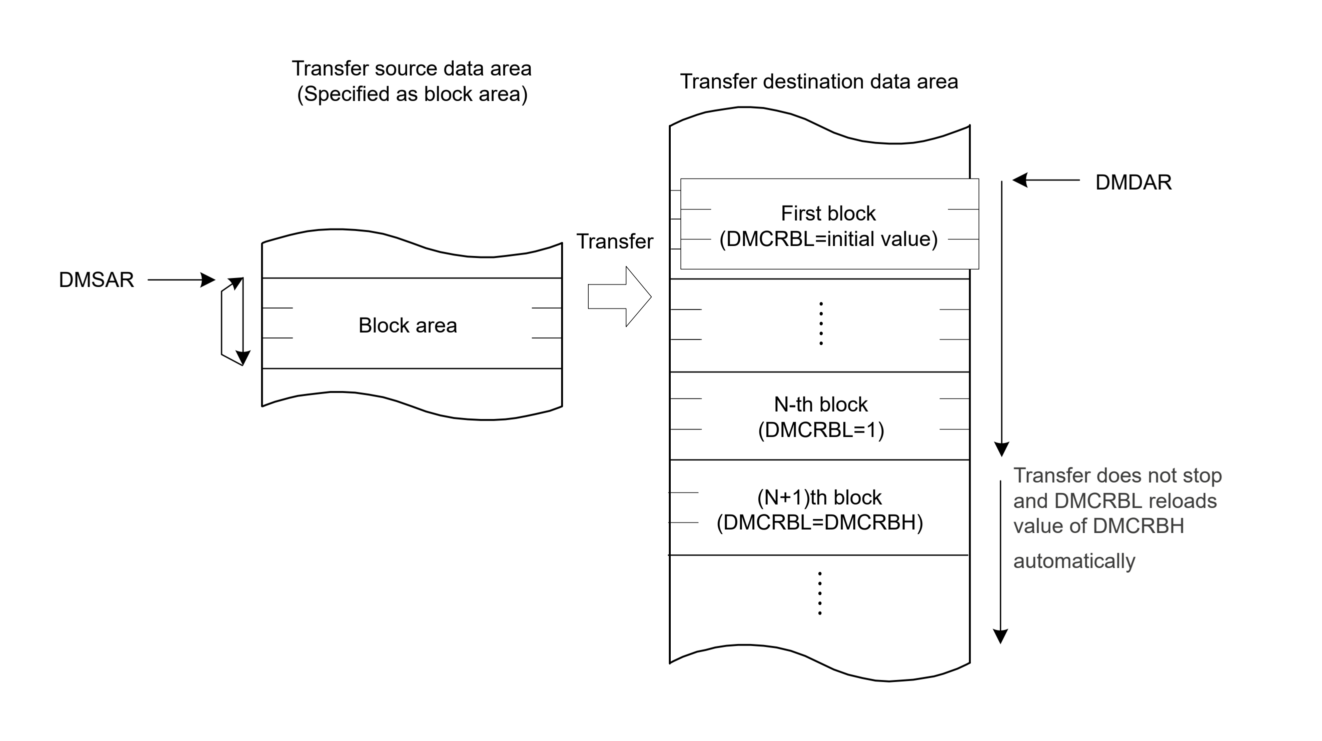


Figure 17.10 Operation in block transfer mode when DMTMD.TKP bit is set to 1

17.3.4 Address Update Function Using Offset

The source and destination addresses can be updated by fixing, increment, decrement, or offset addition. In normal, repeat and block transfer mode, when the offset addition is selected, the offset specified by the DMA offset register (DMOFR) is added to the address every time the DMAC performs one data transfer. This function realizes a data transfer where addresses are allocated to separated areas.

Offset subtraction can also be realized by setting a negative value in DMOFR. In this case, the negative value must be 2’s complement.

DMSBS or DMDBS are used instead of DMOFR in repeat-block transfer mode. For more information section 17.3.1.4. Repeat-Block Transfer Mode

Table 17.14 shows the address update method in each address update mode.

Table 17.14 Address update method in each address update mode

Address update mode	Settings of DMAMD.SM[1:0] and DMAMD.DM[1:0] for address update modes	Address update method (for different SZ[1:0] settings in DMTMD)			
		SZ[1:0] = 00b	SZ[1:0] = 01b	SZ[1:0] = 10b	SZ[1:0] = 11b
Address fixed	00b	Fixed			
Offset addition	01b	+DMOFR*1			
Increment	10b	+1	+2	+4	+8
Decrement	11b	–1	–2	–4	–8

Note 1. When setting a negative value in the DMA Offset Register, the value must be in two's complement, obtained by the following formula:
two's complement of a negative offset value = ~ (offset) + 1 (~ = bit inversion)

17.3.4.1 Basic Transfer Using Offset Addition

Figure 17.11 shows an example of address updating using offset addition.

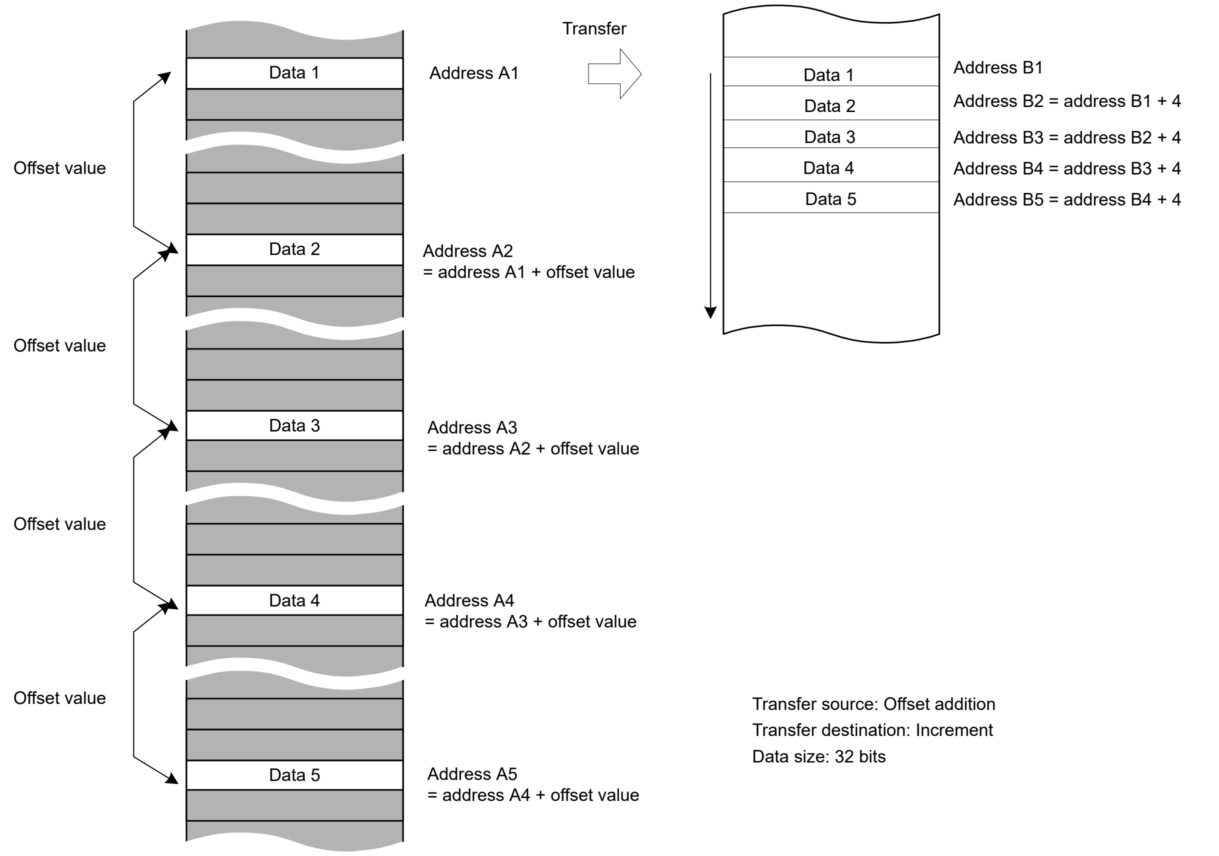


Figure 17.11 Example of address updating by offset addition

Figure 17.11 shows the setting of the following.

- The transfer data is 32 bits long.
- Offset addition is set as the transfer source address update mode.

- Increment is set as the transfer destination address update mode.

The second and subsequent data is each read from the transfer source address obtained by adding the offset value to the previous address. The data read from the addresses at the specified intervals is written to the continuous locations on the destination.

17.3.4.2 Example of XY Conversion Using Offset Addition

Figure 17.12 shows the XY conversion using offset addition in repeat transfer mode.

Settings are as follows:

- DMAMD.SM — Transfer source address update mode: Offset addition.
- DMAMD.DM — Transfer destination address update mode: Destination address is incremented.
- DMTMD.SZ — Transfer data size select: 32 bits.
- DMTMD.MD — Transfer mode select: Repeat transfer.
- DMTMD.DTS — Repeat area select: The source is specified as the repeat area.
- DMOFR — Offset address: 0x10.
- DMCRA — Repeat size: 0x4.
- DMINT.RPTIE — The repeat size end interrupt is enabled.

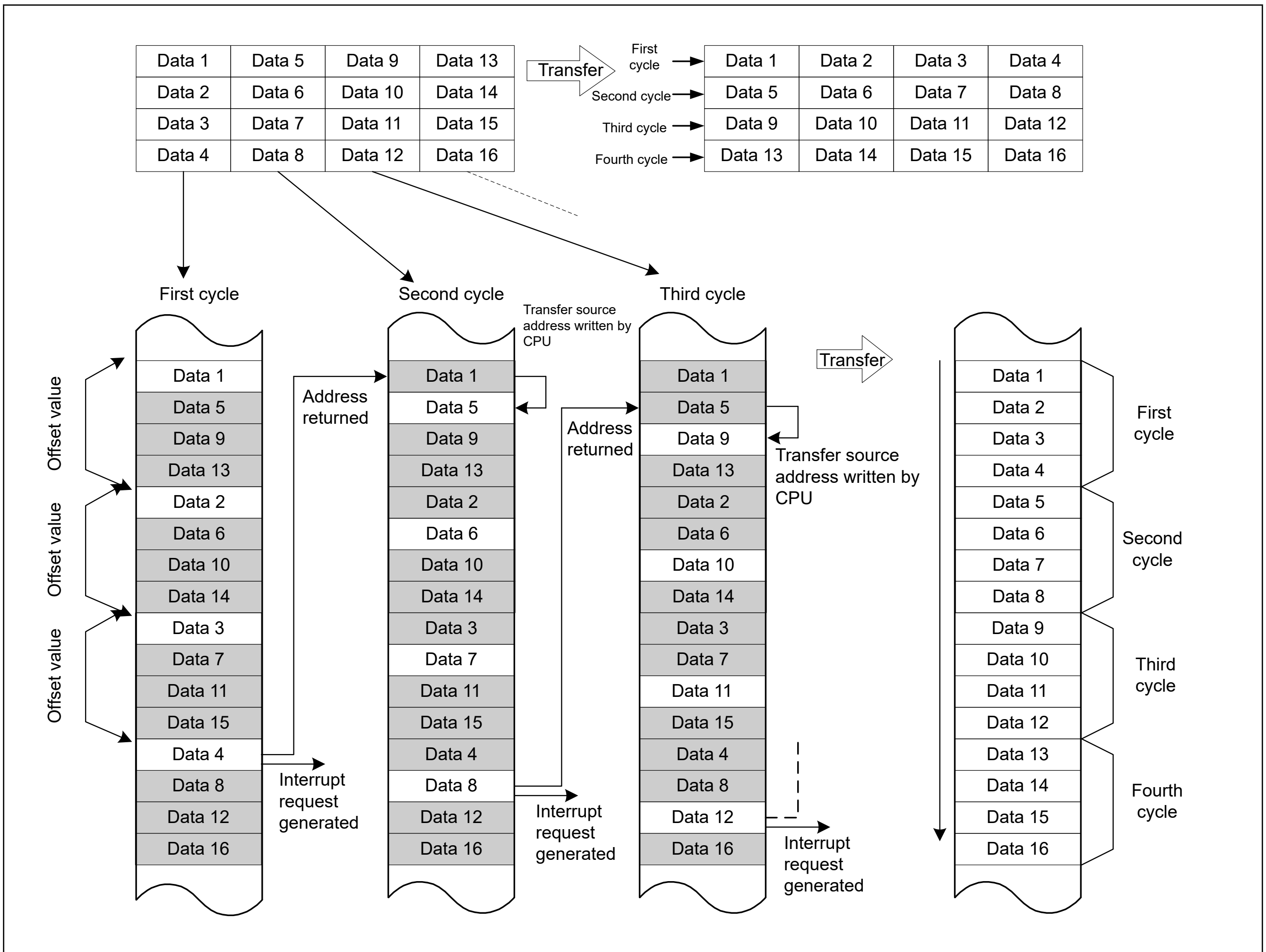


Figure 17.12 XY conversion operation using offset addition in repeat transfer mode

When a transfer starts, the offset value is added to the transfer source address every time data is transferred. The transfer data is written to continuous transfer destination addresses. When data 4 is transferred:

- The repeat size of data transfer is complete.
- The transfer source address returns to the transfer start address (the address of data 1 on the transfer source).
- A repeat size end interrupt is requested.

During the time this interrupt pauses the transfer, the following operations are performed.

- DMSAR — Rewrite the DMA transfer source address to the address of data 5
(with the above example, the data 1 address + 4).
- DMCNT — Set the DTE bit to 1.

The DMA transfer is resumed from the state when the DMA transfer is stopped. After that, the operations described above are repeated until the transfer source data is transposed to the destination area (XY conversion).

Figure 17.13 shows a flowchart of the XY conversion.

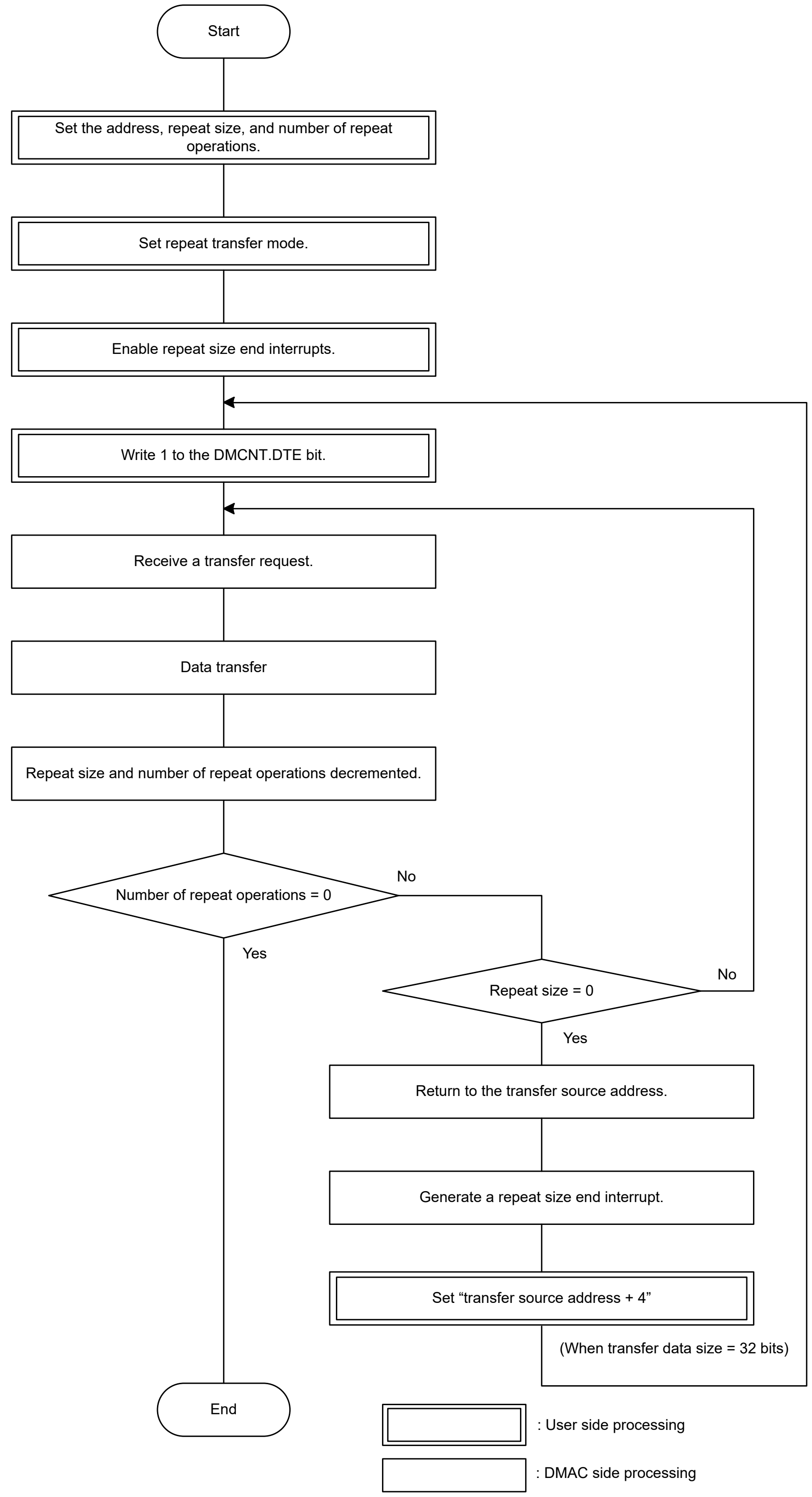


Figure 17.13 XY conversion flowchart using offset addition in repeat transfer mode

17.3.5 Address Update Function in Repeat-Block Transfer Mode

Repeat-block transfer mode is an extension of repeat transfer mode and block transfer mode. However, the detailed behavior of the address update is different from these two modes. Here are the details of the address update function in repeat-block transfer mode.

17.3.5.1 Fixed Address Mode

When DMAMD.SM[1:0] is set to 00b, the address update mode of the source is fixed address. And when DMAMD.DM[1:0] is set to 00b, the address update mode of the destination is fixed address.

In fixed address, the address is not updated from the initial value of DMSAR and DMDAR. If the block size (DMCRA) is larger than 1, the same data will be transferred multiple times for one request.

Figure 17.14 shows address update in fixed address.

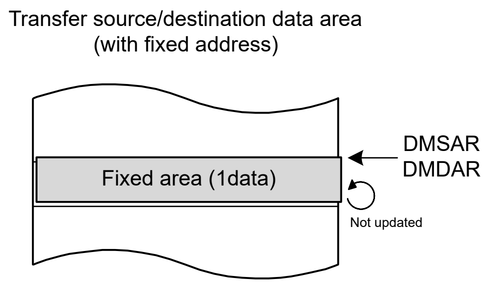


Figure 17.14 Address update in fixed address

17.3.5.2 Incremental and Decremental Address Mode

When DMAMD.SM[1:0] is set to 10b, the address update mode of the source is incremental address. And when DMAMD.DM[1:0] is set to 10b, the address update mode of the destination is incremental address. When DMAMD.SM[1:0] is set to 11b, the address update mode of the source is decremental address. And when DMAMD.DM[1:0] is set to 11b, the address update mode of the destination is decremental address.

In these update modes, the address is incremented or decremented according to the setting of DMTMD.SZ[1:0].

In these update modes DMSBS and DMDBS indicates a reload area. The unit of DMSBS and DMDBS is "number of data". At the start of transfer, DMSBSL and DMDBSL, which is the lower 16 bits of DMSBS and DMDBS, operates as a down counter and decrements each time one data transfer is performed. When the value becomes 1, DMSAR and DMDAR reloads the value of DMSRR and DMDRR.

Figure 17.15 shows address update in incremental address.

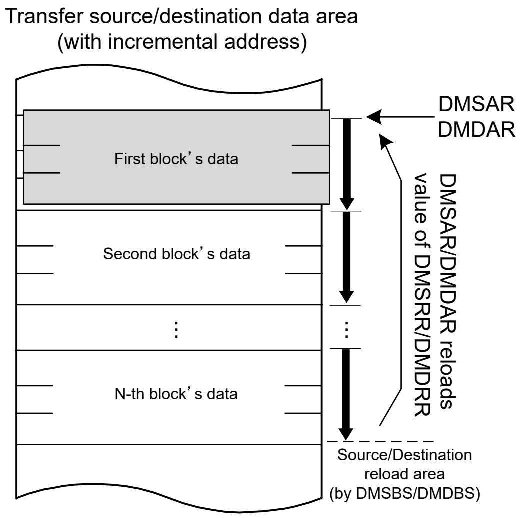


Figure 17.15 Address update in incremental address

17.3.5.3 Offset Addition Mode

When DMAMD.SM[1:0] is set to 01b, the address update mode of the source is offset addition. And when DMAMD.DM[1:0] is set to 01b, the address update mode of the destination is offset addition.

In offset addition, DMSBS and DMDBS indicates reload area and also works as an access offset value. Unlike other transfer modes, DMOFR register is not used in repeat-block transfer mode. In offset addition, the unit of DMSBS and DMDBS is the number of blocks. When the transfer starts, DMCRAL operates as a down counter, DMSAR and DMDAR reloads the value of DMSRR and DMDRR every time one block is transferred. In addition, DMSBSL and DMDBSL, which is the lower 16 bits of DMSBS and DMDBS, also operates as a down counter and decrements every time one block is transferred. When the DMSBS and DMDBS value becomes 1, DMSAR and DMDAR reloads the value of DMSRR and DMDRR.

When DMAMD.SADR and DMAMD.DADR is set to 0, offset addition operation of the same area is repeated. DMDAR only reloads DMDRR. Figure 17.16 shows address update in offset addition with DMAMD.SADR and DMAMD.DADR=0.

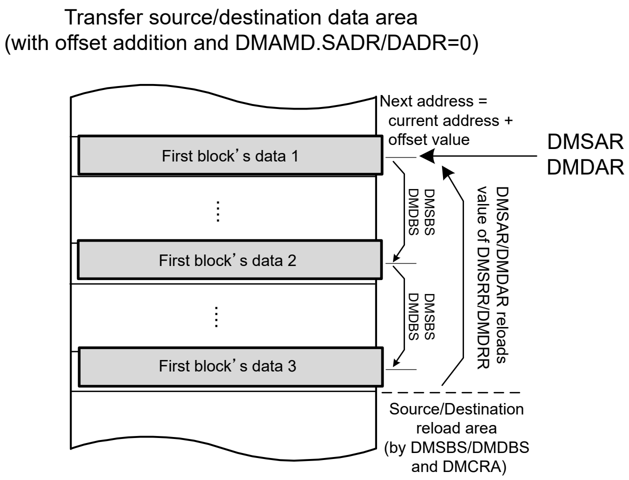


Figure 17.16 Address update in offset addition with DMAMD.SADR and DMAMD.DADR = 0

When DMAMD.SADR and DMAMD.DADR is set to 1, the address is incremented by one data unit after DMSRR and DMDRR is reloaded by DMCRAL=1. In other words, an index value ((DMDBSH-DMDBSL) × DataSize) is added to DMDAR after DMDRR is reloaded. This behavior is used to implement multiple ring buffers. Figure 17.17 shows address update in offset addition with DMAMD.SADR and DMAMD.DADR=1.

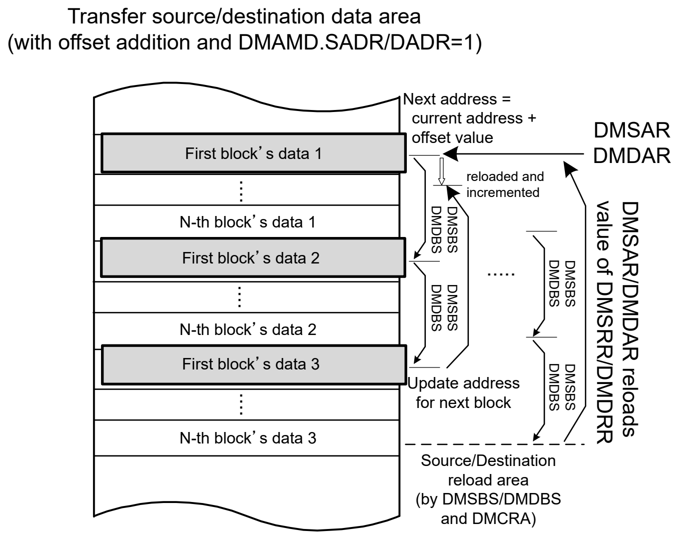


Figure 17.17 Address update in offset addition with DMAMD.SADR and DMAMD.DADR = 1

17.3.6 Example of Using Repeat-Block Transfer Mode

In repeat-block transfer mode, it is possible to realize repeated access to interval data and single or multiple ring buffers by combining the above address update modes. Following sections shows some usage examples.

17.3.6.1 Interval Address to Single Ring Buffer

Figure 17.18 shows an example of reading interval ADDRn registers (data register) of ADC16H module and storing it in single ring buffer. It transfers 2 data every 4 halfwords per 1 request. The block size (DMCRA) to 2, and the transfer source offset (DMSBS) to 4. Table 17.15 shows setting of this example.

Table 17.15 Setting of use case: from interval address to single ring buffer

Register	Value	Description
DMSAR, DMSRR	0x4033_A000	Initial source address
DMDAR, DMDRR	0x2200_0000	Initial destination address
DMTMD.SZ[1:0]	01b	Data size is halfword
DMAMD.SADR	0	DMSAR only reloads DMSRR
DMAMD.SM[1:0]	01b	Source update mode is offset addition
DMAMD.DM[1:0]	10b	Destination update mode is incremental address
DMCRAH, DMCRAL	2	Transfer block size
DMSBSH, DMSBSL	8	Source access offset (unit is 'data')
DMDBSH, DMDBSL	N × 2 (DMCRA)	Destination buffer size (unit is 'data')

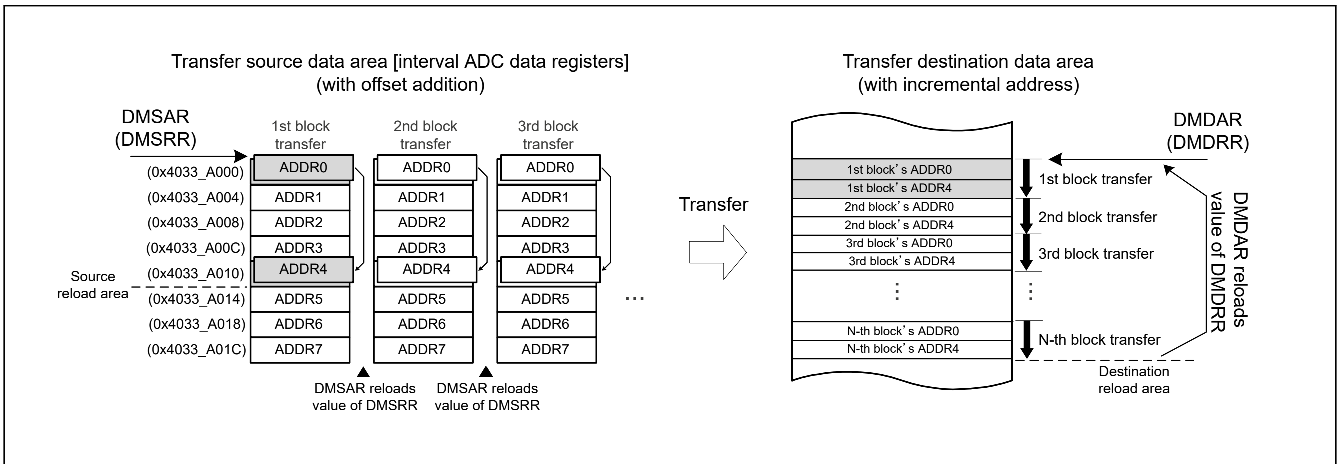


Figure 17.18 Example of use case: from interval address to single ring buffer

17.3.6.2 Single Block to Multi Ring Buffer

Figure 17.19 shows an example of storing the continuous ADDRn registers (data register) of ADC16H module individually in multiple ring buffers. In this example, a ring buffer in which only the first element (ADDR0) in a single block is arranged in transfer order is created at the destination. Also, in the next area, create a ring buffer in which only the second element (ADDR1) is arranged in transfer order. In the following case, create a ring buffer of length N, which is defined by DMDBS. And the number of data elements in the block is 3, which is defined by DMCRA. Table 17.16 shows setting of this example.

Table 17.16 Setting of use case: from single block to multi ring buffer (1 of 2)

Register	Value	Description
DMSAR, DMSRR	0x4033_A000	Initial source address
DMDAR, DMDRR	0x2200_0000	Initial destination address
DMTMD.SZ[1:0]	01b	Data size is halfword
DMAMD.DADR	1	Incremental destination address after reloading
DMAMD.SM[1:0]	01b	Source update mode is offset addition
DMAMD.DM[1:0]	01b	Destination update mode is offset addition

Table 17.16 Setting of use case: from single block to multi ring buffer (2 of 2)

Register	Value	Description
DMCRAH, DMCRAL	3	Transfer block size
DMSBSH, DMSBSL	2	Source buffer size (unit is ‘data’)
DMDBSH, DMDBSL	N	Destination whole buffer size (unit is ‘blocks’) and Destination access offset (unit is ‘data’)

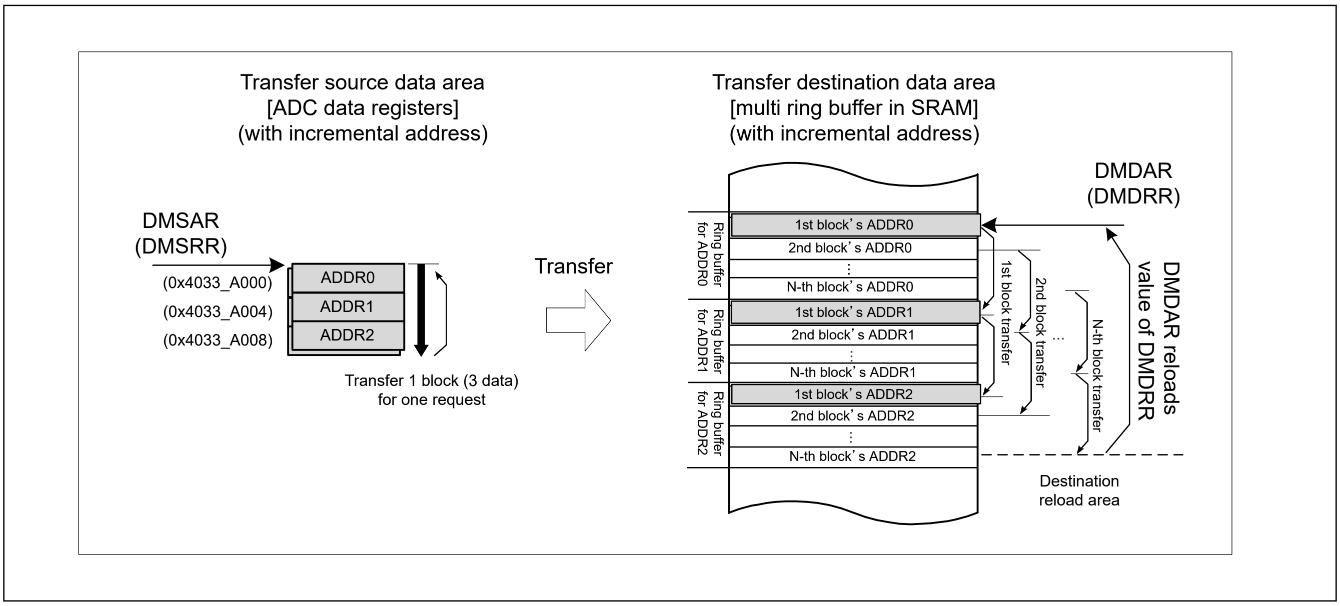


Figure 17.19 Example of use case: from single block to multi ring buffer

17.3.7 Activation Sources

Software, interrupt requests from the peripheral modules, and external interrupt requests can all be specified as DMAC activation sources. Set the DMTMD.DCTG[1:0] bits to select the activation source.

17.3.7.1 DMAC Activation by Software

When DMA transfer is started by software, follow below procedure.

1. Set the DMTMD.DCTG[1:0] bits to 00b.
2. Set the DMCNT.DTE bit to 1 (DMA transfer is enabled).
3. Set the DMAST.DMST bit set to 1 (DMAC activation enabled).
4. Set the DMREQ.SWREQ bit to 1 (DMA requested).

When the DMAC is activated by software while the DMREQ.CLRS bit is 0, the DMREQ.SWREQ bit is cleared to 0 after data transfer is started in response to a DMA transfer request.

When the DMAC is activated by software while the CLRS bit is 1, the SWREQ bit is not cleared to 0 after data transfer is started. In this case, a DMA transfer request is issued again after completion of a transfer.

17.3.7.2 DMAC Activation through Interrupt Requests from On-Chip Peripheral Modules or External Interrupt Requests

You can specify interrupt requests from on-chip peripheral modules and external interrupt requests as DMAC activation sources. The activation sources can be selected individually for each channel in the ICU.DELSRn.DELS[9:0] (n = 0 to 7) bits.

To start DMA transfer through an interrupt request from an on-chip peripheral module or an external interrupt request, follow the procedures as indicated below.

1. Set the ICU.DELSRn.DELS[9:0] (n = 0 to 7) bits to the event number (select the DMAC event link).

2. Set the DMTMD.DCTG[1:0] bits to 01b (interrupts from the peripheral modules and the external interrupt pins).
3. Set the DMCNT.DTE bit to 1 (enable DMA transfer).
4. Set the DMAST.DMST bit set to 1 (DMAC activation enabled)

For interrupt requests specified as DMAC activation sources, see Table 14.3 in section 14, Interrupt Controller Unit (ICU).

17.3.8 Operation Timing

The following timing charts show the minimum number of execution cycles.

Figure 17.20 and Figure 17.21 show DMAC operation timing examples.

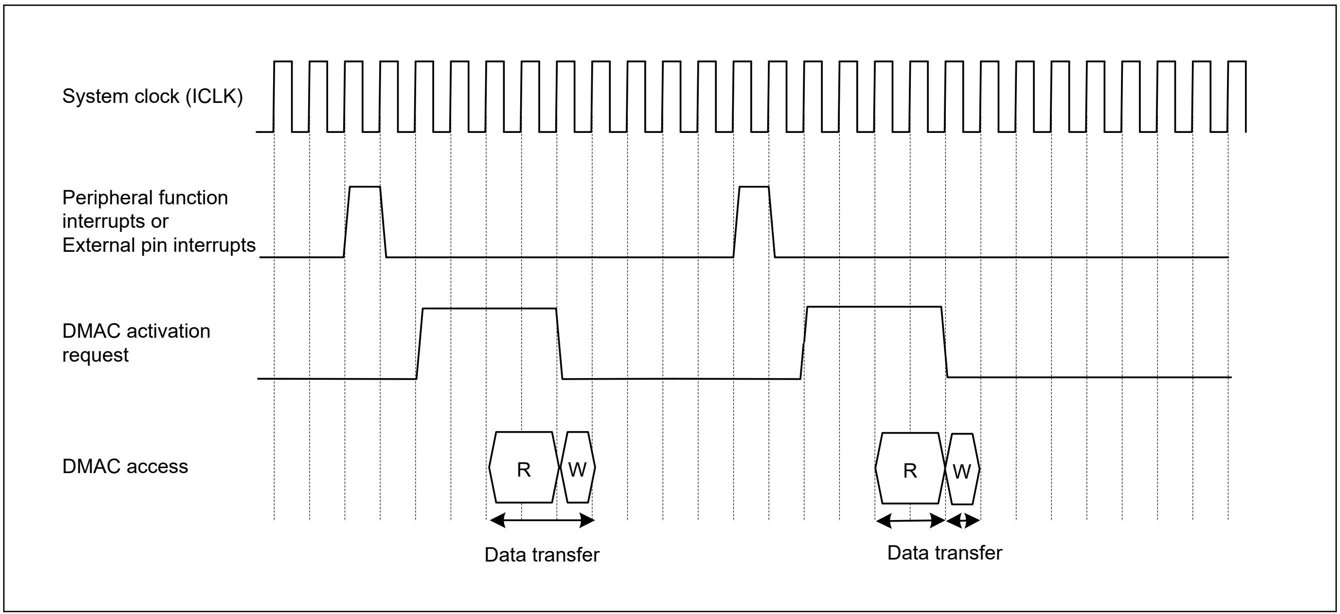


Figure 17.20 DMAC operation timing example 1 with DMAC activation by Interrupt from peripheral module or external interrupt input pin, in normal transfer mode or repeat transfer mode

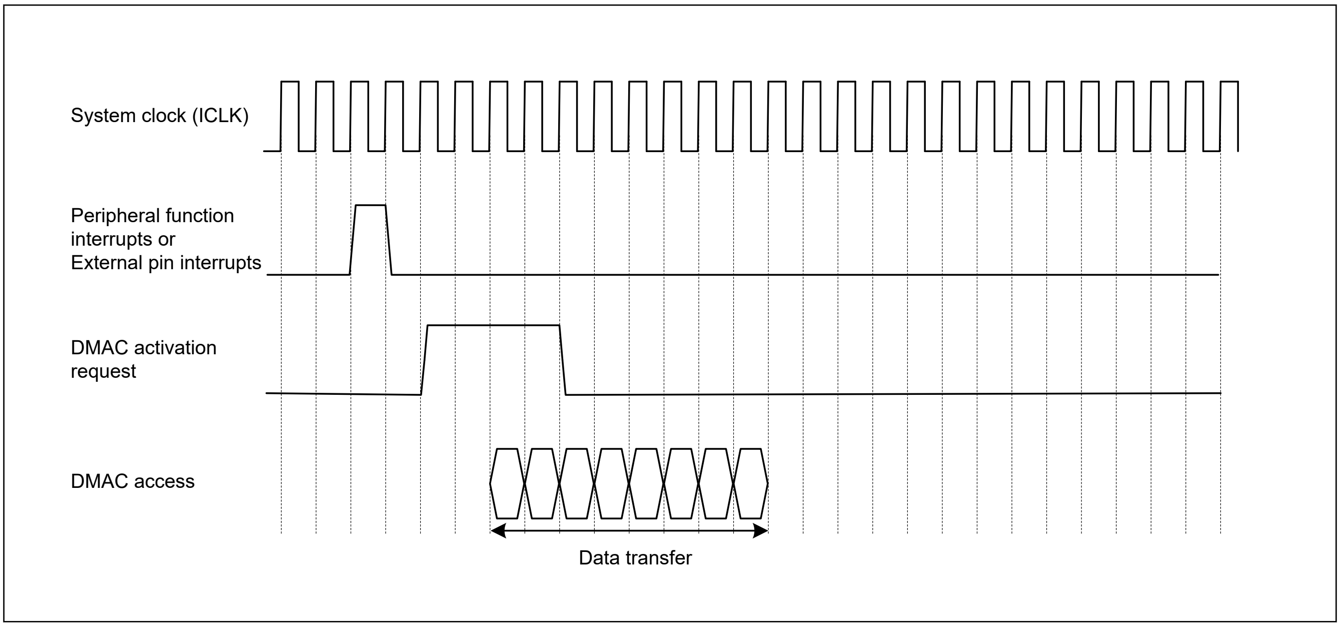


Figure 17.21 DMAC operation timing example 2 with DMAC activation by interrupt from peripheral module or external interrupt input pin, in block transfer mode with block size = 4

17.3.9 Activating the DMAC

Table 17.17 shows the register setting procedure of normal, repeat and block transfer mode and Table 17.18 shows register setting procedure of repeat-block transfer mode.

Table 17.17 Register Setting Procedure of Normal Transfer Mode, Repeat Transfer Mode and Block Transfer Mode

No.	Step Name	Description
1	Disable the peripheral function as the DMAC request source.	To use peripheral function interrupts as DMAC activation sources. Disable the control register for the peripheral function.
2	Disable the IRQn pin as the DMAC request source.	To use external pin interrupts as DMAC activation sources.
3	Set the ICU.DELSRn.DELS[9:0] bits to 0x00.	Disable the DMAC request.
4	Clear the DMCNT.DTE bit to 0	Disable DMA transfer.
5	Set the interrupt request as a DMAC request source in the DELSRn register by using the ICU.	To use internal peripheral interrupts or external pin interrupts as DMAC activation sources. Enable the interrupt bit for the activation source. Set the DMAC activation source.
6	Set the peripheral module as a DMAC request source	To use peripheral function interrupt as a DMAC activation source. Set the control register for the peripheral function without starting it.
7	Set the IRQn pin function by using the ICU.	To use external pin interrupt as a DMAC activation source. Set the IRQn pin function by using the Interrupt Controller Unit.
8	Set the DMAMD.DM[1:0] bits Set the DMAMD.SM[1:0] bits Set the DMAMD.DARA[4:0] bits Set the DMAMD.SARA[4:0] bits	Set the Transfer destination address update mode bits Set the Transfer source address update mode bits Set the Transfer destination address extended repeat area bits Set the Transfer source address extended repeat area bits
9	Set the DMTMD.DCTG[1:0] bits Set the DMTMD.SZ[1:0] bits Set the DMTMD.DTS[1:0] bits Set the DMTMD.MD[1:0] bits Set the DMTMD.TKP bit	Set the Transfer request select bits Set the Data transfer size bits Set the Repeat area select bits Set the Transfer mode select bits Set the transfer keeping select bit
10	Set the DMSAR register Set the DMDAR register Set the DMCRA register	Set the transfer source start address. Set the transfer destination start address. Set the number of transfer operations.
11	Set the DMCRB register	To use block transfer mode or repeat transfer mode. Set the number of block transfer operations.
12	Set the DMOFR register	To use the address update function with offset. Set the offset value.
13	Set the DMINT.DTIE bit to 1	To use the DMA transfer end interrupts. Enable DMAC transfer end interrupts.
14	Set the DMINT.RPTIE bit Set the DMINT.SARIE bit Set the DMINT.DARIE bit Set the DMINT.ESIE bit to 1	To use the DMA transfer escape end interrupts Set the repeat size end interrupt. Set the transfer source address extended repeat area overflow interrupt. Set the transfer destination address extended repeat area overflow interrupt. Enable the DMA transfer escape end interrupt.
15	Set the DMCNT.DTE bit to 1	Enable DMA transfer.
16	Set the DMAST.DMST bit to 1	Enable DMAC operation. *1 Common settings for DMAC
17	Start the peripheral function as a DMAC request source	To use peripheral function interrupt as a DMAC activation source
18	Enable the IRQn pin as a DMAC request source	To use external pin interrupt as a DMAC activation source
19	End of initial settings	For activation by software On completion of the initial settings, writing 1 to the DMA software start bit (DMREQ.SWREQ) starts DMA transfer.

Note 1. The DMAST.DMST bit setting does not necessarily have to follow the settings for the individual activation sources.

Table 17.18 Register Setting Procedure of Repeat-Block Transfer Mode

No.	Step Name	Description
1	Disable the peripheral function as the DMAC request source.	To use peripheral function interrupts as DMA activation sources. Disable the control register for the peripheral function.
2	Disable the IRQ pin as the DMAC request source.	To use external pin interrupts as DMA activation sources.
3	Set the ICU.DELSRn.DELS[9:0] bits to 0x00.	Disable the DMAC request.
4	Clear the DMCNT.DTE bit to 0	Disable DMAC transfer.
5	Set the interrupt request as a DMAC request source in the DELSRn register by using the ICU.	To use internal peripheral interrupts or external pin interrupts as DMA activation sources. Enable the interrupt bit for the activation source. Set the DMAC activation source.
6	Set the peripheral module as a DMAC request source	To use peripheral function interrupt as a DMA activation source. Set the control register for the peripheral function without starting it.
7	Set the IRQ pin function by using the Interrupt Controller Unit.	To use external pin interrupt as a DMA activation source. Set the IRQ pin function by using the Interrupt Controller Unit.
8	Set the DMAMD.DM[1:0] bits Set the DMAMD.SM[1:0] bits Set the DMAMD.DADR bit Set the DMAMD.SADR bit	Set the Transfer destination address update mode bits Set the Transfer source address update mode bits Set the Transfer destination address update select after reload Set the Transfer source address update select after reload
9	Set the DMTMD.DCTG[1:0] bits Set the DMTMD.SZ[1:0] bits Set the DMTMD.MD[1:0] bits Set the DMTMD.TKP bit	Set the Transfer request select bits Set the Data transfer size bits Set the Transfer mode to repeat-block transfer mode Set the transfer keeping select bit
10	Set the DMSAR register Set the DMDAR register Set the DMSRR register Set the DMDRR register Set the DMCRA register Set the DMCRB register	Set the transfer source start address Set the transfer destination start address Set the initial value of source start address Set the initial value of destination start address Set the number of transfer operations Set the number of block transfer operations
11	Set the DMSBS register Set the DMDBS register	To use the address update function with incremental, decremental or offset Set the source buffer size and access offset Set the destination buffer size and access offset
12	Set the DMINT.DTIE bit to 1	To use DMA transfer end interrupts. Enable DMAC transfer end interrupts.
13	Set the DMCNT.DTE bit to 1	Enable DMAC transfer
14	Set the DMAST.DMST bit to 1	Enable DMAC operation. *1
15	Start the peripheral function as a DMAC request source	To use peripheral function interrupt as a DMA activation source
16	Enable the IRQ pin as a DMAC request source	To use external pin interrupt as a DMA activation source
17	End of initial settings	For activation by software On completion of the initial settings, writing 1 to the DMA software start bit (DMREQ.SWREQ) starts DMA transfer.

Note 1. The DMAST.DMST bit setting does not necessarily have to follow the settings for the individual activation sources.

17.3.10 Starting DMA Transfer

To enable the DMA transfer, set the DMCNT.DTE bit to 1 (enable the DMA transfer), and then set the DMAST.DMST bit to 1 (enable the DMAC activation).

New activation requests are not accepted during the transfer of another DMAC channel or DTC. When the preceding transfer is complete, channel arbitration selects the DMA transfer request of the highest priority channel, and the DMA transfer of that channel starts. When the DMA transfer starts, the DMSTS.ACT flag is set to 1 (the DMAC is in the active state).

17.3.11 Registers During DMA Transfer

The DMAC registers are updated by a DMA transfer. The value to be updated differs according to the other settings and the transfer state. The registers to be updated are DMSAR, DMDAR, DMCRA, DMCRB, DMSBS, DMDBS, DMCNT, and DMSTS.

DMA Source Address Register (DMSAR)

When data has been transferred in response to one transfer request, the contents of DMSAR are updated to the address to be accessed by the next transfer request.

For details on register update operation in each transfer mode, see Table 17.5 to Table 17.13.

DMA Destination Address Register (DMDAR)

When data has been transferred in response to one transfer request, the contents of DMDAR are updated to the address to be accessed by the next transfer request.

For details on register update operation in each transfer mode, see Table 17.5 to Table 17.13.

DMA Transfer Count Register (DMCRA)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.

For details on register update operation in each transfer mode, see Table 17.5 to Table 17.13.

DMA Block Transfer Count Register (DMCRB)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.

For details on register update operation in each transfer mode, see Table 17.5 to Table 17.13.

DMA Source Buffer Size Register (DMSBS)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.

For details on register update operation in each transfer mode, see Table 17.8 to Table 17.13.

DMA Destination Buffer Size Register (DMDBS)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.

For details on register update operation in each transfer mode, see Table 17.8 to Table 17.13.

DMA Transfer Enable Bit (DMCNT.DTE)

Although the DMCNT.DTE bit enables or disables data transfer by the register write access, it is automatically cleared to 0 by the DMAC according to the DMA transfer state.

The conditions for clearing this bit by the DMAC are as follows:

- When the specified total volume of data transfer is completed
- When DMA transfer is stopped by the repeat size end interrupt
- When DMA transfer is stopped by the extended repeat area overflow interrupt
- When DMA transfer error occurs

Writing to the registers for the channels when the corresponding DMCNT.DTE bit is set to 1 is prohibited (except for DMCNT). In this case, writing must be performed after the bit is cleared to 0.

DMAC Active Flag (DMSTS.ACT)

The DMSTS.ACT flag indicates whether the DMACn is in the idle or active state.

This flag is set to 1 when the DMAC starts data transfer, and is cleared to 0 when data transfer in response to one transfer request is completed.

Even when DMA transfer is stopped by writing 0 to the DMCNT.DTE bit during DMA transfer, this flag remains 1 until DMA transfer is completed.

Transfer End Interrupt Flag (DMSTS.DTIF)

The DMSTS.DTIF flag is set to 1 after DMA transfer of the total transfer size of data is completed.

When both this flag and the DMINT.DTIE bit are set to 1, a transfer end interrupt is requested.

This flag is set to 1 when the DMA transfer bus cycle is completed and the DMSTS.ACT flag is cleared to 0 indicating the DMA transfer end.

This flag is automatically cleared to 0 when the DMCNT.DTE bit is set to 1 during the interrupt handling.

Transfer Escape End Interrupt Flag (DMSTS.ESIF)

The DMSTS.ESIF flag is set to 1 when a repeat size end interrupt or extended repeat area overflow interrupt is requested. When this bit and the DMINT.ESIE bit are set to 1, a transfer escape end interrupt is requested.

This flag is set to 1 when the bus cycle of the DMA transfer having caused the interrupt request is completed and the DMSTS.ACT flag is cleared to 0 indicating the DMA transfer end.

This flag is automatically cleared to 0 when the DMCNT.DTE bit is set to 1 during an interrupt handling.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set.

For details, see section 14, Interrupt Controller Unit (ICU).

17.3.12 Channel Priority

When multiple DMA transfer requests and a DTC transfer request are present, the DMAC and DTC determine the priority of channels that have DMA transfer requests and a DTC transfer request.

- When DMCTL.PR = 0, the channel priority is fixed as follows: channel 0 > channel 1 > channel 2 > channel 3 … > channel 7 > DTC (channel 0: Highest).

When DMCTL.PR = 1, the DMAC and DTC determine the priority of channels at round-robin.

When a DMA transfer request is generated during data transfer, channel arbitration is started after the final data has been transferred, and DMA transfer of the higher-priority channel starts.

17.3.13 Channel Security

The secure attribute can be set with CPSCU.DMACCHSAR.SADMACm0n (m = 0, 1, n = 0 to 7) for each DMAC channel.

When the CPSCU.DMACCHSAR.SADMACm0n bit is 0.

- When the corresponding channel transfers, it behaves as a secure master.
- The register of corresponding channel has a secure attribute.

When the CPSCU.DMACCHSAR.SADMACm0n bit is 1.

- When the corresponding channel transfers, it behaves as a non-secure master.
- The register of corresponding channel has a non-secure attribute.

Refer to the Security chapter and BUS chapter for areas accessible to secure and non-secure masters.

Do not change to the CPSCU.DMACCHSAR while DMA transfer.

17.3.14 Channel Privilege

The privileged attribute can be set with CPSCU.DMACCHPAR.PADMACm0n (m = 0, 1, n = 0 to 7) for each DMAC channel n.

When the CPSCU.DMACCHPAR.PADMACm0n bit is 0.

- When the corresponding channel transfers, it behaves as a privileged master.
- The registers of channel are protected from an unprivileged access.

When the CPSCU.DMACCHPAR.PADMACm0n bit is 1.

- The transfer of DMAC channel is unprivileged access for both read and write.
- The registers of channel are unprivileged attributes.

Do not change to the CPSCU.DMACCHPAR while DMA transfer.

17.4 Ending DMA Transfer

The operation for ending DMA transfer depends on the transfer end conditions. When DMA transfer ends, the DMCNT.DTE bit and the DMSTS.ACT flag are changed from 1 to 0, indicating that DMA transfer has ended.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set.

For details, see section 14, Interrupt Controller Unit (ICU).

17.4.1 Transfer End by Completion of Specified Total Number of Transfer Operations

(1) In Normal Transfer Mode (DMTMD.MD[1:0] = 00b)

When the value of DMCRAL changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMCNT.DTE bit is cleared to 0 and the DMSTS.DTIF flag is set to 1 at the same time. If the DMINT.DTIE bit is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

(2) In Repeat Transfer Mode (DMTMD.MD[1:0] = 01b)

When the value of DMCRBL changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMCNT.DTE bit is cleared to 0 and the DMSTS.DTIF flag is set to 1 at the same time. If the DMINT.DTIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC.

If the DMTMD.TKP bit is 1 (in free-running function), the DMSTS.DTIF bit is set to 1, but the DMCNT.DTE bit is not cleared to 0.

(3) In Block Transfer Mode (DMTMD.MD[1:0] = 10b)

When the value of DMCRBL changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMCNT.DTE bit is cleared to 0 and the DMSTS.DTIF flag is set to 1 at the same time. If the DMINT.DTIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC.

If the DMTMD.TKP bit is 1 (in free-running function), the DMSTS.DTIF bit is set to 1, but the DMCNT.DTE bit is not cleared to 0.

(4) In Repeat-Block Transfer Mode (DMTMD.MD[1:0] = 11b)

When the value of DMCRBL changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMCNT.DTE bit is cleared to 0 and the DMSTS.DTIF flag is set to 1 at the same time. If the DMINT.DTIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC.

If the DMTMD.TKP bit is 1 (in free-running function), the DMSTS.DTIF bit is set to 1, but the DMCNT.DTE bit is not cleared to 0.

17.4.2 Transfer End by Repeat Size End Interrupt

In repeat transfer mode, a repeat size end interrupt is requested when transfer of a 1-repeat size of data is completed while the DMINT.RPTIE bit is set to 1. When the interrupt is requested to complete DMA transfer, the DMCNT.DTE bit is cleared to 0 and the DMSTS.ESIF flag is set to 1 even if the DMTMD.TKP bit is 1 (in free-running function). If the DMINT.ESIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC. Here, the transfer can be resumed by writing 1 to the DMCNT.DTE bit.

A repeat size end interrupt can be requested also in block transfer mode. In block transfer mode, the interrupt is requested in the same way as in repeat transfer mode when transfer of a 1-block size data is completed.

Repeat size end interrupt cannot be requested in repeat-block transfer mode.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 14, Interrupt Controller Unit (ICU).

17.4.3 Transfer End by Interrupt on Extended Repeat Area Overflow

When an overflow on the extended repeat area occurs while the extended repeat area is specified and the DMINT.SARIE or DMINT.DARIE bit is set to 1 even if the DMTMD.TKP bit is 1 (in free-running function), an interrupt by an extended repeat area overflow is requested. When the interrupt is requested, the DMA transfer is terminated, the DMCNT.DTE bit is cleared to 0, and the ESIF flag in DMSTS is set to 1. If the DMINT.ESIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC.

Even if an interrupt by an extended repeat area overflow is requested during a read cycle, the following write cycle is performed.

In block transfer mode, even if an interrupt by an extended repeat area overflow is requested during a 1-block transfer, the remaining data in the block is transferred; transfer is terminated after a block transfer.

An interrupt by an extended repeat area overflow cannot be requested in repeat-block transfer mode.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 14, Interrupt Controller Unit (ICU).

17.5 Processing on DMA Transfer Error

DMA transfer error occurs with MSAU error, the Slave TrustZone Filter error, the Master MPU error, the Slave Bus Error or the Illegal Access Error. If the access error occurs during the DMA transfer, the DMAC immediately stops the transfer of error occurred channel. If there is a request other than the channel which caused the error, it will be re-arbitration as it is.

When the transfer error occurs, DMCNT.DTE of the error causing channel is set to 0. Also, ICU.DELSRn of the corresponding channel is cleared. Write back to each register is not performed. The information of the channel that caused the error is set in DMECHR.

17.6 Interrupts

17.6.1 Transfer End Interrupt

Each DMAC channel can output an interrupt request DMACmn_INT (m = 0, 1, n = 0 to 7) to the CPU or the DTC after transfer in response to one request is completed.

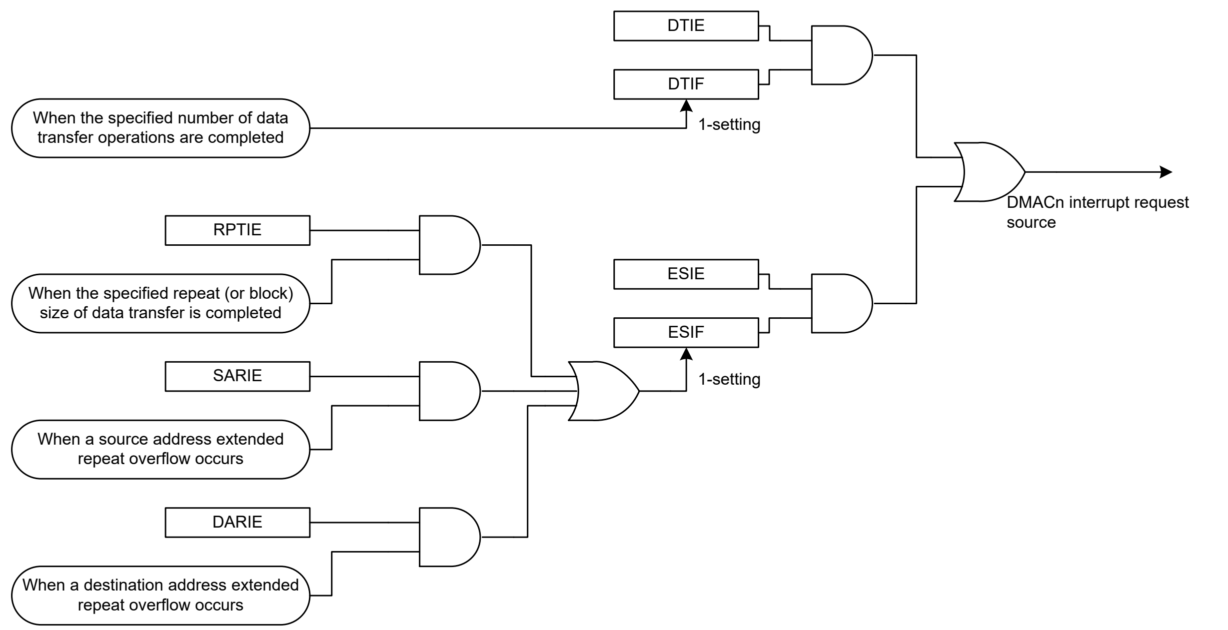
The interrupt request is generated after the bus response to the final write of the DMA transfer. The timing of the bus response to a DMA write access depends on the DMBWR.BWE and transfer destination. For details, see section 17.9.1. Bus Response to DMA Write Access.

In repeat-block transfer mode, do not enable escape transfer end interrupt.

Table 17.19 lists the relation among the interrupt sources, the interrupt status flags, and the interrupt enable bits. Figure 17.22 shows the schematic logic diagram of interrupt outputs (DMACn (n = 0 to 7)). Figure 17.23 shows the DMAC interrupt handling routine to resume/terminate DMA transfer.

Table 17.19 Relation among interrupt sources, interrupt status flags, and interrupt enable bits

Interrupt sources		Interrupt enable bits	Interrupt status flags	Request output enable bits
Transfer end		—	DMSTS.DTIF	DMINT.DTIE
Escape transfer end	Repeat size end	DMINT.RPTIE	DMSTS.ESIF	DMINT.ESIE
	Source address extended repeat area overflow	DMINT.SARIE		
	Destination address extended repeat area overflow	DMINT.DARIE		



Note: n = 0 to 7

Figure 17.22 Schematic logic diagram of interrupt output source (DMACn)

Specifically, the different procedures are used for canceling an interrupt to restart DMA transfer in the following two cases:

- When terminating a DMA transfer
- When continuing a DMA transfer

17.6.1.1 When Terminating a DMA Transfer

Write 0 to the DMSTS.DTIF flag to clear a transfer end interrupt, and to the DMSTS.ESIF flag to clear a repeat size interrupt and an extended repeat area overflow interrupt. The DMACn remains in the stop state. When starting another DMA transfer after that, set the appropriate registers, and set the DMCNT.DTE bit to 1 (DMA transfer enabled).

17.6.1.2 When Continuing a DMA Transfer

Write 1 to the DMCNT.DTE bit. The DMSTS.ESIF flag is automatically cleared to 0 (interrupt source cleared), and DMA transfer is resumed.

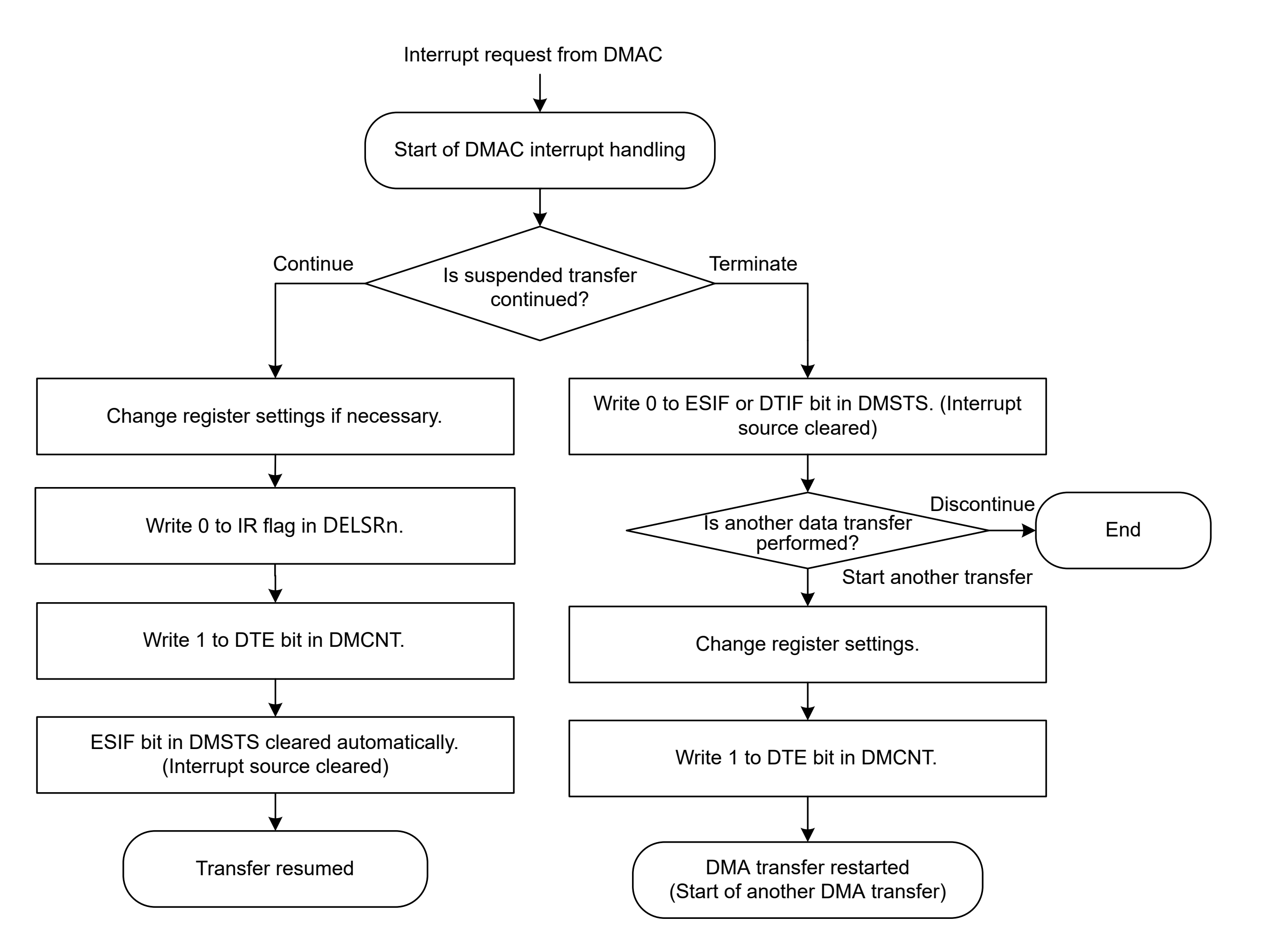


Figure 17.23 DMAC Interrupt Handling Routine to Resume/Terminate DMA Transfer

17.6.2 Transfer Error Interrupt

Error response detection interrupt request DMAm_TRANSERR (m = 0, 1) is generated from the DMAC/DTC when the transfer error is detected during DMAC transfer.

The types of interrupts that occur when a DMAC transfer error occurs are listed in the Table 17.20. The Table 17.20 also shows error information stored when a transfer error occurs.

Table 17.20 Interrupt and error information due to DMAC transfer error cause

Transfer error factor	NMI/RESET*1 Request	Interrupt Request	Bus Error Status	Error Channel Information
MSAU Error	ICU.NMISR.BUSST	DMAm_TRANSERR*1	BUS.BUSERRSTATDMACm.MESRRSTAT	DMAC. DMECHR
Master MPU Error	ICU.NMISR.BUSST	DMAm_TRANSERR*1	BUS.BUSERRSTATDMACm.MMERRSTAT	DMAC. DMECHR
Illegal Access Error	ICU.NMISR.BUSST	DMAm_TRANSERR*1	BUS.BUSERRSTATDMACm.ILERRSTAT	DMAC. DMECHR
Slave Bus Error	ICU.NMISR.BUSST	DMAm_TRANSERR*1	BUS.BUSERRSTATDMACm.SLERRSTAT	DMAC. DMECHR
TrustZone Filter error	ICU.NMISR.BUSST	DMAm_TRANSERR*1	BUS.BUSERRSTATDMACm.SLERRSTAT	DMAC. DMECHR

Note 1. If ICU.NMIER.BUSEN is enabled with DMAm_TRANSERR set in ICU.IELSR, NMI and an interrupt will occur due to a transfer error caused by DMA. Only NMI can be generated by not setting DMAm_TRANSERR to ICU.IELSR.

17.7 Event Link

Each DMAC channel outputs an event link request each time one data transfer is completed in normal transfer mode or repeat transfer mode, and each time one block transfer is completed in block transfer mode or repeat-block transfer mode.

The event link request DMACmn_INT (m = 0, 1, n = 0 to 7) is generated when the bus responds to the last destination access of DMA transfer.

The timing of the bus response to a DMA write access depends on the DMBWR.BWE and transfer destination. For details, see section 17.9.1. Bus Response to DMA Write Access.

For details, see section 19, Event Link Controller (ELC).

17.8 Low-Power Consumption Function

Before entering the module-stop state, Software Standby mode or Deep Software Standby mode, you must first set the DMAST.DMST bit to 0 (the DMAC module suspended) and use the settings in the sections that follow.

(1) Module-stop function

Setting the module-stop bit to 1 stops the DMAC corresponding to that module-stop bit.

If the module-stop bit is set to 1 while DMAC transfers data, the DMAC corresponding to that bit becomes the module-stop state after completing it.

When the module-stop bit is 1, access to the register of the DMAC corresponding to that module-stop bit is prohibited.

(2) Software Standby mode and Deep Software Standby mode

Use the settings described in section 11.6.2.1. Entering Software Standby Mode, or in section 11.6.3.1. Transition to Deep Software Standby Mode .

If DMA transfer operations are in progress when the WFI instruction is executed, the DMA transfer completes before the transition to Software Standby mode or Deep Software Standby mode.

(3) Notes on low power consumption function

For information on the WFI instruction and register settings, see section 11.8.7. Timing of WFI Instructions.

To perform a DMA transfer after returning from a low power mode, set the DMAST.DMST bit to 1 again. To use a request that is generated in Software Standby mode as an interrupt request to the CPU but not as a DMAC startup request, specify the CPU as the interrupt request destination, as described in section 14.4.1. Interrupt Detection Selection, and then execute the WFI instruction.

17.9 Usage Notes

17.9.1 Bus Response to DMA Write Access

The timing of the bus response to a DMA write access depends on the DMBWR.BWE setting and the destination of the transfer. When DMBWR.BWE is "1", the bus response is returned from the intermediate point described in section 15.2.5.2. Bufferable Write Access in the bus chapter. Also, when writing to an access destination described in section 15.2.5.3. Notes on Writing Completion Timing in the bus chapter, a bus response is returned before the write is completed.

17.9.2 Access to the Registers during DMA Transfer

Do not write to the following registers while the DMSTS.ACT flag of the same channel is set to 1 (DMAC active state) or the DMCNT.DTE bit of the same channel is set to 1 (DMA transfer enabled):

- ICU.DELSR
- DMSAR
- DMDAR
- DMCRA
- DMCRB
- DMTMD
- DMINT
- DMAMD
- DMOFR

- DMSBS
- DMDBS
- DMSRR
- DMDRR
- DMBWR
- ICUSARC
- DMACSAR
- DMCTL

17.9.3 DMA Transfer to Reserved Areas

DMA transfer to the reserved areas is prohibited. If such an access is made, transfer results are not guaranteed. For details on the reserved areas, see section 5, Address Space.

17.9.4 Setting of DMAC Event Link Setting Register of the Interrupt Controller Unit (ICU.DELSRn)

The DMAC event link setting register (ICU.DELSRn) should be set while the DMA transfer enable bit (DMCNT.DTE) is cleared to 0 (DMA transfer is disabled). Moreover, the DTC activation enable register (ICU.IELSRn.DTCE) that corresponds to the same event number that has been set by the ICU.DELSRn register should not be set to 1. For details on the ICU.IELSRn.DTCE, see section 14, Interrupt Controller Unit (ICU).

17.9.5 Suspending or Restarting DMAC Activation

To suspend a DMAC activation request, write 0x000 to the DMAC Event Link select bits (ICU.DELSRn.DELS[9:0]). To restart the DMA transfer, write the event number to the ICU.DELSRn.DELS[9:0] bits following the settings shown in section 17.3.9. Activating the DMAC.

17.9.6 Precautions for Resuming DMA Transfer

A DMAC activation request might occur in the next request after a DMA transfer completes. If this happens, the DMA transfer starts and the DMAC activation request is held in the DMAC. To prevent this, stop the DMAC activation requests by setting the ICU.DELSRn.DELS[9:0] bits to 0.

When a DMAC activation request occurs after the last round of the DMA transfer is generated, clear the DMAC activation request with the following steps:

1. Set the DMCNT.DTE bit to 0.
2. Set the ICU.DELSRn.IR flag to 0.

18. Data Transfer Controller (DTC)

18.1 Overview

A Data Transfer Controller (DTC) module is provided for transferring data when activated by an interrupt request.

Table 18.1 lists the DTC specifications and Figure 18.1 shows DTC block diagram.

Table 18.1 DTC specifications

Item	DTC0 (for CPU0)/DTC1 (for CPU1)
Transfer modes	● Normal transfer mode A single activation leads to a single data transfer. ● Repeat transfer mode A single activation leads to a single data transfer. The transfer address returns to the start address after the number of data transfers reaches the specified repeat size. The maximum number of repeat transfers is 256 and the maximum data transfer size is 256 × 64 bits (2048 bytes). ● Block transfer mode A single activation leads to a transfer of a single block. The maximum block size is 256 × 64 bits = 2048 bytes.
Transfer channel	● Channel transfer corresponding to the interrupt source is possible (transferred by a DTC activation request from the ICU). ● Multiple data units can be transferred on a single activation source (chain transfer). ● Chain transfers are selectable to either execute when the counter is 0 or always execute.
Transfer space	● 4 GB area from 0x0000_0000 to 0xFFFF_FFFF excluding reserved areas.
Data transfer units	● Single data unit: 1 byte (8 bits), 1 halfword (16 bits), 1 word (32 bits), 1 doubleword (64 bits). ● Single block size: 1 to 256 data units.
CPU interrupt source	● An interrupt request can be generated to the CPU on a DTC activation interrupt ● An interrupt request can be generated to the CPU after a single data transfer ● An interrupt request can be generated to the CPU after a data transfer of a specified volume.
Processing on DTC transfer error	● When the DTC transfer error occurs, it stops the transfer that caused the error ● Request to clear the register for activation request of DTC error number to ICU
Error response detection interrupt	Generated when the DTC transfer error occurs
Event link function	Generated after one data transfer (for block, after one block transfer)
Read skip	Read of transfer information can be skipped
Write-back skip	When the transfer source or destination address is specified as fixed, a write-back of transfer information can be skipped
Module-stop function	Module-stop state can be set to reduce power consumption
TrustZone Filter	Security attribution and privilege attribution can be set for each activation source.

Note: Security attribution Register of DTC is described in ICU.ICUSARG, ICU.ICUSARH, ICU.ICUSARI, ICU.ICUSARJ, ICU.ICUSARK and ICU.ICUSARL

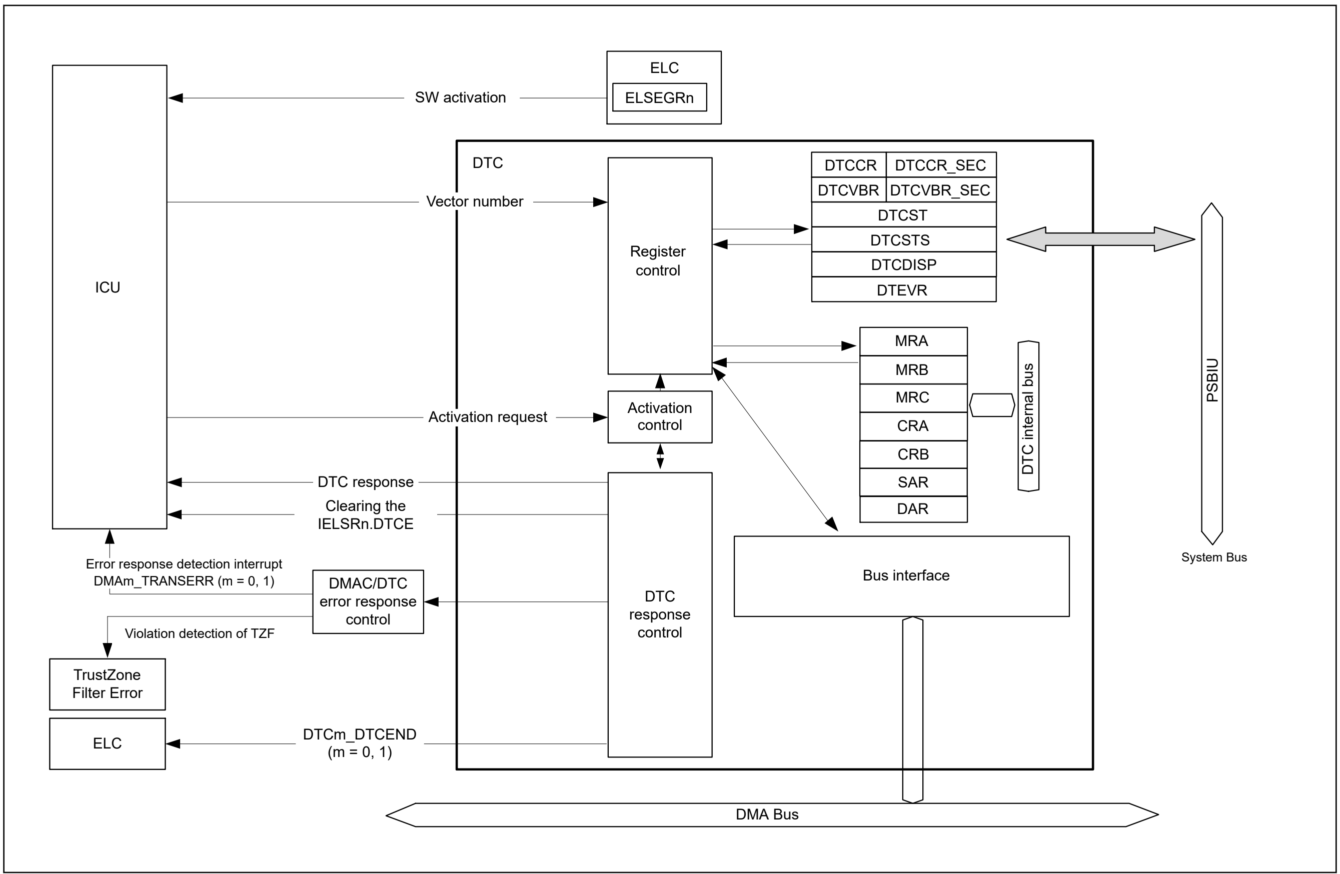


Figure 18.1 DTC block diagram

See section 14.1. Overview in section 14, Interrupt Controller Unit (ICU) for the connections between the DTC and NVIC in the CPU.

18.2 Register Descriptions

MRA, MRB, MRC, SAR, DAR, CRA, and CRB are all DTC internal registers that cannot be directly accessed from the CPU. Values to be set in these DTC internal registers are placed in the SRAM area as transfer information. When an activation request is generated, the DTC reads the transfer information from the SRAM area and sets it in its internal registers. After the data transfer ends, the internal register contents are written back to the SRAM area as transfer information.

The DTC0 and DTC1 registers have the same base address. DTC0 is dedicated to CPU0 and can only be accessed from CPU0. DTC1 is dedicated to CPU1 and can only be accessed from CPU1.

18.2.1 DTCSAR : DTC Controller Security Attribution Register

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x30

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DTCSTSA1
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DTCSTSA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DTCSTSA0	DTC0 Security Attribution 0: Secure 1: Non-secure	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W
16	DTCSTSA1	DTC1 Security Attribution 0: Secure 1: Non-secure	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

This register sets the DTCST and DTCSTS security attribution.

DTCSTSA0 bit (DTC0 Security Attribution)

This bit specifies security attribution of registers for DTC0.DTCST and DTC0.DTCSTS.

Do not write to the DTCSTSA0 bit while DTC0 transfer is enabled or a bus master is writing to the DTC0 registers.

DTCSTSA1 bit (DTC1 Security Attribution)

This bit specifies security attribution of registers for DTC1.DTCST and DTC1.DTCSTS.

Do not write to the DTCSTSA1 bit while DTC1 transfer is enabled or a bus master is writing to the DTC1 registers.

18.2.2 MRA : DTC Mode Register A

Base address: DTCVBR_SEC
DTCVBR

Offset address: 0x03 + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 18.3.1. Allocating Transfer Information and DTC Vector Table.)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MD[1:0]		SZ[1:0]		SM[1:0]		—	WBDIS
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
0	WBDIS	Write Back Disable 0: Write back to the DTC transfer information at the end of DTC data transfer. 1: Do not write back to the DTC transfer information at the end of DTC data transfer.	—
1	—	The read value is undefined. The write value should be 0.	—
3:2	SM[1:0]	Transfer Source Address Addressing Mode 0 0: Address in the SAR register is fixed (write-back to SAR is skipped.) 0 1: Address in the SAR register is fixed (write-back to SAR is skipped.) 1 0: SAR value is incremented after data transfer: +1 when SZ[1:0] = 00b +2 when SZ[1:0] = 01b +4 when SZ[1:0] = 10b +8 when SZ[1:0] = 11b 1 1: SAR value is decremented after data transfer: -1 when SZ[1:0] = 00b -2 when SZ[1:0] = 01b -4 when SZ[1:0] = 10b -8 when SZ[1:0] = 11b	—
5:4	SZ[1:0]	DTC Data Transfer Size 0 0: Byte (8-bit) transfer 0 1: Halfword (16-bit) transfer 1 0: Word (32-bit) transfer 1 1: Doubleword (64-bit) transfer	—

Bit	Symbol	Function	R/W
7:6	MD[1:0]	DTC Transfer Mode Select 0 0: Normal transfer mode 0 1: Repeat transfer mode 1 0: Block transfer mode 1 1: Setting prohibited	—

The MRA register cannot be accessed directly from the CPU, however the CPU can access the SRAM area (transfer information (n) start address + 0x03) and DTC transfers it automatically to and from the MRA register. See section 18.3.1. Allocating Transfer Information and DTC Vector Table.

WBDIS bit (Write Back Disable)

This bit selects whether to write back the transfer information at the end of data transfer.

When the WBDIS bit is 0, the updated DTC transfer information is written back.

When the WBDIS bit is 1, the transfer information is not written back even if the address is set to be incremented after transfer, and the same data transfer is performed for each start request. Since the transfer information is not written back, you can put the DTC transfer information in ROM. When the WBDIS bit is 1, the following operations are performed for each transfer mode.

When setting the MRC.DISPE bit to 1, also set the WBDIS bit to 1 (do not write back). If there is at least one transfer information with the WBDIS bit set to 1, set DTCCR.RRS to 0 (do not perform read skip). When used in secure state, access DTCCR_SEC.RRS instead of DTCCR.RRS.

(1) Normal transfer mode and repeat transfer mode

One startup factor transfers 1 byte, 1 halfword, 1 word, and 1 doubleword. The transfer address and the number of transfers are not updated, and the same transfer is repeated for each transfer request. Even if the number of transfers is 1, ICU.DTCERn.DTCE will not be cleared and DTC transfer will continue due to the next activation factor.

(2) Block transfer mode

One block is transferred by one activation factor. The transfer address and block transfer count are not updated, and the same block transfer is repeated for each transfer request. Even if the number of block transfers is 1, ICU.DTCERn.DTCE does not clear to 0 and continues DTC transfer due to the next activation factor.

18.2.3 MRB : DTC Mode Register B

Base address: DTCVBR_SEC
DTCVBR

Offset address: 0x02 + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 18.3.1. Allocating Transfer Information and DTC Vector Table.)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CHNE	CHNS	DISEL	DTS	DM[1:0]		—	—
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
1:0	—	The read values are undefined. The write value should be 0.	—

Bit	Symbol	Function	R/W
3:2	DM[1:0]	Transfer Destination Address Addressing Mode 0 0: Address in the DAR register is fixed (write-back to DAR is skipped) 0 1: Address in the DAR register is fixed (write-back to DAR is skipped) 1 0: DAR value is incremented after data transfer: +1 when MRA.SZ[1:0] = 00b +2 when SZ[1:0] = 01b +4 when SZ[1:0] = 10b +8 when SZ[1:0] = 11b 1 1: DAR value is decremented after data transfer: -1 when MRA.SZ[1:0] = 00b -2 when SZ[1:0] = 01b -4 when SZ[1:0] = 10b -8 when SZ[1:0] = 11b	—
4	DTS	DTC Transfer Mode Select 0: Select transfer destination as repeat or block area. 1: Select transfer source as repeat or block area.	—
5	DISEL	DTC Interrupt Select 0: Generate an interrupt request to the CPU when specified data transfer is complete. 1: Generate an interrupt request to the CPU each time DTC data transfer is performed.	—
6	CHNS	DTC Chain Transfer Select 0: Chain transfer is continuous. 1: Chain transfer occurs only when the transfer counter changes from 1 to 0 or 1 to CRAH.	—
7	CHNE	DTC Chain Transfer Enable 0: Chain transfer is disabled. 1: Chain transfer is enabled.	—

The MRB register cannot be accessed directly from the CPU, however the CPU can access the SRAM area (transfer information (n) start address + 0x02) and DTC transfers it automatically to and from the MRB register. See section 18.3.1. Allocating Transfer Information and DTC Vector Table.

DM[1:0] bits (Transfer Destination Address Addressing Mode)

The DM[1:0] bits are to fix the address of the DAR register or specify increment / decrement of the DAR register after transfer.

DTS bit (DTC Transfer Mode Select)

The DTS bit specifies whether the transfer source or destination is the repeat or block area in repeat or block transfer mode.

DISEL bit (DTC Interrupt Select)

The DISEL bit specifies the condition for generating an interrupt request to the CPU.

CHNS bit (DTC Chain Transfer Select)

The CHNS bit selects the chain transfer condition. When CHNE is 0, the CHNS setting is ignored. For details on the conditions for chain transfer, see Table 18.3.

When the next transfer is chain transfer, completion of the specified number of transfers is not determined, the activation source flag is not cleared, and an interrupt request to the CPU is not generated.

CHNE bit (DTC Chain Transfer Enable)

The CHNE bit enables chain transfer. The chain transfer condition is selected by the CHNS bit. For details on chain transfer, see section 18.4.6. Chain Transfer.

Note: When the chain transfer settings are CHNE = 1 and CHNS = 0, the transfer count according to the transfer settings will be -1 for each transfer request. Therefore, the transfer count for each transfer information must match.

18.2.4 MRC : DTC Mode Register C

Base address: DTCVBR_SEC
DTCVBR

Offset address: 0x01 + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 18.3.1. Allocating Transfer Information and DTC Vector Table.)

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

Bit	Symbol	Function	R/W
0	DISPE	Displacement Added 0: Do not add the displacement value to the forwarding source address. 1: Add the displacement value to the forwarding source address.	—
7:1	—	The read values are undefined. The write value should be 0.	—

The MRC register cannot be accessed directly from the CPU, however the CPU can access the SRAM area (transfer information (n) start address + 0x01) and DTC transfers it automatically to and from the MRC register. See section 18.3.1. Allocating Transfer Information and DTC Vector Table

DISPE bit (Displacement Added)

The DISPE bit specifies whether to use the value of SAR + DTCDISP as the forwarding address. When setting the DISPE bit to 1, be sure to set the MRA.WBDIS bit to 1 (do not write back) and the DTCCR.RRS bit to 0 (do not perform read skip). When used in secure state, access DTCCR_SEC.RRS instead of DTCCR.RRS.

18.2.5 SAR : DTC Transfer Source Register

Base address: DTCVBR_SEC
DTCVBR

Offset address: 0x04 + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 18.3.1. Allocating Transfer Information and DTC Vector Table.)

Bit position: 31 ... 0

Bit field: (empty)

Value after reset: x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x

The SAR sets the transfer source start address and cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 0x04) and DTC transfers it automatically to and from the SAR register. See section 18.3.1. Allocating Transfer Information and DTC Vector Table.

Misalignment is prohibited for DTC transfers. Bit[0] must be 0 when MRA.SZ[1:0] = 01b, and bit[1] and bit[0] must be 0 when MRA.SZ[1:0] = 10b, and bit[2], bit[1], and bit[0] must be 0 when MRA.SZ[1:0] = 11b.

18.2.6 DAR : DTC Transfer Destination Register

Base address: DTCVBR_SEC
DTCVBR

Offset address: 0x08 + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 18.3.1. Allocating Transfer Information and DTC Vector Table.)

Bit position: 31 ... 0

Bit field: (empty)

Value after reset: x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x

The DAR sets the transfer destination start address and cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 0x08) and DTC transfers it automatically to and from the DAR register. See section 18.3.1. Allocating Transfer Information and DTC Vector Table.

Misalignment is prohibited for DTC transfers. Bit[0] must be 0 when MRA.SZ[1:0] = 01b, and bit[1] and bit[0] must be 0 when MRA.SZ[1:0] = 10b, and bit[2], bit[1], and bit[0] must be 0 when MRA.SZ[1:0] = 11b.

18.2.7 CRA : DTC Transfer Count Register A

Base address: DTCVBR_SEC
DTCVBR

Offset address: 0x0E + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 18.3.1. Allocating Transfer Information and DTC Vector Table.)

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

Bit	Symbol	Function	R/W
7:0	CRAL	Transfer Counter A Lower Register Specify the transfer count.	—
15:8	CRAH	Transfer Counter A Upper Register Specify the transfer count.	—

Note: The function depends on the transfer mode.
Note: Set CRAH and CRAL to the same value in repeat transfer mode and block transfer mode.

The CRA register consists of 16 bits. CRAL is the lower 8 bits and CRAH is the upper 8 bits. CRA is used in normal mode.

CRAL and CRAH are used in repeat transfer mode and block transfer mode.

The CRA register cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 0x0E) and DTC transfers it automatically to and from the CRA register. See section 18.3.1. Allocating Transfer Information and DTC Vector Table.

(1) Normal transfer mode (MRA.MD[1:0] = 00b)

In normal transfer mode, the CRA register functions as a 16-bit transfer counter. The transfer count is 1, 65535, and 65536 when the set value is 0x0001, 0xFFFF, and 0x0000, respectively. The CRA value is decremented (-1) on each data transfer.

(2) Repeat transfer mode (MRA.MD[1:0] = 01b)

In repeat transfer mode, the CRAH register holds the transfer count and the CRAL register functions as an 8-bit transfer counter. The transfer count is 1, 255, and 256 when the set value is 0x01, 0xFF, and 0x00, respectively. The CRAL value is decremented (-1) on each data transfer. When it reaches 0x00, the CRAH value is transferred to CRAL.

(3) Block transfer mode (MRA.MD[1:0] = 10b)

In block transfer mode, the CRAH register holds the block size and the CRAL register functions as an 8-bit block size counter. The transfer count is 1, 255, and 256 when the set value is 0x01, 0xFF, and 0x00, respectively. The CRAL value is decremented (-1) on each data transfer. When it reaches 0x00, the CRAH value is transferred to CRAL.

18.2.8 CRB : DTC Transfer Count Register B

Base address: DTCVBR_SEC
DTCVBR

Offset address: 0x0C + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 18.3.1. Allocating Transfer Information and DTC Vector Table.)

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

The CRB sets the block transfer count for block transfer mode. The transfer count is 1, 65535, and 65536 when the set value is 0x0001, 0xFFFF, and 0x0000, respectively. The CRB value is decremented (-1) when the final data of a single block size is transferred. When normal transfer mode or repeat transfer mode is selected, this register is not used, and the set value is ignored.

The CRB cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 0x0C) and DTC transfers it automatically to and from the CRB register. See section 18.3.1. Allocating Transfer Information and DTC Vector Table.

18.2.9 DTCCR : DTC Control Register

Base address: DTC0_NS = 0x5000_AC00

Offset address: 0x00

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

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	—	This bit is read as 1. The write value should be 1.	R/W
4	RRS	DTC Transfer Information Read Skip Enable for Non-secure 0: Transfer information read is not skipped. 1: Transfer information read is skipped when vector numbers match	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE7, P-TYPE2

RRS bit (DTC Transfer Information Read Skip Enable for Non-secure)

The RRS bit enables skipping of transfer information reads when vector numbers match. The DTC vector number is compared with the vector number in the previous activation process. When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the transfer information. However, when the previous transfer is a chain transfer, the transfer information is read regardless of the RRS bit.

When the transfer counter (CRA register) becomes 0 during the previous normal transfer and when the transfer counter (CRB register) becomes 0 during the previous block transfer, the transfer information is read regardless of the RRS bit value.

Set the RRS bit to 0 when there is at least one transfer with the MRA.WBDIS bit set to 1.

If you want to set the MRC.DISPE bit to 1, you need to set the MRA.WBDIS bit to 1.

For information on the function of the RRS bit and how to set it, see section 18.4.1. Transfer Information Read Skip Function.

18.2.10 DTCCR_SEC : DTC Control Register for Secure Region

Base address: DTC0 = 0x4000_AC00

Offset address: 0x10

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

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	—	This bit is read as 1. The write value should be 1.	R/W
4	RRS	DTC Transfer Information Read Skip Enable for Secure 0: Transfer information read is not skipped. 1: Transfer information read is skipped when vector numbers match.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE6, P-TYPE2

RRS bit (DTC Transfer Information Read Skip Enable for Secure)

The RRS bit enables skipping of transfer information reads when vector numbers match. The DTC vector number is compared with the vector number in the previous activation process. When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the transfer information. However, when the previous transfer is a chain transfer, the transfer information is read regardless of the RRS bit.

When the transfer counter (CRA register) becomes 0 during the previous normal transfer and when the transfer counter (CRB register) becomes 0 during the previous block transfer, the transfer information is read regardless of the RRS bit value.

18.2.11 DTCVBR : DTC Vector Base Register

Base address: DTC0_NS = 0x5000_AC00

Offset address: 0x04

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

Bit	Symbol	Function	R/W
31:0	n/a	DTC Vector Base Address for Non-secure region Set the DTC vector base address for non-secure region. The lower 10 bits should be 0.	R/W

Note: S-TYPE7, P-TYPE2

The DTCVBR sets the base address for calculating the DTC vector table address, which can be set in the range of 0x0000_0000 to 0xFFFF_FFFF (4 GB) in 1 KB units.

For information on the function of the RRS bit and how to set it, see section 18.4.1. Transfer Information Read Skip Function.

18.2.12 DTCVBR_SEC : DTC Vector Base Register for Secure Region

Base address: DTC0 = 0x4000_AC00

Offset address: 0x14

Bit position:	31 … 0
Bit field:	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	DTC Vector Base Address for Secure region Set the DTC vector base address for secure region. The lower 10 bits should be 0.	R/W

Note: S-TYPE6, P-TYPE2

The DTCVBR_SEC sets the base address for calculating the DTC vector table address, which can be set in the range of 0x0000_0000 to 0xFFFF_FFFF (4 GB) in 1-KB units.

18.2.13 DTCST : DTC Module Start Register

Base address: DTC0 = 0x4000_AC00
DTC0_NS = 0x5000_AC00

Offset address: 0x0C

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

Bit	Symbol	Function	R/W
0	DTCST	DTC Module Start 0: DTC module stopped 1: DTC module started	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE2

DTCST bit (DTC Module Start)

Set the DTCST bit to 1 to enable the DTC to accept transfer requests. When this bit is set to 0, transfer requests are no longer accepted. If this bit is set to 0 during a data transfer, the accepted transfer request is active until processing completes.

DTCST must be set to 0 before transitioning to one of the following state or mode:

- Module-stop state
- Software Standby mode
- Deep Software Standby mode

For details on these transitions, see section 18.10. Low Power Consumption Function and section 11, Low Power Mode.

18.2.14 DTCSTS : DTC Status Register

Base address: DTC0 = 0x4000_AC00
DTC0_NS = 0x5000_AC00

Offset address: 0x0E

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

Bit	Symbol	Function	R/W
7:0	VECN[7:0]	DTC-Activating Vector Number Monitoring These bits indicate the vector number for the activation source when a DTC transfer is in progress. The value is only valid if a DTC transfer is in progress (ACT flag is 1).	R
14:8	—	These bits are read as 0.	R
15	ACT	DTC Active Flag 0: DTC transfer operation is not in progress. 1: DTC transfer operation is in progress.	R

Note: S-TYPE3, P-TYPE2

VECN[7:0] bits (DTC-Activating Vector Number Monitoring)

While transfer by the DTC is in progress, the VECN[7:0] bits indicate the vector number associated with the activation source for the transfer. The value read from the VECN[7:0] bits is valid if the ACT flag is 1, indicating a DTC transfer in progress, and invalid if the ACT flag is 0, indicating no DTC transfer is in progress.

ACT flag (DTC Active Flag)

The ACT flag indicates the state of the DTC transfer operation.

[Setting condition]

- When the DTC is activated by a transfer request.

[Clearing condition]

- When transfer by the DTC, in response to a transfer request, is complete.

18.2.15 DTCDISP : DTC Address Displacement Register

Base address: DTC0 = 0x4000_AC00
DTC0_NS = 0x5000_AC00

Offset address: 0x18

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

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

Bit	Symbol	Function	R/W
31:0	DTCDISP[31:0]	DTC Address Displacement Displacement value to be added to the DTC forwarding source address.	R/W

Note: S-TYPE-7, P-TYPE-2

The DTCDISP register specifies the displacement value to be added to the DTC forwarding source address. If the MRC.DISPE bit is 1, the value of SAR + DTCDISP is used as the transfer source address.

18.2.16 DTEVR : DTC Error Vector Register

Base address: DTC0 = 0x4000_AC00
DTC0_NS = 0x5000_AC00

Offset address: 0x20

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

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

Bit	Symbol	Function	R/W
7:0	DTEV[7:0]	DTC Error Vector Number These bits represent error vector of the DTC.	R
8	DTEVSAM	DTC Error Vector Number SA Monitor Indicates the SA of vector number causing the error. 0: Secure vector number 1: Non-secure vector number	R
15:9	—	These bits are read as 0. The write value should be 0.	R
16	DTESTA	DTC Error Status Flag 0: No DTC transfer error occurred 1: DTC transfer error occurred	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R

Note: P-TYPE2 See below for access by security attribution.
Note: Writing to DTESTA depends on the value of DTEVSAM

When reading this register, it can be accessed from both secure and non-secure access.

When writing this register, it depends on DTEVR.DTEVSAM.

- When DTEVSAM = 1, it can be accessed from Secure and Non-secure access.
- When DTEVSAM = 0, it can be accessed from Secure. An error is returned when write access is performed in a Non-secure access.

This register is cleared by a reset caused by a transfer error. Please select interrupt DMAm_TRANSERR (m = 0, 1) in BUS.BUSOAD and TZF.MSAOAD when you want to debug the program.

DTEV[7:0] bit (DTC Error Vector Number)

When a transfer error due to DTC transfer occurs, it stores the channel of DTC that was violated.

[Setting condition]

- When the DTC transfer error occurs and DTESTA = 0.

[Clearing condition]

- When 1 is written to DTEVR.DTESTA.

DTEVSAM bit (DTC Error Vector Number SA Monitor)

When a transfer error due to DTC transfer occurs, it indicates the SA of the violating DTC vector number.

[Setting condition]

- When the DTC transfer error occurs and DTESTA = 0.

[Clearing condition].

- When 1 is written to DTEVR.DTESTA.

DTESTA bit (DTC Error Status Flag)

Indicates whether or not a DTC transfer error occurred.

DTEV, DTEVSAM, DTESTA are cleared by writing 1 to DTESTA.

Writing 0 to DTESTA is ignored.

[Setting condition]

- When the DTC transfer error occurs.

[Clearing condition]

- When 1 is written to DTEVR.DTESTA.

18.3 Activation Sources

The DTC is activated by an interrupt request. Setting the ICU.IELSRn.DTCE bit to 1 enables activation of the DTC by the associated interrupt. The selector output *n* number set in ICU.IELSRn is defined as the interrupt vector number. For an enabled interrupt, the specific DTC interrupt source associated with each interrupt vector number n is selected in ICU.IELSRn.IELS[9:0] (n = 0 to 95), as listed in section 14.3.2. Event Number. For activation by software, see section 19.2.2. ELSEGRn : Event Link Software Event Generation Register n (n = 0 to 3).

The interrupt vector number is equivalent to the DTC vector table number. After the DTC accepted an activation request, it does not accept another activation request until the transfer for that single request is complete, regardless of the priority of the requests. When multiple activation requests are generated during a DTC transfer, the highest priority request is accepted on completion of the transfer. When multiple activation requests are generated while the DTC Module Start bit (DTCST.DTCST) is 0, the DTC accepts the highest priority request when DTCST.DTCST is subsequently set to 1. The smaller interrupt vector number has higher priority.

The DTC performs the following operations at the start of a single data transfer or for a chain transfer, after the last of the consecutive transfers:

- On completion of a specified round of data transfer, the ICU.IELSRn.DTCE bit is set to 0, and an interrupt request is sent to the CPU.
- If the MRB.DISEL bit is 1, an interrupt request is sent to the CPU on completion of a data transfer.
- For other transfers, the ICU.IELSRn.IR flag of the activation source is set to 0 at the start of the data transfer.

18.3.1 Allocating Transfer Information and DTC Vector Table

The DTC reads the start address of the transfer information associated with each activation source from the vector table and reads the transfer information starting at that address.

The DTC has a vector table for secure DTC activation requests and a vector table for non-secure DTC activation requests. Place the vector table of the interrupt vector number of SA = 1 in DTCVBR which is the Non-secure side. Place the vector table of interrupt number SA = 0 in DTCVBR_SEC which is the Secure side.

The vector table must be located so that the lower 10 bits of the base address (start address) are 0. Use the DTC Vector Base Register (DTCVBR) to set the base address of the DTC vector table. Transfer information is allocated in the SRAM area. However, if the MRA.WBDIS bit is set to 1 (no write back), it can also be placed in the ROM area. The start address of the transfer information n with vector number n must be 4n added to the base address in the vector table. Each transfer information start address must be a multiple of 16.

When setting the DTC transfer attribution for the privilege access or unprivileged access, allocating the vector table and transfer information as follows. The vector table should be located in the privileged area. Transfer information with the privileged attribution is located in the privileged area. Transfer information with the unprivileged attribution is located in any privileged area. Privileged area is set by MMPU (see section 16, Memory Protection Unit (MPU)). Privileged and non-privileged access settings are located at bit 0 of the address corresponding to each vector number in the vector table.

Figure 18.2 shows the relationship between the DTC vector table and transfer information.

Figure 18.3 shows the bit arrangement in the vector table.

Figure 18.4 shows the allocation of transfer information in the SRAM area.

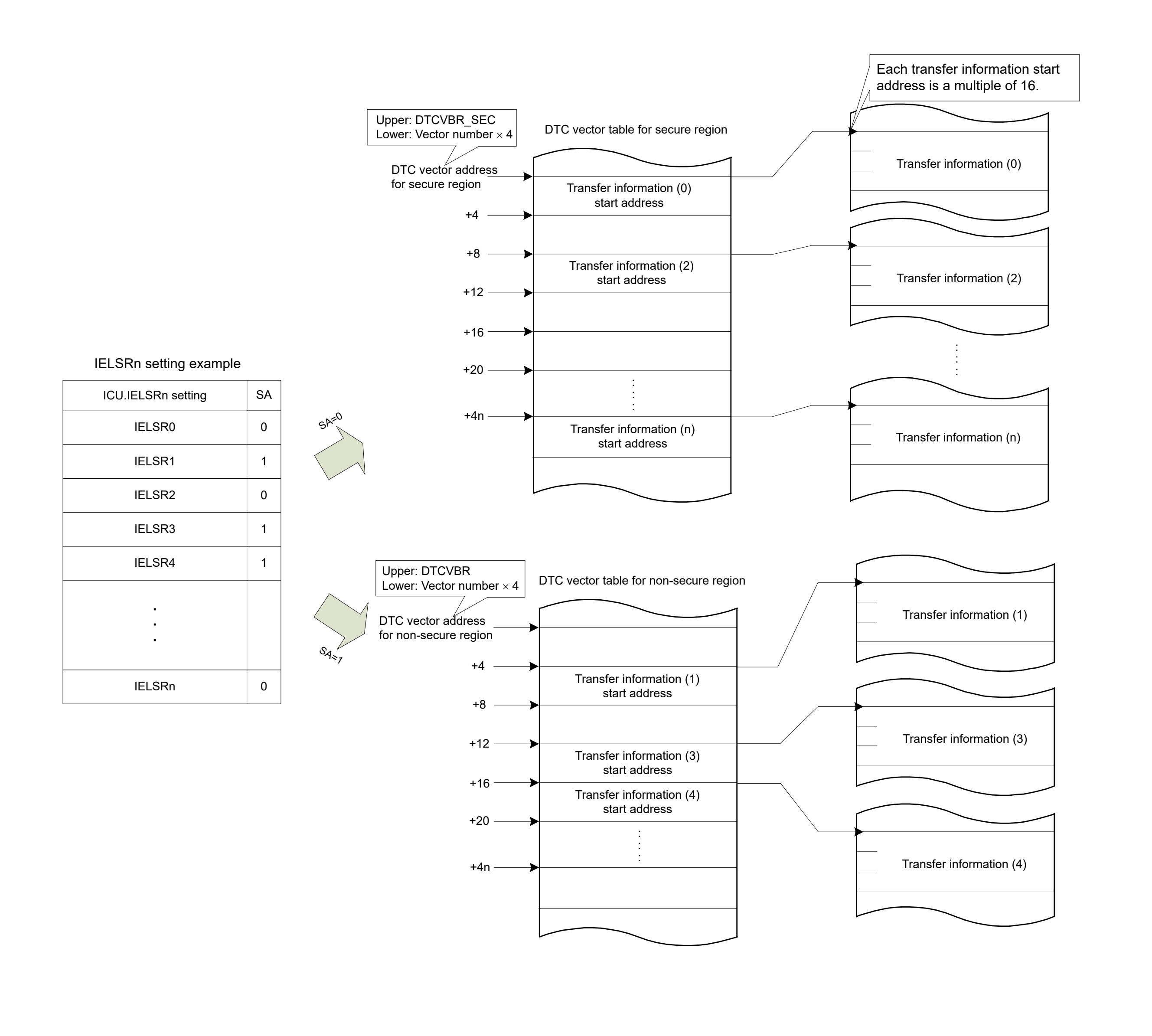


Figure 18.2 DTC vector table and transfer information

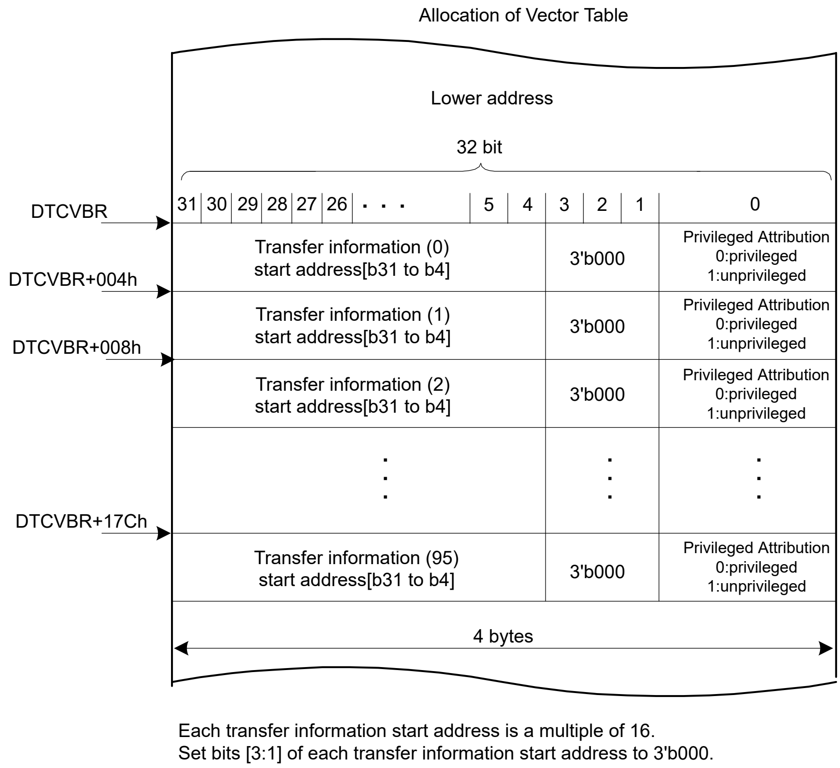


Figure 18.3 Bit arrangement in the vector table

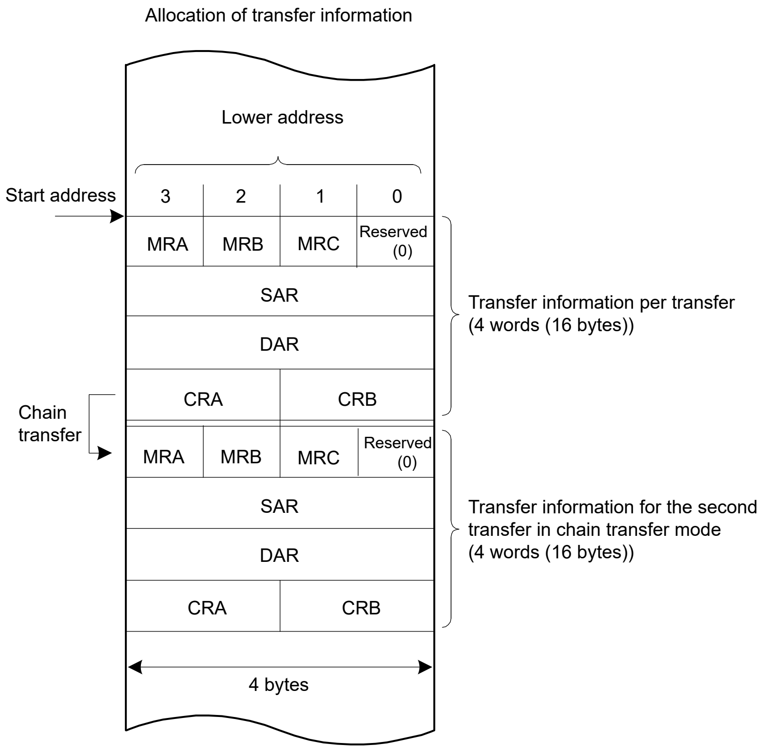


Figure 18.4 Allocation of transfer information in the SRAM area

18.4 Operation

The DTC transfers data according to the transfer information. Storage of the transfer information in the SRAM area is required before a DTC operation. When the DTC is activated, it reads the DTC vector associated with the vector number. The DTC reads the transfer information from the transfer information store address referenced by the DTC vector and transfers the data. After the data transfer, the DTC writes back the transfer information. Storing the transfer information in the SRAM area allows data transfer of any number of channels.

The transfer modes include:

- Normal transfer mode
- Repeat transfer mode
- Block transfer mode.

The DTC specifies a transfer source address in the SAR register and a transfer destination address in the DAR register. The values of these registers are incremented, decremented, or address-fixed independently after the data transfer.

Table 18.2 describes the DTC transfer modes.

Table 18.2 DTC transfer modes (1 of 2)

Transfer mode	Data size transferred on single transfer request	Increment or decrement of memory address	Settable transfer count
Normal transfer mode	1 byte (8 bits), 1 halfword (16 bits), 1 word (32 bits), or 1 doubleword (64 bits).	Incremented or decremented by 1, 2, 4, or 8 or address-fixed	1 to 65536
Repeat transfer mode*1	1 byte (8 bits), 1 halfword (16 bits), 1 word (32 bits), or 1 doubleword (64 bits).	Incremented or decremented by 1, 2, 4, or 8 or address-fixed	1 to 256*3

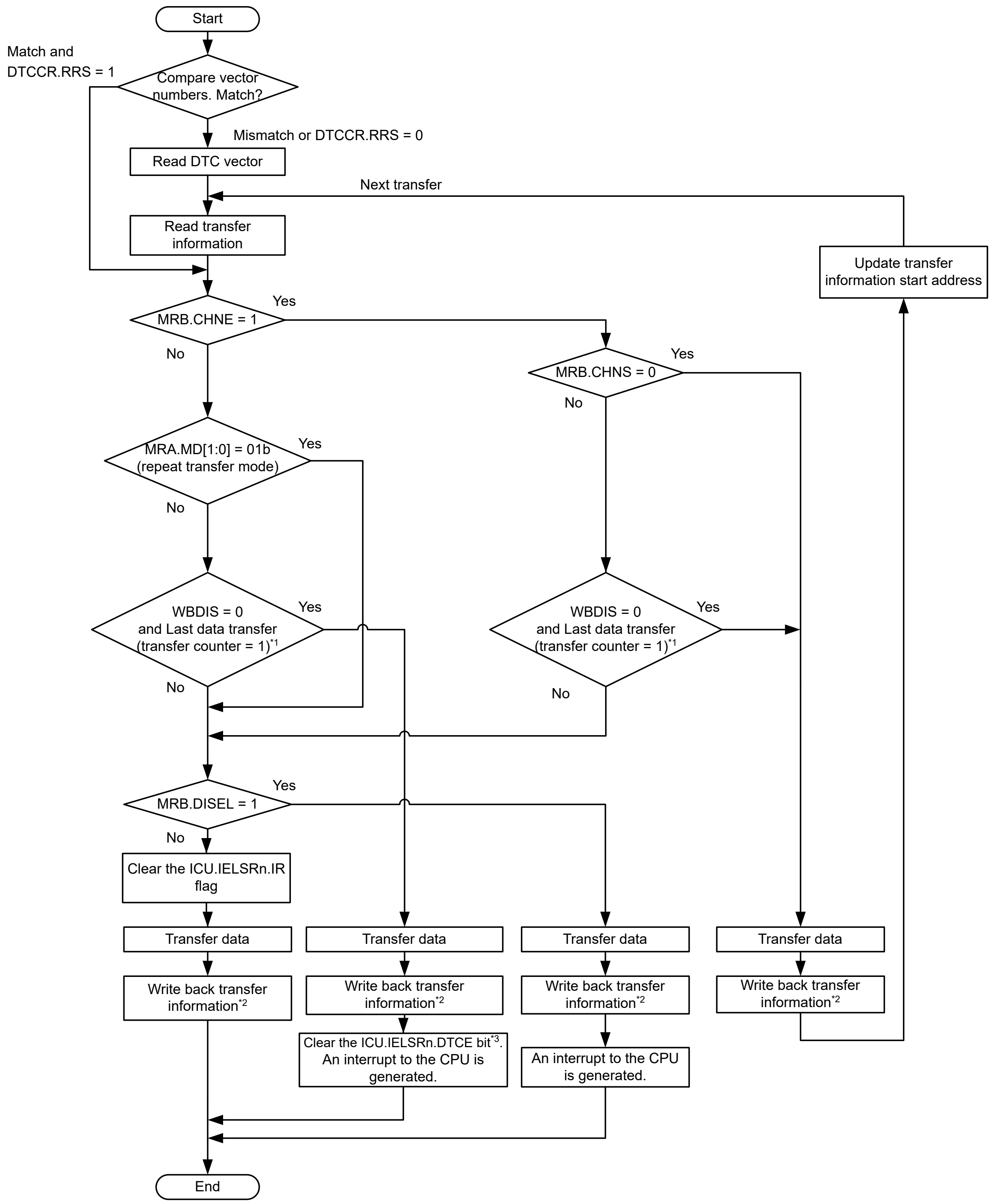
Table 18.2 DTC transfer modes (2 of 2)

Transfer mode	Data size transferred on single transfer request	Increment or decrement of memory address	Settable transfer count
Block transfer mode*2	Block size specified in CRAH (1 to 256 bytes, 1 to 256 halfwords (2 to 512 bytes), 1 to 256 words (4 to 1024 bytes), or 1 to 256 doublewords (8 to 2048 bytes)).	Incremented or decremented by 1, 2, 4, or 8 or address-fixed	1 to 65536

Note 1. Set the transfer source or transfer destination as the repeat area.
Note 2. Set the transfer source or transfer destination as the block area.
Note 3. After a data transfer of the specified count, the initial state is restored and operation restarts.

Setting the MRB.CHNE bit to 1 allows multiple transfers or chain transfer on a single activation source. It also enables a chain transfer when the specified data transfer is complete.

Figure 18.5 shows the operation flow of the DTC. Table 18.3 lists the chain transfer conditions. The combination of control information for the second and subsequent transfers are omitted in this table.



Note 1. Counter value before starting data transfer.
Note 2. Write back is skipped when the WBDIS bit is 1.
Note 3. The DTCE bit is not cleared when the WBDIS bit is 1.

Figure 18.5 DTC operation flow

Table 18.3 Chain transfer conditions

First transfer				Second transfer[*3]				DTC transfer
CHNE bit	CHNS bit	DISEL bit	Transfer counter[*1 *2]	CHNE bit	CHNS bit	DISEL bit	Transfer counter[*1 *2]	
0	—	0	Other than (1 → 0)	—	—	—	—	Ends after the first transfer
0	—	0	(1 → 0)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU
0	—	1	—	—	—	—	—	
1	0	—	—	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	0	Other than (1 → *)	—	—	—	—	Ends after the first transfer
1	1	—	(1 → *)	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	1	Other than (1 → *)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU

Note 1. The transfer counter used depends on the transfer modes as follows:
Normal transfer mode — CRA register
Repeat transfer mode — CRAL register
Block transfer mode — CRB register

Note 2. On completion of a data transfer, the counters operate as follows:
1 → 0 in normal and block transfer modes
1 → CRAH in repeat transfer mode
(1 → *) in the table indicates both of these two operations, depending on the mode.

Note 3. Chain transfer can be selected for the second or subsequent transfers. The conditions for the combination of the second transfer and CHNE = 1 is omitted.

18.4.1 Transfer Information Read Skip Function

Reading of vector addresses and transfer information can be skipped by setting the DTCCR.RRS bit. When a DTC activation request is generated, the current DTC vector number is compared with the DTC vector number in the previous activation process. When these vector numbers match and the RRS bit is set to 1, the DTC data transfer is performed without reading the vector address and transfer information. However, when the previous transfer is a chain transfer, the vector address and transfer information are read. Additionally, when the transfer counter (CRA register) becomes 0 during the previous normal transfer, and when the transfer counter (CRB register) becomes 0 during the previous block transfer, transfer information is read regardless of the RRS bit. Figure 18.13 shows an example when reading the transfer information is skipped.

To update the vector table and transfer information, set the RRS bit to 0, update the vector table and transfer information, then set the RRS bit to 1. The stored vector number is discarded by setting the RRS bit to 0. The updated DTC vector table and transfer information are read in the next activation process.

If there is at least one transfer information with the MRA.WBDIS bit set to 1, the transfer information read skip function cannot be used. Set the DTCCR.RRS bit to 0 (do not perform read skip). When used in secure state, access DTCCR_SEC.RRS instead of DTCCR.RRS.

18.4.2 Transfer Information Write-Back Skip Function

When the MRA.SM[1:0] bits or the MRB.DM[1:0] bits are set to address fixed, a part of the transfer information is not written back. Table 18.4 lists the transfer information write-back skip conditions and the associated registers. The CRA and CRB registers are written back, and the write-back of the MRA and MRB registers is skipped.

When the MRA.WBDIS bit is 1, all transfer information (SAR, DAR, CRA, CRB) is not written back. Since the transfer information in memory is not updated, DTC data transfer can be performed without copying the transfer information from ROM to RAM. Also, by omitting writeback, the time required for post-processing of data transfer can be shortened.

Table 18.4 Transfer information write-back skip conditions and applicable registers

MRA.SM[1:0] bits		MRB.DM[1:0] bits		SAR register	DAR register
b3	b2	b3	b2		
0	0	0	0	Skip	Skip
0	0	0	1		
0	1	0	0		
0	1	0	1		
0	0	1	0	Skip	Write-back
0	0	1	1		
0	1	1	0		
0	1	1	1		
1	0	0	0	Write-back	Skip
1	0	0	1		
1	1	0	0		
1	1	0	1		
1	0	1	0	Write-back	Write-back
1	0	1	1		
1	1	1	0		
1	1	1	1		

18.4.3 Normal Transfer Mode

The normal transfer mode allows a 1-byte (8 bits), 1-halfword (16 bits), 1-word (32 bits), or 1-doubleword (64 bits) data transfer on a single activation source. The transfer count can be set from 1 to 65536. Transfer source and destination addresses can be independently set to increment, decrement, or fixed. This mode enables an interrupt request to the CPU to be generated at the end of a specified-count transfer.

Table 18.5 lists register functions in normal transfer mode, and Figure 18.6 shows the memory map of normal transfer mode.

Table 18.5 Register functions in normal transfer mode

Register	Description	Value written back by writing transfer information*2
SAR	Transfer source address	Increment, decrement, or fixed*1
DAR	Transfer destination address	Increment, decrement, fixed*1
CRA	Transfer counter A	CRA - 1
CRB	Transfer counter B	Not updated

Note 1. Write-back operation is skipped in address-fixed mode.
Note 2. Write-back is skipped when MRA.WBDIS = 1.

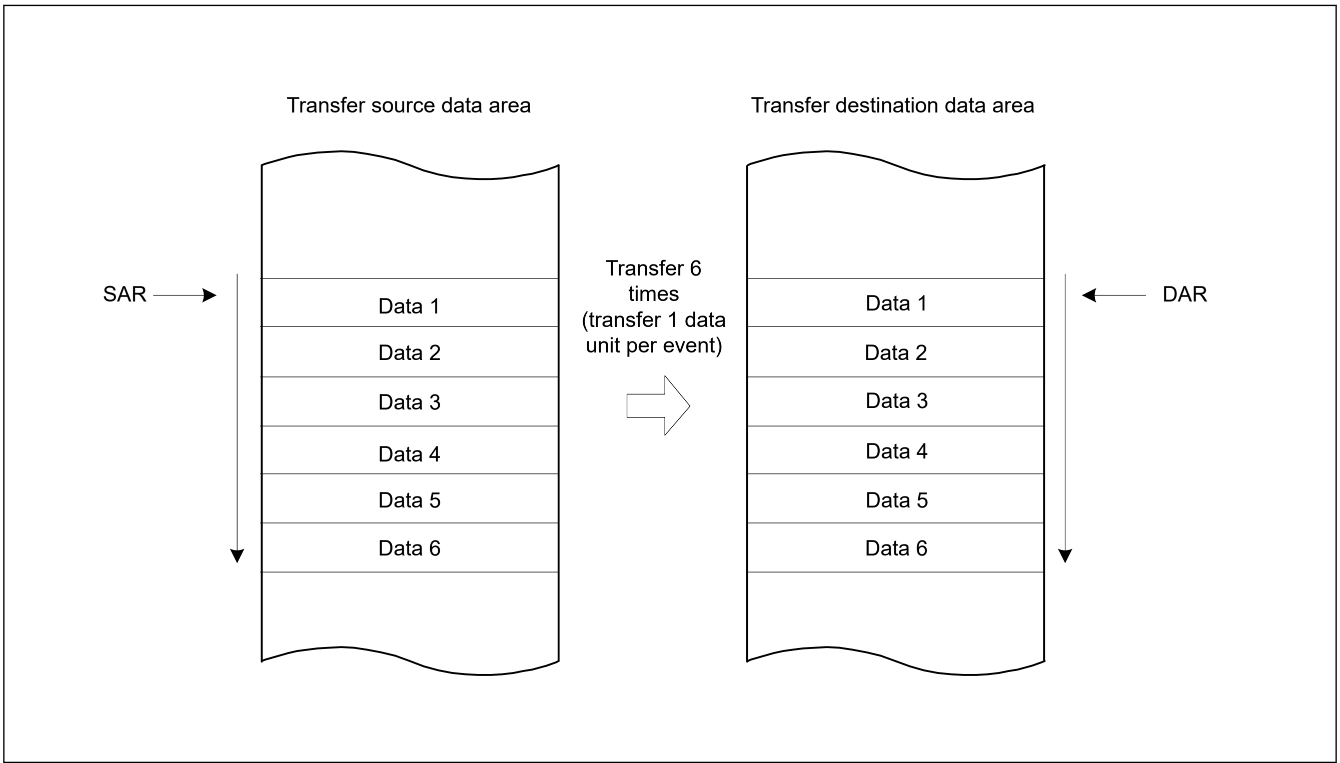


Figure 18.6 Memory map of normal transfer mode (MRA.SM[1:0] = 10b, MRB.DM[1:0] = 10b, CRA = 0x0006)

18.4.4 Repeat Transfer Mode

The repeat transfer mode allows a 1-byte (8 bits), 1-halfword (16 bits), 1-word (32 bits), or 1-doubleword (64 bits) data transfer on a single activation source. Transfer source or transfer destination for the repeat area must be specified in the MRB.DTS bit. The transfer count can be set from 1 to 256. When the specified transfer count is complete, the initial value of the address register specified in the repeat area is restored, the initial value of the transfer counter is restored, and transfer is repeated. The other address register is incremented or decremented continuously or remains unchanged.

When the transfer counter CRAL decrements to 0x00 in repeat transfer mode, the CRAL value is updated to the value set in the CRAH register. As a result, the transfer counter does not clear to 0x00, which disables interrupt requests to the CPU when the MRB.DISEL bit is set to 0. An interrupt request to the CPU is generated when the specified data transfer completes.

Table 18.6 lists the register functions in repeat transfer mode, and Figure 18.7 shows the memory map of repeat transfer mode.

Table 18.6 Register functions in repeat transfer mode

Register	Description	Value written back by writing transfer information[*2]	
		When CRAL is not 1	When CRAL is 1
SAR	Transfer source address	Increment, decrement, fixed[*1]	• When the MRB.DTS bit is 0 Increment, decrement, or fixed[*1] • When the MRB.DTS bit is 1 SAR register initial value
DAR	Transfer destination address	Increment, decrement, or fixed[*1]	• When the MRB.DTS bit is 0 DAR register initial value • When the MRB.DTS bit is 1 Increment, decrement, or fixed[*1]
CRAH	Retains transfer counter	CRAH	CRAH
CRAL	Transfer counter A	CRAL - 1	CRAH
CRB	Transfer counter B	Not updated	Not updated

Note 1. Write-back is skipped in address-fixed mode.
Note 2. Write-back is skipped when MRA.WBDIS = 1.

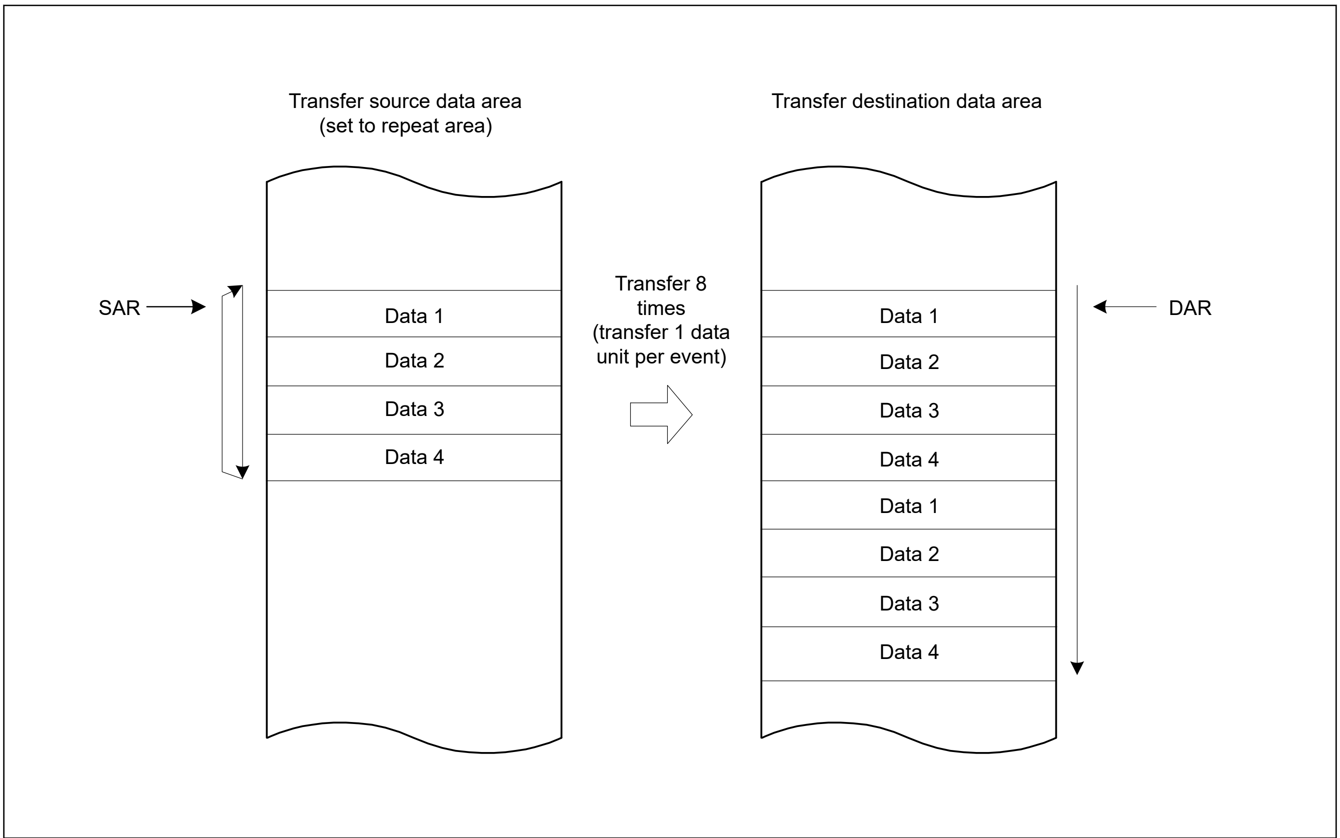


Figure 18.7 Memory map of repeat transfer mode when transfer source is a repeat area (MRA.SM[1:0] = 10b, MRB.DM[1:0] = 10b, CRAH = 0x04)

18.4.5 Block Transfer Mode

The block transfer mode allows single-block data transfer on a single activation source. Transfer source or transfer destination for the block area must be specified in the MRB.DTS bit. The block size can be set from 1 to 256 bytes, 1 to 256 halfwords (2 to 512 bytes), 1 to 256 words (4 to 1024 bytes), or 1 to 256 doublewords (8 to 2048 bytes). When transfer of the specified block completes, the initial values of the block size counter CRAL and the address register (the SAR register when the MRB.DTS = 1 or the DAR register when the DTS = 0) specified in the block area are restored. The other address register is incremented or decremented continuously or remains unchanged.

The transfer count (block count) can be set from 1 to 65536. This mode enables an interrupt request to the CPU to be generated at the end of the specified-count block transfer.

Table 18.7 lists the register functions in block transfer mode, and Figure 18.8 shows the memory map for block transfer mode.

Table 18.7 Register functions in block transfer mode

Register	Description	Value written back by writing transfer information*2
SAR	Transfer source address	• When MRB.DTS bit is 0 Increment, decrement, or fixed*1 • When MRB.DTS bit is 1 SAR register initial value.
DAR	Transfer destination address	• When MRB.DTS bit is 0 DAR register initial value • When MRB.DTS bit is 1 Increment, decrement, or fixed*1.
CRAH	Holds block size	CRAH
CRAL	Block size counter	CRAH
CRB	Block transfer counter	CRB - 1

Note 1. Write-back is skipped in address-fixed mode.

Note 2. Write-back is skipped when MRA.WBDIS = 1.

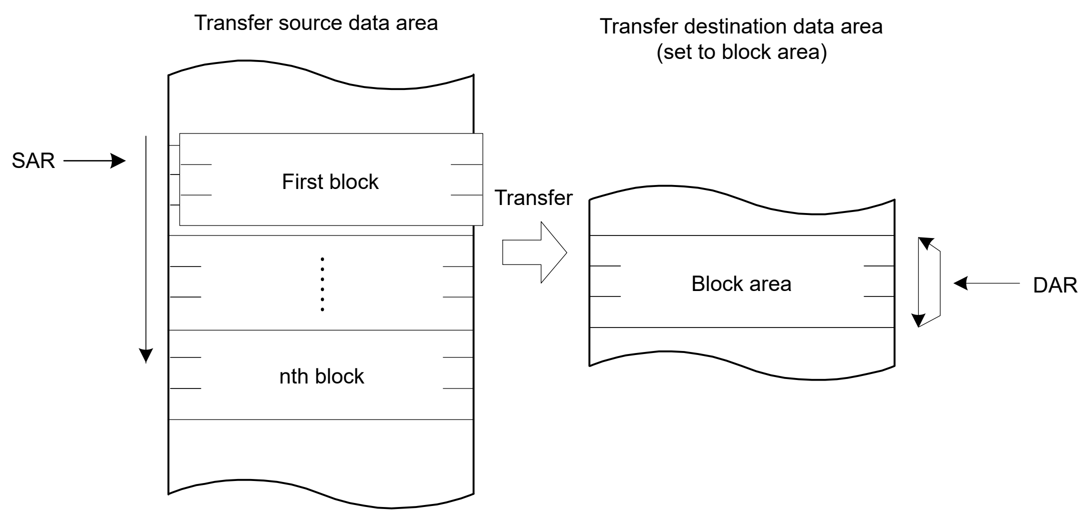


Figure 18.8 Memory map of block transfer mode

18.4.6 Chain Transfer

Setting the MRB.CHNE bit to 1 allows chain transfer to be performed continuously on a single activation source. If the MRB.CHNE is set to 1 and CHNS to 0, an interrupt request to the CPU is not generated on completion of the specified number of rounds of transfer or by setting the MRB.DISEL bit to 1. An interrupt request is sent to the CPU each time DTC data transfer is performed. Data transfer has no effect on the ICU.IELSRn.IR flag of the activation source.

The SAR, DAR, CRA, CRB, MRA, MRB, and MRC registers can be set independently of each other to define the data transfer. Figure 18.9 shows a chain transfer operation.

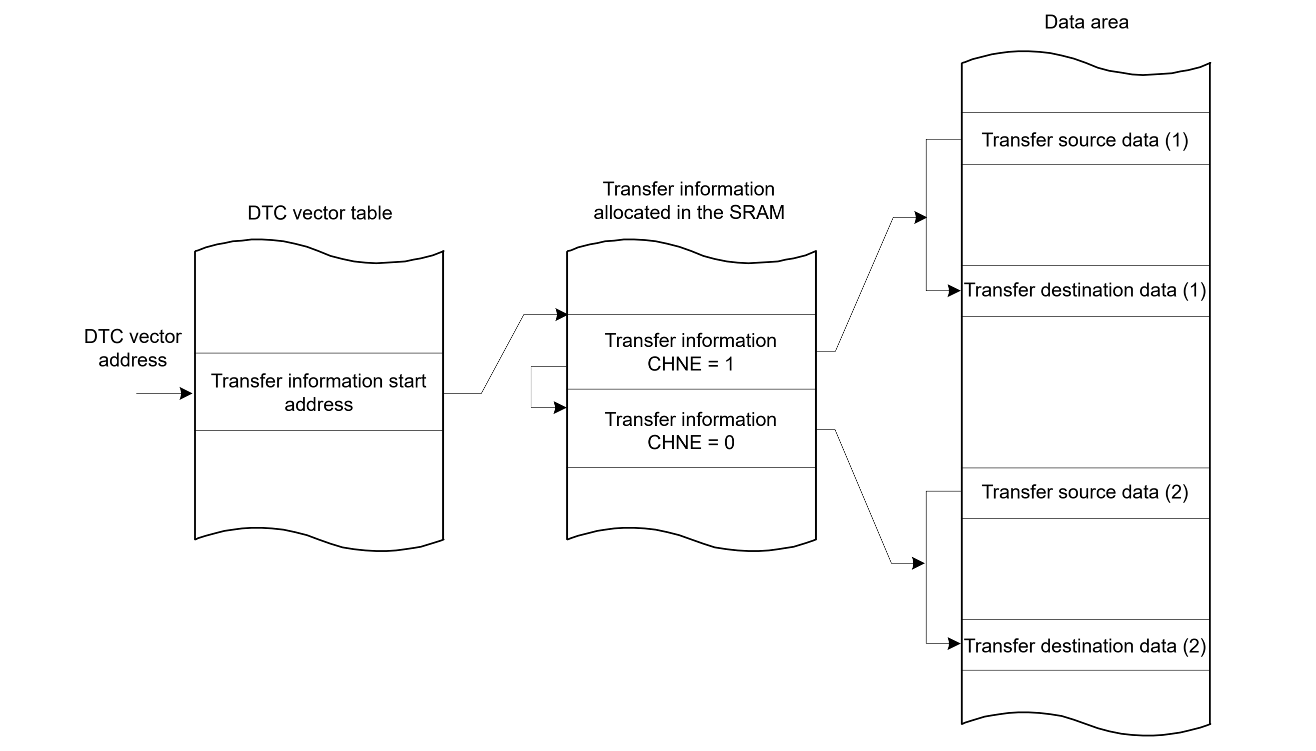


Figure 18.9 Chain transfer operation

Writing 1 to the MRB.CHNE and CHNS bits enables chain transfer to be performed only after completion of the specified data transfer. In repeat transfer mode, chain transfer is performed after completion of the specified data transfer. For details on chain transfer conditions, see Table 18.3.

18.4.7 Operation Timing

Figure 18.10 to Figure 18.13 are timing diagrams that show the minimum number of execution cycles.

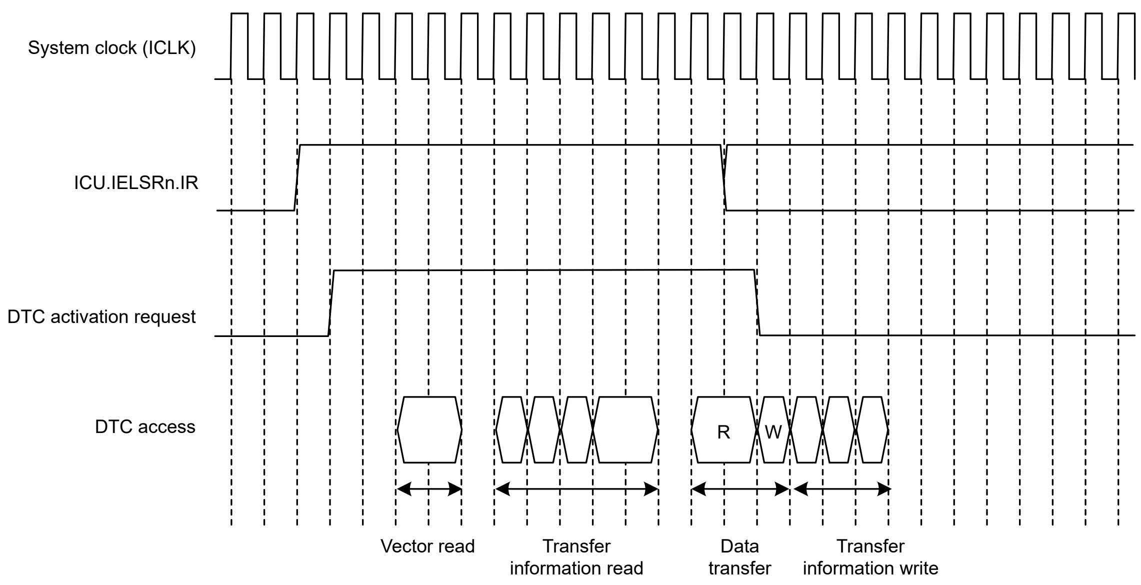


Figure 18.10 Example 1 of DTC operation timing in normal transfer and repeat transfer modes

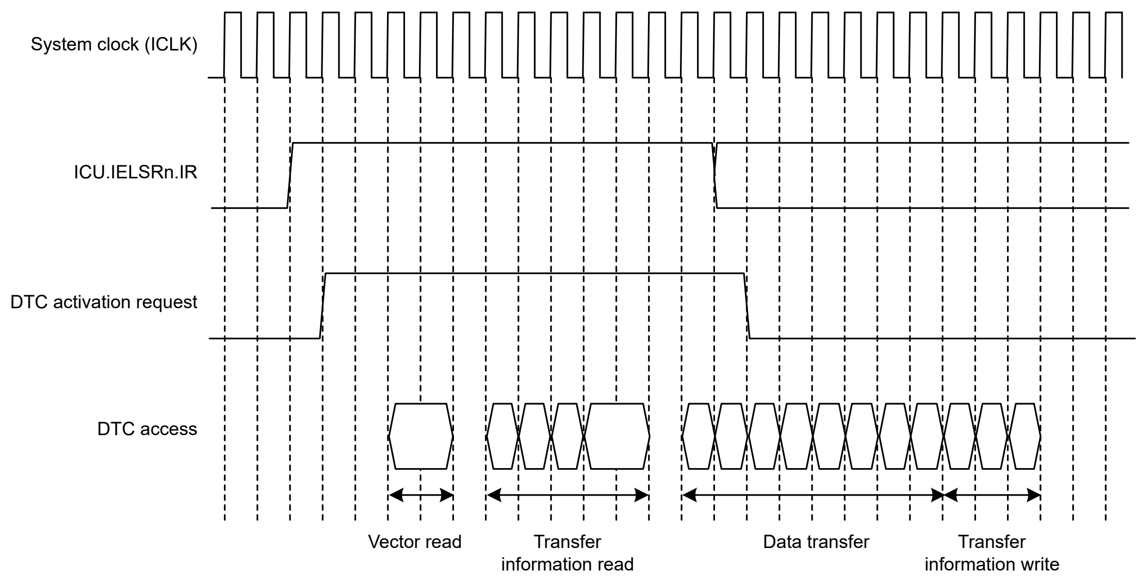
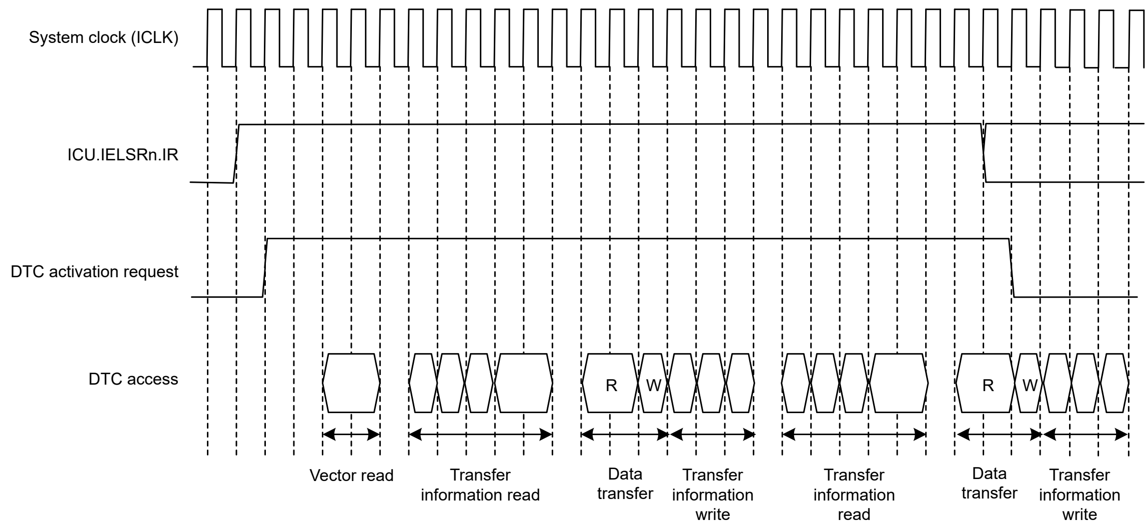
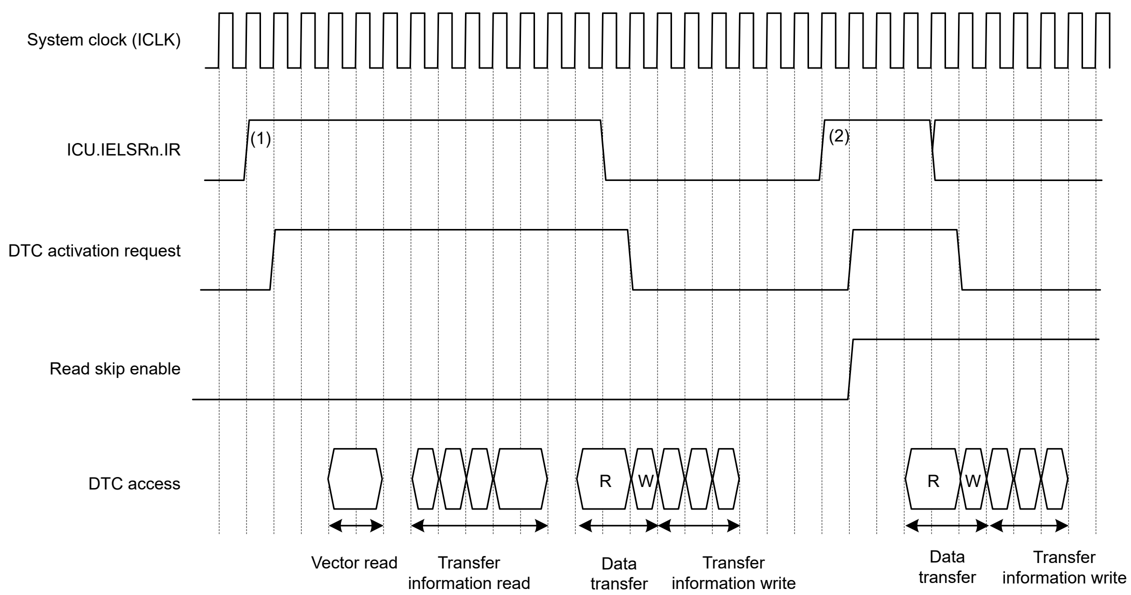


Figure 18.11 Example 2 of DTC operation timing in block transfer mode when the block size = 4



Note: n = Vector number

Figure 18.12 Example 3 of DTC operation timing for chain transfer



Note: n = Vector number
Note: When activation sources (vector numbers) of (1) and (2) are the same and the RRS = 1, the transfer information read for request (2) is skipped.

Figure 18.13 Example of operation when a transfer information read is skipped with the vector, transfer information, and transfer destination data on the SRAM, and the transfer source data on the peripheral module

18.4.8 Vector Security

The security attribute of transfer access of DTC vector n and security attribute of access to the IELSRn (n = 0 to 95) register of ICU are controlled by SAIELSRn bit of ICUSARx (x = G, H, I, J, K or L) registers in CPSCU. For details on the CPSCU.ICUSARx registers, see section 14, Interrupt Controller Unit (ICU).

When the CPSCU.ICUSARx.SAIELSRn bit is 0

- IELSRn register have security attribute
- The vector number of DTC activated by the interrupt request set in IELS behaves as a secure master.

When the CPSCU.ICUSARx.SAIELSRn bit is 1

- IELSRn register have non-security attribute
- The vector number of DTC activated by the interrupt request set in IELS behaves as a non-secure master.

For areas accessible to secure or non-secure masters, See section 15, Buses, and section 51, Security Features.

18.4.9 Vector Privilege

The privileged attribute of transfer access of DTC vector n is controlled bit0 in vector table.

If bit0 corresponding to each vector number in the vector table is set to 0, DTC transfer is privileged access.

If bit0 is set to 1, DTC transfers are unprivileged access.

Refer to for vector table and transfer information.

18.5 DTC Setting Procedure

Before using the DTC, set the DTC Vector Base Register (DTCVBR). When used in secure state, access DTCVBR_SEC instead of DTCVBR. Set the ICU.IELSRn.IELS[8:0] bits to 0 to disable the interrupt in the NVIC and follow the procedure in Table 18.8 to set the DTC.

Table 18.8 DTC setting procedure

No.	Step Name	Description
1	Set the DTCCR.RRS bit to 0	Set the DTCCR.RRS bit to 0 to reset the transfer information read skip flag. After that, the transfer information read is not skipped while the DTC is activated. Be sure to specify this setting when the transfer information is updated.
2	Set transfer information (MRA, MRB, MRC, SAR, DAR, CRA, and CRB)	Allocate transfer information (MRA, MRB, MRC, SAR, DAR, CRA, and CRB) in the data area. To set transfer information, see section 18.2. Register Descriptions. To allocate transfer information, see section 18.3.1. Allocating Transfer Information and DTC Vector Table.
3	Set transfer information start addresses in the DTC vector table	Set the transfer information start addresses in the DTC vector table. To set the DTC vector table, see section 18.3.1. Allocating Transfer Information and DTC Vector Table.
4	Set the DTCCR.RRS bit to 1	Set the DTCCR.RRS bit to 1 to enable skipping of the second and subsequent transfer information read cycles for continuous DTC activation from the same interrupt source. The RRS bit can be set to 1, but if this is set during DTC transfer, it becomes valid from the next transfer. Leave the value of the RRS bit as 0 if the value of the MRA.WBDIS bit or the MRC.DISPE bit in any transfer information is 1.
5	Set the ICU.IELSRn.DTCE bit to 1. Set the ICU.IELSRn.IELS[8:0] as interrupt source. The interrupt should be enabled in the NVIC.	Set the ICU.IELSRn.DTCE bit to 1. Set ICU.IELSRn.IELS[8:0] as interrupt sources that trigger DTC. The interrupt must be enabled in the NVIC. See section 14.3.2. Event Number in section 14, Interrupt Controller Unit (ICU).
6	Set the enable bit for an activation source interrupt	Set the enable bit for the activation source interrupts to 1. When a source interrupt is generated, the DTC is activated. To set the interrupt source enable bit, see the settings for the modules that are to be the activation sources.
7	Set the DTCST.DTCST bit to 1	Set the DTC Module Start bit (DTCST.DTCST) to 1.

Note: The DTCST.DTCST bit can be set even if the setting for each activation source is not completed.
Note: When DTCSAR.DTCSTSA = 0 is set and DTC is used in both the Secure and the Non-secure state, DTCST.DTCST bit must be written in the Secure state.
Note: When used in secure state for step 1 and step 4, access DTCCR_SEC.RRS instead of DTCCR.RRS.

18.6 Examples of DTC Usage

18.6.1 Normal Transfer

This section provides an example of DTC usage and its application when receiving 128 bytes of data from an SCI.

(1) Transfer information settings

In the MRA register, select a fixed source address (MRA.SM[1:0] = 00b), normal transfer mode (MRA.MD[1:0] = 00b), and byte-sized transfer (MRA.SZ[1:0] = 00b). In the MRB register, specify incrementation of the destination address (MRB.DM[1:0] = 10b) and single data transfer by a single interrupt (MRB.CHNE = 0 and MRB.DISEL = 0). The MRB.DTS bit can be set to any value. Set the RDR register address of the SCI in the SAR register, the start address of the SRAM area for data storage in the DAR register, and 128 (0x0080) in the CRA register. The CRB register can be set to any value.

(2) DTC vector table settings

The start address of the transfer information for the RXI interrupt is set in the vector table for the DTC.

(3) ICU settings and DTC module activation

Set the ICU.IELSRn.DTCE bit to 1 and set ICU.IELSRn.IELS[9:0] as the SCI interrupt. The interrupt must be enabled in the NVIC. Set the DTCST.DTCST bit to 1.

(4) SCI settings

Enable the SCIn_RXI (n = 0 to 9) interrupt by setting the SCR.RIE bit in the SCI to 1. If a reception error occurs during the SCI receive operation, reception stops. To manage this, use settings that allow the CPU to accept receive error interrupts.

(5) DTC transfer

Each time a reception of 1 byte by the SCI is complete, an SCIn_RXI interrupt is generated to activate the DTC. The DTC transfers the received byte from the RDR of the SCI to the SRAM, after which the DAR register is incremented and the CRA register is decremented.

(6) Interrupt handling

After 128 rounds of data transfer are complete and the value in the CRA register becomes 0, an SCIn_RXI interrupt request is generated for the CPU. Complete the process in the handling routine for this interrupt.

18.6.2 Chain Transfer

This section provides an example of chain transfer by the DTC and describes its use in the output of pulses by the General PWM Timer (GPT). You can use chain transfer to transfer PWM timer compare data and change the period of the PWM timer for the GPT.

For the first of the chain transfers, normal transfer mode is specified for transfer to the GPT32n.GTCCRC register (n = 0 to 13). For the second of the chain transfers, normal transfer mode is specified for transfer to the GPT32n.GTCCRE register (n = 0 to 13). For the third of the chain transfers, normal transfer mode is specified for transfer to the GPT32n.GTPBR register (n = 0 to 13). This is because clearing of the activation source and generation of an interrupt on completion of the specified number of transfers are restricted to the third of the chain transfers, that is, transfer while MRB.CHNE = 0.

The following example shows how to use the counter overflow interrupt with the GPT320.GTPR register as an activating source for the DTC.

(1) First transfer information set

Set up transfer to the GPT320.GTCCRC register.

1. In the MRA register, select incrementation of the source address (MRA.SM[1:0] = 10b).
2. Set the transfer to normal transfer mode (MRA.MD[1:0] = 00b) and word-sized transfer (MRA.SZ[1:0] = 10b).
3. In the MRB register, select the destination address as fixed (MRB.DM[1:0] = 00b) and set up chain transfer (MRB.CHNE = 1 and MRB.CHNS = 0).
4. Set the SAR register to the first address of the data table.

5. Set the DAR register to the address of the GPT320.GTCCRC register.
6. Set the CRA register to the size of the data table.

(2) Second transfer information set

Set up for transfer to the GPT320.GTCCRE register.

1. In the MRA register, select incrementation of the source address (MRA.SM[1:0] = 10b).
2. Set the transfer to normal transfer mode (MRA.MD[1:0] = 00b) and word-sized transfer (MRA.SZ[1:0] = 10b).
3. In the MRB register, select the destination address as fixed (MRB.DM[1:0] = 00b) and set up chain transfer (MRB.CHNE = 1, MRB.CHNS = 0).
4. Set the SAR register to the first address of the data table.
5. Set the DAR register to the address of the GPT320.GTCCRE register.
6. Set the CRA register to the size of the data table.

(3) Third transfer information set

Set up transfer to the GPT320.GTPBR register.

1. In the MRA register, select incrementation of the source address (MRA.SM[1:0] = 10b).
2. Set the transfer to normal transfer mode (MRA.MD[1:0] = 00b) and word-sized transfer (MRA.SZ[1:0] = 10b).
3. In the MRB register, select the destination address as fixed (MRB.DM[1:0] = 00b) and end chain transfer (MRB.CHNE = 0, MRB.CHNS = 0, MRB.DISEL = 0).
4. Set the SAR register to the first address of the data table.
5. Set the DAR register to the address of the GPT320.GTPBR register.
6. Set the CRA register to the size of the data table.

(4) Transfer information assignment

Place the transfer information for use in the transfer to the GPT320.GTPBR immediately after the transfer control information for use in the GPT320.GTCCRC and GPT320.GTCCRE registers.

(5) DTC vector table

In the DTC vector table, set the address where the transfer control information for use in transfer to the GPT320.GTCCRC and GPT320.GTCCRE registers starts.

(6) ICU setting and DTC module activation

1. Set the ICU.IELSRn.DTCE bit associated with the GPT320 counter overflow interrupt.
2. Set the ICU.IELSRn.IELS[9:0] bits and specify the GPT320 counter overflow.
3. Set the DTCST.DTCST bit to 1.

(7) GPT settings

1. Set the GPT320.GTIOR register so that the GTCCRA and GTCCRB registers operate as output compare registers.
2. Set the default PWM timer compare values in the GPT320.GTCCRA and GPT320.GTCCRB registers and the next PWM timer compare values in the GPT320.GTCCRC and GPT320.GTCCRE registers.
3. Set the default PWM timer period values in the GPT320.GTPR register and the next PWM timer period values in the GPT320.GTPBR register.
4. Set 1 to the output bit in PmnPFS.PDR, and set 00011b to the Peripheral Select bits in PmnPFS.PSEL[4:0].

(8) GPT activation

Set the GPT320.GTSTR.CSTRT bits to 1 to start the GPT320.GTCNT counter.

(9) DTC transfer

Each time a GPT320 counter overflow is generated with the GPT320.GTPR register, the next PWM timer compare values are transferred to the GPT320.GTCCRC and GPT320.GTCCRE registers. The setting for the next PWM timer period is transferred to the GPT320.GTPBR register.

(10) Interrupt handling

After the specified rounds of data transfer are complete, for example when the value in the CRA register for GPT transfer becomes 0, a GPT320 counter overflow interrupt request is issued for the CPU. Complete the process for this interrupt in the handling routine.

18.6.3 Chain Transfer when Counter = 0

The second data transfer is performed only when the transfer counter is set to 0 in the first data transfer, and the first data transfer information is repeatedly changed in the second transfer. Chain transfer enables transfers to be repeated 256 times or more.

The following procedure shows an example of configuring a 1-KB input buffer, where the input buffer is set so that its lower address starts with 0x00. Figure 18.14 shows a chain transfer when the counter = 0.

1. Set the normal transfer mode to input data for the first data transfer. Set the following:
   (a) Transfer source address = fixed.
   (b) CRA register = 0x0200 (512) times.
   (c) MRB.CHNE bit = 1 (chain transfer is enabled).
   (d) MRB.CHNS bit = 1 (chain transfer is performed only when the transfer counter is 0).
   (e) MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when the specified data transfer completes).
2. Prepare the upper 8-bit address of the start address at every 512 times of the transfer destination address for the first data transfer in different area such as the flash. For example, when setting the input buffer to 0x8000 to 0x83FF, prepare 0x82 and 0x80.
3. For the second data transfer:
   (a) Set the repeat transfer mode (with transfer source and destination address = fixed.) to reset the transfer counter of the first data transfer.
   (b) Specify the CRA register in the first transfer information area for the transfer destination.
   (c) Set the MRB.CHNE bit = 1 (chain transfer is enabled).
   (d) Set the MRB.CHNS bit = 0 (select continuous chain transfer).
   (e) Set the MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when the specified data transfer completes).
   (f) CRA register = 0x0101 (The transfer count is 1).
4. For the third data transfer:
   (a) Set the repeat transfer mode (with the source as the repeat area) to reset the transfer destination address of the first data transfer.
   (b) Specify the upper 8 bits of the DAR register in the first transfer information area for the transfer destination.
   (c) Set the MRB.CHNE bit = 0 (chain transfer is disabled).
   (d) Set the MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when the specified data transfer completes).
   (e) When setting the input buffer to 0x8000 to 0x83FF, also set the transfer counter to 2.
5. The first data transfer is performed by an interrupt 512 times. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the transfer counter of the first data transfer to 0x0200. The lower 8 bits of the transfer destination address and the transfer counter of the first data transfer becomes 0x0200.
6. The second data transfer is performed by an interrupt 1 times. When the transfer counter of the first data transfer becomes 0, the third data transfer starts. Set the upper 8 bits of the transfer destination address of the first data transfer to

0x82. The lower 8 bits of the transfer destination address becomes 0x00 and the transfer counter of the first data transfer becomes 0x0200.

7. In succession, the first data transfer is performed by an interrupt 512 times as specified for the first data transfer. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the transfer counter of the first data transfer to 0x0200. The lower 8 bits of the transfer destination address and the transfer counter of the first data transfer becomes 0x0200.
8. The second data transfer is performed by an interrupt 1 times. When the transfer counter of the first data transfer becomes 0, the third data transfer starts. Set the upper 8 bits of the transfer destination address of the first data transfer to 0x80. The lower 8 bits of the transfer destination address becomes 0x00 and the transfer counter of the first data transfer becomes 0x0200.
9. Steps 5 to 8 are repeated indefinitely. Because the second data transfer is in repeat transfer mode, no interrupt request to the CPU is generated.

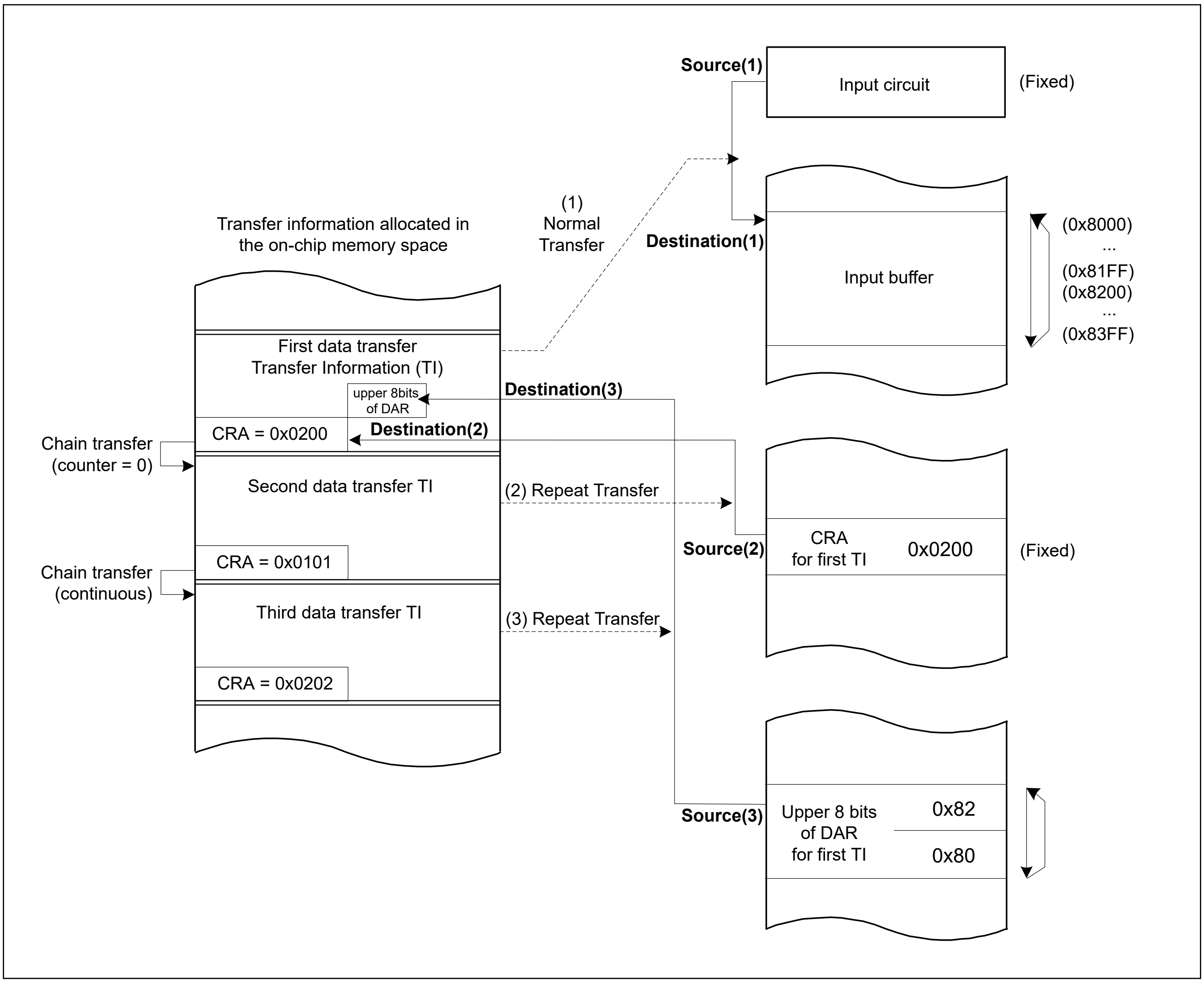


Figure 18.14 Chain transfer when counter = 0

18.7 Processing on DTC Transfer Error

If the access error occurs during DTC transfer, the DTC immediately stops access during transfer. DTC is stopped only for the vector number that caused the error.

The error response is informed to ICU when the transfer error occurs. ICU clears the ICU.IELSRn of the corresponding vector number which caused the transfer error. Write back to SRAM is not performed.

After that, if there is a request other than the vector number which caused the error, it will be re-arbitration as it is.

18.8 Interrupt

18.8.1 Interrupt Request of Transfer End

When the DTC completes data transfer of the specified count or when data transfer with MRB.DISEL set to 1 is complete, a DTC activation source generates an interrupt to the CPU. Interrupts to the CPU are controlled according to the settings in the NVIC and the ICU.IELSRn.IELS[9:0] bits. See section 14, Interrupt Controller Unit (ICU). The DTC prioritizes activation sources by granting the smaller interrupt vector numbers higher priority. The priority of interrupts to the CPU is determined by the NVIC priority.

18.8.2 Interrupt Request of Transfer Error

The error response detection interrupt request DMAm_TRANSERR (m = 0, 1) is generated from the DMAC/DTC when the transfer error is detected during DTC transfer.

The types of interrupts that occur when the DTC transfer error occurs are listed in Table 18.9. The Table 18.9 also shows error information stored when a transfer error occurs.

Table 18.9 Interrupt and error information due to DTC transfer error cause

Transfer error factor	NMI/RESET*1 Request	Interrupt Request	Bus Error Status	Error Channel Information
MSAU Error	ICU.NMISR.BUSST	DMAm_TRANSERR*1	BUS.BUSERRSTATDMACm.MESRRSTAT	DTC.DTEVR
Master MPU Error	ICU.NMISR.BUSST	DMAm_TRANSERR*1	BUS.BUSERRSTATDMACm.MMERRSTAT	DTC.DTEVR
Illegal Access Error	ICU.NMISR.BUSST	DMAm_TRANSERR*1	BUS.BUSERRSTATDMACm.ILERRSTAT	DTC.DTEVR
Slave Bus Error	ICU.NMISR.BUSST	DMAm_TRANSERR*1	BUS.BUSERRSTATDMACm.SLERRSTAT	DTC.DTEVR
TrustZone Filter error	ICU.NMISR.BUSST	DMAm_TRANSERR*1	BUS.BUSERRSTATDMACm.SLERRSTAT	DTC.DTEVR

Note 1. If ICU.NMIER.BUSEN is enabled with DMA0_TRANSERR set in ICU.IELSR, NMI and an interrupt will occur due to a transfer error caused by DMA. Only NMI can be generated by not setting DMAm_TRANSERR to ICU.IELSR.

18.9 Event Link

The DTC can produce an event link request on completion of one transfer request. The event link request is generated when the bus responds to the last destination access of DTC transfer. The timing of the bus response to DTC write access depends on the bus configuration and transfer destination. For details on bus responses, see section 15.2.5. Restrictions. Note that all DTC transfers are non-bufferable.

18.10 Low Power Consumption Function

Before transitioning to the module-stop state, Software Standby mode or Deep Software Standby mode, set the DTCST.DTCST bit to 0, and then perform the operations described in the following sections.

(1) Module-stop function

Setting the module-stop bit to 1 stops the DTC corresponding to that module-stop bit.

If the module-stop bit is set to 1 while DTC transfers data, the DTC corresponding to that bit becomes the module-stop state after completing it.

When the module-stop bit is 1, access to the register of the DTC corresponding to that module-stop bit is prohibited.

(2) Software Standby mode and Deep Software Standby mode

Use the settings described in section 11.6.2.1. Entering Software Standby Mode or section 11.6.3.1. Transition to Deep Software Standby Mode.

If DTC transfer operations are in progress when the WFI instruction is executed, the transition to Software Standby mode or Deep Software Standby mode follows the completion of the DTC transfer.

(3) Notes on Low Power Consumption Function

For the WFI instruction and the register setting procedure, see section 11, Low Power Mode.

To perform a DTC transfer after returning from a low power mode.

To use a request that is generated in Software Standby mode as an interrupt request to the CPU but not as a DTC activation request, specify the CPU as the interrupt request destination as described in section 14.4.1. Interrupt Detection Selection, then execute the WFI instruction.

18.11 Usage Notes

18.11.1 Transfer Information Start Address

You must set multiples of 16 for the transfer information start addresses in the vector table.

18.11.2 Arrangement of Transfer Information

When allocating the transfer information to the memory, arrange it as shown in Figure 18.15.

For example, when writing CRA and CRB setting data in 16 bits, write the CRB setting data to the lower address 0 and the CRA setting data to the lower address 2. When writing in 32 bits, place the CRA setting data on the 32-bit MSB side and the CRB setting data on the LSB side, and write them to the lower address 0.

Also, arrange the transfer information so that it does not straddle the 4KB boundary.

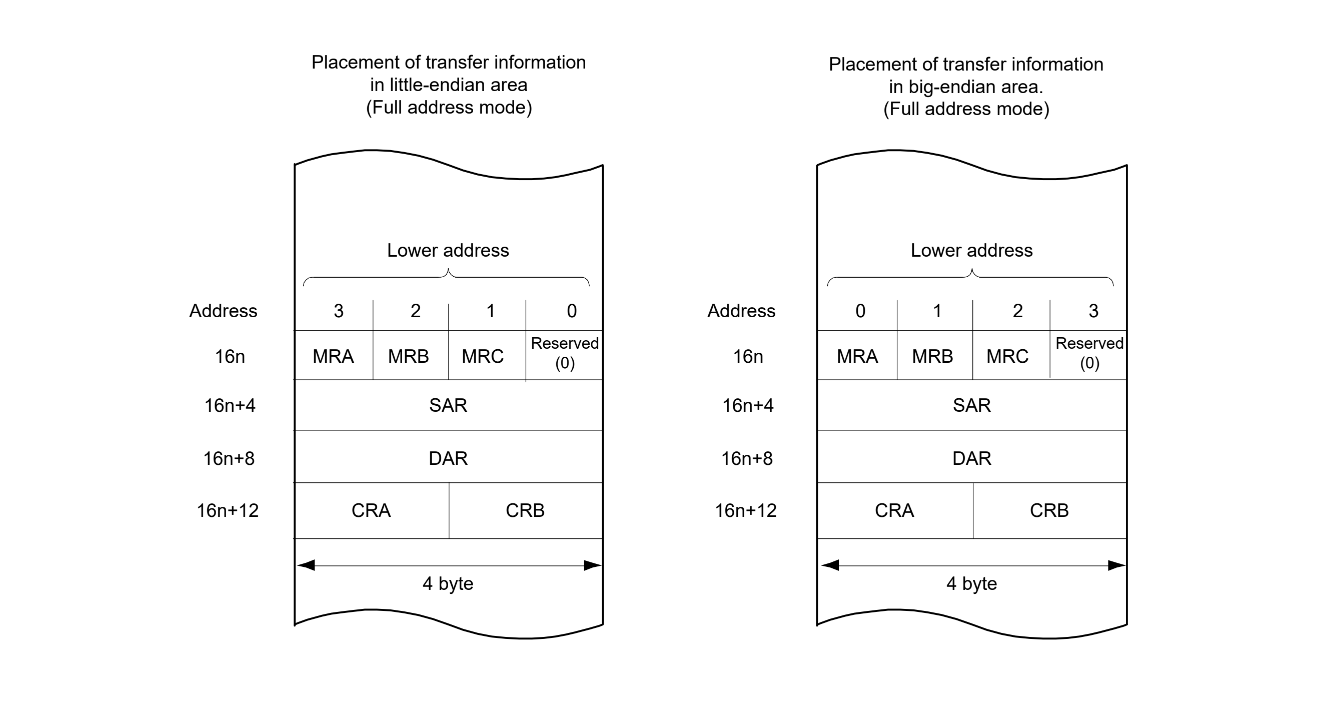


Figure 18.15 Arrangement of transfer information

18.11.3 Bus Response to DTC Write Access

The timing of the bus response to DTC write access depends on the transfer destination.

When writing to an access destination described in section 15.2.5.3. Notes on Writing Completion Timing, a bus response is returned before the write operation is completed.

19. Event Link Controller (ELC)

19.1 Overview

The Event Link Controller (ELC) uses the event requests generated by various peripheral modules as source signals to connect them to different modules, allowing direct link between the modules without CPU intervention.

Table 19.1 lists the ELC specifications, and Figure 19.1 shows a block diagram.

Table 19.1 ELC specifications

Item	Description
Event link function	322 types of event signals can be directly connected to modules. The ELC generates the ELC event signal, and events that activate the DTC.
Module-stop function	Module-stop state can be set to reduce power consumption
TrustZone Filter	Security and Privilege attribution can be set for each registers

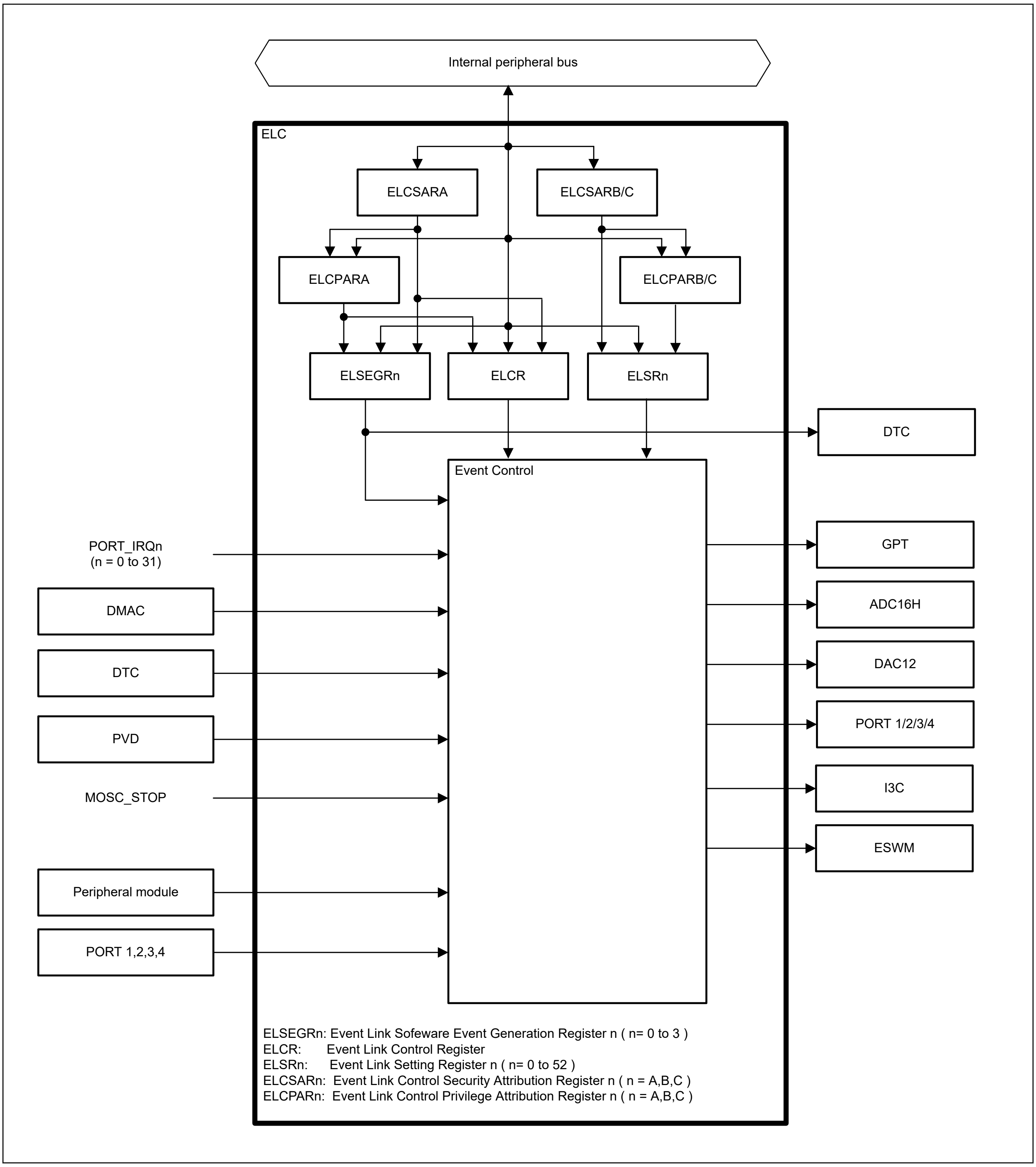


Figure 19.1 ELC block diagram

19.2 Register Descriptions

19.2.1 ELCR : Event Link Controller Register

Base address: ELC = 0x4020_1000
ELC_NS = 0x5020_1000

Offset address: 0x000

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

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	ELCON	All Event Link Enable 0: ELC function is disabled. 1: ELC function is enabled.	R/W

Note: S-TYPE-3, P-TYPE-3

The ELCR register controls the ELC operation.

19.2.2 ELSEGRn : Event Link Software Event Generation Register n (n = 0 to 3)

Base address: ELC = 0x4020_1000
ELC_NS = 0x5020_1000

Offset address: 0x004 + 0x004 × n

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

Bit	Symbol	Function	R/W
0	SEG	Software Event Generation 0: Normal operation 1: Software event is generated.	W
5:1	—	These bits are read as 0. The write value should be 0.	R/W
6	WE	SEG Bit Write Enable 0: Write to SEG bit disabled. 1: Write to SEG bit enabled.	R/W
7	WI	ELSEGR Register Write Disable 0: Write to ELSEGR register enabled. 1: Write to ELSEGR register disabled.	W

Note: S-TYPE-3, P-TYPE-3

SEG bit (Software Event Generation)

When 1 is written to the SEG bit while the WE bit is 1, a software event is generated. This bit is read as 0. Even when 1 is written to this bit, data is not stored. The WE bit must be set to 1 before writing to this bit.

A software event can trigger a linked DTC event.

WE bit (SEG Bit Write Enable)

The SEG bit can only be written to when the WE bit is 1. Clear the WI bit to 0 before writing to this bit.

[Setting condition]

- If 1 is written to this bit while the WI bit is 0, this bit becomes 1.

[Clearing condition]

- If 0 is written to this bit while the WI bit is 0, this bit becomes 0.

WI bit (ELSEGR Register Write Disable)

The ELSEGR register can only be written to when the write value to the WI bit is 0. This bit is read as 1. Before setting the WE or SEG bit, the WI bit must be set to 0.

19.2.3 ELSRn:Event Link Setting Register n (n = 0 to 52)

Base address: ELC = 0x4020_1000
ELC_NS = 0x5020_1000

Offset address: 0x020 + 0x004 × n

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

Bit	Symbol	Function	R/W
9:0	ELS[9:0]	Event Link Select 0x000: Event output disabled for the associated peripheral module 0x001: Number setting for the event signal to be linked ⋮ 0x1F8: Number setting for the event signal to be linked Others: Settings prohibited	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: P-TYPE-3

Note: This register has different secure access permission depending on the setting of Trusted Event Route Control Register (TEVTRCR). If the security attribution is configured as Secure,

- Secure access is allowed.
- Non-secure write access is ignored and Non-secure read access is read as 0, TrustZone access error is generated.

If the security attribution is configured as Non-secure and the trusted event route is disabled,

- Secure write access is ignored and Secure read access is read as 0, TrustZone access error is generated.
- Non-secure access is allowed.

If the security attribution is configured as Non-secure and the trusted event route is enabled,

- Secure access is allowed.
- Non-secure write access is ignored and Non-secure read access is allowed.
- TrustZone access error is not generated.

The ELSRn register specifies an event signal to be linked to each peripheral module. Table 19.2 shows the association between the ELSRn register and the peripheral modules. Table 19.3 shows the association between the event signal names set in the ELSRn register and the signal numbers.

Table 19.2 Association between the ELSRn registers and peripheral functions (1 of 2)

Register name	Peripheral function (module)	Event name
ELSR0	GPT (A)	ELC_GPTA
ELSR1	GPT (B)	ELC_GPTB
ELSR2	GPT (C)	ELC_GPTC
ELSR3	GPT (D)	ELC_GPTD
ELSR4	GPT (E)	ELC_GPTE
ELSR5	GPT (F)	ELC_GPTF
ELSR6	GPT (G)	ELC_GPTG
ELSR7	GPT (H)	ELC_GPTH

Table 19.2 Association between the ELSRn registers and peripheral functions (2 of 2)

Register name	Peripheral function (module)	Event name
ELSR12	DAC12 channel 0	ELC_DA0
ELSR13	DAC12 channel 1	ELC_DA1
ELSR14	PORT1	ELC_PORT1
ELSR15	PORT2	ELC_PORT2
ELSR16	PORT3	ELC_PORT3
ELSR17	PORT4	ELC_PORT4
ELSR19	ADC16HA0	ELC_HAD00
ELSR20	ADC16HB0	ELC_HAD01
ELSR21	ADC16HC0	ELC_HAD02
ELSR22	ADC16HA1	ELC_HAD10
ELSR23	ADC16HB1	ELC_HAD11
ELSR24	ADC16HC1	ELC_HAD12
ELSR25	ADC16HA2	ELC_HAD20
ELSR26	ADC16HB2	ELC_HAD21
ELSR27	ADC16HC2	ELC_HAD22
ELSR30	I3C	ELC_I3C
ELSR31	ESWM	ELC_GPTP0
ELSR32	ESWM	ELC_GPTP1

Table 19.3 Association between event signal names set in ELSRn.ELS[9:0] bits and signal numbers (1 of 9)

Event number	Interrupt request source	Name	Description
0x001	Port	PORT_IRQ0*1	External pin interrupt 0
0x002		PORT_IRQ1*1	External pin interrupt 1
0x003		PORT_IRQ2*1	External pin interrupt 2
0x004		PORT_IRQ3*1	External pin interrupt 3
0x005		PORT_IRQ4*1	External pin interrupt 4
0x006		PORT_IRQ5*1	External pin interrupt 5
0x007		PORT_IRQ6*1	External pin interrupt 6
0x008		PORT_IRQ7*1	External pin interrupt 7
0x009		PORT_IRQ8*1	External pin interrupt 8
0x00A		PORT_IRQ9*1	External pin interrupt 9
0x00B		PORT_IRQ10*1	External pin interrupt 10
0x00C		PORT_IRQ11*1	External pin interrupt 11
0x00D		PORT_IRQ12*1	External pin interrupt 12
0x00E		PORT_IRQ13*1	External pin interrupt 13
0x00F		PORT_IRQ14*1	External pin interrupt 14
0x010		PORT_IRQ15*1	External pin interrupt 15
0x011		PORT_IRQ16*1	External pin interrupt 16
0x012		PORT_IRQ17*1	External pin interrupt 17
0x013		PORT_IRQ18*1	External pin interrupt 18
0x014		PORT_IRQ19*1	External pin interrupt 19
0x015		PORT_IRQ20*1	External pin interrupt 20
0x016		PORT_IRQ21*1	External pin interrupt 21
0x017		PORT_IRQ22*1	External pin interrupt 22
0x018		PORT_IRQ23*1	External pin interrupt 23
0x019		PORT_IRQ24*1	External pin interrupt 24
0x01A		PORT_IRQ25*1	External pin interrupt 25
0x01B		PORT_IRQ26*1	External pin interrupt 26
0x01C		PORT_IRQ27*1	External pin interrupt 27
0x01D		PORT_IRQ28*1	External pin interrupt 28
0x01E		PORT_IRQ29*1	External pin interrupt 29
0x01F		PORT_IRQ30*1	External pin interrupt 30
0x020		PORT_IRQ31*1	External pin interrupt 31

Table 19.3 Association between event signal names set in ELSRn.ELS[9:0] bits and signal numbers (2 of 9)

Event number	Interrupt request source	Name	Description
0x040	DMAC0	DMAC00_INT	DMAC0 transfer end 0
0x041		DMAC01_INT	DMAC0 transfer end 1
0x042		DMAC02_INT	DMAC0 transfer end 2
0x043		DMAC03_INT	DMAC0 transfer end 3
0x044		DMAC04_INT	DMAC0 transfer end 4
0x045		DMAC05_INT	DMAC0 transfer end 5
0x046		DMAC06_INT	DMAC0 transfer end 6
0x047		DMAC07_INT	DMAC0 transfer end 7
0x048	DMAC1	DMAC10_INT	DMAC1 transfer end 0
0x049		DMAC11_INT	DMAC1 transfer end 1
0x04A		DMAC12_INT	DMAC1 transfer end 2
0x04B		DMAC13_INT	DMAC1 transfer end 3
0x04C		DMAC14_INT	DMAC1 transfer end 4
0x04D		DMAC15_INT	DMAC1 transfer end 5
0x04E		DMAC16_INT	DMAC1 transfer end 6
0x04F		DMAC17_INT	DMAC1 transfer end 7
0x052	DTC0	DTC0_DTCEND	DTC0 transfer end
0x055	DTC1	DTC1_DTCEND	DTC1 transfer end
0x070	PVD	PVD_PVD1	Voltage monitor 1 interrupt
0x071		PVD_PVD2	Voltage monitor 2 interrupt
0x076	MOSC	MOSC_STOP	Mail Clock oscillation stop
0x080	ULPT0	ULPT0_ULPTI	Underflow
0x081		ULPT0_ULPTCMAI	Compare match A
0x082		ULPT0_ULPTCMBI	Compare match B
0x083	ULPT1	ULPT1_ULPTI	Underflow
0x084		ULPT1_ULPTCMAI	Compare match A
0x085		ULPT1_ULPTCMBI	Compare match B
0x086	AGT0	AGT0_AGTI	AGT interrupt
0x087		AGT0_AGTCMAI	Compare match A
0x088		AGT0_AGTCMBI	Compare match B
0x089	AGT1	AGT1_AGTI	AGT interrupt
0x08A		AGT1_AGTCMAI	Compare match A
0x08B		AGT1_AGTCMBI	Compare match B
0x092	IWDT	IWDT_NMIUNDF	IWDT underflow
0x093	WDT	WDT0_NMIUNDF	WDT underflow
0x094		WDT1_NMIUNDF	WDT underflow
0x096	RTC	RTC_PRD	Periodic interrupt
0x09C	IIC0	IIC0_RXI	Receive data full
0x09D		IIC0_TXI	Transmit data empty
0x09E		IIC0_TEI	Transmit end
0x09F		IIC0_EEI	Transfer error

Table 19.3 Association between event signal names set in ELSRn.ELS[9:0] bits and signal numbers (3 of 9)

Event number	Interrupt request source	Name	Description
0x0A1	IIC1	IIC1_RXI	Receive data full
0x0A2		IIC1_TXI	Transmit data empty
0x0A3		IIC1_TEI	Transmit end
0x0A4		IIC1_EEI	Transfer error
0x0A6	IIC2	IIC2_RXI	Receive data full
0x0A7		IIC2_TXI	Transmit data empty
0x0A8		IIC2_TEI	Transmit end
0x0A9		IIC2_EEI	Transfer error
0x0C6	ACMPHS	ACMP_HS0*1	Comparator interrupt 0
0x0C7		ACMP_HS1*1	Comparator interrupt 1
0x0C8		ACMP_HS2*1	Comparator interrupt 2
0x0C9		ACMP_HS3*1	Comparator interrupt 3
0x0CC	ELC	ELC_SWEVT0	Software event 0
0x0CD		ELC_SWEVT1	Software event 1
0x0CE		ELC_SWEVT2	Software event 2
0x0CF		ELC_SWEVT3	Software event 3
0x0D0	I/O Port	IOPORT_GROUP1	Port 1 event
0x0D1		IOPORT_GROUP2	Port 2 event
0x0D2		IOPORT_GROUP3	Port 3 event
0x0D3		IOPORT_GROUP4	Port 4 event
0x180	GPT	GPT_UVWEDGE	UVW edge event
0x181	GPT0	GPT0_CCMPA	Compare match A
0x182		GPT0_CCMPB	Compare match B
0x183		GPT0_CMPC	Compare match C
0x184		GPT0_CMPD	Compare match D
0x185		GPT0_CMPE	Compare match E
0x186		GPT0_CMPF	Compare match F
0x187		GPT0_OVF	Overflow
0x188		GPT0_UDF	Underflow
0x189		GPT0_PC	Cycle count function end
0x18A	GPT1	GPT1_CCMPA	Compare match A
0x18B		GPT1_CCMPB	Compare match B
0x18C		GPT1_CMPC	Compare match C
0x18D		GPT1_CMPD	Compare match D
0x18E		GPT1_CMPE	Compare match E
0x18F		GPT1_CMPF	Compare match F
0x190		GPT1_OVF	Overflow
0x191		GPT1_UDF	Underflow
0x192		GPT1_PC	Cycle count function end

Table 19.3 Association between event signal names set in ELSRn.ELS[9:0] bits and signal numbers (4 of 9)

Event number	Interrupt request source	Name	Description
0x193	GPT2	GPT2_CCMPA	Compare match A
0x194		GPT2_CCMPB	Compare match B
0x195		GPT2_CMPC	Compare match C
0x196		GPT2_CMPD	Compare match D
0x197		GPT2_CMPE	Compare match E
0x198		GPT2_CMPF	Compare match F
0x199		GPT2_OVF	Overflow
0x19A		GPT2_UDF	Underflow
0x19B		GPT2_PC	Cycle count function end
0x19C	GPT3	GPT3_CCMPA	Compare match A
0x19D		GPT3_CCMPB	Compare match B
0x19E		GPT3_CMPC	Compare match C
0x19F		GPT3_CMPD	Compare match D
0x1A0		GPT3_CMPE	Compare match E
0x1A1		GPT3_CMPF	Compare match F
0x1A2		GPT3_OVF	Overflow
0x1A3		GPT3_UDF	Underflow
0x1A4		GPT3_PC	Cycle count function end
0x1A5	GPT4	GPT4_CCMPA	Compare match A
0x1A6		GPT4_CCMPB	Compare match B
0x1A7		GPT4_CMPC	Compare match C
0x1A8		GPT4_CMPD	Compare match D
0x1A9		GPT4_CMPE	Compare match E
0x1AA		GPT4_CMPF	Compare match F
0x1AB		GPT4_OVF	Overflow
0x1AC		GPT4_UDF	Underflow
0x1AE	GPT5	GPT5_CCMPA	Compare match A
0x1AF		GPT5_CCMPB	Compare match B
0x1B0		GPT5_CMPC	Compare match C
0x1B1		GPT5_CMPD	Compare match D
0x1B2		GPT5_CMPE	Compare match E
0x1B3		GPT5_CMPF	Compare match F
0x1B4		GPT5_OVF	Overflow
0x1B5		GPT5_UDF	Underflow
0x1B7	GPT6	GPT6_CCMPA	Compare match A
0x1B8		GPT6_CCMPB	Compare match B
0x1B9		GPT6_CMPC	Compare match C
0x1BA		GPT6_CMPD	Compare match D
0x1BB		GPT6_CMPE	Compare match E
0x1BC		GPT6_CMPF	Compare match F
0x1BD		GPT6_OVF	Overflow
0x1BE		GPT6_UDF	Underflow

Table 19.3 Association between event signal names set in ELSRn.ELS[9:0] bits and signal numbers (5 of 9)

Event number	Interrupt request source	Name	Description
0x1C0	GPT7	GPT7_CCMPA	Compare match A
0x1C1		GPT7_CCMPB	Compare match B
0x1C2		GPT7_CMPC	Compare match C
0x1C3		GPT7_CMPD	Compare match D
0x1C4		GPT7_CMPE	Compare match E
0x1C5		GPT7_CMPF	Compare match F
0x1C6		GPT7_OVF	Overflow
0x1C7		GPT7_UDF	Underflow
0x1C9	GPT8	GPT8_CCMPA	Compare match A
0x1CA		GPT8_CCMPB	Compare match B
0x1CB		GPT8_CMPC	Compare match C
0x1CC		GPT8_CMPD	Compare match D
0x1CD		GPT8_CMPE	Compare match E
0x1CE		GPT8_CMPF	Compare match F
0x1CF		GPT8_OVF	Overflow
0x1D0		GPT8_UDF	Underflow
0x1D2	GPT9	GPT9_CCMPA	Compare match A
0x1D3		GPT9_CCMPB	Compare match B
0x1D4		GPT9_CMPC	Compare match C
0x1D5		GPT9_CMPD	Compare match D
0x1D6		GPT9_CMPE	Compare match E
0x1D7		GPT9_CMPF	Compare match F
0x1D8		GPT9_OVF	Overflow
0x1D9		GPT9_UDF	Underflow
0x1DB	GPT10	GPT10_CCMPA	Compare match A
0x1DC		GPT10_CCMPB	Compare match B
0x1DD		GPT10_CMPC	Compare match C
0x1DE		GPT10_CMPD	Compare match D
0x1DF		GPT10_CMPE	Compare match E
0x1E0		GPT10_CMPF	Compare match F
0x1E1		GPT10_OVF	Overflow
0x1E2		GPT10_UDF	Underflow
0x1E3		GPT10_PC	Cycle count function end
0x1E4	GPT11	GPT11_CCMPA	Compare match A
0x1E5		GPT11_CCMPB	Compare match B
0x1E6		GPT11_CMPC	Compare match C
0x1E7		GPT11_CMPD	Compare match D
0x1E8		GPT11_CMPE	Compare match E
0x1E9		GPT11_CMPF	Compare match F
0x1EA		GPT11_OVF	Overflow
0x1EB		GPT11_UDF	Underflow
0x1EC		GPT11_PC	Cycle count function end

Table 19.3 Association between event signal names set in ELSRn.ELS[9:0] bits and signal numbers (6 of 9)

Event number	Interrupt request source	Name	Description
0x1ED	GPT12	GPT12_CCMPA	Compare match A
0x1EE		GPT12_CCMPB	Compare match B
0x1EF		GPT12_CMPC	Compare match C
0x1F0		GPT12_CMPD	Compare match D
0x1F1		GPT12_CMPE	Compare match E
0x1F2		GPT12_CMPF	Compare match F
0x1F3		GPT12_OVF	Overflow
0x1F4		GPT12_UDF	Underflow
0x1F5		GPT12_PC	Cycle count function end
0x1F6	GPT13	GPT13_CCMPA	Compare match A
0x1F7		GPT13_CCMPB	Compare match B
0x1F8		GPT13_CMPC	Compare match C
0x1F9		GPT13_CMPD	Compare match D
0x1FA		GPT13_CMPE	Compare match E
0x1FB		GPT13_CMPF	Compare match F
0x1FC		GPT13_OVF	Overflow
0x1FD		GPT13_UDF	Underflow
0x1FE		GPT13_PC	Cycle count function end
0x2B4	ESWM	GPTP_PTPOUT0	PTP Pulse Out 0
0x2B5		GPTP_PTPOUT1	PTP Pulse Out 1
0x2B6		GPTP_PTPOUT2	PTP Pulse Out 2
0x2B7		GPTP_PTPOUT3	PTP Pulse Out 3
0x2B8		GPTP_MATCH0	Media Clock Recovery Match 0
0x2B9		GPTP_MATCH1	Media Clock Recovery Match 1
0x2C4	SCI0	SCI0_RXI*2	Receive data full
0x2C5		SCI0_TXI*2	Transmit data empty
0x2C6		SCI0_TEI*2	Transmit end
0x2C7		SCI0_ERI	Receive error
0x2C8		SCI0_AED	Effective edge detection
0x2CA		SCI0_AM	Address match event
0x2CB	SCI1	SCI1_RXI*2	Receive data full
0x2CC		SCI1_TXI*2	Transmit data empty
0x2CD		SCI1_TEI*2	Transmit end
0x2CE		SCI1_ERI	Receive error
0x2CF		SCI1_AED	Effective edge detection
0x2D1		SCI1_AM	Address match event

Table 19.3 Association between event signal names set in ELSRn.ELS[9:0] bits and signal numbers (7 of 9)

Event number	Interrupt request source	Name	Description
0x2D2	SCI2	SCI2_RXI*2	Receive data full
0x2D3		SCI2_TXI*2	Transmit data empty
0x2D4		SCI2_TEI*2	Transmit end
0x2D5		SCI2_ERI	Receive error
0x2D6		SCI2_AED	Effective edge detection
0x2D8		SCI2_AM	Address match event
0x2D9	SCI3	SCI3_RXI*2	Receive data full
0x2DA		SCI3_TXI*2	Transmit data empty
0x2DB		SCI3_TEI*2	Transmit end
0x2DC		SCI3_ERI	Receive error
0x2DD		SCI3_AED	Effective edge detection
0x2DF		SCI3_AM	Address match event
0x2E0	SCI4	SCI4_RXI*2	Receive data full
0x2E1		SCI4_TXI*2	Transmit data empty
0x2E2		SCI4_TEI*2	Transmit end
0x2E3		SCI4_ERI	Receive error
0x2E4		SCI4_AED	Effective edge detection
0x2E6		SCI4_AM	Address match event
0x2E7	SCI5	SCI5_RXI*2	Receive data full
0x2E8		SCI5_TXI*2	Transmit data empty
0x2E9		SCI5_TEI*2	Transmit end
0x2EA		SCI5_ERI	Receive error
0x2EB		SCI5_AED	Effective edge detection
0x2ED		SCI5_AM	Address match event
0x2EE	SCI6	SCI6_RXI*2	Receive data full
0x2EF		SCI6_TXI*2	Transmit data empty
0x2F0		SCI6_TEI*2	Transmit end
0x2F1		SCI6_ERI	Receive error
0x2F2		SCI6_AED	Effective edge detection
0x2F4		SCI6_AM	Address match event
0x2F5	SCI7	SCI7_RXI*2	Receive data full
0x2F6		SCI7_TXI*2	Transmit data empty
0x2F7		SCI7_TEI*2	Transmit end
0x2F8		SCI7_ERI	Receive error
0x2F9		SCI7_AED	Effective edge detection
0x2FB		SCI7_AM	Address match event

Table 19.3 Association between event signal names set in ELSRn.ELS[9:0] bits and signal numbers (8 of 9)

Event number	Interrupt request source	Name	Description
0x2FC	SCI8	SCI8_RXI*2	Receive data full
0x2FD		SCI8_TXI*2	Transmit data empty
0x2FE		SCI8_TEI*2	Transmit end
0x2FF		SCI8_ERI	Receive error
0x300		SCI8_AED	Effective edge detection
0x302		SCI8_AM	Address match event
0x303	SCI9	SCI9_RXI*2	Receive data full
0x304		SCI9_TXI*2	Transmit data empty
0x305		SCI9_TEI*2	Transmit end
0x306		SCI9_ERI	Receive error
0x307		SCI9_AED	Effective edge detection
0x309		SCI9_AM	Address match event
0x318	SPI0	SPI0_SPRI	Receive buffer full
0x319		SPI0_SPTI	Transmit buffer empty
0x31A		SPI0_SPII	Idle
0x31B		SPI0_SPEI	Error
0x31C		SPI0_SPCEND	Communication complete event
0x31D	SPI1	SPI1_SPRI	Receive buffer full
0x31E		SPI1_SPTI	Transmit buffer empty
0x31F		SPI1_SPII	Idle
0x320		SPI1_SPEI	Error
0x321		SPI1_SPCEND	Transmission complete event
0x33A	I3C	I3C_RESP	Normal Response queue full
0x33B		I3C_CMD	Normal Command queue empty
0x33C		I3C_IBI	Normal IBI Queue Empty/Full
0x33D		I3C_RX	Normal Rx Data buffer full
0x33E		I3C_TX	Normal Tx Data buffer empty
0x33F		I3C_RCV	Normal Receive Status queue full
0x340		I3C_HRESP	High priority response queue full
0x341		I3C_HCMD	High priority command queue empty
0x342		I3C_HRX	High priority Rx Data buffer full
0x343		I3C_HTX	High priority Tx Data buffer empty
0x344		I3C_TEND	Transmit end
0x345		I3C_EEI	Transfer error or event occurrence
0x346		I3C_STEV	Synchronous Timing
0x347		I3C_MREFOVF	MREF Counter Overflow
0x348		I3C_MREFCPT	MREF Capture
0x349		I3C_AMEV	Additional Master-initiated bus Event

Table 19.3 Association between event signal names set in ELSRn.ELS[9:0] bits and signal numbers (9 of 9)

Event number	Interrupt request source	Name	Description
0x34D	ADC16H	ADC_ADI0	A/D scan end interrupt for group 0
0x34E		ADC_ADI1	A/D scan end interrupt for group 1
0x34F		ADC_ADI2	A/D scan end interrupt for group 2
0x350		ADC_ADI3	A/D scan end interrupt for group 3
0x351		ADC_ADI4	A/D scan end interrupt for group 4
0x352		ADC_ADI5	A/D scan end interrupt for group 5
0x353		ADC_ADI6	A/D scan end interrupt for group 6
0x354		ADC_ADI7	A/D scan end interrupt for group 7
0x355		ADC_ADI8	A/D scan end interrupt for group 8
0x361		ADC_CCMPM0	Composite condition compare 0 match
0x362		ADC_CCMPUM0	Composite condition compare 0 mismatch
0x369		ADC_CCMPM1	Composite condition compare 1 match
0x36A		ADC_CCMPUM1	Composite condition compare 1 mismatch
0x36F	DOC	DOC_DOPCI	Data operation circuit interrupt

Note 1. Only pulse (edge detection) is supported.
Note 2. This event is not supported in FIFO mode.

19.2.4 ELCSARA : Event Link Controller Security Attribution Register A

Base address: ELC = 0x4020_1000
ELC_NS = 0x5020_1000

Offset address: 0x100

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	ELSEGR3	ELSEGR2	ELSEGR1	ELSEGR0	ELCR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ELCR	Event Link Controller Register Security Attribution Target register: ELCR 0: Secure 1: Non-secure	R/W
1	ELSEGR0	Event Link Software Event Generation Register 0 Security Attribution 0: Secure 1: Non-secure	R/W
2	ELSEGR1	Event Link Software Event Generation Register 1 Security Attribution 0: Secure 1: Non-secure	R/W
3	ELSEGR2	Event Link Software Event Generation Register 2 Security Attribution 0: Secure 1: Non-secure	R/W
4	ELSEGR3	Event Link Software Event Generation Register 3 Security Attribution 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1
Note: This register is write-protected by PRCR_S register.

This register specifies the security attribution for the ELCR, ELSEGR0, ELSEGR1, ELSEGR2, and ELSEGR3 registers.

19.2.5 ELCSARB : Event Link Controller Security Attribution Register B

Base address: ELC = 0x4020_1000
ELC_NS = 0x5020_1000

Offset address: 0x104

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

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

Bit	Symbol	Function	R/W
7:0	ELSR7 to ELSR0	Event Link Setting Register n Security Attribution Target register: ELSRn (n = 0 to 7) 0: Secure 1: Non-secure	R/W
11:8	—	These bits are read as 0. The write value should be 0.	R/W
17:12	ELSR17 to ELSR12	Event Link Setting Register n Security Attribution Target register: ELSRn (n = 12 to 17) 0: Secure 1: Non-secure	R/W
18	—	This bit is read as 0. The write value should be 0.	R/W
27:19	ELSR27 to ELSR19	Event Link Setting Register n Security Attribution Target register: ELSRn (n = 19 to 27) 0: Secure 1: Non-secure	R/W
29:28	—	These bits are read as 0. The write value should be 0.	R/W
31:30	ELSR31 to ELSR30	Event Link Setting Register n Security Attribution Target register: ELSRn (n = 30, 31) 0: Secure 1: Non-secure	R/W

Note: S-TYPE-1, P-TYPE-1
Note: This register is write-protected by PRCR_S register.

This register specifies the security attribution for the Register ELSRn (n = 0 to 7, 12 to 17, 19 to 27, 30, 31).

19.2.6 ELCSARC : Event Link Controller Security Attribution Register C

Base address: ELC = 0x4020_1000
ELC_NS = 0x5020_1000

Offset address: 0x108

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	ELSR[52:48]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ELSR[47:32]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
20:0	ELSR[52:32]	Event Link Setting Register n Security Attribution Target register: ELSRn (n = 32 to 52) 0: Secure 1: Non-secure	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1
Note: This register is write-protected by PRCR_S register.

This register specifies the security attribution for the Register ELSRn (n = 32 to 52).

19.2.7 ELCPARA : Event Link Controller Privilege Attribution Register A

Base address: ELC = 0x4020_1000
ELC_NS = 0x5020_1000

Offset address: 0x110

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	ELSEGR3	ELSEGR2	ELSEGR1	ELSEGR0	ELCR
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	ELCR	Event Link Controller Register Privilege Attribution Target register: ELCR 0: Privileged 1: Unprivileged	R/W
1	ELSEGR0	Event Link Software Event Generation Register 0 Privilege Attribution Target register: ELSEGR0 0: Privileged 1: Unprivileged	R/W
2	ELSEGR1	Event Link Software Event Generation Register 1 Privilege Attribution Target register: ELSEGR1 0: Privileged 1: Unprivileged	R/W

Bit	Symbol	Function	R/W
3	ELSEGR2	Event Link Software Event Generation Register 2 Privilege Attribution Target register: ELSEGR2 0: Privileged 1: Unprivileged	R/W
4	ELSEGR3	Event Link Software Event Generation Register 3 Privilege Attribution Target register: ELSEGR3 0: Privileged 1: Unprivileged	R/W
31:5	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-2, P-TYPE-1
Note: This register is write-protected by PRCR_S and PRCR_NS register.

This register specifies the privilege attribution for the Register ELCR, ELSEGR0, ELSEGR1, ELSEGR2, and ELSEGR3 registers.

19.2.8 ELCPARB : Event Link Controller Privilege Attribution Register B

Base address: ELC = 0x4020_1000
ELC_NS = 0x5020_1000

Offset address: 0x114

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

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

Bit	Symbol	Function	R/W
7:0	ELSR7 to ELSR0	Event Link Setting Register n Privilege Attribution Target register: ELSRn (n = 0 to 7) 0: Privileged 1: Unprivileged	R/W
11:8	—	These bits are read as 1. The write value should be 1.	R/W
17:12	ELSR17 to ELSR12	Event Link Setting Register n Privilege Attribution Target register: ELSRn (n = 12 to 17) 0: Privileged 1: Unprivileged	R/W
18	—	This bit is read as 1. The write value should be 1.	R/W
27:19	ELSR27 to ELSR19	Event Link Setting Register n Privilege Attribution Target register: ELSRn (n = 19 to 27) 0: Privileged 1: Unprivileged	R/W
29:28	—	These bits are read as 1. The write value should be 1.	R/W
31:30	ELSR31 to ELSR30	Event Link Setting Register n Privilege Attribution Target register: ELSRn (n = 30, 31) 0: Privileged 1: Unprivileged	R/W

Note: S-TYPE-2, P-TYPE-1
Note: This register is write-protected by PRCR_S and PRCR_NS register.

This register specifies the privilege attribution for the Register ELSRn (n = 0 to 7, 12 to 17, 19 to 27, 30, 31) registers.

19.2.9 ELCPARC : Event Link Controller Privilege Attribution Register C

Base address: ELC = 0x4020_1000
ELC_NS = 0x5020_1000

Offset address: 0x118

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	ELSR[52:48]				
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ELSR[47:32]															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
20:0	ELSR[52:32]	Event Link Setting Register n Privilege Attribution Target register: ELSRn (n = 32 to 52) 0: Privileged 1: Unprivileged	R/W
31:21	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-2, P-TYPE-1
Note: This register is write-protected by PRCR_S and PRCR_NS register.

This register specifies the privilege attribution for the Register ELSRn (n = 32 to 52) registers.

19.3 Operation

19.3.1 Relation Between Interrupt Handling and Event Linking

Event number for an event link is the same as that for the associated interrupt source. For information on generating event signals, see the explanation in the chapter for each event source module.

19.3.2 Linking Events

When an event occurs and that event is already set as a trigger in the Event Link Setting Register (ELSRn), the associated module is activated. The operation of the module must be set up in advance. Table 19.4 lists the operations of modules when an event occurs.

Table 19.4 Module operations when event occurs (1 of 2)

Module	Operations when event is input
GPT	• Start counting • Stop counting • Clear counting • Up counting • Down counting • Input capture
DAC12	Start D/A conversion
I/O Ports	• Change pin output based on the EORR (reset) or EOSR (set) • Latch pin state to EIDR • The following ports can be used for the ELC: PORT 1 PORT 2 PORT 3 PORT 4
I3C	Start operation
ADC16H	Start A/D conversion

Table 19.4 Module operations when event occurs (2 of 2)

Module	Operations when event is input
DTC	Start DTC data transfer
ESWM	Media clock capture event occurs

19.3.3 Example of Procedure for Linking Events

To link events:

1. Set the operation of the module for which an event is to be linked.
2. Set the appropriate ELSRn.ELS[9:0] bits for the module to be linked.
3. Set the ELCR.ELCON bit to 1 to enable linkage of all events.
4. Configure the module from which an event is output and activate the module. The link between the two modules is now active.
5. To stop event linkage of modules individually, set 0 to the ELSRn.ELS[9:0] bits associated with the modules. To stop linkage of all the events, set the ELCR.ELCON bit to 0.

If event link output from the PVD is to be used, set the ELC after setting the PVD. To disable the PVD, do so after setting 0x00 to the associated ELSRn register.

19.4 Usage Notes

19.4.1 Linking DMAC/DTC Transfer End Signals as Events

When linking the DMAC/DTC transfer end signals as events, do not set the same peripheral module as the DMAC/DTC transfer destination and event link destination. If set, the peripheral module might be started before DMAC/DTC transfer to the peripheral module is complete.

19.4.2 Setting Clocks

To link events, you must enable the ELC and the related modules. The modules cannot operate if the related modules are in the module-stop state or in low power mode in which the module is stopped (CPU Deep Sleep mode, Software Standby mode or Deep Software Standby mode). For more information, seeTable 19.3 and section 11, Low Power Mode.

19.4.3 Module-Stop Function Setting

The Module Stop Control Register C (MSTPCRC) can enable or disable ELC operation. The ELC is initially stopped after reset. Releasing the module-stop state enables access to the registers. The ELCON bit must be set to 0 before disabling ELC operation using the Module Stop Control Register. For more information, see Table 19.3 and section 11, Low Power Mode.

19.4.4 ELC Delay Time

In Figure 19.2, module A accesses module B through the ELC. There is a delay time in the ELC between module A and module B. Table 19.5 shows the ELC delay time.

Table 19.6 shows the clock domain of each module A. Table 19.7 shows the clock domain of each module B.

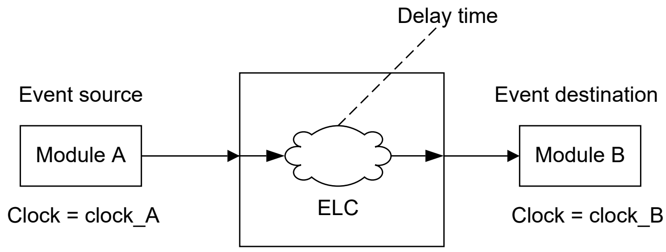


Figure 19.2 ELC delay time

Table 19.5 ELC delay time

Clock domain	Clock frequency	ELC delay time
clock_A = clock_B	clock_A = clock_B	0 cycle
clock_A ≠ clock_B	clock_A = clock_B	1 cycle to 2 cycles
	clock_A > clock_B	1 cycle to 2 cycles of clock_B
	clock_A < clock_B	1 cycle to 2 cycles of clock_A
clock_A ≠ clock_B	clock_A = clock_B*2	0 cycle
(Asynchronous)*1	—	5 cycles of clock_A + 4 cycles of clock_B

Note 1. When GPTCLK is selected as the GPT count clock, use this formula.
When ADC16H is destination and ADCCLK or GPTCLK is selected as the A/D conversion clock, use this formula.
Note 2. In case GPTCLK is selected as the GPT count clock, ADC conversion clock.

Table 19.6 Clock domain of each module A

Module A	Clock domain of clock_A
SCI, SPI, I3C, ADC16H, DOC, ESWM	PCLKA
Port, DMAC, PVD1, PVD2, MOSC, ULPT, AGT, IWDT, WDT, IIC, ACMPHS, ELC, I/O Ports	PCLKB
GPT	GPTCLK or PCLKD

Table 19.7 Clock domain of each module B

Module B	Clock domain of clock_B
I3C, ESWM	PCLKA
I/O Ports, DAC12	PCLKB
GPT	GPTCLK or PCLKD
ADC16H	ADCLK, GPTCLK, or PCLKA

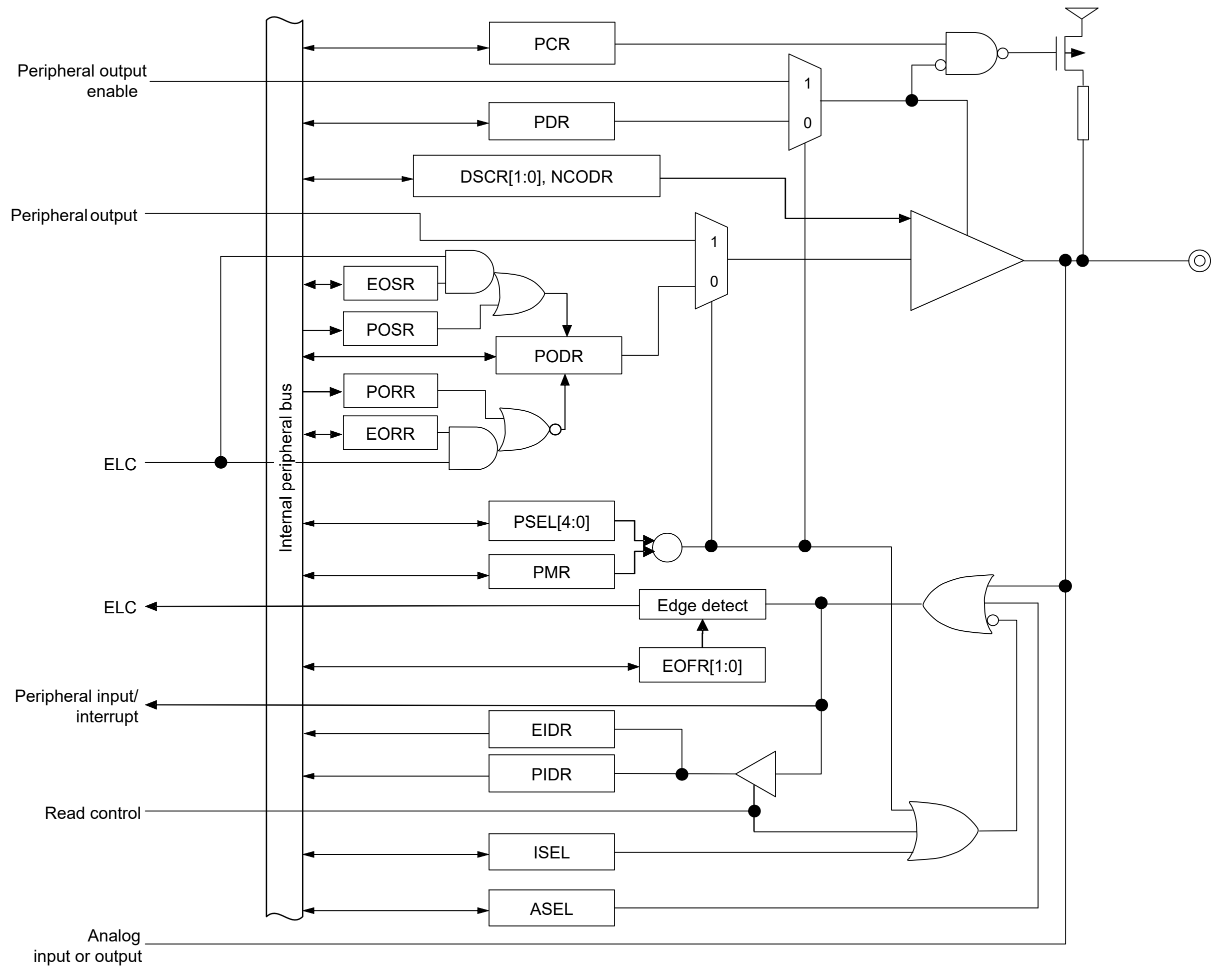
20. I/O Ports

20.1 Overview

The I/O port pins operate as general I/O port pins, I/O pins for peripheral modules, interrupt input pins, analog I/O, port group function for the ELC or bus control pins.

All pins except P209 (as TDO of JTAG ports) operate as input pins immediately after a reset, and pin functions are switched by register settings. The I/O ports and peripheral modules for each pin are specified in the associated registers.

Figure 20.1 shows a connection diagram for the I/O port registers. The configuration of the I/O ports differs for different packages. Table 20.1 lists the I/O port specifications by package, and Table 20.2 lists the port functions.



Note: This figure shows a basic port configuration. The configuration differs depending on the ports.

Figure 20.1 Connection diagram for I/O port registers

Table 20.1 I/O port specifications (1 of 2)

Port	Package 289, 303 pins	Number of Pins	Package 224 pins	Number of Pins	Package 176 pins	Number of Pins	Package 169 pins	Number of Pins
PORT0	P000 to P015	16	P000 to P009, P014, P015	12	P000 to P009, P014, P015	12	P000 to P009, P014, P015	12
PORT1	P100 to P115	16	P100 to P107, P112 to P115	12	P100 to P107, P112 to P115	12	P100 to P104, P106, P107, P112 to P115	11

Table 20.1 I/O port specifications (2 of 2)

Port	Package 289, 303 pins	Number of Pins	Package 224 pins	Number of Pins	Package 176 pins	Number of Pins	Package 169 pins	Number of Pins
PORT2	P200 to P215	16	P200 to P206, P208 to P215	15	P200, P201, P206, P208 to P215	11	P200, P201, P206, P208 to P215	11
PORT3	P300 to P315	16	P300 to P302, P304 to P315	15	P300 to P302, P304 to P308, P310 to P312	11	P300 to P302, P306, P307, P310 to P312	8
PORT4	P400 to P415	16	P400 to P411, P414, P415	14	P400 to P410, P414, P415	13	P400 to P405, P407 to P410, P414, P415	12
PORT5	P500 to P515	16	P503 to P512	10	P503 to P512	10	P500 to P502, P509 to P515	10
PORT6	P600 to P615*1	16	P600 to P604, P608 to P615	13	P600, P601, P608 to P615	10	P602, P603, P609, P610, P612 to P615	8
PORT7	P700 to P715	16	P700 to P714	15	P700 to P705, P707 to P709	9	P700 to P703, P708, P709	6
PORT8	P800 to P815	16	P800 to P804, P808, P809, P813 to P815	10	P800 to P804, P808, P809, P813 to P815	10	P800 to P806, P808, P809, P814, P815	11
PORT9	P900 to P915	16	P902, P905 to P909	6	P905 to P909	5	P905 to P909	5
PORTA	PA00 to PA15	16	PA00 to PA15	16	PA00 to PA15	16	PA00, PA02 to PA05, PA08 to PA14	12
PORTB	PB00 to PB07	8	PB00 to PB04	5	PB00 to PB04	5	—	0
PORTC	PC00 to PC15*1	16	PC11 to PC15	5	PC11 to PC15	5	PC11, PC13, PC14	3
PORTD	PD00 to PD07	8	PD00	1	PD00	1	—	0

Note 1. For BGA303 package, P603 to P607 and PC00 to PC07 are connected to the SiP Flash memory inside the package.

Table 20.2 I/O port functions (1 of 2)

Port	Port name	Input pull-up	Open-drain output	Drive capability switching	5Vtolerant	I/O	Power supply
PORT0	P000 to P015	✓	✓	Low	—	Input / Output	AVCC0
PORT1	P100 to P107	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC2
	P108 to P115	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC
PORT2	P200	✓	—	—	—	Input	VCC
	P201	✓	✓	Low	—	Input / Output	VCC
	P202, P203	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC2
	P204, P205	✓	✓	Low, middle, high, high speed high drive	✓	Input / Output	VCC2
	P206 to P213	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC
	P214, P215	✓	—	—	—	Input	VBATT_R
PORT3	P300 to P302, P304 to P312	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC
	P303	✓	✓	Low, middle, high, high speed high drive	✓	Input / Output	VCC
	P313 to P315	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC2

Table 20.2 I/O port functions (2 of 2)

Port	Port name	Input pull-up	Open-drain output	Drive capability switching	5Vtolerant	I/O	Power supply
PORT4	P400 to P406, P414, P415	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC
	P407 to P413	✓	✓	Low, middle, high, high speed high drive	✓	Input / Output	VCC
PORT5	P500 to P502	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC2
	P503 to P510, P513	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC
	P511, P512, P514, P515	✓	✓	Low, middle, high, high speed high drive	✓	Input / Output	VCC
PORT6	P600 to P607	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC2
	P608 to P615	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC
PORT7	P700 to P707	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC
	P708 to P715	✓	✓	Low, middle, high, high speed high drive	✓	Input / Output	VCC
PORT8	P800 to P804, P808 to P812	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC2
	P805 to P807, P813	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC
	P814, P815	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC_USB
PORT9	P900, P901	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC2
	P902 to P915	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC
PORTA	PA00 to PA15	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC
PORTB	PB00 to PB07	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC
PORTC	PC00 to PC10	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC2
	PC11 to PC15	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC
PORTD	PD00 to PD07	✓	✓	Low, middle, high, high speed high drive	—	Input / Output	VCC

Note: ✓: Available
—: Setting prohibited

20.2 Register Descriptions

20.2.1 PCNTR1/PODR/PDR : Port Control Register 1

Base address: PORTm = 0x4040_0000 + 0x0020 × m (m = 0 to 9, A, B, C, D)
PORTm_NS = 0x5040_0000 + 0x0020 × m (m = 0 to 9, A, B, C, D)

Offset address: 0x000 (PCNTR1/PDR)
0x002 (PODR)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PODR15	PODR14	PODR13	PODR12	PODR11	PODR10	PODR09	PODR08	PODR07	PODR06	PODR05	PODR04	PODR03	PODR02	PODR01	PODR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PDR15	PDR14	PDR13	PDR12	PDR11	PDR10	PDR09	PDR08	PDR07	PDR06	PDR05	PDR04	PDR03	PDR02	PDR01	PDR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	PDR15 to PDR00	Pmn Direction 0: Input (functions as an input pin) 1: Output (functions as an output pin)	R/W
31:16	PODR15 to PODR00	Pmn Output Data 0: Low output 1: High output	R/W

Note: S-TYPE-4, P-TYPE-5
Note: m = 0 to 9, A, B, C, D, n = 00 to 15

The Port Control Register 1 (PCNTR1/PODR/PDR) is a 32-bit or 16-bit read/write register that controls port direction and port output data. The PCNTR1 specifies the port direction and output data, and is accessed in 32-bit units. The PDRn (bits [15:0] in PCNTR1) and PODRn (bits [31:16] in PCNTR1) respectively, are accessed in 16-bit units.

PDRn bits (Pmn Direction)

The PDRn bits select the input or output direction for individual pins on the associated port when the pins are configured as general I/O pins. Each pin on port m is associated with a PORTm.PCNTR1.PDRn bit. The I/O direction can be specified in 1-bit unit. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0. In the case of input only ports, PDRn bits are reserved. See Table 20.2. The PDRn bit in the PORTm.PCNTR1 register serves the same function as the PDR bit in the PFS.PmnPFS register.

For BGA169 package, set the direction of reserved pins to output. See section 20.5.9. Handling of Reserved Pins for BGA169 Package for details.

PODRn bits (Pmn Output Data)

The PODRn bits hold data to be output from the general I/O pins. Bits of non-existent port m are reserved. Reserved bits are read as 0. The write value should be 0. In the case of input only ports, PODRn bits are reserved. See Table 20.2. The PODRn bit in the PORTm.PCNTR1 register serves the same function as the PODR bit in the PFS.PmnPFS register.

20.2.2 PCNTR2/EIDR/PIDR : Port Control Register 2

Base address: PORTm = 0x4040_0000 + 0x0020 × m (m = 0 to 9, A, B, C, D)
PORTm_NS = 0x5040_0000 + 0x0020 × m (m = 0 to 9, A, B, C, D)

Offset address: 0x004 (PCNTR2/PIDR)
0x006 (EIDR)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	EIDR15	EIDR14	EIDR13	EIDR12	EIDR11	EIDR10	EIDR09	EIDR08	EIDR07	EIDR06	EIDR05	EIDR04	EIDR03	EIDR02	EIDR01	EIDR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PIDR15	PIDR14	PIDR13	PIDR12	PIDR11	PIDR10	PIDR09	PIDR08	PIDR07	PIDR06	PIDR05	PIDR04	PIDR03	PIDR02	PIDR01	PIDR00
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
15:0	PIDR15 to PIDR00	Pmn State 0: Low level 1: High level	R
31:16	EIDR15 to EIDR00 [*2]	Port Event Input Data[*1] When an ELC_PORTx signal occurs 0: Low input 1: High input	R

Note: S-TYPE-4, P-TYPE-5
Note: m = 0 to 9, A, B, C, D, n = 00 to 15
Note 1. x = 1, 2, 3 or 4 for EIDR only
Note 2. Supported by ports 1, 2, 3 or 4.

The Port Control Register 2 (PCNTR2/EIDR/PIDR) allows read access to the Pmn state and the port event input data using 32-bit or 16-bit access.

The PCNTR2 specifies the Pmn state and the port event input data, and is accessed in 32-bit units.

The PIDRn (bits [15:0] in PCNTR2) and EIDRn (bits [31:16] in PCNTR2) respectively, are accessed in 16-bit units. Bits associated with non-existent pins are reserved. Reserved bits are read as undefined.

PIDRn bits (Pmn State)

The PIDRn bits reflect the individual pin states of the port, regardless of the values set in PmnPFS.PMR and PORTm.PCNTR1.PDRn. The PIDRn bit in the PORTm.PCNTR2 register serves the same function as the PIDR bit in the PFS.PmnPFS register.

A pin state cannot be reflected in PIDRn when one of the following functions is enabled:

- RTC Time Capture input and tamper detection (RTCIC)
- Analog function (ASEL = 1)
- I3C bus interface (I3C_SCL0, I3C_SDA0)
- JTAG/SWD interface (TCK/SWCLK, TMS/SWDIO, TDI, TDO)
- Trace interface (TCLK, TDATAn (n = 0 to 3), SWO)

EIDRn bits (Port Event Input Data)

The EIDRn bits latch a pin state when an ELC_PORTx signal occurs. Pin states can only be input to EIDRn when PmnPFS.PMR and PORTm.PCNTR1.PDRn are 0. When the PmnPFS.ASEL bit is set to 1, the associated pin state is not reflected in EIDRn.

20.2.3 PCNTR3/PORR/POSR : Port Control Register 3

Base address: PORTm = 0x4040_0000 + 0x0020 × m (m = 0 to 9, A, B, C, D)
PORTm_NS = 0x5040_0000 + 0x0020 × m (m = 0 to 9, A, B, C, D)

Offset address: 0x008 (PCNTR3/POSR)
0x00A (PORR)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PORR 15	PORR 14	PORR 13	PORR 12	PORR 11	PORR 10	PORR 09	PORR 08	PORR 07	PORR 06	PORR 05	PORR 04	PORR 03	PORR 02	PORR 01	PORR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	POSR 15	POSR 14	POSR 13	POSR 12	POSR 11	POSR 10	POSR 09	POSR 08	POSR 07	POSR 06	POSR 05	POSR 04	POSR 03	POSR 02	POSR 01	POSR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	POSR15 to POSR00	Pmn Output Set 0: No effect on output 1: High output	W
31:16	PORR15 to PORR00	Pmn Output Reset 0: No effect on output 1: Low output	W

Note: S-TYPE-4, P-TYPE-5
Note: m = 0 to 9, A, B, C, D, n = 00 to 15

The Port Control Register 3 (PCNTR3/PORR/POSR) is a 32-bit or 16-bit write register that controls the setting or resetting of the port output data.

The PCNTR3 controls the setting or resetting of the port output data, and is accessed in 32-bit units.

The POSRn (bits [15:0] in PCNTR3) and the PORRn (bits [31:16] in PCNTR3) respectively, are accessed in 16-bit units.

POSRn bits (Pmn Output Set)

POSR changes PODR when set by a software write. For example, for P100, when PORT1.PCNTR3.POSR00 = 1, PORT1.PCNTR1.PODR00 outputs 1. Bits associated with non-existent pins are reserved. The write value should always be 0. In the case of input only ports, POSRn bits are reserved. See Table 20.2.

PORRn bits (Pmn Output Reset)

PORR changes PODR when reset by a software write. For example, for P100, when PORT1.PCNTR3.PORR00 = 1, PORT1.PCNTR1.PODR00 outputs 0. Bits associated with non-existent pins are reserved. The write value should always be 0. In the case of input only ports, PORRn bits are reserved. See Table 20.2.

Note: When EORRn or EOSRn is set, writing is prohibited to PODRn, PORRn, and POSRn.
Note: PORRn and POSRn should not be set at the same time.

20.2.4 PCNTR4/EORR/EOSR : Port Control Register 4

Base address: PORTm = 0x4040_0000 + 0x0020 × m (m = 1 to 4)
PORTm_NS = 0x5040_0000 + 0x0020 × m (m = 1 to 4)

Offset address: 0x00C (PCNTR4/EOSR)
0x00E (EORR)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	EORR 15	EORR 14	EORR 13	EORR 12	EORR 11	EORR 10	EORR 09	EORR 08	EORR 07	EORR 06	EORR 05	EORR 04	EORR 03	EORR 02	EORR 01	EORR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EOSR 15	EOSR 14	EOSR 13	EOSR 12	EOSR 11	EOSR 10	EOSR 09	EOSR 08	EOSR 07	EOSR 06	EOSR 05	EOSR 04	EOSR 03	EOSR 02	EOSR 01	EOSR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	EOSR15 to EOSR00	Pmn Event Output Set When an ELC_PORTx signal occurs 0: No effect on output 1: High output	R/W
31:16	EORR15 to EORR0	Pmn Event Output Reset When an ELC_PORTx signal occurs 0: No effect on output 1: Low output	R/W

Note: S-TYPE-4, P-TYPE-5
Note: m = 1 to 4, n = 00 to 15, x = 1 to 4

The Port Control Register 4 (PCNTR4/EORR/EOSR) is a 32-bit or 16-bit read/write register that controls the setting or resetting of the port output data by an event input from the ELC.

The PCNTR4 controls the setting or resetting of the port output data by an event input from the ELC, and is accessed in 32-bit units.

The EOSRn (bits [15:0] in PCNTR4) and EORRn (bits [31:16] in PCNTR4) respectively, are accessed in 16-bit units.

EOSRn bits (Pmn Event Output Set)

EOSR changes PODR when set because an ELC_PORTx signal occurs. For example, for P100 if PORT1.PCNTR4.EOSR00 is set to 1 when the ELC_PORTx occurs, PORT1.PCNTR1.PODR00 outputs 1. Bits associated with non-existent pins are reserved. The write value should always be 0. For input only ports, EOSRn bits are reserved. See Table 20.2.

EORRn bits (Pmn Event Output Reset)

EORR changes PODR when reset because an ELC_PORTx signal occurs. For example, for P100 if PORT1.PCNTR4.EORR00 = 1 when the ELC_PORTx occurs, PORT1.PCNTR1.PODR00 outputs 0. Bits associated with non-existent pins are reserved. The write value should always be 0. For input only ports, EORRn bits are reserved. See Table 20.2.

Note: When EORRn or EOSRn is set, writing is prohibited to PODRn, PORRn, and POSRn.

Note: EORRn and EOSRn should not be set at the same time.

20.2.5 PmnPFS/PmnPFS_HA/PmnPFS_BY : Port mn Pin Function Select Register (m = 0 to 9, A, B, C, D, n = 00 to 15)

Base address: PFS = 0x4040_0800
PFS_NS = 0x5040_0800

Offset address: 0x000 + 0x040 × m + 0x004 × n (PmnPFS/PmnPFS_HA/PmnPFS_BY)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	PSEL[4:0]					—	—	—	—	—	—	—	PMR
Value after reset:	0	0	0	0*1	0	0	0*1	0*1	0	0	0	0	0	0	0	0*1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ASEL	ISEL	EOFR[1:0]		DSCR[1:0]		—	—	—	NCODR	—	PCR	—	PDR	PIDR	PODR
Value after reset:	0	0	0	0	0	0*1	0	0	0	0	0	0*1	0	0	x	0

Bit	Symbol	Function	R/W
0	PODR	Port Output Data 0: Low output 1: High output	R/W
1	PIDR	Pmn State 0: Low level 1: High level	R
2	PDR	Port Direction 0: Input (functions as an input pin) 1: Output (functions as an output pin)	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	PCR	Pull-up Control 0: Disable input pull-up 1: Enable input pull-up	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	NCODR	N-Channel Open-Drain Control 0: CMOS output 1: NMOS open-drain output	R/W
9:7	—	These bits are read as 0. The write value should be 0.	R/W
11:10	DSCR[1:0]	Port Drive Capability 0 0: Low drive 0 1: Middle drive 1 0: High speed high drive 1 1: High drive	R/W
13:12	EOFR[1:0]	Event on Falling/Event on Rising*2 0 0: Don't care 0 1: Detect rising edge 1 0: Detect falling edge 1 1: Detect both edges	R/W
14	ISEL	IRQ Input Enable 0: Not used as an IRQn input pin 1: Used as an IRQn input pin	R/W
15	ASEL	Analog Input Enable 0: Not used as an analog pin 1: Used as an analog pin	R/W
16	PMR	Port Mode Control 0: Used as a general I/O pin 1: Used as an I/O port for peripheral functions	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
28:24	PSEL[4:0]	Peripheral Select These bits select the peripheral function. For individual pin functions, see the associated tables in this chapter.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note 1. The initial value of P201, P208, P209. P210, P211, P814 and P815 is not 0x0000_0000. P201 is 0x0000_0010, P208 is 0x0001_0010, P209 is 0x0001_0400, P210 is 0x0001_0410, P211 is 0x0001_0010, P814 is 0x1301_0000 and P815 is 0x1301_0000.

Note 2. Supported by PORTn (n = 1 to 4).

Port mn Pin Function Select Register (PmnPFS/PmnPFS_HA/PmnPFS_BY) is a 32-bit, 16-bit, or 8-bit read/write control register that selects the port mn pin function, and is accessed in 32-bit units. PmnPFS_HA (PmnPFS [15:0] bits) is accessed in 16-bit units. PmnPFS_BY (PmnPFS[7:0] bits) is accessed in 8-bit units.

The available Port mn pin depends on the product. For details, see Table 20.1

PODR bit (Port Output Data), PIDR bit (Port State), PDR bit (Port Direction)

The PDR, PIDR, and PODR bits serve the same function as the PCNTR. When these bits are read, the PCNTR value is read.

PCR bit (Pull-up Control)

The PCR bit enables or disables an input pull-up resistor on the individual port pins. When a pin is in the input state with the associated bit in PmnPFS.PCR set to 1, the pull-up resistor connected to the pin is enabled. When a pin is set as a general port output pin, or a peripheral function output pin, the pull-up resistor for the pin is disabled regardless of the PCR setting. The pull-up resistor is also disabled in the reset state. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

NCODR bit (N-Channel Open-Drain Control)

The NCODR bit specifies the output type for the port pins. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

DSCR[1:0] bits (Port Drive Capability)

The DSCR[1:0] bits switch the drive capacity of the port. If the drive capacity of a pin is fixed, the associated bit is a read/write bit, but the drive capacity cannot be changed. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

EOFR[1:0] bits (Event on Falling/Event on Rising)

The EOFR[1:0] bits select the edge detection method for the port group input signal. These bits support rising, falling, or both edge detections. When the EOFR[1:0] bits are set to 01b, 10b, or 11b, the input enable of the I/O cell is asserted. Following that, the event pulse is input from the external pin, and the GPIO outputs the event pulse to the ELC. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

ISEL bit (IRQ Input Enable)

The ISEL bit specifies IRQ input pins. This setting can be used in combination with the peripheral functions, although an IRQn (external pin interrupt) of the same number must only be enabled for one pin. The ISEL bit for an unspecified IRQn is reserved.

ASEL bit (Analog Input Enable)

The ASEL bit specifies analog pins. When a pin is set as an analog pin by this bit:

1. Specify it as a general I/O port in the Port Mode Control bit (PmnPFS.PMR).
2. Disable the pull-up resistor in the Pull-up Control bit (PmnPFS.PCR).
3. Specify the input in the Port Direction bit (PmnPFS.PDR). The pin state cannot be read at this point. The PmnPFS register is protected by the Write-Protect Register for Non-secure (PWPR_NS). Release write-protect before modifying the register.

The ASEL bit for an unspecified analog I/O pin is reserved.

PMR bit (Port Mode Control)

The PMR bit specifies the port pin function. Bits associated with non-existent pins are reserved. The write value should be 0.

PSEL[4:0] bits (Peripheral Select)

The PSEL[4:0] bits assign the peripheral function.

20.2.6 PWPR_NS : Write-Protect Register for Non-secure

Base address: PFS_NS = 0x5040_0800

Offset address: 0x50C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	B0WI	PFSWE	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	PFSWE	PmnPFS Register Write Enable 0: Writing to the PmnPFS register is disabled 1: Writing to the PmnPFS register is enabled	R/W
7	B0WI	PFSWE Bit Write Disable 0: Writing to the PFSWE bit is enabled 1: Writing to the PFSWE bit is disabled	R/W

Note: S-TYPE-7, P-TYPE-2
Note: The access to this register is not controlled by any security attribution register.

PFSWE bit (PmnPFS Register Write Enable)

Writing to the PmnPFS register is enabled only when the PFSWE bit is set to 1. You must first write 0 to the B0WI bit before setting PFSWE to 1.

B0WI bit (PFSWE Bit Write Disable)

Writing to the PFSWE bit is enabled only when the B0WI bit is set to 0.

20.2.7 PFENET : Ethernet Control Register

Base address: PFS = 0x4040_0800
PFS_NS = 0x5040_0800

Offset address: 0x500

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	ETHIOMD1	ETHIOMD0	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	ETHIOMD0	Ethernet I/O Mode Setting ch0 0: GMII mode (ESWM channel 0) 1: MII mode (ESWM channel 0)	R/W
5	ETHIOMD1	Ethernet I/O Mode Setting ch1 0: GMII mode (ESWM channel 1) 1: MII mode (ESWM channel 1)	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: The access to the ETHIOMD0/1 bit is controlled by the PSARC.PSARC30 bit and PPARC.PPARC30 bit.

ETHIOMD0/ETHIOMD1 bit (Ethernet I/O Mode Setting ch0 and Ethernet I/O Mode Setting ch1)

When the transmit data output ports of ESWM are used as GMII/RGMII ports, the drive capability can be set by the ETHIOMDn bit and DSCR[1:0] bit. The details of the setting are given by the following table:

Interface type	PSEL[4:0]	ETHIOMDn	VCC voltage [V]	DSCR [1:0]	Drive capability
MII only	0x16	1b	3.6-2.7	01b	Middle drive
MII/GMII		0b	3.6-3.0	01b	RGMII/3.3 V
			2.7-2.3	11b	RGMII/2.5 V
RMII	0x17	—	3.6-2.7	11b	High drive
RGMII	0x18		3.6-3.0	01b	RGMII/3.3 V
			2.7-2.3	11b	RGMII/2.5 V

If the settings of the ETHIOMDn bit and DSCR[1:0] bit other than those in the above table are performed, the electrical characteristics cannot be guaranteed.

20.2.8 PWPR_S : Write-Protect Register for Secure

Base address: PFS = 0x4040_0800

Offset address: 0x514

Bit position:	7	6	5	4	3	2	1	0
Bit field:	B0WI	PFSWE	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	PFSWE	PmnPFS Register Write Enable 0: Disable writes to the PmnPFS register 1: Enable writes to the PmnPFS register	R/W
7	B0WI	PFSWE Bit Write Disable 0: Enable writes the PFSWE bit 1: Disable writes to the PFSWE bit	R/W

Note: S-TYPE-6, P-TYPE-2

PFSWE bit (PmnPFS Register Write Enable)

Writing to the PmnPFS register of the IO port pin set as secure by the PmSAR register is enabled only when the PFSWE bit is set to 1. You must first write 0 to the B0WI bit before setting PFSWE to 1.

B0WI bit (PFSWE Bit Write Disable)

Writing to the PFSWE bit is enabled only when the B0WI bit is set to 0.

20.2.9 PmSAR : Port Security Attribution Register (m = 0 to 9, A, B, C, D)

Base address: PFS = 0x4040_0800
PFS_NS = 0x5040_0800

Offset address: 0x530 + 0x004 × m

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

Bit	Symbol	Function	R/W
15:0	PMNSA[15:0]	Pmn Security Attribution Target I/O port pin : Pmn 0: Secure 1: Non-secure	R/W

Note: S-TYPE-1, P-TYPE-1
Note: This register is write-protected by PRCR register.
Note: m = 0 to 9, A, B, C, D, n = 00 to 15

Port Security Attribution Register is a 16-bit register that setting the Security Attribution of each port, the registers are accessed only in 16-bit units.

PMNSA[15:0] bits (Pmn Security Attribution)

The PmnSA bit specifies the Security Attribution of Pmn.

20.2.10 LVOCR : Low Voltage Operation Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xAB0

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	LVO1E	LVO0E
Value after reset:	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	LVO0E	Low Voltage Operation 0 Enable 0: Disable 1: Enable	R/W
1	LVO1E	Low Voltage Operation 1 Enable 0: Disable 1: Enable	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Low voltage operation control register controls the IO when VCC/VCC2 is lower than 2.7V.

LVO0E bit (Low Voltage Operation 0 Enable)

When VCC is lower than 2.7V, set the LVO0E bit to 1.

LVO1E bit (Low Voltage Operation 1 Enable)

When VCC2 is lower than 2.7V, set the LVO1E bit to 1.

20.3 Operation

20.3.1 General I/O Ports

All pins except P208 to P211, P814 and P815 operate as general I/O ports after reset. General I/O ports are organized as 16 bits per port and can be accessed by port with the Port Control Registers (PCNTRn, where n = 1 to 4), or by individual pins with the Port mn Pin Function Select register. For details on these registers, see section 20.2. Register Descriptions.

Each port has the following bits:

- Port Security Attribution register (PmSAR) (m = 0 to 9, A, B, C, D), which indicates the security attribution.
- Port Direction bit (PDRn), which selects input or output direction
- Port Output Data bit (PODRn), which holds data for output
- Port Input Data bit (PIDRn), which indicates the pin states
- Event Input Data bit (EIDRn), which indicates the pin state when an ELC_PORTn (n = 1, 2, 3 or 4) signal occurs
- Port Output Set bit (POSRn), which indicates the output value when a software write occurs
- Port Output Reset bit (PORRn), which indicates the output value when a software write occurs
- Event Output Set bit (EOSRn), which indicates the output value when an ELC_PORTn (n = 1, 2, 3 or 4) signal occurs
- Event Output Reset bit (EORRn), which indicates the output value when an ELC_PORTn (n = 1, 2, 3 or 4) signal occurs.

20.3.2 Port Function Select

The following port functions are available for configuring each pin:

- Security function: Security attribution for each pins
- I/O configuration: CMOS output or NMOS open-drain output, pull-up control, and drive capacity
- General I/O port: Port direction, output data setting, and read input data
- Alternate function: Configured function mapping to the pin.

Each pin is associated with a Port mn Pin Function Select register (PmnPFS), which includes the associated PODR, PIDR, and PDR bits. In addition, the PmnPFS register includes the following:

- PCR: Pull-up resistor control bit that turns the input pull-up MOS on or off
- NCODR: N-channel open-drain control bit that selects the output type for each pin
- DSCR[1:0]: Drive capacity control bit that selects the drive capacity
- EOFR[1:0]: For selecting the edge of the event that input from the port group
- ISEL: IRQ input enable bit to specify an IRQ input pin
- ASEL: Analog input enable bit to specify an analog pin
- PMR: Port mode bit to specify the pin function of each port
- PSEL[4:0]: Port function select bits to select the associated peripheral function.

These configurations can be made by a single-register access to the Port mn Pin Function Select register. For details, see section 20.2.5. PmnPFS/PmnPFS_HA/PmnPFS_BY : Port mn Pin Function Select Register (m = 0 to 9, A, B, C, D, n = 00 to 15).

20.3.3 Port Group Function for ELC

In the MCU, PORT1 to PORT4 are assigned for the ELC port group function.

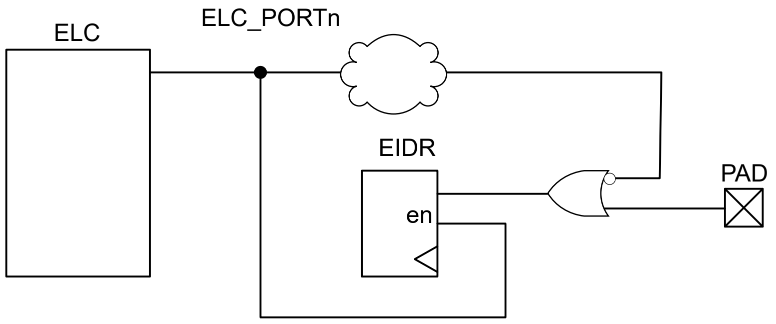
20.3.3.1 Behavior When ELC_PORTn (n = 1, 2, 3 or 4) is Input from ELC

The MCU supports the two functions described in this section when an ELC_PORTn (n = 1, 2, 3 or 4) signal comes from the ELC.

(1) Input to EIDR

For the GPI function (PDR = 0 and PMR = 0 in the PmnPFS register), when an ELC_PORTn (n = 1, 2, 3 or 4) signal comes from the ELC, the input enable of the I/O cell is asserted, and data from the external pins is read into the EIDR bit. See Figure 20.2

For the GPO function (PDR = 1) or the peripheral mode (PMR = 1), 0 is input into the EIDR bit from the external pins.



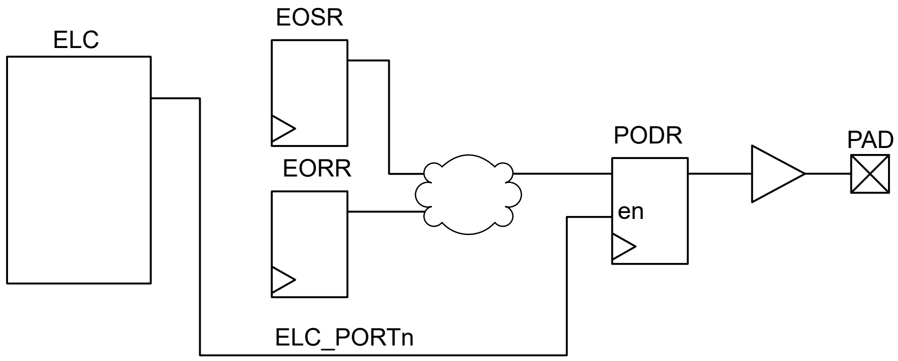
Note: n = 1, 2, 3 or 4

Figure 20.2 Event ports input data

(2) Output from PODR by EOSR and EORR

When an ELC_PORTn (n = 1, 2, 3 or 4) signal occurs, the data is output from the PODR to the external pin based on the settings in the EOSR and EORR registers.

- If EOSR is set to 1, when an ELC_PORTn (n = 1, 2, 3 or 4) signal occurs, the PODR register outputs 1 to the external pin. Otherwise, when EOSR = 0, the PODR value is retained.
- If EORR is set to 1, when ELC_PORTn (n = 1, 2, 3 or 4) signal occurs, the PODR register outputs 0 to the external pin. Otherwise, when EORR = 0, the PODR value is retained.



Note: n = 1, 2, 3 or 4

Figure 20.3 Event ports output data

20.3.3.2 Behavior When an Event Pulse is Output to ELC

To output the event pulse from the external pins to the ELC, set the EOFR[1:0] bits in the PmnPFS register. For details, see section 20.2.5. PmnPFS/PmnPFS_HA/PmnPFS_BY : Port mn Pin Function Select Register (m = 0 to 9, A, B, C, D, n = 00 to 15). When the EOFR[1:0] bits are set, the input enable of the I/O cell is asserted.

Data from the external pin is the input. For example, for Port 1, when the data is input from P100 to P115, the data of those 16 pins is organized by OR logic. This data is formed into a one-shot pulse that goes to the ELC. The operation of Port n (n = 2 to 4) is also the same as Port 1. See Figure 20.4.

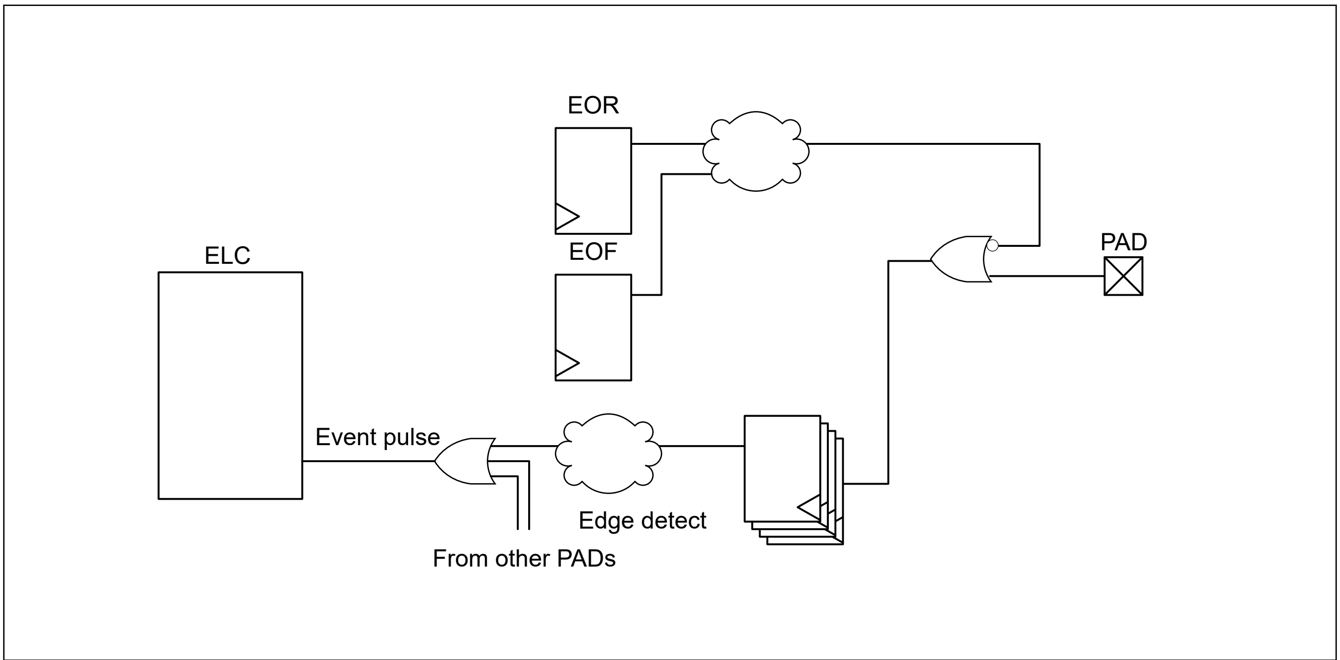


Figure 20.4 Generation of event pulse

20.4 Handling of Unused Pins

Table 20.3 shows how to handle unused pins.

Table 20.3 Handling of unused pins (1 of 2)

Pin name	Description
MD	Use as a mode selection pin
RES	Connect to VCC through a resistor
USB_DP, USB_DM	• Preconditions: – VCC_USB: Connect to VCC – VSS_USB: Connect to VSS – Set the module-stop state for USBFS (MSTPCRB.MSTPB11 = 1) • Processing details: – When both P814PFS.PMR and P815PFS.PMR bits are set to 1, keep these pins open. – When P814PFS.PMR or P815PFS.PMR bit is set to 0, configure it in the same way as P814, P815.
P200/NMI	Connect to VCC through a resistor
EXTAL	When the main clock oscillator is not used, set the MOSCCR.MOSTP bit to 1 (general port P212). When this pin is not used as port P212, configure it in the same way as Other ports.
XTAL	When the main clock oscillator is not used, set the MOSCCR.MOSTP bit to 1 (general port P213). When the external clock is input to the EXTAL pin, the XTAL pin functions as P213. When this pin is not used as port P213, configure it in the same way as Other ports.
XCIN/EXCIN	When the sub clock oscillator is not used, set the SOSCCR.SOSTP bit to 1 and set the SOMCR.SOSEL bit to 0 (general port P215). When this pin is not used as port P215, configure it in the same way as P214, P215.

Table 20.3 Handling of unused pins (2 of 2)

Pin name	Description
XCOUT	When the sub clock oscillator is not used, set the SOSCCR.SOSTP bit to 1 (general port P214). When the external clock is input to the EXCIN pin, the XCOUT pin functions as P214. When this pin is not used as port P214, configure it in the same way as P214, P215.
P000 to P015	• If the direction is set to input (PCNTR1.PDRn = 0), connect the associated pin to AVCC0 through a resistor or to AVSS0 through a resistor *1 • If the direction is set to output (PCNTR1.PDRn = 1), keep pin open *1
P100 to P107, P202 to P205, P313 to P315, P500 to P502, P600 to P607, P800 to P804, P808 to P812, P900, P901, PC00 to PC10	• If the direction is set to input (PCNTR1.PDRn = 0), connect the associated pin to VCC2 through a resistor or to VSS through a resistor *1 • If the direction is set to output (PCNTR1.PDRn = 1), keep pin open *1
P214, P215	Connect to VSS through a resister
P814, P815	• If the direction is set to input (PCNTR1.PDRn = 0), connect the associated pin to VCC_USB through a resistor or to VSS through a resistor *1 • If the direction is set to output (PCNTR1.PDRn = 1), keep pin open *1
Other ports	• If the direction is set to input (PCNTR1.PDRn = 0), connect the associated pin to VCC through a resistor or to VSS through a resistor *1 *2 • If the direction is set to output (PCNTR1.PDRn = 1), keep pin open *1 *3
USBHS_DP, USBHS_DM, USBHS_RREF	• Preconditions: – AVCC_USBHS = VCC_USBHS: Connect to VCC – VSS1_USBHS = VSS2_USBHS: Connect to VSS – Set the module-stop state for USBHS (MSTPCRB.MSTPB12 = 1) • Processing details: – USBHS_DP, USBHS_DM, and USBHS_RREF: Open
VREFH0, VREFH	Connect to AVCC0
VREFL0, VREFL	Connect to AVSS0
VLO	Keep pin open*4
VBATT	Connect to VCC

Note 1. Clear the PmnPFS.PMR, PmnPFS.ISEL, PmnPFS.PCR, and PmnPFS.ASEL bits to 0.
Note 2. P208, P210 and P211 are recommended for pull up VCC through a resistor, because these pins are input pull-up enabled from the initial value (PmnPFS.PCR = 1).
Note 3. P209 is recommended for setting the direction to output (PCNTR1.PDRn = 1), because this pin is output from the initial value.
Note 4. For more information, see section 61, Internal Voltage Regulator

20.5 Usage Notes

20.5.1 Procedure for Specifying the Pin Functions

To specify the I/O pin functions:

1. Clear the B0WI bit in the PWPR_NS register. This enables writing to the PFSWE bit in the PWPR_NS register.*1
2. Set 1 to the PFSWE bit in the PWPR_NS register. This enables writing to the PmnPFS register.*1
3. Clear the Port Mode Control bit in the PMR to 0 for the target pin to select the general I/O port.
4. Specify the I/O function for the pin through the PSEL[4:0] bits settings in the PmnPFS register.
5. Set the PMR bit to 1 as required to switch to the selected I/O function for the pin.
6. Clear the PFSWE bit in the PWPR_NS register. This disables writing to the PmnPFS register.*1
7. Set 1 to the B0WI bit in the PWPR_NS register. This disables writing to the PFSWE bit in the PWPR_NS register.*1

Note 1. When the security attribution of Pmn is set to 0, set the PWPR_S register to write to the PmnPFS register.

20.5.2 Procedure for Using Port Group Input

To use the port group input (port n (n = 1 to 4)):

1. Set the ELSRx.ELS[8:0] bits to all 0 to ignore unexpected pulses. For more information, see section 19, Event Link Controller (ELC).
2. Set the EOFR[1:0] bits of the PmnPFS register to specify the rising, falling, or both edge detections.
3. Execute a dummy read or wait for a short time, for example 100 ns. Ignoring of unexpected pulses depends on the initial value of the external pin.
4. Set the ELSRx.ELS[8:0] bits to enable the event signals.

20.5.3 Port Output Data Register (PODR) Summary

This register outputs data as follows:

1. Outputs 0 if PCNTR4.EORR is set to 1 when ELC_PORTn (n = 1, 2, 3 or 4) signal occurs.
2. Outputs 1 if PCNTR4.EOSR is set to 1 when ELC_PORTn (n = 1, 2, 3 or 4) signal occurs.
3. Outputs 0 if PCNTR3.PORR is set to 1.
4. Outputs 1 if PCNTR3.POSR is set to 1.
5. Outputs 0 or 1 because PCNTR1.PODRn is set.
6. Outputs 0 or 1 because PmnPFS.PODRn is set.

Numbers in this list correspond to the priority for writing to the PODRn. For example, if 1. and 3. from the list occur at the same time, the higher priority event 1. is executed.

20.5.4 Notes on Using Analog Functions

To use an analog function, set the Port Mode Control bit (PMR) and the Port Direction bit (PDRn) to 0 so that the pin acts as a general input port. Next, set the Analog Input Enable bit (ASEL) in the Port mn Pin Function Select Register (PmnPFS.ASEL) to 1.

20.5.5 I/O Buffer Specification

The P402, P403, and P404 can be used as the RTC input, tamper detection and other peripheral functions. Table 20.4 lists the P402, P403, P404 specifications.

Table 20.4 P402, P403, P404 specifications

I/O port	RTC and tamper detection			Other peripheral	
	RTC and tamper detection input enable register	RTC	Tamper detection	Other peripheral enable register	ADC16H, GPT, CAC, ESWM, CANFD, SCI, SSIE, SDHI and interrupt
P402	VBTICTLR.VCH0INEN	RTCIC0	RTCIC0	P402PFS.PSEL and PMR	For details, see section 20.6. Peripheral Select Settings for Each Product.
P403	VBTICTLR.VCH1INEN	RTCIC1	RTCIC1	P403PFS.PSEL and PMR	
P404	VBTICTLR.VCH2INEN	RTCIC2	RTCIC2	P404PFS.PSEL and PMR	

These RTC inputs and tamper detection are controlled by the VBTICTLR register and this register has the highest priority for selecting the functions.

P402, P403, and P404 can be used as IRQn-DS (n = 4, 14, 15) whether RTC inputs and tamper detection are selected or not. When using these interrupts, set the interrupt procedure after setting the VBTICTLR register. See section 12.2.8. VBTICTLR : VBATT Input Control Register.

See Figure 20.5.

The VBTICTLR register is only initialized by VBATT_POR reset. Therefore, when not using the RTC inputs and tamper detection, the associated bit of VBTICTLR register must be set to 0 after except VBATT_POR reset.

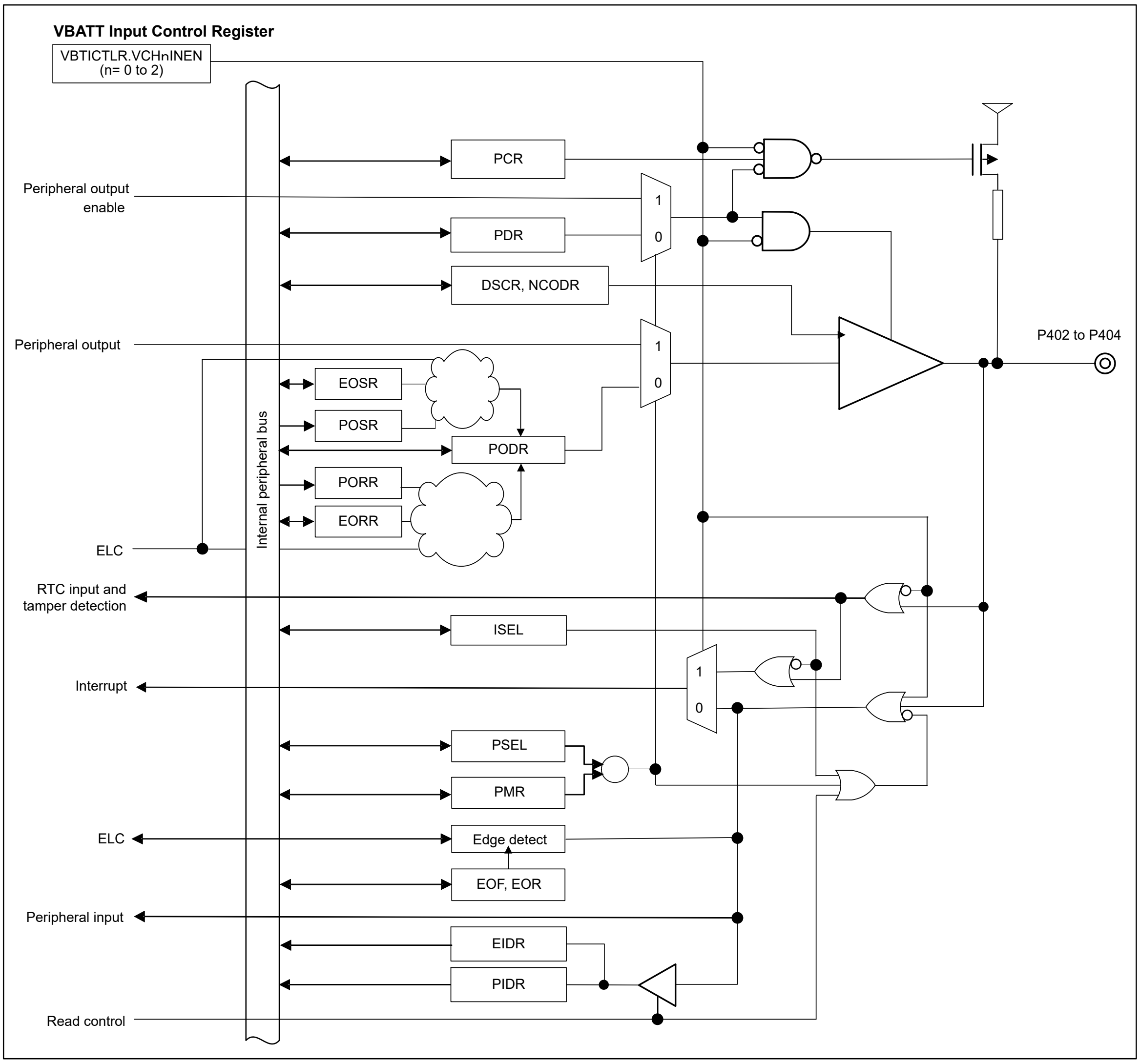


Figure 20.5 P402, P403, P404 diagram

20.5.6 Selecting Pins USB_DP and USB_DM

Pins USB_DP and USB_DM are shared with pins P814 and P815, respectively. Pins USB_DP and P814 can be set with the PFS.P814PFS.PMR bit, and pins USB_DM and P815 can be set with the PFS.P815PFS.PMR bit. Table 20.5 shows setting values of bits PFS.P814PFS.PMR and PFS.P815PFS.PMR with each selected pin.

Table 20.5 Selecting the USB/PORT pins

PMR Bits Settings		Pins Selected	
P814PFS.PMR bit	P815PFS.PMR bit	P814/USB_DP Pin	P815/USB_DM Pin
0	0	P814	P815
0	1	P814	P815
1	0	P814	P815
1	1	USB_DP	USB_DM

Note: When using P814/USB_DP and P815/USB_DM as GPIO pins (P814 and P815), use the USB related registers with their initial values.

Note: When using P814/USB_DP and P815/USB_DM as USB pins (USB_DP and USB_DM), use the GPIO related registers for P814 and P815 with their initial values.
Note: When using P814/USB_DP and P815/USB_DM as GPIO pins or USB pins, set these pins only once after a reset.

20.5.7 Pull-up/Pull-down Setting for P814 and P815 at USBFS/GPIO Function

When the GPIO function of P814 and P815 is used, the Pull-up/Pull-down function of USBFS register affects to P814 and P815. Therefore, before using GPIO function, the pull-up and pull-down control of USBFS needs to be disabled. The control register is SYSCFG.DPRPU and SYSCFG.DRPD bits.

20.5.8 Notes on using P215 and P214

Table 20.6 shows the pin selection and notes for P215 and P214.

Table 20.6 Pin selection and notes for P215 and P214

No.	P215	P214	Description
1	XCIN	XCOUT	Permitted*3
2	EXCIN*1	General purpose input*1	Permitted*4
3	EXCIN*1	Unused*2	Permitted*4
4	General purpose input*1	General purpose input*1	Permitted*5
5	General purpose input*1	Unused*2	Permitted*5
6	Unused*2	General purpose input*1	Permitted*6
7	Unused*2	Unused*2	Permitted*6

Note 1. Do not input a voltage higher than VCC to this pin when VCC > $V_{DETBATT}$.
Do not input a voltage higher than V_{BATT} to this pin when VCC ≤ $V_{DETBATT}$.
Note 2. Unused pin should be connected to VSS through a resister.
Note 3. Set the SOMCR.SOSEL bit to 0 and set SOSCCR.SOSTP bit to 0
Note 4. Set the SOMCR.SOSEL bit to 1 and set SOSCCR.SOSTP bit to 0
Note 5. Set the SOMCR.SOSEL bit to 0 and set SOSCCR.SOSTP bit to 1
Note 6. Set the SOSCCR.SOSTP bit to 1.

20.5.9 Handling of Reserved Pins for BGA169 Package

Table 20.7 shows how to handle reserved pins for BGA169 package.

Table 20.7 Handling of reserved pins for BGA169 package

Pin name	Description
P010, P011, P012, P013, P105, P108, P109, P110, P111, P202, P203, P204, P205, P207, P303, P304, P305, P308, P309, P313, P314, P315, P406, P411, P412, P413, P503, P504, P505, P506, P507, P508, P600, P601, P604, P605, P606, P607, P608, P611, P704, P705, P706, P707, P710, P711, P712, P713, P714, P715, P807, P810, P811, P812, P813, P900, P901, P902, P903, P904, P910, P911, P912, P913, P914, P915, PA01, PA06, PA07, PA15, PB00, PB01, PB02, PB03, PB04, PB05, PB06, PB07, PB08, PB09, PB10, PB11, PB12, PB13, PB14, PC00, PC01, PC02, PC03, PC04, PC05, PC06, PC07, PC08, PC09, PC10, PC12, PC15, PD00, PD01, PD02, PD03, PD04, PD05, PD06, PD07	Set direction to output (PCNTR1.PDRn = 1)

20.6 Peripheral Select Settings for Each Product

This section describes the pin function select configuration using the PmnPFS register. Some pin names have added _A, _B, or _C suffixes. When assigning IIC, SPI, SSIE, and SDHI functionality, select the functional pins having the same suffix. The other pins can be selected regardless of the suffix. Assigning the same function to two or more pins simultaneously is prohibited.

1. In Pmn pin function select register(PmnPFS), the PSEL bits have to be set when the PMR bit of the target pin is 0. If the PSEL bits are set when the PMR bit is 1, the unexpected edges may be input at the input function or the unexpected pulses may be output to the external pin at the output function.
2. Only the allowed values (functions) should be specified in the PSEL bits of PmnPFS register. If a value which is not allowed for the register is specified, the correct operation is not guaranteed.
3. The single function should not be assigned to the multiple pins by PmnPFS register. When the GPT1, GPT5, SCI3, IIC0 or SPI0 are configured as secure and these pin function is being assigned to the pin which security attribution is set as secure by the PmSAR register, the write access to the PSEL bits for setting same function as secure pin in other pins is ignored when the security attribution of that pin is non-secure. For example, if the PSARE.PSARE30 bit is 0 (GPT1 is secure) and the P209PFS.PSEL bits is 00011b (pin function is GTIOC1A) and the P2SAR.P209SA bit is 0 (P209 is secure), the write 00011b to the P405PFS.PSEL bits is ignored when the P4SAR.P405SA bit is 1 (P405 is non-secure).
4. The PORT0, PORT5 and PORT8 have the analog functions such as A/D converter. When these pins are used as an analog function, for avoiding the loss of resolution, the PMR bit should be set to 0 and PDR bit should be set to 0. After that, ASEL bit should be set to 1

Table 20.8 Register settings for input/output pin function (PORT0)

PSEL [4:0]Settings	Function	Pin															
		P000	P001	P002	P003	P004	P005	P006	P007	P008	P009	P010	P011	P012	P013	P014	P015
00000b (value after reset)	Hi-z/ JTAG/SWD	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
ASEL bit		AN000/ IVCMP2	AN001/ IVCMP3	AN002/ IVCMP2	AN003/ IVCMP3	AN004/ IVCMP2	AN005/ IVCMP3	AN006/ IVCMP2	AN007/ IVCMP3	AN008/ IVREF0	AN009/ IVREF1	AN010	AN011	AN012	AN013	AN014/ DA0/ IVCMP0	AN015/ DA1/ IVCMP0
ISEL		IRQ6-DS	IRQ7-DS	IRQ8-DS	IRQ29	IRQ9-DS	IRQ10-DS	IRQ11-DS	IRQ28	IRQ12-DS	IRQ13-DS	IRQ14	IRQ16	IRQ15	IRQ14	IRQ27	IRQ13
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NCODR bit	N-ch Open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DSCR[1:0] bit	Drive capability control*1	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
289, 303 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—	—	✓	✓
176 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—	—	✓	✓
169 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—	—	✓	✓

✓: Available
—: Setting prohibited

Note 1. The drive strength of this port cannot be controlled by PmnPFS.DSCR[1:0] bits.

Table 20.9 Register settings for input/output pin function (PORT1) (1 of 2)

PSEL [4:0]Settings	Function	Pin															
		P100	P101	P102	P103	P104	P105	P106	P107	P108	P109	P110	P111	P112	P113	P114	P115
00000b (value after reset)	Hi-z/JTAG/SWD	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
00001b	AGT	AGTIO0	AGTEE0	AGTO0	—	—	—	AGTOB0	AGTOA0	—	—	—	—	—	—	—	—
00010b	GPT*2	GTETRGA	GTETRGB	GTOWLO	GTOWUP	GTETRGB	—	GTOWLO	GTOWUP	—	—	—	—	GTETRGA	GTETRGB	GTETRGC	GTETRGD
00011b	GPT*2	GTIOC8B	GTIOC8A	GTIOC2B	GTIOC2A	GTIOC1B	GTIOC1A	GTIOC8B	GTIOC8A	GTIOC10B	GTIOC10A	GTIOC9B	GTIOC9A	GTIOC3B	GTIOC2A	GTIOC2B	GTIOC5A
00100b	SCI*1	—	—	—	—	—	CTS_RTS8_B/ SS8_B	CTS8_B	CTS4_A	—	—	—	—	TXD0_A/ SDA0_A/ MOSI0_A	RXD0_A/ SCL0_A/ MISO0_A	CTS_RTS0_A/ SS0_A	CTS0_A
00101b	SCI*1	SCK9_A	RXD9_A/ SCL9_A/ MISO9_A	TXD9_A/ SDA9_A/ MOSI9_A	CTS_RTS9_A/ SS9_A	CTS9_A	—	—	—	—	—	—	—	—	—	—	—

Table 20.9 Register settings for input/output pin function (PORT1) (2 of 2)

PSEL [4:0]Settings	Function	Pin P100	P101	P102	P103	P104	P105	P106	P107	P108	P109	P110	P111	P112	P113	P114	P115
00110b	SPI*1	MISOB_A	MOSIB_A	RSPCKB_A	SSLB0_A	SSLB1_A	SSLB2_A	SSLB3_A	—	—	—	—	—	SSLA2_B	SSLA1_B	SSLA0_B	MOSIA_B
01010b	CAC/ADC16H	—	—	ADTRG0	AD1FLAG1	AD0FLAG1	ADSYNC	ADST1	ADST0	—	—	—	—	ADST0	ADST1	ADSYNC	AD0FLAG1
01011b	BUS	—	—	—	—	—	—	—	—	—	—	—	—	D3/DQ3	D4/DQ4	D5/DQ5	D6/DQ6
01101b	SCI	—	—	—	—	—	DE8	—	—	—	—	—	—	—	—	DE0	—
01110b	SCI	DE9	—	—	DE9	—	—	—	—	—	—	—	—	—	—	—	—
10000b	CANFD	—	—	CRX0	CTX0	—	—	—	—	—	—	—	—	—	—	—	—
10010b	SSIE*1	—	—	—	—	—	—	—	—	—	—	—	—	SSIBCK0_B	SSILRCK0/SSIFS0_B	SSIRXD0_B	SSITXD0_B
10101b	SDHI*1	—	—	—	—	—	—	—	—	SD0DAT6_C	SD0DAT5_C	SD0DAT4_C	SD0DAT3_C	SD0DAT4_B	SD0DAT5_B	SD0DAT6_B	SD0DAT7_B
10110b	ESWM (GMII/MII)	GPTP_MATCH1	GPTP_CAPTURE1	—	GPTP_PPS0	GPTP_MATCH0	GPTP_CAPTURE0	ET1_LINKSTA	ET1_INT	—	—	—	—	—	—	—	—
11100b	OSPI	OM_0_SIO0	OM_0_SIO3	OM_0_SIO4	OM_0_SIO2	OM_0_CS1	OM_0_ECSINT1	OM_0_RESET	OM_0_CS0	—	—	—	—	—	—	—	—
11110b	ULPT	—	—	—	—	—	ULPTO1-DS	ULPTE1-DS	—	—	—	—	—	ULPTOB0-DS	ULPTOA0-DS	—	—
ASEL bit		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
ISEL bit		IRQ2	IRQ1	IRQ17	IRQ16	IRQ1	IRQ0	IRQ16	IRQ31	IRQ24	IRQ23	IRQ20	IRQ19	IRQ27	IRQ28	IRQ30-DS	IRQ31-DS
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NCODR bit	N-ch Open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DSCR [1:0] bit	Drive capability control	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH
289, 303 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		✓	✓	✓	✓	✓	✓	✓	✓	—	—	—	—	✓	✓	✓	✓
176 pins product		✓	✓	✓	✓	✓	✓	✓	✓	—	—	—	—	✓	✓	✓	✓
169 pins product		✓	✓	✓	✓	✓	—	✓	✓	—	—	—	—	✓	✓	✓	✓

✓: Available
—: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance “_A”, “_B”, or “_C” to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t_{GTISK}).

Table 20.10 Register settings for input/output pin function (PORT2) (1 of 2)

PSEL [4:0]Settings	Function	Pin P200*4	P201	P202	P203	P204	P205	P206	P207	P208	P209	P210	P211	P212	P213	P214	P215
00000b (value after reset)	Hi-z/JTAG/SWD	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	TDI	TDO	TMS/SWDIO	TCK/SWCLK	Hi-z	Hi-z	Hi-z	Hi-z
00001b	AGT	—	—	—	—	AGTIO1	AGTO1	—	—	—	—	—	—	AGTEE1	—	—	—
00010b	GPT*2	—	—	—	—	GTIW	GTIV	GTIU	—	GTOVLO	GTOVUP	GTOULO	GTOUUP	GTETRGD	GTETRGC	—	—
00011b	GPT*2	—	—	GTIOC5B	GTIOC5A	GTIOC4B	GTIOC4A	GTCPPO0	GTCPPO3	GTIOC1B	GTIOC1A	GTIOC0B	GTIOC0A	GTIOC0B	GTIOC0A	—	—
00100b	SCI*1	—	—	CTS_RTS4_A/SS4_A	RXD4_A/SCL4_A/MISO4_A	SCK4_A	TXD4_A/SDA4_A/MOSI4_A	—	—	—	—	—	—	—	—	—	—
00101b	SCI*1	—	—	—	—	—	—	—	—	RXD9_B/SCL9_B/MISO9_B	TXD9_B/SDA9_B/MOSI9_B	CTS_RTS9_B/SS9_B	SCK9_B	RXD1_C/SCL1_C/MISO1_C	TXD1_C/SDA1_C/MOSI1_C	—	—

Table 20.10 Register settings for input/output pin function (PORT2) (2 of 2)

PSEL [4:0]Settings	Function	P200*4	P201	P202	P203	P204	P205	P206	P207	P208	P209	P210	P211	P212	P213	P214	P215
00110b	SPI*1	—	—	MOSIA_A	RSPCKA_A	SSLA0_A	SSLA1_A	—	—	—	—	—	—	—	—	—	—
00111b	I3C/IIC*1	—	—	—	—	SDA1_B	SCL1_B	—	—	—	—	—	—	—	—	—	—
01001b	RTC/CLKOUT/ACMPHS/ETHPHYCLK	—	—	—	—	—	CLKOUT	CLKOUT	—	VCOUT	CLKOUT	—	—	—	—	—	—
01010b	CAC/ADC16H	—	—	—	—	CACREF	—	—	—	—	—	—	—	—	ADTRG1	—	—
01011b	BUS	—	—	—	—	—	—	CS7	—	—	—	—	—	—	—	—	—
01101b	SCI	—	—	DE4	—	DE4	—	—	—	—	—	—	—	—	—	—	—
01110b	SCI	—	—	—	—	—	—	—	—	—	—	DE9	DE9	—	—	—	—
10000b	CANFD	—	—	CRX0	CTX0	—	—	—	—	CRX1	CTX1	—	—	—	—	—	—
10010b	SSIE*1	—	—	—	—	—	—	SSIDATA1_A	—	—	—	—	—	—	—	—	—
10011b	USBFS	—	—	USB_EXICEN	USB_VBUSEN	USB_OVRCURB	USB_OVRCURA	USB_VBUSEN	—	—	—	—	—	—	—	—	—
10101b	SDHI*1	—	—	SD1CMD_A	SD1CLK_A	SD1WP	SD1CD	SD0DAT7_C	—	—	—	—	—	—	—	—	—
10110b	ESWM (GMII/MII)	—	—	—	—	—	—	ET1_RX_DV	ET1_RXD5	—	—	—	—	—	—	—	—
10111b	ESWM (RMII)	—	—	—	—	—	—	RMII1_CRS_DV	—	—	—	—	—	—	—	—	—
11000b	ESWM (RGMII)	—	—	—	—	—	—	RGMII1_RX_CTL	—	—	—	—	—	—	—	—	—
01111b	CEU	—	—	—	—	—	—	VIO_D0	—	—	—	—	—	—	—	—	—
11110b	ULPT	—	—	ULPTOB1	ULPTOA1	—	—	ULPTOB1	—	—	—	—	—	—	ULPTEE0	—	—
Don't care	Trace (Debug)/Clock	—	—	—	—	—	—	—	—	—	SWO*5	—	—	EXTAL	XTAL	XCOUT	XCIN/EXCIN
ASEL bit		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
ISEL bit		NMI	IRQ4	IRQ3-DS	IRQ2-DS	IRQ26	IRQ1-DS	IRQ0-DS	IRQ25	IRQ3	IRQ25	IRQ24	IRQ23	IRQ3	IRQ2	IRQ21	IRQ20
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NCODR bit	N-ch Open-drain	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
DSCR[1:0] bit	Drive capability control	—	L*3	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	—	—
289, 303 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		✓	✓	✓	✓	✓	✓	✓	—	✓	✓	✓	✓	✓	✓	✓	✓
176 pins product		✓	✓	—	—	—	—	✓	—	✓	✓	✓	✓	✓	✓	✓	✓
169 pins product		✓	✓	—	—	—	—	✓	—	✓	✓	✓	✓	✓	✓	✓	✓

✓: Available
—: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance “_A”, “_B”, or “_C” to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t_{GTISK}).

Note 3. The driver strength of this port is controlled by PmnPFS.DSCR[1:0] bits. Set the PmnPFS.DSCR[1:0] bits to Low Drive (00b).

Note 4. When using NMI pin interrupt, Port related registers setting are not required.

Note 5. To use this pin function, select the CoreSight TPIU.SPPR.PProtocol bits to SWO and set the TRPORTCR.OE bit to 1. The drive strength of this pin function is controlled by TRPORTCR.DRV[1:0] bits.

Table 20.11 Register settings for input/output pin function (PORT3)

PSEL [4:0]Settings	Function	Pin P300	P301	P302	P303	P304	P305	P306	P307	P308	P309	P310	P311	P312	P313	P314	P315
00000b (value after reset)	Hi-z/JTAG/SWD	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
00001b	AGT	—	AGTIO0	—	—	—	—	—	—	—	—	AGTEE1	AGTOB1	AGTOA1	—	—	—
00010b	GPT*2	—	GTOULO	GTOUUP	—	GTOVLO	GTOVUP	GTIW	GTIV	GTIU	—	—	GTADSM1	GTADSM0	—	—	—
00011b	GPT*2	GTIOC3A	GTIOC4B	GTIOC4A	GTIOC7B	GTIOC7A	GTCPPO12	GTCPPO11	GTCPPO10	GTCPPO9	GTCPPO8	GTCPPO7	GTCPPO6	GTCPPO5	—	—	—
00100b	SCI*1	SCK0_A	TXD6_B/SDA6_B/MOSI6_B	RXD6_B/SCL6_B/MISO6_B	SCK6_B	—	—	—	CTS_RTS6_A/SS6_A	—	—	—	—	—	—	—	—
00101b	SCI*1	—	—	—	—	—	—	—	—	CTS3_B	CTS9_B	TXD3_B/SDA3_B/MOSI3_B	SCK3_B	CTS_RTS3_B/SS3_B	TXD3_C/SDA3_C/MOSI3_C	RXD3_C/SCL3_C/MISO3_C	SCK3_C
00110b	SPI*1	SSLA3_B	—	—	—	—	—	—	—	—	—	—	—	—	MISOA_A	SSLA2_A	SSLA3_A
01001b	RTC/CLKOUT/ACMPHS/ETHPHYCLK	—	—	—	—	—	—	—	—	ETHPHYCLK	VCOUT	—	—	—	—	—	—
01010b	CAC/ADC16H	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADTRG0	—
01011b	BUS	D2/DQ2	D1/DQ1	D0/DQ0	—	—	—	—	—	—	—	—	—	—	—	—	—
01101b	SCI	DE0	—	—	DE6	—	—	—	DE6	—	—	—	—	—	—	—	—
01110b	SCI	—	—	—	—	—	—	—	—	—	—	—	DE3	DE3	—	—	DE3
10000b	CANFD	—	—	—	—	—	—	—	—	—	—	—	CRX0	CTX0	—	—	—
10011b	USBFS	—	—	—	—	—	—	—	—	—	—	—	—	—	USB_ID	—	—
10101b	SDHI*1	SD0DAT3_B	SD0DAT2_B	SD0DAT1_B	—	SD0DAT0_B	SD0WP	SD0CD	SD0CMD_B	SD0CLK_B	—	—	—	—	SD1DAT0_A	SD1DAT1_A	—
10110b	ESWM (GMII/MII)	—	—	—	—	ET1_TXD3	ET1_TXD2	ET1_TXD1	ET1_TXD0	ET1_TX_ER	ET1_GTX_CLK	ET1_TX_EN	ET1_TX_CLK	ET1_RX_ER	—	—	—
10111b	ESWM (RMII)	—	—	—	—	—	—	RMII1_TXD1	RMII1_TXD0	—	—	RMII1_TX_EN	—	RMII1_RX_ER	—	—	—
11000b	ESWM (RGMII)	—	—	—	—	RGMII1_TXD3	RGMII1_TXD2	RGMII1_TXD1	RGMII1_TXD0	—	RGMII1_TXC	RGMII1_TX_CTL	—	—	—	—	—
11011b	PDMIF	—	—	—	—	—	—	—	—	—	—	—	—	PDMDAT2	—	—	—
01111b	CEU	—	—	—	—	VIO_D15	VIO_D14	VIO_D13	VIO_D12	VIO_D11	VIO_D10	VIO_D9	VIO_D8	VIO_D7	—	—	—
11110b	ULPT	ULPTEVI0-DS	ULPTEE0-DS	ULPTO0-DS	—	ULPTO1	ULPTEE1	ULPTEVI1	ULPTOA1	ULPTOB1	—	—	—	—	—	—	—
Don't care	Trace (Debug)	—	—	—	—	TDATA3*3	TDATA2*3	TDATA1*3	TDATA0*3	TCLK*3	—	—	—	—	—	—	—
ASEL bit		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
ISEL bit		IRQ4	IRQ6	IRQ5	IRQ29-DS	IRQ9	IRQ8	IRQ28-DS	IRQ27-DS	IRQ26-DS	IRQ25-DS	IRQ24-DS	IRQ23-DS	IRQ22-DS	IRQ27	IRQ28	IRQ29
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NCODR bit	N-ch Open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DSCR[1:0] bit	Drive capability control	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH
289, 303 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		✓	✓	✓	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
176 pins product		✓	✓	✓	—	✓	✓	✓	✓	✓	—	✓	✓	✓	—	—	—
169 pins product		✓	✓	✓	—	—	—	✓	✓	—	—	✓	✓	✓	—	—	—

✓: Available

—: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance "_A", "_B", or "_C" to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t_{GTISK}).

Note 3. To use this pin function, select the CoreSight TPIU.SPPR.PProtocol bits to parallel port and set the TRPORTCR.OE bit to 1. The drive strength of this pin function is controlled by TRPORTCR.DRV[1:0] bits. How many pins of Trace Data are used can be controlled by TRPORTSZ.PORTSIZE[31:0] bits.

Table 20.12 Register settings for input/output pin function (PORT4) (1 of 2)

PSEL [4:0]Settings	Function	Pin															
		P400	P401	P402	P403	P404	P405	P406	P407	P408	P409	P410	P411	P412	P413	P414	P415
00000b (value after reset)	Hi-z/ JTAG/SWD	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
00001b	AGT	AGTIO1	—	—	—	—	AGTIO1	—	AGTIO0	—	—	AGTOB1	AGTOA1	AGTEE1	—	—	—
00010b	GPT*2	—	GTETRGA	—	—	—	—	—	—	GTOWLO	GTOWUP	GTOVLO	GTOVUP	GTOULO	GTOUUP	—	—
00011b	GPT*2	GTIOC6A	GTIOC6B	—	GTIOC3A	GTIOC3B	GTIOC1A	GTIOC1B	GTIOC10B	GTIOC10A	—	GTIOC9B	GTIOC9A	GTCPPO8	GTCPPO7	GTIOC0B	GTIOC0A
00100b	SCI*1	—	—	—	—	—	SCK2_B	TXD2_B/ SDA2_B/ MOSI2_B	—	—	—	—	—	—	—	RXD4_B/ SCL4_B/ MISO4_B	TXD4_B/ SDA4_B/ MOSI4_B
00101b	SCI*1	TXD1_A/ SDA1_A/ MOSI1_A	RXD1_A/ SCL1_A/ MISO1_A	SCK1_A	CTS_RTS1_A/ SS1_A	CTS1_A	—	—	SCK1_C	RXD3_A/ SCL3_A/ MISO3_A	TXD3_A/ SDA3_A/ MOSI3_A	SCK3_A	CTS_RTS3_A/ SS3_A	CTS3_A	—	—	—
00110b	SPI*1	—	—	—	—	—	—	SSLA3_C	—	—	—	—	—	—	—	SSLB0_B	RSPCKB_B
00111b	I3C/IIC*1	I3C_SCL0	I3C_SDA0	—	—	—	—	—	SDA0_B	SCL0_B	SDA0_A	SCL0_A	—	—	—	—	—
01001b	RTC/ CLKOUT/ ACMPHS/ ETHPHYCLK	—	—	—	—	—	—	—	RTCOUT	—	—	—	—	—	—	—	—
01010b	CAC/ ADC16H	ADTRG1	—	CACREF	AD0FLAG1	AD1FLAG1	—	—	ADTRG0	ADSYNC	ADST1	ADST0	CACREF	—	—	—	—
01011b	BUS	—	—	—	—	—	—	—	CS6	A17	A18	A19	A20	A21	A22	A23	WAIT
01101b	SCI	—	—	—	—	—	DE2	—	—	—	—	—	—	—	—	—	—
01110b	SCI	—	—	DE1	DE1	—	—	—	DE1	—	—	DE3	DE3	—	—	—	—
10000b	CANFD	—	CTX0	CRX0	—	—	—	—	—	—	—	—	—	—	—	CRX1	CTX1
10010b	SSIE*1	AUDIO_CLK	—	AUDIO_CLK	SSIBCK0_A	SSILRCK0/ SSIFS0_A	SSITXD0_A	SSIRXD0_A	—	—	—	—	—	—	—	—	—
10011b	USBFS	—	—	—	—	—	—	—	USB_VBUS	USB_VBUSEN	USB_OVRCURA-DS	USB_OVRCURB-DS	USB_ID	USB_EXICEN	—	—	—
10100b	USBHS	—	—	—	—	—	—	—	USBHS_VBUSEN	USBHS_VBUS	USBHS_OVRCURA	USBHS_OVRCURB	USBHS_ID	USBHS_EXICEN	—	—	—
10101b	SDHI*1	SD1CLK_B	SD1CMD_B	SD1DAT0_B	SD1DAT1_B	SD1DAT2_B	SD1DAT3_B	SD1CD	—	—	—	—	—	—	—	—	—
10110b	ESWM (GMII/MII)	—	—	ET0_LINKSTA	ET1_WOL	ET0_WOL	ET0_RX_DV	ET0_RXD3	GPTP_PTPOUT3	GPTP_PTPOUT2	GPTP_CAPTURE0	GPTP_MATCH0	GPTP_PTPOUT1	GPTP_PTPOUT0	ET_TAS_STA3	ET1_MDIO	ET1_MDC
10111b	ESWM (RMII)	—	—	—	—	—	RMII0_CRS_DV	—	—	—	—	—	—	—	—	—	—
11000b	ESWM (RGMII)	—	—	—	—	—	RGMII0_RX_CTL	RGMII0_RXD3	—	—	—	—	—	—	—	—	—
01111b	CEU	VIO_D0	VIO_D1	—	—	—	VIO_D2	VIO_D3	—	—	—	—	—	—	—	VIO_CLK	VIO_HD
11110b	ULPT	—	—	—	—	—	—	—	—	ULPTOB0	ULPTOA0	—	—	—	ULPTEE1	—	—

Table 20.12 Register settings for input/output pin function (PORT4) (2 of 2)

PSEL [4:0]Settings	Function	P400	P401	P402	P403	P404	P405	P406	P407	P408	P409	P410	P411	P412	P413	P414	P415
Don't care	RTC	—	—	RTCIC0*3	RTCIC1*3	RTCIC2*3	—	—	—	—	—	—	—	—	—	—	—
ASEL bit		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
ISEL bit		IRQ0	IRQ5-DS	IRQ4-DS	IRQ14-DS	IRQ15-DS	IRQ30	IRQ31	IRQ22	IRQ7	IRQ6	IRQ5	IRQ4	IRQ20-DS	IRQ18	IRQ9	IRQ8
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NCODR bit	N-ch Open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DSCR[1:0] bit	Drive capability control	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH
289, 303 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—	✓	✓
176 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—	✓	✓
169 pins product		✓	✓	✓	✓	✓	✓	—	✓	✓	✓	✓	—	—	—	✓	✓

✓: Available
—: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance “_A”, “_B”, or “_C” to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t_{GTISK}).

Note 3. To use this pin function, set the associated pin as a general input (set the PmnPFS.PDR and PmnPFS.PMR bits to 0).

Table 20.13 Register settings for input/output pin function (PORT5) (1 of 2)

PSEL [4:0]Settings	Function	P500	P501	P502	P503	P504	P505	P506	P507	P508	P509	P510	P511	P512	P513	P514	P515
00000b (value after reset)	Hi-z/JTAG/SWD	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
00010b	GPT*2	—	—	—	GTOUUP	GTOULO	GTOWUP	GTOWLO	GTADSM0	GTADSM1	—	—	—	—	—	—	—
00011b	GPT*2	GTIOC11B	GTIOC12A	GTIOC12B	GTCPPO6	GTCPPO1	GTCPPO2	GTCPPO3	GTIOC0A	GTIOC0B	GTIOC1A	GTIOC1B	GTIOC0B	GTIOC0A	GTIOC13B	GTIOC13B	GTIOC13A
00100b	SCI*1	RXD8_B/SCL8_B/MISO8_B	TXD8_B/SDA8_B/MOSI8_B	SCK8_B	—	—	—	—	—	—	—	—	CTS_RTS8_A/SS8_A	CTS8_A	SCK8_A	SCK4_C	CTS_RTS4_C/SS4_C
00101b	SCI*1	—	—	—	—	—	—	—	CTS_RTS7_A/SS7_A	CTS5_A	CTS_RTS5_A/SS5_A	RXD5_A/SCL5_A/MISO5_A	—	—	—	—	—
00111b	I3C/IIC*1	—	—	—	—	—	—	—	—	—	—	—	SDA1_A	SCL1_A	—	SDA2_B	SCL2_B
01010b	CAC/ADC16H	CACREF	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
01011b	BUS	—	—	—	A6	A7	A8	A9	A10	A11	A12	A13	—	—	—	—	—
01101b	SCI	—	—	DE8	—	—	—	—	—	—	—	—	DE8	—	DE8	DE4	DE4
01110b	SCI	—	—	—	—	—	—	—	DE7	—	DE5	—	—	—	—	—	—
10000b	CANFD	—	—	—	—	—	—	—	—	—	—	—	CRX1	CTX1	—	—	—
10011b	USBFS	USB_VBUSEN	USB_OVRCURA	USB_OVRCURB	—	—	—	—	—	—	—	—	—	—	—	—	—
10101b	SDHI*1	SD1DAT5_A	SD1DAT6_A	SD1DAT7_A	SD0CD	SD0WP	SD0CLK_A	SD0CMD_A	—	—	—	—	—	—	—	—	—
10110b	ESWM (GMII/MII)	—	—	—	—	—	—	—	ET_TAS_STA0	ET_TAS_STA1	ET_TAS_STA2	ET_TAS_STA3	ET1_LINKSTA	ET1_INT	ET0_INT	ET_TAS_STA1	ET_TAS_STA0
11011b	PDMIF	PDMDAT0	PDMDAT1	PDMDAT2	—	—	—	—	—	—	—	—	—	—	—	—	—
01111b	CEU	—	—	—	VIO_D4	VIO_D3	VIO_D2	VIO_D1	VIO_D0	VIO_VD	VIO_HD	VIO_CLK	—	—	VIO_FLD	—	—

Table 20.13 Register settings for input/output pin function (PORT5) (2 of 2)

PSEL [4:0]Settings	Function	Pin P500	P501	P502	P503	P504	P505	P506	P507	P508	P509	P510	P511	P512	P513	P514	P515
11110b	ULPT	—	—	—	—	—	—	—	—	—	ULPTEVI1	ULPTEVI0	—	—	—	—	—
ASEL bit		AN021	AN020	AN019	—	—	—	—	—	—	—	—	—	—	AN016/IVCMP0	—	—
ISEL bit		IRQ24	IRQ25	IRQ26	IRQ6	IRQ7	IRQ8	IRQ9	IRQ10	IRQ1	IRQ2	IRQ3	IRQ15	IRQ14	IRQ31	IRQ13	IRQ12
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NCODR bit	N-ch Open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DSCR[1:0] bit	Drive capability control	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH
289, 303 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—
176 pins product		—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—
169 pins product		✓	✓	✓	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓

✓: Available
—: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance "_A", "_B", or "_C" to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t_{GTISK}).

Table 20.14 Register settings for input/output pin function (PORT6) (1 of 2)

PSEL [4:0]Settings	Function	Pin P600	P601	P602	P603	P604	P605	P606	P607	P608	P609	P610	P611	P612	P613	P614	P615
00000b (value after reset)	Hi-z/JTAG/SWD	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
00001b	AGT	—	—	—	—	—	—	—	—	—	—	—	—	—	AGTO1	AGTO0	—
00010b	GPT[*2]	—	—	—	—	—	—	—	—	GTOWUP	GTIU	GTOUUP	GTOULO	—	GTETRGA	GTETRGB	GTETRGC
00011b	GPT[*2]	GTIOC6B	GTIOC6A	GTIOC7B	GTIOC7A	GTIOC8B	GTIOC8A	—	—	GTCPPO4	GTIOC5B	GTIOC4A	GTIOC4B	GTIOC9A	GTIOC9B	GTCPPO9	GTCPPO10
00100b	SCI[*1]	—	SCK0_B	RXD0_B/SCL0_B/MISO0_B	TXD0_B/SDA0_B/MOSI0_B	CTS_RTS0_B/SS0_B	CTS0_B	—	—	—	TXD0_C/SDA0_C/MOSI0_C	RXD0_C/SCL0_C/MISO0_C	SCK0_C	CTS_RTS0_C/SS0_C	CTS0_C	—	—
00101b	SCI[*1]	—	—	—	—	—	—	—	—	TXD5_A/SDA5_A/MOSI5_A	—	—	—	—	—	RXD7_A/SCL7_A/MISO7_A	TXD7_A/SDA7_A/MOSI7_A
00110b	SPI[*1]	—	—	—	—	—	—	—	—	—	MISOA_B	RSPCKA_B	MOSIA_B	SSLA0_B	—	—	—
01001b	RTC/CLKOUT/ACMPHS/ETHPHYCLK	—	RTCOUT	—	—	—	—	—	—	—	—	—	CLKOUT	—	—	—	—
01010b	CAC/ADC16H	CACREF	—	—	—	—	—	—	—	CACREF	AD1FLAG1	—	CACREF	—	—	—	—
01011b	BUS	—	—	—	—	—	—	WR3/BC3	D31/DQ31	A14	D7/DQ7	D12/DQ12	D13/DQ13	D14/DQ14	D15/DQ15	WR/WR0/DQM0	WR2/BC2/DQM2
01101b	SCI	—	DE0	—	—	DE0	—	—	—	—	—	—	DE0	DE0	—	—	—
10000b	CANFD	—	—	—	—	—	—	—	—	—	CTX1	CRX1	—	—	—	—	—
10100b	USBHS	—	—	—	—	—	—	—	—	—	—	—	USBHS_VBUSEN	USBHS_OVRCURA	USBHS_OVRCURB	USBHS_ID	USBHS_EXICEN
10110b	ESWM (GMII/MII)	ET1_WOL	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
11100b	OSPI	OM_0_RSTO1	OM_0_WP1	OM_1_SCLKN	OM_1_SCLK	OM_1_SIO7	OM_1_SIO1	OM_1_SIO6	OM_1_DQS	—	—	—	—	—	—	—	—

Table 20.14 Register settings for input/output pin function (PORT6) (2 of 2)

PSEL [4:0]Settings	Function	Pin															
		P600	P601	P602	P603	P604	P605	P606	P607	P608	P609	P610	P611	P612	P613	P614	P615
01111b	CEU	—	—	—	—	—	—	—	—	VIO_FLD	—	—	—	—	—	—	—
11110b	ULPT	ULPTEVI1-DS	ULPTEVI0	ULPTEE0	ULPTO0	—	—	—	—	—	ULPTOA1-DS	ULPTOB1-DS	—	—	—	—	—
ASEL bit		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
ISEL bit		IRQ30	IRQ29	IRQ28	IRQ27	IRQ26	IRQ25	IRQ24	IRQ23	IRQ22	IRQ29	IRQ16	IRQ17	IRQ18	IRQ19	IRQ20	IRQ7
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NCODR bit	N-ch Open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DSCR[1:0] bit	Drive capability control	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH
289, 303 pins product*3		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		✓	✓	✓	✓	✓	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓
176 pins product		✓	✓	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓
169 pins product		—	—	✓	✓	—	—	—	—	—	✓	✓	—	✓	✓	✓	✓

✓: Available
—: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance "_A", "_B", or "_C" to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t_{GTISK}).

Note 3. For BGA303 package, P603 to P607 are connected to the SiP flash inside the package. To access SiP Flash memory, set OSPI in PSEL settings.

Table 20.15 Register settings for input/output pin function (PORT7) (1 of 2)

PSEL [4:0]Settings	Function	Pin															
		P700	P701	P702	P703	P704	P705	P706	P707	P708	P709	P710	P711	P712	P713	P714	P715
00000b (value after reset)	Hi-z/ JTAG/SWD	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
00001b	AGT	—	—	—	AGTO1	AGTO0	AGTIO0	AGTIO0	—	—	—	—	AGTEE0	AGTOB0	AGTOA0	—	—
00010b	GPT*2	—	—	—	—	GTADSM0	GTADSM1	—	—	—	—	—	—	—	—	—	—
00011b	GPT*2	GTIOC5A	GTIOC5B	GTIOC6A	GTIOC6B	GTCPPO0	GTCPPO1	GTCPPO2	GTCPPO3	GTCPPO6	GTCPPO5	GTIOC11B	GTIOC11A	GTIOC2B	GTIOC2A	GTIOC12B	GTIOC12A
00100b	SCI*1	RXD2_B/ SCL2_B/ MISO2_B	CTS_RTS2_B/ SS2_B	CTS2_B	—	—	—	—	—	SCK4_B	CTS_RTS4_B/ SS4_B	CTS4_B	—	—	CTS4_C	TXD4_C/ SDA4_C/ MOSI4_C	RXD4_C/ SCL4_C/ MISO4_C
00101b	SCI*1	—	—	—	—	—	CTS1_B	RXD1_B/ SCL1_B/ MISO1_B	TXD1_B/ SDA1_B/ MOSI1_B	—	—	—	CTS_RTS1_C/ SS1_C	CTS1_C	—	—	—
00110b	SPI*1	MISOA_C	MOSIA_C	RSPCKA_C	SSLA0_C	SSLA1_C	SSLA2_C	—	—	MOSIB_B	MISOB_B	SSLB3_B	SSLB2_B	SSLB1_B	—	—	—
00111b	I3C/IIC*1	—	—	—	—	—	—	—	—	SDA2_A	SCL2_A	—	—	—	—	—	—
01001b	RTC/ CLKOUT/ ACMPHS/ ETHPHYCLK	—	—	—	VCOUT	—	—	ETHPHYCLK	ETHPHYCLK	—	—	—	—	—	—	—	—
01010b	CAC/ ADC16H	—	—	—	—	—	—	—	—	CACREF	—	—	—	—	—	—	—
01011b	BUS	—	—	—	—	—	—	—	—	WR1/BC1	CS4	CS5	—	—	—	—	—
01101b	SCI	—	DE2	—	—	—	—	—	—	DE4	DE4	—	—	—	—	—	—
01110b	SCI	—	—	—	—	—	—	—	—	—	—	—	DE1	—	—	—	—
10000b	CANFD	—	—	—	—	CTX0	CRX0	—	—	—	—	—	—	—	—	—	—

Table 20.15 Register settings for input/output pin function (PORT7) (2 of 2)

PSEL [4:0]Settings	Function	Pin P700	P701	P702	P703	P704	P705	P706	P707	P708	P709	P710	P711	P712	P713	P714	P715
10010b	SSIE*1	SSIDATA1_B	SSILRCK1/ SSIFS1_B	SSIBCK1_B	—	—	—	—	—	AUDIO_CLK	—	—	—	—	—	—	—
10100b	USBHS	—	—	—	—	—	—	USBHS_OVRCURB-DS	USBHS_OVRCURA-DS	—	—	—	—	—	—	—	—
10101b	SDHI*1	SD1WP	SD1DAT4_B	SD1DAT5_B	SD1DAT6_B	SD1DAT7_B	—	—	—	—	—	—	—	—	—	—	—
10110b	ESWM (GMII/MII)	ET0_RXD2	ET0_RXD1	ET0_RXD0	ET0_RX_CLK	ET0_RX_ER	ET0_TX_EN	ET0_GTX_CLK	ET0_TX_ER	ET0_MDC	ET0_MDIO	ET0_LINKSTA	GPTP_PPS0	GPTP_CAPTURE1	GPTP_MATCH1	GPTP_PPS1	ET_TAS_STA2
10111b	ESWM (RMII)	—	RMII0_RXD1	RMII0_RXD0	RMII0_REF50CK	RMII0_RX_ER	RMII0_TX_EN	—	—	—	—	—	—	—	—	—	—
11000b	ESWM (RGMII)	RGMII0_RXD2	RGMII0_RXD1	RGMII0_RXD0	RGMII0_RXC	—	RGMII0_TX_CTL	RGMII0_TXC	—	—	—	—	—	—	—	—	—
11011b	PDMIF	—	—	—	PDMCLK0	PDMCLK1	PDMCLK2	PDMDAT0	PDMDAT1	—	—	—	—	—	—	—	—
01111b	CEU	VIO_D4	VIO_D5	VIO_D6	VIO_D7	VIO_D8	VIO_D9	VIO_D10	VIO_D11	VIO_VD	VIO_D13	VIO_D12	—	—	—	—	—
11110b	ULPT	—	ULPTO1	ULPTO0	—	—	—	—	—	—	—	—	—	—	—	—	—
ASEL bit		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
ISEL bit		IRQ16-DS	IRQ17-DS	IRQ18-DS	IRQ19-DS	IRQ26	IRQ19	IRQ7	IRQ8	IRQ11	IRQ10	IRQ17	IRQ3	IRQ2	IRQ14	IRQ13	IRQ12
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NCODR bit	N-ch Open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DSCR[1:0] bit	Drive capability control	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH
289, 303 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
176 pins product		✓	✓	✓	✓	✓	✓	—	✓	✓	✓	—	—	—	—	—	—
169 pins product		✓	✓	✓	✓	—	—	—	—	✓	✓	—	—	—	—	—	—

✓: Available
—: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance “_A”, “_B”, or “_C” to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t_{GTISK}).

Table 20.16 Register settings for input/output pin function (PORT8) (1 of 2)

PSEL [4:0]Settings	Function	Pin P800	P801	P802	P803	P804	P805	P806	P807	P808	P809	P810	P811	P812	P813	P814	P815
00000b (value after reset)	Hi-z/ JTAG/SWD	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
00001b	AGT	AGTOA0	AGTOB0	—	—	—	—	—	—	—	—	—	—	—	—	—	—
00010b	GPT*2	GTIU	GTIV	GTIW	GTETRGC	GTETRGD	—	—	—	—	—	—	—	—	GTETRGA	—	—
00011b	GPT*2	GTIOC11A	GTIOC11B	GTIOC12A	GTIOC12B	GTIOC13A	—	—	GTIOC13A	GTIOC13B	—	GTIOC10A	GTIOC10B	GTIOC11A	GTIOC7B	GTIOC8B	GTIOC8A
00100b	SCI*1	CTS2_A	TXD2_A/ SDA2_A/ MOSI2_A	RXD2_A/ SCL2_A/ MISO2_A	SCK2_A	CTS_RTS2_A/ SS2_A	TXD8_A/ SDA8_A/ MOSI8_A	RXD8_A/ SCL8_A/ MISO8_A	—	—	—	—	—	—	—	—	—

Table 20.16 Register settings for input/output pin function (PORT8) (2 of 2)

PSEL [4:0]Settings	Function	P800	P801	P802	P803	P804	P805	P806	P807	P808	P809	P810	P811	P812	P813	P814	P815
00101b	SCI*1	—	—	—	—	—	—	—	—	RXD7_B/ SCL7_B/ MISO7_B	TXD7_B/ SDA7_B/ MOSI7_B	SCK7_B	CTS7_B	CTS_RTS7_B/ SS7_B	SCK7_A	—	—
01011b	BUS	—	—	—	—	—	—	—	—	—	—	—	—	—	SDCS	—	—
01101b	SCI	—	—	—	DE2	DE2	—	—	—	—	—	—	—	—	—	—	—
01110b	SCI	—	—	—	—	—	—	—	—	—	—	DE7	—	DE7	DE7	—	—
10000b	CANFD	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CRX0	CTX0
10011b	USBFS	—	—	—	—	—	—	—	—	—	—	—	USB_ID	USB_EXICEN	—	USB_DP	USB_DM
10101b	SDHI*1	—	—	—	—	—	—	—	—	—	—	SD1DAT2_A	SD1DAT3_A	SD1DAT4_A	—	—	—
10110b	ESWM (GMII/MII)	—	GPTP_PPS1	—	—	—	ET1_MDIO	ET1_MDC	—	—	—	—	—	—	—	—	—
11011b	PDMIF	—	—	—	—	—	—	—	—	—	—	PDMCLK0	PDMCLK1	PDMCLK2	PDMCLK2	—	—
11100b	OSPI	OM_0_SIO5	OM_0_DQS	OM_0_SIO6	OM_0_SIO1	OM_0_SIO7	—	—	—	OM_0_SCLK	OM_0_SCLKN	—	—	—	—	—	—
01111b	CEU	—	—	—	—	—	VIO_D15	VIO_D14	—	—	—	—	—	—	VIO_D13	—	—
11110b	ULPT	—	—	—	—	—	—	—	—	—	—	ULPTOA0	ULPTOB0	—	—	—	—
Don't care	USBFS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	USB_DP	USB_DM
ASEL bit		—	—	—	—	—	AN017/ IVCMP0	AN018	—	—	—	—	—	AN022	—	—	—
ISEL bit		IRQ11	IRQ12	IRQ18	IRQ19	IRQ14	IRQ30	IRQ0	IRQ11	IRQ15	IRQ20	IRQ21	IRQ22	IRQ23	IRQ15	IRQ16	IRQ15
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NCODR bit	N-ch Open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DSCR[1:0] bit	Drive capability control	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH
289, 303 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		✓	✓	✓	✓	✓	—	—	—	✓	✓	—	—	—	✓	✓	✓
176 pins product		✓	✓	✓	✓	✓	—	—	—	✓	✓	—	—	—	✓	✓	✓
169 pins product		✓	✓	✓	✓	✓	✓	✓	—	✓	✓	—	—	—	—	✓	✓

✓: Available
—: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance "_A", "_B", or "_C" to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t_{GTISK}).

Table 20.17 Register settings for input/output pin function (PORT9) (1 of 2)

PSEL [4:0]Settings	Function	P900	P901	P902	P903	P904	P905	P906	P907	P908	P909	P910	P911	P912	P913	P914	P915
00000b (value after reset)	Hi-z/ JTAG/SWD	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
00001b	AGT	—	AGTIO1	—	—	—	—	—	—	—	—	—	—	—	—	—	—
00010b	GPT*2	GTADSM0	GTADSM1	—	—	—	—	—	—	—	—	—	—	—	—	—	—
00011b	GPT*2	—	—	GTCPPO13	GTIOC11A	GTIOC11B	GTCPPO13	GTIOC13B	GTIOC13A	GTIOC12B	GTIOC12A	GTCPPO12	GTIOC3B	GTIOC3A	GTCPPO11	GTIOC5B	GTIOC5A
00100b	SCI*1	—	—	—	—	—	—	CTS6_A	SCK6_A	TXD6_A/ SDA6_A/ MOSI6_A	RXD6_A/ SCL6_A/ MISO6_A	—	—	—	—	CTS_RTS6_B/ SS6_B	CTS6_B

Table 20.17 Register settings for input/output pin function (PORT9) (2 of 2)

PSEL [4:0]Settings	Function	Pin															
		P900	P901	P902	P903	P904	P905	P906	P907	P908	P909	P910	P911	P912	P913	P914	P915
00101b	SCI*1	CTS3_C	CTS_RTS3_C/SS3_C	—	—	—	RXD3_B/SCL3_B/MISO3_B	—	—	—	—	—	—	—	—	—	—
01001b	RTC/CLKOUT/ACMPHS/ETHPHYCLK	—	—	ETHPHYCLK	—	—	—	—	—	—	—	—	—	—	CLKOUT	—	—
01010b	CAC/ADC16H	—	—	—	—	—	AD1FLAG1	AD0FLAG1	ADSYNC	ADST1	ADST0	—	—	—	—	—	—
01011b	BUS	—	—	ALE	—	—	—	—	—	—	—	—	—	—	—	—	—
01101b	SCI	—	—	—	—	—	—	—	DE6	—	—	—	—	—	—	DE6	—
01110b	SCI	—	DE3	—	—	—	—	—	—	—	—	—	—	—	—	—	—
10000b	CANFD	—	—	—	—	—	—	—	—	CRX1	CTX1	—	—	—	—	—	—
10010b	SSIE*1	—	—	AUDIO_CLK	—	—	—	SSILRCK1/SSIFS1_A	SSIBCK1_A	—	—	—	—	—	—	—	—
10011b	USBFS	—	—	—	—	—	—	USB_ID	USB_EXICEN	USB_OVRCURB	USB_OVRCURA	—	—	—	—	—	—
10100b	USBHS	—	—	—	—	—	—	—	—	USBHS_ID	USBHS_EXICEN	—	—	—	—	—	—
10110b	ESWM (GMII/MII)	—	—	—	—	ET1_RXD4	ET1_RX_CLK	ET1_RXD0	ET1_RXD1	ET1_RXD2	ET1_RXD3	ET1_TXD4	ET1_TXD5	ET1_TXD6	ET1_TXD7	—	—
10111b	ESWM (RMII)	—	—	—	—	—	RMII1_REF50CK	RMII1_RXD0	RMII1_RXD1	—	—	—	—	—	—	—	—
11000b	ESWM (RGMII)	—	—	—	—	—	RGMII1_RXC	RGMII1_RXD0	RGMII1_RXD1	RGMII1_RXD2	RGMII1_RXD3	—	—	—	—	—	—
11011b	PDMIF	—	—	—	—	—	PDMDAT1	PDMDAT0	PDMCLK2	PDMCLK1	PDMCLK0	—	—	—	—	—	—
01111b	CEU	—	—	VIO_D1	—	—	VIO_D6	VIO_D5	VIO_D4	VIO_D3	VIO_D2	—	—	—	—	—	—
11110b	ULPT	—	—	—	—	—	—	ULPTO1	ULPTEE1	ULPTEVI1	ULPTOA1	—	—	—	—	—	—
ASEL bit		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
ISEL bit		IRQ30	IRQ31	IRQ0	IRQ1	IRQ2	IRQ8	IRQ9	IRQ10	IRQ11	IRQ21-DS	IRQ7	IRQ6	IRQ5	IRQ3	IRQ9	IRQ8
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NCODR bit	N-ch Open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DSCR[1:0] bit	Drive capability control	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH
289, 303 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		—	—	✓	—	—	✓	✓	✓	✓	✓	—	—	—	—	—	—
176 pins product		—	—	—	—	—	✓	✓	✓	✓	✓	—	—	—	—	—	—
169 pins product		—	—	—	—	—	✓	✓	✓	✓	✓	—	—	—	—	—	—

✓: Available
—: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance"_A", "_B", or "_C" to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t_{GTISK}).

Table 20.18 Register settings for input/output pin function (PORTA) (1 of 2)

PSEL [4:0]Settings	Function	Pin															
		PA00	PA01	PA02	PA03	PA04	PA05	PA06	PA07	PA08	PA09	PA10	PA11	PA12	PA13	PA14	PA15
00000b (value after reset)	Hi-z/JTAG/SWD	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z

Table 20.18 Register settings for input/output pin function (PORTA) (2 of 2)

PSEL[4:0]Settings	Function	Pin															
		PA00	PA01	PA02	PA03	PA04	PA05	PA06	PA07	PA08	PA09	PA10	PA11	PA12	PA13	PA14	PA15
00010b	GPT*2	GTOVLO	GTOVUP	GTIW	GTIV	GTIU	GTETRGD	GTETRGC	GTETRGB	GTETRGD	—	—	GTIV	GTIW	GTOVUP	GTOVLO	—
00011b	GPT*2	GTCPPO7	GTCPPO8	GTCPPO9	GTCPPO10	GTIOC4B	GTIOC4A	GTIOC7B	GTIOC7A	GTCPPO11	GTCPPO12	GTCPPO13	GTIOC6A	GTIOC6B	GTIOC10A	GTIOC10B	GTIOC7A
00100b	SCI*1	—	—	RXD2_C/SCL2_C/MISO2_C	TXD2_C/SDA2_C/MOSI2_C	SCK2_C	CTS_RTS2_C/SS2_C	CTS2_C	—	—	—	—	—	—	—	—	—
00101b	SCI*1	CTS_RTS5_B/SS5_B	CTS5_B	—	—	—	—	—	CTS7_A	RXD5_B/SCL5_B/MISO5_B	TXD5_B/SDA5_B/MOSI5_B	SCK5_B	SCK9_C	RXD9_C/SCL9_C/MISO9_C	CTS_RTS9_C/SS9_C	TXD9_C/SDA9_C/MOSI9_C	CTS9_C
01001b	RTC/CLKOUT/ACMPHS/ETHPHYCLK	—	—	—	—	—	—	—	VCOUT	—	—	—	—	—	—	—	—
01010b	CAC/ADC16H	AD1FLAG1	AD0FLAG1	ADSYNC	ADST1	ADST0	—	—	—	—	—	—	—	—	—	—	—
01011b	BUS	A5	A4	A3	A2	A1/DQM3	A0/BC0/DQM1	CS1/CKE	RD	CS0/WE	CS3/CAS	CS2/RAS	D8/DQ8	D9/DQ9	D10/DQ10	D11/DQ11	EBCLK/SDCLK
01101b	SCI	—	—	—	—	DE2	DE2	—	—	—	—	—	—	—	—	—	—
01110b	SCI	DE5	—	—	—	—	—	—	—	—	—	DE5	DE9	—	DE9	—	—
10101b	SDHI*1	SD0DAT7_A	SD0DAT6_A	SD0DAT5_A	SD0DAT4_A	SD0DAT3_A	SD0DAT2_A	SD0DAT1_A	SD0DAT0_A	—	—	—	—	—	—	—	—
11011b	PDMIF	—	—	—	—	—	PDMDAT2	PDMDAT1	PDMDAT0	—	—	PDMCLK0	—	—	—	—	PDMCLK1
01111b	CEU	VIO_D5	VIO_D6	VIO_D7	VIO_D8	VIO_D9	VIO_D10	VIO_D11	VIO_D12	—	—	VIO_D15	—	—	—	—	VIO_D14
ASEL bit		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
ISEL bit		IRQ22	IRQ21	IRQ31	IRQ20	IRQ19	IRQ18	IRQ17	IRQ16	IRQ6	IRQ5	IRQ4	IRQ10	IRQ11	IRQ12	IRQ13	IRQ14
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NCODR bit	N-ch Open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DSCR[1:0] bit	Drive capability control	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH
289, 303 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
176 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
169 pins product		✓	—	✓	✓	✓	✓	—	—	✓	✓	✓	✓	✓	✓	✓	—

✓: Available
—: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance "_A", "_B", or "_C" to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t_{GTISK}).

Table 20.19 Register settings for input/output pin function (PORTB) (1 of 2)

PSEL[4:0]Settings	Function	Pin							
		PB00	PB01	PB02	PB03	PB04	PB05	PB06	PB07
00000b (value after reset)	Hi-z/JTAG/SWD	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
00011b	GPT*2	GTCPPO4	GTCPPO2	GTCPPO0	GTCPPO1	GTCPPO3	GTCPPO4	GTIOC9A	GTIOC9B
00101b	SCI*1	SCK1_B	CTS_RTS1_B/SS1_B	RXD5_C/SCL5_C/MISO5_C	TXD5_C/SDA5_C/MOSI5_C	SCK5_C	CTS5_C	CTS_RTS5_C/SS5_C	—
01010b	CAC/ADC16H	ADST0	AD1FLAG1	ADST1	ADSYNC	AD0FLAG1	—	—	—
01011b	BUS	—	ALE	—	—	—	—	—	—
01110b	SCI	DE1	DE1	—	—	DE5	—	DE5	—

Table 20.19 Register settings for input/output pin function (PORTB) (2 of 2)

PSEL[4:0]Settings	Function	Pin							
		PB00	PB01	PB02	PB03	PB04	PB05	PB06	PB07
10100b	USBHS	USBHS_VBUSEN	—	—	—	—	—	—	—
10110b	ESWM (GMII/MII)	ET0_TXD0	ET0_TX_CLK	ET0_TXD1	ET0_TXD2	ET0_TXD3	ET0_TXD7	ET0_TXD6	ET0_TXD5
10111b	ESWM (RMII)	RMII0_TXD0	—	RMII0_TXD1	—	—	—	—	—
11000b	ESWM (RGMII)	RGMII0_TXD0	—	RGMII0_TXD1	RGMII0_TXD2	RGMII0_TXD3	—	—	—
11011b	PDMIF	PDMDAT2	—	—	—	—	—	—	—
01111b	CEU	—	VIO_FLD	VIO_VD	VIO_HD	VIO_CLK	—	—	—
ASEL bit		—	—	—	—	—	—	—	—
ISEL bit		IRQ10	IRQ12	IRQ11	IRQ13	IRQ9	IRQ15	IRQ0	IRQ1
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓
NCODR bit	N-ch Open-drain	✓	✓	✓	✓	✓	✓	✓	✓
DSCR[1:0] bit	Drive capability control	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH
289, 303 pins product		✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		✓	✓	✓	✓	✓	—	—	—
176 pins product		✓	✓	✓	✓	✓	—	—	—
169 pins product		—	—	—	—	—	—	—	—

✓: Available
—: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance “_A”, “_B”, or “_C” to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t_{GTISK}).

Table 20.20 Register settings for input/output pin function (PORTC) (1 of 2)

PSEL [4:0]Settings	Function	Pin															
		PC00	PC01	PC02	PC03	PC04	PC05	PC06	PC07	PC08	PC09	PC10	PC11	PC12	PC13	PC14	PC15
00000b (value after reset)	Hi-z/ JTAG/SWD	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
00010b	GPT*2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GTADSM1	GTADSM0
00011b	GPT*2	GTCPPO7	GTCPPO6	GTCPPO5	GTCPPO4	GTCPPO3	GTCPPO2	GTCPPO1	GTCPPO0	GTCPPO8	—	GTCPPO13	GTCPPO12	GTCPPO11	GTCPPO10	GTCPPO9	—
00100b	SCI*1	—	—	—	—	—	—	—	—	—	—	—	CTS_RTS6_C/ SS6_C	SCK6_C	RXD6_C/ SCL6_C/ MISO6_C	TXD6_C/ SDA6_C/ MOSI6_C	CTS6_C
00101b	SCI*1	CTS_RTS7_C/ SS7_C	CTS7_C	SCK7_C	TXD7_C/ SDA7_C/ MOSI7_C	RXD7_C/ SCL7_C/ MISO7_C	—	—	—	—	—	—	—	—	—	—	—
01011b	BUS	D30/ DQ30	D29/ DQ29	D28/ DQ28	D27/ DQ27	D26/ DQ26	D25/ DQ25	D24/ DQ24	D23/ DQ23	D22/ DQ22	D21/ DQ21	D20/ DQ20	D19/ DQ19	D18/ DQ18	D17/ DQ17	D16/ DQ16	A16
01101b	SCI	—	—	—	—	—	—	—	—	—	—	—	DE6	DE6	—	—	—
01110b	SCI	DE7	—	DE7	—	—	—	—	—	—	—	—	—	—	—	—	—
10000b	CANFD	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CRX1
10110b	ESWM (GMII/MII)	—	—	—	—	—	—	—	—	—	—	—	ET0_MDC	ET0_MDIO	ET0_INT	ET0_WOL	—
11100b	OSPI	OM_1_SIO5	OM_1_SIO0	OM_1_SIO3	OM_1_SIO4	OM_1_SIO2	OM_1_CS1	OM_1_ECSINT1	OM_1_RESET	OM_1_CS0	OM_1_RSTO1	OM_1_WP1	—	—	—	—	—
ASEL bit		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
ISEL bit		IRQ28	IRQ27	IRQ26	IRQ25	IRQ24	IRQ23	IRQ22	IRQ21	IRQ29	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0	IRQ30
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NCODR bit	N-ch Open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 20.20 Register settings for input/output pin function (PORTC) (2 of 2)

PSEL [4:0]Settings	Function	Pin															
		PC00	PC01	PC02	PC03	PC04	PC05	PC06	PC07	PC08	PC09	PC10	PC11	PC12	PC13	PC14	PC15
DSCR[1:0] bit	Drive capability control	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH
289, 303 pins product*3		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		—	—	—	—	—	—	—	—	—	—	—	✓	✓	✓	✓	✓
176 pins product		—	—	—	—	—	—	—	—	—	—	—	✓	✓	✓	✓	✓
169 pins product		—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	✓	—

✓: Available
—: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance "_A", "_B", or "_C" to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t_{GTISK}).

Note 3. For BGA303 package, PC00 to PC07 are connected to the SiP Flash memory inside the package. To access SiP Flash memory, set OSPI in PSEL settings.

Table 20.21 Register settings for input/output pin function (PORTD)

PSEL[4:0]Settings	Function	Pin							
		PD00	PD01	PD02	PD03	PD04	PD05	PD06	PD07
00000b (value after reset)	Hi-z/JTAG/SWD	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
00010b	GPT*2	GTOWLO	—	—	—	—	—	—	—
00011b	GPT*2	GTCPPO5	GTCPPO2	GTCPPO1	GTIOC3B	GTIOC3A	GTIOC2B	GTIOC2A	GTCPPO0
00100b	SCI*1	—	SCK8_C	TXD8_C/ SDA8_C/ MOSI8_C	RXD8_C/ SCL8_C/ MISO8_C	CTS_RTS8_C/ SS8_C	CTS8_C	—	—
00101b	SCI*1	SCK5_A	—	—	—	—	—	—	—
01011b	BUS	A15	—	—	—	—	—	—	—
01101b	SCI	—	DE8	—	—	DE8	—	—	—
01110b	SCI	DE5	—	—	—	—	—	—	—
10000b	CANFD	CTX1	—	—	—	—	—	—	—
10100b	USBHS	—	—	—	USBHS_EXICEN	USBHS_ID	USBHS_OVRCURB	USBHS_OVRCURA	USBHS_VBUSEN
10101b	SDHI*1	—	SD0DAT2_C	SD0DAT1_C	SD0DAT0_C	SD0CMD_C	SD0CLK_C	SD0WP	SD0CD
10110b	ESWM (GMII/MII)	—	ET1_RXD6	ET1_RXD7	ET0_RXD4	ET0_RXD5	ET0_RXD6	ET0_RXD7	ET0_TXD4
ASEL bit		—	—	—	—	—	—	—	—
ISEL bit		IRQ23	IRQ22	IRQ21	IRQ21	IRQ20	IRQ19	IRQ18	IRQ17
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓
NCODR bit	N-ch Open-drain	✓	✓	✓	✓	✓	✓	✓	✓
DSCR[1:0] bit	Drive capability control	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH
289, 303 pins product		✓	✓	✓	✓	✓	✓	✓	✓
224 pins product		✓	—	—	—	—	—	—	—
176 pins product		✓	—	—	—	—	—	—	—
169 pins product		—	—	—	—	—	—	—	—

✓: Available
—: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance "_A", "_B", or "_C" to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t_{GTISK}).

21. Port Output Enable for GPT (POEG)

21.1 Overview

The Port Output Enable (POEG) function can place the General PWM Timer (GPT) output pins in the output disable state in one of the following ways:

- Input level detection of the GTETRGn (n = A to D) pins
- Output-disable request from the GPT
- Comparator interrupt request detection
- Oscillation stop detection of the clock generation circuit
- Register settings

The GTETRGn (n = A to D) pins can be used as GPT external trigger input pins.

Table 21.1 lists the POEG specifications, Figure 21.1 shows a block diagram, and Table 21.2 lists the input pins.

Table 21.1 POEG specifications

Parameter	Specifications
Output-disable control through input level detection	The GPT output pins can be disabled when a GTETRGn rising edge or high level is sampled after polarity and filter selection.
Output-disable request from the GPT	When the GTIOCxA pin and the GTIOCxB pin are driven to an active level simultaneously, the GPT generates an output-disable request to the POEG. Through reception of these requests, the POEG can control whether the GTIOCxA and GTIOCxB pins are output-disabled.
Output-disable control through the comparator (ACMPHS) interrupt request detection	The GPT output pins can be disabled when an interrupt request is generated by a change in the output results of any of the comparators.
Output-disable control through oscillation stop detection	The GPT output pins can be disabled when oscillation of the clock generation circuit stops.
Output-disable control by software (registers)	The GPT output pins can be disabled by modifying the register settings.
Interrupt	Interrupts can be generated by detecting the input level of external trigger input pins (GTETRGn pins). Interrupts can be generated when all GPT or ACMPHS output pins are driven to an active level simultaneously.
External trigger output to the GPT	The GTETRGn signals can be output to the GPT after polarity and filter selection. (count start, count stop, count clear, up-count, down-count, or input capture function)
Noise filtering	For input from the GTETRGn pins, PCLKB/1, PCLKB/8, PCLKB/32, or PCLKB/128 can be selected as the noise filtering clock. (Filtering is performed by sampling the input signals three times using the selected clock.) Positive or negative polarity can be selected for any of the GTETRGn input pins. Signal state after polarity and filter selection can be monitored.
Module-stop function	Module-stop state can be set for each group to reduce power consumption.
TrustZone Filter	Security and Privilege attribution can be set for each group.

Note: n = A to D, x = 0 to 13

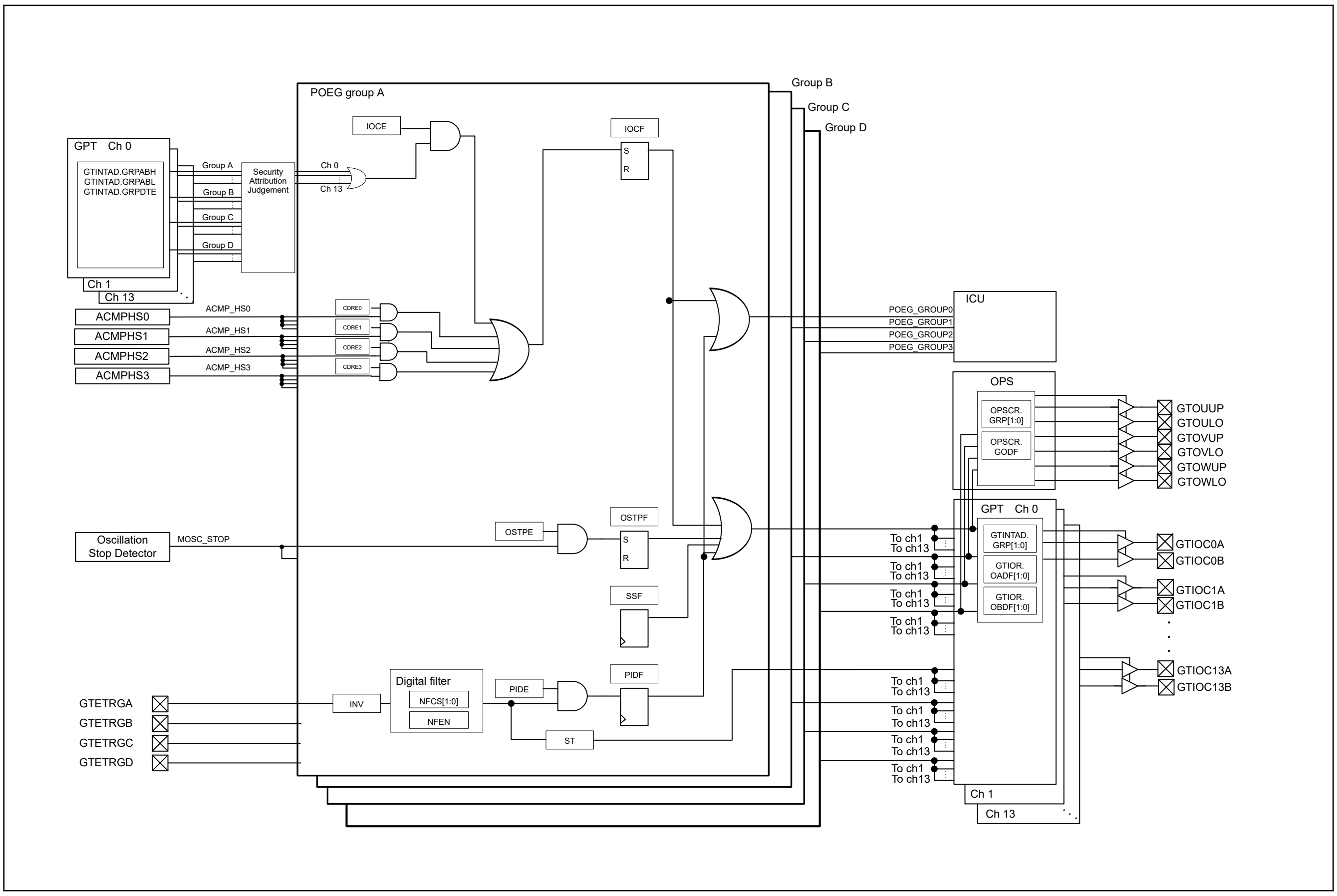


Figure 21.1 POEG block diagram

Table 21.2 POEG input pins

Pin name	I/O	Description
GTETRGA	Input	GPT output pin output-disable request signal or GPT external trigger input pin A
GTETRGB	Input	GPT output pin output-disable request signal or GPT external trigger input pin B
GTETRGC	Input	GPT output pin output-disable request signal or GPT external trigger input pin C
GTETRGD	Input	GPT output pin output-disable request signal or GPT external trigger input pin D

21.2 Register Descriptions

21.2.1 POEGGn : POEG Group n Setting Register (n = A to D)

Base address: POEG = 0x4021_2000
POEG_NS = 0x5021_2000

Offset address: 0x000 (POEGGA)
0x100 (POEGGB)
0x200 (POEGGC)
0x300 (POEGGD)

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	CDRE3	CDRE2	CDRE1	CDRE0	—	OSTPE	IOCE	PIDE	SSF	OSTPF	IOCF	PIDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIDF	Port Input Detection Flag 0: No output-disable request from the GTETRGn pin occurred 1: Output-disable request from the GTETRGn pin occurred.	R/W*1
1	IOCF	Detection Flag for GPT or ACMPHS Output-Disable Request 0: No output-disable request from GPT or comparator interrupt occurred. 1: Output-disable request from GPT or comparator interrupt occurred.	R/W*1
2	OSTPF	Oscillation Stop Detection Flag 0: No output-disable request from oscillation stop detection occurred 1: Output-disable request from oscillation stop detection occurred	R/W*1
3	SSF	Software Stop Flag 0: No output-disable request from software occurred 1: Output-disable request from software occurred	R/W
4	PIDE	Port Input Detection Enable 0: Disable output-disable requests from the GTETRGn pins 1: Enable output-disable requests from the GTETRGn pins	R/W*2
5	IOCE	Enable for GPT Output-Disable Request 0: Disable output-disable requests from GPT 1: Enable output-disable requests from GPT	R/W*2
6	OSTPE	Oscillation Stop Detection Enable 0: Disable output-disable requests from oscillation stop detection 1: Enable output-disable requests from oscillation stop detection	R/W*2
7	—	This bit is read as 0. The write value should be 0.	R/W
8	CDRE0	ACMP_HS0 Enable 0: Disable output-disable request from ACMPHS0 1: Enable output-disable request from ACMPHS0	R/W*2
9	CDRE1	ACMP_HS1 Enable 0: Disable output-disable request from ACMPHS1 1: Enable output-disable request from ACMPHS1	R/W*2
10	CDRE2	ACMP_HS2 Enable 0: Disable output-disable request from ACMPHS2 1: Enable output-disable request from ACMPHS2	R/W*2
11	CDRE3	ACMP_HS3 Enable 0: Disable output-disable request from ACMPHS3 1: Enable output-disable request from ACMPHS3	R/W*2
15:12	—	These bits are read as 0. The write value should be 0.	R/W
16	ST	GTETRGn Input Status Flag 0: GTETRGn input after filtering was 0 1: GTETRGn input after filtering was 1	R
27:17	—	These bits are read as 0. The write value should be 0.	R/W
28	INV	GTETRGn Input Reverse 0: Input GTETRGn as-is 1: Input GTETRGn in reverse	R/W
29	NFEN	Noise Filter Enable 0: Disable noise filtering 1: Enable noise filtering	R/W
31:30	NFCS[1:0]	Noise Filter Clock Select 0 0: Sample GTETRGn pin input level three times every PCLKB 0 1: Sample GTETRGn pin input level three times every PCLKB/8 1 0: Sample GTETRGn pin input level three times every PCLKB/32 1 1: Sample GTETRGn pin input level three times every PCLKB/128	R/W

Note: S-TYPE3, P-TYPE3
Note 1. Only 0 can be written to clear the flag.
Note 2. Can be modified only once after a reset.

The POEGGn (n = A to D) registers control the output-disable state of the GPT pins, interrupts, and the external trigger input to the GPT.

In the descriptions, POEGGn represents the POEGGn (n = A to D) registers.

21.3 Output-Disable Control Operation

If any of the following conditions is satisfied, the GTIOCxA, GTIOCxB, and the 3-phase PWM output for BLDC motor control pins can be set to output-disable:

- Input level or edge detection of the GTETRGn pins
  When POEGGn.PIDE is 1, the POEGGn.PIDF flag is set to 1.
- Output-disable request from the GPT
  When POEGGn.IOCE is 1, the POEGGn.IOCF flag is set to 1 if the disable request is enabled by GTINTAD.
  The GTINTAD.GRPABH and GTINTAD.GRPABL settings apply to the group selected by the GPT register GTINTAD.GRP[1:0] or OPSCR.GRP[1:0].

  Note: The disable request is valid only when the security attributions of GPT and POEG are same. The disable request from GPT with a security attribution different from the security attribution of POEG is invalidated by the security attribution judgment.
- Comparator (ACMPHS) interrupt request detection
  Comparator interrupt detection is activated when any of the POEGGn.CDREi (i = 0 to 3) is 1. When the associated comparator interrupt is generated, the GPT output pins are disabled. POEGGn.IOCF indicates the detection status.
- Oscillation stop detection for the clock generation circuit
  While POEGGn.OSTPE is 1, the halt status of the main clock oscillator is detected and the POEGGn. OSTPF flag is set to 1.
- SSF bit setting
  When POEGGn.SSF is set to 1, the GPT and PWM output are disabled.

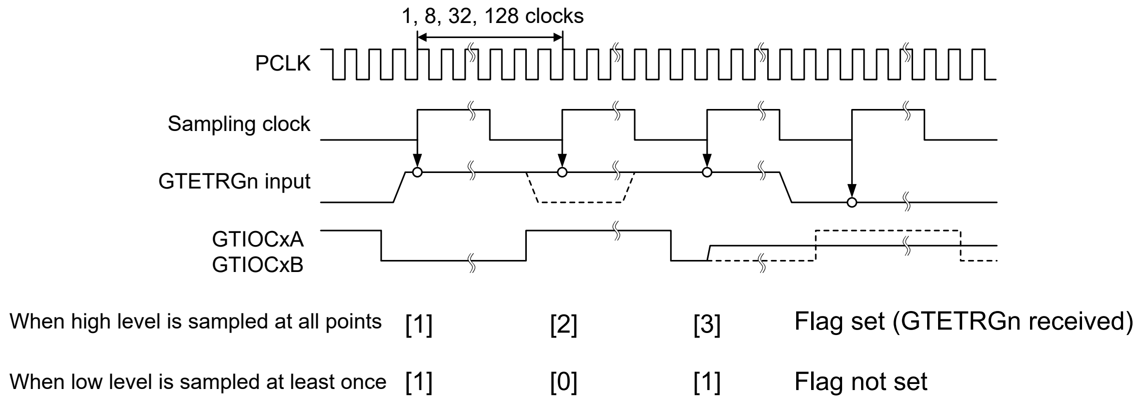
The output-disable state is controlled in the GPT module. The output-disable of the GTIOCxA and GTIOCxB pins is set in the GTINTAD.GRP[1:0], GTIOR.OADF[1:0], and GTIOR.OBDF[1:0] bits in GPTx. The output-disable of the 3-phase PWM output for BLDC motor control pins is set in the OPSCR.GRP[1:0] bits and OPSCR.GODF bit in GPT_OPS.

21.3.1 Pin Input Level Detection Operation

If the input conditions set in POEGGn.PIDE, POEGGn.NFCS[1:0], POEGGn.NFEN, and POEGGn.INV occur on the GTETRGn pins, the GPT output pins are output-disabled.

21.3.1.1 Digital Filter

Figure 21.2 shows high-level detection by the digital filter. When a high level associated with the POEGGn.INV polarity setting is detected three times consecutively with the sampling clock selected in POEGGn.NFCS[1:0], the detected level is recognized as high, and the GPT output pins are output-disabled. If even one low level is detected during this interval, the detected level is not recognized as high. In addition, in an interval where the sampling clock is not output, changes of the levels on the GTETRGn pins are ignored.



Note: PCLK = PCLKB
Note: Each channel output can be set in the GPT setting. Low level sampling can be set in the POEGGn.INV setting.

Figure 21.2 Example of digital filter operation

21.3.2 Output-Disable Requests from the GPT

For details on the operation, see the description for GTIOC Pin Output Negate Control in section 22, General PWM Timer (GPT).

21.3.3 Comparator Interrupt Detection

If POEGGn.CDREi (i = 0 to 3) is 1 when an associated comparator interrupt request is generated, the GPT output pins are output-disabled for each group. The status flag is POEGGn.IOCF which is shared with GPT output-disable detection.

21.3.4 Output-Disable Control Using Detection of Stopped Oscillation

When the oscillation stop detection function in the clock generation circuit detects stopped oscillation while POEGGn.OSTPE is 1, the GPT output pins are output-disabled for each group.

21.3.5 Output-Disable Control Using Registers

The GPT output pins can be directly controlled by writing 1 to the Software Stop flag, POEGGn.SSF.

21.3.6 Release from Output-Disable

To release the GPT output pins placed in the output-disable state, either return them to their initial state with a reset or clear all of the following flags:

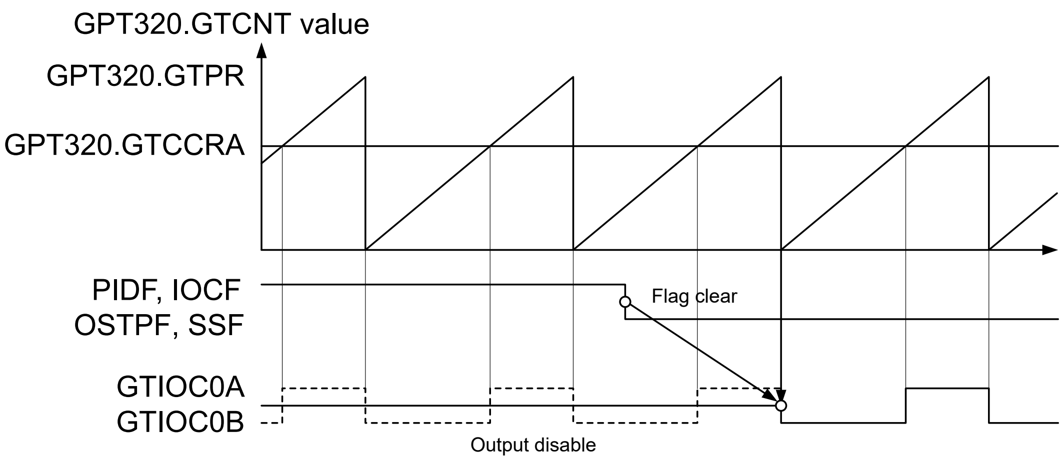
- POEGGn.PIDF
- POEGGn.IOCF
- POEGGn.OSTPF
- POEGGn.SSF

Writing 0 to the POEGGn.PIDF flag is ignored (the flag is not cleared) if the external input pins, GTETRGn are not disabled and the POEGGn.ST bit is not set to 0.

Writing 0 to the POEGGn.IOCF flag is valid (the flag is cleared) only if all of the GTST.OABHF and GTST.OABLF flags in the GPT are set to 0.

Writing 0 to the POEGGn.OSTPF flag is ignored (the flag is not cleared) if the OSTDSR.OSTDF flag in the clock generation circuit is not set to 0. In addition, when the flag set and release occur at the same time, the flag set takes precedence.

Figure 21.3 shows the release timing for output-disable. The output-disable is released at the beginning of the next count cycle of the GPT after the flag is cleared.



Note: The base address of the registers are the same.

Figure 21.3 Output-disable release timing for GPT pin outputs

21.4 Interrupt Sources

The POEG generates an interrupt request for the following factors:

- Output-disable control by the input level detection
- Output-disable request from the GPT or the comparator (ACMPHS) interrupt request detection

Table 21.3 lists the conditions for interrupt requests.

Table 21.3 Interrupt sources and conditions

Interrupt source	Symbol	Associated flag	Trigger conditions
POEG group A interrupt	POEG_GROUPA	POEGGA.IOCF	An output-disable request from a GPT disable request occurred
			An output-disable request from ACMPHS interrupt occurred
		POEGGA.PIDF	An output-disable request from the GTETRGA pin occurred
POEG group B interrupt	POEG_GROUPB	POEGGB.IOCF	An output-disable request from a GPT disable request occurred
			An output-disable request from ACMPHS interrupt occurred
		POEGGB.PIDF	An output-disable request from the GTETRGB pin occurred
POEG group C interrupt	POEG_GROUPC	POEGGC.IOCF	An output-disable request from a GPT disable request occurred
			An output-disable request from ACMPHS interrupt occurred
		POEGGC.PIDF	An output-disable request from the GTETRGC pin occurred
POEG group D interrupt	POEG_GROUPD	POEGGD.IOCF	An output-disable request from a GPT disable request occurred
			An output-disable request from ACMPHS interrupt occurred
		POEGGD.PIDF	An output-disable request from the GTETRGD pin occurred

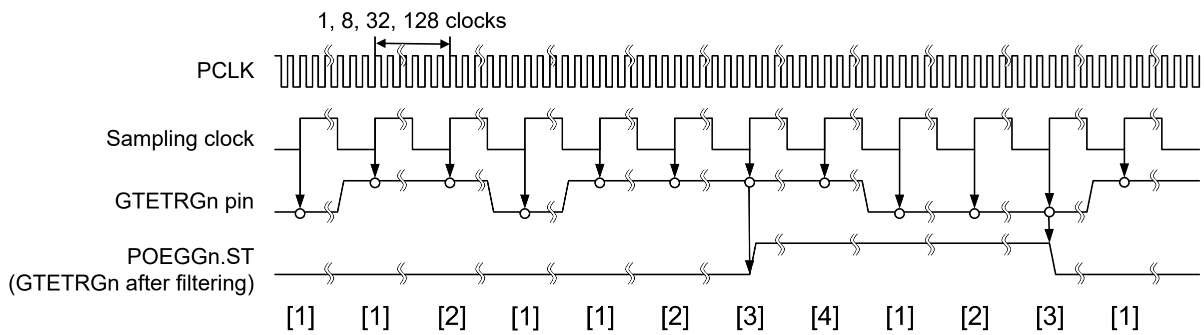
21.5 External Trigger Output to the GPT

The POEG outputs signals generated by filtering and level detection of GTETRGn pins input signals as the GPT operation trigger signal for the following:

- Count start
- Count stop
- Count clear
- Up-count
- Down-count
- Input capture

For the POEGGn.INV polarity setting signal, when the same level is input three times continuously with the sampling clock selected in POEGGn.NFCS[1:0], that value is output. Set the control registers the same as for the input level detection operation described in section 21.3.1. Pin Input Level Detection Operation The state after filtering can be monitored in POEGGn.ST.

Figure 21.4 shows the output timing of an external trigger to the GPT.



Note: PCLK = PCLKB
Note: Each channel output can be set in the GPT settings. Polarity can be reversed in POEGGn.INV.

Figure 21.4 Output timing of external trigger to the GPT

21.6 Usage Notes

21.6.1 Transition to Software Standby Mode

When using the POEG, do not invoke Software Standby mode. In this mode, the POEG stops and therefore output disable of the pins cannot be controlled.

21.6.2 Specifying Pins Associated with the GPT

The POEG controls output-disable only when a pin is associated with the GPT in the PmnPFS.PMR and PmnPFS.PSEL settings. When the pin is specified as a general I/O pin, the POEG does not perform output-disable control.

22. General PWM Timer (GPT)

22.1 Overview

The General PWM Timer (GPT) is a 32-bit timer with GPT32 × 14 channels. PWM waveforms can be generated by controlling the up-counter, down-counter, or the up- and down-counter. In addition, PWM waveforms can be generated for controlling brushless DC motors. The GPT can also be used as a general-purpose timer.

This GPT provides the high resolution PWM waveform generation function for the channel 0 to 3, the A/D conversion start request function, the asymmetric automatic dead time setting function, and the enhanced interrupt skipping function. In addition, it is enhanced with the external pulse width measuring function for the channel 0 to 3, the additional PWM modes for the channel 4 to 9, the extended buffer transfer function, the GTCPPO pin output function, and the inter-channel cooperation function.

Table 22.1 lists the GPT specifications, Table 22.2 shows the GPT functions, and Figure 22.1 shows a block diagram.

Table 22.1 GPT specifications

Item	Description
Functions	● 32 bits × 14 channels (GPT32n (n = 0 to 13)) ● Up-counting or down-counting (saw waves) or up/down-counting (triangle waves) for each counter ● Clock sources independently selectable for each channel ● Two input/output pins per channel ● Two output compare/input capture registers per channel ● For the two output compare/input capture registers of each channel, four registers are provided as buffer registers and are capable of operating as comparison registers when buffering is not in use ● In output compare operation, buffer switching can be at crests or troughs, enabling the generation of laterally asymmetric PWM waveforms ● Registers for setting up frame cycles in each channel with capability for generating interrupts at overflow or underflow ● Generation of dead times in PWM operation ● Generation of high accuracy duty in the vicinity of duty 0% and 100% PWM waveform ● In output compare operation, setting compare register is immediately used to generate PWM waveform with dead times ● Synchronous starting, stopping and clearing counters for arbitrary channels ● Count start, count stop, count clear, up-count, down-count, or input capture operation in response to a maximum of 8 ELC events ● Count start, count stop, count clear, up-count, down-count, or input capture operation in response to the status of two input pins ● Count start, count stop, count clear, up-count, down-count, or input capture operation in response to a maximum of 4 external triggers ● Output pin disable function by dead time error and detected short-circuits between output pins ● A/D conversion start request generation function ● PWM waveform for controlling brushless DC motors can be generated ● Compare match A to F event, overflow/underflow event, and input UVW edge event can be output to the ELC ● Enables the noise filter for input capture and input UVW ● Period count function ● External pulse width measuring function ● Logical operation between the channel output ● Synchronous counter clearing/counter setting/input capture among channels ● Bus clock: PCLKA, Core clock: GTCLK*1 ● Frequency ratio: PCLKA:PCLKD = 1:N (N = 1/2/4/8/16/32/64) (when using synchronous clock), PCLKA ≤ GTCLK (when using asynchronous clock GTIOCnA/B input/output pin polarity reversal function)

Note 1. GPT core clock (GTCLK) is PCLKD when synchronous clock is selected, and GPTCLK when asynchronous clock is selected. See Figure 22.3.

Table 22.2 GPT functions (1 of 2)

Parameter		Description
Count clock		GTCLK GTCLK/2 GTCLK/4 GTCLK/8 GTCLK/16 GTCLK/32 GTCLK/64 GTCLK/128 GTCLK/256 GTCLK/512 GTCLK/1024 GTETRGA, GTETRGB, GTETRGC, GTETRGD
Output compare/input capture registers (GTCCR)		GTCCRA GTCCRB
Compare/buffer registers		GTCCRC GTCCRD GTCCRE GTCCRF
Cycle setting register		GTPR
Cycle setting buffer register		GTPBR GTPDBR
I/O pins		GTIOCnA GTIOCnB (n = 0 to 13)
External trigger input pin*1		GTETRGA GTETRGB GTETRGC GTETRGD
Counter clear sources		GTPR register compare match Input capture Input pin status ELC event input GTETRGn (n = A to D) pin input GTCCR register compare match (GPT32n (n = 4 to 9)) Other channel's counter clear sources (GPT32n (n = 4 to 9))
Period count function		Available (GPT32n (n = 0 to 3, 10 to 13))
Compare match output	Low output	Available
	High output	Available
	Toggle output	Available
Input capture function		Available
Automatic addition of dead time		Available
PWM mode		Available
High accuracy PWM waveform		Available (GPT32n (n = DEADBEEF))
Phase count function		Available
External pulse width measuring function		Available
Buffer operation		Double buffer Simultaneous operation disable control for multiple channels Buffer operation by counter clearing/compare match
One-shot operation		Available
DMAC/DTC activation		All the interrupt sources
A/D conversion start request		Compare match of GTADTRA or GTADTRB register
Brushless DC motor control function		Available

Table 22.2 GPT functions (2 of 2)

Parameter	Description
Interrupt sources	11 sources ● GTCCRA compare match/input capture(GPTn_CCMPA) ● GTCCRB compare match/input capture(GPTn_CCMPB) ● GTCCRC compare match(GPTn_CMPC) ● GTCCRD compare match(GPTn_CMPD) ● GTCCRE compare match(GPTn_CMPE) ● GTCCRF compare match(GPTn_CMPF) ● GTCNT overflow (GTPR compare match) (GPTn_OVF) ● GTCNT underflow (GPTn_UDF) ● GTPC count stop(GPTx_PC) (x = 0 to 3)
Interrupt skipping function	● Skipping of interrupts of GTCNT counter overflow (GTPR register compare match) (GPTn_OVF) and GTCNT counter underflow (GPTn_UDF) (interlocked with other interrupts and A/D conversion start requests) ● Skipping of GTADTRA and GTADTRB register compare match (GPT32y (y = 4 to 9)) ● Buffer operation skipping function
Event linking (ELC) function	Available*2
Noise filtering function	Available
Logical operation between the channel output	Available
Synchronous counter clearing/counter setting/input capture	Available (GPT32n (n = 4 to 9))
TrustZone Filter	Security and Privilege attribution can be set for each UNIT.*3

Note:

Note 1. GTETRGn connects to GPT through the POEG module. Therefore, to use the GPT function, supply the POEG clock by clearing the MSTPCRD.MSTPDn (n = 11 to 14) bit.

Note 2. See section 22.6. Operations Linked by ELC.

Note 3. UNIT0 to UNIT3 is CH0 to CH3.
UNIT4 is CH4 to CH9.
UNIT5 to 8 is CH10 to CH13.

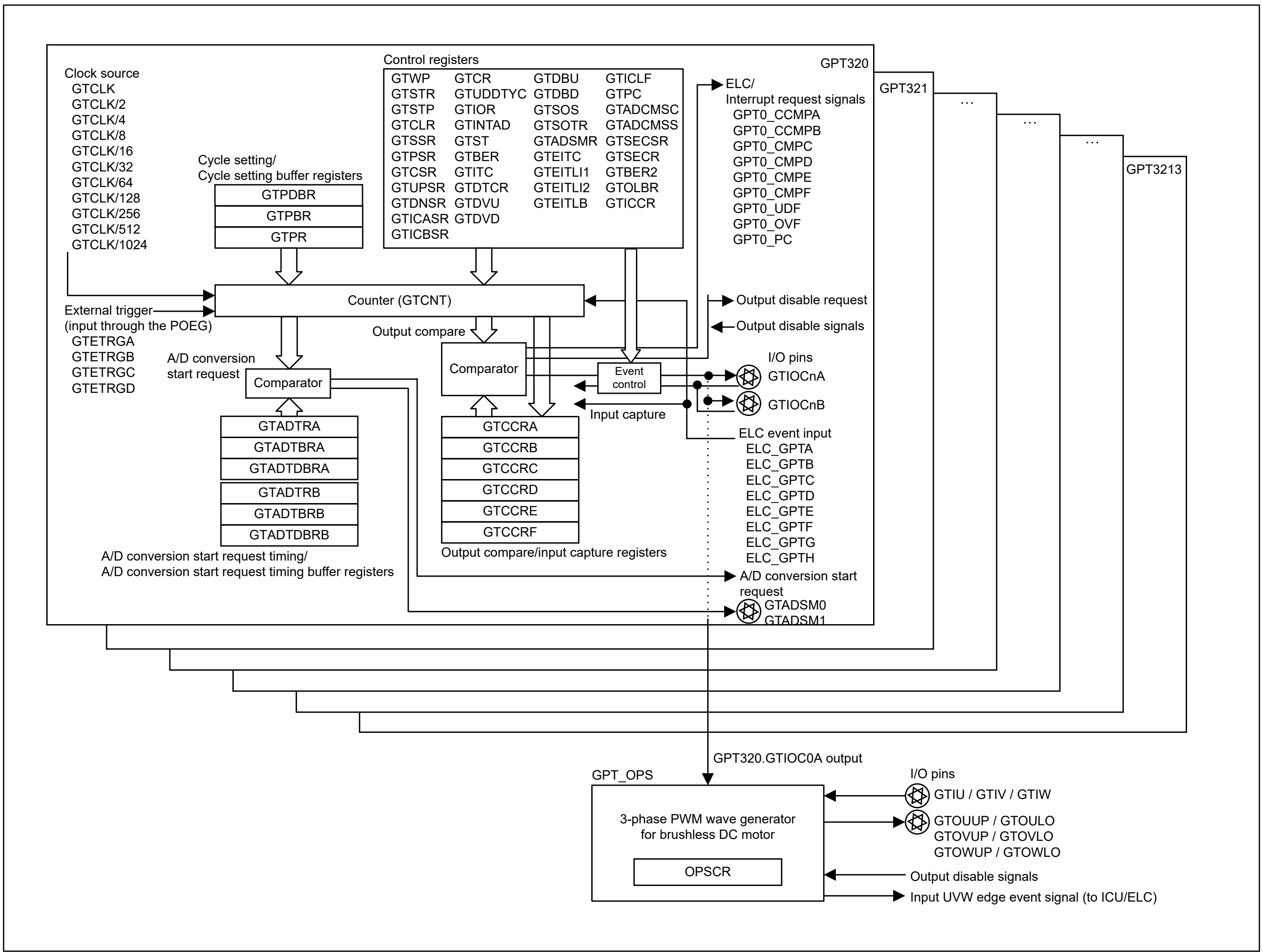


Figure 22.1 GPT block diagram (Saw-wave PWM mode 1, Saw-wave one-shot pulse mode, Triangle-wave PWM mode1,2,3)

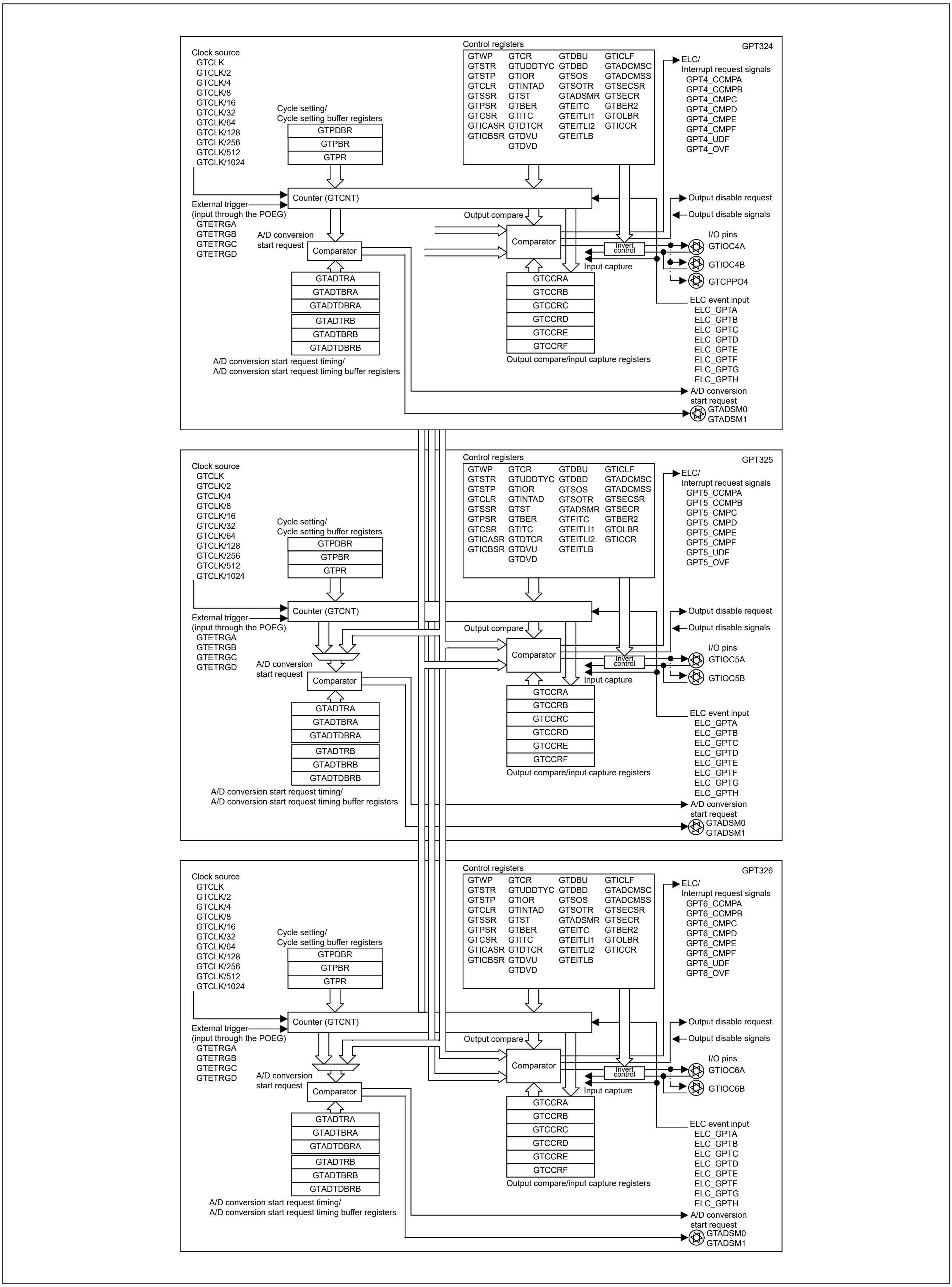


Figure 22.2 GPT block diagram (Saw-wave PWM mode 1, 2, Saw-wave one-shot pulse mode, Triangle-wave PWM mode1,2,3, Complementary PWM mode 1, 2, 3, 4)

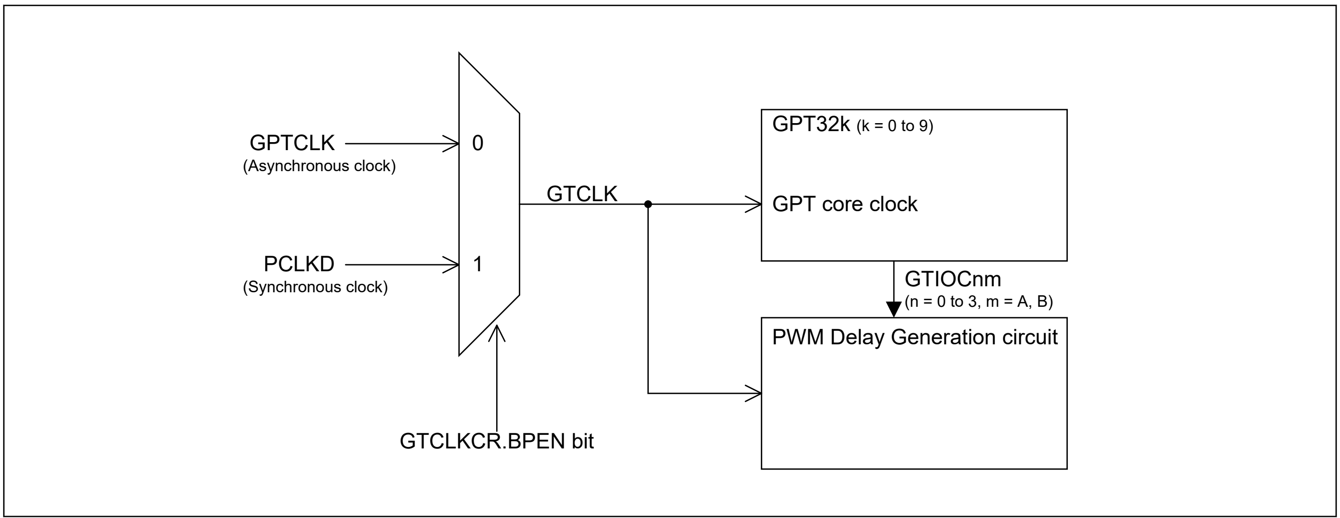


Figure 22.3 GPT core clock selection diagram

In this specification, three consecutive channels to configure complementary PWM mode is defined as complementary PWM mode channel group. The lowest position channel of complementary PWM mode channel group is defined as master channel. The second channel is defined as slave channel 1. The highest position channel is defined as slave channel 2.

Figure 22.4 shows an example using multiple GPTs.

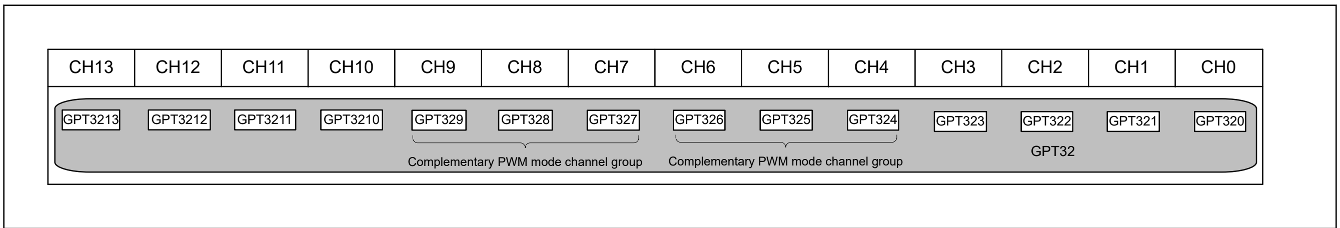


Figure 22.4 Association between GPT channels and module names

Table 22.3 lists the I/O pins.

Table 22.3 GPT I/O pins

Channel	Pin name	I/O	Function
Common	GTETRGx	Input	External trigger input pin x (input through the POEG)
	GTADSM0	Output	A/D conversion start request monitor 0 output pin
	GTADSM1	Output	A/D conversion start request monitor 1 output pin
GPT32n	GTIOCnA	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOCnB	I/O	GTCCRB register input capture input/output compare output/PWM output pin
	GTCPPOk	Output	Toggle output synchronized with PWM period
GPT_OPS	GTIU	Input	Hall sensor input pin U
	GTIV	Input	Hall sensor input pin V
	GTIW	Input	Hall sensor input pin W
	GTOUUP	Output	3-phase PWM output for BLDC motor control (positive U-phase)
	GTOULO	Output	3-phase PWM output for BLDC motor control (negative U-phase)
	GTOVUP	Output	3-phase PWM output for BLDC motor control (positive V-phase)
	GTOVLO	Output	3-phase PWM output for BLDC motor control (negative V-phase)
	GTOWUP	Output	3-phase PWM output for BLDC motor control (positive W-phase)
	GTOWLO	Output	3-phase PWM output for BLDC motor control (negative W-phase)

Note: x: A to D
n: 0 to 13
k: 0 to 13

22.2 Register Descriptions

22.2.1 GTWP : General PWM Timer Write-Protection Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x00

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PRKEY[7:0]								—	—	—	CMNWP	CLRWP	STPWP	STRWP	WP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	WP	Register Write Disable 0: Write to the register enabled 1: Write to the register disabled	R/W
1	STRWP	GTSTR.CSTRT Bit Write Disable 0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
2	STPWP	GTSTP.CSTOP Bit Write Disable 0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
3	CLRWP	GTCLR.CCLR Bit Write Disable 0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
4	CMNWP	Common Register Write Disabled 0: Write to the register is enabled 1: Write to the register is disabled	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PRKEY[7:0]	GTWP Key Code When 0xA5 is written to these bits, writing to the WP, STRWP, STPWP, CLRWP, and CMNWP bits are permitted. These bits are read as 0.	W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

GTWP enables or disables writing to registers to prevent accidental modification. Protection by the GTWP register is only for the writes by the CPU. GTWP does not protect registers from updates that occur in association with CPU writes.

WP bit (Register Write Disable)

The following is a list of write enabled or disabled registers:

GTSSR, GTPSR, GTCSR, GTUPSR, GTDNSR, GTICASR, GTICBSR, GTCR, GTUDDTYC, GTIOR, GTINTAD, GTST, GTBER, GTITC, GTCNT, GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, GTCCRF, GTPR, GTPBR, GTPDBR, GTADTRA, GTADTBRA, GTADTDBRA, GTADTRB, GTADTBRB, GTADTDBRB, GTDTCR, GTDVU, GTDVD, GTDBU, GTDBD, GTSOS, GTSOTR, GTADSMR, GTEITC, GTEITLI1, GTEITLI2, GTEITLB, GTICLF, GTPC, GTADCMSC, GTADCMSS, GTBER2, GTOLBR, GTICCR.

STRWP bit (GTSTR.CSTRT Bit Write Disable)

The STRWP bit enables or disables starting the updating of counter values by writing to the CSTRTn bit (n = 0 to 13) corresponding to a channel number in the GTSTR register.

The bit position of each CSTRTn bit in the GTSTR register is allocated to the channel with the corresponding number, and writing to the GTSTR register for any channel results in writing to the registers of all channels. The STRWP bit for each channel does not control writing but only controls updating of the CSTRT bit for the corresponding channel when simultaneously writing to all channels.

Therefore, when writing to the CSTRT bits of a channel for which the setting of the STRWP bit is 1 (disabling writing), the CSTRT bit for the given channel is not updated, but the CSTRT bits corresponding to channel for which the setting of the STRWP bit is 0 (enabling writing) are updated. For example, when the setting of the GPT320.GTWP.STRWP bit is 0 (enabling writing), writing 1 to the GPT321.GTSTR.CSTRT0 bit when its current setting is 0 causes the value to be updated, and the GPT320.GTCNT counter starts to run. When the setting of the GPT320.GTWP.STRWP bit is 1 (disabling writing), writing 1 to the GPT321.GTSTR.CSTRT0 bit when its current setting is 0 leaves the bit with the value 0, and the GPT320.GTCNT counter does not run.

If you want to protect all bits in the GTSTR register from being updated, set the STRWP bits of all channels to 1.

STPWP bit (GTSTP.CSTOP Bit Write Disable)

The STPWP bit enables or disables starting the updating of counter values by writing to the CSTOPn bit (n = 0 to 13) corresponding to a channel number in the GTSTP register.

The bit position of each CSTOPn bit in the GTSTP registers is allocated to the channel with the corresponding number, and the writing to the GTSTP register for any channel results in writing to the registers of all channels. The STPWP bit for each channel does not control writing but only controls updating of the CSTOP bit for the corresponding channel when simultaneously writing to all channels.

Therefore, when writing to the CSTOP bits of a channel for which the setting of the STPWP bit is 1 (disabling writing), the CSTOP bit for the given channel is not updated, but the CSTOP bits corresponding to channel for which the setting of the STPWP bit is 0 (enabling writing) are updated. For example, when the setting of the GPT320.GTWP.STPWP bit is 0 (enabling writing), writing 1 to the GPT321.GTSTP.CSTOP0 bit when its current setting is 0 causes the value to be updated, and the GPT320.GTCNT counter is stopped. When the setting of the GPT320.GTWP.STPWP bit is 1 (disabling writing), writing 1 to the GPT321.GTSTP.CSTOP0 bit when its current setting is 0 leaves the bit with the value 0, and the GPT320.GTCNT counter is not stopped.

If you want to protect all bits in the GTSTP register from being updated, set the STPWP bits of all channels to 1.

CLRWP bit (GTCLR.CCLR Bit Write Disable)

CLRWP bit enables or disables starting the updating of counter values by writing to the CCLRn bit (n = 0 to 13) corresponding to a channel number in the GTCLR register.

The bit position of each CCLRn bit in the GTCLR registers is allocated to the channel with the corresponding number, and the writing to the GTCLR register for any channel results in writing to the registers of all channels. The CLRWP bit for each channel does not control writing but only controls updating of the CCLR bit for the corresponding channel when simultaneously writing to all channels.

Therefore, when writing to the CCLR bits of a channel for which the setting of the CLRWP bit is 1 (disabling writing), the CCLR bit for the given channel is not updated, but the CCLR bits corresponding to channel for which the setting of the CLRWP bit is 0 (enabling writing) are updated. For example, when the setting of the GPT320.GTWP.CLRWP bit is 0 (enabling writing), writing 1 to the GPT321.GTCLR.CCLR0 bit when its current setting is 0 causes the value to be updated, and the GPT320.GTCNT counter is cleared. When the setting of the GPT320.GTWP.CLRWP bit is 1 (disabling writing), writing 1 to the GPT321.GTCLR.CCLR0 bit when its current setting is 0 leaves the bit with the value 0, and the GPT320.GTCNT counter is not cleared.

If you want to protect all bits in the GTCLR register from being updated, set the CLRWP bits of all channels to 1.

CMNWP bit (Common Register Write Disabled)

CMNWP bit enables or disables starting the updating of counter values by writing to the SECSELn bit (n = 0 to 13) corresponding to a channel number in the GTSECSR register or to the GTSECR register.

The bit position of each SECSEL bit in the GTSECSR registers is allocated to the channel with the corresponding number, and the writing to the GTSECSR register for any channel results in writing to the registers of all channels. Writing to the GTSECR register of any channel leads to writing to the registers of all channels. The CMNWP bit for each channel does not control writing but only controls updating of the SECSEL bit and the GTSECR register value for the corresponding channel when simultaneously writing to all channels.

Therefore, when writing to the SECSEL bit and the GTSECR register value of a channel for which the setting of the CMNWP bit is 1 (disabling writing), the SECSEL bit and the GTSECR register value for the given channel is not updated, but the SECSEL bit and the GTSECR register value corresponding to channel for which the setting of the CMNWP bit is 0 (enabling writing) are updated.

For example, when the setting of the GPT320.GTWP.CMNWP bit is 0 (enabling writing), writing to the GPT321.GTSECSR.SECSEL0 bit causes the value of the GPT320.GTSECSR.SECSEL0 bit to be updated. In the same way, writing to the GPT321.GTSECR register updates the value of the GPT320.GTSECR register. When the setting of the GPT320.GTWP.CMNWP bit is 1 (disabling writing), writing to the GPT321.GTSECSR.SECSEL0 bit does not cause the value of the GPT320.GTSECSR.SECSEL0 bit to be updated. In the same way, writing to the GPT321.GTSECR register does not update the value of the GPT320.GTSECR register.

If you want to protect all bits in the GTSECSR and GTSECR registers from being updated, set the CMNWP bits of all channels to 1.

PRKEY[7:0] bit (GTWP Key Code)

This bit controls whether the WP, STRWP, STPWP, CLRWP, and CMNWP bits can be overwritten.

22.2.2 GTSTR : General PWM Timer Software Start Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x04

Bit position:	31																														0
Bit field:	CSTRT31 to CSTRT0																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CSTRT0 to CSTRT31*1	Channel n GTCNT Count Start (n is the same as the bit position value) 0: GTCNT counter is not started 1: GTCNT counter is started	R/W

Note 1. The bits that can be used vary depending on the product. The n in CSTRTn is the same as the GPT channel number. For this product, n is 0 to 13.

The GTSTR starts the GTCNT counter operation for each channel n, where n = 0 to 13.

The GTSTR bit number represents the channel number. The GTSTR register of each channel is shared by all of the channels. The GTCNT counter starts for the channel associated with the GTSTR bit number where 1 is written. Writing 0 has no effect on the status of GTCNT counter and the value of GTSTR register.

In complementary PWM mode, writing to the CSTRTn bit of master channel is only valid. The bit on slave channels reflects the bit value of master channel.

For the association between module names and channel numbers, see Figure 22.4.

CSTRTn bits (Channel n GTCNT Count Start (n = 0 to 13))

The CSTRTn bits start channel n of the GTCNT counter operation. Writing to the GTSTR.CSTRTn bit (n = 0 to 13) has no effect unless the GTSSR.CSTRT bit is set to 1.

The read data shows the counter status of each channel (GTCR.CST bit). A value of 0 means the counter is stopped and 1 means the counter is running.

22.2.3 GTSTP : General PWM Timer Software Stop Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x08

Bit position:	31 to 0
Bit field:	CSTOP31 to CSTOP0
Value after reset:	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Bit	Symbol	Function	R/W
31:0	CSTOP0 to CSTOP31*1	Channel n GTCNT Count Stop (n is the same as the bit position value) 0: GTCNT counter is not stopped 1: GTCNT counter stopped	R/W

Note 1. The bits that can be used vary depending on the product. The n in CSTOPn is the same as the GPT channel number. For this product, n is 0 to 13.

The GTSTP stops the GTCNT counter operation for each channel n, where n = 0 to 13.

The GTSTP bit number represents the channel number. The GTSTP register of each channel is shared by all the channels. The GTCNT counter stops for the channel associated with the GTSTP bit number where 1 is written. Writing 0 has no effect on the status of the GTCNT counter and the value of GTSTP register.

In complementary PWM mode, writing to the CSTOPn bit of master channel is only valid. The bit on slave channels reflects the bit value of master channel.

For the association between module names and channel numbers, see Figure 22.4.

CSTOPn bits (Channel n GTCNT Count Stop (n = 0 to 13))

The CSTOPn bits stop channel n of the GTCNT counter operation. Writing to the GTSTP.CSTOPn bit (n = 0 to 13) has no effect unless the GTPSR.CSTOP bit is set to 1. The read data shows the counter status of each channel (invert of GTCR.CST bit). A value of 0 means the counter is running and 1 means the counter is stopped.

22.2.4 GTCLR : General PWM Timer Software Clear Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x0C

Bit position:	31 to 0
Bit field:	CCLR31 to CCLR0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	CCLR0 to CCLR31*1	Channel n GTCNT Count Clear (n : the same as bit position value) 0: GTCNT counter is not cleared 1: GTCNT counter is cleared	W

Note 1. The bits that can be used vary depending on the product. The n of CCLRn is the same as the GPT channel number. For this product, n is 0 to 13.

The GTCLR is a write-only register that clears the GTCNT counter operation for each channel n, where n = 0 to 13.

The GTCLR bit number represents the channel number. The GTCLR register of each channel is shared by all the channels. The GTCNT counter is cleared for the channel associated with the GTCLR bit number where 1 is written. Writing 0 has no effect on the status of GTCNT counter.

In complementary PWM mode, writing to the CCLRn bit of master channel is only valid. The bit on slave channels reflects the bit value of master channel.

For the association between module names and channel numbers, see Figure 22.4.

CCLRn bits (Channel n GTCNT Count Clear (n = 0 to 13))

When the counting direction flag is set for decrement (GTST.TUCF flag = 0) with saw-wave mode selected in the GTCR.MD[2:0] bits, the value of the GTCNT counter becomes that of the corresponding GTPR register in response to writing 1 to the CCLRn bit. The value of the counter becomes 0x0000 0000 with other settings. These bits are read as 0.

22.2.5 GTSSR : General PWM Timer Start Source Select Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CSTRT	—	—	—	—	—	—	—	SSELCH	SSELCG	SSELCF	SSELCE	SSELCD	SSELCC	SSELCB	SSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SSCBFAH	SSCBFAL	SSCBRAH	SSCBRAL	SSCAFBH	SSCAFBL	SSCARBH	SSCARBL	SSGTRGDF	SSGTRGDR	SSGTRGCF	SSGTRGCR	SSGTRGBF	SSGTRGBR	SSGTRGAF	SSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SSGTRGAR	GTETRGA Pin Rising Input Source Counter Start Enable 0: Counter start disabled on the rising edge of GTETRGA input 1: Counter start enabled on the rising edge of GTETRGA input	R/W*1
1	SSGTRGAF	GTETRGA Pin Falling Input Source Counter Start Enable 0: Counter start disabled on the falling edge of GTETRGA input 1: Counter start enabled on the falling edge of GTETRGA input	R/W*1
2	SSGTRGBR	GTETRGB Pin Rising Input Source Counter Start Enable 0: Counter start disabled on the rising edge of GTETRGB input 1: Counter start enabled on the rising edge of GTETRGB input	R/W*1
3	SSGTRGBF	GTETRGB Pin Falling Input Source Counter Start Enable 0: Counter start disabled on the falling edge of GTETRGB input 1: Counter start enabled on the falling edge of GTETRGB input	R/W*1
4	SSGTRGCR	GTETRGC Pin Rising Input Source Counter Start Enable 0: Counter start disabled on the rising edge of GTETRGC input 1: Counter start enabled on the rising edge of GTETRGC input	R/W*1
5	SSGTRGCF	GTETRGC Pin Falling Input Source Counter Start Enable 0: Counter start disabled on the falling edge of GTETRGC input 1: Counter start enabled on the falling edge of GTETRGC input	R/W*1
6	SSGTRGDR	GTETRGD Pin Rising Input Source Counter Start Enable 0: Counter start disabled on the rising edge of GTETRGD input 1: Counter start enabled on the rising edge of GTETRGD input	R/W*1
7	SSGTRGDF	GTETRGD Pin Falling Input Source Counter Start Enable 0: Counter start disabled on the falling edge of GTETRGD input 1: Counter start enabled on the falling edge of GTETRGD input	R/W*1
8	SSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Start Enable 0: Counter start disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter start enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	SSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Start Enable 0: Counter start disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter start enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W

Bit	Symbol	Function	R/W
10	SSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Start Enable 0: Counter start disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter start enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	SSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Start Enable 0: Counter start disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter start enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	SSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Start Enable 0: Counter start disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter start enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	SSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Start Enable 0: Counter start disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter start enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	SSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Start Enable 0: Counter start disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter start enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	SSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Start Enable 0: Counter start disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter start enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	SSELCA	ELC_GPTA Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTA input 1: Counter start enabled at the ELC_GPTA input	R/W*1
17	SSELCB	ELC_GPTB Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTB input 1: Counter start enabled at the ELC_GPTB input	R/W*1
18	SSELCC	ELC_GPTC Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTC input 1: Counter start enabled at the ELC_GPTC input	R/W*1
19	SSELCD	ELC_GPTD Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTD input 1: Counter start enabled at the ELC_GPTD input	R/W*1
20	SSELCE	ELC_GPTE Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTE input 1: Counter start enabled at the ELC_GPTE input	R/W*1
21	SSELCF	ELC_GPTF Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTF input 1: Counter start enabled at the ELC_GPTF input	R/W*1
22	SSELCG	ELC_GPTG Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTG input 1: Counter start enabled at the ELC_GPTG input	R/W*1
23	SSELCH	ELC_GPTH Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTH input 1: Counter start enabled at the ELC_GPTH input	R/W*1
30:24	—	These bits are read as 0. The write value should be 0.	R/W
31	CSTRT	Software Source Counter Start Enable 0: Counter start disabled by the GTSTR register 1: Counter start enabled by the GTSTR register	R/W*1

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

The GTSSR sets the source to start the GTCNT counter.

Input from GTETRGn (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

SSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Start Enable)

The SSGTRGAR bit enables or disables the GTCNT counter start on the rising edge of the GTETRGA pin input.

SSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Start Enable)

The SSGTRGAF bit enables or disables the GTCNT counter start on the falling edge of the GTETRGA pin input.

SSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Start Enable)

The SSGTRGBR bit enables or disables the GTCNT counter start on the rising edge of the GTETRGB pin input.

SSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Start Enable)

The SSGTRGBF bit enables or disables the GTCNT counter start on the falling edge of the GTETRGB pin input.

SSGTRGCR bit (GTETRGC Pin Rising Input Source Counter Start Enable)

The SSGTRGCR bit enables or disables the GTCNT counter start on the rising edge of the GTETRGC pin input.

SSGTRGCF bit (GTETRGC Pin Falling Input Source Counter Start Enable)

The SSGTRGCF bit enables or disables the GTCNT counter start on the falling edge of the GTETRGC pin input.

SSGTRGDR bit (GTETRGD Pin Rising Input Source Counter Start Enable)

The SSGTRGDR bit enables or disables the GTCNT counter start on the rising edge of the GTETRGD pin input.

SSGTRGDF bit (GTETRGD Pin Falling Input Source Counter Start Enable)

The SSGTRGDF bit enables or disables the GTCNT counter start on the falling edge of the GTETRGD pin input.

SSCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Start Enable)

The SSCARBL bit enables or disables the GTCNT counter start on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 0.

In complementary PWM mode, this setting is invalid.

SSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Start Enable)

The SSCARBH bit enables or disables the GTCNT counter start on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

In complementary PWM mode, this setting is invalid.

SSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Start Enable)

The SSCAFBL bit enables or disables the GTCNT counter start on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

In complementary PWM mode, this setting is invalid.

SSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Start Enable)

The SSCAFBH bit enables or disables the GTCNT counter start on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

In complementary PWM mode, this setting is invalid.

SSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Start Enable)

The SSCBRAL bit enables or disables the GTCNT counter start on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

In complementary PWM mode, this setting is invalid.

SSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Start Enable)

The SSCBRAH bit enables or disables the GTCNT counter start on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

In complementary PWM mode, this setting is invalid.

SSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Start Enable)

The SSCBFAL bit enables or disables the GTCNT counter start on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

In complementary PWM mode, this setting is invalid.

SSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Start Enable)

The SSCBFAH bit enables or disables the GTCNT counter start on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

In complementary PWM mode, this setting is invalid.

SSELCm bit (ELC_GPTm Event Source Counter Start Enable) (m = A to H)

The SSELCm bit enables or disables the GTCNT counter start at the ELC_GPTm event input.

CSTRT bit (Software Source Counter Start Enable)

The CSTRT bit enables or disables the GTCNT counter start by GTSTR register.

22.2.6 GTPSR : General PWM Timer Stop Source Select Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CSTOP	—	—	—	—	—	—	—	PSELCH	PSELCG	PSELCF	PSELCE	PSELCD	PSELCC	PSELCB	PSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PSCBFAH	PSCBFAL	PSCBRAH	PSCBRAL	PSCAFBH	PSCAFBL	PSCARBH	PSCARBL	PSGTRGDF	PSGTRGDR	PSGTRGCF	PSGTRGCR	PSGTRGBF	PSGTRGBR	PSGTRGAF	PSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PSGTRGAR	GTETRGA Pin Rising Input Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTETRGA input 1: Counter stop enabled on the rising edge of GTETRGA input	R/W*1
1	PSGTRGAF	GTETRGA Pin Falling Input Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTETRGA input 1: Counter stop enabled on the falling edge of GTETRGA input	R/W*1
2	PSGTRGBR	GTETRGB Pin Rising Input Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTETRGB input 1: Counter stop enabled on the rising edge of GTETRGB input	R/W*1
3	PSGTRGBF	GTETRGB Pin Falling Input Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTETRGB input 1: Counter stop enabled on the falling edge of GTETRGB input	R/W*1
4	PSGTRGCR	GTETRGC Pin Rising Input Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTETRGC input 1: Counter stop enabled on the rising edge of GTETRGC input	R/W*1
5	PSGTRGCF	GTETRGC Pin Falling Input Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTETRGC input 1: Counter stop enabled on the falling edge of GTETRGC input	R/W*1

Bit	Symbol	Function	R/W
6	PSGTRGDR	GTETRGD Pin Rising Input Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTETRGD input 1: Counter stop enabled on the rising edge of GTETRGD input	R/W*1
7	PSGTRGDF	GTETRGD Pin Falling Input Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTETRGD input 1: Counter stop enabled on the falling edge of GTETRGD input	R/W*1
8	PSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter stop enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	PSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter stop enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	PSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter stop enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	PSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter stop enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	PSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter stop enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	PSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter stop enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	PSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter stop enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	PSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter stop enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	PSELCA	ELC_GPTA Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTA input 1: Counter stop enabled at the ELC_GPTA input	R/W*1
17	PSELCB	ELC_GPTB Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTB input 1: Counter stop enabled at the ELC_GPTB input	R/W*1
18	PSELCC	ELC_GPTC Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTC input 1: Counter stop enabled at the ELC_GPTC input	R/W*1
19	PSELCD	ELC_GPTD Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTD input 1: Counter stop enabled at the ELC_GPTD input	R/W*1

Bit	Symbol	Function	R/W
20	PSELCE	ELC_GPTE Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTE input 1: Counter stop enabled at the ELC_GPTE input	R/W*1
21	PSELCF	ELC_GPTF Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTF input 1: Counter stop enabled at the ELC_GPTF input	R/W*1
22	PSELCG	ELC_GPTG Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTG input 1: Counter stop enabled at the ELC_GPTG input	R/W*1
23	PSELCH	ELC_GPTH Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTH input 1: Counter stop enabled at the ELC_GPTH input	R/W*1
30:24	—	These bits are read as 0. The write value should be 0.	R/W
31	CSTOP	Software Source Counter Stop Enable 0: Counter stop disabled by the GTSTP register 1: Counter stop enabled by the GTSTP register	R/W*1

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

The GTPSR sets the source to stop the GTCNT counter.

Inputs from GTETRGn (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

PSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Stop Enable)

The PSGTRGAR bit enables or disables the GTCNT counter stop on the rising edge of the GTETRGA pin input.

PSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Stop Enable)

The PSGTRGAF bit enables or disables the GTCNT counter stop on the falling edge of the GTETRGA pin input.

PSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Stop Enable)

PSGTRGBR bit enables or disables the GTCNT counter stop on the rising edge of the GTETRGB pin input.

PSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Stop Enable)

The PSGTRGBF bit enables or disables the GTCNT counter stop on the falling edge of the GTETRGB pin input.

PSGTRGCR bit (GTETRGC Pin Rising Input Source Counter Stop Enable)

PSGTRGCR bit enables or disables the GTCNT counter stop on the rising edge of the GTETRGC pin input.

PSGTRGCF bit (GTETRGC Pin Falling Input Source Counter Stop Enable)

The PSGTRGCF bit enables or disables the GTCNT counter stop on the falling edge of the GTETRGC pin input.

PSGTRGDR bit (GTETRGD Pin Rising Input Source Counter Stop Enable)

PSGTRGDR bit enables or disables the GTCNT counter stop on the rising edge of the GTETRGD pin input.

PSGTRGDF bit (GTETRGD Pin Falling Input Source Counter Stop Enable)

The PSGTRGDF bit enables or disables the GTCNT counter stop on the falling edge of the GTETRGD pin input.

PSCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Stop Enable)

The PSCARBL bit enables or disables the GTCNT counter stop on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 0.

In complementary PWM mode, this setting is invalid.

PSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Stop Enable)

The PSCARBH bit enables or disables the GTCNT counter stop on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

In complementary PWM mode, this setting is invalid.

PSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Stop Enable)

The PSCAFBL bit enables or disables the GTCNT counter stop on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

In complementary PWM mode, this setting is invalid.

PSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Stop Enable)

The PSCAFBH bit enables or disables the GTCNT counter stop on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

In complementary PWM mode, this setting is invalid.

PSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Stop Enable)

The PSCBRAL bit enables or disables the GTCNT counter stop on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

In complementary PWM mode, this setting is invalid.

PSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Stop Enable)

The PSCBRAH bit enables or disables the GTCNT counter stop on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

In complementary PWM mode, this setting is invalid.

PSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Stop Enable)

The PSCBFAL bit enables or disables the GTCNT counter stop on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

In complementary PWM mode, this setting is invalid.

PSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Stop Enable)

The PSCBFAH bit enables or disables the GTCNT counter stop on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

In complementary PWM mode, this setting is invalid.

PSELCm bit (ELCm Event Source Counter Stop Enable) (m = A to H)

The PSELCm bit enables or disables the GTCNT counter stop at the ELC_GPTm event input.

CSTOP bit (Software Source Counter Stop Enable)

The CSTOP bit enables or disables the GTCNT counter stop by the GTSTP register.

22.2.7 GTCSR : General PWM Timer Clear Source Select Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x18

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CCLR	—	—	—	CP1CCE	CSCMSC[2:0]			CSELCH	CSELCG	CSELCF	CSELCE	CSELCD	CSELCC	CSELCB	CSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CSCBFAH	CSCBFAL	CSCBRAH	CSCBRAL	CSCAFBH	CSCAFBL	CSCARBH	CSCARBL	CSGTRGDF	CSGTRGDR	CSGTRGCF	CSGTRGCR	CSGTRGBF	CSGTRGBR	CSGTRGAF	CSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CSGTRGAR	GTETRGA Pin Rising Input Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTETRGA input 1: Counter clear enabled on the rising edge of GTETRGA input	R/W*1
1	CSGTRGAF	GTETRGA Pin Falling Input Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTETRGA input 1: Counter clear enabled on the falling edge of GTETRGA input	R/W*1
2	CSGTRGBR	GTETRGB Pin Rising Input Source Counter Clear Enable 0: Disable counter clear on the rising edge of GTETRGB input 1: Enable counter clear on the rising edge of GTETRGB input	R/W*1
3	CSGTRGBF	GTETRGB Pin Falling Input Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTETRGB input 1: Counter clear enabled on the falling edge of GTETRGB input	R/W*1
4	CSGTRGCR	GTETRGC Pin Rising Input Source Counter Clear Enable 0: Disable counter clear on the rising edge of GTETRGC input 1: Enable counter clear on the rising edge of GTETRGC input	R/W*1
5	CSGTRGCF	GTETRGC Pin Falling Input Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTETRGC input 1: Counter clear enabled on the falling edge of GTETRGC input	R/W*1
6	CSGTRGDR	GTETRGD Pin Rising Input Source Counter Clear Enable 0: Disable counter clear on the rising edge of GTETRGD input 1: Enable counter clear on the rising edge of GTETRGD input	R/W*1
7	CSGTRGDF	GTETRGD Pin Falling Input Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTETRGD input 1: Counter clear enabled on the falling edge of GTETRGD input	R/W*1
8	CSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter clear enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	CSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter clear enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	CSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter clear enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	CSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter clear enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	CSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter clear enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	CSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter clear enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	CSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter clear enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W

Bit	Symbol	Function	R/W
15	CSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter clear enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	CSELCA	ELC_GPTA Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTA input 1: Counter clear enabled at the ELC_GPTA input	R/W*1
17	CSELCB	ELC_GPTB Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTB input 1: Counter clear enabled at the ELC_GPTB input	R/W*1
18	CSELCC	ELC_GPTC Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTC input 1: Counter clear enabled at the ELC_GPTC input	R/W*1
19	CSELCD	ELC_GPTD Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTD input 1: Counter clear enabled at the ELC_GPTD input	R/W*1
20	CSELCE	ELC_GPTE Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTE input 1: Counter clear enabled at the ELC_GPTE input	R/W*1
21	CSELCF	ELC_GPTF Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTF input 1: Counter clear enabled at the ELC_GPTF input	R/W*1
22	CSELCG	ELC_GPTG Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTG input 1: Counter clear enabled at the ELC_GPTG input	R/W*1
23	CSELCH	ELC_GPTH Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTH input 1: Counter clear enabled at the ELC_GPTH input	R/W*1
26:24	CSCMSC[2:0]	Compare Match/Input Capture/Synchronous counter clearing Source Counter Clear Enable*2 0 0 0: Counter clear disabled by Compare match/ Input capture/ Synchronous counter clearing group 0 0 1: Counter clear enabled at the GTCCRA register compare match/ Input capture 0 1 0: Counter clear enabled at the GTCCRB register compare match/ Input capture 0 1 1: Counter clear enabled at the GTCCRC register compare match 1 0 0: Counter clear enabled at the GTCCRD register compare match 1 0 1: Counter clear enabled at the GTCCRE register compare match 1 1 0: Counter clear enabled at the GTCCRF register compare match 1 1 1: Counter clear enabled at the synchronous counter clearing group	R/W
27	CP1CCE	Complementary PWM mode1 Crest Source Counter Clear Enable*2 0: Counter clear disabled at the crest of complementary PWM mode1 1: Counter clear enabled at the crest of complementary PWM mode1	R/W*1
30:28	—	These bits are read as 0. The write value should be 0.	R/W
31	CCLR	Software Source Counter Clear Enable 0: Counter clear disabled by the GTCLR register 1: Counter clear enabled by the GTCLR register	R/W*1

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

Note 2. This bit is only available in GPT324 to GPT329.
In GPT320 to GPT323 and GPT3210 to GPT3213, this bit is read as 0. The write value should be 0.

The GTCSR sets the source to clear the GTCNT counter.

Counter clearing can be executed whether the counter is running (GTCR.CST=1) or stopped (GTCR.CST=0).

Inputs from GTETRGn (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

CSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Clear Enable)

The CSGTRGAR bit enables or disables the GTCNT counter clear on the rising edge of the GTETRGA pin input.

CSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Clear Enable)

The CSGTRGAF bit enables or disables the GTCNT counter clear on the falling edge of the GTETRGA pin input.

CSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Clear Enable)

The CSGTRGBR bit enables or disables the GTCNT counter clear on the rising edge of the GTETRGB pin input.

CSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Clear Enable)

The CSGTRGBF bit enables or disables the GTCNT counter clear on the falling edge of the GTETRGB pin input.

CSGTRGCR bit (GTETRGC Pin Rising Input Source Counter Clear Enable)

The CSGTRGCR bit enables or disables the GTCNT counter clear on the rising edge of the GTETRGC pin input.

CSGTRGCF bit (GTETRGC Pin Falling Input Source Counter Clear Enable)

The CSGTRGCF bit enables or disables the GTCNT counter clear on the falling edge of the GTETRGC pin input.

CSGTRGDR bit (GTETRGD Pin Rising Input Source Counter Clear Enable)

The CSGTRGDR bit enables or disables the GTCNT counter clear on the rising edge of the GTETRGD pin input.

CSGTRGDF bit (GTETRGD Pin Falling Input Source Counter Clear Enable)

The CSGTRGDF bit enables or disables the GTCNT counter clear on the falling edge of the GTETRGD pin input.

CSCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Clear Enable)

The CSCARBL bit enables or disables the GTCNT counter clear on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 0.

In complementary PWM mode, this setting is invalid.

CSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Clear Enable)

The CSCARBH bit enables or disables the GTCNT counter clear on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

In complementary PWM mode, this setting is invalid.

CSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Clear Enable)

The CSCAFBL bit enables or disables the GTCNT counter clear on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

In complementary PWM mode, this setting is invalid.

CSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Clear Enable)

The CSCAFBH bit enables or disables the GTCNT counter clear on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

In complementary PWM mode, this setting is invalid.

CSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Clear Enable)

The CSCBRAL bit enables or disables the GTCNT counter clear on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

In complementary PWM mode, this setting is invalid.

CSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Clear Enable)

The CSCBRAH bit enables or disables the GTCNT counter clear on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

In complementary PWM mode, this setting is invalid.

CSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Clear Enable)

The CSCBFAL bit enables or disables the GTCNT counter clear on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

In complementary PWM mode, this setting is invalid.

CSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Clear Enable)

The CSCBFAH bit enables or disables the GTCNT counter clear on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

In complementary PWM mode, this setting is invalid.

CSELCm bit (ELCm Event Source Counter Clear Enable) (m = A to H)

The CSELCm bit enables or disables the GTCNT counter clear at the ELC_GPTm event input.

CSCMSC[2:0] bit (Compare Match/Input Capture/Synchronous counter clearing Source Counter Clear Enable*2)

Select enable or disable for the counter clear of the GTCNT counter by compare match/input capture/synchronous counter clearing group.

Since the compare match by the register that is performing the buffer operation (including the wave mode specific case) does not occur, the counter clear enable setting that makes the target register of the buffer operation the compare match factor is invalid.

In complementary PWM mode, the counter clear enable setting for compare match of the GTCCRB register, GTCCRE register, and GTCCRF register is invalid even when the buffer operation is not performed.

CP1CCE bit (Complementary PWM mode1 Crest Source Counter Clear Enable)

Select enable or disable for the counter clear at the crest of complementary PWM mode 1.

To enable this bit, do not set 1 to the PSYE bit of the GTIOR register.

It is valid only for the master channel in complementary PWM mode. The master channel setting also clears the GTCNT counter of the slave channel in complementary PWM mode.

CCLR bit (Software Source Counter Clear Enable)

The CCLR bit enables or disables the GTCNT counter clear by the GTCLR register.

22.2.8 GTUPSR : General PWM Timer Up Count Source Select Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x1C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	USILVL[3:0]				USELCH	USELCG	USELCF	USELCE	USELCD	USELCC	USELCB	USELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	USCBFAH	USCBFAL	USCBRAH	USCBRAL	USCAFBH	USCAFBL	USCARBH	USCARBL	USGTRGDF	USGTRGDR	USGTRGCF	USGTRGCR	USGTRGBF	USGTRGBR	USGTRGAF	USGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	USGTRGAR	GTETRGA Pin Rising Input Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTETRGA input 1: Counter count up enabled on the rising edge of GTETRGA input	R/W

Bit	Symbol	Function	R/W
1	USGTRGAF	GTETRGA Pin Falling Input Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTETRGA input 1: Counter count up enabled on the falling edge of GTETRGA input	R/W
2	USGTRGBR	GTETRGB Pin Rising Input Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTETRGB input 1: Counter count up enabled on the rising edge of GTETRGB input	R/W
3	USGTRGBF	GTETRGB Pin Falling Input Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTETRGB input 1: Counter count up enabled on the falling edge of GTETRGB input	R/W
4	USGTRGCR	GTETRGC Pin Rising Input Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTETRGC input 1: Counter count up enabled on the rising edge of GTETRGC input	R/W
5	USGTRGCF	GTETRGC Pin Falling Input Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTETRGC input 1: Counter count up enabled on the falling edge of GTETRGC input	R/W
6	USGTRGDR	GTETRGD Pin Rising Input Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTETRGD input 1: Counter count up enabled on the rising edge of GTETRGD input	R/W
7	USGTRGDF	GTETRGD Pin Falling Input Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTETRGD input 1: Counter count up enabled on the falling edge of GTETRGD input	R/W
8	USCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter count up enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	USCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter count up enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	USCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter count up enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	USCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter count up enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	USCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter count up enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	USCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter count up enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	USCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter count up enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W

Bit	Symbol	Function	R/W
15	USCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter count up enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	USELCA	ELC_GPTA Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTA input 1: Counter count up enabled at the ELC_GPTA input	R/W
17	USELCB	ELC_GPTB Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTB input 1: Counter count up enabled at the ELC_GPTB input	R/W
18	USELCC	ELC_GPTC Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTC input 1: Counter count up enabled at the ELC_GPTC input	R/W
19	USELCD	ELC_GPTD Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTD input 1: Counter count up enabled at the ELC_GPTD input	R/W
20	USELCE	ELC_GPTE Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTE input 1: Counter count up enabled at the ELC_GPTE input	R/W
21	USELCF	ELC_GPTF Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTF input 1: Counter count up enabled at the ELC_GPTF input	R/W
22	USELCG	ELC_GPTG Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTG input 1: Counter count up enabled at the ELC_GPTG input	R/W
23	USELCH	ELC_GPTH Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTH input 1: Counter count up enabled at the ELC_GPTH input	R/W
27:24	USILVL[3:0]	External Input Level Source Count-Up Enable 0 0 0 0: Disables count-up by external input level 0 0 0 1: Setting prohibited 0 0 1 0: Enables count-up by GTIOCnA pin input level 0 0 0 1 1: Enables count-up by GTIOCnA pin input level 1 0 1 0 0: Enables count-up by GTIOCnB pin input level 0 0 1 0 1: Enables count-up by GTIOCnB pin input level 1 0 1 1 0: Setting prohibited 0 1 1 1: Setting prohibited 1 0 0 0: Enables count-up by GTETRGA pin input level 0 1 0 0 1: Enables count-up by GTETRGA pin input level 1 1 0 1 0: Enables count-up by GTETRGB pin input level 0 1 0 1 1: Enables count-up by GTETRGB pin input level 1 1 1 0 0: Enables count-up by GTETRGC pin input level 0 1 1 0 1: Enables count-up by GTETRGC pin input level 1 1 1 1 0: Enables count-up by GTETRGD pin input level 0 1 1 1 1: Enables count-up by GTETRGD pin input level 1	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The GTUPSR sets the source to count up the GTCNT counter.

When at least one bit in the GTUPSR register is set to 1, the GTCNT counter is counted up by the source that is set to 1 in this register. In this case, GTCR.TPCS has no effect.

Number of increment in counting is one even when multiple sources are generated simultaneously.

Inputs from GTETRGn (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

USGTRGAR bit (GTETRGA Pin Rising Input Source Counter Count Up Enable)

The USGTRGAR bit enables or disables the GTCNT counter count up on the rising edge of the GTETRGA pin input.

USGTRGAF bit (GTETRGA Pin Falling Input Source Counter Count Up Enable)

The USGTRGAF bit enables or disables the GTCNT counter count up on the falling edge of the GTETRGA pin input.

USGTRGBR bit (GTETRGB Pin Rising Input Source Counter Count Up Enable)

The USGTRGBR bit enables or disables the GTCNT counter count up on the rising edge of the GTETRGB pin input.

USGTRGBF bit (GTETRGB Pin Falling Input Source Counter Count Up Enable)

The USGTRGBF bit enables or disables the GTCNT counter count up on the falling edge of the GTETRGB pin input.

USGTRGCR bit (GTETRGC Pin Rising Input Source Counter Count Up Enable)

The USGTRGCR bit enables or disables the GTCNT counter count up on the rising edge of the GTETRGC pin input.

USGTRGCF bit (GTETRGC Pin Falling Input Source Counter Count Up Enable)

The USGTRGCF bit enables or disables the GTCNT counter count up on the falling edge of the GTETRGC pin input.

USGTRGDR bit (GTETRGD Pin Rising Input Source Counter Count Up Enable)

The USGTRGDR bit enables or disables the GTCNT counter count up on the rising edge of the GTETRGD pin input.

USGTRGDF bit (GTETRGD Pin Falling Input Source Counter Count Up Enable)

The USGTRGDF bit enables or disables the GTCNT counter count up on the falling edge of the GTETRGD pin input.

USCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Count Up Enable)

The USCARBL bit enables or disables GTCNT counter count up on the rising edge of GTIOCnA pin input, when GTIOCnB input is 0.

USCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Count Up Enable)

The USCARBH bit enables or disables the GTCNT counter count up on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

USCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Count Up Enable)

The USCAFBL bit enables or disables the GTCNT counter count up on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

USCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Count Up Enable)

The USCAFBH bit enables or disables the GTCNT counter count up on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

USCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Count Up Enable)

The USCBRAL bit enables or disables the GTCNT counter count up on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

USCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Count Up Enable)

The USCBRAH bit enables or disables the GTCNT counter count up on the rising edge of the GTIOCnB pin input, when the GTIOCnA input is 1.

USCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Count Up Enable)

The USCBFAL bit enables or disables the GTCNT counter count up on the falling edge of the GTIOCnB pin input, when the GTIOCnA input is 0.

USCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Count Up Enable)

The USCBFAH bit enables or disables the GTCNT counter count up on the falling edge of the GTIOCnB pin input, when the GTIOCnA input is 1.

USELCm bit (ELC_GPTm Event Source Counter Count Up Enable) (m = A to H)

The USELCm bit enables or disables the GTCNT counter count up at the ELC_GPTm event input.

USILVL[3:0] bit (External Input Level Source Count-Up Enable)

Select enable or disable for the count-up of the GTCNT counter by the GTIOCnA pin input level, GTIOCnB pin input level, and GTETRGA/GTETRGB/GTETRGC/GTETRGD input level.

22.2.9 GTDNSR : General PWM Timer Down Count Source Select Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	DSILVL[3:0]				DSELCH	DSELCG	DSELCF	DSELCE	DSELCD	DSELCC	DSELCB	DSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DSCBFAH	DSCBFAL	DSCBRAH	DSCBRAL	DSCAFBH	DSCAFBL	DSCARBH	DSCARBL	DSGTRGDF	DSGTRGDR	DSGTRGCF	DSGTRGCR	DSGTRGBF	DSGTRGBR	DSGTRGAF	DSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DSGTRGAR	GTETRGA Pin Rising Input Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTETRGA input 1: Counter count down enabled on the rising edge of GTETRGA input	R/W
1	DSGTRGAF	GTETRGA Pin Falling Input Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTETRGA input 1: Counter count down enabled on the falling edge of GTETRGA input	R/W
2	DSGTRGBR	GTETRGB Pin Rising Input Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTETRGB input 1: Counter count down enabled on the rising edge of GTETRGB input	R/W
3	DSGTRGBF	GTETRGB Pin Falling Input Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTETRGB input 1: Counter count down enabled on the falling edge of GTETRGB input	R/W
4	DSGTRGCR	GTETRGC Pin Rising Input Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTETRGC input 1: Counter count down enabled on the rising edge of GTETRGC input	R/W
5	DSGTRGCF	GTETRGC Pin Falling Input Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTETRGC input 1: Counter count down enabled on the falling edge of GTETRGC input	R/W
6	DSGTRGDR	GTETRGD Pin Rising Input Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTETRGD input 1: Counter count down enabled on the rising edge of GTETRGD input	R/W
7	DSGTRGDF	GTETRGD Pin Falling Input Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTETRGD input 1: Counter count down enabled on the falling edge of GTETRGD input	R/W
8	DSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter count down enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	DSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter count down enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W

Bit	Symbol	Function	R/W
10	DSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter count down enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	DSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter count down enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	DSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter count down enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	DSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter count down enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	DSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter count down enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	DSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter count down enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	DSELCA	ELC_GPTA Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTA input 1: Counter count down enabled at the ELC_GPTA input	R/W
17	DSELCB	ELC_GPTB Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTB input 1: Counter count down enabled at the ELC_GPTB input	R/W
18	DSELCC	ELC_GPTC Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTC input 1: Counter count down enabled at the ELC_GPTC input	R/W
19	DSELCD	ELC_GPTD Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTD input 1: Counter count down enabled at the ELC_GPTD input	R/W
20	DSELCE	ELC_GPTE Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTE input 1: Counter count down enabled at the ELC_GPTE input	R/W
21	DSELCF	ELC_GPTF Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTF input 1: Counter count down enabled at the ELC_GPTF input	R/W
22	DSELCG	ELC_GPTG Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTG input 1: Counter count down enabled at the ELC_GPTG input	R/W

Bit	Symbol	Function	R/W
23	DSELCH	ELC_GPTF Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTF input 1: Counter count down enabled at the ELC_GPTF input	R/W
27:24	DSILVL[3:0]	External Input Level Source Count-Down Enable 0 0 0 0: Disables count-down by external input level 0 0 0 1: Setting prohibited 0 0 1 0: Enables count-down by GTIOCnA pin input level 0 0 0 1 1: Enables count-down by GTIOCnA pin input level 1 0 1 0 0: Enables count-down by GTIOCnB pin input level 0 0 1 0 1: Enables count-down by GTIOCnB pin input level 1 0 1 1 0: Setting prohibited 0 1 1 1: Setting prohibited 1 0 0 0: Enables count-down by GTETRGA pin input level 0 1 0 0 1: Enables count-down by GTETRGA pin input level 1 1 0 1 0: Enables count-down by GTETRGB pin input level 0 1 0 1 1: Enables count-down by GTETRGB pin input level 1 1 1 0 0: Enables count-down by GTETRGC pin input level 0 1 1 0 1: Enables count-down by GTETRGC pin input level 1 1 1 1 0: Enables count-down by GTETRGD pin input level 0 1 1 1 1: Enables count-down by GTETRGD pin input level 1	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The GTDNSR sets the source to count down the GTCNT counter.

When at least one bit in the GTDNSR register is set to 1, the GTCNT counter is counted down by the source that is set to 1 in this register. In this case, GTCR.TPCS has no effect.

Number of decrement in counting is one even when multiple sources are generated simultaneously.

Inputs from GTETRGn (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

DSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Count Down Enable)

The DSGTRGAR bit enables or disables the GTCNT counter count down on the rising edge of the GTETRGA pin input.

DSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Count Down Enable)

The DSGTRGAF bit enables or disables the GTCNT counter count down on the falling edge of the GTETRGA pin input.

DSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Count Down Enable)

The DSGTRGBR bit enables or disables the GTCNT counter count down on the rising edge of the GTETRGB pin input.

DSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Count Down Enable)

The DSGTRGBF bit enables or disables the GTCNT counter count down on the falling edge of the GTETRGB pin input.

DSGTRGCR bit (GTETRGC Pin Rising Input Source Counter Count Down Enable)

The DSGTRGCR bit enables or disables the GTCNT counter count down on the rising edge of the GTETRGC pin input.

DSGTRGCF bit (GTETRGC Pin Falling Input Source Counter Count Down Enable)

The DSGTRGCF bit enables or disables the GTCNT counter count down on the falling edge of the GTETRGC pin input.

DSGTRGDR bit (GTETRGD Pin Rising Input Source Counter Count Down Enable)

The DSGTRGDR bit enables or disables the GTCNT counter count down on the rising edge of the GTETRGD pin input.

DSGTRGDF bit (GTETRGD Pin Falling Input Source Counter Count Down Enable)

The DSGTRGDF bit enables or disables the GTCNT counter count down on the falling edge of the GTETRGD pin input.

DSCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Count Down Enable)

The DSCARBL bit enables or disables the GTCNT counter count down on the rising edge of the GTIOCnA pin input, when the GTIOCnB input is 0.

DSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Count Down Enable)

The DSCARBH bit enables or disables the GTCNT counter count down on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

DSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Count Down Enable)

The DSCAFBL bit enables or disables the GTCNT counter count down on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

DSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Count Down Enable)

The DSCAFBH bit enables or disables the GTCNT counter count down on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

DSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Count Down Enable)

The DSCBRAL bit enables or disables the GTCNT counter count down on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

DSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Count Down Enable)

The DSCBRAH bit enables or disables the GTCNT counter count down on the rising edge of GTIOCnB pin input, when GTIOCnA input is 1.

DSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Count Down Enable)

The DSCBFAL bit enables or disables the GTCNT counter count down on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

DSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Count Down Enable)

The DSCBFAH bit enables or disables the GTCNT counter count down on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

DSELCm bit (ELC_GPTm Event Source Counter Count Down Enable) (m = A to H)

The DSELCm bit enables or disables the GTCNT counter count down at the ELC_GPTm event input.

DSILVL[3:0] bit (External Input Level Source Count-Down Enable)

Select enable or disable for the count-down of the GTCNT counter by the GTIOCnA pin input level, GTIOCnB pin input level, and GTETRGA/GTETRGB/GTETRGC/GTETRGD input level.

22.2.10 GTICASR : General PWM Timer Input Capture Source Select Register A

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x24

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	ASOC	ASELCH	ASELCG	ASELCF	ASELCE	ASELCD	ASELCC	ASELCB	ASELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ASCBFAH	ASCBFAL	ASCBRAH	ASCBRAL	ASCAFBH	ASCAFBL	ASCARBH	ASCARBL	ASGTRGDF	ASGTRGDR	ASGTRGCF	ASGTRGCR	ASGTRGBF	ASGTRGBR	ASGTRGAF	ASGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ASGTRGAR	GTETRGA Pin Rising Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTETRGA input 1: GTCCRA input capture enabled on the rising edge of GTETRGA input	R/W
1	ASGTRGAF	GTETRGA Pin Falling Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTETRGA input 1: GTCCRA input capture enabled on the falling edge of GTETRGA input	R/W
2	ASGTRGBR	GTETRGB Pin Rising Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTETRGB input 1: GTCCRA input capture enabled on the rising edge of GTETRGB input	R/W
3	ASGTRGBF	GTETRGB Pin Falling Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTETRGB input 1: GTCCRA input capture enabled on the falling edge of GTETRGB input	R/W
4	ASGTRGCR	GTETRGC Pin Rising Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTETRGC input 1: GTCCRA input capture enabled on the rising edge of GTETRGC input	R/W
5	ASGTRGCF	GTETRGC Pin Falling Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTETRGC input 1: GTCCRA input capture enabled on the falling edge of GTETRGC input	R/W
6	ASGTRGDR	GTETRGD Pin Rising Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTETRGD input 1: GTCCRA input capture enabled on the rising edge of GTETRGD input	R/W
7	ASGTRGDF	GTETRGD Pin Falling Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTETRGD input 1: GTCCRA input capture enabled on the falling edge of GTETRGD input	R/W
8	ASCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: GTCCRA input capture enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	ASCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: GTCCRA input capture enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	ASCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: GTCCRA input capture enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W

Bit	Symbol	Function	R/W
11	ASCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: GTCCRA input capture enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	ASCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: GTCCRA input capture enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	ASCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: GTCCRA input capture enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	ASCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: GTCCRA input capture enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	ASCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: GTCCRA input capture enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	ASELCA	ELC_GPTA Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTA input 1: GTCCRA input capture enabled at the ELC_GPTA input	R/W
17	ASELCB	ELC_GPTB Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTB input 1: GTCCRA input capture enabled at the ELC_GPTB input	R/W
18	ASELCC	ELC_GPTC Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTC input 1: GTCCRA input capture enabled at the ELC_GPTC input	R/W
19	ASELCD	ELC_GPTD Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTD input 1: GTCCRA input capture enabled at the ELC_GPTD input	R/W
20	ASELCE	ELC_GPTE Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTE input 1: GTCCRA input capture enabled at the ELC_GPTE input	R/W
21	ASELCF	ELC_GPTF Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTF input 1: GTCCRA input capture enabled at the ELC_GPTF input	R/W
22	ASELCG	ELC_GPTG Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTG input 1: GTCCRA input capture enabled at the ELC_GPTG input	R/W
23	ASELCH	ELC_GPTH Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTH input 1: GTCCRA input capture enabled at the ELC_GPTH input	R/W
24	ASOC	Other channel Source GTCCRA Input Capture Enable*1 0: Disables GTCCRA input capture by other channel factor 1: Enables GTCCRA input capture by other channel factor	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit is only available in GPT324 to GPT329.

In GPT320 to GPT323 and GPT3210 to GPT3213, this bit is read as 0. The write value should be 0.

The GTICASR sets the source of input capture for GTCCRA.

When at least one bit among bits in the GTICASR register is set to 1, input capture operation making the GTCCRA register as an input capture register is performed.

Inputs from GTETRGn (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

ASGTRGAR bit (GTETRGA Pin Rising Input Source GTCCRA Input Capture Enable)

The ASGTRGAR bit enables or disables the input capture for GTCCRA on the rising edge of the GTETRGA pin input.

ASGTRGAF bit (GTETRGA Pin Falling Input Source GTCCRA Input Capture Enable)

The ASGTRGAF bit enables or disables the input capture for GTCCRA on the falling edge of the GTETRGA pin input.

ASGTRGBR bit (GTETRGB Pin Rising Input Source GTCCRA Input Capture Enable)

The ASGTRGBR bit enables or disables the input capture for GTCCRA on the rising edge of the GTETRGB pin input.

ASGTRGBF bit (GTETRGB Pin Falling Input Source GTCCRA Input Capture Enable)

The ASGTRGBF bit enables or disables the input capture for GTCCRA on the falling edge of the GTETRGB pin input.

ASGTRGCR bit (GTETRGC Pin Rising Input Source GTCCRA Input Capture Enable)

The ASGTRGCR bit enables or disables the input capture for GTCCRA on the rising edge of the GTETRGC pin input.

ASGTRGCF bit (GTETRGC Pin Falling Input Source GTCCRA Input Capture Enable)

The ASGTRGCF bit enables or disables the input capture for GTCCRA on the falling edge of the GTETRGC pin input.

ASGTRGDR bit (GTETRGD Pin Rising Input Source GTCCRA Input Capture Enable)

The ASGTRGDR bit enables or disables the input capture for GTCCRA on the rising edge of the GTETRGD pin input.

ASGTRGDF bit (GTETRGD Pin Falling Input Source GTCCRA Input Capture Enable)

The ASGTRGDF bit enables or disables the input capture for GTCCRA on the falling edge of the GTETRGD pin input.

ASCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source GTCCRA Input Capture Enable)

The ASCARBL bit enables or disables the input capture for GTCCRA on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 0.

ASCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source GTCCRA Input Capture Enable)

The ASCARBH bit enables or disables the input capture for GTCCRA on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

ASCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source GTCCRA Input Capture Enable)

The ASCAFBL bit enables or disables the input capture for GTCCRA on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

ASCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source GTCCRA Input Capture Enable)

The ASCAFBH bit enables or disables the input capture for GTCCRA on the falling edge of the GTIOCnA pin input, when the GTIOCnB input is 1.

ASCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source GTCCRA Input Capture Enable)

The ASCBRAL bit enables or disables the input capture for GTCCRA on the rising edge of the GTIOCnB pin input, when the GTIOCnA input is 0.

ASCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source GTCCRA Input Capture Enable)

The ASCBRAH bit enables or disables the input capture for GTCCRA on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

ASCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source GTCCRA Input Capture Enable)

The ASCBFAL bit enables or disables the input capture for GTCCRA on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

ASCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source GTCCRA Input Capture Enable)

The ASCBFAH bit enables or disables the input capture for GTCCRA on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

ASELCm bit (ELC_GPTm Event Source Counter GTCCRA Input Capture Enable) (m = A to H)

The ASELCm bit enables or disables the input capture for GTCCRA at the ELC_GPTm event input.

ASOC bit (Other channel Source GTCCRA Input Capture Enable)

Select enable or disable for an input capture to the GTCCRA register by other channel factor.

Input capture of other channel factors is not subject to input capture factors to other channels set by the ICAFA and ICBFA bits of the GTICCR register.

22.2.11 GTICBSR : General PWM Timer Input Capture Source Select Register B

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x28

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	BSOC	BSELCH	BSELCG	BSELCF	BSELCE	BSELCD	BSELCC	BSELCB	BSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSCBFAH	BSCBFAL	BSCBRAH	BSCBRAL	BSCAFBH	BSCAFBL	BSCARBH	BSCARBL	BSGTRGDF	BSGTRGDR	BSGTRGCF	BSGTRGCR	BSGTRGBF	BSGTRGBR	BSGTRGAF	BSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BSGTRGAR	GTETRGA Pin Rising Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTETRGA input 1: GTCCRB input capture enabled on the rising edge of GTETRGA input	R/W
1	BSGTRGAF	GTETRGA Pin Falling Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTETRGA input 1: GTCCRB input capture enabled on the falling edge of GTETRGA input	R/W
2	BSGTRGBR	GTETRGB Pin Rising Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTETRGB input 1: GTCCRB input capture enabled on the rising edge of GTETRGB input	R/W
3	BSGTRGBF	GTETRGB Pin Falling Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTETRGB input 1: GTCCRB input capture enabled on the falling edge of GTETRGB input	R/W
4	BSGTRGCR	GTETRGC Pin Rising Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTETRGC input 1: GTCCRB input capture enabled on the rising edge of GTETRGC input	R/W

Bit	Symbol	Function	R/W
5	BSGTRGCF	GTETRGC Pin Falling Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTETRGC input 1: GTCCRB input capture enabled on the falling edge of GTETRGC input	R/W
6	BSGTRGDR	GTETRGD Pin Rising Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTETRGD input 1: GTCCRB input capture enabled on the rising edge of GTETRGD input	R/W
7	BSGTRGDF	GTETRGD Pin Falling Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTETRGD input 1: GTCCRB input capture enabled on the falling edge of GTETRGD input	R/W
8	BSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: GTCCRB input capture enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	BSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: GTCCRB input capture enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	BSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: GTCCRB input capture enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	BSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: GTCCRB input capture enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	BSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: GTCCRB input capture enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	BSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: GTCCRB input capture enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	BSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: GTCCRB input capture enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	BSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: GTCCRB input capture enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	BSELCA	ELC_GPTA Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTA input 1: GTCCRB input capture enabled at the ELC_GPTA input	R/W

Bit	Symbol	Function	R/W
17	BSELCB	ELC_GPTB Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTB input 1: GTCCRB input capture enabled at the ELC_GPTB input	R/W
18	BSELCC	ELC_GPTC Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTC input 1: GTCCRB input capture enabled at the ELC_GPTC input	R/W
19	BSELCD	ELC_GPTD Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTD input 1: GTCCRB input capture enabled at the ELC_GPTD input	R/W
20	BSELCE	ELC_GPTE Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTE input 1: GTCCRB input capture enabled at the ELC_GPTE input	R/W
21	BSELCF	ELC_GPTF Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTF input 1: GTCCRB input capture enabled at the ELC_GPTF input	R/W
22	BSELCG	ELC_GPTG Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTG input 1: GTCCRB input capture enabled at the ELC_GPTG input	R/W
23	BSELCH	ELC_GPTH Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTH input 1: GTCCRB input capture enabled at the ELC_GPTH input	R/W
24	BSOC	Other channel Source GTCCRB Input Capture Enable*1 0: Disables GTCCRB input capture by other channel factor 1: Enables GTCCRB input capture by other channel factor	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit is only available in GPT324 to GPT329.
In GPT320 to GPT323 and GPT3210 to GPT3213, this bit is read as 0. The write value should be 0.

The GTICBSR sets the source of input capture for GTCCRB.

When at least one bit among bits in the GTICBSR register is set to 1, input capture operation making the GTCCRB register as an input capture register is performed.

Inputs from GTETRGn (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

BSGTRGAR bit (GTETRGA Pin Rising Input Source GTCCRB Input Capture Enable)

The BSGTRGAR bit enables or disables the input capture for GTCCRB on the rising edge of the GTETRGA pin input.

BSGTRGAF bit (GTETRGA Pin Falling Input Source GTCCRB Input Capture Enable)

The BSGTRGAF bit enables or disables the input capture for GTCCRB on the falling edge of the GTETRGA pin input.

BSGTRGBR bit (GTETRGB Pin Rising Input Source GTCCRB Input Capture Enable)

The BSGTRGBR bit enables or disables the input capture for GTCCRB on the rising edge of GTETRGB pin input.

BSGTRGBF bit (GTETRGB Pin Falling Input Source GTCCRB Input Capture Enable)

The BSGTRGBF bit enables or disables the input capture for GTCCRB on the falling edge of the GTETRGB pin input.

BSGTRGCR bit (GTETRGC Pin Rising Input Source GTCCRB Input Capture Enable)

The BSGTRGCR bit enables or disables the input capture for GTCCRB on the rising edge of GTETRGC pin input.

BSGTRGCF bit (GTETRGC Pin Falling Input Source GTCCRB Input Capture Enable)

The BSGTRGCF bit enables or disables the input capture for GTCCRB on the falling edge of the GTETRGC pin input.

BSGTRGDR bit (GTETRGD Pin Rising Input Source GTCCRB Input Capture Enable)

The BSGTRGDR bit enables or disables the input capture for GTCCRB on the rising edge of GTETRGD pin input.

BSGTRGDF bit (GTETRGD Pin Falling Input Source GTCCRB Input Capture Enable)

The BSGTRGDF bit enables or disables the input capture for GTCCRB on the falling edge of the GTETRGD pin input.

BSCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source GTCCRB Input Capture Enable)

The BSCARBL bit enables or disables the input capture for GTCCRB on the rising edge of the GTIOCnA pin input, when the GTIOCnB input is 0.

BSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source GTCCRB Input Capture Enable)

The BSCARBH bit enables or disables the input capture for GTCCRB on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

BSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source GTCCRB Input Capture Enable)

The BSCAFBL bit enables or disables the input capture for GTCCRB on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

BSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source GTCCRB Input Capture Enable)

The BSCAFBH bit enables or disables the input capture for GTCCRB on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

BSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source GTCCRB Input Capture Enable)

The BSCBRAL bit enables or disables the input capture for GTCCRB on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

BSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source GTCCRB Input Capture Enable)

The BSCBRAH bit enables or disables the input capture for GTCCRB on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

BSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source GTCCRB Input Capture Enable)

The BSCBFAL bit enables or disables the input capture for GTCCRB on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

BSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source GTCCRB Input Capture Enable)

The BSCBFAH bit enables or disables the input capture for GTCCRB on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

BSELCm bit (ELC_GPTm Event Source Counter GTCCRB Input Capture Enable) (m = A to H)

The BSELCm bit enables or disables the input capture for GTCCRB at the ELC_GPTm event input.

BSOC bit (Other channel Source GTCCRB Input Capture Enable)

Select enable or disable for an input capture to the GTCCRB register by other channel factor.

Input capture of other channel factors is not subject to input capture factors to other channels set by the ICAFB and ICBFB bits of the GTICCR register.

22.2.12 GTCR : General PWM Timer Control Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x2C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	CKEG[1:0]		TPCS[3:0]				—	—	—	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SSCEN	—	—	CPSCD	SSCGRP[1:0]		SCGTIOC	ICDS	—	—	BINV	AINV	—	—	—	CST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CST	Count Start 0: Count operation is stopped 1: Count operation is performed	R/W*1
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	AINV	GTIOCnA input/output pin polarity reversal control 0: Not invert 1: Invert	R/W
5	BINV	GTIOCnB input/output pin polarity reversal control 0: Not invert 1: Invert	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	ICDS	Input Capture Operation Select During Count Stop 0: Input capture is operated during count stop. 1: Input capture is not operated during count stop.	R/W
9	SCGTIOC	GTIOC input Source Synchronous Clear Enable*2 0: Disables to use the counter clear by GTIOC input as the clear factor for other channels 1: Enables to use the counter clear by GTIOC input as the clear factor for other channels	R/W
11:10	SSCGRP[1:0]	Synchronous Set/Clear Group Select*2 0 0: Select synchronous set/clear group A 0 1: Select synchronous set/clear group B 1 0: Select synchronous set/clear group C 1 1: Select synchronous set/clear group D	R/W*1
12	CPSCD	Complementary PWM Mode Synchronous Clear Disable*2 0: Enable synchronous counter clear by other channel other than the section of trough in complementary PWM mode 1: Disable synchronous counter clear by other channel other than the section of trough in complementary PWM mode	R/W*1
14:13	—	These bits are read as 0. The write value should be 0.	R/W
15	SSCEN	Synchronous Set/Clear Enable*2 0: Disable Synchronous set/clear of the GTCNT counter 1: Enable Synchronous set/clear of the GTCNT counter	R/W*1

Bit	Symbol	Function	R/W
19:16	MD[3:0]	Mode Select*3 0 0 0 0: Saw-wave PWM mode 1 (single buffer or double buffer possible) 0 0 0 1: Saw-wave one-shot pulse mode (fixed buffer operation) 0 0 1 0: Saw-wave PWM mode 2 (single buffer or double buffer possible) 0 0 1 1: Setting prohibited 0 1 0 0: Triangle-wave PWM mode 1 (32-bit transfer at trough) (single buffer or double buffer possible) 0 1 0 1: Triangle-wave PWM mode 2 (32-bit transfer at crest and trough) (single buffer or double buffer possible) 0 1 1 0: Triangle-wave PWM mode 3 (64-bit transfer at trough) (fixed buffer operation) 0 1 1 1: Setting prohibited 1 0 0 0: Setting prohibited 1 0 0 1: Setting prohibited 1 0 1 0: Setting prohibited 1 0 1 1: Setting prohibited 1 1 0 0: Complementary PWM mode 1 (transfer at crest) 1 1 0 1: Complementary PWM mode 2 (transfer at trough) 1 1 1 0: Complementary PWM mode 3 (transfer at crest and trough) 1 1 1 1: Complementary PWM mode 4 (immediate transfer)	R/W*1
22:20	—	These bits are read as 0. The write value should be 0.	R/W
26:23	TPCS[3:0]	Timer Prescaler Select 0 0 0 0: GTCLK/1 0 0 0 1: GTCLK/2 0 0 1 0: GTCLK/4 0 0 1 1: GTCLK/8 0 1 0 0: GTCLK/16 0 1 0 1: GTCLK/32 0 1 1 0: GTCLK/64 0 1 1 1: GTCLK/128 1 0 0 0: GTCLK/256 1 0 0 1: GTCLK/512 1 0 1 0: GTCLK/1024 1 0 1 1: Setting prohibited 1 1 0 0: GTETRGA (Via the POEG) 1 1 0 1: GTETRGB (Via the POEG) 1 1 1 0: GTETRGC (Via the POEG) 1 1 1 1: GTETRGD (Via the POEG)	R/W*1
28:27	CKEG[1:0]	Clock Edge Select 0 0: Select rising edge of GTETRG for clock count 0 1: Select falling edge of GTETRG for clock count Others: Select both edge of GTETRG for clock count	R/W*1
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

Note 2. This bit is only available in GPT324 to GPT329.
In GPT320 to GPT323 and GPT3210 to GPT3213, this bit is read as 0. The write value should be 0.

Note 3. MD[3] bit is only available in GPT324 to GPT329. GPT320 to GPT323 and GPT3210 to GPT3213 only support Saw-wave PWM mode and Triangle-wave PWM mode except Saw-wave PWM mode 2.

The GTCR controls GTCNT.

Access in 8 bit unit to GTCR is prohibited.

CST bit (Count Start)

The CST bit controls the GTCNT counter start and stop.

[Setting conditions]

- The GTSTR value where the channel number associated with the bit number is set to 1 with the GTSSR.CSTRT bit at 1
- The ELC event input, the external trigger, or the GTIOCnA/GTIOCnB input that are enabled by GTSSR for the starting counter source, occurs (n = 0 to 13)
- 1 is written by software directly.

[Clearing conditions]

- The GTSTP value where the channel number associated with the bit number is set to 1 with the GTPSR.CSTOP bit at 1
- The ELC event input, the external trigger, or the GTIOCnA/GTIOCnB input enabled by GTPSR as the counter stop source, occurs (n = 0 to 13)
- 0 is written by software directly.
- When the period count function is finished while the GTPC.ASTP bit is (n = 0 to 3, 10 to 13).

AINV bit (GTIOCnA input/output pin polarity reversal control)

This bit selects GTIOCnA input/output pin polarity reversal control.

BINV bit (GTIOCnB input/output pin polarity reversal control)

This bit selects GTIOCnB input/output pin polarity reversal control.

ICDS bit (Input Capture Operation Select During Count Stop)

This bit selects input capture operation during count stop, when the input capture function is selected.

SCGTIOC bit (GTIOC input Source Synchronous Clear Enable)

Select enable or disable the use of the counter clear by GTIOCnA/GTIOCnB input pin selected by the GTCSR register as the counter clear factor for other channels.

SSCGRP[1:0] bits (Synchronous Set/Clear Group Select)

Select the channel group of synchronous set/clear.

In complementary PWM mode, slave channels are also controlled by setting the SSCGRP[1:0] bits of the master channel.

CPSCD bit (Complementary PWM Mode Synchronous Clear Disable)

Select disable or enable of counter clear when synchronous clear from other channel occurs except section of trough in complementary PWM mode.

The slave channel is also controlled by setting the CPSCD bit of the master channel.

SSCEN bit (Synchronous Set/Clear Enable)

Select disable or enable of Synchronous set/clear.

In complementary PWM mode, slave channels are also controlled by setting the SSCEN bits of the master channel.

MD[3:0] bits (Mode Select)

These bits select the GPT operating mode.

In complementary PWM mode, slave channels are also controlled by setting the MD bits of the master channel.

Only the MD[3:2] bit (MD[2] for GPT320 to GPT323) is valid at input capture. Counting in saw-wave mode is performed with 00 for the MD[3:2] bit (0 for the MD[2] of GPT320 to GPT323), and counting in triangle-wave mode is performed with 01 for the MD[3:2] bit (1 for MD[2] of GPT320 to GPT323), and counting in complementary PWM mode is performed with 1 for the MD[3].

The MD bits must be set while the GTCNT operation is stopped.

During the event count operation (when at least one bit among the bits for the GTUPSR and GTDNSR registers is set to 1), setting of the MD bits are ignored, where counting in saw-wave or triangle-wave or complementary PWM modes is not performed. Instead, up-counting or down-counting by a source set by the GTUPSR and GTDNSR registers is performed.

TPCS[3:0] bits (Timer Prescaler Select)

The TPCS[3:0] bits select the clock for GTCNT. A clock prescaler can be selected independently for each channel. The TPCS[3:0] bits must be set while the GTCNT operation is stopped.

CKEG[1:0] bits (Clock Edge Select)

When GTETRG input is selected by TPCS[3:0] bits, select the edge of GTETRG input used as the clock of GTCNT counter.

Set the CKEG[1:0] bits only when the GTCNT counter operation is stopped.

22.2.13 GTUDDTYC : General PWM Timer Count Direction and Duty Setting Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x30

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	OABDTYT	OBDTYR	OBDTYF	OBDTY[1:0]		—	—	—	—	OADTYR	OADTYF	OADTY[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	UD	Count Direction Setting 0: GTCNT counts down 1: GTCNT counts up	R/W
1	UDF	Forcible Count Direction Setting 0: Not forcibly set 1: Forcibly set	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
17:16	OADTY[1:0]	GTIOCnA Output Duty Setting 0 0: GTIOCnA pin duty depends on the compare match 0 1: GTIOCnA pin duty depends on the compare match 1 0: GTIOCnA pin duty 0% 1 1: GTIOCnA pin duty 100%	R/W
18	OADTYF	Forcible GTIOCnA Output Duty Setting 0: Not forcibly set 1: Forcibly set	R/W
19	OADTYR	GTIOCnA Output Value Selecting after Releasing 0%/100% Duty Setting 0: The function selected by the GTIOA[3:2] bits is applied to the output value when the duty cycle is set after release from the 0 or 100% duty-cycle setting. 1: The function selected by the GTIOA[3:2] bits is applied to the compare match output value which is masked after release from the 0 or 100% duty-cycle setting.	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
25:24	OBDTY[1:0]	GTIOCnB Output Duty Setting 0 0: GTIOCnB pin duty depends on the compare match 0 1: GTIOCnB pin duty depends on the compare match 1 0: GTIOCnB pin duty 0% 1 1: GTIOCnB pin duty 100%	R/W
26	OBDTYF	Forcible GTIOCnB Output Duty Setting 0: Not forcibly set 1: Forcibly set	R/W
27	OBDTYR	GTIOCnB Output Value Selecting after Releasing 0%/100% Duty Setting 0: The function selected by the GTIOB[3:2] bits is applied to the output value when the duty cycle is set after release from the 0 or 100% duty-cycle setting. 1: The function selected by the GTIOB[3:2] bits is applied to the compare match output value which is masked after release from the 0 or 100% duty-cycle setting.	R/W
28	OABDTYT	GTIOCnA, B pin output 0%/100% duty setting reflection timing setting 0: The output duty-cycle setting changed at an underflow is reflected. 1: The output duty-cycle setting changed at an underflow and at the top of triangle-wave is reflected. The setting value "1" is invalid except the triangle-wave PWM mode.	R/W

Bit	Symbol	Function	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

The GTUDDTYC sets the direction in which the GTCNT counts (up-counting or down-counting), and sets the duty of the GTIOCnA/GTIOCnB pin output.

The setting is invalid during the event count operation, Saw-wave PWM mode 2, Complementary PWM mode.

Count Direction:

- In saw-wave mode.
  When the UD value is set to 0 during up-counting, the count direction changes at an overflow (the timing synchronous with count clock after the GTCNT value becomes the GTPR value). When the UD value is set to 1 during down-counting, the count direction changes at an underflow (the timing synchronous with count clock after the GTCNT value becomes 0).
  When the UD value changes from 1 to 0 with the UDF bit being 0 and while counting stops, the counter starts up-counting and the count direction changes at an overflow (the timing synchronous with count clock after the GTCNT value becomes the GTPR value). When the UD value changes from 0 to 1 with the UDF bit being 0 and while counting stops, the counter starts down-counting and the count direction changes at an underflow (the timing synchronous with count clock after the GTCNT value becomes 0).
  When the UDF bit is set to 1 while counting stops, the UD bit value is reflected in the count direction when counting starts.
- In triangle-wave mode.
  When the UD value changes during counting, the count direction does not change. When the UD value changes while the UDF bit is 0 and counting stops, the change is not reflected in the count direction when counting starts.
  When the UDF bit is set to 1 while counting is stopped, the UD value is reflected in the count direction when counting starts.

UD bit (Count Direction Setting)

The UD bit sets the count direction (up-counting or down-counting) for GTCNT.

UDF bit (Forcible Count Direction Setting)

The UDF bit forcibly sets the count direction when GTCNT starts operation as the UD value. Only 0 should be written to this bit during counter operation. When 1 is written to this bit while counting stops, return this bit to 0 before counting starts.

Output duty

- In saw-wave mode.
  When the OADTY/OBDTY value changes during up-counting, the duty is reflected at an overflow (GTCNT = GTPR). When the OADTY/OBDTY value is changed during down-counting, the duty is reflected at an underflow (GTCNT = 0).
  When the OADTY/OBDTY value is changed with the OADTYF/OBDTYF bit being 0 and while counting stops the output duty is not reflected at the starting counter operation. When the count direction is up, the output duty is reflected at an overflow (GTCNT = GTPR). When the count direction is down, the output duty is reflected at an underflow (GTCNT = 0).
  When the OADTY/OBDTY value is changed with the OADTYF/OBDTYF bit being 1 and while counting stops, the output duty is reflected at starting counter operation.
- In triangle-wave mode.
  When the OADTY/OBDTY value changes during counting, the duty is reflected at an underflow.
  When the OADTY/OBDTY value is changed with the OADTYF/OBDTYF bit being 0 and while counting stops, the output duty is not reflected at the starting counter operation. The output duty is reflected at an underflow.
  When the OADTY/OBDTY value is changed with the OADTYF/OBDTYF bit being 1 and while counting stops, the output duty is reflected at starting counter operation.

In both saw-wave mode and triangle-wave mode, when the OADTYF/OBDTYF bit is set back to 0 and the OADTY[1:0]/OBDTY[1:0] bits are set after setting the OADTYF/OBDTYF bit to 1 and setting the OADTY[1:0]/OBDTY[1:0] bits for the duty of first cycle while count operation is stopped, these duty-cycle set during stopping count operation are reflected in the first cycle and the second cycle after starting count operation.

OmDTY[1:0] bits (GTIOCnm Output Duty Setting) (m = A, B)

The OmDTY[1:0] bits set the output duty (0%, 100% or compare match control) of the GTIOCnm pin.

OmDTYF bit (Forcible GTIOCnm Output Duty Setting) (m = A, B)

The OmDTYF bit forcibly sets the output duty cycle to the OmDTY setting. Set this bit to 0 during counter operation.

OmDTYR bit (GTIOCnm Output Value Selecting after Releasing 0%/100% Duty Setting) (m = A, B)

The OmDTYR bit selects the value that is the object of output retained or toggled at cycle end, when the control changes from 0% or 100% duty setting to compare match for the GTIOCnm pin and GTIOR.GTIOm[3:2] bits are set to 00b (output retained at cycle end) or the GTIOR.GTIOm[3:2] bits are set to 11b (output toggled at cycle end).

The GPT internally continues to perform compare match operation during duty-cycle 0% or 100% operation. When the OmDTYR bit is 1, the value after the period has elapsed due this compare match operation is target for the GTIOm[3:2] bits.

OABDTYT Bit (GTIOCnm pin output 0%/100% duty setting reflection timing setting) (n = 0 − (value of P_NUMCHALL − 1), m = A, B)

Select the reflection timing of the 0%/100% duty setting for the GTIOCnm pin in triangle-wave mode.

If the OABDTYT bit is 0, the output duty setting set in the GTUDDTYC.OADTY[1:0] and OBDTY[1:0] bits is reflected when an underflow occurs.

If the OABDTYT bit is “1”, the output duty setting set in the GTUDDTYC.OADTY[1:0] and OBDTY[1:0] bits is reflected when underflow occurs and at the top of the triangle-wave.

22.2.14 GTIOR : General PWM Timer I/O Control Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x34

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	NFCSB[1:0]		NFBEN	—	OBEOCD	OBDF[1:0]		OBE	OBHLD	OBDFLT	—	GTIOB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	NFCSA[1:0]		NFAEN	PSYE	OAEOCD	OADF[1:0]		OAE	OAHLD	OADFLT	CPSCIR	GTIOA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	GTIOA[4:0]	GTIOCnA Pin Function Select See Table 22.4 and Table 22.5.	R/W
5	CPSCIR*1	Complementary PWM Mode Initial Output at Synchronous Clear Disable 0: Output the initial value set by the GTIOR.GTIOA and GTIOB bits when synchronous clear occurs in Trough section of complementary PWM mode 1: Disable output the initial value	R/W
6	OADFLT	GTIOCnA Pin Output Value Setting at the Count Stop 0: The GTIOCnA pin outputs low when counting stops 1: The GTIOCnA pin outputs high when counting stops	R/W
7	OAHLD	GTIOCnA Pin Output Setting at the Start/Stop Count 0: The GTIOCnA pin output level at the start or stop of counting depends on the register setting 1: The GTIOCnA pin output level is retained at the start or stop of counting	R/W
8	OAE	GTIOCnA Pin Output Enable 0: Output is disabled 1: Output is enabled	R/W

Bit	Symbol	Function	R/W
10:9	OADF[1:0]	GTIOCnA Pin Disable Value Setting 0 0: None of the below options are specified 0 1: GTIOCnA pin is set to Hi-Z in response to controlling the output negation 1 0: GTIOCnA pin is set to 0 in response to controlling the output negation 1 1: GTIOCnA pin is set to 1 in response to controlling the output negation	R/W
11	OAEOCD*1	GTCCRA Compare Match Cycle End Output Invalidate 0: Validate GTIOA[3:2] setting 1: Invalidate GTIOA[3:2] setting (GTIOCnA pin output is retained)	R/W
12	PSYE	PWM Synchronous output Enable 0: Disable GTCPPOm pin output 1: Enable GTCPPOm pin output	R/W
13	NFAEN	Noise Filter A Enable 0: The noise filter for the GTIOCnA pin is disabled 1: The noise filter for the GTIOCnA pin is enabled	R/W
15:14	NFCSA[1:0]	Noise Filter A Sampling Clock Select 0 0: GTCLK/1 0 1: GTCLK/4 1 0: GTCLK/16 1 1: GTCLK/64	R/W
20:16	GTIOB[4:0]	GTIOCnB Pin Function Select See Table 22.4 and Table 22.5.	R/W
21	—	This bit is read as 0. The write value should be 0.	R/W
22	OBDFLT	GTIOCnB Pin Output Value Setting at the Count Stop 0: The GTIOCnB pin outputs low when counting stops 1: The GTIOCnB pin outputs high when counting stops	R/W
23	OBHLD	GTIOCnB Pin Output Setting at the Start/Stop Count 0: The GTIOCnB pin output level at the start/stop of counting depends on the register setting 1: The GTIOCnB pin output level is retained at the start/stop of counting	R/W
24	OBE	GTIOCnB Pin Output Enable 0: Output is disabled 1: Output is enabled	R/W
26:25	OBDF[1:0]	GTIOCnB Pin Disable Value Setting 0 0: None of the below options are specified 0 1: GTIOCnB pin is set to Hi-Z in response to controlling the output negation 1 0: GTIOCnB pin is set to 0 in response to controlling the output negation 1 1: GTIOCnB pin is set to 1 in response to controlling the output negation	R/W
27	OBEOCD*1	GTCCRB Compare Match Cycle End Output Invalidate 0: When Saw-wave PWM mode 1, validate GTIOB[3:2] setting When Saw-wave PWM mode 2, validate GTIOA[3:2] setting 1: When Saw-wave PWM mode 1, invalidate GTIOB[3:2] setting (GTIOCnB pin output is retained) When Saw-wave PWM mode 2, invalidate GTIOA[3:2] setting (GTIOCnA pin output is retained)	R/W
28	—	This bit is read as 0. The write value should be 0.	R/W
29	NFBEN	Noise Filter B Enable 0: The noise filter for the GTIOCnB pin is disabled 1: The noise filter for the GTIOCnB pin is enabled	R/W
31:30	NFCSB[1:0]	Noise Filter B Sampling Clock Select 0 0: GTCLK/1 0 1: GTCLK/4 1 0: GTCLK/16 1 1: GTCLK/64	R/W

Note 1. This bit is only available in GPT324 to GPT329.
In GPT320 to GPT323 and GPT3210 to GPT3213, this bit is read as 0. The write value should be 0.

The GTIOR sets the functions of the GTIOCnA, GTIOCnB and GTCPPOm pins. (n = 0 to 13, m = 0 to 13)

GTIOA[4:0] bits (GTIOCnA Pin Function Select)

The GTIOA[4:0] bits select the GTIOCnA pin function. For details, see Table 22.4 and Table 22.5.

CPSCIR bit (Complementary PWM Mode Initial Output at Synchronous Clear Disable)

Select the output waveform when synchronous clear occurs in complementary PWM mode.

The initial output is disabled by this function only when synchronous clear occurs in the trough section in complementary PWM mode. If a synchronous clear occurs at any other time, the initial value set by the GTIOA[4]/GTIOB[4] bits is output regardless of the CPSCIR bit setting. In addition, the initial value set by the GTIOA[4]/GTIOB[4] bits is output even when the synchronous clear occurs in the trough section immediately after the count starts.

OADFLT bit (GTIOCnA Pin Output Value Setting at the Count Stop)

The OADFLT bit sets whether the GTIOCnA pin outputs high or low when counting stops.

OAHLD bit (GTIOCnA Pin Output Setting at the Start/Stop Count)

The OAHLD bit specifies whether the GTIOCnA pin output level is retained or the level at the start or stop of counting depends on the register setting.

When the OAHLD bit is set to 0:

- The value specified in bit [4] of the GTIOA[4:0] bits is output when counting starts
- The value specified in the OADFLT bit is output when counting stops
- If the OADFLT bit is modified while counting stops, the new value is immediately reflected in the output.

When the OAHLD bit is set to 1:

- The output is retained when counting starts or stops.

OAE bit (GTIOCnA Pin Output Enable)

The OAE bit disables or enables the GTIOCnA pin output.

When GTCCRA register is used as the input capture register (at least one bit in the GTICASR register is set to 1), the GTIOCnA pin does not output regardless of the OAE bit value.

OADF[1:0] bits (GTIOCnA Pin Disable Value Setting)

The OADF[1:0] bits select the output value of the GTIOCnA pin in response to a request to disable output from the POEG.

OAEOCD bit (GTCCRA Compare Match Cycle End Output Invalidate)

If the cycle end matches the GTCCRA compare match timing in Saw-wave PWM mode 1 and 2, select invalid/valid of the setting of GTIOA[3:2] bits. When 1 (disabled) is set, the GTIOCnA pin holds the output when the cycle end and the GTCCRA compare match timing match.

PSYE bit (PWM Synchronous output Enable)

This bit set Enable/disable of output signal from GTCPPOm pin synchronized with the PWM cycle that toggles at the crest/trough/GTCNT counter clear of complementary PWM mode and Triangle-wave mode or the end of the cycle of the Saw-wave mode.

The initial output of the GTCPPOn output pin is Low, and the count start results in High.

NFAEN bit (Noise Filter A Enable)

The NFAEN bit disables or enables the noise filter for input from the GTIOCnA pin. Because changing the value of the bit might lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the GTIOR register before doing so.

NFCSA[1:0] bits (Noise Filter A Sampling Clock Select)

The NFCSA[1:0] bits set the sampling interval for the noise filter of the GTIOCnA pin. When setting these bits, wait for 2 cycles of the selected sampling interval before setting the input capture function.

GTIOB[4:0] bits (GTIOCnB Pin Function Select)

The GTIOB[4:0] bits select the GTIOCnB pin function. For details, see Table 22.4 and Table 22.5.

In Saw-wave PWM mode 2, only the GTIOB[1:0] bits are valid, and the GTIOCnA pin output is selected instead of the GTIOCnB pin by the GTCCRB register compare match.

OBDFLT bit (GTIOCnB Pin Output Value Setting at the Count Stop)

The OBDFLT bit sets whether the GTIOCnB pin outputs high or low when counting stops.

OBHLD bit (GTIOCnB Pin Output Setting at the Start/Stop Count)

The OBHLD bit specifies whether the GTIOCnB pin output level is retained or the level at the start or stop of counting depends on the register setting.

When the OBHLD bit is set to 0:

- The value specified in bit [4] of the GTIOB[4:0] bits is output when counting starts
- The value specified in the OBDFLT bit is output when counting stops
- If the OBDFLT bit is modified while counting stops, the new value is immediately reflected in the output.

When the OBHLD bit is set to 1:

- The output is retained when counting starts or stops.

OBE bit (GTIOCnB Pin Output Enable)

The OBE bit disables or enables the GTIOCnB pin output.

When GTCCRB register is used as the input capture register (at least one bit in the GTICBSR register is set to 1), the GTIOCnB pin does not output regardless of the OBE bit value.

OBDF[1:0] bits (GTIOCnB Pin Disable Value Setting)

The OBDF[1:0] bits select the output value of the GTIOCnB pin in response to a request to disable output from the POEG.

OBEOCD bit (GTCCRB Compare Match Cycle End Output Invalidate)

If the cycle end matches the GTCCRB compare match timing in Saw-wave PWM mode 1 and 2, select invalid/valid of the GTIOB[3:2] bits setting in saw-wave PWM mode 1 or the GTIOA[3:2] bit setting in saw-wave PWM mode 2. When 1 (disabled) is set, the GTIOCnB pin in saw-wave PWM mode 1 or the GTIOCnA pin in saw-wave PWM mode 2 holds the output when the cycle end and the GTCCRB compare match timing match.

NFBEN bit (Noise Filter B Enable)

The NFBEN bit disables or enables the noise filter for input from the GTIOCnB pin. Because changing the value of the bit might lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the GTIOR register before doing so.

NFCSB[1:0] bits (Noise Filter B Sampling Clock Select)

The NFCSB[1:0] bits set the sampling interval for the noise filter of the GTIOCnB pin. When setting these bits, wait for 2 cycles of the selected sampling interval before setting the input capture function.

Table 22.4 Settings of GTIOA[4:0] and GTIOB[4:0] bits (Saw-wave mode, Triangle-wave mode)

GTIOA/GTIOB[4:0] bits					Function		
b4	b3	b2	b1	b0	b4*4	b3, b2*1 *2 *3 *4	b1, b0*2
0	0	0	0	0	Initial output is low	Output retained at cycle end	Output retained at GTCCRA/GTCCRB compare match
0	0	0	0	1			Low output at GTCCRA/GTCCRB compare match
0	0	0	1	0			High output at GTCCRA/GTCCRB compare match
0	0	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	0	1	0	0		Low output at cycle end	Output retained at GTCCRA/GTCCRB compare match
0	0	1	0	1			Low output at GTCCRA/GTCCRB compare match
0	0	1	1	0			High output at GTCCRA/GTCCRB compare match
0	0	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	1	0	0	0		High output at cycle end	Output retained at GTCCRA/GTCCRB compare match
0	1	0	0	1			Low output at GTCCRA/GTCCRB compare match
0	1	0	1	0			High output at GTCCRA/GTCCRB compare match
0	1	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	1	1	0	0		Output toggled at cycle end	Output retained at GTCCRA/GTCCRB compare match
0	1	1	0	1			Low output at GTCCRA/GTCCRB compare match
0	1	1	1	0			High output at GTCCRA/GTCCRB compare match
0	1	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	0	0	0	0	Initial output is high	Output retained at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	0	0	0	1			Low output at GTCCRA/GTCCRB compare match
1	0	0	1	0			High output at GTCCRA/GTCCRB compare match
1	0	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	0	1	0	0		Low output at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	0	1	0	1			Low output at GTCCRA/GTCCRB compare match
1	0	1	1	0			High output at GTCCRA/GTCCRB compare match
1	0	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	1	0	0	0		High output at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	1	0	0	1			Low output at GTCCRA/GTCCRB compare match
1	1	0	1	0			High output at GTCCRA/GTCCRB compare match
1	1	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	1	1	0	0		Output toggled at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	1	1	0	1			Low output at GTCCRA/GTCCRB compare match
1	1	1	1	0			High output at GTCCRA/GTCCRB compare match
1	1	1	1	1			Output toggled at GTCCRA/GTCCRB compare match

Note 1. The cycle end means an overflow (GTCNT changes from GTPR to 0 in up-counting), an underflow (GTCNT changes from 0 to GTPR in down-counting), or counter clearing for saw-wave mode, and means a trough (GTCNT changes from 0 to 1) for triangle-wave mode.

Note 2. When the timing of a cycle end and the timing of a GTCCRA/GTCCRB compare match are the same in a compare-match operation, if the OAEOCD and OBEOCD bits are set to 0 and the end of cycle output is enabled, the b3 and b2 settings are given priority in saw-wave PWM mode, and the b1 and b0 settings are given priority in any other mode.

Note 3. In event count operation where at least one bit in GTUPSR or GTDNSR is set to 1, the setting of b3 and b2 is ignored.

Note 4. In Saw-wave PWM mode 2, GTIOB[4:2] bits is invalid. Since only GTIOCnA pins are output pins, set GTIOA[4] bit for initial output. Set GTIOA[3:2] bits for output at the end of a cycle.

Table 22.5 Settings of GTIOA[4:0] and GTIOB[4:0] bits (Complementary PWM mode)

GTIOA/GTIOB[4:0] bits*1 *2 *3					Function		
					Initial output, Active level	Up count Compare Match output	Down count Compare Match output
b4	b3	b2	b1	b0	b4	b3, b2	b1, b0
0	0	1	1	0	Initial output is Low. Active level is High.	Low output	High output
0	1	0	0	1		High output	Low output
1	0	1	1	0	Initial output is High. Active level is Low.	Low output	High output
1	1	0	0	1		High output	Low output

Note 1. In complementary PWM mode, the only values that can be set in the GTIOA[4:0] bits are 01001b, and 10110b. Setting other values is prohibited.

Note 2. In complementary PWM mode, the only values that can be set in the GTIOB[4:0] bits are 00110b, and 11001b. Setting other values is prohibited.

Note 3. In complementary PWM mode, setting GTIOB[4:0] bits does not use compare match of GTCCRB register. The combination of counter and register that is the target of compare match depends on the operation period of complementary PWM mode. For details, see section 22.3.3.7. Complementary PWM mode 1, 2, 3.

22.2.15 GTINTAD : General PWM Timer Interrupt Output Setting Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x38

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	GRPABL	GRPABH	GRPDTE	—	—	GRP[1:0]		—	—	—	—	ADTRBDEN	ADTRBUEN	ADTRADEN	ADTRAUEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SCFPU	SCFPO	SCFF	SCFE	SCFD	SCFC	SCFB	SCFA	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	SCFA	GTCCRA Register Compare Match/Input Capture Source Synchronous Clear Enable*1 0: Disable use of GTCCRA register compare match/input capture as a clear factor for other channels. 1: Enable use of GTCCRA register compare match/input capture as a clear factor for other channels.	R/W
9	SCFB	GTCCRB Register Compare Match/Input Capture Source Synchronous Clear Enable*1 0: Disable use of GTCCRB register compare match/input capture as a clear factor for other channels. 1: Enable use of GTCCRB register compare match/input capture as a clear factor for other channels.	R/W
10	SCFC	GTCCRC Register Compare Match Source Synchronous Clear Enable*1 0: Disable use of GTCCRC register compare match as a clear factor for other channels. 1: Enable use of GTCCRC register compare match as a clear factor for other channels.	R/W
11	SCFD	GTCCRD Register Compare Match Source Synchronous Clear Enable*1 0: Disable use of GTCCRD register compare match as a clear factor for other channels. 1: Enable use of GTCCRD register compare match as a clear factor for other channels.	R/W

Bit	Symbol	Function	R/W
12	SCFE	GTCCRE Register Compare Match Source Synchronous Clear Enable*1 0: Disable use of GTCCRE register compare match as a clear factor for other channels. 1: Enable use of GTCCRE register compare match as a clear factor for other channels	R/W
13	SCFF	GTCCRF Register Compare Match Source Synchronous Clear Enable*1 0: Disable use of GTCCRF register compare match as a clear factor for other channels. 1: Enable use of GTCCRF register compare match as a clear factor for other channels	R/W
14	SCFPO	Overflow Source Synchronous Clear Enable*1 0: Disable use of overflow as a clear factor for other channels. 1: Enable use of overflow as a clear factor for other channels.	R/W
15	SCFPU	Underflow Source Synchronous Clear Enable*1 0: Disable use of underflow as a clear factor for other channels 1: Enable use of underflow as a clear factor for other channels	R/W
16	ADTRAUEN	GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Enable 0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
17	ADTRADEN	GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Enable 0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
18	ADTRBUEN	GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Enable 0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
19	ADTRBDEN	GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Enable 0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
25:24	GRP[1:0]	Output Disable Source Select 0 0: Group A output disable source is selected 0 1: Group B output disable source is selected 1 0: Group C output disable source is selected 1 1: Group D output disable source is selected	R/W
27:26	—	These bits are read as 0. The write value should be 0.	R/W
28	GRPDTE	Dead Time Error Output Disable Request Enable 0: Dead time error output disable request is disabled. 1: Dead time error output disable request is enabled.	R/W
29	GRPABH	Same Time Output Level High Disable Request Enable 0: Same time output level high disable request disabled 1: Same time output level high disable request enabled	R/W
30	GRPABL	Same Time Output Level Low Disable Request Enable 0: Same time output level low disable request disabled 1: Same time output level low disable request enabled	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. This bit is only available in GPT324 to GPT329.
In GPT320 to GPT323 and GPT3210 to GPT3213, this bit is read as 0. The write value should be 0.

The GTINTAD enables or disables interrupt requests, A/D conversion start request, and output disable requests.

SCFA bit (GTCCRA Register Compare Match/Input Capture Source Synchronous Clear Enable)

This bit enables or disables use of GTCCRA register compare match/input capture as a clear factor for other channels.

SCFB bit (GTCCRB Register Compare Match/Input Capture Source Synchronous Clear Enable)

This bit enables or disables use of GTCCRB register compare match/input capture as a clear factor for other channels.

The setting is invalid in complementary PWM mode.

SCFC bit (GTCCRC Register Compare Match Source Synchronous Clear Enable)

This bit enables or disables use of GTCCRC register compare match as a clear factor for other channels.

The setting is invalid in complementary PWM mode.

SCFD bit (GTCCRD Register Compare Match Source Synchronous Clear Enable)

This bit enables or disables use of GTCCRD register compare match as a clear factor for other channels.

The setting is invalid in complementary PWM mode.

SCFE bit (GTCCRE Register Compare Match Source Synchronous Clear Enable)

This bit enables or disables use of GTCCRE register compare match as a clear factor for other channels.

The setting is invalid in complementary PWM mode.

SCFF bit (GTCCRF Register Compare Match Source Synchronous Clear Enable)

This bit enables or disables use of GTCCRF register compare match as a clear factor for other channels.

The setting is invalid in complementary PWM mode.

SCFPO bit (Overflow Source Synchronous Clear Enable)

This bit enables or disables use of overflow as a clear factor for other channels.

In complementary PWM mode, it is valid only for the master channel.

SCFPU bit (Underflow Source Synchronous Clear Enable)

This bit enables or disables use of underflow as a clear factor for other channels.

In complementary PWM mode, it is valid only for the master channel.

ADTRAUEN bit (GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Enable)

This bit enables or disables A/D conversion start requests generated by GTADTRA register compare matches during GTCNT counter up-counting.

In complementary PWM mode, it is valid only for the master channel.

The setting is invalid during the event count operation, and A/D conversion start request is not generated.

ADTRADEN bit (GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Enable)

This bit enables or disables A/D conversion start requests generated by GTADTRA register compare matches during GTCNT counter down-counting.

In complementary PWM mode, it is valid only for the master channel.

The setting is invalid during the event count operation, and A/D conversion start request is not generated.

ADTRBUEN bit (GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Enable)

This bit enables or disables A/D conversion start requests generated by GTADTRB register compare matches during GTCNT counter up-counting.

In complementary PWM mode, it is valid only for the master channel.

The setting is invalid during the event count operation, and A/D conversion start request is not generated.

ADTRBDEN bit (GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Enable)

This bit enables or disables A/D conversion start requests generated by GTADTRB register compare matches during GTCNT counter down-counting.

In complementary PWM mode, it is valid only for the master channel.

The setting is invalid during the event count operation, and A/D conversion start request is not generated.

GRP[1:0] bits (Output Disable Source Select)

These bits select the group of output disable request from GPT to POEG and the group of output disable for GTIOCnA pin and GTIOCnB pin from POEG to GPT.

The output disable request to POEG is output to the group selected in the GRP[1:0] bit, with dead-time errors, simultaneous high output, and simultaneous low output factors following their respective disable request enable bits.

GTST.ODF shows the request of the output disable source group that is selected with the GRP[1:0] bits. Set the GRP[1:0] bits when both GTIOR.OAE and GTIOR.OBE bits are 0.

GRPDTE bit (Dead Time Error Output Disable Request Enable)

This bit enables or disables the output disable request by a dead time error.

The dead time error output disable request is not generated during the event count operation.

GRPABH bit (Same Time Output Level High Disable Request Enable)

The GRPABH bit enables or disables the output disable request when the GTIOCnA pin and GTIOCnB pin output 1 at the same time.

GRPABL bit (Same Time Output Level Low Disable Request Enable)

The GRPABL bit enables or disables the output disable request when the GTIOCnA pin and GTIOCnB pin output 0 at the same time.

22.2.16 GTST : General PWM Timer Status Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x3C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PCF	OABLF	OABHF	DTEF	—	—	—	ODF	—	—	—	—	ADTRBDF	ADTRBUF	ADTRADF	ADTRAUF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TUCF	—	—	—	—	ITCNT[2:0]			TCFPU	TCFPO	TCFF	TCFE	TCFD	TCFC	TCFB	TCFA
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TCFA	Input Capture/Compare Match Flag A 0: No input capture/compare match of GTCCRA is generated 1: An input capture/compare match of GTCCRA is generated	R/W*1
1	TCFB	Input Capture/Compare Match Flag B 0: No input capture/compare match of GTCCRB is generated 1: An input capture/compare match of GTCCRB is generated	R/W*1
2	TCFC	Input Compare Match Flag C 0: No compare match of GTCCRC is generated 1: A compare match of GTCCRC is generated	R/W*1
3	TCFD	Input Compare Match Flag D 0: No compare match of GTCCRD is generated 1: A compare match of GTCCRD is generated	R/W*1
4	TCFE	Input Compare Match Flag E 0: No compare match of GTCCRE is generated 1: A compare match of GTCCRE is generated	R/W*1

Bit	Symbol	Function	R/W
5	TCFF	Input Compare Match Flag F 0: No compare match of GTCCRF is generated 1: A compare match of GTCCRF is generated	R/W*1
6	TCFPO	Overflow Flag 0: No overflow (crest) occurred 1: An overflow (crest) occurred	R/W*1
7	TCFPU	Underflow Flag 0: No underflow (trough) occurred 1: An underflow (trough) occurred	R/W*1
10:8	ITCNT[2:0]	GPTn_OVF/GPTn_UDF Interrupt Skipping Count Counter	R
14:11	—	These bits are read as 0. The write value should be 0.	R/W
15	TUCF	Count Direction Flag 0: GTCNT counter counts downward 1: GTCNT counter counts upward	R
16	ADTRAUF	GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Flag 0: No GTADTRA register compare match has occurred in up-counting. 1: A GTADTRA register compare match has occurred in up-counting.	R/W*1
17	ADTRADF	GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Flag 0: No GTADTRA register compare match has occurred in down-counting. 1: A GTADTRA register compare match has occurred in down-counting.	R/W*1
18	ADTRBUF	GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Flag 0: No GTADTRB register compare match has occurred in up-counting. 1: A GTADTRB register compare match has occurred in up-counting.	R/W*1
19	ADTRBDF	GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Flag 0: No GTADTRB register compare match has occurred in down-counting. 1: A GTADTRB register compare match has occurred in down-counting.	R/W*1
23:20	—	These bits are read as 0. The write value should be 0.	R/W
24	ODF	Output Disable Flag 0: No output disable request is generated 1: An output disable request is generated	R
27:25	—	These bits are read as 0. The write value should be 0.	R/W
28	DTEF	Dead Time Error Flag 0: No dead time error has occurred. 1: A dead time error has occurred.	R
29	OABHF	Same Time Output Level High Flag 0: No simultaneous generation of 1 both for the GTIOCA and GTIOCB pins has occurred. 1: A simultaneous generation of 1 both for the GTIOCA and GTIOCB pins has occurred.	R
30	OABLF	Same Time Output Level Low Flag 0: No simultaneous generation of 0 both for the GTIOCA and GTIOCB pins has occurred. 1: A simultaneous generation of 0 both for the GTIOCA and GTIOCB pins has occurred.	R
31	PCF*2	Period Count Function Finish Flag 0: No period count function finish has occurred 1: A period count function finish has occurred	R/W*1

Note 1. Only 0 can be written to this bit. Do not write 1.
When clearing the ADTRAUF, ADTRADF, ADTRBUF, or ADTRBDF flag, be sure to write 0 only to the target flag or flags for clearing and to write 1 to the other flags not for clearing.

Note 2. This bit is only available in GPT320 to GPT322 and GPT3210 to GPT3213.
In GPT324 to GPT329, this bit is read as 0. The write value should be 0.

The GTST indicates the status of the GPT.

TCFA flag (Input Capture/Compare Match Flag A)

The TCFA flag indicates the status for the input capture or compare match of GTCCRA.

[Setting conditions]

- GTCNT = GTCCRA, when the GTCCRA register functions as a compare match register
- GTCNT counter value is transferred to GTCCRA by the input capture signal when the GTCCRA register functions as an input capture register.

[Clearing condition]

- 0 is written to this flag.

TCFB flag (Input Capture/Compare Match Flag B)

The TCFB flag indicates the status for the input capture or compare match of GTCCRB.

[Setting conditions]

- GTCNT = GTCCRB, when the GTCCRB register functions as a compare match register
- GTCNT counter value is transferred to GTCCRB by the input capture signal when the GTCCRB register functions as an input capture register.

[Clearing condition]

- 0 is written to this flag.

TCFC flag (Input Compare Match Flag C)

The TCFC flag indicates the status for the compare match of GTCCRC.

When GTCCRC performs buffer operation, GTCCRC does not perform compare match.

[Setting condition]

- GTCNT = GTCCRC.

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 01b, 10b, 11b (GTCCRC performs buffer operation).

TCFD flag (Input Compare Match Flag D)

The TCFD flag indicates the status for the compare match of GTCCRD.

When GTCCRD performs buffer operation, GTCCRD does not perform compare match.

[Setting condition]

- GTCNT = GTCCRD.

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (Triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 10b, 11b (GTCCRD performs buffer operation).

TCFE flag (Input Compare Match Flag E)

The TCFE flag indicates the status for the compare match of GTCCRE.

When GTCCRE performs buffer operation, GTCCRE does not perform compare match.

[Setting condition]

- GTCNT = GTCCRE.

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (Triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 01b, 10b, 11b (GTCCRE performs buffer operation).

TCFF flag (Input Compare Match Flag F)

The TCFF flag indicates the status for the compare match of GTCCRF.

When GTCCRF performs buffer operation, GTCCRF does not perform compare match.

[Setting condition]

- GTCNT = GTCCRF.

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (Triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 10b, 11b (GTCCRF performs buffer operation).

TCFPO flag (Overflow Flag)

The TCFPO flag indicates when an overflow or crest has occurred.

[Setting conditions]

- In saw-wave mode, an overflow (GTCNT changes from GTPR to 0 in up-counting) has occurred
- In triangle-wave mode, a crest (GTCNT changes from GTPR to GTPR - 1) has occurred
- In counting by hardware sources, an overflow (GTCNT changes from GTPR to 0 in up-counting) has occurred.

[Clearing condition]

- 0 is written to this flag.

TCFPU flag (Underflow Flag)

The TCFPU flag indicates when an underflow or trough has occurred.

[Setting conditions]

- In saw-wave mode, an underflow (GTCNT changes from 0 to GTPR in down-counting) has occurred
- In triangle-wave mode, a trough (GTCNT changes from 0 to 1) has occurred
- In counting by hardware sources, an underflow (GTCNT changes from 0 to GTPR in down-counting) has occurred.

[Clearing condition]

- 0 is written to this bit.

ITCNT[2:0] flag (GPTn_OVF/GPTn_UDF Interrupt Skipping Count Counter)

When the GPTn_OVF/GPTn_UDF interrupt skipping function is used (the GTITC.IVTC[1:0] bits are set to a value other than 00b), the counter is incremented by 1 every time the GPTn_OVF/GPTn_UDF interrupt source which is selected in GTITC.IVTC[1:0] is generated.

These bits are operated independently of the extended interrupt skipping by the GTEITC register.

[Clearing conditions]

- The GPTn_OVF/GPTn_UDF interrupt skipping function is not used (the GTITC.IVTT[2:0] bits are 000b when the IVTC[1:0] bits are 00b).
- The GPTn_OVF/GPTn_UDF interrupt skipping count matches the specified count (the ITCNT[2:0] bits match the skipping count specified by the IVTT[2:0] bits).
- When the count operation is stopped

TUCF flag (Count Direction Flag)

The TUCF flag indicates the count direction of GTCNT. In event count operation, this flag is set to 1 in up-counting and to 0 in down-counting.

ADTRAUF flag (GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Flag)

This status flag indicates generation of a GTADTRA register compare match in up-counting.

[Setting condition]

- The GTCNT counter matches the GTADTRA register in up-counting.

[Clearing condition]

- 0 is written to the ADTRAUF flag.

ADTRADF flag (GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Flag)

This status flag indicates generation of a GTADTRA register compare match in down-counting.

[Setting condition]

- The GTCNT counter matches the GTADTRA register in down-counting.

[Clearing condition]

- 0 is written to the ADTRADF flag.

ADTRBUF flag (GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Flag)

This status flag indicates generation of a GTADTRB register compare match in up-counting.

[Setting condition]

- The GTCNT counter matches the GTADTRB register in up-counting.

[Clearing condition]

- 0 is written to the ADTRBUF flag.

ADTRBDF flag (GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Flag)

This status flag indicates generation of a GTADTRB register compare match in down-counting.

[Setting condition]

- The GTCNT counter matches the GTADTRB register in down-counting.

[Clearing condition]

- 0 is written to the ADTRBDF flag.

ODF flag (Output Disable Flag)

The ODF flag shows the request of the output disable source group that is selected in the GRP[1:0] bits.

When output is disabled, an output disable control is not released within the same cycle in which an output disable request is negated. It is released in the next cycle.

DTEF flag (Dead Time Error Flag)

This flag indicates that the timer output toggle point after the automatic addition of dead time has exceeded the count period.

This flag returns to 0 when the timer output toggle point after the automatic addition of dead time is back within the period.

This flag can only be read from. (Writing 0 to clear the flag is not allowed.)

When the output disable request by the DTEF flag is enabled (when GTINTAD.GRPDTE bit is 1), the DTEF flag is output as the output disable request to the POEG. The GPT does not have a dead time error interrupt. Use the interrupt function in the POEG if this is necessary.

[Setting condition]

- When a change point of the waveform after the automatic setting of dead time has exceeded the count period (in the following cases).
  - Up-counting in triangle-wave mode:
    GTCCRA register – GTDVU register ≤ 0
  - Down-counting in triangle-wave mode:
    GTCCRA register – GTDVD register < 0
  - Up-counting in saw-wave one-shot pulse mode:
    GTCCRA register – GTDVU register < 0, or
    GTCCRA register + GTDVD register > GTPR register
  - Down-counting in saw-wave one-shot pulse mode:
    GTCCRA register + GTDVU register > GTPR register, or
    GTCCRA register - GTDVD register < 0

[Clearing condition]

- The timer output toggle point after the automatic addition of dead time is within the count period.

OABHF flag (Same Time Output Level High Flag)

The OABHF flag indicates that the GTIOCnA pin and GTIOCnB pin output 1 at the same time.

When the GTIOCnA or GTIOCnB pin outputs 0, this flag returns to 0. This flag is read only. Writing 0 to clear the flag is prohibited.

When the output disable request by the OABHF flag is enabled (GTINTAD.GRPABH = 1), the OABHF flag is output to POEG as an output disable request. The GPT does not have an interrupt to indicate that outputs have been simultaneous driven to the high level. Use the interrupt function in the POEG if this is necessary.

[Setting condition]

- The GTIOCnA and GTIOCnB pins output 1 at the same time when both OAE and OBE bits are set to 1.

[Clearing conditions]

- The GTIOCnA pin output value is different from the GTIOCnB pin output value when both OAE and OBE bits are set to 1
- The GTIOCnA and GTIOCnB pins output 0 at the same time when both OAE and OBE bits are set to 1
- Either the OAE bit or OBE bit is set to 0.

OABLF flag (Same Time Output Level Low Flag)

The OABLF flag indicates that the GTIOCnA and GTIOCnB pins output 0 at the same time.

When the GTIOCnA pin or GTIOCnB pin outputs 1, this flag returns to 0. This flag is read only. Writing 0 to clear the flag is prohibited.

When the output disable request by the OABLF flag is enabled (GTINTAD.GRPABL = 1), the OABLF flag is output to POEG as an output disable request. The GPT does not have an interrupt to indicate that outputs have been simultaneous driven to the low level. Use the interrupt function in the POEG if this is necessary.

[Setting condition]

- The GTIOCnA and GTIOCnB pins output 0 at the same time when both OAE and OBE bits are set to 1.

[Clearing conditions]

- The GTIOCnA pin output value is different from the GTIOCnB pin output value when both OAE and OBE bits are set to 1
- The GTIOCnA and GTIOCnB pins output 1 at the same time when both OAE and OBE bits are set to 1
- Either the OAE bit or the OBE bit is set to 0.

The compare-target signals to generate the OABHF/OABLF flag are the compare match outputs (PWM outputs) signals before they are masked by the output disable function. Even during the output disable condition, compare match operation continues internally, where the OABHF or OABLF flag is updated based on the operation results.

PCF flag (Period Count Function Finish Flag)

This bit is status flag of period count function finish.

[Setting condition]

- The GTPC.PCEN bit is 1 and the GTPC.PCNT counter is 1 at the end of cycle.
- The GTPC.PCEN bit is 1 and the GTPC.PCNT counter is 0 at the count clock.

[Clearing condition]

- 0 is written to this bit.

22.2.17 GTBER : General PWM Timer Buffer Enable Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x40

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	ADTDB	ADTTB[1:0]		—	ADTDA	ADTTA[1:0]		—	CCRSWT	PR[1:0]		CCRB[1:0]		CCRA[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	DBRTECB	—	DBRTECA	—	—	—	—	BD3	BD2	BD1	BD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BD0	GTCCR Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
1	BD1	GTPR Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W*1
2	BD2	GTADTRA/GTADTRB Registers Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
3	BD3	GTDVU/GTDVD Registers Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	DBRTECA	GTCCRA Register Double Buffer Repeat Operation Enable 0: GTCCRA register double buffer repeat operation is disabled 1: GTCCRA register double buffer repeat operation is enabled	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
10	DBRTECB	GTCCRB Register Double Buffer Repeat Operation Enable 0: GTCCRB register double buffer repeat operation is disabled 1: GTCCRB register double buffer repeat operation is enabled	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
17:16	CCRA[1:0]	GTCCRA Buffer Operation 0 0: No buffer operation 0 1: Single buffer operation (GTCCRA ↔GTCCRC) Others: Double buffer operation (GTCCRA ↔ GTCCRC ↔ GTCCRD)	R/W
19:18	CCRB[1:0]	GTCCRB Buffer Operation 0 0: No buffer operation 0 1: Single buffer operation (GTCCRB ↔ GTCCRE) Others: Double buffer operation (GTCCRB ↔ GTCCRE ↔ GTCCRF)	R/W
21:20	PR[1:0]	GTPR Buffer Operation 0 0: No buffer operation 0 1: Single buffer operation (GTPBR → GTPR) Others: Double buffer operation (GTPDBR → GTPBR → GTPR)	R/W
22	CCRSWT	GTCCRA and GTCCRB Forcible Buffer Operation Writing 1 to this bit forces a buffer transfer of GTCCRA and GTCCRB. This bit automatically returns to 0 after 1 is written. This bit is read as 0.	W
23	—	This bit is read as 0. The write value should be 0.	R/W
25:24	ADTTA[1:0]	GTADTRA Register Buffer Transfer Timing Select 0 0: In triangle wave or complementary PWM mode, no transfer. In saw-wave mode, no transfer. 0 1: In triangle wave or complementary PWM mode, transfer at crest. In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing. 1 0: In triangle wave or complementary PWM mode, transfer at trough. In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing. 1 1: In triangle wave or complementary PWM mode, transfer at both crest and trough. In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing.	R/W
26	ADTDA	GTADTRA Register Double Buffer Operation 0: Single buffer operation (GTADTBRA → GTADTRA) 1: Double buffer operation (GTADTDBRA → GTADTBRA → GTADTRA)	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
29:28	ADTTB[1:0]	GTADTRB Register Buffer Transfer Timing Select 0 0: In triangle wave or complementary PWM mode, no transfer. In saw-wave mode, no transfer. 0 1: In triangle wave or complementary PWM mode, transfer at crest. In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing. 1 0: In triangle wave or complementary PWM mode, transfer at trough. In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing. 1 1: In triangle wave or complementary PWM mode, transfer at both crest and trough. In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing.	R/W
30	ADTDB	GTADTRB Register Double Buffer Operation 0: Single buffer operation (GTADTBRB → GTADTRB) 1: Double buffer operation (GTADTDBRB → GTADTBRB → GTADTRB)	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

The GTBER register provides settings for the buffer operation. Set the GTBER register except the BDx (x = 0 to 3) bits while the GTCNT counter is stopped.

BD0 bit (GTCCR Buffer Operation Disable)

The BD0 bit disables the buffer operation using GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, and GTCCRF combined.

When GTDTCR.TDE is 1 and when BD0 is set to 0, GTCCRB does not perform buffer operation in Saw-wave one-shot pulse mode or Triangle-wave PWM mode. The GTCCRB register is automatically set to a compare match value for negative-phase waveform with dead time.

In complementary PWM mode, it is valid only for the buffer operation of GTCCRC register and GTCCRE register. The buffer operation of GTCCRA registers cannot be disabled. Buffer operation of GTCCRE register and GTCCRF register is enabled/disabled by CP3DB bit of GTBER2 register. No buffer transfer to GTCCRB is performed in complementary PWM mode.

A value for the BD0 bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the GTSECR.SBDCE or GTSECR.SBDCD.

When the DBRTECm (m = A, B) bit is 1, setting the BD0 bit to 1 while the mode of operation is saw-wave one-shot pulse mode or triangle-wave PWM mode 3 results in transfer from the intermediate buffer to the GTCCRm register.

BD1 bit (GTPR Buffer Operation Disable)

The BD1 bit disables the buffer operation using GTPR, GTPDBR, and GTPBR combined.

A value for the BD1 bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the GTSECR.SBDPE or GTSECR.SBDPD.

In complementary PWM mode, the slave channel is also controlled by setting the BD1 bit of the master channel.

BD2 bit (GTADTRA/GTADTRB Registers Buffer Operation Disable)

This bit disables buffer operation using the GTADTRA, GTADTBRA, and GTADTDBRA registers together and buffer operation using the GTADTRB, GTADTBRB, and GTADTDBRB registers together.

The setting is invalid during the event count operation, and the buffer operation using the GTADTRA and GTADTRB registers is not performed.

A value for the BD2 bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the SBDAE or SBDAD bit in the GTSECR register.

BD3 bit (GTDVU/GTDVD Registers Buffer Operation Disable)

In Saw-wave PWM mode 1, Saw-wave one-shot pulse mode or Triangle-wave PWM mode, this bit disables buffer operation using the GTDVU and GTDBU registers together and buffer operation using the GTDVD and GTDBD registers together.

Even though the BD3 bit is set to 0, buffer operation in the GTDVD register is not performed if the GTDTCR.TDFER bit is set to 1. Instead, the value in the GTDVU register is set automatically.

In Saw-wave PWM mode 2 or complementary PWM mode, this bit is invalid and GTDVU and GTDVD registers don't perform buffer operation.

The setting is invalid during the event count operation, and the buffer operation using the GTDVU and GTDVD registers is not performed.

A value for the BD3 bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the SBDDE or SBDDD bit in the GTSECR register.

DBRTECm bits (GTCCRm Register Double Buffer Repeat Operation Enable) (m = A, B)

This setting enables the operation to repeat a transfer to the GTCCRm register from the intermediate buffer by cycle during the buffer transfer disable period when performing the double buffer operation using the GTCCRm register.

It is valid in saw-wave one-shot pulse mode and triangle-wave PWM mode 3.

The disabling period of buffer transfer indicates the period that buffer transfer is stopped by setting of the BD0 bit (CPU writing or simultaneous buffer operation control by the GTSECSR register) and the period for buffer transfer extended skipping (except the case for skipping by counting both crests and troughs) by the GTEITLB register.

When the DBRTECm bit is 1, writing by the CPU to the GTCCRm register sets the same value in temporary register x (x = C, E). The value of the GTCCRx (x = C, E) register is also transferred to the temporary register x (x = C, E) by forcible buffer transfer.

CCRA[1:0] bits (GTCCRA Buffer Operation)

The CCRA[1:0] bits set the buffer operation with GTCCRA, GTCCRC, and GTCCRD combined. When the buffer operation is restricted by the operating mode set in GTCR, the GTCR setting is given priority.

The buffer operation mode is fixed in saw-wave one-shot pulse mode or triangle-wave PWM mode 3 (64-bit transfer at trough), or complementary PWM mode.

CCRB[1:0] bits (GTCCRB Buffer Operation)

The CCRB[1:0] bits set the buffer operation using GTCCRB, GTCCRE, and GTCCRF combined. When the buffer operation is restricted by the operating mode set in GTCR, the GTCR setting is given priority.

The buffer operation mode is fixed in saw-wave one-shot pulse mode or triangle-wave PWM mode 3 (64-bit transfer at trough), or complementary PWM mode.

PR[1:0] bits (GTPR Buffer Operation)

The PR[1:0] bits set the buffer operation with GTPR, GTPDBR, and GTPBR combined.

The setting is invalid in complementary PWM mode. Buffer operation unique to complementary PWM mode is performed regardless of the setting value of the PR[1:0] bits.

CCRSWT bit (GTCCRA and GTCCRB Forcible Buffer Operation)

Writing 1 to the CCRSWT bit forces a buffer transfer of GTCCRA and GTCCRB. This bit automatically returns to 0 after the 1 is written. This bit is read as 0, and is valid only when counting is stopped with a compare match operation specified.

The setting is invalid in complementary PWM mode.

ADTTA[1:0] bits (GTADTRA Register Buffer Transfer Timing Select)

These bits set the transfer timing for buffer operation of the GTADTRA, GTADTBRA, and GTADTDBRA registers.

The setting is invalid during the event count operation.

ADTDA bit (GTADTRA Register Double Buffer Operation)

These bits set buffer operation with the GTADTRA, GTADTBRA, and GTADTDBRA registers combined.

The setting is invalid during the event count operation.

ADTTB[1:0] bits (GTADTRB Register Buffer Transfer Timing Select)

These bits set the transfer timing for buffer operation of the GTADTRB, GTADTBRB, and GTADTDBRB registers.

The setting is invalid during the event count operation.

ADTDB bit (GTADTRB Register Double Buffer Operation)

These bits set buffer operation with the GTADTRB, GTADTBRB, and GTADTDBRB registers combined.

The setting is invalid during the event count operation.

22.2.18 GTITC : General PWM Timer Interrupt and A/D Conversion Start Request Skipping Setting Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x44

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	ADTBL	—	ADTAL	—	IVTT[2:0]			IVTC[1:0]		ITLF	ITLE	ITLD	ITLC	ITLB	ITLA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ITLA	GTCCRA Register Compare Match/Input Capture Interrupt Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function. 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function.	R/W
1	ITLB	GTCCRB Register Compare Match/Input Capture Interrupt Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function. 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function.	R/W
2	ITLC	GTCCRC Register Compare Match Interrupt Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function. 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function.	R/W
3	ITLD	GTCCRD Register Compare Match Interrupt Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function. 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function.	R/W
4	ITLE	GTCCRE Register Compare Match Interrupt Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function. 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function.	R/W
5	ITLF	GTCCRF Register Compare Match Interrupt Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function. 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function.	R/W
7:6	IVTC[1:0]	GPTn_OVF/GPTn_UDF Interrupt Skipping Function Select 0 0: Skipping is not performed. 0 1: Both overflow and underflow for saw waves*1 and crest for triangle waves and complementary PWM mode are counted and skipped. 1 0: Both overflow and underflow for saw waves*1 and trough for triangle waves and complementary PWM mode are counted and skipped. 1 1: Both overflow and underflow for saw waves*1 and both crest and trough for triangle waves and complementary PWM mode are counted and skipped.	R/W
10:8	IVTT[2:0]	GPTn_OVF/GPTn_UDF Interrupt Skipping Count Select 0 0 0: Skipping is not performed 0 0 1: Skipping count of 1 0 1 0: Skipping count of 2 0 1 1: Skipping count of 3 1 0 0: Skipping count of 4 1 0 1: Skipping count of 5 1 1 0: Skipping count of 6 1 1 1: Skipping count of 7	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	ADTAL	GTADTRA Register A/D Conversion Start Request Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function. 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function.	R/W
13	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
14	ADTBL	GTADTRB Register A/D Conversion Start Request Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function. 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function.	R/W
31:15	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Saw-wave PWM mode 2 is not the target of this function.

The GTITC register sets the skipping function for the GTCNT counter overflow (GTPR compare match) interrupt (GPTn_OVF) and underflow interrupt (GPTn_UDF) and also sets whether to link the other interrupts and A/D conversion start requests with the GPTn_OVF/GPTn_UDF interrupt skipping function. Note the output disable request to POEG cannot be linked with the GPTn_OVF/GPTn_UDF interrupt skipping function. Additionally, if the interrupt skipping function is performed, the change in the status flag is also skipped.

The setting is invalid during the event count operation.

The setting is operated independently of the extended interrupt skipping by the GTEITC register.

ITLA bit (GTCCRA Register Compare Match/Input Capture Interrupt Link)

This bit specifies whether to link the GTCCRA compare match/input capture interrupt (GPTn_CCMPA) with the GPTn_OVF/GPTn_UDF interrupt skipping function.

ITLB bit (GTCCRB Register Compare Match/Input Capture Interrupt Link)

This bit specifies whether to link the GTCCRB compare match/input capture interrupt (GPTn_CCMPB) with the GPTn_OVF/GPTn_UDF interrupt skipping function.

ITLC bit (GTCCRC Register Compare Match Interrupt Link)

This bit specifies whether to link the GTCCRC compare match interrupt (GPTn_CMPC) with the GPTn_OVF/GPTn_UDF interrupt skipping function.

ITLD bit (GTCCRD Register Compare Match Interrupt Link)

This bit specifies whether to link the GTCCRD compare match interrupt (GPTn_CMPD) with the GPTn_OVF/GPTn_UDF interrupt skipping function.

ITLE bit (GTCCRE Register Compare Match Interrupt Link)

This bit specifies whether to link the GTCCRE compare match interrupt (GPTn_CMPE) with the GPTn_OVF/GPTn_UDF interrupt skipping function.

ITLF bit (GTCCRF Register Compare Match Interrupt Link)

This bit specifies whether to link the GTCCRF compare match interrupt (GPTn_CMPF) with the GPTn_OVF/GPTn_UDF interrupt skipping function.

IVTC[1:0] bit (GPTn_OVF/GPTn_UDF Interrupt Skipping Function Select)

These bits set the skipping function for the GTPR compare match (GTCNT counter overflow) interrupt (GPTn_OVF) and GTCNT counter underflow interrupt (GPTn_UDF).

IVTT[2:0] bit (GPTn_OVF/GPTn_UDF Interrupt Skipping Count Select)

These bits set the skipping count for the GTPR compare match (GTCNT counter overflow) interrupt (GPTn_OVF) and GTCNT counter underflow interrupt (GPTn_UDF).

When modifying the IVTT[2:0] bits, first set the IVTC[1:0] bits to 00b.

ADTAL bit (GTADTRA Register A/D Conversion Start Request Link)

This bit specifies whether to link the GTADTRA A/D conversion start request, which is generated in response to a compare match with the GTCNT counter and the GTADTRA register, with the GPTn_OVF/GPTn_UDF interrupt skipping function.

ADTBL bit (GTADTRB Register A/D Conversion Start Request Link)

This bit specifies whether to link the GTADTRB A/D conversion start request, which is generated in response to a compare match with the GTCNT counter and the GTADTRB register, with the GPTn_OVF/GPTn_UDF interrupt skipping function.

22.2.19 GTCNT : General PWM Timer Counter

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x48

Bit position:	31 … 0
Bit field:	GTCNT[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	GTCNT[31:0]	GTCNT is a 32-bit read/write counter for GPT32n (n = 0 to 13). GTCNT can only be written to after counting stops. Access in 8-bit or 16-bit units to the GTCNT counter is prohibited, and it should be accessed in 32-bit units. In saw waves or triangle waves, GTCNT must be set within the range of 0 ≤ GTCNT ≤ GTPR.	R/W

22.2.20 GTCCRk : General PWM Timer Compare Capture Register k (k = A to F)

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x4C (GTCCRA)
0x50 (GTCCRB)
0x54 (GTCCRC)
0x58 (GTCCRE)
0x5C (GTCCRD)
0x60 (GTCCRF)

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

Bit	Symbol	Function	R/W
31:0	GTCCRk[31:0]	GTCCRk registers are read/write registers. Access in 8-bit or 16-bit units to the GTCCRk register is prohibited, and it should be accessed in 32-bit units. GTCCRA and GTCCRB are registers used for both output compare and input capture. GTCCRC and GTCCRE are compare match registers, and can also function as buffer registers for GTCCRA and GTCCRB. GTCCRD and GTCCRF are compare match registers, and can also function as buffer registers for GTCCRC and GTCCRE (double-buffer registers for GTCCRA and GTCCRB).	R/W

22.2.21 GTPR : General PWM Timer Cycle Setting Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x64

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

Bit	Symbol	Function	R/W
31:0	GTPR[31:0]	GTPR is a read/write register that sets the maximum count value of GTCNT. Access in 8-bit or 16-bit units to the GTPR register is prohibited, and it should be accessed in 32-bit units. The setting is invalid in the Saw-wave PWM mode 2. For Saw waves except the Saw-wave PWM mode 2, the value of (GTPR + 1) is the cycle. For triangle waves or complementary PWM mode, the value of (GTPR value × 2) is the cycle. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.	R/W

22.2.22 GTPBR : General PWM Timer Cycle Setting Buffer Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x68

Bit position: 31 … 0

Bit field: GTPBR[31:0]

Value after reset: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Bit	Symbol	Function	R/W
31:0	GTPBR[31:0]	GTPBR is a read/write register that functions as a buffer register for GTPR. Access in 8-bit or 16-bit units to the GTPBR register is prohibited, and it should be accessed in 32-bit units. The setting is invalid in Saw-wave PWM mode 2. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.	R/W

22.2.23 GTPDBR : General PWM Timer Cycle Setting Double-Buffer Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x6C

Bit position: 31 … 0

Bit field: GTPDBR[31:0]

Value after reset: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Bit	Symbol	Function	R/W
31:0	GTPDBR[31:0]	The buffer register for the GTPBR register (double buffer register for the GTPR register) Access in 8-bit or 16-bit units to the GTPDBR register is prohibited, and it should be accessed in 32-bit units. The setting is invalid in Saw-wave PWM mode 2. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.	R/W

22.2.24 GTADTRk : A/D Conversion Start Request Timing Register k (k = A, B)

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x70 (GTADTRA)
0x7C (GTADTRB)

Bit position: 31 … 0

Bit field: GTADTRk[31:0]

Value after reset: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Bit	Symbol	Function	R/W
31:0	GTADTRk[31:0]	Set the timing of A/D conversion start request generation Access in 8-bit or 16-bit units to the GTADTRk register is prohibited, and it should be accessed in 32-bit units. When the GTADTRk register value matches the GTCNT counter value, an A/D conversion start request is generated. In complementary PWM mode, A/D conversion start request is generated when the GTCNT counter of the master channel matches this register.	R/W

22.2.25 GTADTBRk : A/D Conversion Start Request Timing Buffer Register k (k = A, B)

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x74 (GTADTBRA)
0x80 (GTADTBRB)

Bit position: 31 … 0

Bit field: GTADTBRk[31:0]

Value after reset: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Bit	Symbol	Function	R/W
31:0	GTADTBRk[31:0]	The buffer registers for the GTADTRk register Access in 8-bit or 16-bit units to the GTADTBRk register is prohibited, and it should be accessed in 32-bit units.	R/W

22.2.26 GTADTDBRk : A/D Conversion Start Request Timing Double-Buffer Register k (k = A, B)

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x78 (GTADTDBRA)
0x84 (GTADTDBRB)

Bit position: 31 … 0

Bit field: GTADTDBRk[31:0]

Value after reset: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Bit	Symbol	Function	R/W
31:0	GTADTDBRk[31:0]	The buffer registers for the GTADTBRk register (double buffer registers for the GTADTRk register) Access in 8-bit or 16-bit units to the GTADTDBRk register is prohibited, and it should be accessed in 32-bit units.	R/W

22.2.27 GTDTCR : General PWM Timer Dead Time Control Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x88

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

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

Bit	Symbol	Function	R/W
0	TDE	Negative-Phase Waveform Setting 0: GTCCRB is set without using GTDVU and GTDVD 1: GTDVU and GTDVD are used to set the compare match value for negative-phase waveform with dead time automatically in GTCCRB	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	TDBUE	GTDVU Register Buffer Operation Enable 0: GTDVU register buffer operation is disabled 1: GTDVU register buffer operation is enabled	R/W
5	TDBDE	GTDVD Register Buffer Operation Enable 0: GTDVD register buffer operation is disabled 1: GTDVD register buffer operation is enabled	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	TDFER	GTDVD Register Setting 0: GTDVU and GTDVD registers are set separately. 1: The value written to GTDVU register is automatically set to GTDVD register.	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

GTDTCR enables automatic setting of a compare match value for negative-phase waveform with dead time. GPT has a dead time control function and GTDVU and GTDVD registers are used for setting dead time value.

This register is invalid in Saw-wave PWM mode 2 or complementary PWM mode.

The setting is invalid during the event count operation.

TDE bit (Negative-Phase Waveform Setting)

The TDE bit specifies whether to use GTDVU and GTDVD. When GTDVU and GTDVD are used, the compare match value for a negative-phase waveform with dead time obtained by the compare match value of a positive-phase waveform (GTCCRA) and the dead time value (GTDVU and GTDVD) is automatically set in GTCCRB.

The TDE bit setting is ignored in saw-wave PWM mode, and the GTCCRB is not automatic setting.

The GTCCRB value is automatically set and has the following upper and lower limit values. If the obtained GTCCRB value is not within the upper or lower limit, the following limit value is set in GTCCRB, and GTST.DTEF flag becomes 1. However, if the obtained GTCCRB value exceeds the upper limit in triangle-wave PWM mode, DTEF flag becomes 0.

- Triangle waves:
  Upper limit value: GTPR − 1
  Lower limit value: 1 in up-counting, 0 in down-counting
- Saw-wave one-shot pulse mode:
  Upper limit value: GTPR
  Lower limit value: 0.

TDBUE bit (GTDVU Register Buffer Operation Enable)

This bit enables buffer operation with the GTDVU and GTDBU registers combined.

The timing of buffer transfer is at troughs in triangle-wave mode, and at overflows or underflows in saw-wave mode.

TDBDE bit (GTDVD Register Buffer Operation Enable)

This bit enables buffer operation with the GTDVD and GTDBD registers combined.

The buffer transfer timing is the trough in triangle-wave mode, and at an overflow or underflow in saw-wave mode.

When this bit and the TDFER bit are set to 1 simultaneously, the TDFER bit setting is given priority.

TDFER bit (GTDVD Register Setting)

This bit sets whether or not the value written to the GTDVU register is also set to the GTDVD register automatically.

22.2.28 GTDVk : General PWM Timer Dead Time Value Register k (k = U, D)

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x8C (GTDVU)
0x90 (GTDVD)

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

Bit	Symbol	Function	R/W
31:0	GTDVk[31:0]	GTDVk is a read/write register that sets the dead time for generating PWM waveforms with dead time. Access in 8-bit or 16-bit units to the GTDVk register is prohibited, and it should be accessed in 32-bit units. The setting is invalid in Saw-wave PWM mode 2. In complementary PWM mode, the GTDVD register is invalid and the GTDVU register is used as the dead time value during both up-counting and down-counting. No matter which GTDVU register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time. In Triangle-waves, The GTDVU register is used for up-counting. The GTDVD register is used for down-counting. In the case of a saw-waves, the GTDVU register controls the front dead time and the GTDVD register controls the rear dead time, regardless of whether the count is up or down. Setting a GTDVk value greater than or equal to GTPR is prohibited. In complementary PWM mode, set the GTDVU register to meet the following conditions: ● GTDVU > 0 ● GTDVU < GTPR/2 ● GTDVU + GTPR ≤ 0xffffffff When using the automatic dead time setting function, do not set a value that makes a change point of the waveform exceeding the count period. The change point of the negative-phase waveform, which is automatically calculated, is obtained by reading the GTCCRB register. When GTDVk is used, writing to GTCCRB is prohibited. Other than complementary PWM mode, when this register is set to 0, waveforms without dead time are output. When the GTDTCR.TDFER bit is 1, the writing to the GTDVD register has no effect. At this time, when the GTDVD register is read, the value for the GTDVU register is read. While GPT is running, changing the GTDVk values is prohibited. To change GTDVk to a new value, stop the GPT with the CST bit in the GTCR register.	R/W

22.2.29 GTDBk: General PWM Timer Dead Time Buffer Register k (k = U, D)

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x94 (GTDBU)
0x98 (GTDBD)

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

Bit	Symbol	Function	R/W
31:0	GTDBk[31:0]	The buffer register for the GTDVk register Access in 8-bit or 16-bit units to the GTDBk register is prohibited, and it should be accessed in 32-bit units.	R/W

22.2.30 GTSOS : General PWM Timer Output Protection Function Status Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0x9C

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

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

Bit	Symbol	Function	R/W
1:0	SOS[1:0]	Output Protection Function Status 0 0: Normal operation 0 1: Protected state (GTCCRA = 0 is set during transfer at trough or crest) 1 0: Protected state (GTCCRA ≥ GTPR is set during transfer at trough) 1 1: Protected state (GTCCRA ≥ GTPR is set during transfer at crest)	R
7:2	—	These bits are read as 0.	R
9:8	—	The read value is undefined.	R
31:10	—	These bits are read as 0.	R

The GTSOS register is a status register that indicates the status of the output protection function. The output protection function is enabled only when the dead time is automatically set (GTDTCR.TDE bit = 1) in triangle-wave mode.

SOS[1:0] bit (Output Protection Function Status)

This bit indicates the status of the output protection function in triangle-wave PWM mode. For details of the output protection function, see section 22.8.4. Output Protection Function for GTIOCnm Pin Output (n = 0 to 13; m = A, B).

22.2.31 GTSOTR : General PWM Timer Output Protection Function Temporary Release Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0xA0

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

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

Bit	Symbol	Function	R/W
0	SOTR	Output Protection Function Temporary Release 0: Protected state is not released 1: Protected state is released	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The GTSOTR register temporarily releases the protected state of GTIOCnB pin output (n = 0 to 13) when output protection has been set.

The protected state can be released only for the case of GTSOS.SOS[1:0] bits are 10b (protected state in which GTCCRA register ≥ GTPR register has occurred during transfer at trough). The protected state cannot be released for any other case.

SOTR bit (Output Protection Function Temporary Release)

This bit sets whether to temporarily release the protected state of the GTIOCnB pin output in an output protected state.

After the SOTR bit has been set to 1, the output protection function is canceled from the first trough. After the SOTR bit has been set to 0, output protection is resumed from the first trough.

22.2.32 GTADSMR : General PWM Timer A/D Conversion Start Request Signal Monitoring Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0xA4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	ADSMEN1	—	—	—	—	—	—	ADSMS1[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ADSMEN0	—	—	—	—	—	—	ADSMS0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ADSMS0[1:0]	A/D Conversion Start Request Signal Monitor 0 Selection 0 0: A/D conversion start request signal generated by the GTADTRA register during up-counting. 0 1: A/D conversion start request signal generated by the GTADTRA register during down-counting. 1 0: A/D conversion start request signal generated by the GTADTRB register during up-counting. 1 1: A/D conversion start request signal generated by the GTADTRB register during down-counting.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	ADSMEN0	A/D Conversion Start Request Signal Monitor 0 Output Enabling 0: Output of A/D conversion start request signal monitor 0 is disabled. 1: Output of A/D conversion start request signal monitor 0 is enabled.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
17:16	ADSMS1[1:0]	A/D Conversion Start Request Signal Monitor 1 Selection 0 0: A/D conversion start request signal generated by the GTADTRA register during up-counting. 0 1: A/D conversion start request signal generated by the GTADTRA register during down-counting. 1 0: A/D conversion start request signal generated by the GTADTRB register during up-counting. 1 1: A/D conversion start request signal generated by the GTADTRB register during down-counting.	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
24	ADSMEN1	A/D Conversion Start Request Signal Monitor 1 Output Enabling 0: Output of A/D conversion start request signal monitor 1 is disabled. 1: Output of A/D conversion start request signal monitor 1 is enabled.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

The GTADSMR register is used to control monitors for the A/D conversion start request signal that is synchronized with a frame period.

ADSMSk[1:0] bits (A/D Conversion Start Request Signal Monitor k Selection) (k = 0, 1)

These bits are used to select A/D conversion start request signal synchronized with a frame period which is monitored by the GTASMk pin.

In triangle-wave PWM mode or complementary PWM mode, the following settings are prohibited:

- Set ADSMSk[1:0] bit to 00b (A/D conversion start request during up-counting) when GTADTRA = 0
- Set ADSMSk[1:0] bit to 10b (A/D conversion start request during up-counting) when GTADTRB = 0
- Set ADSMSk[1:0] bit to 01b (A/D conversion start request during down-counting) when GTADTRA = GTPR
- Set ADSMSk[1:0] bit to 11b (A/D conversion start request during down-counting) when GTADTRB = GTPR

ADSMENk bit (A/D Conversion Start Request Signal Monitor k Output Enabling) (k = 0, 1)

This bit enables or disables the monitor output to the GTADSMk pin.

When the output is disabled, the GTADSMk pin goes to the low level.

When the bit is 1, the signal on the GTADSMk pin goes to the high level on assertion of the signal to request to the start of A/D conversion selected by the ADSMSk[1:0] bits and returns to the low level at the end of the current cycle of the timer for the channel that generated the given signal to request the start of A/D conversion. When the counter stops, the value when the counter stopped is retained for output. Set the ADSMENk bit to 0 to output the low level.

When a signal to request the start of A/D conversion is generated at the end of a timer period, the generation of this signal has priority in terms of monitoring output and the output remains at the high level till the end of the next period.

When the output of the same A/D conversion start request signal monitoring output is enabled for multiple channels, ORed signals will be output from the GPT.

22.2.33 GTEITC : General PWM Timer Extended Interrupt Skipping Counter Control Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0xA8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	EITCNT2[3:0]				EITCNT2IV[3:0]				EIVTT2[3:0]				—	—	EIVTC2[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EITCNT1[3:0]				—	—	—	—	EIVTT1[3:0]				—	—	EIVTC1[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	EIVTC1[1:0]	Extended Interrupt Skipping Counter 1 Count Source Select 0 0: Not counted (not skipped) 0 1: Counting both at overflow or underflow in saw-wave mode, and counting crests in triangle-wave mode or complementary PWM mode 1 0: Counting both at overflow or underflow in saw-wave mode, and counting troughs in triangle-wave mode or complementary PWM mode 1 1: Counting both at overflow or underflow in saw-wave mode, and counting both crests and troughs in triangle-wave mode or complementary PWM mode	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
7:4	EIVTT1[3:0]	Extended Interrupt Skipping 1 Skipping Count Setting Skipping count for the extended interrupt skipping 1	R/W
11:8	—	These bits are read as 0. The write value should be 0.	R/W
15:12	EITCNT1[3:0]	Extended Interrupt Skipping Counter 1	R
17:16	EIVTC2[1:0]	Extended Interrupt Skipping Counter 2 Count Source select 0 0: Not counted (not skipped) 0 1: Counting both at overflow or underflow in saw-wave mode, and counting crests in triangle-wave mode or complementary PWM mode 1 0: Counting both at overflow or underflow in saw-wave mode, and counting troughs in triangle-wave mode or complementary PWM mode 1 1: Counting both at overflow or underflow in saw-wave mode, and counting both crests and troughs in triangle-wave mode or complementary PWM mode	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
23:20	EIVTT2[3:0]	Extended Interrupt Skipping 2 Skipping Count Setting Skipping count for the extended interrupt skipping 2	R/W
27:24	EITCNT2IV[3:0]	Extended Interrupt Skipping Counter 2 Initial Value	R/W*1
31:28	EITCNT2[3:0]	Extended Interrupt Skipping Counter 2	R

Note 1. The EITCNT2IV[3:0] bits are only writable when the value other than 00b is written to the EIVTC2[1:0] bits that have been 00b.

GTEITC register sets the extended interrupt skipping function to skip the interrupts, A/D conversion start requests, and buffer transfers independently by counting at overflow and underflow in the GTCNT counter.

The setting is operated independently of the interrupt skipping by the GTITC register or the GTADCMSC register.

The setting is invalid during the event count operation.

Access in 8-bit units to GTEITC is prohibited.

EIVTCk[1:0] bits (Extended Interrupt Skipping Counter k Count Source Select) (k = 1, 2)

These bits select the way of counting for the extended interrupt skipping counter k.

Setting only with these bits does not skip the interrupts, A/D conversion start requests, and buffer transfers. Skipping function for the interrupt, A/D conversion start request, and buffer transfer, all of which are a target of skipping, is set individually with the GTEITLI1, GTEITLI2, and GTEITLB registers.

EIVTTk[3:0] bits (Extended Interrupt Skipping k Skipping Count Setting) (k = 1, 2)

A count for the period with continuous skipping is set as a skipping count, where a period is from a generation of a count source selected by the EIVTCk[1:0] bits to the next generation of the count source.

When the count source is generated while the EIVTTk[3:0] bits match the EITCNTk[3:0] bits, the EITCNTk[3:0] bits are cleared.

When these bits are 0x0, skipping is not performed.

EITCNT1[3:0] bits (Extended Interrupt Skipping Counter 1)

The counting is incremented by 1 every time a count source (overflow/underflow/crest/trough) selected by the EIVTC1[1:0] bits is generated.

Counting is performed periodically within the range between 0 and the EIVTT1[3:0] bits.

Even if the GTCNT counter is stopped, the value is not cleared, and value at stop of the GTCNT counter is retained.

[Clearing conditions]

- 00b is written to the EIVTC1[1:0] bits.
- 0x0 is written to the EIVTT1[3:0] bits.
- A count source (overflow/underflow/crest/trough) selected by the EIVTC1[1:0] bits is generated when the extended interrupt skipping 1 skipping count set by the EIVTT1[3:0] bits match the value for the EITCNT1[3:0]

EITCNT2IV[3:0] bit (Extended Interrupt Skipping Counter 2 Initial Value)

These bits are the value of the initial value for the extended interrupt skipping counter 2.

Writing to the EITCNT2IV[3:0] is performed only when the writing value to the EIVTC2[1:0] bits are other than 00b and when the GTEITC register is written by the access of upper 16 bits or 32 bits while the EITCNT2[3:0] bits are set not to count (EIVTC2[1:0] bits are 00b). When the EITCNT2IV[3:0] bits are written, the value written to the EITCNT2IV[3:0] bits is written to the EITCNT2[3:0] bits simultaneously.

The writing to the EITCNT2IV[3:0] bits are ignored when the EITCNT2[3:0] bits are set to count (EIVTC2[1:0] bits are other than 00b) or perform the setting of not to count (00b is written to the EIVTC2[1:0] bits).

The EITCNT2IV[3:0] bits are not reset by the writing 00b to the EIVTC2[1:0] bits.

EITCNT2[3:0] bit (Extended Interrupt Skipping Counter 2)

The counting is incremented by 1 every time a count source (overflow/underflow/crest/trough) selected by the EIVTC2[1:0] bits is generated.

Counting is performed periodically within the range between 0 and the EIVTT2[3:0] bits.

Even if the GTCNT counter is stopped, the value is not cleared, and value at stop of the GTCNT counter is retained.

Setting of the initial value for the EITCNT2[3:0] bits are performed only when the writing value to the EIVTC2[1:0] bits are other than 00b and when the GTEITC register is written by the access of upper 16 bits or 32 bits while the extended interrupt skipping counter 2 is set as not to count (EIVTC2[1:0] bits are 00b).

When the initial value is set, the written value to the EITCNT2IV[3:0] bits is written to the EITCNT2[3:0] bits as the initial value.

[Clearing condition]

- 00b is written to the EIVTC2[1:0] bits.
- 0x0 is written to the EIVTT2[3:0] bits.
- The value other than 00b is written to the EIVTC2[1:0] bits and the 0x0 is written to the EITCNT2IV[3:0] bits simultaneously while 00b is set to the EIVTC2[1:0] bits.
- A count source (overflow/underflow/crest/trough) selected by the EIVTC2[1:0] bits is generated when the extended interrupt skipping 2 skipping count set by the EIVTT2[3:0] bits match the value for the EITCNT2[3:0].

22.2.34 GTEITLI1 : General PWM Timer Extended Interrupt Skipping Setting Register 1

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0xAC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	EITLU[2:0]			—	EITLV[2:0]			—	EITLF[2:0]			—	EITLE[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	EITLA[2:0]	GTCCRA Register Compare Match/Input Capture Interrupt Extended Skipping Function Select See Table 22.6.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	EITLB[2:0]	GTCCRB Register Compare Match/Input Capture Interrupt Extended Skipping Function Select See Table 22.6.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
10:8	EITLC[2:0]	GTCCRC Register Compare Match Interrupt Extended Skipping Function Select See Table 22.6.	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
14:12	EITLD[2:0]	GTCCRD Register Compare Match Interrupt Extended Skipping Function Select See Table 22.6.	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
18:16	EITLE[2:0]	GTCCRE Register Compare Match Interrupt Extended Skipping Function Select See Table 22.6.	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
22:20	EITLF[2:0]	GTCCRF Register Compare Match Interrupt Extended Skipping Function Select See Table 22.6.	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
26:24	EITLV[2:0]	Overflow Interrupt Extended Skipping Function Select See Table 22.6.	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
30:28	EITLU[2:0]	Underflow Interrupt Extended Skipping Function Select See Table 22.6.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The GTEITLI1 register sets the extended skipping function for interrupts such as compare match/input capture, overflow and underflow.

Setting only with this register does not perform skipping. The GTEITC register should be set so that a corresponding extended interrupt skipping counter operates the counting.

The setting is operated independently of the interrupt skipping by the GTITC register or the GTADCMSC register.

The setting is invalid during the event count operation.

EITLm[2:0] bits (GTCCRm Register Compare Match/Input Capture Interrupt Extended Skipping Function Select) (m = A, B)

These bits select the extended interrupt skipping function to skip the compare match/input capture interrupt (GPTn_CCMPm) in the GTCCRm register. See Table 22.6.

EITLx[2:0] bits (GTCCRx Register Compare Match Interrupt Extended Skipping Function Select) (x = C, D, E, F)

These bits select the extended interrupt skipping function to skip the compare match interrupt (GPTn_CMPx) in the GTCCRx register. See Table 22.6.

EITLV[2:0] bit (Overflow Interrupt Extended Skipping Function Select)

These bits select the extended interrupt skipping function to skip the interrupt at overflow(GPTn_OVF). See Table 22.6.

EITLU[2:0] bit (Underflow Interrupt Extended Skipping Function Select)

These bits select the extended interrupt skipping function to skip the interrupt at underflow (GPTn_UDF). See Table 22.6.

Table 22.6 Setting the Function select for the GTEITLI1

EITLy[2:0]	Function
0 0 0	Do not perform an extended interrupt skipping
0 0 1	Skip an interrupt in the period when the value for the extended interrupt skipping counter 1 is other than 0 (An interrupt is output in the period of the EITCNT1[3:0] bits = 0)
0 1 0	Skip an interrupt in the period when the value for the extended interrupt skipping counter 2 is other than 0 (An interrupt is output in the period of the EITCNT2[3:0] bits = 0)
0 1 1	Skip an interrupt in the period the value for the extended interrupt skipping counter 1 or 2 is other than 0 (An interrupt is output in the period of the EITCNT1[3:0] bits = 0 and the EITCNT2[3:0] bits = 0)
1 0 0	Setting prohibited
1 0 1	Skip an interrupt in the period when the value for the extended interrupt skipping counter 1 is other than the skipping count (An interrupt is output in the period of the EITCNT1[3:0] bits = the EIVTT1[3:0] bits)
1 1 0	Skip an interrupt in the period when the value for the extended interrupt skipping counter 2 is other than the skipping count (An interrupt is output in the period of the EITCNT2[3:0] bits = the EIVTT2[3:0] bits)
1 1 1	Skip an interrupt in the period when the value for the extended interrupt skipping counter 1 or 2 is other than the skipping count (An interrupt is output in the period of the EITCNT1[3:0] bits = the EIVTT1[3:0] bits and the EITCNT2 [3:0] bits = the EIVTT2[3:0] bits)

Note:
- y = A, B, C, D, E, F, V, U
- When the intended skipping counter is set as not to count (the EIVTCk[1:0] bits = 00b or the EIVTTk[3:0] bits = 0x0), skipping is not performed.(k = 1, 2)
- When the EITLy[2:0] bits are set to 011b or 111b, and when one of the skipping counter 1 or 2 is set as not to count, skipping is not performed.

22.2.35 GTEITLI2 : General PWM Timer Extended Interrupt Skipping Setting Register 2

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0xB0

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

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

Bit	Symbol	Function	R/W
2:0	EADTAL[2:0]	GTADTRA Register A/D Conversion Start Request Extended Skipping Function Select See Table 22.7.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	EADTBL[2:0]	GTADTRB Register A/D Conversion Start Request Extended Skipping Function Select See Table 22.7.	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

The GTEITLI2 register sets the extended skipping function for A/D conversion start requests.

Setting only with this register does not perform skipping. The GTEITC register should be set so that a corresponding extended interrupt skipping counter operates the counting.

The setting is operated independently of the interrupt skipping by the GTITC register or the GTADCMSC register.

The setting is invalid during the event count operation.

EADTmL[2:0] bits (GTADTRm Register A/D Conversion Start Request Extended Skipping Function Select) (m = A, B)

These bits select the extended skipping function to skip A/D conversion start requests for the compare match in the GTADTRm register. See Table 22.7.

Table 22.7 Setting the Function Select for the GTEITLI2 Register

EADTmL[2:0]	Function
0 0 0	Do not perform an extended interrupt skipping
0 0 1	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 1 is other than 0 (An A/D conversion start request is output in the period of the EITCNT1[3:0] bits = 0)
0 1 0	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 2 is other than 0 (An A/D conversion start request is output in the period of the EITCNT2[3:0] bits = 0)
0 1 1	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 1 or 2 is other than 0 (An A/D conversion start request is output in the period of the EITCNT1[3:0] bits = 0 and the EITCNT2[3:0] bits = 0)
1 0 0	Setting prohibited
1 0 1	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 1 is other than the skipping count (An A/D conversion start request is output in the period of the EITCNT1[3:0] bits = the EIVTT1[3:0] bits)
1 1 0	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 2 is other than the skipping count (An A/D conversion start request is output in the period of the EITCNT2[3:0] bits = the EIVTT2[3:0] bits)
1 1 1	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 1 or 2 is other than the skipping count (An A/D conversion start request is output in the period of the EITCNT1[3:0] bits = the EIVTT1[3:0] bits and the EITCNT2[3:0] bits = the EIVTT2[3:0] bits)

Note:
- m = A, B
- When the intended skipping counter is set as not to count (the EIVTCk[1:0] bits = 00b or the EIVTTk[3:0] bits = 0x0), skipping is not performed. (k = 1, 2)
- When the EADTmL[2:0] bits are set to 011b or 111b, and when one of the skipping counter 1 or 2 is set as not to count, skipping is not performed.

22.2.36 GTEITLB : General PWM Timer Extended Buffer Transfer Skipping Setting Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0xB4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	EBTLDVD[2:0]			—	EBTLDVU[2:0]			—	EBTLADB[2:0]			—	EBTLADA[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	EBTLCA[2:0]	GTCCRA Register Buffer Transfer Extended Skipping Function Select See Table 22.8.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	EBTLCB[2:0]	GTCCRB Register Buffer Transfer Extended Skipping Function Select See Table 22.8.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
10:8	EBTLPR[2:0]	GTPR Register Buffer Transfer Extended Skipping Function Select See Table 22.8.	R/W*1
15:11	—	These bits are read as 0. The write value should be 0.	R/W
18:16	EBTLADA[2:0]	GTADTRA Register Buffer Transfer Extended Skipping Function Select See Table 22.8.	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
22:20	EBTLADB[2:0]	GTADTRB Register Buffer Transfer Extended Skipping Function Select See Table 22.8.	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
26:24	EBTLDVU[2:0]	GTDVU Register Buffer Transfer Extended Skipping Function Select See Table 22.8.	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
30:28	EBTLDVD[2:0]	GTDVD Register Buffer Transfer Extended Skipping Function Select See Table 22.8.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

The GTEITLB register sets the extended skipping function for buffer transfers.

Setting only this register does not perform skipping. The GTEITC register should be set so that a corresponding extended interrupt skipping counter operates the counting.

The setting is operated independently of the interrupt skipping by the GTITC register.

The buffer transfer from the GTOLBR register to the GTIOR.GTIOA[4:0], GTIOB[4:0] bits is not target of the extended buffer transfer skipping function.

The setting is invalid during the event count operation.

EBTLCA[2:0] bit (GTCCRA Register Buffer Transfer Extended Skipping Function Select)

These bits select the extended buffer transfer skipping function to skip buffer transfers (transfers among the GTCCRA, GTCCRC, GTCCRD registers and a temporary register A) in the GTCCRA register. See Table 22.8.

An extended skipping of buffer transfers in the GTCCRA register is valid for forcible buffer transfers by the GTBER.CCRSWT bit while the count operation is stopped. Forcible buffer transfers in the GTCCRA register should be performed in the condition of not performing the extended buffer transfer skipping.

The buffer transfer between the GTCCRC, GTCCRE, and GTCCRA in complementary PWM mode cannot be skipped.

EBTLCB[2:0] bit (GTCCRB Register Buffer Transfer Extended Skipping Function Select)

These bits select the extended buffer transfer skipping function to skip buffer transfers (transfers among the GTCCRB, GTCCRE, GTCCRF registers and a temporary register B) in the GTCCRB register. See Table 22.8.

An extended skipping of buffer transfers in the GTCCRB register is valid for forcible buffer transfers by the GTBER.CCRSWT bit while the count operation is stopped. Forcible buffer transfers in the GTCCRB register should be performed in the condition of not performing the extended buffer transfer skipping.

EBTLPR[2:0] bit (GTPR Register Buffer Transfer Extended Skipping Function Select)

These bits select the extended buffer transfer skipping function to skip buffer transfers (transfers among the GTPR, GTPBR, and GTPDBR registers) in the GTPR register. See Table 22.8.

If the buffer transfer of GTPR is skipped in complementary PWM mode, GTEITC setting of slave channel should be matched to the master channel so that the buffer transfer timing of the slave channel matches the master channel.

EBTLADm[2:0] bits (GTADTRm Register Buffer Transfer Extended Skipping Function Select) (m = A, B)

These bits select the extended buffer transfer skipping function to skip buffer transfers (transfers among the GTADTRm, GTADTBRm, and GTADTDBRm registers) in the GTADTRm register. See Table 22.8.

EBTLDVm[2:0] bits (GTDVm Register Buffer Transfer Extended Skipping Function Select) (m = U, D)

These bits select the extended buffer transfer skipping function to skip buffer transfers (transfers between the GTDVm and GTDBm registers) in the GTDVm register. See Table 22.8.

Table 22.8 Setting the Function Select for the GTEITLB Register

EBTLx[2:0] bits	Function
0 0 0	Do not perform an extended interrupt skipping
0 0 1	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 is other than 0 (Buffer is transferred in the period of the EITCNT1[3:0] bits = 0)
0 1 0	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 2 is other than 0 (Buffer is transferred in the period of the EITCNT2[3:0] bits = 0)
0 1 1	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 or 2 is other than 0 (Buffer is transferred in the period of the EITCNT1[3:0] bits = 0 and the EITCNT2[3:0] bits = 0)
1 0 0	Setting prohibited
1 0 1	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 is other than the skipping count (Buffer is transferred in the period of the EITCNT1[3:0] bits = the EIVTT1[3:0] bits)
1 1 0	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 2 is other than the skipping count (Buffer is transferred in the period of the EITCNT2[3:0] bits = the EIVTT2[3:0] bits)
1 1 1	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 or 2 is other than the skipping count (Buffer is transferred in the period of the EITCNT1[3:0] bits = the EIVTT1[3:0] bits and the EITCNT2[3:0] bits = the EIVTT2[3:0] bits)

Note:
- x = CA, CB, PR, ADA, ADB, DVU, DVD
- When the intended skipping counter is set as not to count (EIVTCk[1:0]bits = 00b or EIVTTk[3:0] bits = 0x0), skipping is not performed. (k = 1, 2)
- When the EBTLx[2:0] bits are set to 011b or 111b, and when one of the skipping counter 1 or 2 is set as not to count, skipping is not performed.

22.2.37 GTICLF : General PWM Timer Inter Channel Logical Operation Function Setting Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0xB8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	ICLFSELD[5:0]						—	ICLFB[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	ICLFA[2:0]	GTIOCnA Output Logical Operation Function Select 0 0 0: A (no delay) 0 0 1: NOT A (no delay) 0 1 0: C (1GTCLK delay) 0 1 1: NOT C (1GTCLK delay) 1 0 0: A AND C (1GTCLK delay)*2 1 0 1: A OR C (1GTCLK delay)*2 1 1 0: A EXOR C (1GTCLK delay)*2 1 1 1: A NOR C (1GTCLK delay)*2	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
9:4	ICLFSELC[5:0]	Inter Channel Signal C Select*1*2 0x00: GTIOC0A 0x01: GTIOC0B 0x02: GTIOC1A 0x03: GTIOC1B ⋮ 0x12: GTIOC9A 0x13: GTIOC9B Others: Setting prohibited	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
18:16	ICLFB[2:0]	GTIOCnB Output Logical Operation Function Select 0 0 0: B (no delay) 0 0 1: NOT B (no delay) 0 1 0: D (1GTCLK delay) 0 1 1: NOT D (1GTCLK delay) 1 0 0: B AND D (1GTCLK delay)*3 1 0 1: B OR D (1GTCLK delay)*3 1 1 0: B EXOR D (1GTCLK delay)*3 1 1 1: B NOR D (1GTCLK delay)*3	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
25:20	ICLFSELD[5:0]	Inter Channel Signal D Select*1*3 0x00: GTIOC0A 0x01: GTIOC0B 0x02: GTIOC1A 0x03: GTIOC1B 0x04: GTIOC2A 0x05: GTIOC2B 0x06: GTIOC3A 0x07: GTIOC3B ⋮ ⋮ 0x3E: GTIOC31A 0x3F: GTIOC31B	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The signal before performing output disable control is selected.
Note 2. When channel's own GTIOCnA is selected, C is treated as "1".
Note 3. When channel's own GTIOCnB is selected, D is treated as "1".

The GTICLF register sets the logical operation function between compare match outputs. The logical operation is performed with the signals that the duty 0%/100% control is performed after compare match control. (The output disable control is performed with the signal after logical operation.)

Access in 8-bit units to GTICLF is prohibited.

ICLFm[2:0] bit (GTIOCnm Output Logical Operation Function Select) (m = A, B)

These bits select the logical operation function between signals before performing output disable control for GTIOCnm. To prevent hazard to the GPT output, the signal after logical operation is latched with GTCLK. After latching, the output disable control is performed. When the logical operation function which causes the delay of 1 GTCLK is selected, the output enable signal is also delayed with 1 GTCLK and input to the output disable control.

When the same signal to operate logical function AND, OR, EXOR and NOR is selected, one signal is treated as "1".

ICLFSELk[5:0] bit (Inter Channel Signal k Select) (k = C, D)

These bits select the signal k that the logical operation is performed with the signal before performing output disable control for GTIOCnm.

22.2.38 GTPC : General PWM Timer Period Count Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 3, 10 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0xBC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	PCNT[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	PCEN	Period Count Function Enable 0: Period count function is disabled 1: Period count function is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
8	ASTP	Automatic Stop Function Enable 0: Automatic stop function is disabled 1: Automatic stop function is enabled	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
27:16	PCNT[11:0]	Period Counter Counter for the number of period	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The GTPC register counts the number of period.

PCEN bit (Period Count Function Enable)

This bit enables or disables period count function.

Writing is available when counting is both in progress and stopped.

When 1 is written to either the GTSECR.SPCE bit or the GTSECR.SPCD bit, the value is simultaneously set to the PCEN bit in the channels set to 1 by the GTSECSR register.

ASTP bit (Automatic Stop Function Enable)

This bit enables or disables the GTCNT counter automatic stopping after finishing counting the number of period.

When the PCEN bis is 0, writing is available.

When the PCEN bit is 1, writing is disabled.

When the PCEN bit is 1, the ASTP bit is 1, and the PCNT counter is stopped at PCNT = 0, the GTCNT counter is also stopped. When the ASTP bit is 0, the GTCNT counter continues to count.

PCNT[11:0] bit (Period Counter)

This counter counts the number of period.

When the PCEN bis is 0, writing the number of period is available.

When the PCEN bit is 1, writing is disabled, and down-counting is performed at the end of period. In saw-wave mode, the end of period refers to overflow, underflow, or counter clearing. In triangle-wave mode or complementary PWM mode, it refers to trough.

When the PCNT counter is 1 at the end of period, it becomes 0 and counting is stopped.

When the GTCNT counter is stopped while period count function is enabled, the PCNT counter keeps its value. When the GTCNT counter restarts counting and the PCEN bit is 1, the PCNT counter restarts down-counting from the hold value.

When the PCEN bit is changed from 0 to 1 while the PCNT counter is 0 and the ASTP bit is 1, the GTCNT counter is stopped at the count clock immediately after that.

22.2.39 GTADCMSC : General PWM Timer A/D Conversion Start Request Compare Match Skipping Control Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0xC0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ADCMSCNT2[3:0]				ADCMSCNT2IV[3:0]				ADCMST2[3:0]				—	—	ADCMSC2[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADCMSCNT1[3:0]				ADCMSCNT1IV[3:0]				ADCMST1[3:0]				—	—	ADCMSC1[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ADCMSC1[1:0]	A/D Conversion Start Request Compare Match Skipping Counter 1 Count Source Select 0 0: Not counted (not skipped) 0 1: Counting GTADTRA register compare match 1 0: Counting GTADTRB register compare match 1 1: Counting both GTADTRA register compare match and GTADTRB register compare match	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
7:4	ADCMST1[3:0]	A/D Conversion Start Request Compare Match Skipping 1 Skipping Count Setting Skipping count for the A/D conversion start request compare match skipping 1.	R/W
11:8	ADCMSCNT1IV[3:0]	A/D Conversion Start Request Compare Match Skipping Counter 1 Initial Value	R/W*1
15:12	ADCMSCNT1[3:0]	A/D Conversion Start Request Compare Match Skipping Counter 1	R
17:16	ADCMSC2[1:0]	A/D Conversion Start Request Compare Match Skipping Counter 2 Count Source Select 0 0: Not counted (not skipped) 0 1: Counting GTADTRA register compare match 1 0: Counting GTADTRB register compare match 1 1: Counting both GTADTRA register compare match and GTADTRB register compare match	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
23:20	ADCMST2[3:0]	A/D Conversion Start Request Compare Match Skipping 2 Skipping Count Setting Skipping count for the A/D conversion start request compare match skipping 2	R/W
27:24	ADCMSCNT2IV[3:0]	A/D Conversion Start Request Compare Match Skipping Counter 2 Initial Value	R/W*2
31:28	ADCMSCNT2[3:0]	A/D Conversion Start Request Compare Match Skipping Counter 2	R

Note 1. Writing is possible only when the ADCMSC1[1:0] bits are 00b and a value other than 00b is written to the ADCMSC1[1:0] bits.
Note 2. Writing is possible only when the ADCMSC2[1:0] bits are 00b and a value other than 00b is written to the ADCMSC2[1:0] bits.

The GTADCMSC register is a register that controls the skipping counter of the A/D conversion start request compare match skipping function that counts compare matches of GTADTRA register and GTADTRB register, and skipping A/D conversion start request and buffer transfer independently.

This register setting is operated independently of the interrupt skipping by the GTITC register or GTEITC register.

Access in 8-bit units to GTADCMSC is prohibited.

ADCMSCk[1:0] bits (A/D Conversion Start Request Compare Match Skipping Counter k Count Source Select) (k = 1, 2)

These bits select the way of counting for the A/D conversion start request compare match skipping counter k.

Setting only with these bits does not skip the A/D conversion start requests and buffer transfers. Skipping function for the A/D conversion start request and buffer transfer which are a target of skipping is set individually with the GTADCMSS register.

ADCMSTk[3:0] bits (A/D Conversion Start Request Compare Match Skipping k Skipping Count Setting) (k = 1, 2)

A count for the period with continuous skipping is set as a skipping count, where a period is from a generation of a count source selected by the ADCMSCk[1:0] bits to the next generation of the count source.

When the count source is generated while the ADCMSTk[3:0] bits match the ADCMSCNTk[3:0] bits, the ADCMSCNTk[3:0] bits are cleared.

When these bits are 0x0, skipping is not performed.

ADCMSCNTkIV[3:0] bits (A/D Conversion Start Request Compare Match Skipping Counter k Initial Value) (k = 1, 2)

These bits are the value of the initial value for the A/D conversion start request compare match skipping counter k.

Writing to the ADCMSCNTkIV[3:0] is performed only when the writing value to the ADCMSCk[1:0] bits are other than 00b and when the GTADCMSC register is written by the access of 16 bits or 32 bits while the ADCMSCNTk[3:0] bits are set not to count (ADCMSCk[1:0] bits are 00b). When the ADCMSCNTkIV[3:0] bits are written, the value written to the ADCMSCNTkIV[3:0] bits is written to the ADCMSCNTk[3:0] bits simultaneously.

The writing to the ADCMSCNTkIV[3:0] bits are ignored when the ADCMSCNTk[3:0] bits are set to count (ADCMSCk[1:0] bits are other than 00b) or perform the setting of not to count (00b is written to the ADCMSCk[1:0] bits).

The ADCMSCNTkIV[3:0] bits are not reset by the writing 00b to the ADCMSCk[1:0] bits.

ADCMSCNTk[3:0] bits (A/D Conversion Start Request Compare Match Skipping Counter k) (k=1,2)

The counting is incremented by 1 every time a count source selected by the ADCMSCk[1:0] bits is generated.

Counting is performed periodically within the range between 0 and the ADCMSTk[3:0] bits.

Even if the GTCNT counter is stopped, the value is not cleared, and value at stop of the GTCNT counter is retained.

Setting of the initial value for the ADCMSCNTk[3:0] bits are performed only when the writing value to the ADCMSCk[1:0] bits are other than 00b and when the GTADCMSC register is written by the access of 16 bits or 32 bits while the A/D conversion start request compare match skipping counter k is set as not to count (ADCMSCk[1:0] bits are 00b).

When the initial value is set, the written value to the ADCMSCNTkIV[3:0] bits is written to the ADCMSCNTk[3:0] bits as the initial value.

[Clearing condition]

- 00b is written to the ADCMSCk[1:0] bits.
- 0x0 is written to the ADCMSTk[3:0] bits.
- The value other than 00b is written to the ADCMSCk[1:0] bits and the 0x0 is written to the ADCMSCNTkIV[3:0] bits simultaneously while 00b is set to the ADCMSCk[1:0] bits.
- A count source selected by the ADCMSCk[1:0] bits is generated when the extended interrupt skipping 2 skipping count set by the ADCMSTk[3:0] bits match the value for the ADCMSCNTk[3:0].

22.2.40 GTADCMSS : General PWM Timer A/D Conversion Start Request Compare Match Skipping Setting Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0xC4

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

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

Bit	Symbol	Function	R/W
2:0	ADCMSAL[2:0]	GTADTRA Register A/D Conversion Start Request Compare Match Skipping Function Select See Table 22.9.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	ADCMSBL[2:0]	GTADTRB Register A/D Conversion Start Request Compare Match Skipping Function Select See Table 22.9.	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W
18:16	ADCMBSA[2:0]	GTADTRA Register Buffer Transfer by A/D Conversion Start Request Compare Match Skipping Function Select See Table 22.10.	R/W

Bit	Symbol	Function	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
22:20	ADCMBSB[2:0]	GTADTRB Register Buffer Transfer by A/D Conversion Start Request Compare Match Skipping Function Select See Table 22.10.	R/W
31:23	—	These bits are read as 0. The write value should be 0.	R/W

The GTADCMSS register is a register that selects the A/D conversion start request compare match skipping function or the GTADTRm (m = A, B) register buffer transfer by A/D conversion start request compare match skipping function.

Setting only with this register does not perform skipping. The GTADCMSC register should be set so that a corresponding A/D conversion start request compare match skipping counter operates the counting.

This register setting is operated independently of the interrupt skipping by the GTITC register or GTEITC register.

ADCMSmL[2:0] bits (GTADTRm Register A/D Conversion Start Request Compare Match Skipping Function Select) (m = A, B)

These bits select the A/D conversion start request compare match skipping function of the GTADTRm (m = A, B) register.Table 22.9.

ADCMBSm[2:0] bits (GTADTRm Register Buffer Transfer by A/D Conversion Start Request Compare Match Skipping Function Select) (m = A, B)

These bits select the GTADTRm (m = A, B) register buffer transfer by A/D conversion start request compare match skipping function which skips the buffer transfer of GTADTRm (m = A, B) register (Transfer between GTADTRm register, GTADTBRm register, GTADTDBRm register). See Table 22.10.

Table 22.9 Setting of the GTADTRm Register A/D Conversion Start Request Compare Match Skipping Function Select bit (m = A, B)

ADCMSmL[2:0]	Function
0 0 0	Do not perform an A/D conversion start request compare match skipping
0 0 1	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 1 is other than 0 (An A/D conversion start request is output in the period of the ADCMSCNT1[3:0] bits = 0)
0 1 0	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 2 is other than 0 (An A/D conversion start request is output in the period of the ADCMSCNT2[3:0] bits = 0)
0 1 1	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 1 or 2 is other than 0 (An A/D conversion start request is output in the period of the ADCMSCNT1[3:0] bits = 0 and the ADCMSCNT2[3:0] bits = 0)
1 0 0	Setting prohibited
1 0 1	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 1 is other than the skipping count (An A/D conversion start request is output in the period of the ADCMSCNT1[3:0] bits = the ADCMST1[3:0] bits)
1 1 0	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 2 is other than the skipping count (An A/D conversion start request is output in the period of the ADCMSCNT2[3:0] bits = the ADCMST2[3:0] bits)
1 1 1	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 1 or 2 is other than the skipping count (An A/D conversion start request is output in the period of the ADCMSCNT1[3:0] bits = the ADCMST1[3:0] bits and the ADCMSCNT2[3:0] bits = the ADCMST2[3:0] bits)

Note:
- m = A, B
- When the intended skipping counter is set as not to count (the ADCMSCk[1:0] bits = 00b or the ADCMSTk[3:0] bits = 0x0), skipping is not performed. (k = 1, 2)
- When the ADCMSmL[2:0] bits are set to 011b or 111b, and when one of the skipping counters is set as not to count, skipping is not performed.

Table 22.10 Setting of the GTADTRm Register Buffer Transfer by A/D Conversion Start Request Compare Match Skipping Function Select bit (m = A, B)

ADCMBSm[2:0]	Function
0 0 0	Do not perform an GTADTRm register buffer transfer by A/D conversion start request compare match skipping
0 0 1	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 1 is other than 0 (Buffer is transferred in the period of the ADCMSCNT1[3:0] bits = 0)
0 1 0	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 2 is other than 0 (Buffer is transferred in the period of the ADCMSCNT2[3:0] bits = 0)
0 1 1	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 1 or 2 is other than 0 (Buffer is transferred in the period of the ADCMSCNT1[3:0] bits = 0 and the ADCMSCNT2[3:0] bits = 0)
1 0 0	Setting prohibited
1 0 1	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 1 is other than the skipping count (Buffer is transferred in the period of the ADCMSCNT1[3:0] bits = the ADCMST1[3:0] bits)
1 1 0	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 2 is other than the skipping count (Buffer is transferred in the period of the ADCMSCNT2[3:0] bits = the ADCMST2[3:0] bits)
1 1 1	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 1 or 2 is other than the skipping count (A Buffer is transferred in the period of the ADCMSCNT1[3:0] bits = the ADCMST1[3:0] bits and the ADCMSCNT2[3:0] bits = the ADCMST2[3:0] bits)

Note:
- m = A, B
- When the intended skipping counter is set as not to count (the ADCMSCk[1:0] bits = 00b or the ADCMSTk[3:0] bits = 0x0), skipping is not performed. (k = 1, 2)
- When the ADCMSmL[2:0] bits are set to 011b or 111b, and when one of the skipping counters is set as not to count, skipping is not performed.

22.2.41 GTSECSR : General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0xD0

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	SECSEL13	SECSEL12	SECSEL11	SECSEL10	SECSEL9	SECSEL8	SECSEL7	SECSEL6	SECSEL5	SECSEL4	SECSEL3	SECSEL2	SECSEL1	SECSEL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SECSEL0	Channel 0 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
1	SECSEL1	Channel 1 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
2	SECSEL2	Channel 2 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W

Bit	Symbol	Function	R/W
3	SECSEL3	Channel 3 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
4	SECSEL4	Channel 4 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
5	SECSEL5	Channel 5 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
6	SECSEL6	Channel 6 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
7	SECSEL7	Channel 7 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
8	SECSEL8	Channel 8 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
9	SECSEL9	Channel 9 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
10	SECSEL10	Channel 10 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
11	SECSEL11	Channel 11 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
12	SECSEL12	Channel 12 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
13	SECSEL13	Channel 13 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

The GTSECSR register selects an intended channel n (n = 0 to 13) for updating an operation enable bit by the GTSECR register. A bit position for the GTSECSR register indicates a channel number. The GTSECSR register of each channel is a common register, and writing 1 to a bit in the GTSECSR register in any channel and updating it changes a channel, related to the position of the bit written with 1 by the GTSECSR register, to be simultaneously controlled of the operation enable bit by the GTSECR register.

Contrary to a security attribute for write or read access, if the security and privilege attributes set on each channel has the security violation or the privilege violation, the bit of the channel number that violates security cannot be written or read, and the read value of the bit is 0.

Access in 8-bit or 16-bit units to GTSECSR is prohibited, and it should be accessed in 32-bit units.

SECSELn bit (Operation Enable Bit Simultaneous Control Channel Select) (n = 0 to 13)

This bit enables or disables the simultaneous control of operation enable in channel n.

When the bit is set to 1, the simultaneous control is enabled, and disabled when the bit is 0.

22.2.42 GTSECR : General PWM Timer Operation Enable Bit Simultaneous Control Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0xD4

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	SBDDD	SBDAD	SBDPD	SBDCD	—	—	—	—	SBDDE	SBDAE	SBDPE	SBDCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SBDCE	GTCCR Register Buffer Operation Simultaneous Enable 0: Disable simultaneous enabling GTCCR buffer operations 1: Enable GTCCR register buffer operations simultaneously	R/W
1	SBDPE	GTPR Register Buffer Operation Simultaneous Enable 0: Disable simultaneous enabling GTPR buffer operations 1: Enable GTPR register buffer operations simultaneously	R/W*1
2	SBDAE	GTADTR Register Buffer Operation Simultaneous Enable 0: Disable simultaneous enabling GTADTR buffer operations 1: Enable GTADTR register buffer operations simultaneously	R/W
3	SBDDE	GTDV Register Buffer Operation Simultaneous Enable 0: Disable simultaneous enabling GTDV buffer operations 1: Enable GTDV register buffer operations simultaneously	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	SBDCD	GTCCR Register Buffer Operation Simultaneous Disable 0: Disable simultaneous disabling GTCCR buffer operations 1: Disable GTCCR register buffer operations simultaneously	R/W
9	SBDPD	GTPR Register Buffer Operation Simultaneous Disable 0: Disable simultaneous disabling GTPR buffer operations 1: Disable GTPR register buffer operations simultaneously	R/W*1
10	SBDAD	GTADTR Register Buffer Operation Simultaneous Disable 0: Disable simultaneous disabling GTADTR buffer operations 1: Disable GTADTR register buffer operations simultaneously	R/W
11	SBDDD	GTDV Register Buffer Operation Simultaneous Disable 0: Disable simultaneous disabling GTDV buffer operations 1: Disable GTDV register buffer operations simultaneously	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
16	SPCE	Period Count Function Simultaneous Enable*2 0: Disable simultaneous enabling period count function 1: Enable period count function simultaneously	R/W
17	SSCE	Synchronous Set/Clear Simultaneous Enable*3 0: Disable simultaneous enabling synchronous set/clear 1: Enable synchronous set/clear simultaneously	R/W*1
23:18	—	These bits are read as 0. The write value should be 0.	R/W
24	SPCD	Period Count Function Simultaneous Disable*2 0: Disable simultaneous disabling period count function 1: Disable period count function simultaneously	R/W

Bit	Symbol	Function	R/W
25	SSCD	Synchronous Set/Clear Simultaneous Disable*3 0: Disable simultaneous disabling synchronous set/clear 1: Disable synchronous set/clear simultaneously	R/W*1
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. When in complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.
Note 2. This bit is only available in GPT320 to GPT323 and GPT3210 to GPT3213.
In GPT324 to GPT329, this bit is read as 0. The write value should be 0.
Note 3. This bit is only available in GPT324 to GPT329.
In GPT320 to GPT323 and GPT3210 to GPT3213, this bit is read as 0. The write value should be 0.

The GTSECR register simultaneously updates the value for operation enable bits of a channel set by the GTSECSR register.

Writing 1 to a bit in the GTSECR register in any channel and updating it updates an operation enable bit for all channels, related to the position of the bit written with 1 by the all GTSECSR registers.

Contrary to a security attribute for write or read access, if the security and privilege attributes set on each channel has the security violation or the privilege violation, the bit of the channel number that violates security cannot be written or read, and the read value of the bit is 0.

Setting enable and disable bits for the same operation enable bit to 1 in the GTSECR is prohibited.

A bit written to 1 is automatically cleared. When the GTSECR is read, 0 is read.

Access in 8-bit or 16-bit units to the GTSECR register is prohibited, and it should be accessed in 32-bit units.

SBDCE bit (GTCCR Register Buffer Operation Simultaneous Enable)

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[0] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTCCRA, GTCCRC, and GTCCRD registers and using the GTCCRB, GTCCRE, and GTCCRF registers are enabled.

Simultaneous setting of SBDCE and SBDCD bits to 1 is prohibited.

SBDPE bit (GTPR Register Buffer Operation Simultaneous Enable)

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[1] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTPR, GTPDBR, and GTPBR registers are enabled.

Simultaneous setting of SBDPE and SBDPD bits to 1 is prohibited.

SBDAE bit (GTADTR Register Buffer Operation Simultaneous Enable)

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[2] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTADTRA, GTADTBRA, and GTADTDBRA registers and using the GTADTRB, GTADTBRB, and GTADTDBRB registers are enabled.

Simultaneous setting of SBDAE and SBDAD bits to 1 is prohibited.

SBDDE bit (GTDV Register Buffer Operation Simultaneous Enable)

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[3] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTDVU and GTDBU registers and using the GTDVD and GTDBD registers are enabled.

Simultaneous setting of SBDDE and SBDDD bits to 1 is prohibited.

SBDCD bit (GTCCR Register Buffer Operation Simultaneous Disable)

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[0] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTCCRA, GTCCRC, and GTCCRD registers and using the GTCCRB, GTCCRE, and GTCCRF registers are disabled.

Simultaneous setting of SBDCE and SBDCD bits to 1 is prohibited.

SBDPD bit (GTPR Register Buffer Operation Simultaneous Disable)

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[1] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTPR, GTPDBR, and GTPBR registers are disabled.

Simultaneous setting of SBDPE and SBDPD bits to 1 is prohibited.

SBDAD bit (GTADTR Register Buffer Operation Simultaneous Disable)

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[2] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTADTRA, GTADTBRA, and GTADTDBRA registers and using the GTADTRB, GTADTBRB, and GTADTDBRB registers are disabled.

Simultaneous setting of SBDAE and SBDAD bits to 1 is prohibited.

SBDDD bit (GTDV Register Buffer Operation Simultaneous Disable)

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[3] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTDVU and GTDBU registers and using the GTDVD and GTDBD registers are disabled.

Simultaneous setting of SBDDE and SBDDD bits to 1 is prohibited.

SPCE bit (Period Count Function Simultaneous Enable)

When 1 is written to this bit, 1 is simultaneously set to GTPC.PCEN bit in the channels set to 1 by the GTSECSR register, and period count function is enabled.

Simultaneous setting of SPCE and SPCD bits to 1 is prohibited.

SSCE bit (Synchronous Set/Clear Simultaneous Enable)

When 1 is written to this bit, 1 is simultaneously set to GTCR.SSCEN bit in the channels set to 1 by the GTSECSR register and Enable synchronous set/clear function.

Simultaneous setting of SSCE and SSCD bits to 1 is prohibited.

SPCD bit (Period Count Function Simultaneous Disable)

When 1 is written to this bit, 0 is simultaneously set to GTPC.PCEN bit in the channels set to 1 by the GTSECSR register, and period count function is disabled.

Simultaneous setting of SPCE and SPCD bits to 1 is prohibited.

SSCD bit (Synchronous Set/Clear Simultaneous Disable)

When 1 is written to this bit, 0 is simultaneously set to GTCR.SSCEN bit in the channels set to 1 by the GTSECSR register and Disable synchronous set/clear function.

Simultaneous setting of SSCE and SSCD bits to 1 is prohibited.

22.2.43 GTBER2 : General PWM Timer Buffer Enable Register 2

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0xE0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	OLTTB[1:0]		OLTTA[1:0]		CPBTD	CP3DB	—		CPTDV	CPTADB	CPTADA	CPTPR	CPTCB	CPTCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	CMTADB	CMTADA	—	CMTCB[1:0]		CMTCA[1:0]		—	—	CCTDV	CCTADB	CCTADA	CCTPR	CCTCB	CCTCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CCTCA	Counter Clear Source GTCCRA Register Buffer Transfer Disable 0: Enable GTCCRA register buffer transfer by counter clear 1: Disable GTCCRA register buffer transfer by counter clear	R/W

Bit	Symbol	Function	R/W
1	CCTCB	Counter Clear Source GTCCRB Register Buffer Transfer Disable 0: Enable GTCCRB register buffer transfer by counter clear 1: Disable GTCCRB register buffer transfer by counter clear	R/W
2	CCTPR	Counter Clear Source GTPR Register Buffer Transfer Disable 0: Enable GTPR register buffer transfer by counter clear 1: Disable GTPR register buffer transfer by counter clear	R/W
3	CCTADA	Counter Clear Source GTADTRA Register Buffer Transfer Disable 0: Enable GTADTRA register buffer transfer by counter clear 1: Disable GTADTRA register buffer transfer by counter clear	R/W
4	CCTADB	Counter Clear Source GTADTRB Register Buffer Transfer Disable 0: Enable GTADTRB register buffer transfer by counter clear 1: Disable GTADTRB register buffer transfer by counter clear	R/W
5	CCTDV	Counter Clear Source GTDVU/GTDVD Register Buffer Transfer Disable 0: Enable GTDVU/GTDVD register buffer transfer by counter clear 1: Disable GTDVU/GTDVD register buffer transfer by counter clear	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
9:8	CMTCA[1:0]	Compare Match Source GTCCRA Register Buffer Transfer Enable 0 0: Disable GTCCRA register Buffer Transfer by compare match of GTCCRA register and GTCCRB register 0 1: Enable GTCCRA register Buffer Transfer by compare match of GTCCRA register 1 0: Enable GTCCRA register Buffer Transfer by compare match of GTCCRB register 1 1: Enable GTCCRA register Buffer Transfer by compare match of GTCCRA register and GTCCRB register	R/W
11:10	CMTCB[1:0]	Compare Match Source GTCCRB Register Buffer Transfer Enable 0 0: Disable GTCCRB register Buffer Transfer by compare match of GTCCRA register and GTCCRB register 0 1: Enable GTCCRB register Buffer Transfer by compare match of GTCCRA register 1 0: Enable GTCCRB register Buffer Transfer by compare match of GTCCRB register 1 1: Enable GTCCRB register Buffer Transfer by compare match of GTCCRA register and GTCCRB register	R/W
12	—	This bit is read as 0. The write value should be 0.	R/W
13	CMTADA	Compare Match Source GTADTRA Register Buffer Transfer Enable 0: Disable GTADTRA register buffer transfer by compare match of GTADTRA register 1: Enable GTADTRA register buffer transfer by compare match of GTADTRA register	R/W
14	CMTADB	Compare Match Source GTADTRB Register Buffer Transfer Enable 0: Disable GTADTRB register buffer transfer by compare match of GTADTRB register 1: Enable GTADTRB register buffer transfer by compare match of GTADTRB register	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
16	CPTCA	Overflow/Underflow Source GTCCRA Register Buffer Transfer Disable 0: Enable GTCCRA register buffer transfer by overflow/underflow 1: Disable GTCCRA register buffer transfer by overflow/underflow	R/W
17	CPTCB	Overflow/Underflow Source GTCCRB Register Buffer Transfer Disable 0: Enable GTCCRB register buffer transfer by overflow/underflow 1: Disable GTCCRB register buffer transfer by overflow/underflow	R/W
18	CPTPR	Overflow/Underflow Source GTPR Register Buffer Transfer Disable 0: Enable GTPR register buffer transfer by overflow/underflow 1: Disable GTPR register buffer transfer by overflow/underflow	R/W
19	CPTADA	Overflow/Underflow Source GTADTRA Register Buffer Transfer Disable 0: Enable GTADTRA register buffer transfer by overflow/underflow 1: Disable GTADTRA register buffer transfer by overflow/underflow	R/W
20	CPTADB	Overflow/Underflow Source GTADTRB Register Buffer Transfer Disable 0: Enable GTADTRB register buffer transfer by overflow/underflow 1: Disable GTADTRB register buffer transfer by overflow/underflow	R/W

Bit	Symbol	Function	R/W
21	CPTDV	Overflow/Underflow Source GTDVU/GTDVD Register Buffer Transfer Disable 0: Enable GTDVU/GTDVD register buffer transfer by overflow/underflow 1: Disable GTDVU/GTDVD register buffer transfer by overflow/underflow	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
24	CP3DB	Complementary PWM mode 3,4 Double Buffer select*1 0: Disable double buffer function in complementary PWM mode 3, 4 1: Enable double buffer function in complementary PWM mode 3, 4	R/W
25	CPBTD	Complementary PWM mode Buffer Transfer Disable*1 0: Enable buffer transfer from temporary register to GTCCRC and GTPBR register 1: Disable buffer transfer from temporary register to GTCCRC and GTPBR register	R/W
27:26	OLTTA[1:0]	GTIOCnA Output Level Buffer Transfer Timing Select*1 0 0: No transfer 0 1: Triangle waves, complementary PWM mode: Transfer at crest Saw waves: Transfer at the end of period 1 0: Triangle waves, complementary PWM mode: Transfer at trough Saw waves: Transfer by compare match of GTCCRA register 1 1: Triangle waves, complementary PWM mode: Transfer at both crest and trough Saw waves: Setting prohibited	R/W
29:28	OLTTB[1:0]	GTIOCnB Output Level Buffer Transfer Timing Select*1 0 0: No transfer 0 1: Triangle waves, complementary PWM mode: Transfer at crest Saw waves: Transfer at the end of period 1 0: Triangle waves, complementary PWM mode: Transfer at trough Saw waves: Transfer by compare match of GTCCRB register 1 1: Triangle waves, complementary PWM mode: Transfer at both crest and trough Saw waves: Setting prohibited	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit is only available in GPT324 to GPT329.
In GPT320 to GPT323 and GPT3210 to GPT3213, this bit is read as 0. The write value should be 0.

The GTBER2 register makes settings for buffer operation.

Set the CP3DB bit and OLTTm[1:0] (m = A, B) bits when the GTCNT counter is stopped.

CCTCA bit (Counter Clear Source GTCCRA Register Buffer Transfer Disable)

This bit disables buffer transfer by counter clear using the GTCCRA, GTCCRC, and GTCCRD registers together.

This bit is effective when the GTBER.BD[0] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.CCRA[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode. Even if the CCTCA bit is 0, buffer transfer is not performed by counter clear.

If there is a conflict with the CMTCA bit setting, the CCTCA bit setting has priority.

The setting is invalid during the event count operation.

CCTCB bit (Counter Clear Source GTCCRB Register Buffer Transfer Disable)

This bit disables buffer transfer by counter clear using the GTCCRB, GTCCRE, and GTCCRF registers together.

This bit is effective when the GTBER.BD[0] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.CCRB[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode. Even if the CCTCB bit is 0, buffer transfer is not performed by counter clear.

If there is a conflict with the CMTCB bit setting, the CCTCB bit setting has priority.

The setting is invalid during the event count operation.

CCTPR bit (Counter Clear Source GTPR Register Buffer Transfer Disable)

This bit disables buffer transfer by counter clear using the GTPR, GTPBR, and GTPDBR registers together.

This bit is effective when the GTBER.BD[1] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.PR[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode. Even if the CCTPR bit is 0, buffer transfer is not performed by counter clear.

The setting is invalid during the event count operation.

CCTADm bit (Counter Clear Source GTADTRm Register Buffer Transfer Disable) (m = A, B)

This bit disables buffer transfer by counter clear using the GTADTRm, GTADTBRm and GTADTDBRm registers together.

This bit is effective when the GTBER.BD[2] bit is 0 (buffer operation enabled) and the buffer operation is selected by the ADTTm[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode. Even if the CCTADm bit is 0, buffer transfer is not performed by counter clear.

If there is a conflict with the CMTADm bit setting, the CCTADm bit setting has priority.

The setting is invalid during the event count operation.

CCTDV bit (Counter Clear Source GTDVU/GTDVD Register Buffer Transfer Disable)

This bit disables buffer transfer by counter clear using the GTDVU and GTDBU or GTDVD and GTDBD registers together.

This bit is effective when the GTBER.BD[3] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTDTCR.TDBUE or GTDTCR.TDBDE bit with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode. Even if the CCTDV bit is 0, buffer transfer is not performed by counter clear.

The setting is invalid during the event count operation.

CMTCA[1:0] bit (Compare Match Source GTCCRA Register Buffer Transfer Enable)

This bit enables buffer transfer by compare match of GTCCRA using the GTCCRA, GTCCRC, and GTCCRD registers together.

This bit is effective when the GTBER.BD[0] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.CCRA[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode.

If there is a conflict with the CCTCA bit setting, the CCTCA bit setting has priority.

The setting is invalid during the event count operation.

CMTCB[1:0] bit (Compare Match Source GTCCRB Register Buffer Transfer Enable)

This bit enables buffer transfer by compare match of GTCCRB using the GTCCRB, GTCCRE, and GTCCRF registers together.

This bit is effective when the GTBER.BD[0] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.CCRB[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode.

If there is a conflict with the CCTCB bit setting, the CCTCB bit setting has priority.

The setting is invalid during the event count operation.

CMTADm bit (Compare Match Source GTADTRm Register Buffer Transfer Enable) (m = A, B)

This bit enables buffer transfer by compare match of GTADTRm using the GTADTRm, GTADTBRm and GTADTDBRm registers together.

This bit is effective when the GTBER.BD[2] bit is 0 (buffer operation enabled) and the buffer operation is selected by the ADTTm[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode.

If there is a conflict with the CCTADm bit setting, the CCTADm bit setting has priority.

The setting is invalid during the event count operation.

CPTCA bit (Overflow/Underflow Source GTCCRA Register Buffer Transfer Disable)

This bit disables buffer transfer by overflow/underflow in saw-waves using the GTCCRA, GTCCRC, and GTCCRD registers together.

This bit is effective when the CCTCA bit is 0 (GTCCRA register buffer transfer by counter clear is enabled), the GTBER.BD[0] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.CCRA[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode.

The setting is invalid during the event count operation.

CPTCB bit (Overflow/Underflow Source GTCCRB Register Buffer Transfer Disable)

This bit disables buffer transfer by overflow/underflow in saw-waves using the GTCCRB, GTCCRE and GTCCRF registers together.

This bit is effective when the CCTCB bit is 0 (GTCCRB register buffer transfer by counter clear is enabled), the GTBER.BD[0] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.CCRB[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode.

The setting is invalid during the event count operation.

CPTPR bit (Overflow/Underflow Source GTPR Register Buffer Transfer Disable)

This bit disables buffer transfer by overflow/underflow in saw-waves using the GTPR, GTPBR, and GTPDBR registers together.

This bit is effective when the CCTPR bit is 0 (GTPR register buffer transfer by counter clear is enabled), the GTBER.BD[1] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.PR[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode.

The setting is invalid during the event count operation.

CPTADm bit (Overflow/Underflow Source GTADTRm Register Buffer Transfer Disable) (m = A, B)

This bit disables buffer transfer by overflow/underflow in saw-waves using the GTADTRm, GTADTBRm and GTADTDBRm registers together.

This bit is effective when the CCTADm bit is 0 (GTADTRm register buffer transfer by counter clear is enabled), the GTBER.BD[2] bit is 0 (buffer operation enabled) and the buffer operation is selected by the ADTTm[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode.

The setting is invalid during the event count operation.

CPTDV bit (Overflow/Underflow Source GTDVU/GTDVD Register Buffer Transfer Disable)

This bit disables buffer transfer by overflow/underflow in saw-waves using the GTDVU and GTDBU or GTDVD and GTDBD registers together.

This bit is effective when the CCTDV bit is 0 (GTDVU/GTDVD register buffer transfer by counter clear is enabled), the GTBER.BD[3] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTDTCR.TDBUE or GTDTCR.TDBDE bit with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode.

The setting is invalid during the event count operation.

CP3DB bit (Complementary PWM mode 3,4 Double Buffer select)

This bit enables buffer transfer using the GTCCRA, GTCCRE and GTCCRF registers together in complementary PWM mode 3,4.

CPBTD bit (Complementary PWM mode Buffer Transfer Disable)

This bit disables buffer transfer from temporary register (temporary register A and temporary register P) to the GTCCRC and GTPBR registers during timer counting in complementary PWM mode 1, 2 and 3. When CP3DB bit is 1, the buffer transfer from temporary register B to GTCCRE register is also disabled. The setting is invalid in complementary PWM mode 4.

OLTTm[1:0] bits (GTIOCnA Output Level Buffer Transfer Timing Select) (m = A, B)

This bit set the timing of buffer transfer from GTOLBR.GTIOmB[4:0] bits to GTIOR.GTIOm[4:0] bits.

22.2.44 GTOLBR : General PWM Timer Output Level Buffer Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0xE4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	GTIOBB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
4:0	GTIOAB[4:0]	GTIOA buffer bits	R/W
15:5	—	These bits are read as 0. The write value should be 0.	R/W
20:16	GTIOBB[4:0]	GTIOB buffer bits	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

The GTOLBR register is a buffer register for the GTIOR.GTIOA[4:0] bits and GTIOR.GTIOB[4:0] bits.

GTIOmB[4:0] bits (GTIOm buffer bits) (m = A, B)

These bits are buffer bits of GTIOR.GTIOm[4:0] bits.

These bits are transferred to the GTIOR.GTIOm[4:0] bits at the transfer timing selected by the GTBER2.OLTTm[1:0] (m = A, B) bits.

22.2.45 GTICCR : General PWM Timer Inter Channel Cooperation Input Capture Control Register

Base address: GPT32n = 0x4032_2000 + 0x0100 × n (n = 0 to 13)
GPT32n_NS = 0x5032_2000 + 0x0100 × n (n = 0 to 13)

Offset address: 0xEC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ICBGRP[1:0]		—	—	—	—	—	ICBCLK	ICBFPU	ICBFPO	ICBFF	ICBFE	ICBFD	ICBFC	ICBFB	ICBFA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ICAGRP[1:0]		—	—	—	—	—	ICACLK	ICAFPU	ICAFPO	ICAFF	ICAFE	ICAFD	ICAFC	ICAFB	ICAFA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ICAFA	Forwarding GTCCRA register Compare Match/Input Capture to Other Channel GTCCRA Input Capture Source Enable 0: Disable forwarding GTCCRA register compare match/input capture to GTCCRA input capture source of other channels 1: Enable forwarding GTCCRA register compare match/input capture to GTCCRA input capture source of other channels	R/W
1	ICAFB	Forwarding GTCCRB register Compare Match/Input Capture to Other Channel GTCCRA Input Capture Source Enable 0: Disable forwarding GTCCRB register compare match/input capture to GTCCRA input capture source of other channels 1: Enable forwarding GTCCRB register compare match/input capture to GTCCRA input capture source of other channels	R/W
2	ICAFC	Forwarding GTCCRC register Compare Match to Other Channel GTCCRA Input Capture Source Enable 0: Disable forwarding GTCCRC register compare match to GTCCRA input capture source of other channels 1: Enable forwarding GTCCRC register compare match to GTCCRA input capture source of other channels	R/W
3	ICAFD	Forwarding GTCCRD register Compare Match to Other Channel GTCCRA Input Capture Source Enable 0: Disable forwarding GTCCRD register compare match to GTCCRA input capture source of other channels 1: Enable forwarding GTCCRD register compare match to GTCCRA input capture source of other channels	R/W
4	ICAFE	Forwarding GTCCRE register Compare Match to Other Channel GTCCRA Input Capture Source Enable 0: Disable forwarding GTCCRE register compare match to GTCCRA input capture source of other channels 1: Enable forwarding GTCCRE register compare match to GTCCRA input capture source of other channels	R/W
5	ICAFF	Forwarding GTCCRF register Compare Match to Other Channel GTCCRA Input Capture Source Enable 0: Disable forwarding GTCCRF register compare match to GTCCRA input capture source of other channels 1: Enable forwarding GTCCRF register compare match to GTCCRA input capture source of other channels	R/W
6	ICAFPO	Forwarding Overflow to Other Channel GTCCRA Input Capture Source Enable 0: Disable forwarding overflow in saw-waves or the crest in triangle-waves or complementary PWM mode to GTCCRA input capture source of other channels 1: Enable forwarding overflow in saw-waves or the crest in triangle-waves or complementary PWM mode to GTCCRA input capture source of other channels	R/W
7	ICAFPU	Forwarding Underflow to Other Channel GTCCRA Input Capture Source Enable 0: Disable forwarding underflow in saw-waves or the trough in triangle-waves or complementary PWM mode to GTCCRA input capture source of other channels 1: Enable forwarding underflow in saw-waves or the trough in triangle-waves or complementary PWM mode to GTCCRA input capture source of other channels	R/W
8	ICACLK	Forwarding Count Clock to Other Channel GTCCRA Input Capture Source Enable 0: Disable forwarding count clock to GTCCRA input capture source of other channels 1: Enable forwarding count clock to GTCCRA input capture source of other channels	R/W
13:9	—	These bits are read as 0. The write value should be 0.	R/W
15:14	ICAGRP[1:0]	GTCCRA Input Capture Group Select 0 0: Select group A 0 1: Select group B 1 0: Select group C 1 1: Select group D	R/W

Bit	Symbol	Function	R/W
16	ICBFA	Forwarding GTCCRA register Compare Match/Input Capture to Other Channel GTCCRB Input Capture Source Enable 0: Disable forwarding GTCCRA register compare match/input capture to GTCCRB input capture source of other channels 1: Enable forwarding GTCCRA register compare match/input capture to GTCCRB input capture source of other channels	R/W
17	ICBFB	Forwarding GTCCRB register Compare Match/Input Capture to Other Channel GTCCRB Input Capture Source Enable 0: Disable forwarding GTCCRB register compare match/input capture to GTCCRB input capture source of other channels 1: Enable forwarding GTCCRB register compare match/input capture to GTCCRB input capture source of other channels	R/W
18	ICBFC	Forwarding GTCCRC register Compare Match to Other Channel GTCCRB Input Source Capture Enable 0: Disable forwarding GTCCRD register compare match to GTCCRB input capture source of other channels 1: Enable forwarding GTCCRD register compare match to GTCCRB input capture source of other channels	R/W
19	ICBFD	Forwarding GTCCRD register Compare Match to Other Channel GTCCRB Input Capture Source Enable 0: Disable forwarding GTCCRD register compare match to GTCCRB input capture source of other channels 1: Enable forwarding GTCCRD register compare match to GTCCRB input capture source of other channels	R/W
20	ICBFE	Forwarding GTCCRE register Compare Match to Other Channel GTCCRB Input Capture Source Enable 0: Disable forwarding GTCCRE register compare match to GTCCRB input capture source of other channels 1: Enable forwarding GTCCRE register compare match to GTCCRB input capture source of other channels	R/W
21	ICBFF	Forwarding GTCCRF register Compare Match to Other Channel GTCCRB Input Capture Source Enable 0: Disable forwarding GTCCRF register compare match to GTCCRB input capture source of other channels 1: Enable forwarding GTCCRF register compare match to GTCCRB input capture source of other channels	R/W
22	ICBFPO	Forwarding Overflow to Other Channel GTCCRB Input Capture Source Enable 0: Disable forwarding overflow in saw-waves or the crest in triangle-waves or complementary PWM mode to GTCCRB input capture source of other channels 1: Enable forwarding overflow in saw-waves or the crest in triangle-waves or complementary PWM mode to GTCCRB input capture source of other channels	R/W
23	ICBFPU	Forwarding Underflow to Other Channel GTCCRB Input Capture Source Enable 0: Disable forwarding underflow in saw-waves or the trough in triangle-waves or complementary PWM mode to GTCCRB input capture source of other channels 1: Enable forwarding underflow in saw-waves or the trough in triangle-waves or complementary PWM mode to GTCCRB input capture source of other channels	R/W
24	ICBCLK	Forwarding Count Clock to Other Channel GTCCRB Input Capture Source Enable 0: Disable forwarding count clock to GTCCRB input capture source of other channels 1: Enable forwarding count clock to GTCCRB input capture source of other channels	R/W
29:25	—	These bits are read as 0. The write value should be 0.	R/W
31:30	ICBGRP[1:0]	GTCCRB Input Capture Group Select 0 0: Select group A 0 1: Select group B 1 0: Select group C 1 1: Select group D	R/W

The GTICCR register is a register that controls input capture by inter channel cooperation.

For channels that perform input capture by inter channel cooperation, the forwarding enable bit of the input capture source corresponding to the GTCCRA register or GTCCRB register where the input capture occurs is invalid.

ICAFm bit (Forwarding GTCCRm register Compare Match/Input Capture to Other Channel GTCCRA Input Capture Source Enable) (m = A, B)

Enables/disables the use of compare match/input capture of GTCCRm register as the input capture source of other channel's GTCCRA register.

ICAFx bit (Forwarding GTCCRx register Compare Match to Other Channel GTCCRA Input Capture Source Enable) (x = C, D, E, F)

Enables/disables the use of compare match of GTCCRx register as the input capture source of other channel's GTCCRA register.

ICAFPO bit (Forwarding Overflow to Other Channel GTCCRA Input Capture Source Enable)

Enable/disable to use the overflow of saw-waves, the crest of triangle-waves or complementary PWM mode as the input capture source of other channel's GTCCRA register.

ICAFPU bit (Forwarding Underflow to Other Channel GTCCRA Input Capture Source Enable)

Enable/disable to use the underflow of saw-waves, the trough of triangle-waves or complementary PWM mode as the input capture source of other channel's GTCCRA register.

ICACLK bit (Forwarding Count Clock to Other Channel GTCCRA Input Capture Source Enable)

Enable/disable to use count clock as the input capture source of other channel's GTCCRA register.

ICAGRP[1:0] bit (GTCCRA Input Capture Group Select)

Select the group of input capture by inter channel cooperation for GTCCRA register.

For channels that accept input capture of the GTCCRA register due to input capture sources from other channels, set the GTICASR.ASOC bit to 1 and select the group of inter channel cooperation with the ICAGRP[1:0] bits.

ICBFm bit (Forwarding GTCCRm register Compare Match/Input Capture to Other Channel GTCCRB Input Capture Source Enable) (m = A, B)

Enables/disables the use of compare match/input capture of GTCCRm register as the input capture source of other channel's GTCCRB register.

ICBFx bit (Forwarding GTCCRx register Compare Match to Other Channel GTCCRB Input Capture Source Enable) (x = C, D, E, F)

Enables/disables the use of compare match of GTCCRx register as the input capture source of other channel's GTCCRB register.

ICBFPO bit (Forwarding Overflow to Other Channel GTCCRB Input Capture Source Enable)

Enable/disable to use the overflow of saw-waves, the crest of triangle-waves or complementary PWM mode as the input capture source of other channel's GTCCRB register.

ICBFPU bit (Forwarding Underflow to Other Channel GTCCRB Input Capture Source Enable)

Enable/disable to use the underflow of saw-waves, the trough of triangle-waves or complementary PWM mode as the input capture source of other channel's GTCCRB register.

ICBCLK bit (Forwarding Count Clock to Other Channel GTCCRB Input Capture Source Enable)

Enable/disable to use count clock as the input capture source of other channel's GTCCRB register.

ICBGRP[1:0] bit (GTCCRB Input Capture Group Select)

Select the group of input capture by inter channel cooperation for GTCCRB register.

For channels that accept input capture of the GTCCRB register due to input capture sources from other channels, set the GTICBSR.BSOC bit to 1 and select the group of inter channel cooperation with the ICBGRP[1:0] bits.

22.2.46 OPSCR : Output Phase Switching Control Register

Base address: GPT_OPS = 0x4032_3F00
GPT_OPS_NS = 0x5032_3F00

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	NFCS[1:0]		NFEN	—	—	GODF	GRP[1:0]		—	—	ALIGN	RV	INV	N	P	FB
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	EN	—	W	V	U	—	WF	VF	UF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	UF	Input Phase Soft Setting These bits set the input phase from software settings. Setting these bits is valid when OPSCR.FB = 1.	R/W
1	VF		R/W
2	WF		R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	U	Input U-Phase Monitor This bit monitors the state of the input phase. OPSCR.FB = 0 : External input that are synchronized by GTCLK OPSCR.FB = 1 : Software settings (UF)	R
5	V	Input V-Phase Monitor This bit monitors the state of the input phase. OPSCR.FB = 0 : External input that are synchronized by GTCLK OPSCR.FB = 1 : Software settings (VF)	R
6	W	Input W-Phase Monitor This bit monitors the state of the input phase. OPSCR.FB = 0 : External input that are synchronized by GTCLK OPSCR.FB = 1 : Software settings (WF)	R
7	—	This bit is read as 0. The write value should be 0.	R/W
8	EN	Output Phase Enable 0: Do not output (Hi-Z external pin) 1: Output*1	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	FB	External Feedback Signal Enable This bit selects the input phase from software settings and external input. 0: Select the external input 1: Select the soft setting (OPSCR.UF, VF, WF)	R/W
17	P	Positive-Phase Output (P) Control 0: Level signal output 1: PWM signal output	R/W
18	N	Negative-Phase Output (N) Control 0: Level signal output 1: PWM signal output	R/W
19	INV	Output Phase Invert Control 0: Positive logic (active-high) output 1: Negative logic (active-low) output	R/W
20	RV	Output Phase Rotation Direction Reversal Control 0: Positive rotation 1: Reverse rotation	R/W
21	ALIGN	Input Phase Alignment 0: Input phase aligned to GTCLK 1: Input phase aligned to the falling edge of PWM	R/W

Bit	Symbol	Function	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
25:24	GRP[1:0]	Output Disabled Source Selection 0 0: Select group A output disable source 0 1: Select group B output disable source 1 0: Select group C output disable source 1 1: Select group D output disable source	R/W
26	GODF	Group Output Disable Function 0: This bit function is ignored 1: Group disable clears the OPSCR.EN bit*1	R/W
28:27	—	These bits are read as 0. The write value should be 0.	R/W
29	NFEN	External Input Noise Filter Enable 0: Do not use a noise filter on the external input 1: Use a noise filter on the external input	R/W
31:30	NFCS[1:0]	External Input Noise Filter Clock Selection Noise filter sampling clock setting of the external input. 0 0: GTCLK/1 0 1: GTCLK/4 1 0: GTCLK/16 1 1: GTCLK/64	R/W

Note 1. When OPSCR.GODF = 1 and the signal value selected by the OPSCR.GRP[1:0] bit is high, the OPSCR.EN bit is set to 0.

The OPSCR register sets the output of the signal waveform required for brushless DC motor control.

UF , VF , WF bits (Input Phase Soft Setting)

The UF , VF , WF bits set the input phase from the software settings. When OPSCR.FB bit is 1, these bits are valid. The set value of the UF /VF /WF takes the place of the U/V/W external input.

U, V, W bits (Input Phase Monitor)

When the OPSCR.FB bit is 0, external inputs that are synchronized by GTCLK are monitored by these bits. When the OPSCR.FB bit is 1, the OPSCR.U, OPSCR.V, and OPSCR.W bits can read the OPSCR.UF , OPSCR.VF , and OPSCR.WF bits.

EN bit (Output Phase Enable)

The EN bit controls the output enable signal of output phase (positive phase/negative phase).

When the OPSCR.EN bit is 1, the signal waveform is output.

When the OPSCR.EN bit is 0, first set OPSCR.FB, OPSCR.UF /VF /WF (software setting is selected), OPSCR.P/N, OPSCR.INV, OPSCR.RV, OPSCR.ALIGN, OPSCR.GRP[1:0], OPSCR.GODF, OPSCR.NFEN, OPSCR.NFCS. Then, set the EN bit to 1. The EN bit should be set when output disable request does not occur from POEG. Also when OPSCR.GODF is 1 and the signal value selected in the OPSCR.GRP[1:0] bit is high, the OPSCR.EN bit is set to 0. Even if 1 is written by software, the EN bit remains at 0.

For the return, after clearing the Output Disable Request by software, set the EN bit to 1.

Priority order of the EN bit is as follows (when the conflict occurs).

When writing 1 by software and clearing to 0 by the Output Disable Request conflict for the EN bit, clearing to 0 by the Output Disable Request is enabled.

FB bit (External Feedback Signal Enable)

The FB bit selects the input phase from the software settings (OPSCR.UF, VF, WF) and external input such as a Hall element.

P bit (Positive-Phase Output (P) Control)

The P bit selects one of the level signal output or PWM signal output for the positive-phase output (GTOUUP, GTOVUP and GTOWUP pins).

N bit (Negative-Phase Output (N) Control)

The N bit selects one of the level signal output or PWM signal output for the negative-phase output (GTOULO, GTOVLO and GTOWLO pins).

INV bit (Output Phase Invert Control)

The INV bit selects one of the positive logic (active-high) output or negative logic (active-low) output for the output phase.

RV bit (Output Phase Rotation Direction Reversal Control)

The RV bit reverses the direction of rotation of the motor by inverting the input phase.

ALIGN bit (Input Phase Alignment)

The ALIGN bit selects the GTCLK or PWM for the sampling of the input phase (input phase is specified in the OPSCR.FB bit).

When OPSCR.ALIGN bit is 0, input phase is aligned to GTCLK.

Note: When the chopping is performed, there are cases where the PWM width of output is shorter than the width of the PWM used to chop just before or after switching of output phase, depending on the phase difference between the phase output switch timing and the phase of PWM.

When OPSCR.ALIGN bit is 1, input phase is aligned with the falling edge of PWM.

GRP[1:0] bit (Output Disabled Source Selection)

The GRP[1:0] bit selects the output disable source.

The GRP bits should be set when GODF bit is 0. If GRP bits select a POEG except for the connected groups, the status of output pin never change to disable.

GODF bit (Group Output Disable Function)

When OPSCR.GODF is 1 and the signal value selected by the OPSCR.GRP[1:0] bit is high, the OPSCR.EN bit is set to 0.

When OPSCR.GODF bit is 0, this bit is ignored.

The GODF bit should be set when output disable request does not occur from POEG.

NFEN bit (External Input Noise Filter Enable)

The NFEN bit selects the noise filter for external input. When OPSCR.NFEN bit is 0, a noise filter for the external input is not used.

Note: Set this bit during the EN bit is 0 to avoid generation of unintentional internal edge caused by switching this bit.

NFCS[1:0] bits (External Input Noise Filter Clock Selection)

The NFCS[1:0] bits select the clock for the external input noise filter. When the OPSCR.NFEN bit is 1, noise filter sampling clock setting of the external input is enabled.

1. Set the NFCS[1:0].
2. Wait for 2 cycles.
3. Set the OPSCR.EN bit to 1.

22.2.47 GTCLKCR : General PWM Timer Clock Control Register

Base address: GPT_GTCLK = 0x4032_3F10
GPT_GTCLK_NS = 0x5032_3F10

Offset address: 0x00

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

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

Bit	Symbol	Function	R/W
0	BPEN	Synchronization Circuit Bypass Enable 0: In case of using Bus Clock and GPT Core Clock asynchronously 1: In case of using Bus Clock and GPT Core Clock synchronously	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The GTCLKCR register is a register that controls the clock.

Set first of initial setting after resetting.

If MSTPCRE.MSTPE31 bit is 0, changing this register is prohibited.

BPEN bit (Synchronization Circuit Bypass Enable)

Enable or disable the synchronization bypass function between the bus clock (PCLKA) and GPT core clock (GTCLK).

Set 0 when using an asynchronous clock (GPTCLK) or 1 when using a synchronous clock (PCLKD) as the core clock.

22.3 Operation

22.3.1 Basic Operation

Each channel has a 32-bit timer that performs a periodic count operation using the count clock and hardware sources. The count function provides both up-counting and down-counting. The GTPR or GTCCRm (m = A to F) controls the count cycle.

When the GTCNT counter value matches the value in GTCCRA or GTCCRB, the output from the associated GTIOCnA or GTIOCnB can be changed (n = 0 to 13). GTCCRA or GTCCRB can be used as an input capture register with hardware resources.

GTCCRC and GTCCRD can function as buffer registers for GTCCRA. GTCCRE and GTCCRF can function as buffer registers for GTCCRB.

22.3.1.1 Counter Operation

(1) Counter start and stop

The counter of each channel starts the count operation when GTCR.CST is set to 1, and stops counting when the bit is set to 0. The GTCR.CST bit value is changed by the following sources:

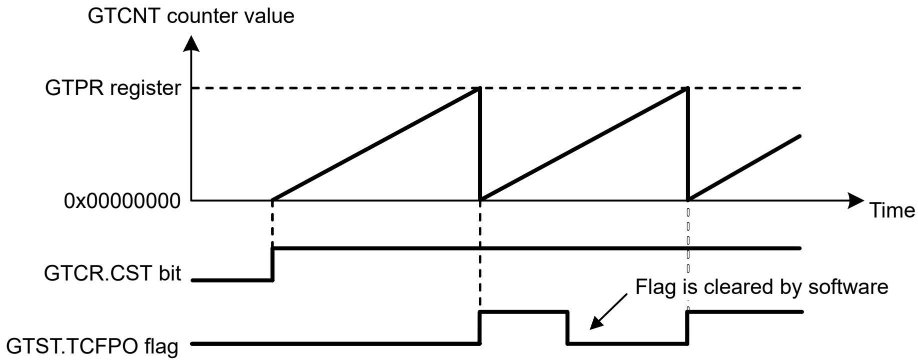
- Writing to GTCR register
- Writing 1 to the bit in GTSTR associated with the GPT channel number when the GTSSR.CSTRT bit set to 1
- Writing 1 to the bit in GTSTP associated with the GPT channel number when the GTPSR.CSTOP bit set to 1
- The hardware source selected in the GTSSR register
- The hardware source selected in the GTPSR register

- Completion of the period count function while the GTPC.ASTP bit is 1

(2) Periodic count operation in up-counting by count clock

The GTCNT counter in each channel starts up-counting when the associated GTCR.CST bit is set to 1 with GTUPSR and GTDNSR registers set to 0x00000000. When the GTCNT value changes from the GTPR value to 0 (overflow) or the GTCCRm (m = A to F) value selected by the GTCSR.CSCMSC[2:0] bits to 0 in Saw-wave PWM mode 2, the overflow interrupt(GPTn_OVF) is also generated. After GTCNT overflows, up-counting resumes from 0x00000000.

Figure 22.5 shows an example of a periodic count operation in up-counting by the count clock.



Note: The base address of registers are the same.

Figure 22.5 Example of periodic count operation in up-counting by the count clock

Table 22.11 shows an example for setting periodic count operation in up-counting by the count clock.

Table 22.11 Example for setting a periodic count operation in up-counting by the count clock

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 22.5, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 22.5, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Other than the saw-wave PWM mode2, set the cycle in the GTPR register. In the saw-wave PWM mode2, select the counter clear source compare match register GTCCRx (x = A to F) by GTCSR.CSCMSC[2:0] bits and set the cycle in that register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 22.5, 0x00000000 is set.
6	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

(3) Periodic count operation in down-counting by count clock

The GTCNT counter in each channel can perform down-counting by setting GTUDDTYC.UD with GTUPSR and GTDNSR registers set to 0x00000000. When GTCNT changes from 0 to the GTPR value (underflow), GTST.TCFPU is set to 1, and the underflow interrupt(GPTn_UDF) is also generated. After the GTCNT counter underflows, down-counting resumes from the GTPR value.

Figure 22.6 shows an example of periodic count operation in down-counting by the count clock.

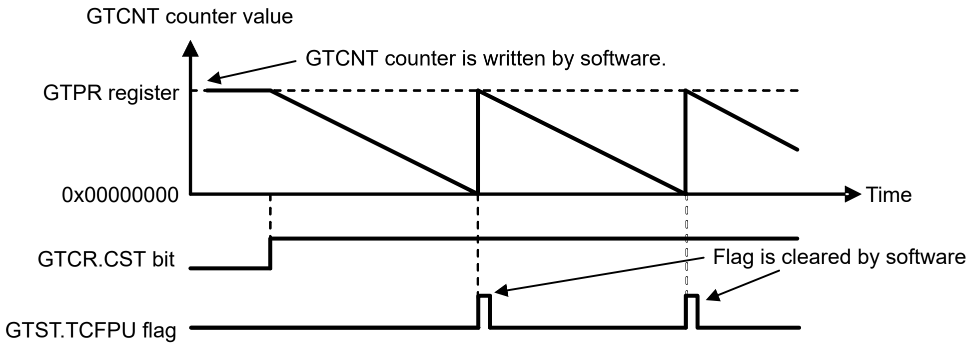


Figure 22.6 Example of periodic count operation in down-counting by the count clock

Table 22.12 shows an example for setting periodic count operation in down-counting by the count clock.

Table 22.12 Example for setting periodic count operation in down-counting by count clock

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 22.6, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction with the GTUDDTYC register. In Figure 22.6, after 10b is set in the GTUDDTYC[1:0] bits, 00b is set in the GTUDDTYC[1:0] bits (down-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 22.6, the GTPR register value is set.
6	Start count operation	Set the GTCR.CST bit to 1 to start count operation. In Figure 22.6, 1 is set in the CST bit.

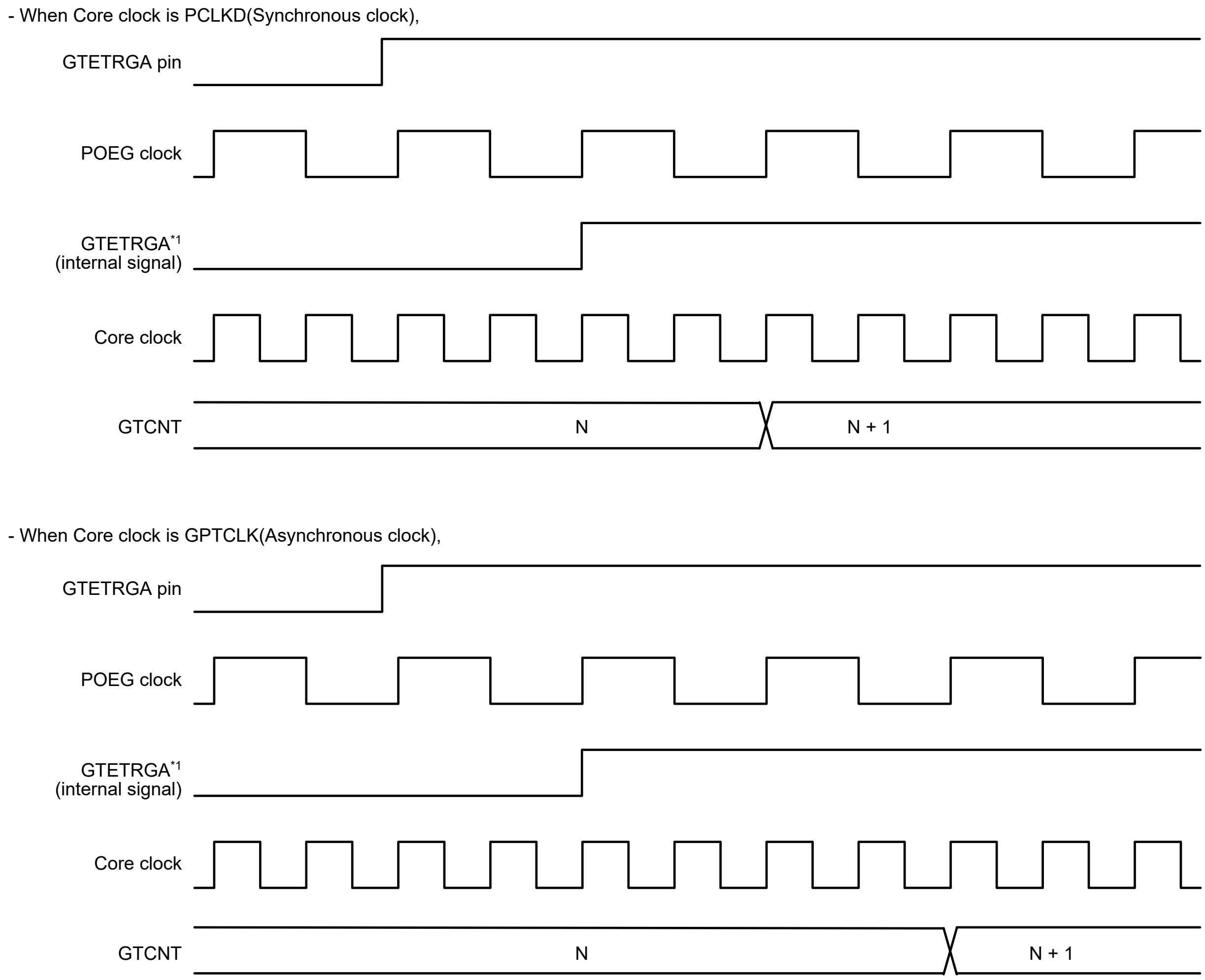
(4) Event count operation in up-counting using hardware sources

The GTCNT counter in each channel can perform up-counting using hardware sources as set in GTUPSR.

When GTUPSR is set to enable, the count clock selected in GTCR.TPCS[3:0] and the count direction selected in GTUDDTYC.UD are ignored. If up-counting and down-counting using hardware sources occur at the same time, the GTCNT counter value does not change. The overflow behavior when up-counting using hardware sources is the same as when up-counting by the count clock.

If you are using a hardware source to count up, set the GTCR.CST bit to 1 to enable the counting operation. After GTCR.CST is set to 1, the counter cannot count up for 1 clock cycle as specified in GTCR.TPCS[3:0] because the count operation is synchronized by the count clock selected in GTCR.TPCS[3:0]. Set GTCR.TPCS[3:0] to 000b to count up with a 1 GTCLK delay after GTCR.CST is set to 1.

Figure 22.7 shows an example of an event count operation in up-counting by a hardware resource (the rising edge of GTETRGA pin input).



Note: POEG clock: PCLKB

Note 1. The GTETRGA internal signal reflects the signal being input to the GPT when the digital noise filter for the GTETRGA pin in the POEG is not in use.

Figure 22.7 Example of event count operation in up-counting using hardware sources

Figure 22.8 shows an example of event count operation by ELC_GPTA event input.

This is an example of event count operation of GPT321.GTCNT counter. An event signal is output to the ELC after matches in comparison with the GPT320.GTCCRA register. This is selected as a trigger for output to the GPT321 by the ELC as ELC_GPTA.

ELC passes the event signal output from GPT320 without delay to GPT321.

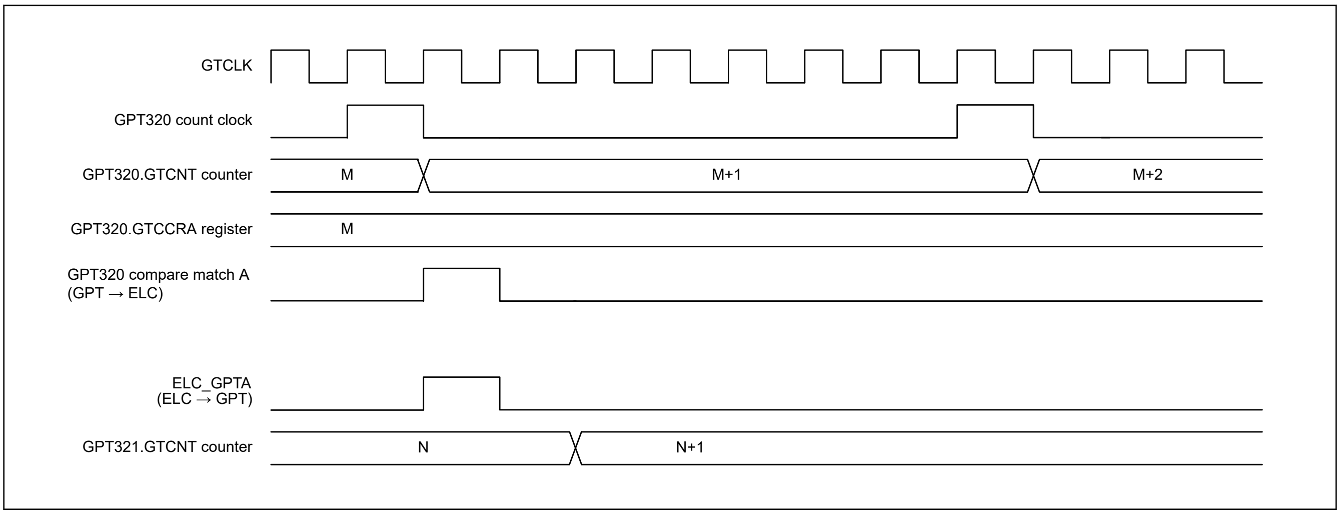


Figure 22.8 Example of Event Count Operation (Up-Counting the Number of Event Signals Input from ELC_GPTA)

Table 22.13 shows an example for setting event count operation in up-counting by a hardware source.

Table 22.13 Example for setting an event count operation in up-counting using hardware sources

No.	Step Name	Description
1	Set count source	Select the counting-up hardware source with the GTUPSR register.
2	Set cycle	Set the cycle in the GTPR register.
3	Set initial value for counter	Set the initial value in the GTCNT counter.
4	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

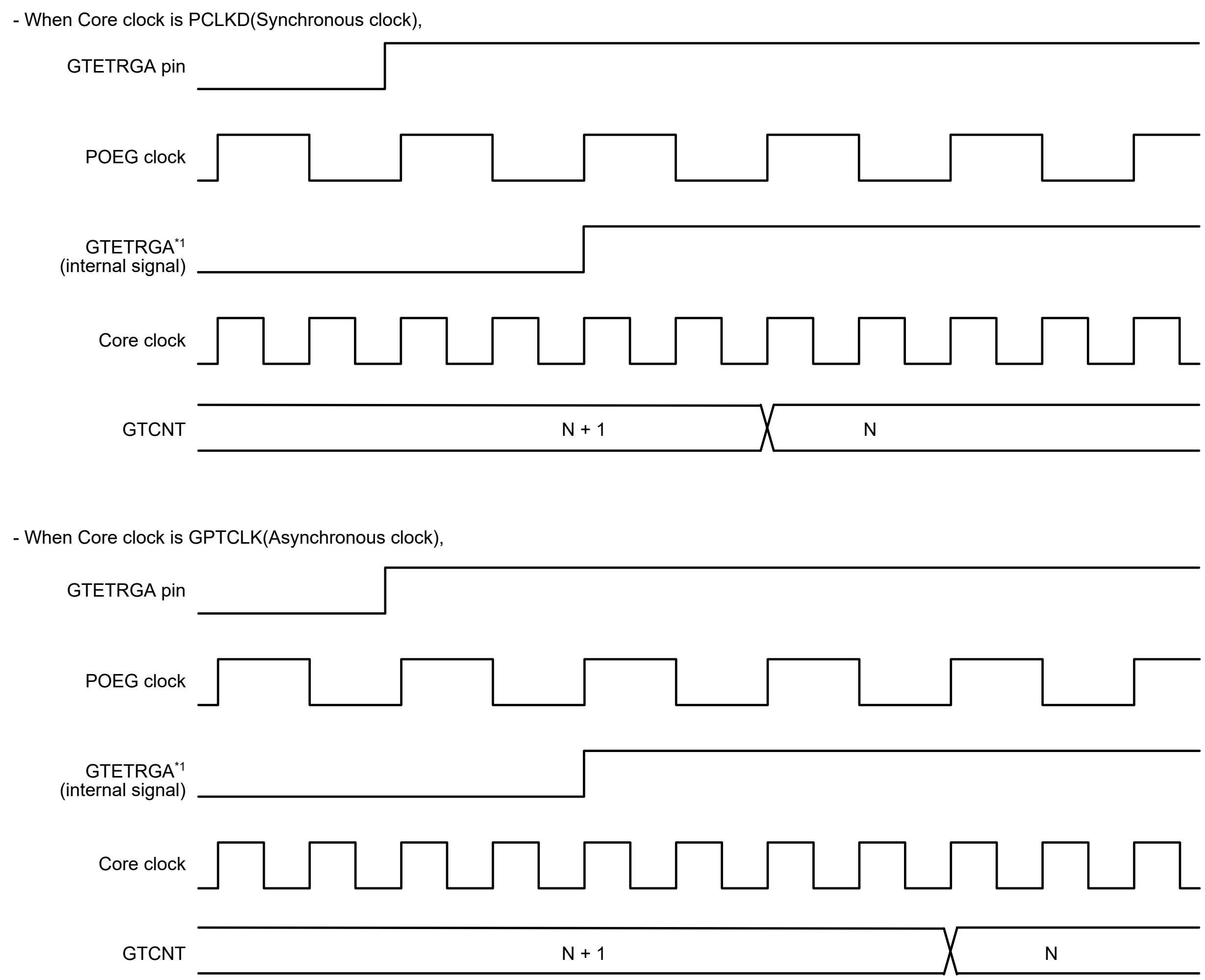
(5) Event count operation in down-counting using hardware sources

The GTCNT counter in each channel can perform down-counting using hardware sources set in the GTDNSR.

When GTDNSR is set to enable, the count clock selected in GTCR.TPCS[3:0] and the count direction selected in GTUDDTYC.UD are ignored. If up-counting and down-counting using hardware sources occur at the same time, the GTCNT counter value does not change. The underflow behavior when down-counting using hardware sources is the same as when down-counting by the count clock.

When GTCR.CST bit is set to 1 to count down using hardware sources, the count operation is enabled. After GTCR.CST is set to 1, the counter cannot count down for 1 clock cycle as specified in GTCR.TPCS[3:0] because the count operation is synchronized with the count clock selected in GTCR.TPCS[3:0]. Set GTCR.TPCS[3:0] to 000b to count down with a 1 GTCLK delay after GTCR.CST is set to 1.

Figure 22.9 shows an example of a event count operation in down-counting by a hardware resource (rising edge of GTETRGA pin).



Note: POEG clock: PCLKB

Note 1. The GTETRGA internal signal reflects the signal being input to the GPT when the digital noise filter for the GTETRGA pin in the POEG is not in use.

Figure 22.9 Example of event count operation in down-counting using hardware sources

Table 22.14 shows an example for setting a periodic count operation in down-counting using a hardware resource.

Table 22.14 Example for setting an event count operation in down-counting using hardware sources

No.	Step Name	Description
1	Set count source	Select the counting-down hardware source with the GTDNSR register.
2	Set cycle	Set the cycle in the GTPR register.
3	Set initial value for counter	Set the initial value in the GTCNT counter.
4	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

(6) Counter clear operation

The counter of each channel is cleared by following sources:

- Writing 0 to GTCNT register
- Writing 1 to the bit in GTCLR associated with the GPT channel number when the GTCSR.CCLR bit set to 1
- The hardware source selected in GTCSR register.

Write access during counting (when CST = 1) is disabled. The GTCNT counter can be cleared both by writing 1 to the GTCLR and by the clear request of hardware sources, whether GTCNT is counting (GTCR.CST is 1) or not (GTCR.CST is 0).

When the count direction flag is set as decrement (GTST.TCUF flag = 0) in saw-wave mode (Except saw-wave PWM mode 2) selected with GTCR.MD[2:0] bits or GTCR.MD[3:0] bits, the GTCNT register is set to the value of the GTPR register when writing 1 to the GTCLR register and when clearing by hardware sources are performed.

When not in saw-waves mode (except saw-wave PWM mode 2) and down-counting, the GTCNT register is set to 0 when writing 1 to the GTCLR register and when clearing by hardware sources are performed.

In event count operation when at least 1 bit in the GTUPSR or GTDNSR is set to 1, after clear sources occur, both writing to GTCLR register and clearing by hardware sources are performed immediately to synchronize with GTCLK. If other settings are used, clear is synchronized with the counter clock selected in GTCR.TPCS[3:0].

22.3.1.2 Waveform Output by Compare Match

Compare match means that the GTCNT counter value matches the value of GTCCRA or GTCCRB. At the same time, the GPT can output low, high, or toggled output from the associated GTIOCnA or GTIOCnB output pin (n = 0 to 13). In addition, the GTIOCnA or GTIOCnB pin output can be low, high, or toggled at the cycle end which is determined by GTPR or GTCCRm selected as a counter clear source by the GTCSR.CSCMSC[2:0] bits in saw-wave PWM mode 2 (n = 0 to 13, m = A to F).

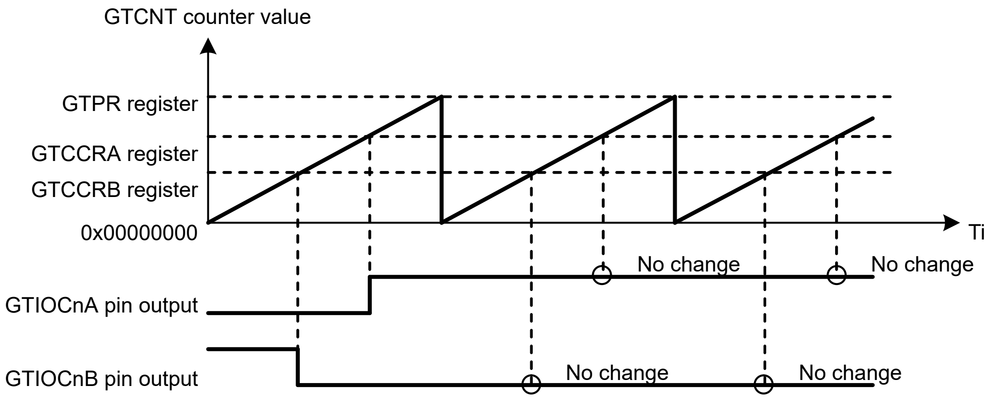
The cycle end is:

- For saw waves (except saw-wave PWM mode 2) in up-counting – when GTCNT changes from the GTPR value to 0 (overflow)
- For saw waves (except saw-wave PWM mode 2) in down-counting – when GTCNT changes from 0 to GTPR value (underflow)
- For saw-wave PWM mode 2 and GTCCRm register (m = A to F) selected as a counter clear source by the GTCSR.CSCMSC[2:0] bits – When the GTCNT counter value changes from the GTCCRm register value to 0x0000 0000
- For saw waves – when the GTCNT counter is cleared
- For triangle waves or complementary PWM mode – when the GTCNT changes from 0 to 1 (trough).

(1) Low output and high output

Figure 22.10 shows an example of low output and high output operation by a compare match of GTCCRA and GTCCRB.

In this example, the GTCNT counter performs up-counting, and settings are made so that high is output from the GTIOCnA pin by a GTCCRA compare match, and low is output from the GTIOCnB pin by a GTCCRB compare match. The pin level does not change when the specified level and pin level match.



[Setting examples]
GTIOR.GTIOA[4:0] bits: Initial output is low, high output at compare match, output retained at cycle end
GTIOR.GTIOB[4:0] bits: Initial output is high, low output at compare match, output retained at cycle end

Note: The base address of registers are the same.

Figure 22.10 Example of low output and high output operation

Table 22.15 shows an example for setting low output and high output operation.

Table 22.15 Example for setting low output and high output operation

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 22.10, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 22.10, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Other than the saw-wave PWM mode2, set the cycle in the GTPR register. In the saw-wave PWM mode2, select the counter clear source compare match register GTCCRx (x = A to F) by GTCSR.CSCMSC[2:0] bits and set the cycle in that register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 22.10, GTIOA[4:0] = 00010b, GTIOB[4:0] = 10001b.
7	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set compare match value*1	Set compare match values in the GTCCRA and GTCCRB registers.
9	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

Note: n: 0 to 13
m: A, B

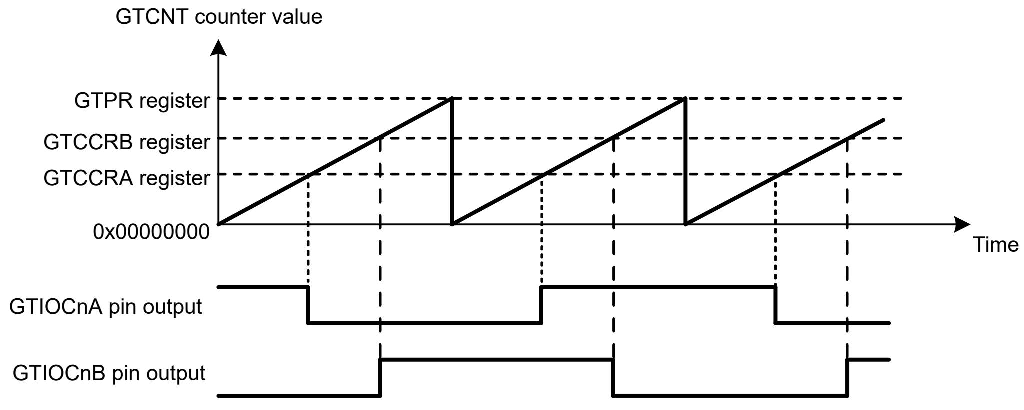
Note 1. When using PWM Delay Generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

(2) Toggled output

Figure 22.11 and Figure 22.12 show examples of toggled output operation by compare matches of GTCCRA and GTCCRB.

In Figure 22.11, the GTCNT counter performs up-counting, and settings are made so that the GTIOCnA pin output by a GTCCRA compare match and GTIOCnB pin output by a GTCCRB compare match are toggled.

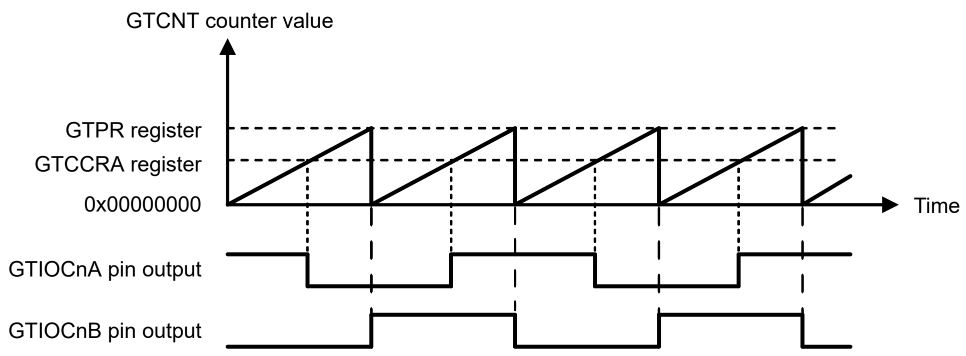
In Figure 22.12, the GTCNT counter performs up-counting, and settings are made so that a GTCCRA compare match toggles the GTIOCnA pin output level and a cycle end toggles the GTIOCnB pin output level.



[Setting examples]
GTIOR.GTIOA[4:0] bits: Initial output is high, output toggled at compare match, output retained at cycle end
GTIOR.GTIOB[4:0] bits: Initial output is low, output toggled at compare match, output retained at cycle end

Note: The base address of registers are the same.

Figure 22.11 Example of toggled output operation (1)



[Setting examples]
GTIOR.GTIOA[4:0] bits: Initial output is high, output toggled at compare match, output retained at cycle end
GTIOR.GTIOB[4:0] bits: Initial output is low, output retained at compare match, output toggled at cycle end

Note: The base address of registers are the same.

Figure 22.12 Example of toggled output operation (2)

Table 22.16 shows an example for setting toggled output operation.

Table 22.16 Example for setting toggled output operation (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 22.11 and Figure 22.12, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 22.11 and Figure 22.12, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Other than the saw-wave PWM mode2, set the cycle in the GTPR register. In the saw-wave PWM mode2, select the counter clear source compare match register GTCCRx (x = A to F) by GTCSR.CSCMSC[2:0] bits and set the cycle in that register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.

Table 22.16 Example for setting toggled output operation (2 of 2)

No.	Step Name	Description
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 22.11, GTIOA[4:0] = 10011b, GTIOB[4:0] = 00011b, and in Figure 22.12, GTIOA[4:0] = 10011b, GTIOB[4:0] = 01100b.
7	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set compare match value*1	Set compare match values in the GTCCRA and GTCCRB registers.
9	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

Note: n: 0 to 13
m: A, B

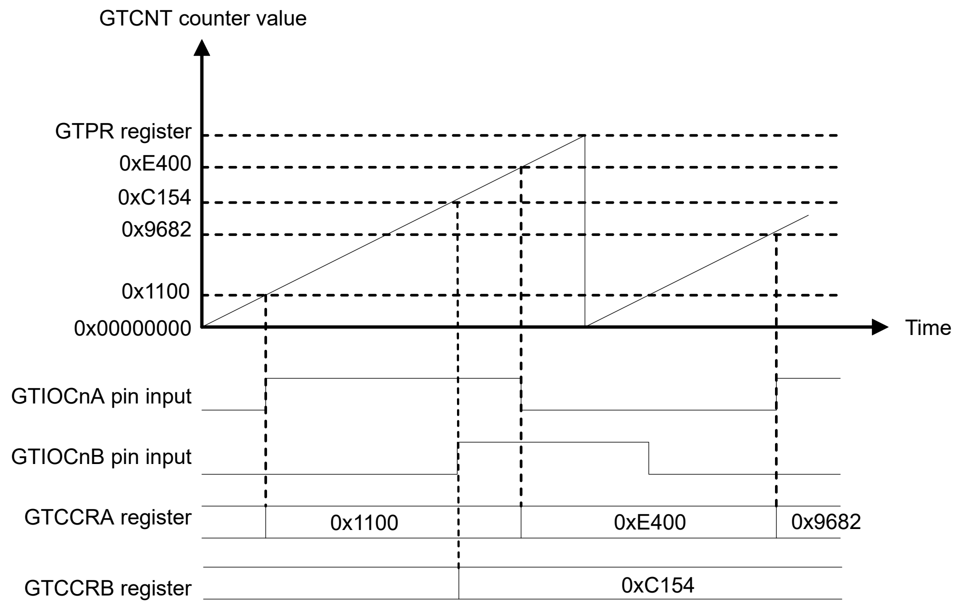
Note 1. When using PWM Delay Generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

22.3.1.3 Input Capture Function

The GTCNT counter value can be transferred to either GTCCRA or GTCCRB on detection of the hardware source that is set in GTICASR and GTICBSR. In complementary PWM mode, GTCCRA and GTCCRB registers do not function as input capture registers.

Figure 22.13 shows an example of the input capture function.

In this example, the GTCNT counter performs up-counting by the count clock, and settings are made so that an input capture is performed to GTCCRA at both edges of the GTIOCnA input pin and to GTCCRB on the rising edge of the GTIOCnB input pin.



Note: The base address of registers are the same.
Note: [Setting examples]
GTICASR setting: input capture at both edges
GTICBSR setting: input capture at the rising edge

Figure 22.13 Example of input capture operation

Table 22.17 and Table 22.21 show the example for setting an input capture operation with count operation by the count clock.

Table 22.17 Example for setting input capture operation

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 22.13, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 22.13, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Other than the saw-wave PWM mode2, set the cycle in the GTPR register. In the saw-wave PWM mode2, select the counter clear source compare match register GTCCRx (x = A to F) by GTCSR.CSCMSC[2:0] bits and set the cycle in that register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Select input capture source	Select the input capture source in the GTICASR and GTICBSR registers. In Figure 22.13, GTICASR = 0x00000F00, GTICBSR = 0x00003000. In GPT324 to GPT329 when input capture by other channel sources is used, set the GTICmSR.mSOC bit (m = A or B) to 1 and allow input capture by other channel sources, and select the group that cooperates with other channel sources by the GTICCR register.
7	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

Figure 22.14 shows an example of the timing of input capture operation in response to ELC_GPTA event input.

This is an example of capture of the counter value by the GPT321.GTCCRA register in response to an input signal. An event signal is output to the ELC after matches in comparison with the GPT320.GTCCRA register. This is selected as a trigger for output to the GPT321 by the ELC as ELC_GPTA.

ELC passes the event signal output from GPT320 without delay to GPT321.

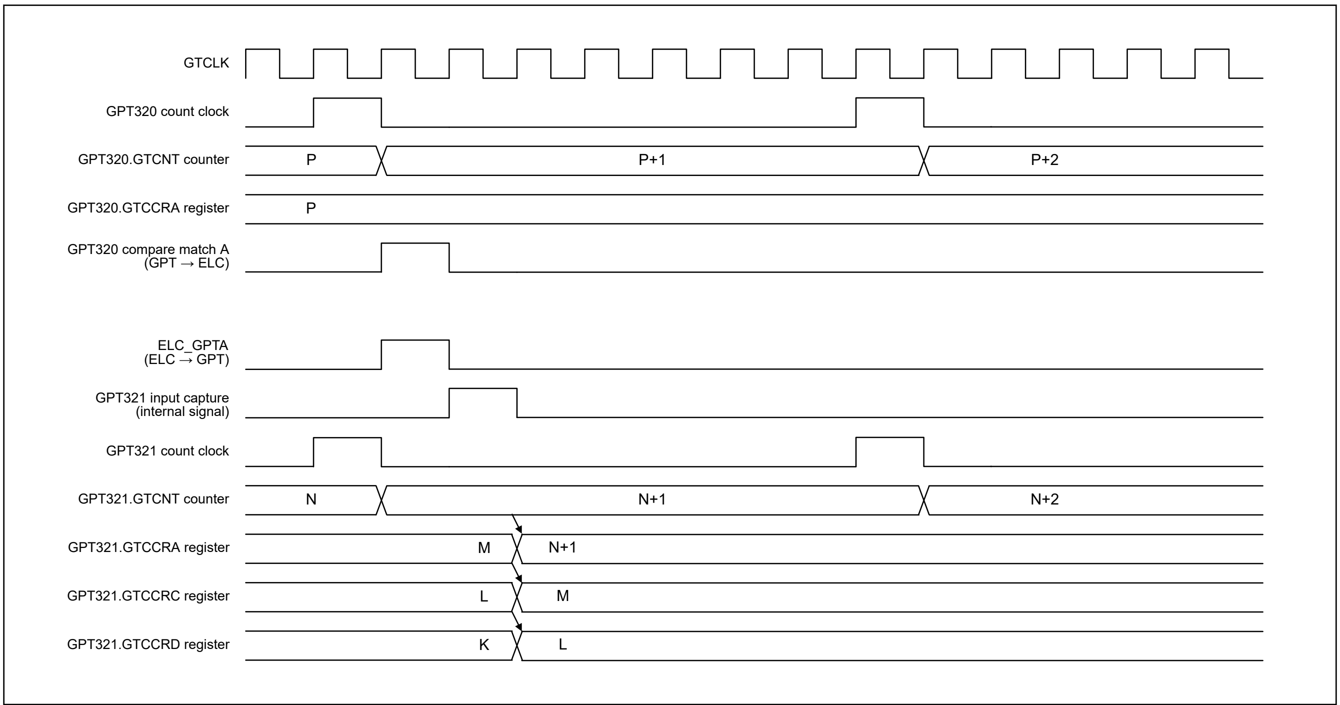


Figure 22.14 Example of the Timing of Input Capture Operation in Response to ELC_GPTA Event Input

Figure 22.15 shows an example of the timing of input capture operation in response to count clock from other channel.

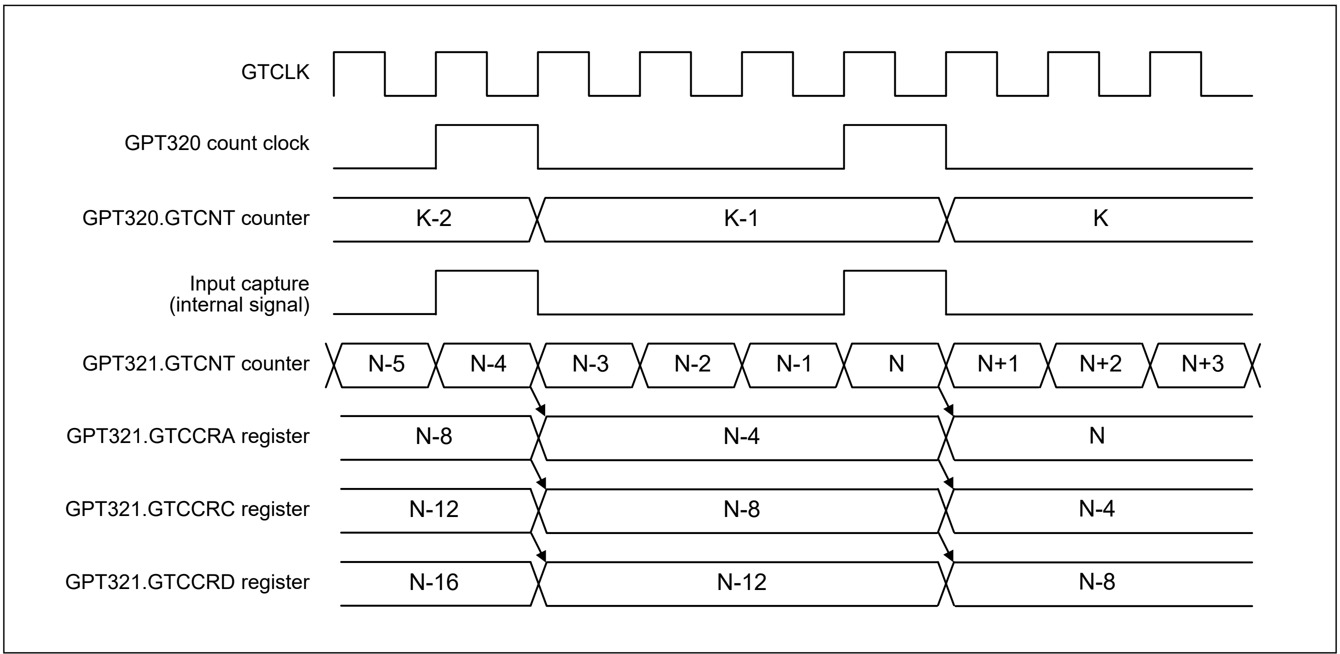


Figure 22.15 Example of the Timing of Input Capture Operation in Response to Count Clock from Other Channel

22.3.2 Buffer Operation

The following buffer operations can be set with GTBER:

- GTPR, GTPBR, and GTPDBR
- GTCCRA, GTCCRC, and GTCCRD
- GTCCRB, GTCCRE, and GTCCRF
- GTADTRA, GTADTBRA, and GTADTDBRA
- GTADTRB, GTADTBRB, and GTADTDBRB

The following buffer operation is enabled by setting GTDTCR:

- GTDVU and GTDBU
- GTDVD and GTDBD

The following buffer operation is enabled by setting GTBER2:

- GTCCRA, GTCCRE, and GTCCRF (in complementary PWM mode 3, 4)
- GTOLBR.GTIOAB[4:0] bits and GTIOR.GTIOA[4:0] bits
- GTOLBR.GTIOBB[4:0] bits and GTIOR.GTIOB[4:0] bits

22.3.2.1 GTPR Register Buffer Operation

GTPBR can function as a buffer register for GTPR.

GTPDBR register can function as a buffer register for the GTPBR register (double-buffer register for the GTPR register).

In complementary PWM mode, the buffer transfer from the GTPDBR register to temporary register P is performed only in the master channel (GPT32n). The temporary register P is transferred to each GTPBR register of master channel, slave channel 1 (GPT32n+1), and slave channel 2 (GPT32n+2). Transfer from the GTPBR register to the GTPR register is made concurrently in three channels. Therefore, the same value is stored in the same register of the three channels. The GTPR register of the master channel represents the GTCNT counter's (GTCNTn) period of the master channel. In the slave channels, periods are controlled using the GTPR register value and the GTDVU register value.

The setting is invalid in the Saw-wave PWM mode 2.

The buffer transfer is performed at an overflow (during up-counting) or an underflow (during down-counting) in saw-wave mode or in event count, and at a trough in triangle-wave mode.

In saw-wave mode or in event count, the buffer transfer is performed when the following counter clear operations occur during counting:

- Clear by hardware sources (the clear source is selected in GTCSR register)
- Clear by software (when GTCSR.CCLR bit is 1 and GTCLR.CCLRn bit is set to 1, n = 0 to 13).

In the case of saw waves, the buffer transfer by clearing counter can be prohibited by GTBER2.CCTPR bit.

Table 22.18 shows the buffer transfer timing in the complementary PWM mode.

Table 22.18 GTPR Buffer Transfer Timing in Complementary PWM Mode

	Complementary PWM mode 1	Complementary PWM mode 2	Complementary PWM mode 3, 4
GTPDBR ↓ Temporary register P	After one GTCLK cycle from GTCCRD register write of slave channel 2 (GPT32n+2)	After one GTCLK cycle from GTCCRD register write of slave channel 2 (GPT32n+2)	After one GTCLK cycle from GTCCRD register write of slave channel 2 (GPT32n+2)
Temporary register P ↓ GTPBR	(1) When data is transferred to temporary register P during up-counting middle section: After one GTCLK cycle from data transfer to temporary register P (2) When data is transferred to temporary register P during a section other than up-counting middle section: At the end of trough section	(1) When data is transferred to temporary register P during down-counting middle section: After one GTCLK cycle from data transfer to temporary register P (2) When data is transferred to temporary register P during a section other than down-counting middle section: At the end of crest section	(1) When data is transferred to temporary register P during middle section: After one GTCLK cycle from data transfer to temporary register P (2) When data is transferred to temporary register P during a section other than middle section: At the end of crest/ trough sections
GTPBR ↓ GTPR	At the end of crest section At the end of crest section Synchronous Counter clear in up counting middle section and crest section (Including counter clear by setting GTCSR.CP1CCE)	At the end of trough section Counter clear in down counting middle section and trough section	At the end of crest section At the end of trough section Counter clear

To set GTPR to function as a buffer, set the GTBER.PR bit to 1. To set GTPR not to function as a buffer, set the GTBER.PR bit to 0.

To set the GTPR register to function as double buffer, set the GTBER.PR[1:0] bits to 10b or 11b. For single buffer operation, set 01b. Not to function as buffer, set 00b.

In complementary PWM mode, complementary PWM mode-specific buffer operation is performed regardless of the GTBER.PR[1:0] bit setting.

Figure 22.16 to Figure 22.21 show examples of GTPR buffer operation and Table 22.19 shows an example for setting GTPR buffer operation. For details of operation settings in complementary PWM mode, see section 22.3.3.7. Complementary PWM mode 1, 2, 3.

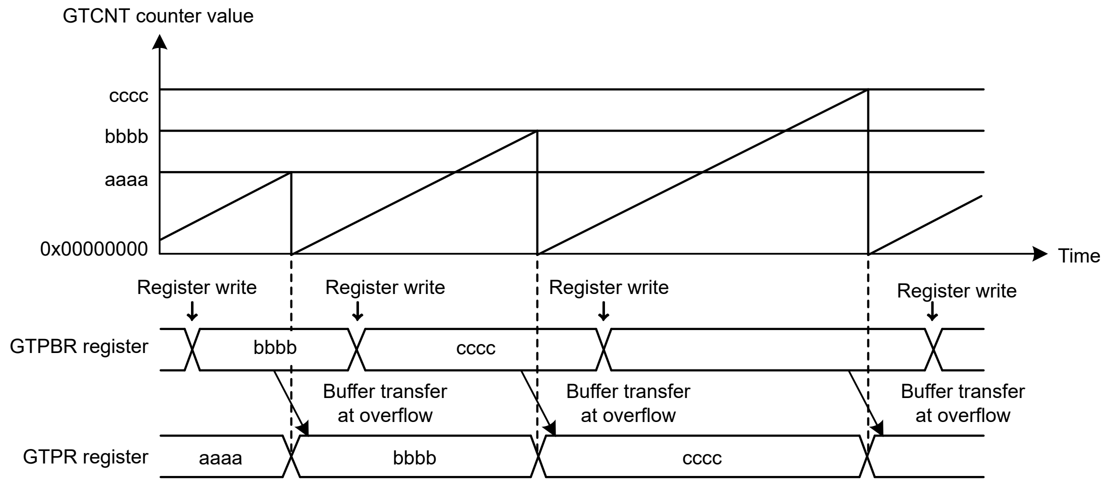


Figure 22.16 Example of GTPR buffer operation with saw waves in up-counting

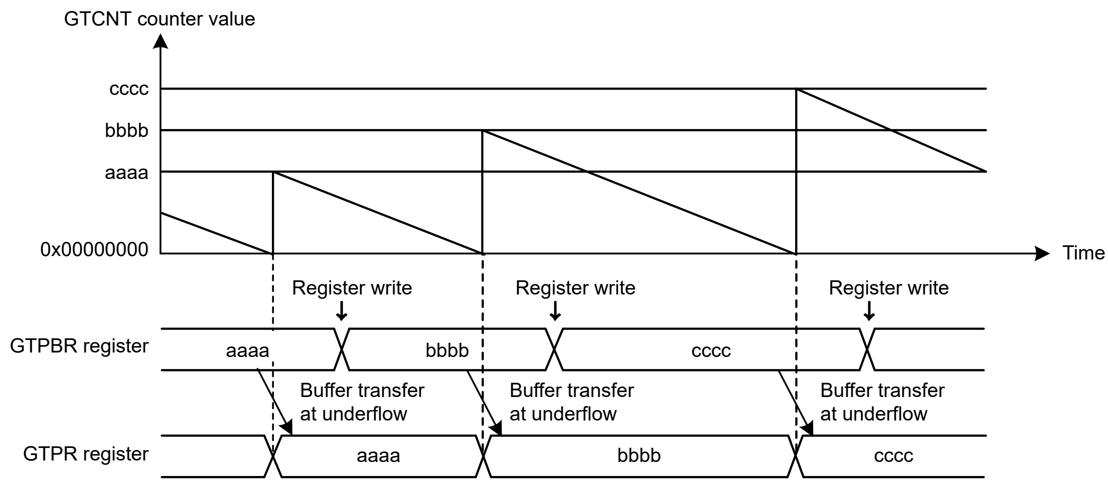


Figure 22.17 Example of GTPR buffer operation with saw waves in down-counting

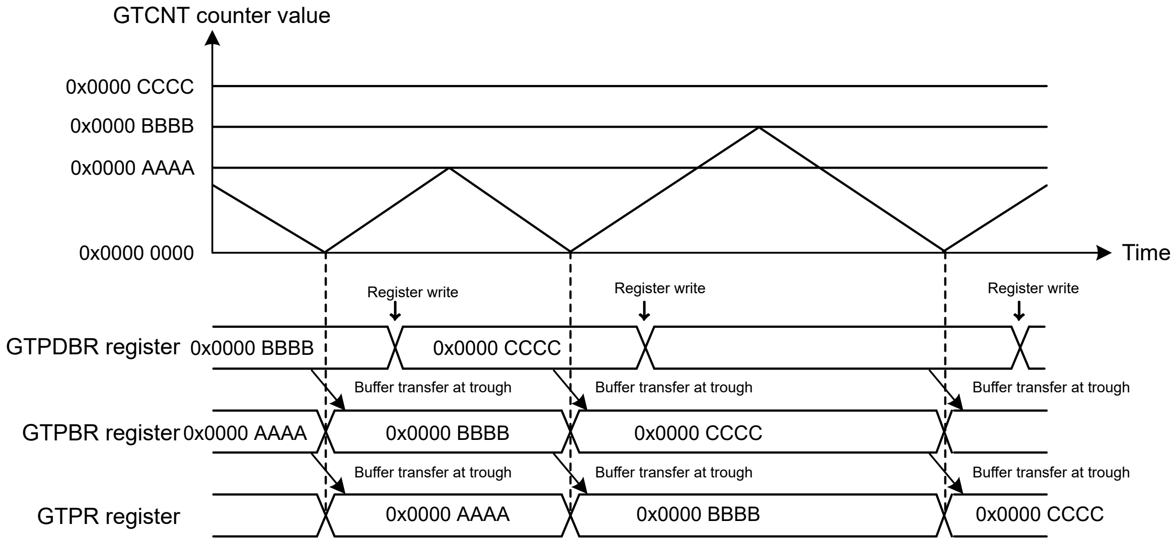


Figure 22.18 Example of GTPR double buffer operation with triangle waves

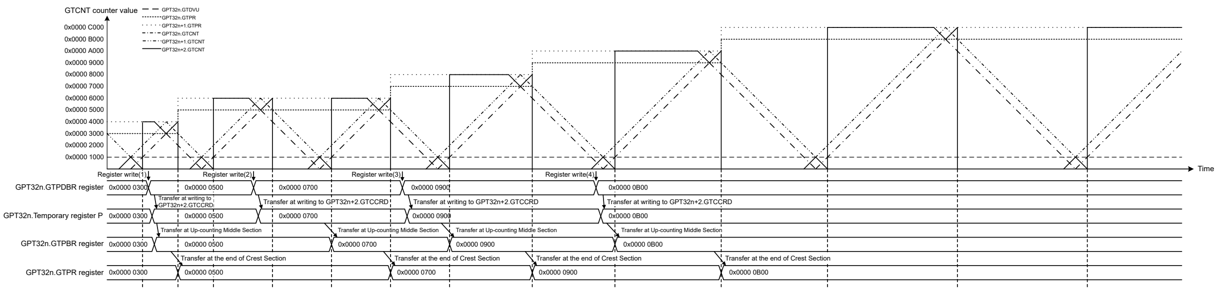


Figure 22.19 Example of GTPR double buffer operation with complementary PWM mode 1

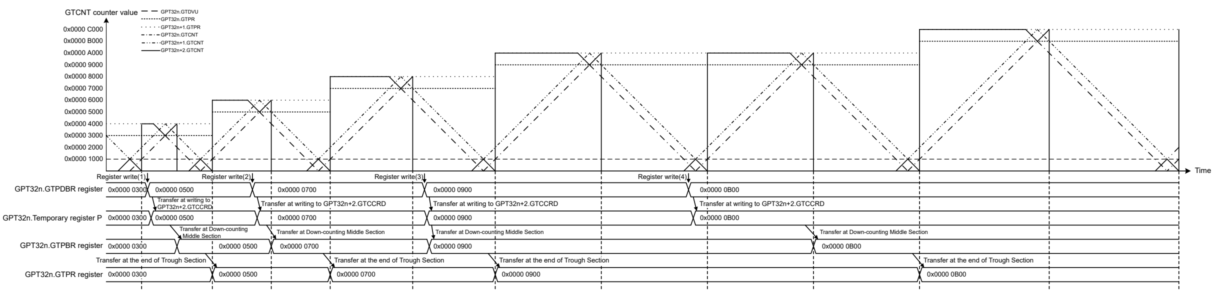


Figure 22.20 Example of GTPR double buffer operation with complementary PWM mode 2

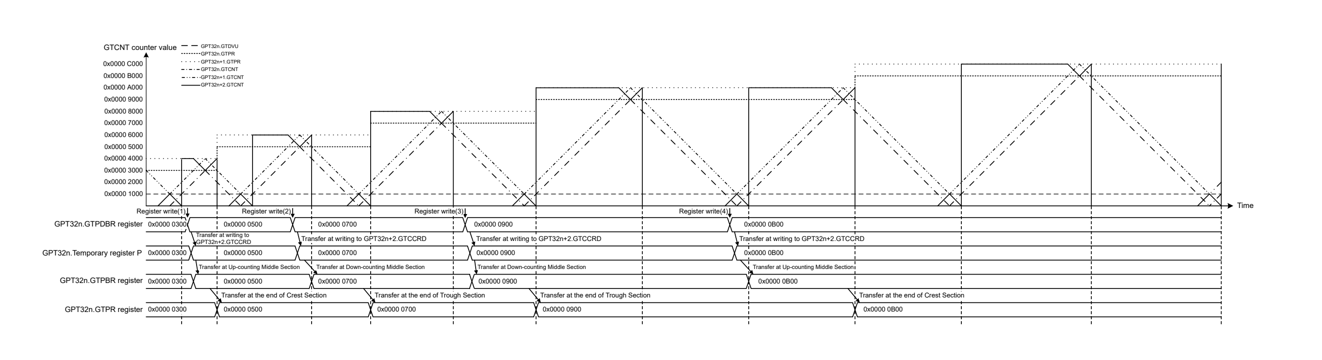


Figure 22.21 Example of GTPR double buffer operation with complementary PWM mode 3, 4

Table 22.19 Example for setting GTPR register buffer operation

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 22.16 and Figure 22.17, 000b or 0000b (saw-wave PWM mode 1) is set, and in Figure 22.18, 100b or 0100b (triangle-wave PWM mode 1) is set. In Figure 22.19 to Figure 22.21, 11xxb (complementary PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 22.16, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting). In Figure 22.17, after 10b is set in the GTUDDTYC[1:0] bits, 00b is set in the GTUDDTYC[1:0] bits (down-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set buffer operation	Set buffer operation with the GTBER.PR[1:0] bits. In Figure 22.16 and Figure 22.17, PR[1:0] = 01b. In Figure 22.18, PR[1:0] = 1xb. In Figure 22.19 to Figure 22.21, no PR[1:0] bits setting required.
7	Set buffer value	For buffer operation, set a value in one cycle after the current cycle in the GTPBR register. For double buffer operation, also set a value of the period for the cycle after the next cycle in the GTPDBR register.
8	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
9	Set buffer value for each cycle	For buffer operation, set a value in one cycle after the current cycle in the GTPBR register. For double buffer operation, also set a value of the period for the cycle after the next cycle in the GTPDBR register. In complementary PWM mode 1 (crest transfer), when the GTPDBR register is set in the up-counting trough or middle section, a value of the period for the next cycle is set. In other sections, a value of the period for the cycle after the next cycle is set. In complementary PWM mode 2 (trough transfer), when the GTPDBR register is set in the up-counting sect ion or the down-counting crest or middle sections, a value of the period for the next cycle is set. In other sections, a value of the period for the cycle after the next cycle is set. In complementary PWM mode 3 or 4 (crest/trough immediate transfer), when the GTPDBR register is set in a section other than the down-counting trough section, a value of the period for the next cycle is set. In the down-counting trough section, a value of the period for the cycle after the next cycle is set.

22.3.2.2 Buffer Operation for GTCCRA and GTCCRB Registers

GTCCRC can function as the GTCCRA buffer register and GTCCRD can function as the GTCCRC buffer register (double-buffer register for GTCCRA). Similarly, GTCCRE can function as the GTCCRB buffer register and GTCCRF can function as the GTCCRE buffer register (double-buffer register for GTCCRB).

To set GTCCRA or GTCCRB to function as a double buffer, set GTBER.CCRA[1:0] or GTBER.CCRB[1:0] to 10b or 11b. For single buffer operation, set 01b. To set GTCCRA or GTCCRB to not function as a buffer, set 00b.

In saw-wave one-shot pulse mode, triangle-wave PWM mode 3, complementary PWM mode, the buffer operations that specific each PWM output operation mode are performed regardless of the setting of GTBER.CCRA [1:0] bits and GTBER.CCRB [1:0] bits.

(1) When GTCCRA or GTCCRB Functions as Output Compare Register

In saw-wave one-shot pulse mode, triangle-wave PWM mode 3, complementary PWM mode, the buffer operations that specific each PWM output operation mode are performed regardless of the setting of GTBER.CCRA [1:0] bits and GTBER.CCRB [1:0] bits. For details, see section 22.3.3. PWM Output Operating Mode. Other than the specified PWM output operation mode, buffer transfer occurs in the following situations:

- Buffer transfer by overflow or underflow
  Buffer transfer is performed at an overflow (during up-counting) or an underflow (during down-counting) in saw-wave mode or in event count operation. In triangle-wave mode, buffer transfer is performed at a trough (triangle-wave PWM mode 1) or a crest and trough (triangle-wave PWM mode 2).
- Buffer transfer by counter clear
  In saw-wave mode or in event count operation, during counting, buffer transfer (which is the same as an overflow during up-counting or an underflow during down-counting) is performed by the counter clear sources similar to the case shown in section 22.3.2.1. GTPR Register Buffer Operation.
  In saw-wave, the buffer transfer of GTCCRm register by counter clearing is possible to be disabled with GTBER2.CCTCm bit (m = A, B).
  In triangle-wave mode, buffer transfer is not performed by the counter clear.
- Buffer transfer by compare match
  In saw-wave, the buffer transfer by the compare match of GTCCRm register enabled by GTBER2.CMTCm (m = A, B) bit is performed.
- Forcible buffer transfer
  When GTBER.CCRSWT bit is set to 1 while the count operation is stopped, the GTCCRA and the GTCCRB register buffer transfer are performed forcibly in saw-wave mode, in event count operation and in triangle-wave mode. Additionally, buffer transfer from the GTCCRD register to temporary register A and from the GTCCRF register to temporary register B are performed in saw-wave one-shot pulse mode or triangle-wave PWM mode 3.

When the setting of the GTBER.DBRTECm (m = A, B) bit is 1 in saw-wave one-shot pulse mode or triangle-wave PWM mode 3, transfer from the intermediate buffers to the GTCCRm (m = A, B) registers is repeated on a cyclic basis even while buffer transfer is disabled by the setting of the GTBER.BD[0] bit or buffer transfer extended skipping function (function for repeated double-buffer operation while buffer transfer is disabled). For details, see section 22.8.2.2. Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer.

Figure 22.22 to Figure 22.25 show examples of GTCCRA and GTCCRB buffer operation and Table 22.20 shows an example for setting GTCCRA and GTCCRB buffer operation.

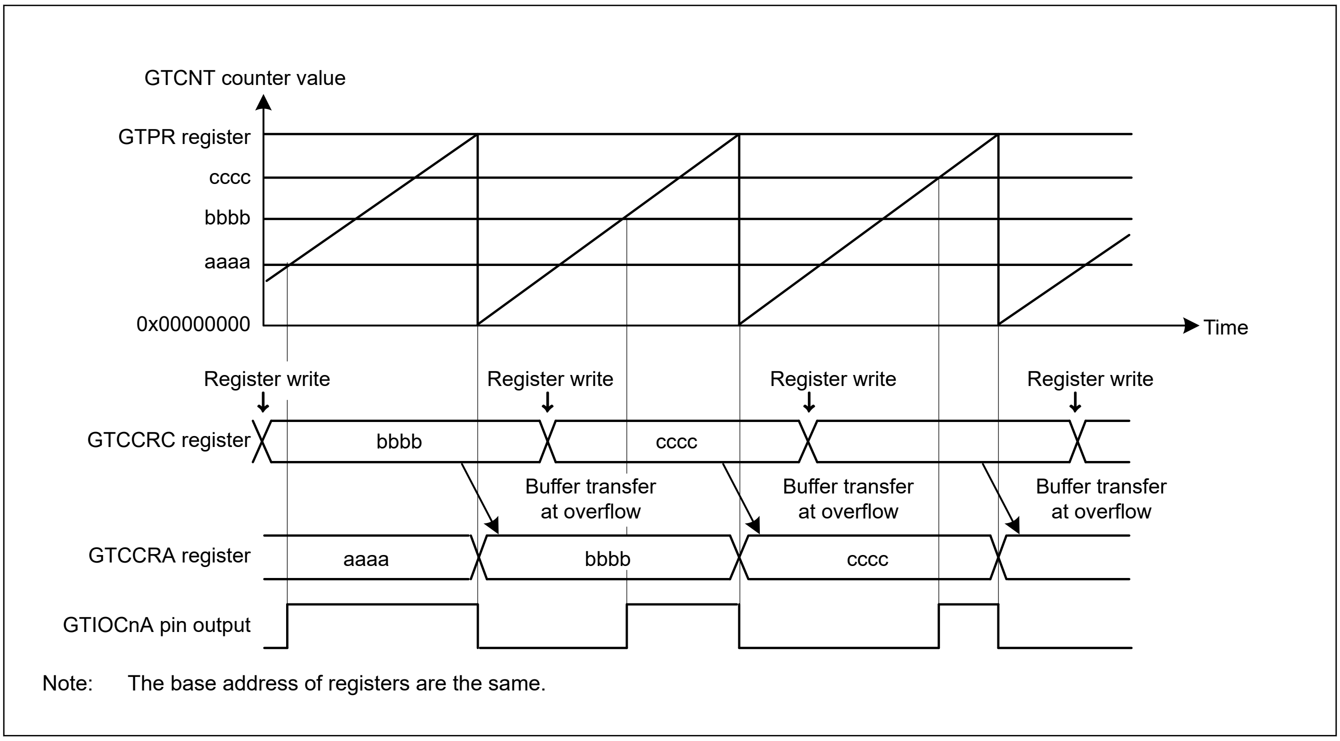


Figure 22.22 Example of GTCCRA and GTCCRB buffer operation with output compare, Saw-wave PWM mode 1 in up- counting, high output at GTCCRA compare match, and low output at cycle end

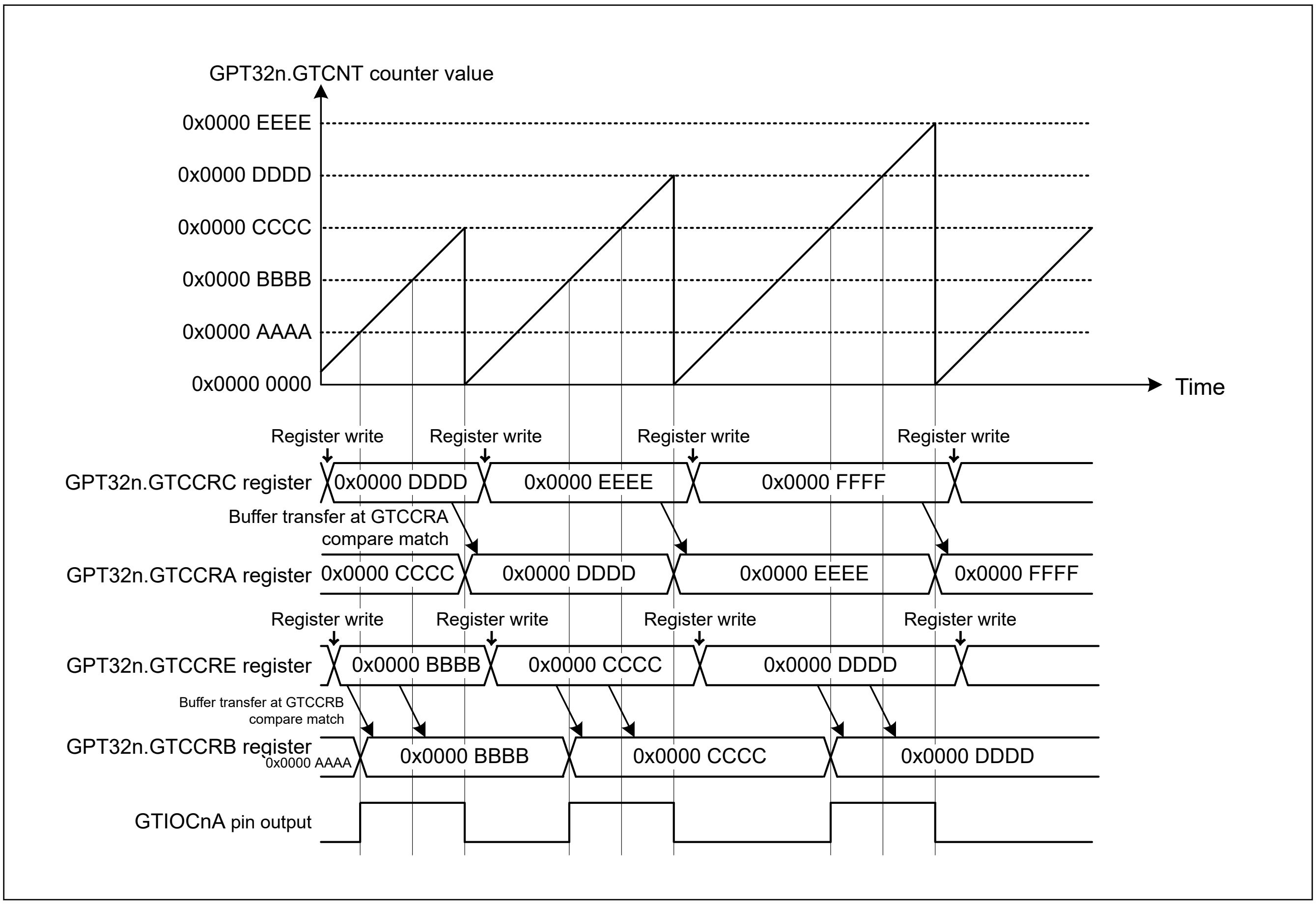
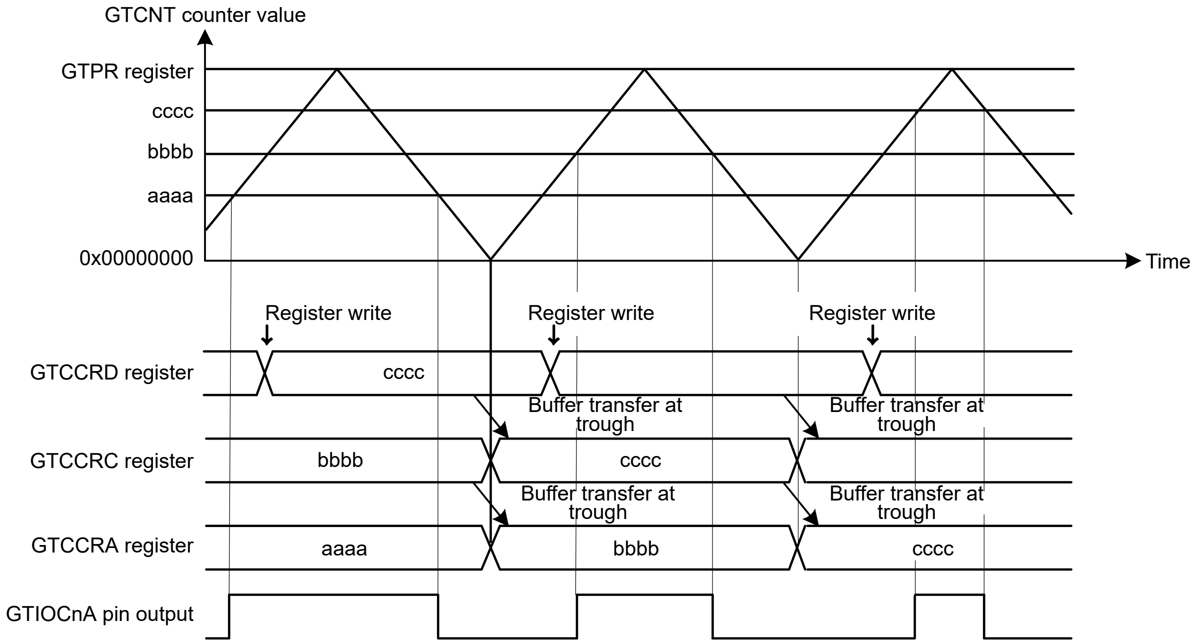
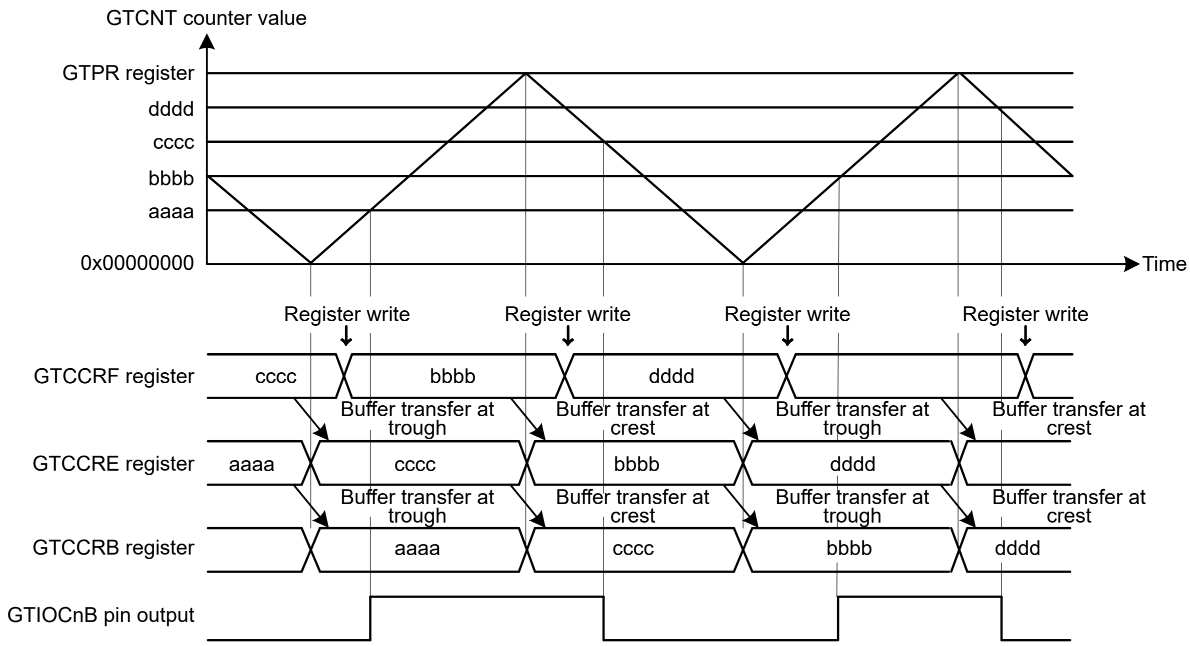


Figure 22.23 Example of GTCCRA and GTCCRB Registers Buffer Operation (Output Compare, Saw-Wave PWM Mode 2, Buffer Transfer at GTCCRA Register Compare Match, Counter Clearing, Low Output, Buffer Transfer at GTCCRB Register Compare Match, High Output) (n = 4 to 9)



Note: The base address of registers are the same.

Figure 22.24 Example of GTCCRA and GTCCRB double buffer operation with output compare, triangle waves, buffer operation at trough, output toggled at GTCCRA compare match, and output retained at cycle end



Note: The base address of registers are the same.

Figure 22.25 Example of GTCCRA and GTCCRB double buffer operation with output compare, triangle waves, buffer operation at both troughs and crests, output toggled at GTCCRB compare match, and output retained at cycle end

Table 22.20 Example for setting GTCCRA and GTCCRB buffer operation for output compare

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 22.22, 000b or 0000b (saw-wave PWM mode 1) is set, in Figure 22.23, 0010b (saw-wave PWM mode 2) is set, in Figure 22.24, 100b or 0100b (triangle-wave PWM mode 1) is set, and in Figure 22.25, 101b or 0101b (triangle-wave PWM mode 2) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 22.22, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Other than the saw-wave PWM mode2, set the cycle in the GTPR register. In the saw-wave PWM mode2, select the counter clear source compare match register GTCCRx (x = A to F) by GTCSR.CSCMSC[2:0] bits and set the cycle in that register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 22.22, GTIOA[4:0] = 00110b, in Figure 22.23, GTIOA[4:0] = 00101b and GTIOB[1:0] = 10b, in Figure 22.24, GTIOA[4:0] = 00011b, and in Figure 22.25, GTIOB[4:0] = 00011b.
7	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set buffer operation	Set buffer operation with the CCRA[1:0] and CCRB[1:0] bits in the GTBER register. In Figure 22.22, CCRA[1:0] = 01b, in Figure 22.23, CCRA[1:0] = 01b and CCRB[1:0] = 01b, in Figure 22.24, CCRA[1:0] = 1xb, and in Figure 22.25, CCRB[1:0] = 1xb.
9	Set compare match value*1	Set the GTIOCnA pin transition in the GTCCRA register and the GTIOCnB pin transition in the GTCCRB register.
10	Set buffer value	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or 1 cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRD and GTCCRF registers, respectively.
11	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
12	Set buffer value for each cycle	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or 1 cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRD and GTCCRF registers, respectively.

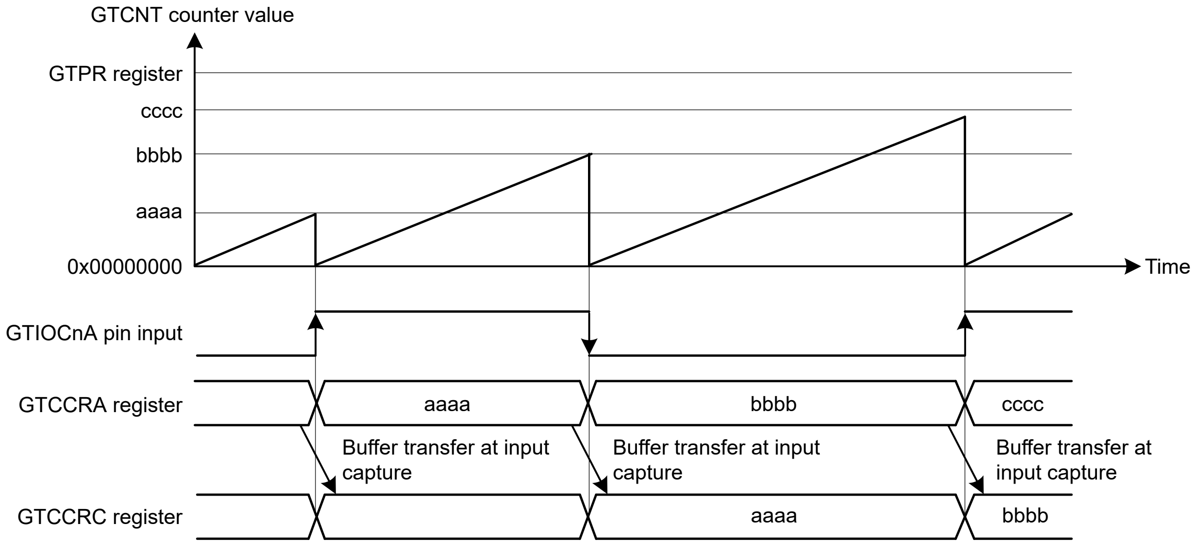
Note: n: 0 to 13
m: A, B

Note 1. When using PWM Delay Generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

(2) When GTCCRA or GTCCRB Functions as Input Capture Register

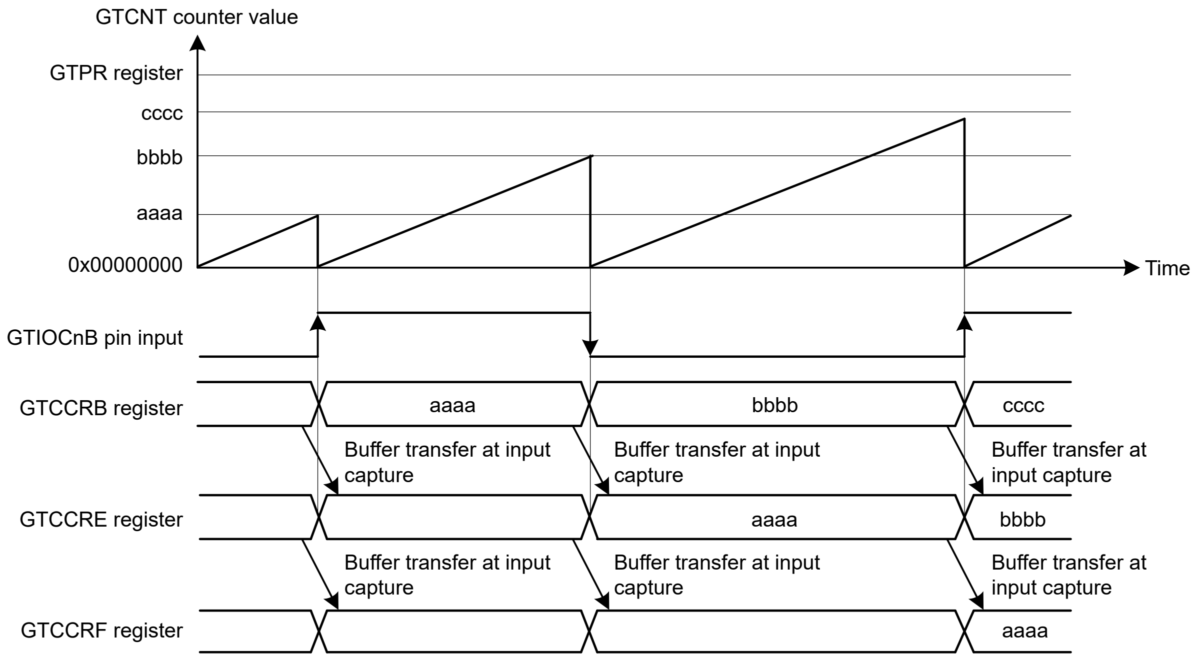
When an input capture is generated, the GTCNT counter value is transferred to GTCCRA and GTCCRB and the stored GTCCRA and GTCCRB register values are transferred to the buffer registers. In input capture operation, the buffer transfer is not performed by the counter clear.

Figure 22.26 and Figure 22.27 show examples of GTCCRA and GTCCRB buffer operation and Table 22.21 shows an example for setting GTCCRA and GTCCRB buffer operation.



Note: The base address of registers are the same.

Figure 22.26 Example of GTCCRA and GTCCRB buffer operation with input capture at both edges of GTIOCnA input, saw waves in up-counting, and GTCNT counter cleared at both edges of GTIOCnA input



Note: The base address of registers are the same.

Figure 22.27 Example of GTCCRA and GTCCRB double buffer operation with input capture at both edges of GTIOCnB input, saw waves in up-counting, and GTCNT counter cleared at both edges of GTIOCnB input

Table 22.21 Example for setting GTCCRA and GTCCRB buffer operation for input capture

No.	Step Name	Description
1	Set operating mode and counter clear sources	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits and count clear source with the GTCSR register. In Figure 22.26, MD[2:0] = 000b or MD[3:0] = 0000b (saw-wave PWM mode 1) and GTCSR = 0x00000F00, and in Figure 22.27, MD[2:0] = 000b or MD[3:0] = 0000b (saw-wave PWM mode 1) and GTCSR = 0x0000F000.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 22.26, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Other than the saw-wave PWM mode2, set the cycle in the GTPR register. In the saw-wave PWM mode2, select the counter clear source compare match register GTCCRx (x = A to F) by GTCSR.CSCMSC[2:0] bits and set the cycle in that register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Select input capture source	Select input capture source in the GTICASR register and GTICBSR register. In Figure 22.26, GTICASR = 0x00000F00, and in Figure 22.27, GTICBSR = 0x0000F000. In GPT324 to GPT329 to perform input capture with sources of other channels, select a group that performs inter-channel cooperation by the GTICCR.ICmGRP bit (m = A or B). For the output-side channel of the input capture sources, set the GTICCR register to enables the input capture sources to output to other channels. For input-side channels, set the GTIOmSR.mSOC bit (m = A or B) to 1 to enable input capture with sources of other channels.
7	Set buffer operation	Set buffer operation with the CCRA and CCRB bits in the GTBER register. In Figure 22.26, CCRA[1:0] = 01b, and in Figure 22.27, CCRB[1:0] = 1xb.
8	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

22.3.2.3 Buffer Operation for GTADTRA and GTADTRB Registers

The GTADTBRA register can function as the GTADTRA buffer register and the GTADTDBRA register can function as the GTADTBRA buffer register (double buffer register for the GTADTRA register). Similarly, the GTADTBRB register can function as the GTADTRB buffer register and the GTADTDBRB register can function as the GTADTBRB buffer register (double buffer register for the GTADTRB register).

To set the GTADTRA or GTADTRB register to function as a double buffer, set the GTBER.ADTDA or ADTDB bit to 1. For single buffer operation, set 0. Not to function as buffer, set the GTBER.ADTTA[1:0] or ADTTB[1:0] bits to 00b.

The buffer transfer timing can be set with the ADTTA[1:0] and ADTTB[1:0] bits to an overflow (in up-counting) or an underflow (in down-counting) in saw-wave mode, with ADTTA[1:0] and ADTTB[1:0] bits to 01b for a crest, to 10b for a trough, or to 11b for both crest and trough in triangle-wave mode or complementary PWM mode.

In saw-wave mode, when the ADTTA[1:0] and ADTTB[1:0] bits are set to value other than 00b and in count operation, the buffer transfer, by similar counter clearing sources in section 22.3.2.1. GTPR Register Buffer Operation, is performed in the same way at an overflow (in up-counting) or an underflow (in down-counting).

In complementary PWM mode, the buffer transfer is performed after one GTCLK cycle from GTCCRD register write of slave channel 2.

In saw-wave mode, the buffer transfer of GTADTRm register by counter clearing can be disabled by GTBER2.CCTADm (m = A, B) bit setting.

In saw-wave mode, the buffer transfer of GTADTRm register by its own compare match can be enabled by the GTBER2.CMTADm (m = A, B) bit setting.

Figure 22.28 to Figure 22.32 show examples of buffer operation of the GTADTRA and GTADTRB registers, and Table 22.22 shows an example for setting buffer operation of the GTADTRA and GTADTRB registers.

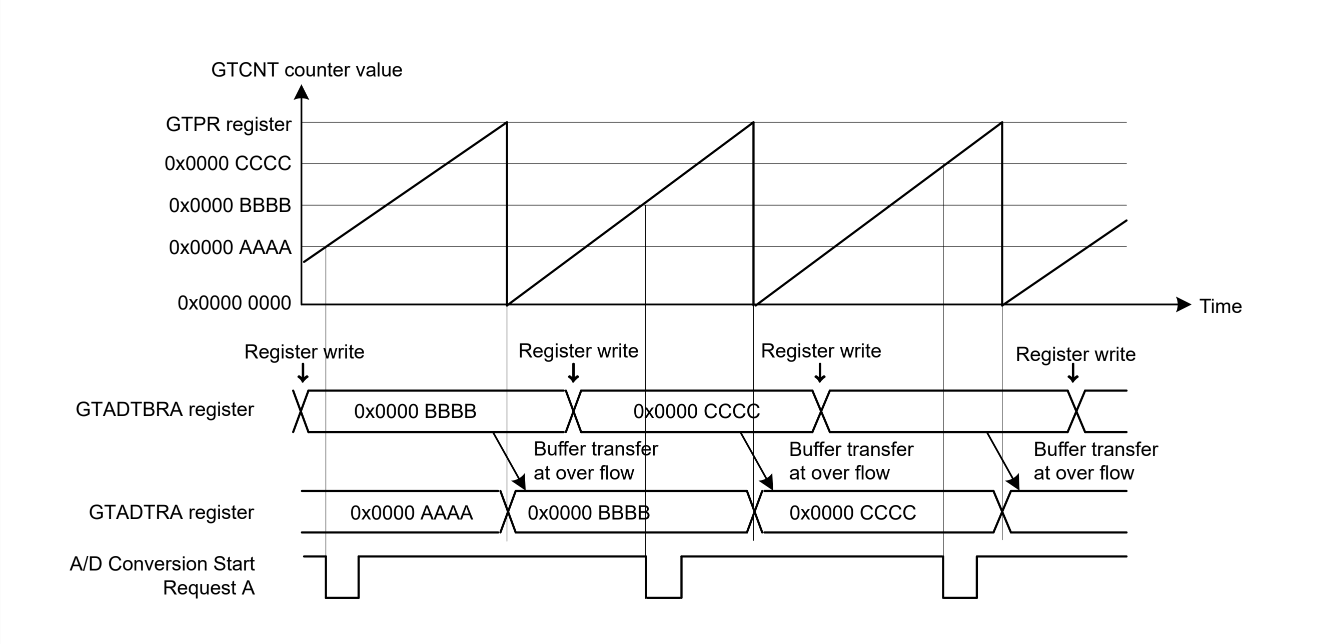


Figure 22.28 Example of Buffer Operation of the GTADTRA and GTADTRB Registers (Saw Waves in Up-Counting, A/D Conversion Start Request Generated by Up-Counting)

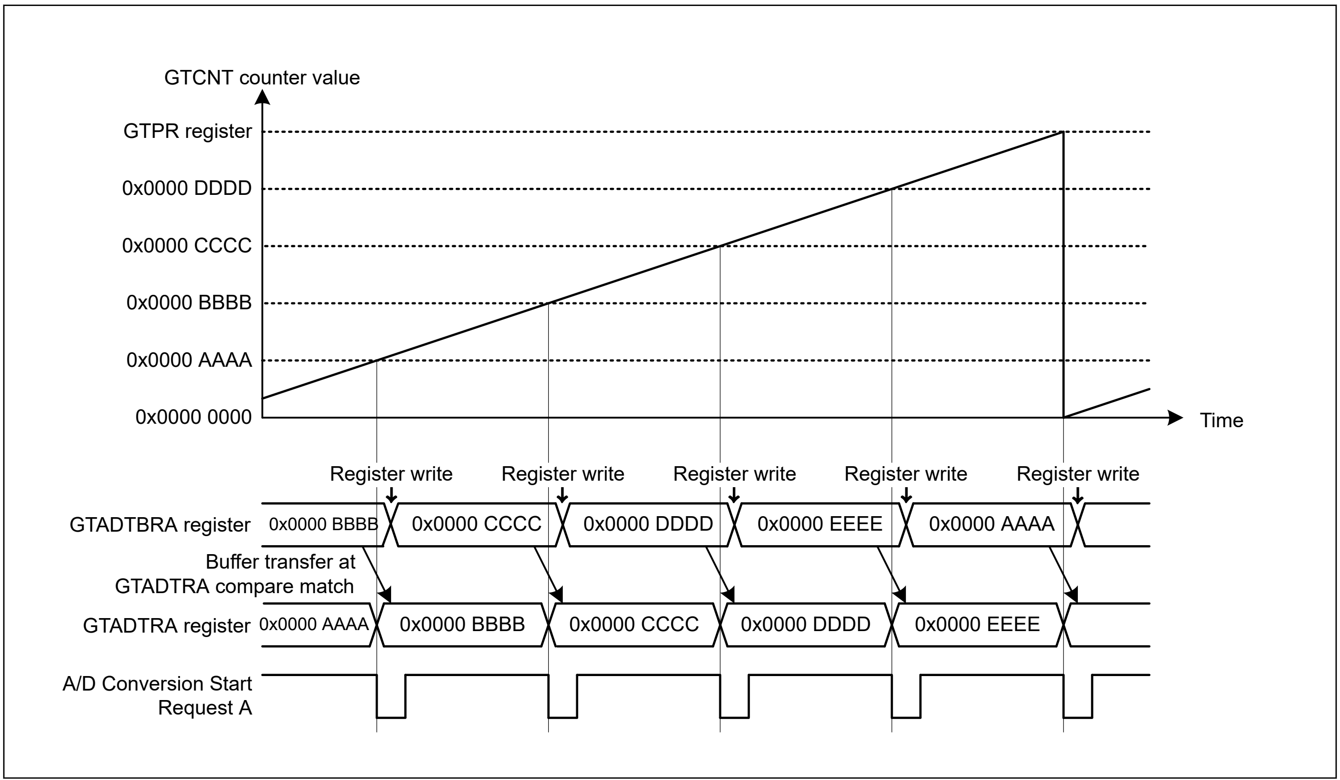


Figure 22.29 Example of Buffer Operation of the GTADTRA and GTADTRB Registers (Saw Waves in Up-Counting, Buffer Transfer at GTADTRA Compare Match, A/D Conversion Start Request Generated by Up-Counting)

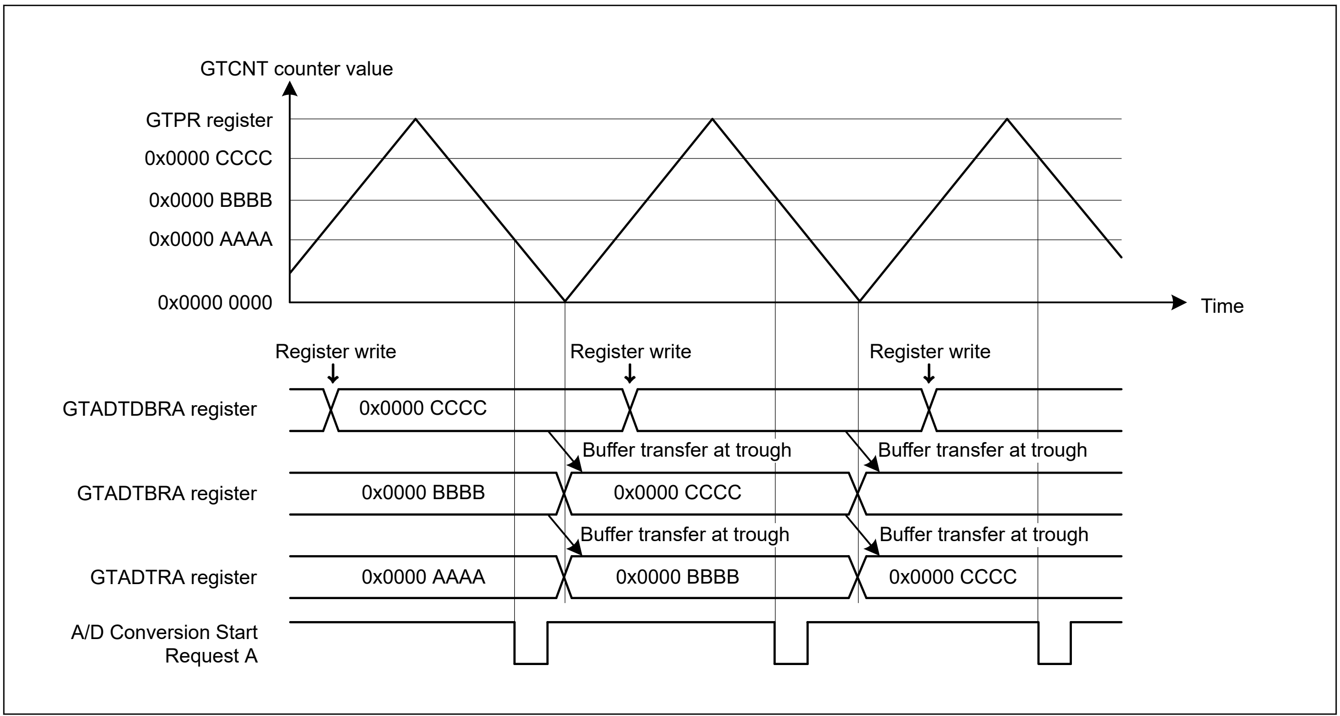


Figure 22.30 Example of Double Buffer Operation of the GTADTRA and GTADTRB Registers (Triangle Waves, Buffer Transfer at Troughs, A/D Conversion Start Request Generated by Down-Counting)

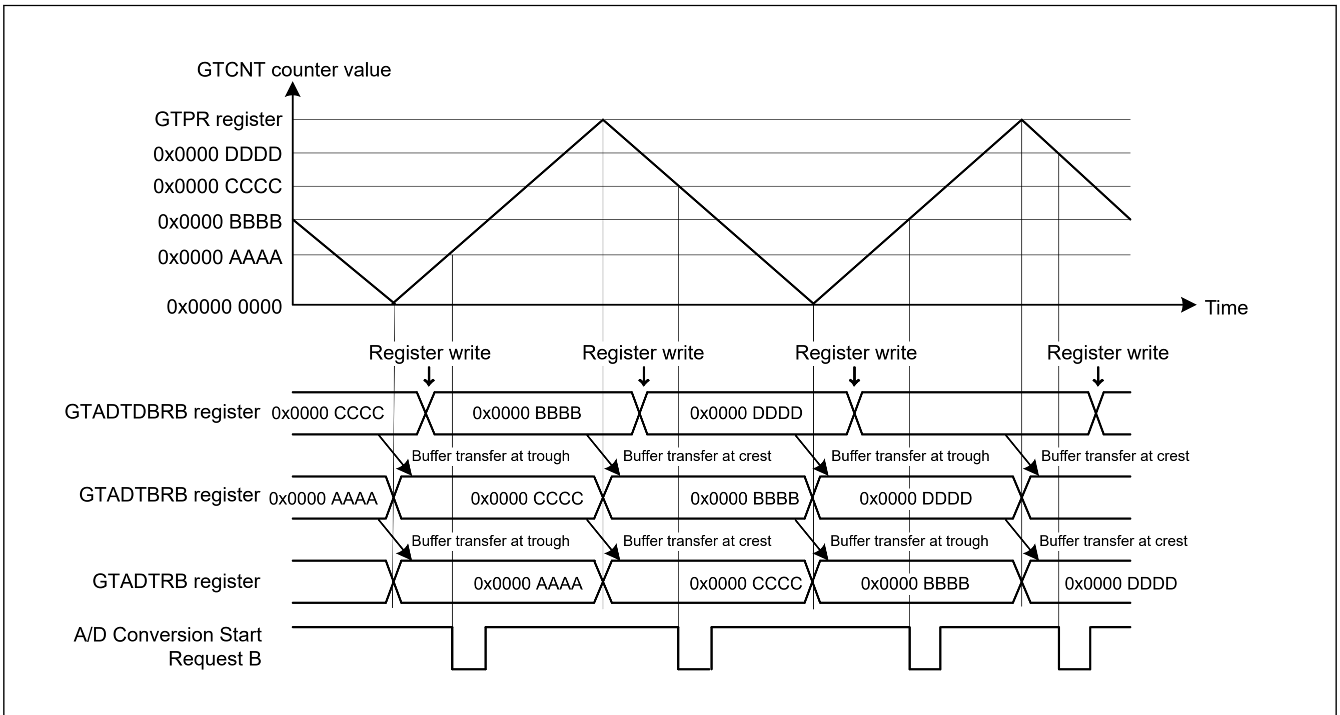


Figure 22.31 Example of Double Buffer Operation of the GTADTRA and GTADTRB Registers (Triangle Waves, Buffer Transfer at Both Troughs and Crests, A/D Conversion Start Request Generated by Both Up- and Down-Counting)

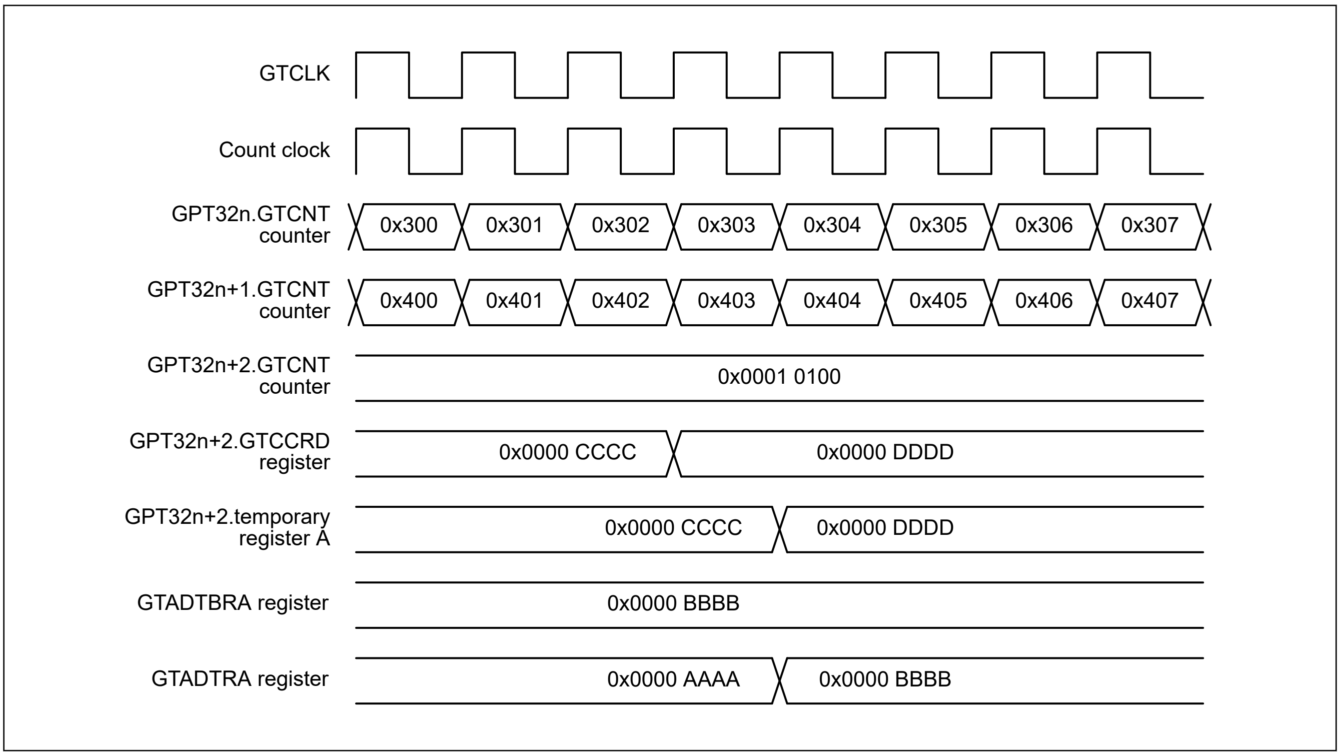


Figure 22.32 Example of Buffer Operation of the GTADTRA and GTADTRB Registers at the GTCCRD Register of Slave Channel 2 Updating in Complementary PWM Mode

Table 22.22 Example for Setting Buffer Operation of the GTADTRA and GTADTRB registers (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. (In Figure 22.28 and Figure 22.29, 000b or 0000b (saw-wave PWM mode 1) is set, in Figure 22.30 and Figure 22.31, 100b, 101b, 110b, 0100b, 0101b, or 0110b (triangle-wave PWM mode) is set.)
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. (In Figure 22.28 and Figure 22.29, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).)
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Other than the saw-wave PWM mode 2, set the cycle in the GTPR register. In the saw-wave PWM mode 2, select the counter clear source compare match register GTCCRx (x = A to F) by GTCSR.CSCMSC[2:0] bits and set the cycle in that register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set buffer operation	Set buffer operation with the ADTTA[1:0], ADTTB[1:0], ADTDA, and ADTDB bits in GTBER register. To perform buffer transfer at a compare match of the ADTRAm (m = A, B) register, set the GTBER2.CMTADm bit to 1. (In Figure 22.28, ADTTA[1:0] bits = 01b, 10b, or 11b and ADTDA bit = 0, in Figure 22.29, CMTADA bit = 1, in Figure 22.30, ADTTA[1:0] bits = 10b and ADTDA bit = 1, and in Figure 22.31, ADTTB [1:0] bits = 11b and ADTDB bit = 1.)
7	Set compare match value	Set the A/D conversion start request point in the GTADTRA and GTADTRB registers.
8	Set buffer value	For buffer operation, set the A/D conversion start request point in one cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers. For double buffer operation, also set the A/D conversion start request point in two cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers.

Table 22.22 Example for Setting Buffer Operation of the GTADTRA and GTADTRB registers (2 of 2)

No.	Step Name	Description
9	Enable A/D conversion start request	Set to enable A/D conversion start request with the ADTRAUEN, ADTRADEN, ADTRBUEN, and ADTRBDEN bits in the GTINTAD register. (In Figure 22.28 and Figure 22.29, ADTRAUEN bit = 1, in Figure 22.30, ADTRADEN bit = 1, and in Figure 22.31, ADTRBUEN bit = 1 and ADTRBDEN bit = 1.)
10	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
11	Set buffer value of each cycle	For buffer operation, set the A/D conversion start request point in one cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers. For double buffer operation, also set the A/D conversion start request point in two cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers.

22.3.2.4 Buffer operation for GTIOA and GTIOB

The GTOLBR.GTIOAB[4:0] bits can function as the buffer register of the GTIOR.GTIOA[4:0] bits and the GTOLBR.GTIOBB[4:0] bits can function as the buffer register of the GTIOR.GTIOB[4:0] bits.

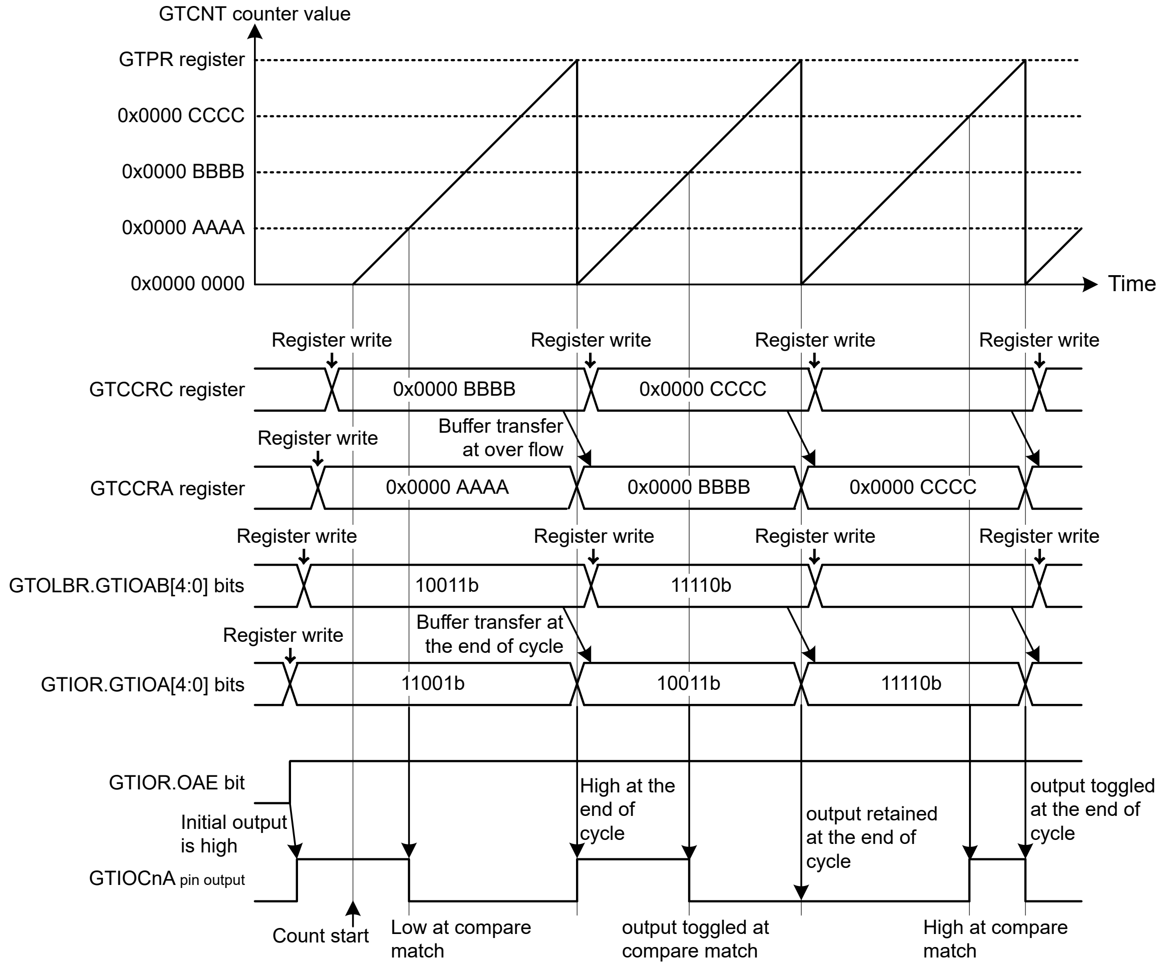
Buffer transfer timing can be set with the GTBER2.OLTTm[1:0] bits (m = A, B). It can be selected from the end of cycle or GTCCR register compare match (in saw-wave mode), from crest, trough, or both crest and trough (in triangle-wave mode and complementary PWM mode). In the case that the GTBER2.OLTTm[1:0] bits are 00b, buffer transfer is not performed.

In complementary PWM mode, the setting to convert GTIOA and GTIOB by buffer transfer during the dead time of complementary PWM is prohibited.

Therefore, when changing GTIOA and GTIOB by buffer transfer, it is necessary to set the crest and trough timings not to overlap with the dead time and change GTOLBR.GTIOAB and GTOLBR.GTIOBB.

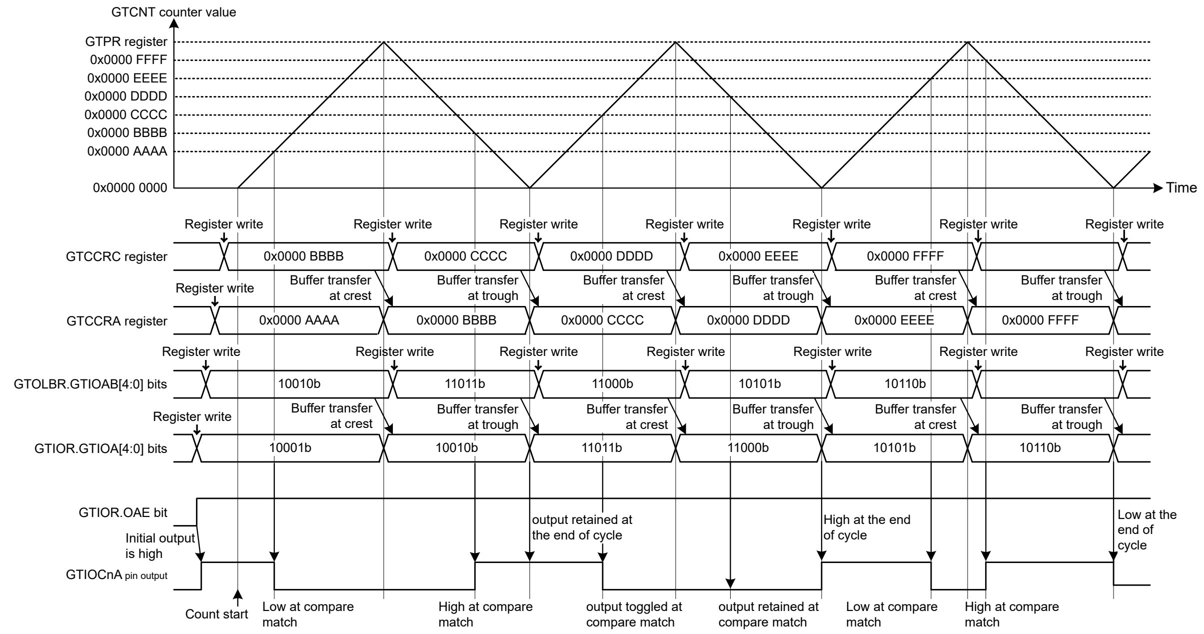
When the buffer transfer timing is crest, set the GTCCRm (m = A, C, E) register to satisfy GTCCRm < GTPR. And when the buffer transfer timing is trough, set the GTCCRm register to satisfy GTDVU < GTCCRm.

Figure 22.33 to Figure 22.35 show examples of buffer operation of the GTIOR.GTIOA[4:0] and GTIOR.GTIOB[4:0] bits. Table 22.23 shows an example for setting buffer operation of the GTIOR.GTIOA[4:0] and GTIOR.GTIOB[4:0] bits.



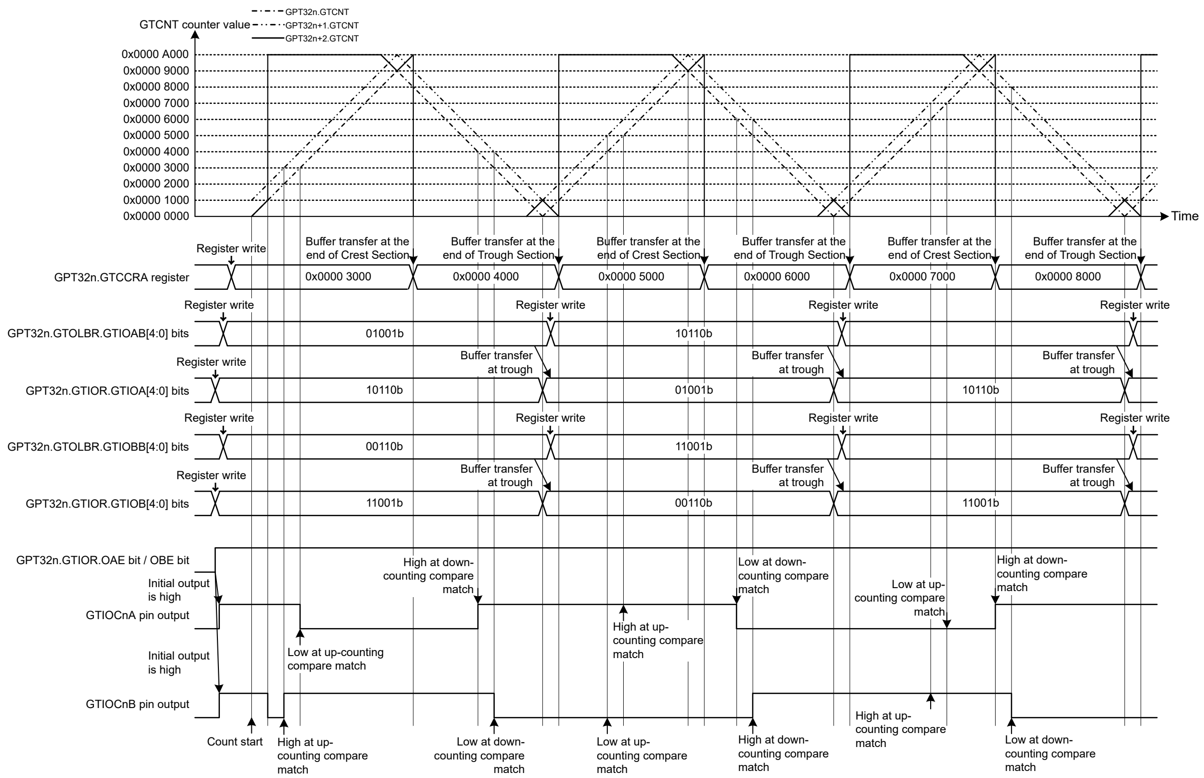
Note: Initial output is synchronized to the count clock.

Figure 22.33 Example of Buffer Operation of the GTIOA and GTIOB Bits (Up-Counting in Saw-Wave PWM Mode 1, Buffer Transfer at the End of Cycle)



Note: Initial output is synchronized to the count clock.

Figure 22.34 Example of Buffer Operation of the GTIOA and GTIOB Bits (Triangle-Wave PWM Mode 2, Buffer Transfer at Crests and Troughs)



Note: Initial output is synchronized to the count clock.

Figure 22.35 Example of Buffer Operation of the GTIOA and GTIOB Bits (Complementary PWM Mode 3, Buffer Transfer at Troughs)

Table 22.23 Example for Setting Buffer Operation of the GTIOA and GTIOB

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[3:0]. (In Figure 22.33, 0000b (saw-wave PWM mode 1) is set, in Figure 22.34, 0101b (triangle-wave PWM mode 2) is set, in Figure 22.35, 1110b (complementary PWM mode 3) is set.)
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. (In Figure 22.33, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).)
3	Select count clock	Select the count clock with GTCR.TPCS[3:0] bits.
4	Set cycle	Other than the saw-wave PWM mode 2, Set the cycle in GTPR. In the saw-wave PWM mode 2, Select the counter clear source compare match register GTCCRx (x = A to F) by GTCSR.CSCMSC[2:0] bits and set the cycle in that register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR. (In Figure 22.33, GTIOA[4:0] = 11001b, in Figure 22.34, GTIOA[4:0] = 10001b, in Figure 22.35, GTIOA[4:0] = 10110b and GTIOB[4:0] = 11001b.)
7	Enable GTIOCnm pin output	Set to enable the GTIOCnm pin output with OAE and OBE in GTIOR.
8	Set buffer operation	Set buffer operation with OLTTm[1:0] bits in GTBER2 register. (In Figure 22.33, 01b is set in OLTTA[1:0], in Figure 22.34, 11b is set in OLTTA[1:0], in Figure 22.35, 10b is set in OLTTA[1:0] and 10b is set in OLTTB[1:0].)
9	Set buffer value	For buffer operation, set the GTIOCnm pin function in one cycle after the current cycle (in saw-wave mode and in triangle-wave/complementary PWM mode with buffer transfer at crest or trough) or half cycle after the current cycle (in triangle-wave/complementary PWM mode with buffer transfer at crest and trough) in the GTOLBR register.
10	Start count operation	Set GTCR.CST to 1 to start count operation.
11	Set buffer value of each cycle	For buffer operation, set the GTIOCnm pin function in one cycle after the current cycle (in saw-wave mode and in triangle-wave/complementary PWM mode with buffer transfer at crest or trough) or half cycle after the current cycle (in triangle-wave/complementary PWM mode with buffer transfer at crest and trough) in the GTOLBR register.

Note: n = 4 to 9
m = A, B

22.3.3 PWM Output Operating Mode

The GPT can output PWM waveforms to the GTIOCnA or GTIOCnB pin (n = 0 to 13) by a compare match between the GTCNT counter and GTCCRA or GTCCRB.

By setting GTDTCR, GTDVU, and GTDVD, the compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

In complementary PWM mode, it is possible to output PWM waveforms (positive phase, negative phase) with dead time that guarantees the linearity of the PWM output pulse width near 0% and 100% duty.

In saw-wave mode other than saw-wave PWM mode 2, or triangle-wave mode, or the master channel of complementary PWM mode, the signal synchronized with the PWM cycle can be output from the GTCPPOn output terminal by setting the GTIOR.PSYE bit to 1. The GTCPPOn output is toggled at the end of cycle in saw-wave mode or at the timing of crest / trough / GTCNT counter clearing in triangle-wave mode or complementary PWM mode. The initial output of GTCPPOn is low, and it becomes high when the count starts.

22.3.3.1 Saw-Wave PWM Mode 1

In saw-wave PWM mode 1, GTCNT performs saw-wave (half-wave) operation by setting the cycle in GTPR and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 13) when a GTCCRA or GTCCRB compare match occurs. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the cycle end according to the GTIOR setting.

When 0 is set in the GTIOR.OxEOCD (x = A, B) bit and the timing of the end of cycle and the timing of GTCCRx register compare match are the same time, the output pin performs along the PWM output setting for the end of cycle set by the GTIOR.GTIOx[3:2] bit.

When 1 is set in the GTIOR.OxEOCD bit, GTIOCnx output is retained.

Figure 22.36 shows an example of saw-wave PWM mode 1 operation, and Table 22.24 shows an example for setting saw-wave PWM mode 1.

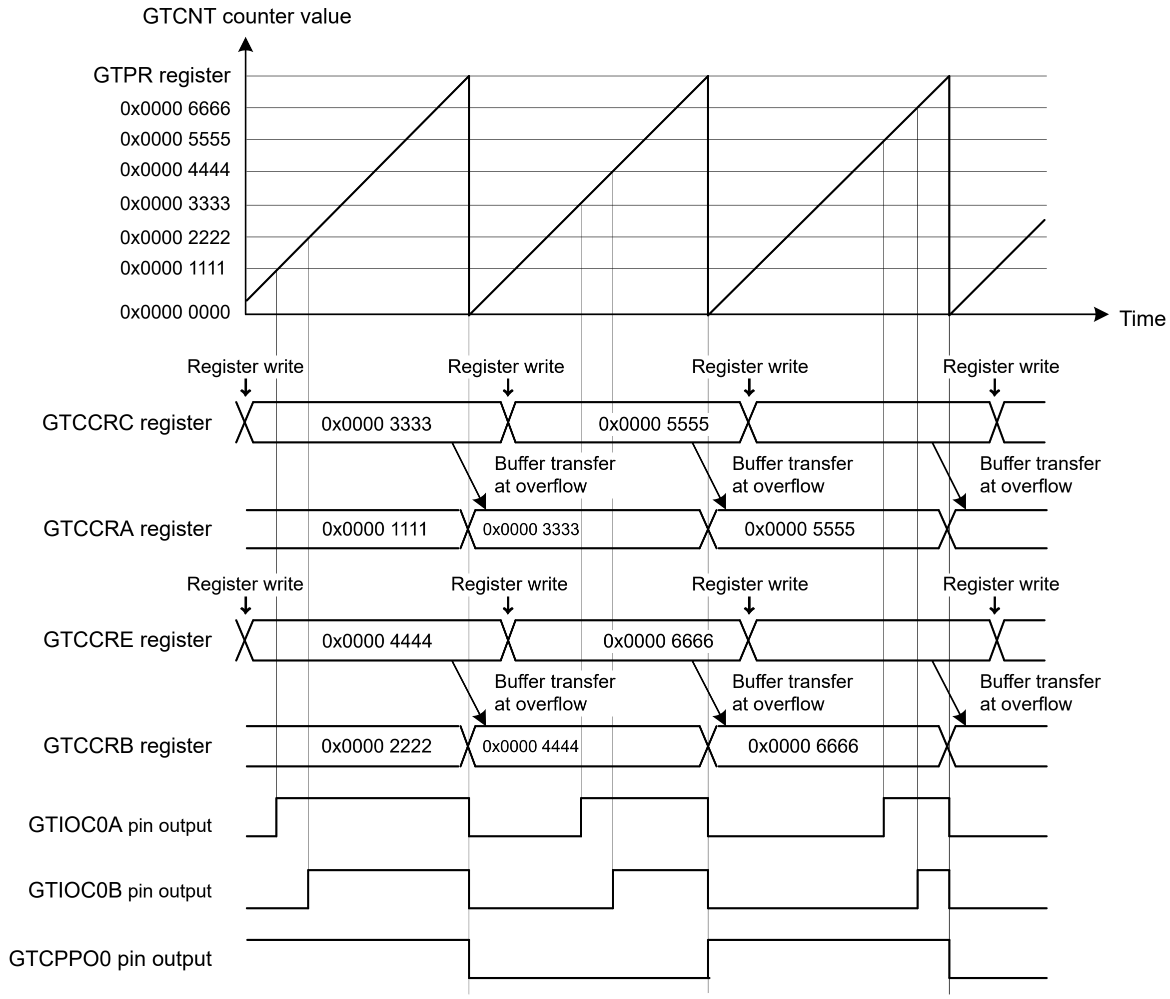


Figure 22.36 Example of saw-wave PWM mode 1 operation with up-counting, buffer operation, high output at GTCCRA/GTCCRB compare match, low output at cycle end, and GTIOR.PSYE = 1

Table 22.24 Example for setting saw-wave PWM mode 1 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 22.36, 000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 22.36, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 22.36, GTIOA[4:0] = 00110b and GTIOB[4:0] = 00110b.
7	Enable GTCPPOn pin output	Set to enable/disable the GTCPPOn pin output with the PSYE bit in the GTIOR register.

Table 22.24 Example for setting saw-wave PWM mode 1 (2 of 2)

No.	Step Name	Description
8	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
9	Set buffer operation	Set buffer operation with the CCRA[1:0] and CCRB[1:0] bits in the GTBER register. In Figure 22.36, CCRA[1:0] = 01b and CCRB[1:0] = 01b.
10	Set compare match value*1	Set the GTIOCnA pin transition in the GTCCRA register and the GTIOCnB pin transition in the GTCCRB register.
11	Set buffer value	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.
12	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
13	Set buffer value for each cycle	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.

Note: n: 0 to 13
m: A, B

Note 1. When using PWM Delay Generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

22.3.3.2 Saw-Wave PWM Mode 2

The saw-wave PWM mode 2 is a mode in which the GTCNT counter is operated as a saw wave by up-counting without using the GTPR register, and the PWM waveform is output by the compare match of the GTCCRA and GTCCRB registers. The pin output level can be selected from low output, high output, or toggle output separately for a compare match according to the GTIOR register setting.

The GTIOCnA pin is used as an output pin. Use the GTIOR.GTIOB[1:0] bits for setting the GTIOCnA pin output at a compare match of the GTCCRB register.

When a counter clear occurs due to the GTCNT counter clearing source selected in the GTCSR register, this is handled at the end of cycle and PWM output operation at the end of the cycle selected with the GTIOR.GTIOA[3:2] bits is performed. If a counter clear (at the end of cycle) conflicts with a PWM output change due to a GTCCRx (x = A, B) register compare match, PWM output operation is performed at the end of cycle (in the case of the GTIOR.OxEOCD bit = 0) or the PWM output is retained (in the case of GTIOR.OxEOCD bit = 1).

Figure 22.37 to Figure 22.39 shows an example of saw-wave PWM mode 2 operation. Table 22.25 shows an example for setting saw-wave PWM mode 2.

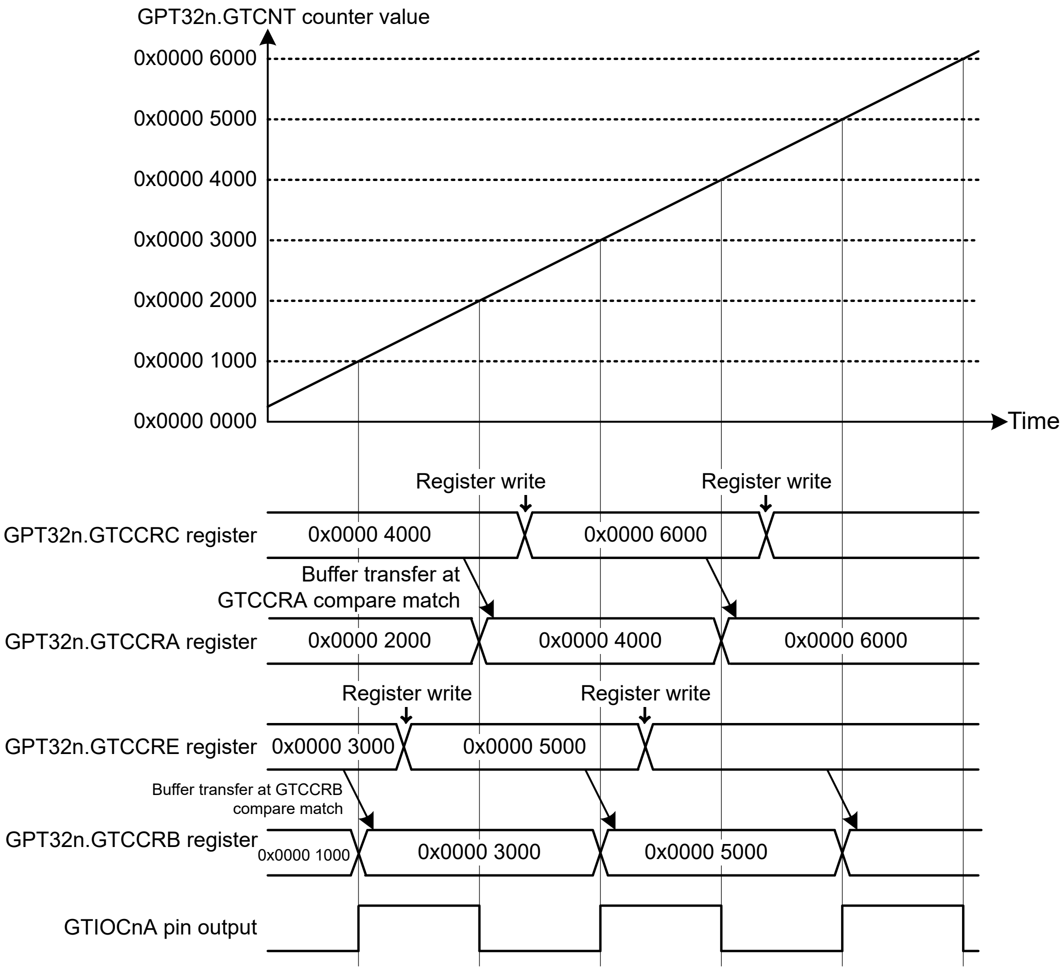


Figure 22.37 Example of Saw-Wave PWM Mode 2 Operation (Low Output at GTCCRA Register Compare Match, High Output at GTCCRB Register Compare Match, Single Buffer Operation, No Clear Setting) (n = 4 to 9)

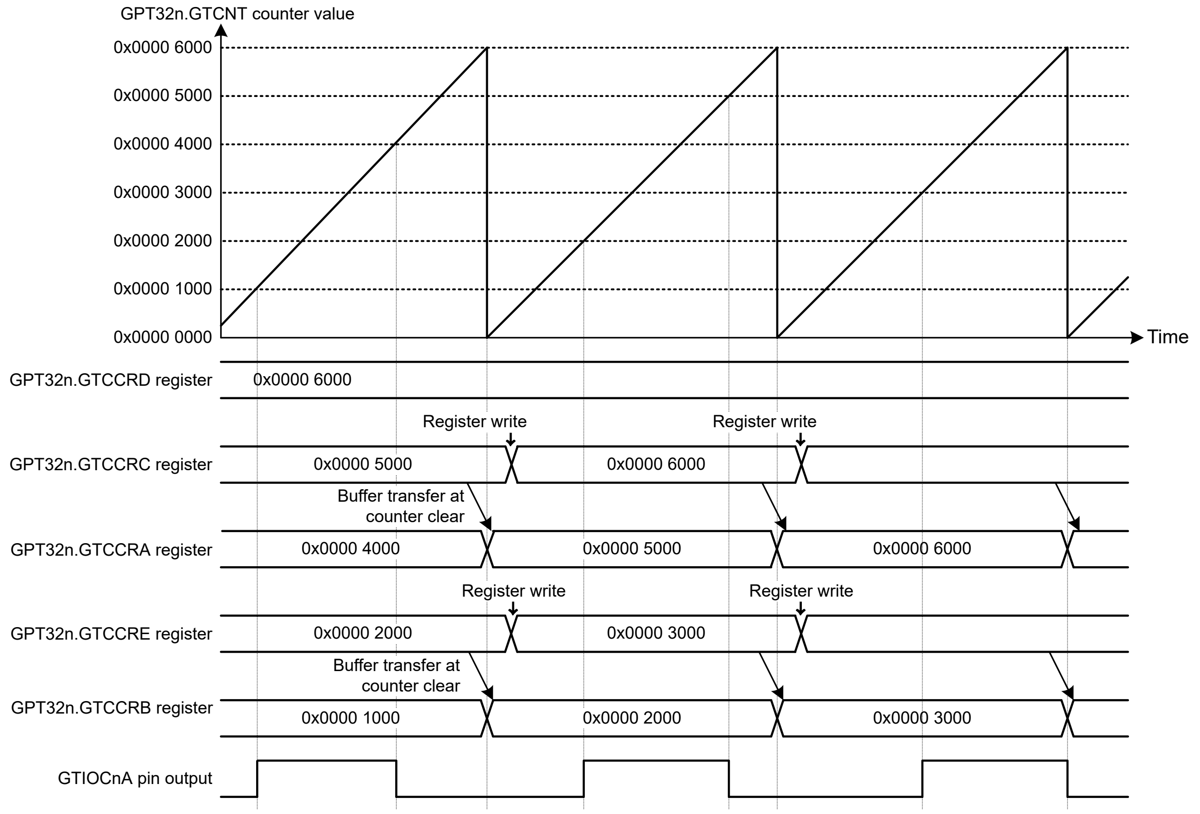


Figure 22.38 Example of Saw-Wave PWM Mode 2 Operation (Low Output at GTCCRA Register Compare Match, High Output at GTCCRB Register Compare Match, Low Output at the End of Cycle, Single Buffer Operation, Cleared at GTCCRD Register Compare Match, GTIOR.OAEOCD Bit = 0) (n = 4 to 9)

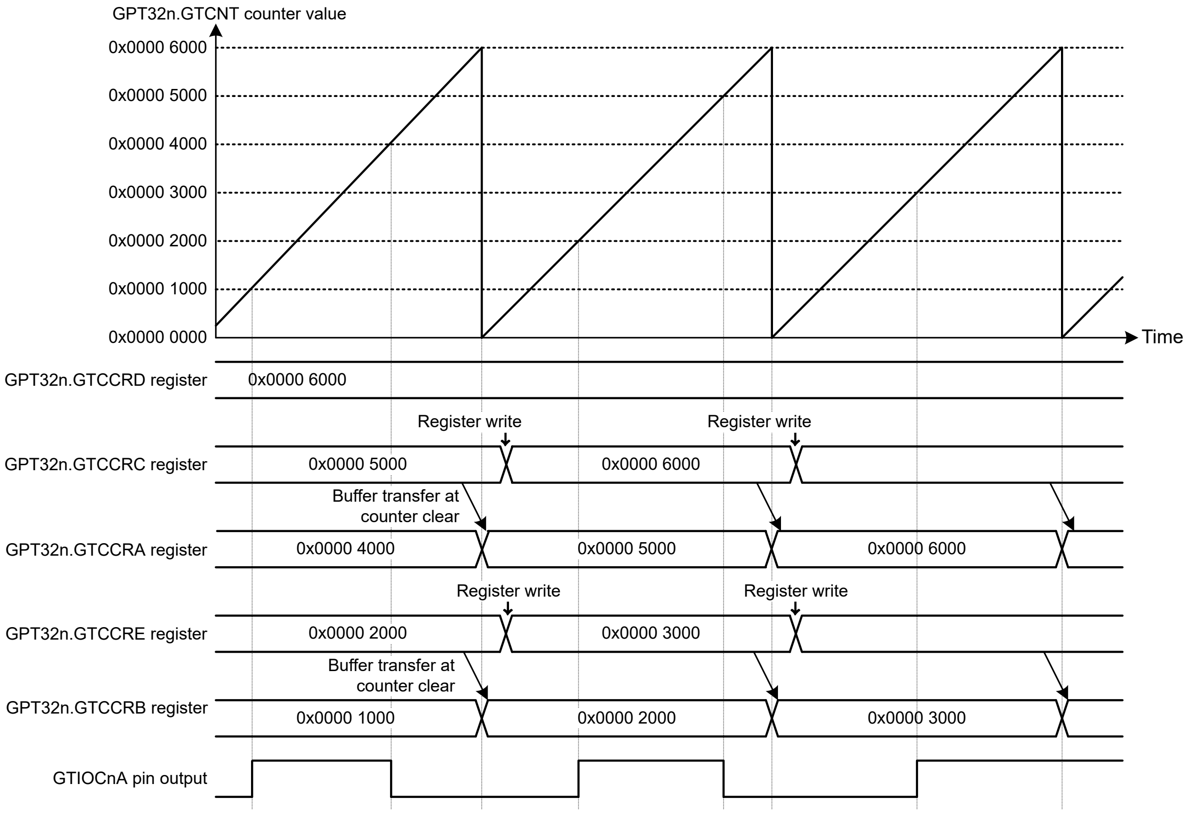


Figure 22.39 Example of Saw-Wave PWM Mode 2 Operation (Low Output at GTCCRA Register Compare Match, High Output at GTCCRB Register Compare Match, Low Output at the End of Cycle, Single Buffer Operation, Cleared at GTCCRD Register Compare Match, GTIOR.OAEOCD Bit =1) (n = 4 to 9)

Table 22.25 Example for Setting Saw-wave PWM mode 2

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[3:0]. (In Figure 22.37 to Figure 22.39, 0010b (saw-wave PWM mode 2) is set.)
2	Select count clock	Select the count clock with GTCR.TPCS[3:0] bits.
3	Set initial value for counter	Set the initial value in the GTCNT counter.
4	Set GTIOCnA pin function	Set the GTIOCnA pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR. (In Figure 22.37 to Figure 22.39, GTIOA[4:0] = 00001b and GTIOB[1:0] = 10b.)
5	Enable GTIOCnA pin output*1	Set to enable the GTIOCnA pin output with OAE in GTIOR.
6	Set buffer operation	Set buffer operation with CCRA[1:0] and CCRB[1:0] bits in GTBER (In Figure 22.37 to Figure 22.39, CCRA[1:0] = 01b and CCRB[1:0] = 01b.)
7	Set compare match value*1	Set the GTIOCnA pin changing point in GTCCRA and GTCCRB.
8	Set buffer value	For buffer operation, set the GTCCRA register value and the GTCCRB register value which used in one cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTCCRA register value and the GTCCRB register value which used in two cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.
9	Start count operation	Set GTCR.CST to 1 to start count operation.
10	Set buffer value of each cycle	For buffer operation, set the GTCCRA register value and the GTCCRB register value which used in one cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTCCRA register value and the GTCCRB register value which used in two cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.

Note: n: 4 to 9
Note 1. When using PWM Delay Generation circuit, change the order of setting for the GTIOCnA pin output enable and setting for a compare match value.

22.3.3.3 Saw-Wave One-Shot Pulse Mode

The saw-wave one-shot pulse mode is a mode in which the cycle is set in GTPR, the GTCNT counter performs saw-wave (half-wave) operation and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 13) at a compare match of GTCCRA or GTCCRB with buffer operation fixed.

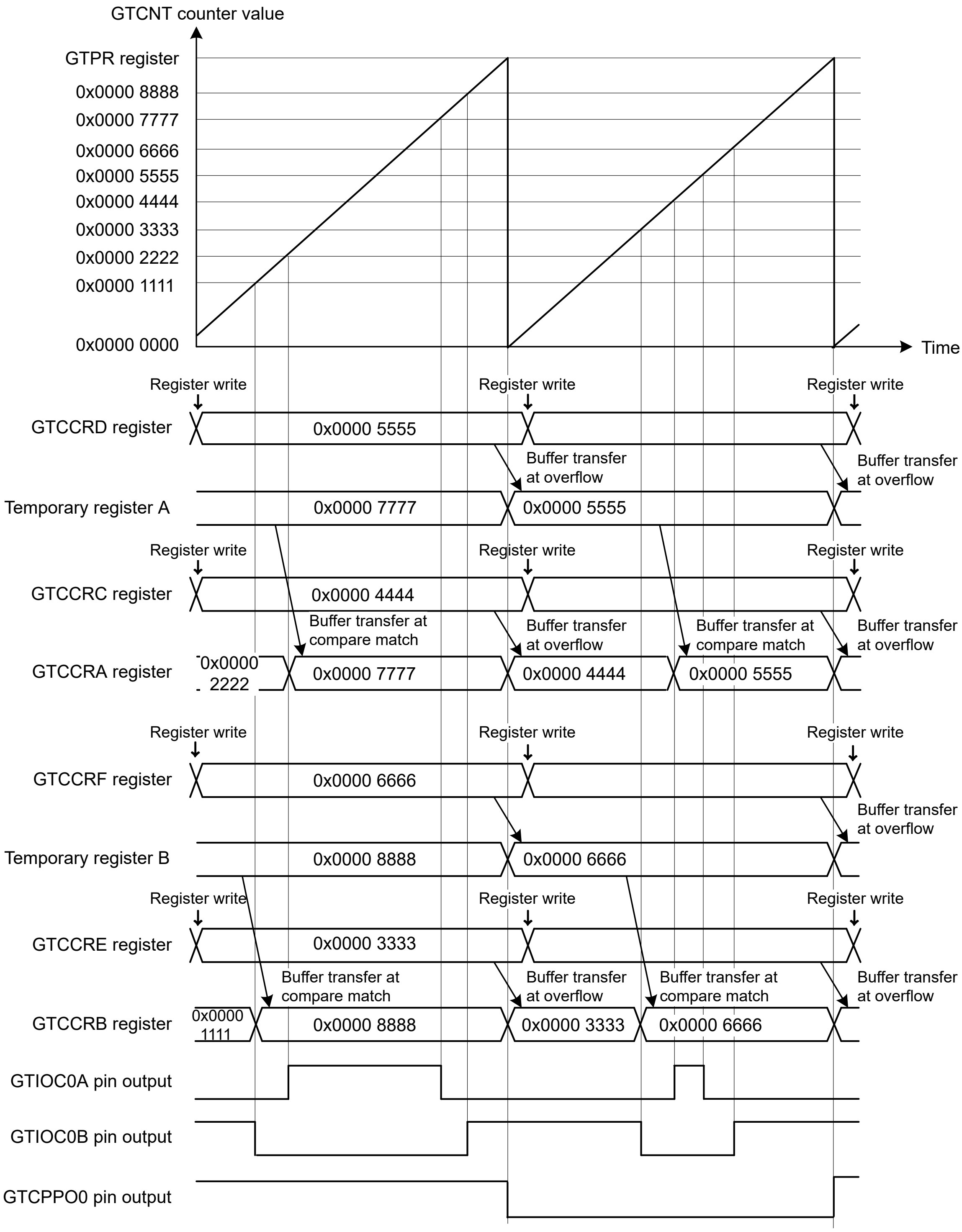
Buffer operation in saw-wave one-shot pulse mode is different from the usual buffer operation. Buffer transfer is performed from:

- GTCCRC to GTCCRA at the cycle end
- GTCCRE to GTCCRB at the cycle end
- GTCCRD to temporary register A at the cycle end
- GTCCRF to temporary register B at the cycle end
- Temporary register A to GTCCRA at a GTCCRA compare match
- Temporary register B to GTCCRB at a GTCCRB compare match.

The pin output value can be selected from low output, high output, or toggled output separately for a compare match and the cycle end according to the GTIOR setting. When the GTBER.CCRSWT bit is set to 1 while count operation is stopped, the buffer is transferred forcibly from the GTCCRD register to temporary register A and from the GTCCRF register to temporary register B. By setting GTDTCR, GTDVU, and GTDVD, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

When the GTBER.DBRTECm (m = A, B) bit is set to 1, transfer from an intermediate buffer to the GTCCRm register is repeated on a cyclic basis with using the temporary register x (x = C, E) and temporary register m which operate as intermediate buffers for the GTCCRx and GTCCRm registers, respectively, even while buffer transfer is disabled (repeated double buffer operation function during disabling of buffer transfer). For details, see section 22.8.2.2. Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer.

Figure 22.40 shows an example of saw-wave one-shot pulse mode operation, and Table 22.26 shows an example for setting saw-wave one-shot pulse mode.



Note: The base address of registers are the same.

Figure 22.40 Example of saw-wave one-shot pulse mode operation with up-counting, low output from the GTIOCnA pin and high output from the GTIOCnB pin at count start, output toggled at GTCCRA/ GTCCRB compare match, output retained at cycle end, and GTIOR.PSYE = 1

Table 22.26 Example setting for saw-wave one-shot pulse mode

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 22.40, 001b (saw-wave one-shot pulse mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 22.40, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 22.40, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
7	Enable GTCPPOn pin output	Set to enable/disable the GTCPPOn pin output with the PSYE bit in the GTIOR register.
8	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
9	Set compare match value*1	Set the GTIOCnA pin transition immediately after the count start in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.
10	Set forcible buffer transfer*1	Set the GTBER.CCRSWT bit to 1 to transfer buffer register data forcibly.
11	Set buffer value	For buffer operation, set the GTIOCnA pin transition in one cycle after the current cycle in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.
12	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
13	Set buffer value for each cycle	For buffer operation, set the GTIOCnA pin transition in one cycle after the current cycle in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.

Note: n: 0 to 13
m: A, B

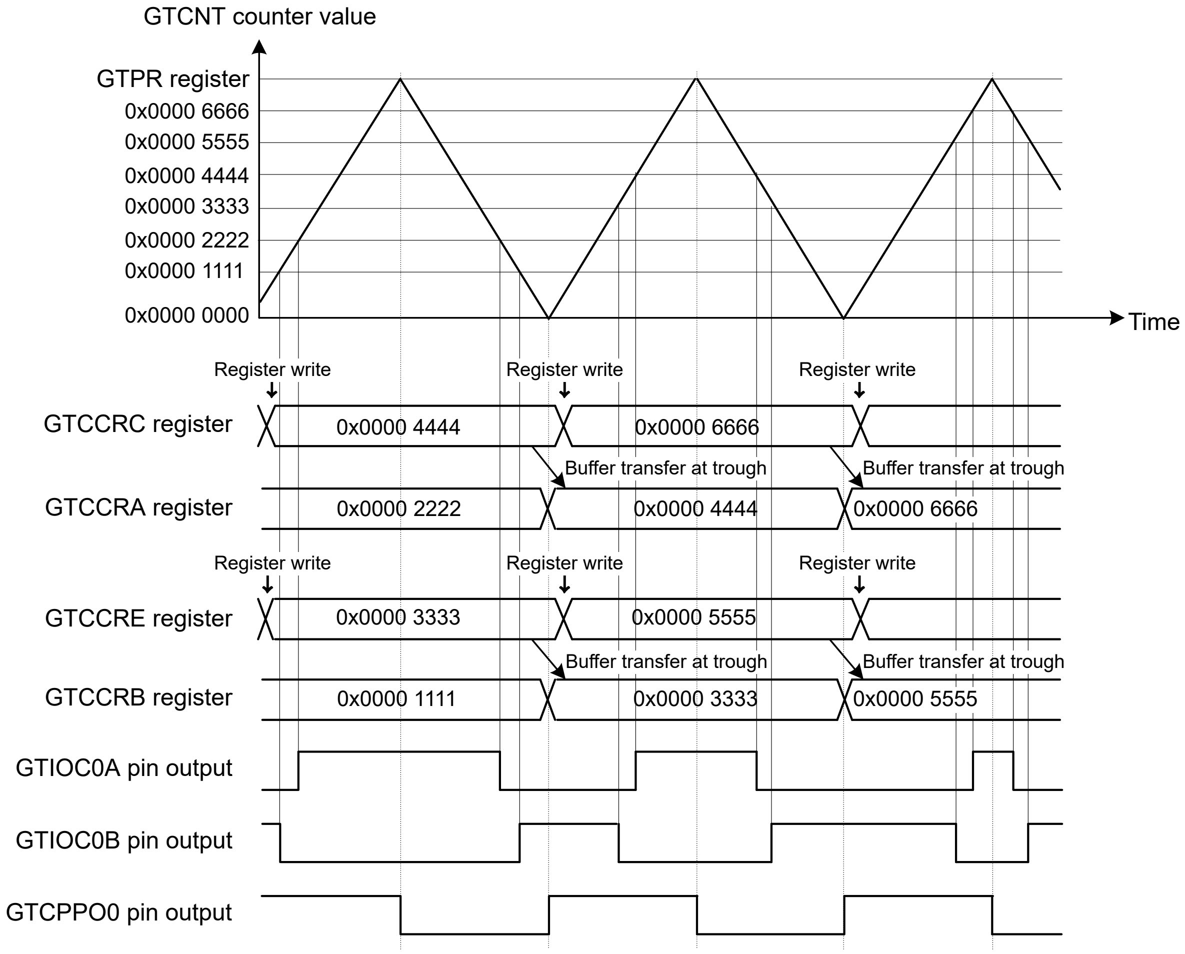
Note 1. When using PWM Delay Generation circuit, change the order of Enable GTIOCnm pin output and Set compare match value + Set forcible buffer transfer.

22.3.3.4 Triangle-Wave PWM Mode 1 (32-Bit Transfer at Trough)

The triangle-wave PWM mode 1 is a mode in which the cycle is set in GTPR. The GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 13) when a GTCCRA or GTCCRB compare match occurs. Buffer transfer is performed at the trough. The pin output value can be selected from low output, high output, or toggled output separately for a compare match and for the cycle end according to the GTIOR setting.

By setting GTDTCR, GTDVU, and GTDVD, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 22.41 shows an example of a triangle-wave PWM mode 1 operation, and Table 22.27 shows an example for setting a triangle-wave PWM mode 1.



Note: The base address of registers are the same.

Figure 22.41 Example of triangle-wave PWM mode 1 operation with buffer operation, low output from the GTIOCnA pin and high output from the GTIOCnB pin at count start, output toggled at GTCCRA/ GTCCRB register compare match, output retained at cycle end, and GTIOR.PSYE = 1

Table 22.27 Example setting for triangle-wave PWM mode 1 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 22.41, 100b (triangle-wave PWM mode 1) is set.
2	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
3	Set cycle	Set the cycle in the GTPR register.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 22.41, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
6	Enable GTCPPOn pin output	Set to enable/disable the GTCPPOn pin output with the PSYE bit in the GTIOR register.
7	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set buffer operation	Set buffer operation with the CCRA[1:0] and CCRB[1:0] bits in the GTBER register. In Figure 22.41, CCRA[1:0] = 01b and CCRB[1:0] = 01b.
9	Set compare match value*1	Set the GTIOCnA and GTIOCnB pins transitions in the GTCCRA and GTCCRB registers, respectively.

Table 22.27 Example setting for triangle-wave PWM mode 1 (2 of 2)

No.	Step Name	Description
10	Set buffer value	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.
11	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
12	Set buffer value for each cycle	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.

Note: n: 0 to 13
m: A, B

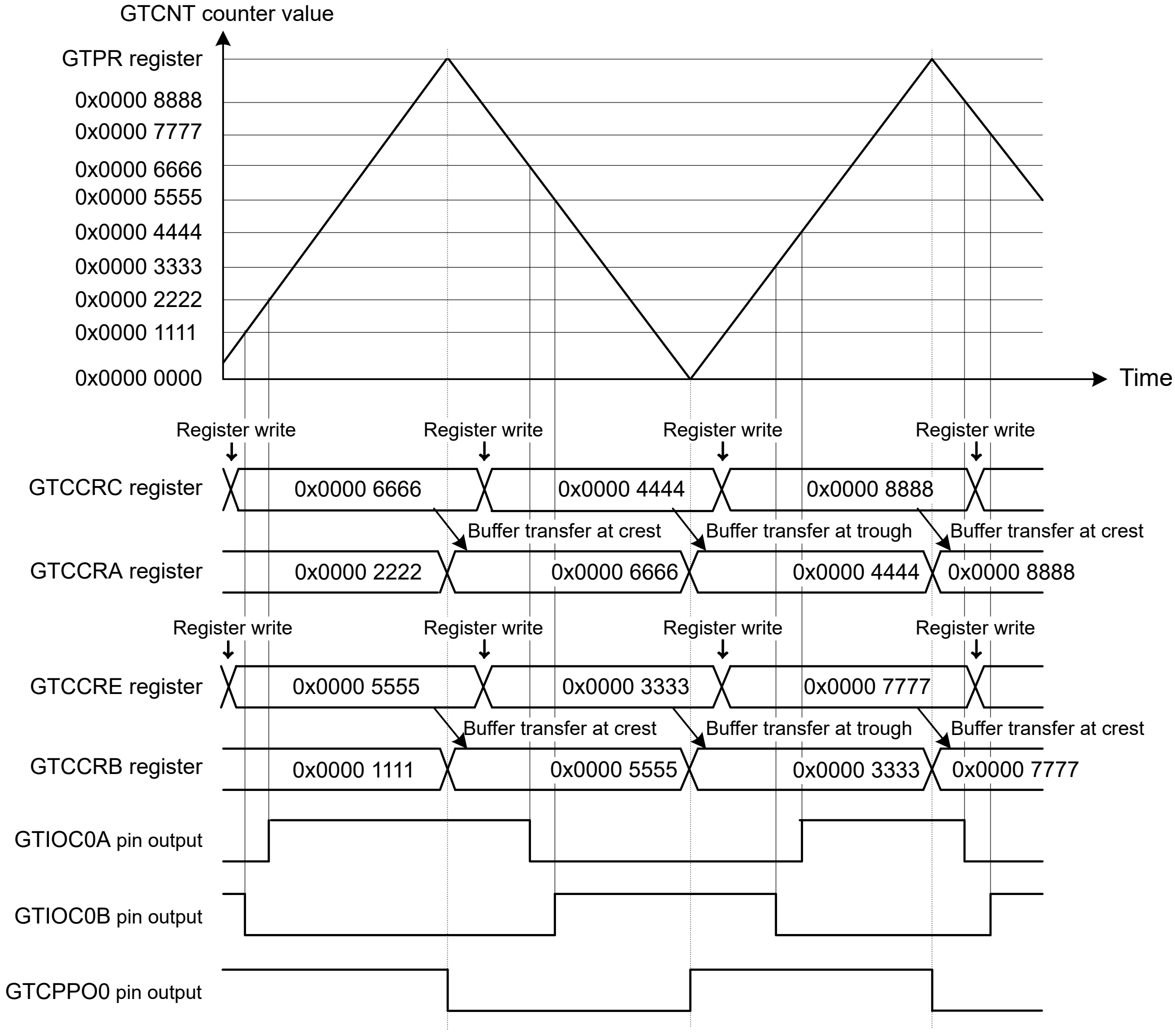
Note 1. When using PWM Delay Generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

22.3.3.5 Triangle-Wave PWM Mode 2 (32-Bit Transfer at Crest and Trough)

Similarly to triangle-wave PWM mode 1, in triangle-wave PWM mode 2 the cycle is set in GTPR. The GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 13) when a GTCCRA or GTCCRB compare match occurs. The buffer transfer is performed at both crests and troughs. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the cycle end according to the GTIOR setting.

By setting GTDTCR, GTDVU, and GTDVD, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 22.42 shows an example of triangle-wave PWM mode 2 operation, and Table 22.28 shows an example for setting triangle-wave PWM mode 2.



Note: The base address of registers are the same.

Figure 22.42 Example of triangle-wave PWM mode 2 operation with buffer operation, low output from the GTIOCnA pin and high output from the GTIOCnB pin at count start, output toggled at GTCCRA/ GTCCRB compare match, output retained at cycle end, and GTIOR.PSYE = 1

Table 22.28 Example for setting triangle-wave PWM mode 2 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 22.42, 101b (triangle-wave PWM mode 2) is set.
2	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
3	Set cycle	Set the cycle in the GTPR register.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 22.42, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
6	Enable GTCPPOn pin output	Set to enable/disable the GTCPPOn pin output with the PSYE bit in the GTIOR register.
7	Enable GTIOCnm pin output[*1]	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set buffer operation	Set buffer operation with the CCRA[1:0] and CCRB[1:0] bits in the GTBER register. In Figure 22.42, CCRA[1:0] = 01b and CCRB[1:0] = 01b.

Table 22.28 Example for setting triangle-wave PWM mode 2 (2 of 2)

No.	Step Name	Description
9	Set compare match value[*1]	Set the GTIOCnA and GTIOCnB pins transitions in the GTCCRA and GTCCRB registers, respectively.
10	Set buffer value	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in half cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in the GTCCRD and GTCCRF registers, respectively.
11	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
12	Set buffer value for each half cycle	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in half cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in GTCCRD and GTCCRF registers, respectively.

Note: n: 0 to 13
m: A, B

Note 1. When using PWM Delay Generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

22.3.3.6 Triangle-Wave PWM Mode 3 (64-Bit Transfer at Trough)

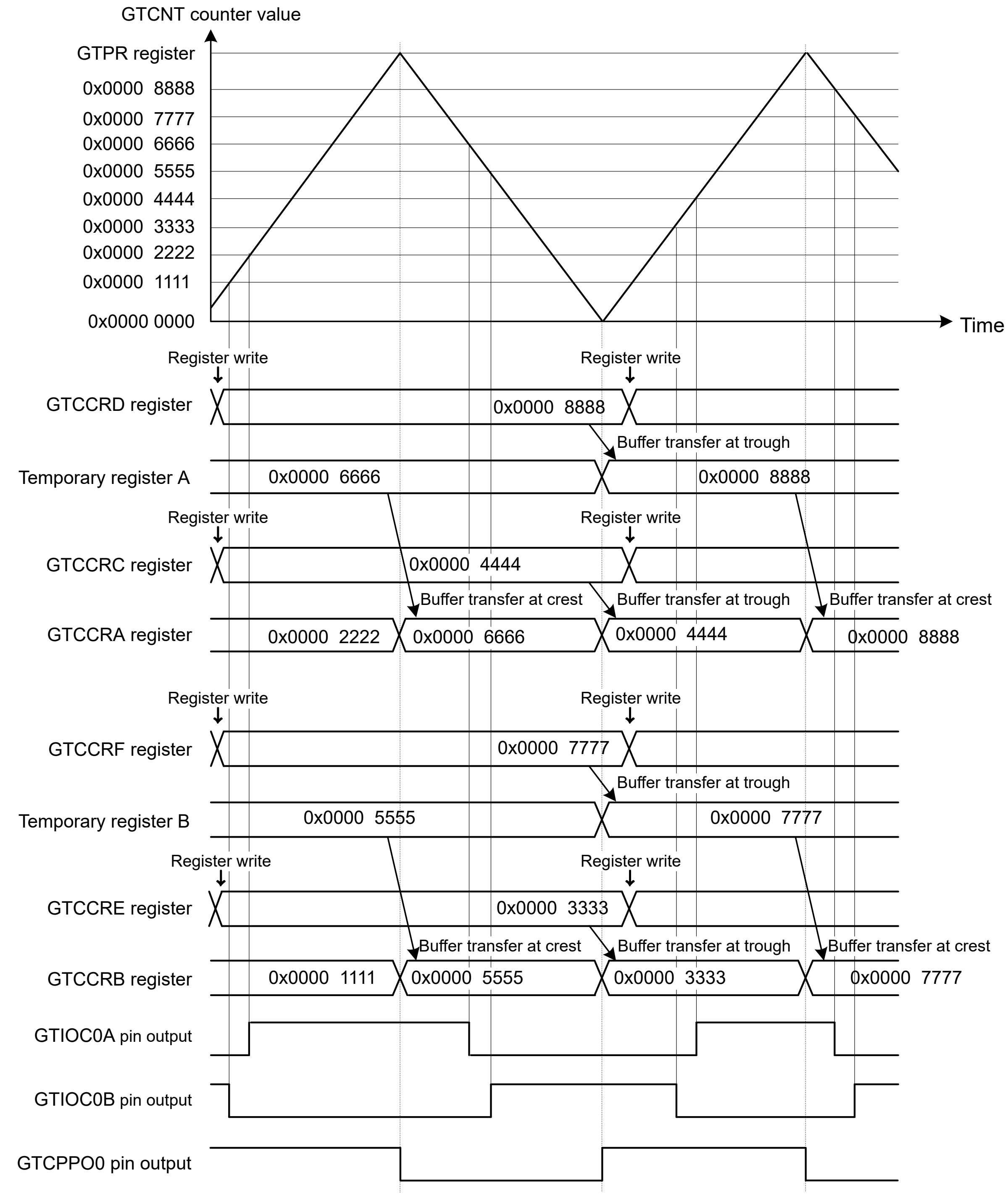
The triangle-wave PWM mode 3 is a mode in which the cycle is set in GTPR. The GTCNT counter performs triangle-wave (full-wave) operation and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 13) at a compare match of GTCCRA or GTCCRB with buffer operation fixed. Buffer operation in triangle-wave PWM mode 3 is different from the usual buffer operation. Buffer transfer is performed from the following:

- GTCCRC to GTCCRA at the trough
- GTCCRE to GTCCRB at the trough
- GTCCRD to temporary register A at the trough
- GTCCRF to temporary register B at the trough
- Temporary register A to GTCCRA at the crest
- Temporary register B to GTCCRB at the crest.

The pin output value can be selected from low output, high output, or toggled output separately for a compare match and for the cycle end according to the GTIOR setting.

By setting GTDTCR, GTDVU, and GTDVD, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 22.43 shows an example of triangle-wave PWM mode 3 operation, and Table 22.29 shows an example for setting triangle-wave PWM mode 3.



Note: The base address of registers are the same.

Figure 22.43 Example of triangle-wave PWM mode 3 operation with low output from the GTIOCnA pin and high output from the GTIOCnB pin at count start, output toggled at GTCCRA/GTCCRB compare match, output retained at cycle end, and GTIOR.PSYE = 1

Table 22.29 Example setting for triangle-wave PWM mode 3 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 22.43, 110b (triangle-wave PWM mode 3) is set.

Table 22.29 Example setting for triangle-wave PWM mode 3 (2 of 2)

No.	Step Name	Description
2	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
3	Set cycle	Set the cycle in the GTPR register.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 22.43, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
6	Enable GTCPPOn pin output	Set to enable/disable the GTCPPOn pin output with the PSYE bit in the GTIOR register.
7	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set compare match value*1	Set the GTIOCnA pin transition immediately after the count start in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.
9	Set forcible buffer transfer*1	Set the GTBER.CCRSWT bit to 1 to transfer buffer register data forcibly.
10	Set buffer value	Set the GTIOCnA pin transition in 1 cycle after the current cycle in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.
11	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
12	Set buffer value for each cycle	Set the GTIOCnA pin transition in 1 cycle after the current cycle in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.

Note: n: 0 to 13
m: A, B

Note 1. When using PWM Delay Generation circuit, change the order of Enable GTIOCnm pin output and Set compare match value + Set forcible buffer transfer.

22.3.3.7 Complementary PWM mode 1, 2, 3

In complementary PWM mode, a three-phase PWM waveform with dead time that ensures the linearity in the vicinity of duty 0% and 100% can be output using the GTCNT counter of consecutive three channels. There are four modes depending on differences in buffer operation: (1) complementary PWM mode 1 (transfer at crests), (2) complementary PWM mode 2 (transfer at troughs), (3) complementary PWM mode 3 (transfer at crests and troughs), and (4) complementary PWM mode 4 (immediate transfer).

Figure 22.44 shows the block diagram in complementary PWM modes 1 to 3.

Among consecutive three channels, the lowest channel is referred to as master channel, and the adjacent upper two channels are referred to as slave channel 1 (lower) and slave channel 2 (upper).

The GTCNT counter of each channel performs individual count operation under the cycle operation by the master channel. In each channel, compare match with the GTCCRA register is performed selecting one of the three GTCNT counters in each operation section, and a positive-phase waveform and a negative-phase waveform are output from the GTIOCn+iA pin (i = 0, 1, 2) and the GTIOCn+iB pin respectively with a non-overlapping section of the dead time value set in the GTDVU register of the master channel.

The GTCCRA register performs buffer operation by the GTCCRC register, temporary register A, and GTCCRD register. In complementary PMW mode 3, setting the GTBER2.CP3DB bit to 1 also enables buffer operation of the GTCCRA register by the GTCCRE register, temporary register B, and GTCCRF register, allowing double buffer operation.

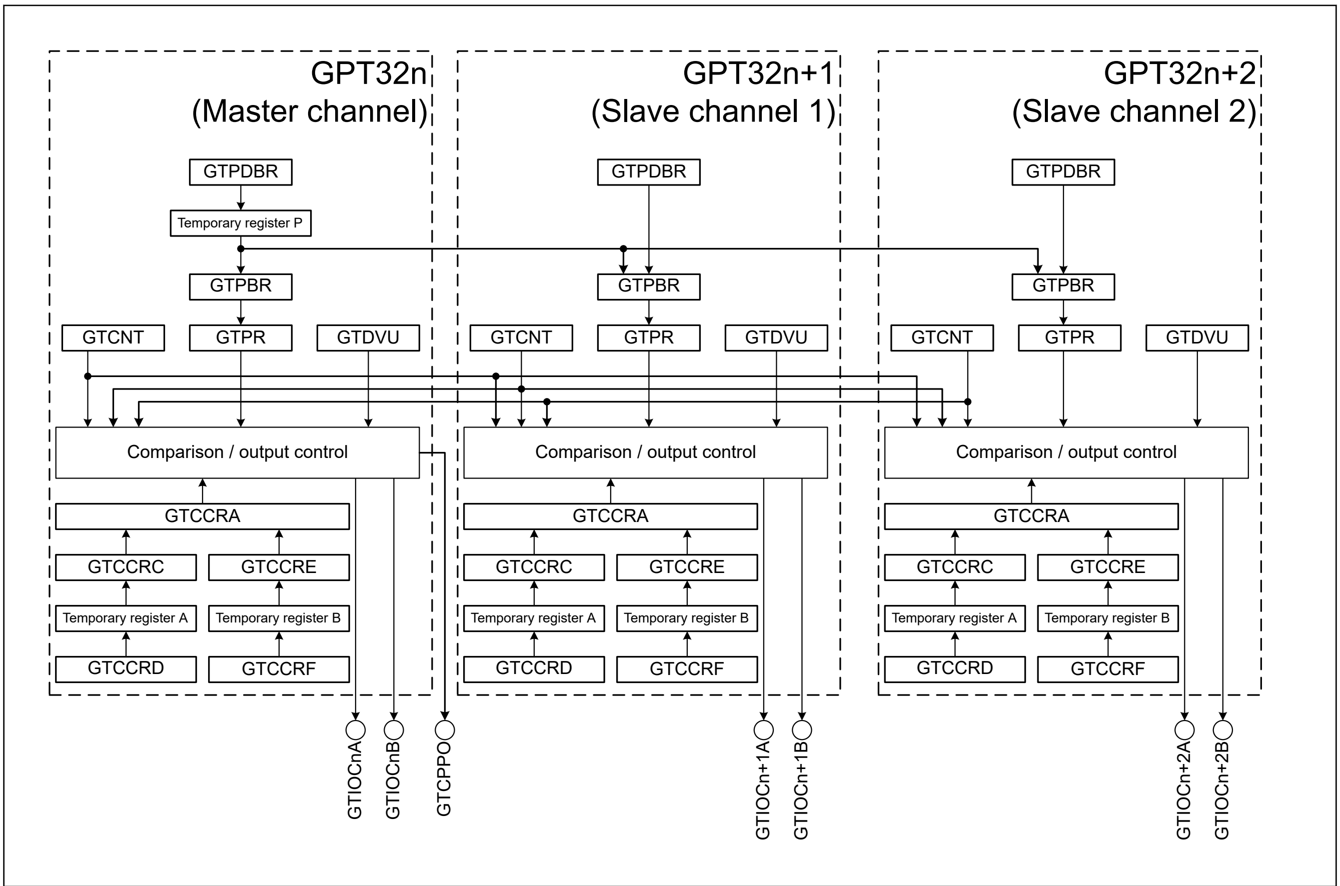


Figure 22.44 Block Diagram in Complementary PWM Mode 1, 2, 3 (n = 4, 7)

The GPT32n.GTCNT counter performs count operation for triangle waves using the GPT32n.GTPR register as a cycle register. A section where the GPT32n.GTCNT counter value is not larger than the dead time value is referred to as trough section.

The GPT32n+1.GTCNT counter performs count operation with the value (GPT32n.GTCNT counter value + dead time value set in the GPT32n.GTDVU register). A section where the GPT32n+1.GTCNT counter value is larger than the GPT32n.GTPR register value is referred to as crest section.

Crest section and trough section are classified into up-counting crest section, down-counting crest section, up-counting trough section, and down-counting trough section according to counting direction. A section between trough section and crest section is referred to as up-counting middle section or down-counting middle section according to counting direction. A section equivalent to the up-counting trough section after starting count operation is referred to as initial output section where operation differs partially from other up-counting trough sections.

The GPT32n+2.GTCNT counter functions as a counter to ensure the linearity in the vicinity of duty 0% and 100%. In a crest section, this counter performs count operation for a triangle wave (up-counting after down-counting) with the value (GPT32n.GTPR register value + dead time value) as an initial value and the GPT32n.GTPR register value as a trough. This counter is cleared to 0 at the end of the crest section, and then stops counting until the next trough section. In a trough section, this counter performs count operation for a triangle wave with an initial value of 0 and dead time value as a crest. This counter becomes the value (GPT32n.GTPR register value + dead time value) at the end of the trough section, and then stops counting until the next crest section. In the initial output section, however, this counter counts up with an initial value of 0 until the dead time value, and then becomes the value (GPT32n.GTPR register value + dead time value).

If the counter stops and then restarts in complementary PWM mode, the counter of each channel returns to the initial value after starting count operation, and then starts counting from the initial output section.

Table 22.30 and Table 22.31 shows count operation (counting direction/counting range) in each section. In these tables, registers with no channel identification indicate that the same value is stored in them of the master channel, slave channel 1, and slave channel 2.

Table 22.30 Count Operation in Complementary PWM Mode

Counter	Initial Value	Initial Output Section (After Start)	Up-Counting Middle Section	Up-Counting Crest Section	Down-Counting Crest Section
GPT32n.GTCNT	0	Up-counting 0 to GTDVU	Up-counting GTDVU+1 to GTPR-GTDVU	Up-counting GTPR-GTDVU+1 to GTPR	Down-counting GTPR-1 to GTPR-GTDVU
GPT32n+1.GTCNT	GTDVU	Up-counting GTDVU to GTDVU×2	Up-counting GTDVU×2+1 to GTPR	Up-counting GTPR+1 to GTPR+GTDVU	Down-counting GTPR+GTDVU-1 to GTPR
GPT32n+2.GTCNT	0	Up-counting 0 to GTDVU	Stop GTPR+GTDVU	Down-counting GTPR+GTDVU-1 to GTPR	Up-counting GTPR+1 to GTPR+GTDVU

Table 22.31 Count Operation in Complementary PWM Mode

Counter	Down-Counting Middle Section	Down-Counting Trough Section	Up-Counting Trough Section
GPT32n.GTCNT	Down-counting GTPR-GTDVU-1 to GTDVU	Down-counting GTDVU-1 to 0	Up-counting 1 to GTDVU
GPT32n+1.GTCNT	Down-counting GTPR-1 to GTDVU×2	Down-counting GTDVU×2-1 to GTDVU	Up-counting GTDVU+1 to GTDVU×2
GPT32n+2.GTCNT	Stop 0	Up-counting 1 to GTDVU	Down-counting GTDVU-1 to 0

In complementary PWM mode, the GTCCRA register performs unusual buffer operation.

Data transfers from the GTCCRD register to the temporary register A and from the GTCCRF register to the temporary register B are performed at the same time in three channels by writing a value to the GTCCRD register of the GPT32n+2 channel.

Data transfers from the temporary register A and temporary register B to the GTCCRC and GTCCRE registers vary depending on the transfer timing to the temporary register A and temporary register B. Data transfers from the GTCCRC and GTCCRE registers to the GTCCRA register are performed according to each complementary PWM mode name (crest transfer, trough transfer, and crest/trough transfer).

Buffer operation of the GTPR register in complementary PWM mode is described in section 22.3.2.1. GTPR Register Buffer Operation. Do not perform buffer operation for the GTDVU register in complementary PWM mode.

Table 22.32 shows buffer transfer timing during single buffer operation in complementary PWM modes 1 to 3. Table 22.33 shows buffer transfer timing during double buffer operation in complementary PWM mode 3.

Table 22.32 Single Buffer Transfer Timing in Complementary PWM Mode 1, 2, 3

Buffer Transfer	Complementary PWM Mode 1	Complementary PWM Mode 2	Complementary PWM Mode 3 (Single Buffer)
GTCCRD ↓ Temporary register A	After one GTCLK cycle from GTCCRD register write of slave channel 2 (GPT32n+2)	After one GTCLK cycle from GTCCRD register write of slave channel 2 (GPT32n+2)	After one GTCLK cycle from GTCCRD register write of slave channel 2 (GPT32n+2)
Temporary register A ↓ GTCCRC	(1) When data is transferred to temporary register A in up-counting middle section After one GTCLK cycle from transfer to temporary register A (2) When data is transferred to temporary register A in a section other than up-counting middle section At the end of trough section	(1) When data is transferred to temporary register A in down-counting middle section After one GTCLK cycle from transfer to temporary register A (2) When data is transferred to temporary register A in a section other than down-counting middle section At the end of crest section	(1) When data is transferred to temporary register A in middle section After one GTCLK cycle from transfer to temporary register A (2) When data is transferred to temporary register A in a section other than middle section At the end of crest and trough sections
GTCCRC ↓ GTCCRA	At the end of crest section Counter clear in up-counting middle section and crest section	At the end of trough section (excluding initial output section) Counter clear in down-counting middle section and trough section	At the end of crest section At the end of trough section (excluding initial output section) Counter clear

Table 22.33 Double Buffer Transfer Timing in Complementary PWM Mode 3

Transfer from GTCCRD to GTCCRA		Transfer from GTCCRF to GTCCRA	
Buffer Transfer	**Transfer Timing**	**Buffer Transfer**	**Transfer Timing**
GTCCRD ↓ Temporary register A	After one GTCLK cycle from GTCCRD register write of slave channel 2 (GPT32n+2)	GTCCRF ↓ Temporary register B	After one GTCLK cycle from GTCCRD register write of slave channel 2 (GPT32n+2)
Temporary register A ↓ GTCCRC	(1) When data is transferred to temporary register A in middle section After one GTCLK cycle from transfer to temporary register A (2) When data is transferred to temporary register A in a section other than middle section At the end of crest and trough sections	Temporary register B ↓ GTCCRE	(1) When data is transferred to temporary register B in middle section After one GTCLK cycle from transfer to temporary register B (2) When data is transferred to temporary register B in a section other than middle section At the end of crest and trough sections
GTCCRC ↓ GTCCRA	At the end of crest section Counter clear	GTCCRE ↓ GTCCRA	At the end of trough section (excluding initial output section)

An output level change in the positive-phase waveform from the GTIOCn+iA pin (i = 0, 1, 2) and the negative-phase waveform from the GTIOCn+iB pin occurs at a compare match in combination of counters and registers determined for each operation section. In middle sections, the positive-phase waveform output level changes at a compare match of the GPT32n.GTCNT counter and the GTCCRA register, and the negative-phase waveform output level changes at a compare match between the GPT32n+1.GTCNT counter and the GTCCRA register. In crest and trough sections, compare match operation is performed using the GPT32n+2.GTCNT counter, GTCCRC register, and GTCCRE register to ensure the linearity in the vicinity of duty 0% and 100%.

In the case that the compare match value equals to or larger than the GPT32n.GTPR register value, the duty becomes 0% (positive-phase waveform OFF, negative-phase waveform ON). In the case that the compare match value is 0, the duty becomes 100% (positive-phase waveform ON, negative-phase waveform OFF).

Table 22.34 lists combinations of counters and registers used for compare match operation to generate a positive-phase waveform and a negative-phase waveform in each operation section.

Table 22.34 Combinations of Counters and Registers for Compare Match Operation in Complementary PWM Mode

	Up-Counting Middle Section	Up-Counting Crest Section	Down-Counting Crest Section	Down-Counting Middle Section	Down-Counting Trough Section	Up-Counting Trough Section
Negative-phase OFF	GPT32n+1.GTCNT	GPT32n+1.GTCNT	—	GPT32n+2.GTCNT*1	GPT32n+2.GTCNT	GPT32n+1.GTCNT
	GTCCRA	GTCCRA	—	GTCCRC (GTCCRE for double buffer operation)	GTCCRC (GTCCRE for double buffer operation)	GTCCRC (GTCCRE for double buffer operation)
Positive-phase ON	GPT32n.GTCNT	GPT32n.GTCNT	GPT32n+2.GTCNT	—	GPT32n.GTCNT*1	GPT32n.GTCNT
	GTCCRA	GTCCRA	GTCCRA	—	GTCCRC (GTCCRE for double buffer operation)	GTCCRC (GTCCRE for double buffer operation)
Positive-phase OFF	GPT32n+2.GTCNT*1	GPT32n+2.GTCNT	GPT32n.GTCNT	GPT32n.GTCNT	GPT32n.GTCNT	—
	GTCCRC	GTCCRC	GTCCRC	GTCCRA	GTCCRA	—
Negative-phase ON	—	GPT32n+1.GTCNT*1	GPT32n+1.GTCNT	GPT32n+1.GTCNT	GPT32n+1.GTCNT	GPT32n+2.GTCNT
	—	GTCCRC	GTCCRC	GTCCRA	GTCCRA	GTCCRA

Note 1. Compare match is performed only at the time of final count in the target section, but is not performed at count values other than the final count.

In the case of normal complementary PWM mode waveform, a PWM waveform change occurs in the order of negative-phase OFF → positive-phase ON → positive-phase OFF → negative-phase ON. However, this order may vary depending on operation section and register values. In this case, OFF takes precedence in trough sections and ON takes precedence in crest sections (for negative-phase waveforms), and ON takes precedence in trough sections and OFF takes precedence in crest sections (for positive-phase waveforms). A lower-priority compare match that occurs at the same time or after a higher-priority compare match is ignored.

In the initial output section, the initial output set in the GTIOR register is retained. In the case that the GTCCRA register value is larger than the GTDVU register value at the end of the initial output section, negative phase is enabled. In the case that the GTCCRA register value is not larger than the GTDVU register value, positive phase is enabled.

As operation examples of normal complementary PWM mode waveform where compare match operation occurs in the middle section, Figure 22.45 and Figure 22.46 show complementary PWM mode 1, Figure 22.47 and Figure 22.48 show complementary PWM mode 2, Figure 22.49 and Figure 22.50 show single buffer complementary PWM mode 3, and Figure 22.51 and Figure 22.52 show double buffer complementary PWM mode 3.

Figure 22.53 to Figure 22.64 show complementary PWM mode waveforms where compare match operation occurs in crest sections and trough sections and differences due to compare match occurrence order.

Figure 22.65 and Figure 22.66 show examples of initial output operation according to the GTCCRA register value.

Table 22.35 shows an example for setting complementary PWM modes 1 to 3.

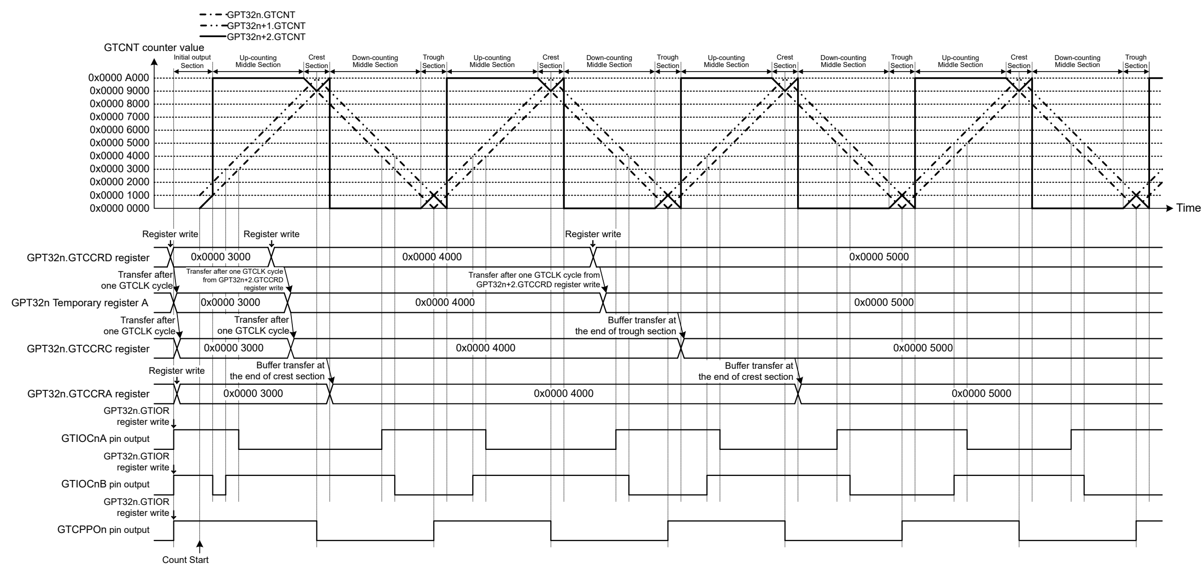


Figure 22.45 Example of Complementary PWM Mode 1 Operation (GTIOCnA pin = High / GTIOCnB pin = High as initial output, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000 and updating GTCCRD register in middle section) (n = 4, 7)

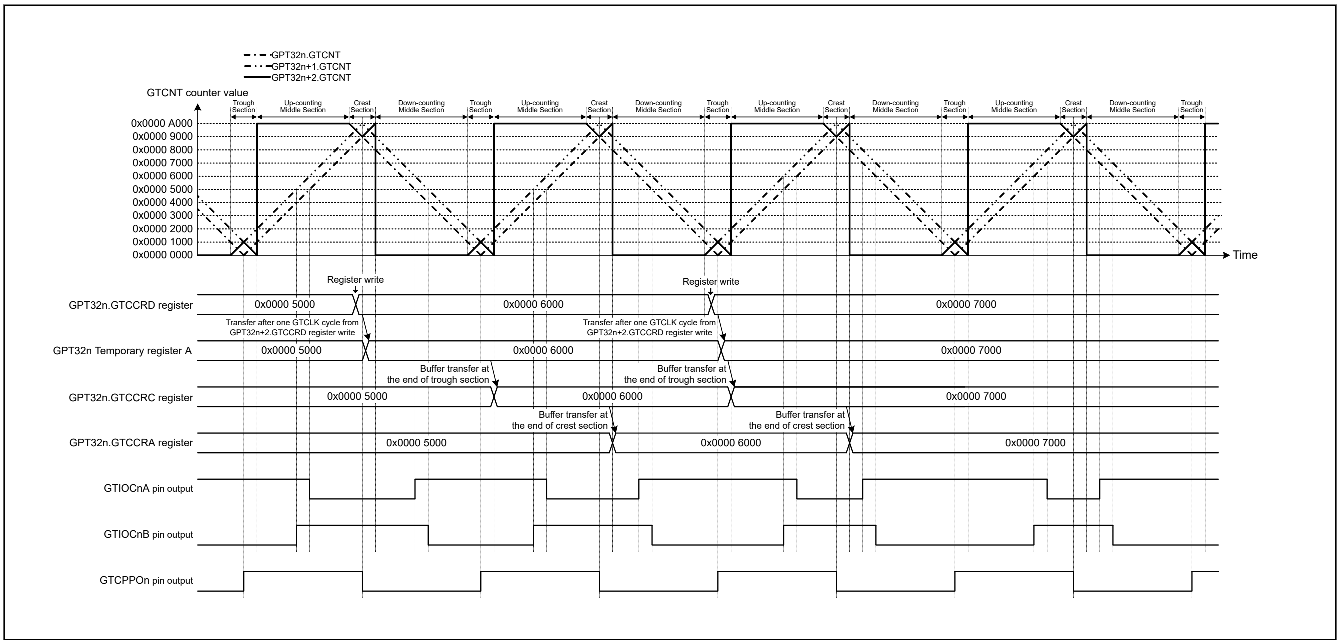


Figure 22.46 Example of Complementary PWM Mode 1 Operation (GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000 and updating GTCCRD register in crest and trough sections) (n = 4, 7)

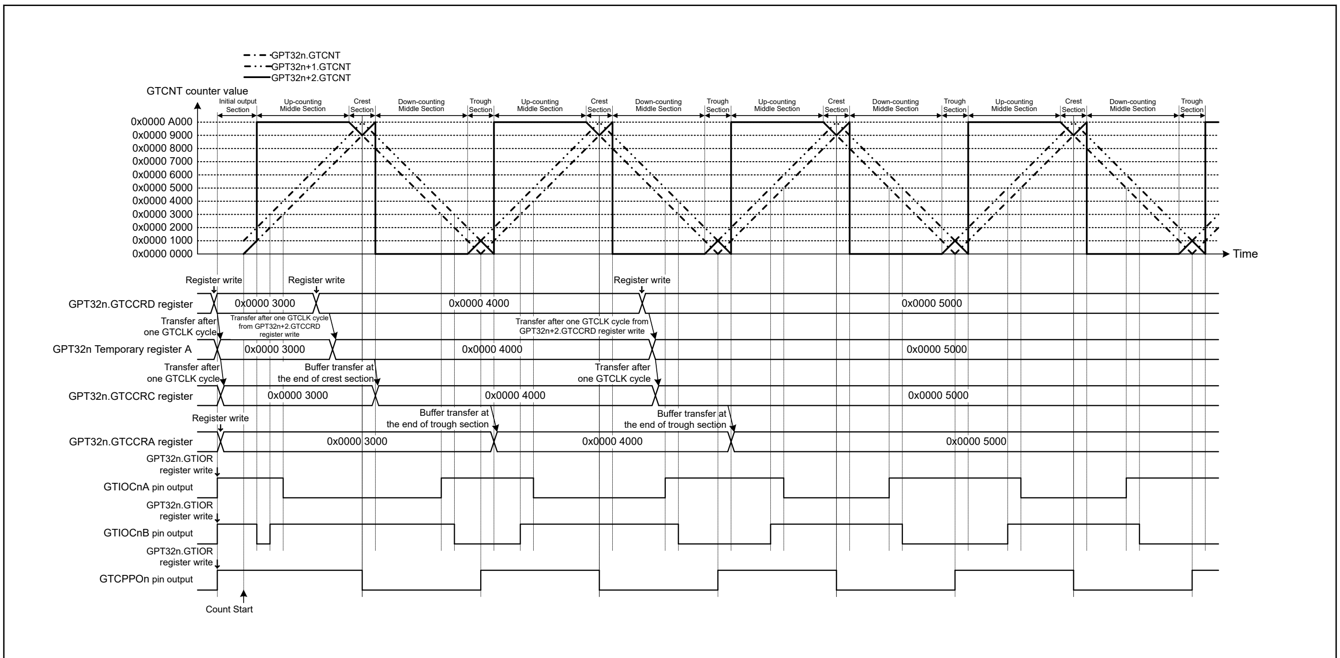


Figure 22.47 Example of Complementary PWM Mode 2 Operation (GTIOCnA pin = High / GTIOCnB pin = High as initial output, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000 and updating GTCCRD register in middle section) (n = 4, 7)

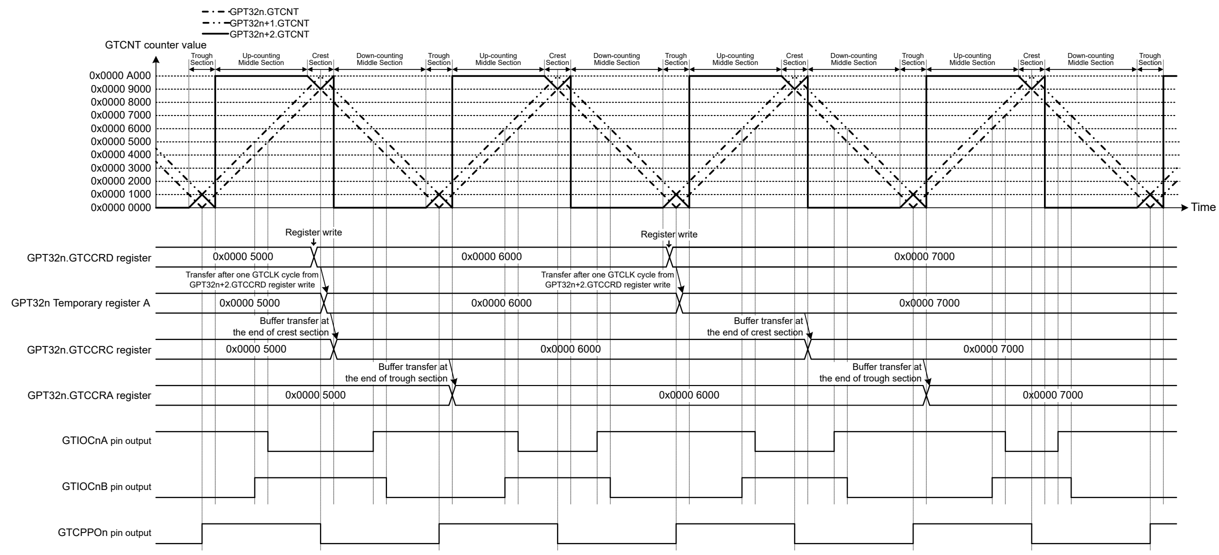


Figure 22.48 Example of Complementary PWM Mode 2 Operation (GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000 and updating GTCCRD register in crest and trough sections) (n = 4, 7)

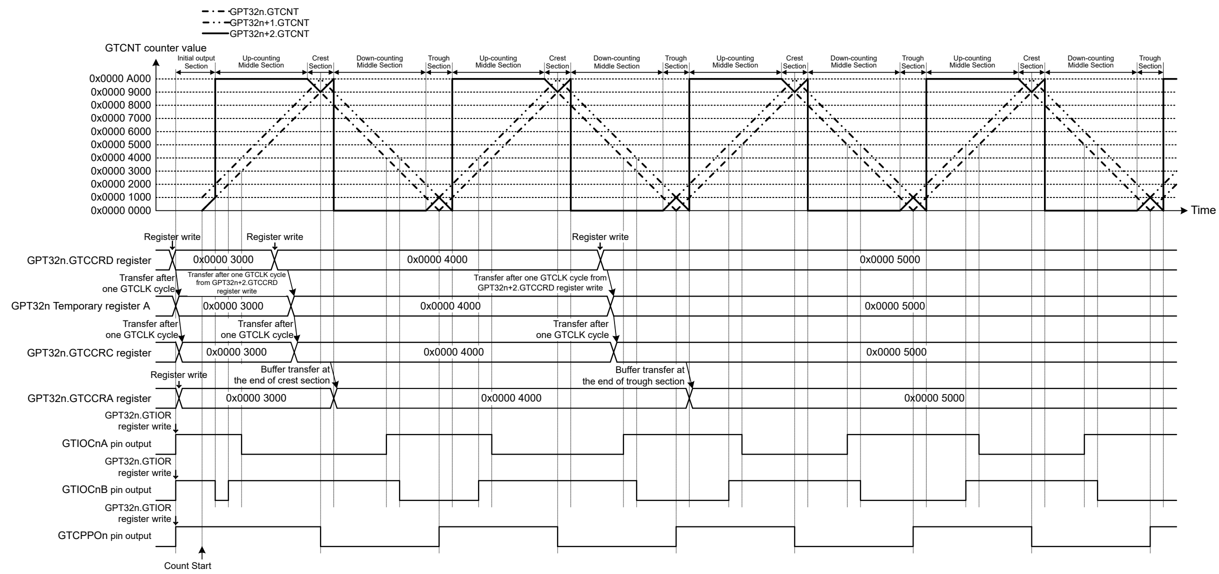


Figure 22.49 Example of Complementary PWM Mode 3 Operation (Single buffer operation, GTIOCnA pin = High / GTIOCnB pin = High as initial output, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000 and updating GTCCRD register in middle section) (n = 4, 7)

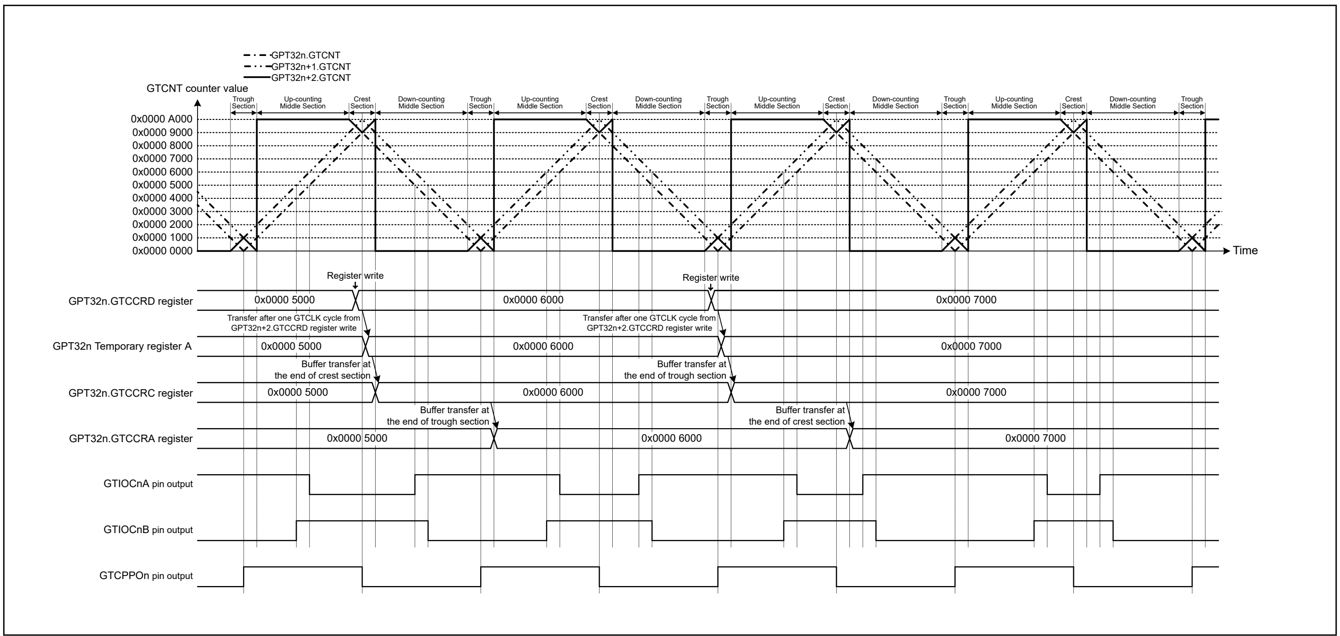


Figure 22.50 Example of Complementary PWM Mode 3 Operation (Single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000 and updating GTCCRD register in crest and trough sections) (n = 4, 7)

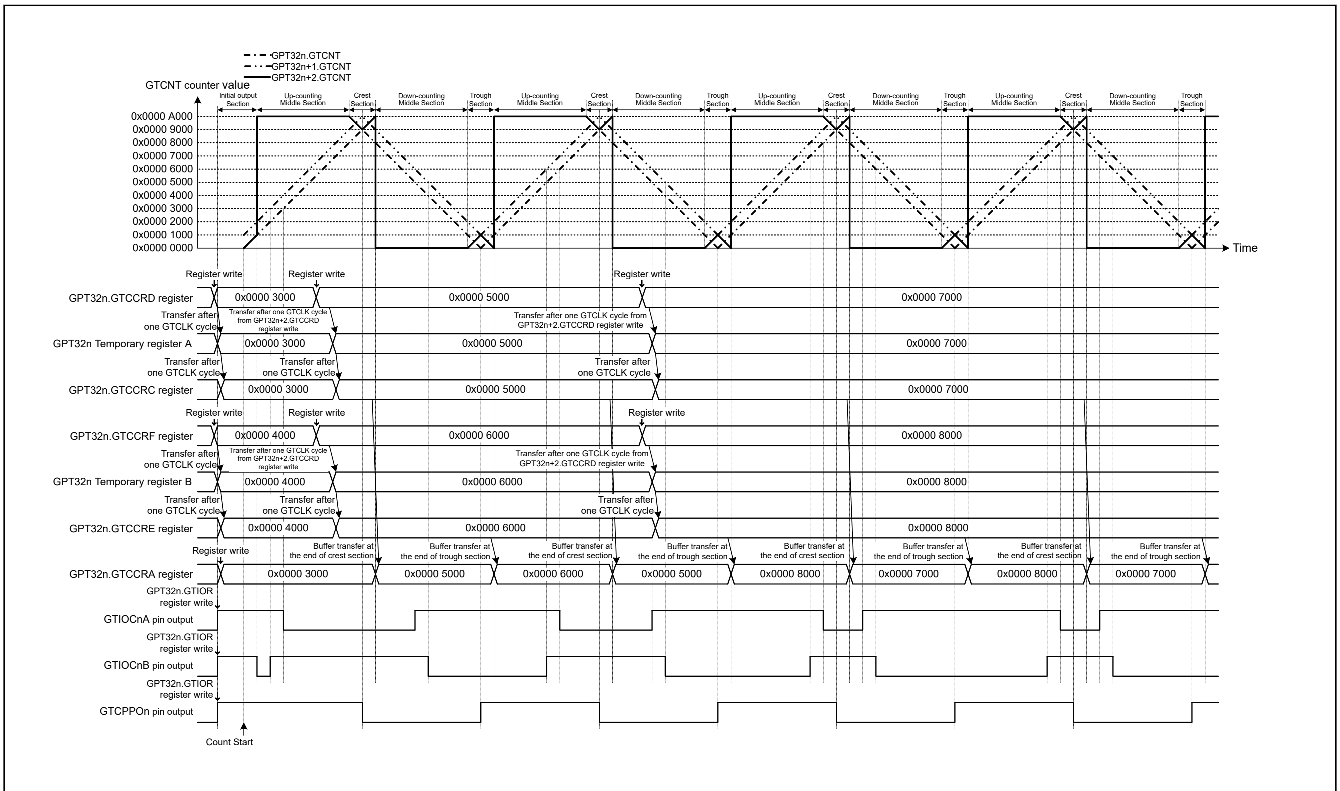


Figure 22.51 Example of Complementary PWM Mode 3 Operation (Double buffer operation, GTIOCnA pin = High / GTIOCnB pin = High as initial output, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000 and updating GTCCRD register in middle section) (n = 4, 7)

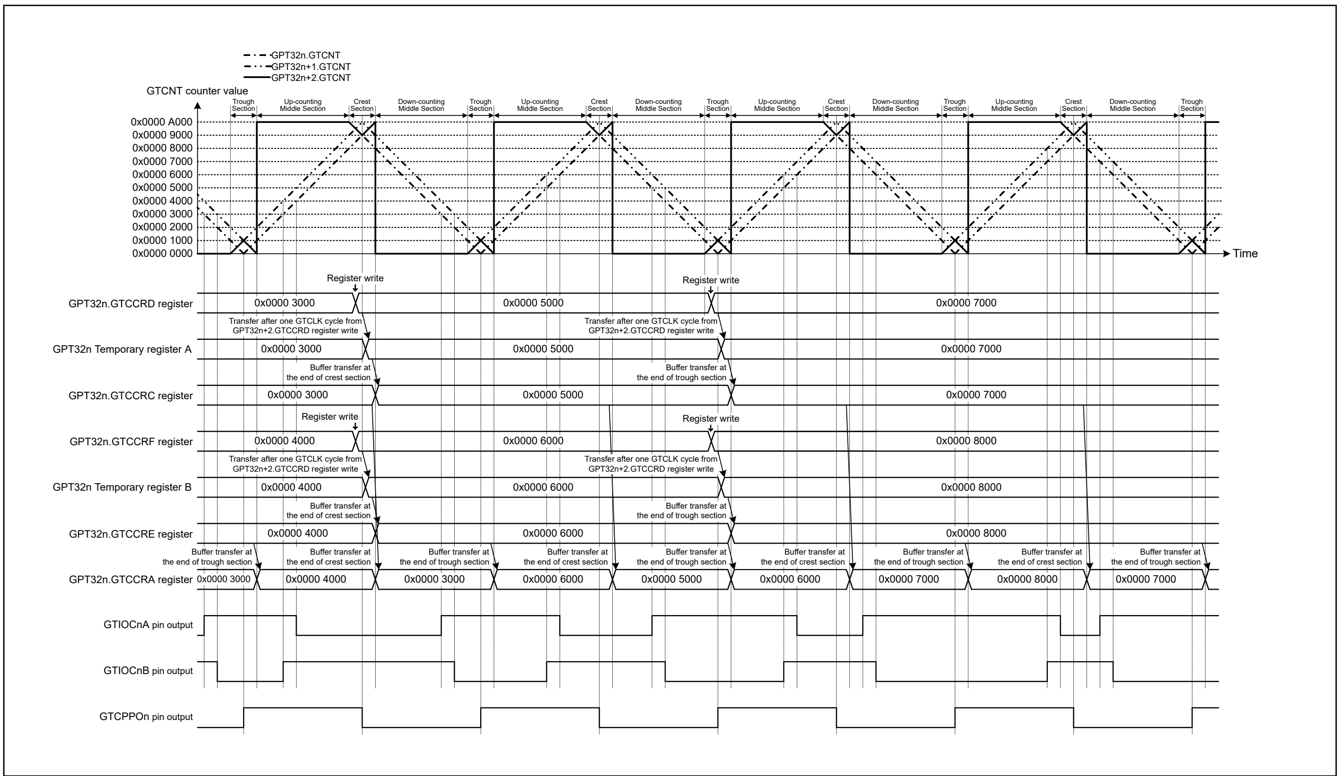


Figure 22.52 Example of Complementary PWM Mode 3 Operation (Double buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000 and updating GTCCRD register in crest and trough sections) (n = 4, 7)

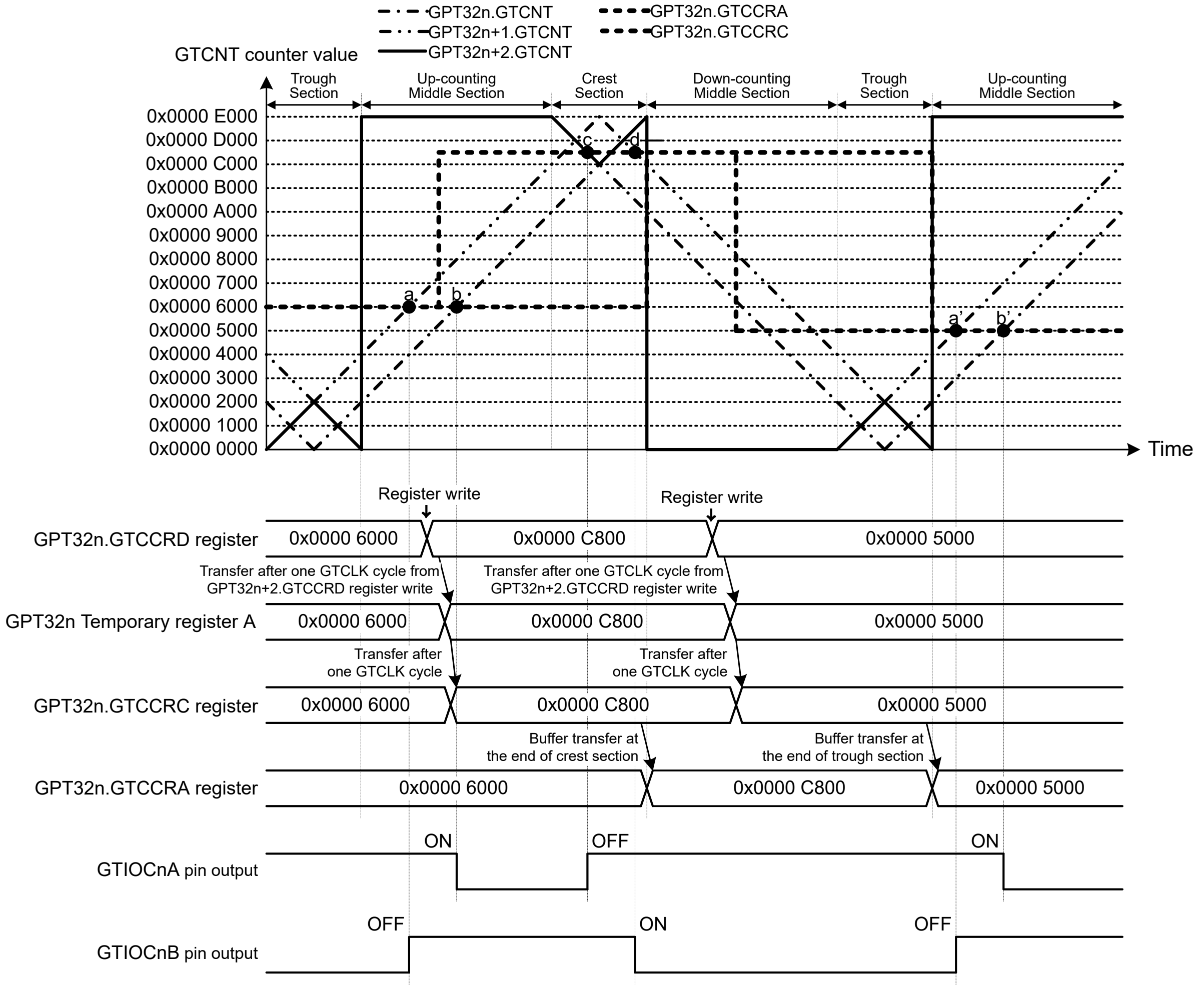


Figure 22.53 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: a → b → c → d) (n = 4, 7)

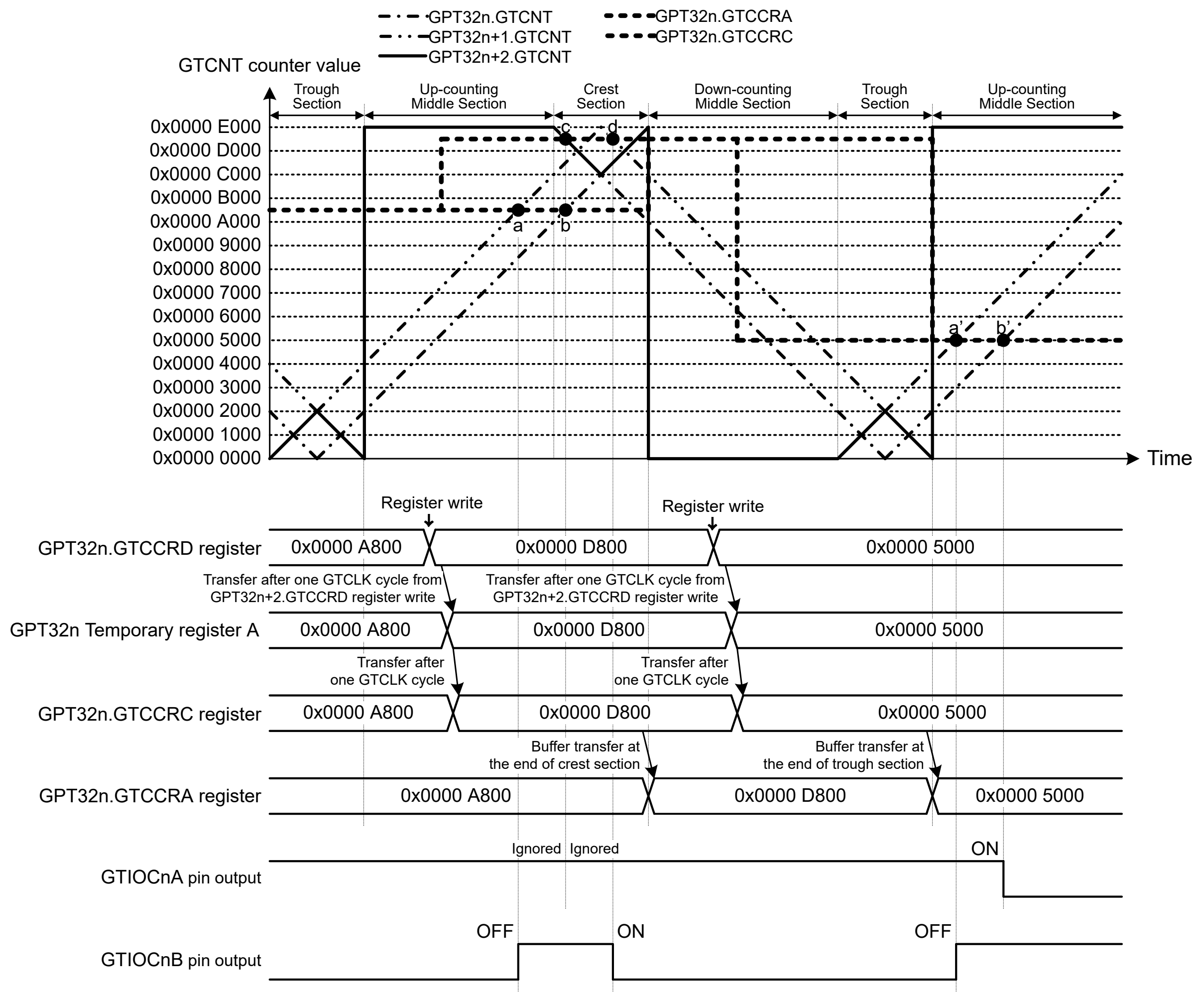


Figure 22.54 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: a → (b, c) → d) (n = 4, 7)

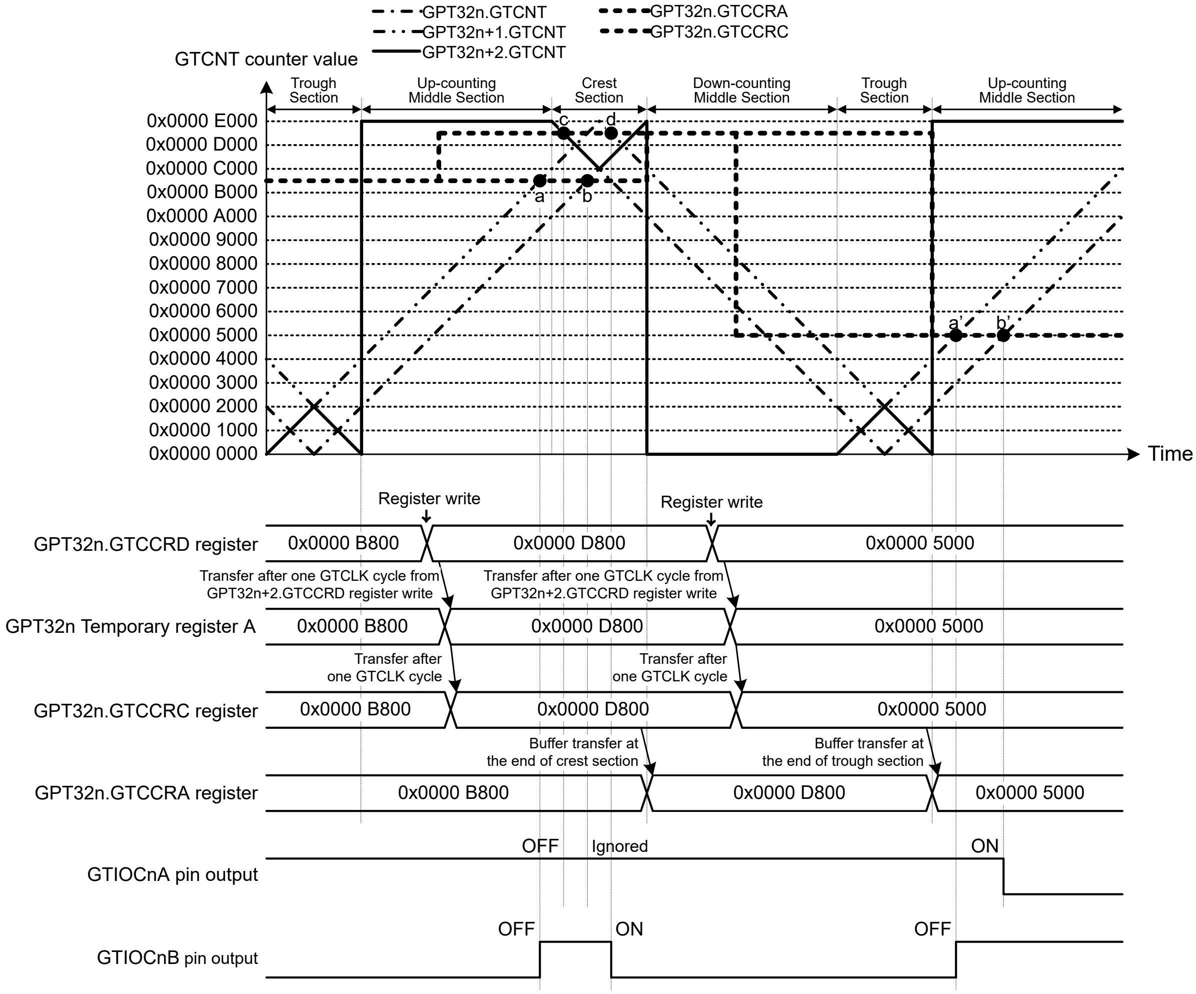


Figure 22.55 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: a → c → b → d) (n = 4, 7)

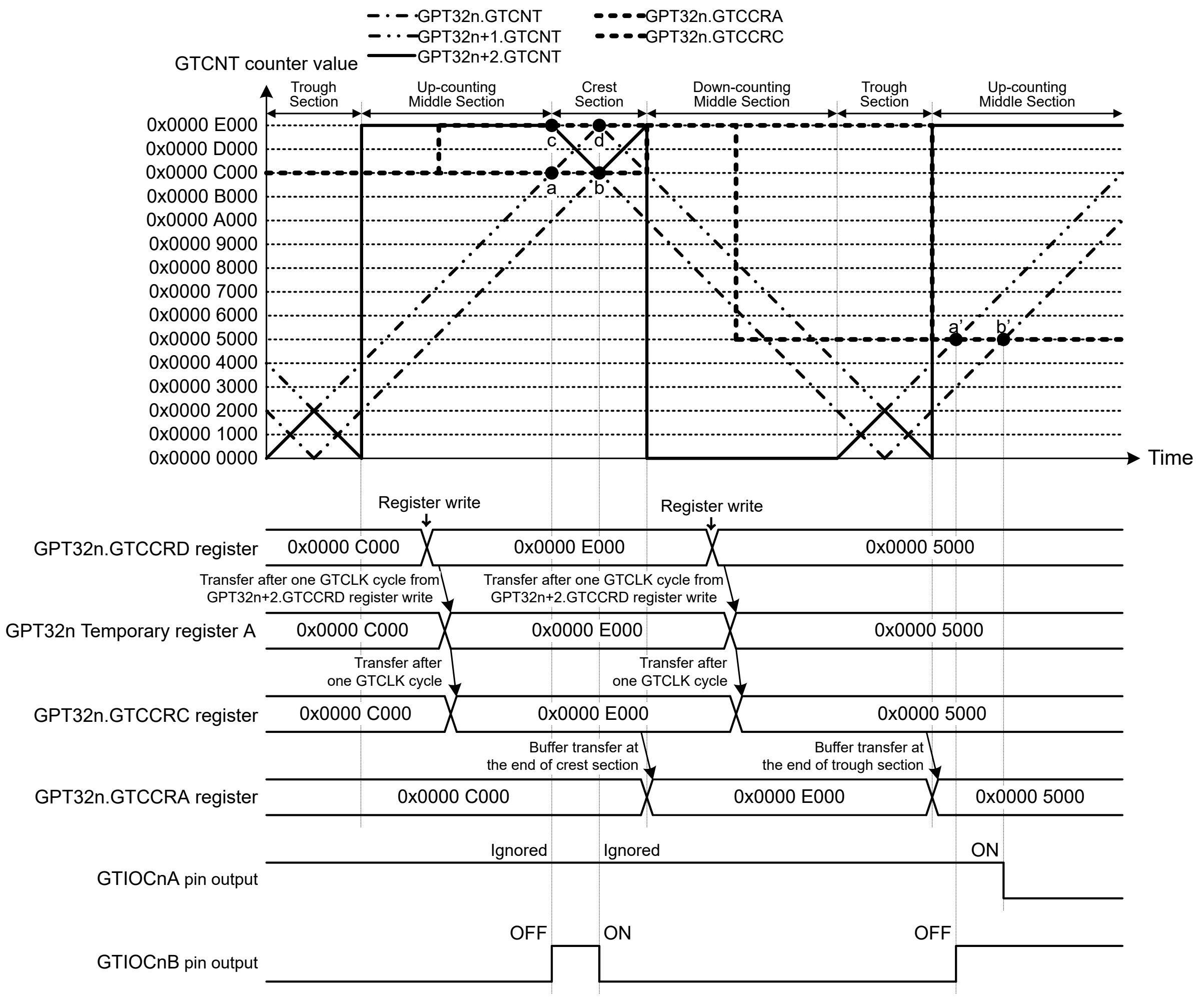


Figure 22.56 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: (a, c) → (b, d)) (n = 4, 7)

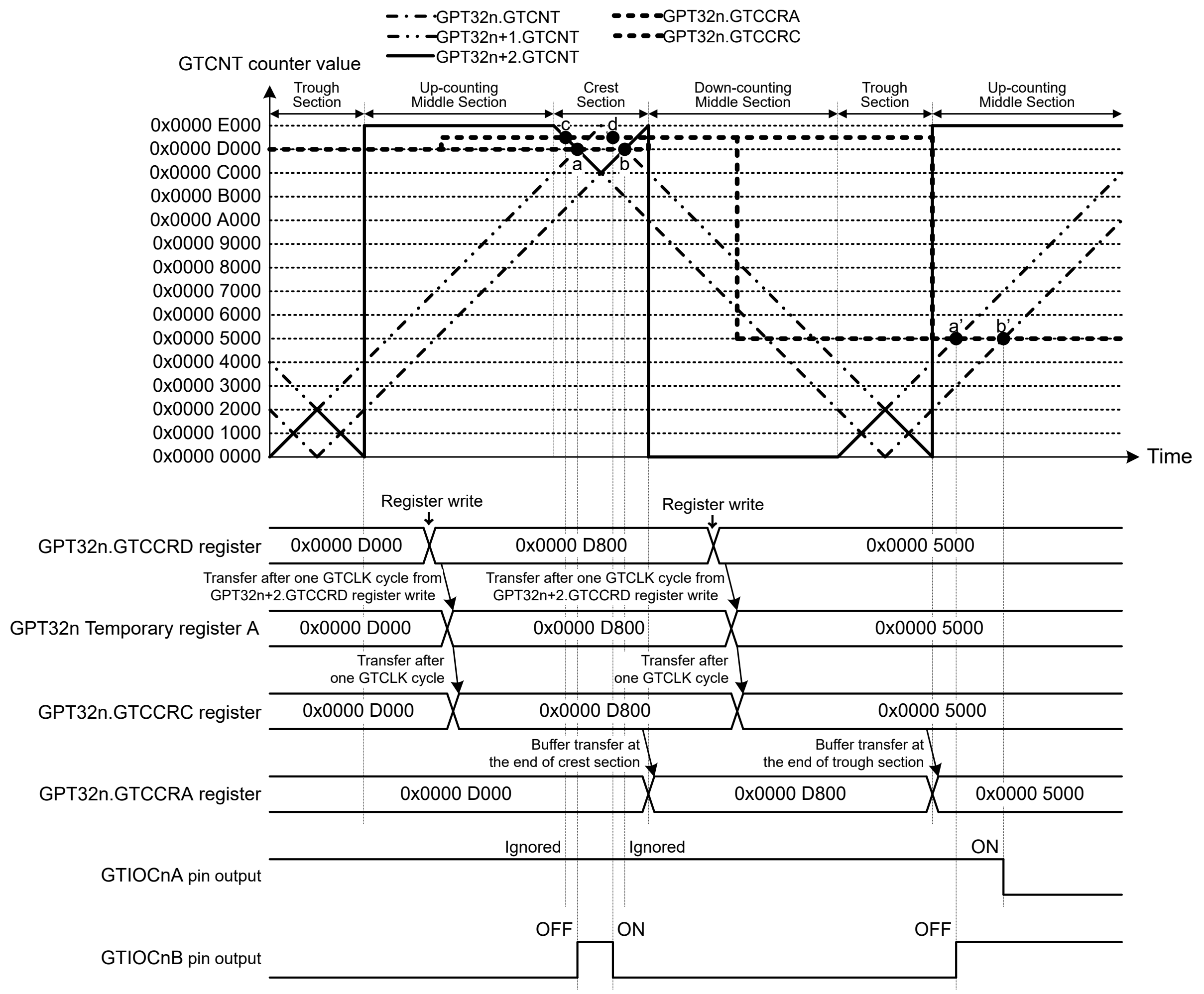


Figure 22.57 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: c → a → d → b) (n = 4, 7)

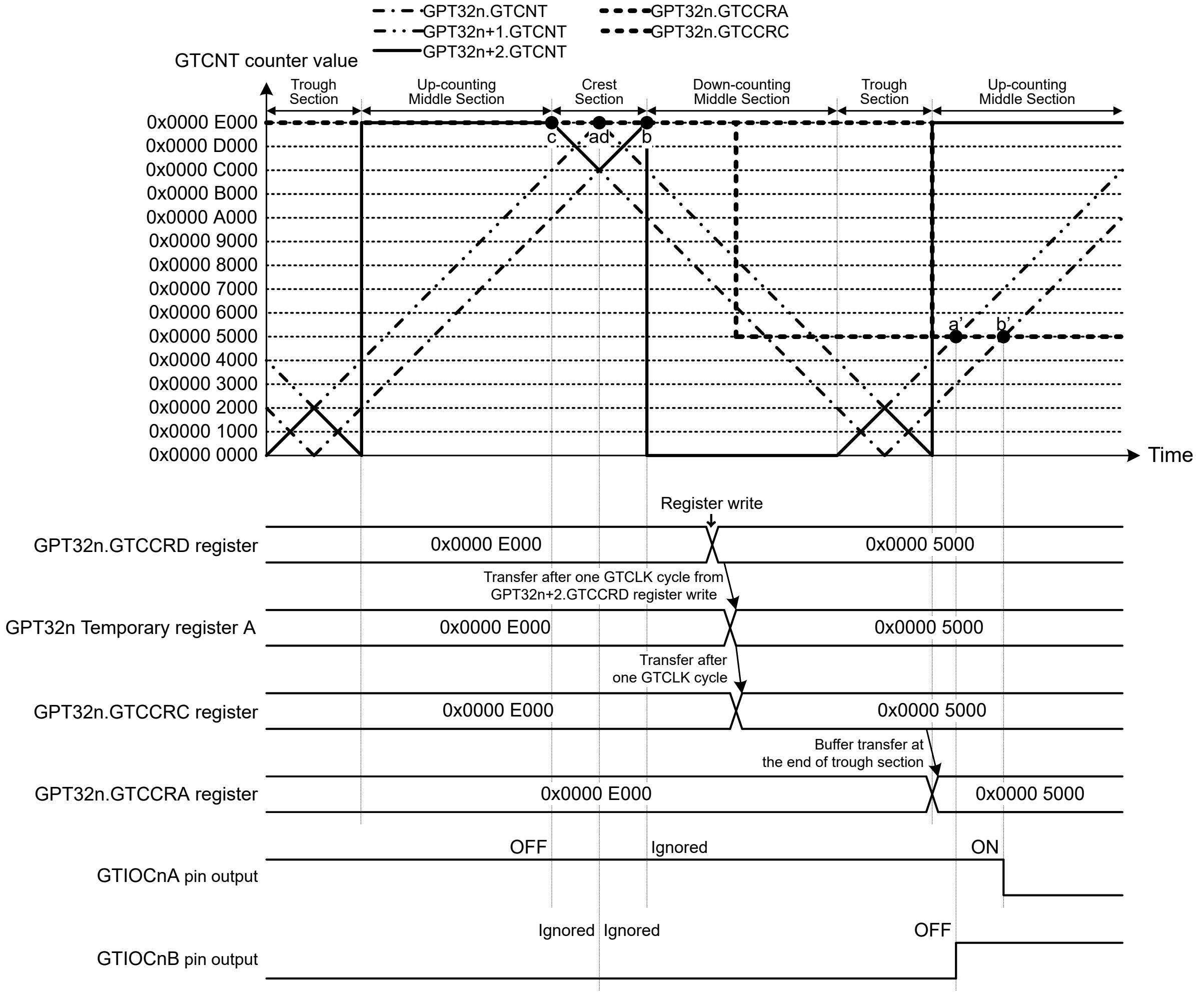


Figure 22.58 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: c → (a, d) → b) (n = 4, 7)

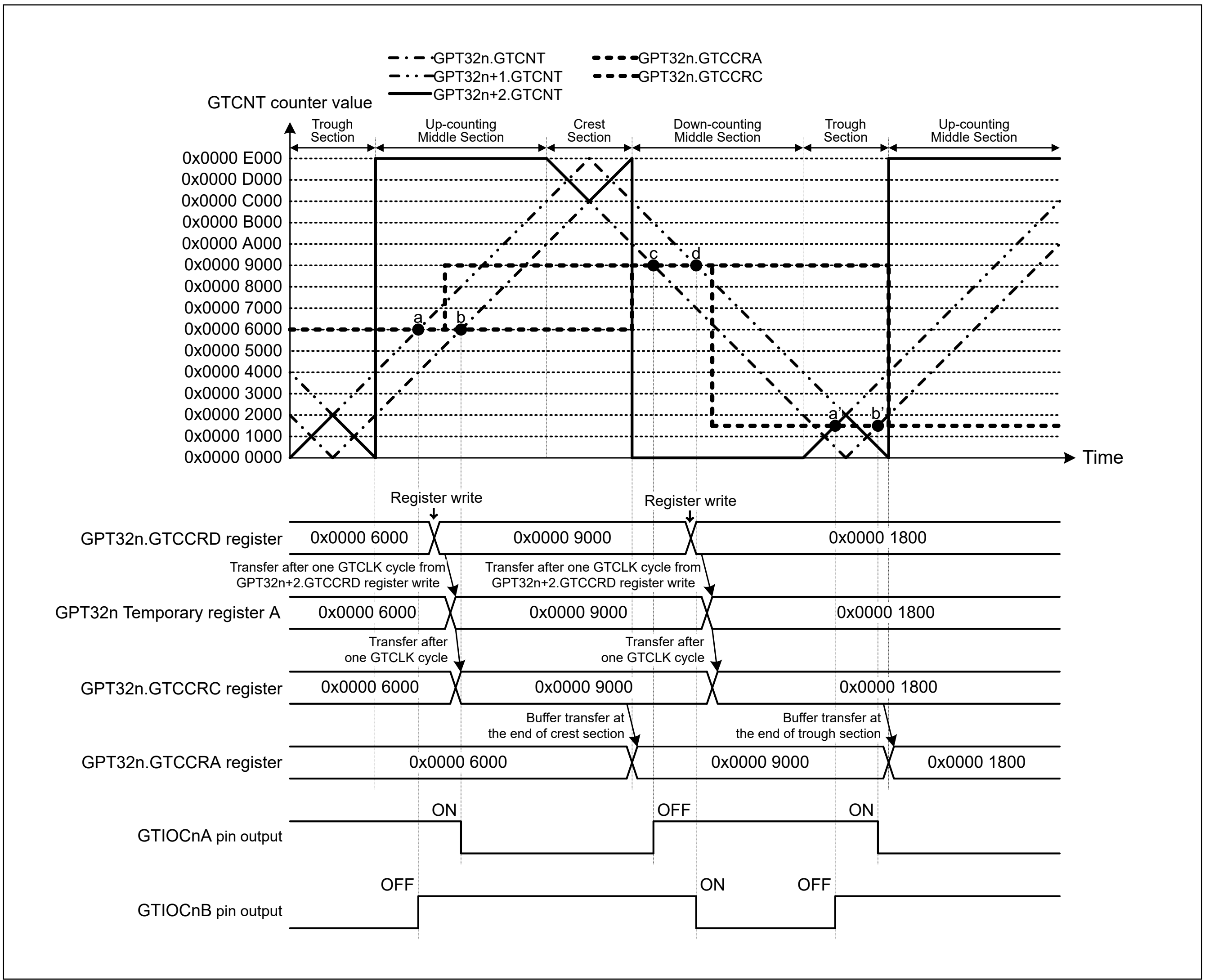


Figure 22.59 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: c → d → a' → b') (n = 4, 7)

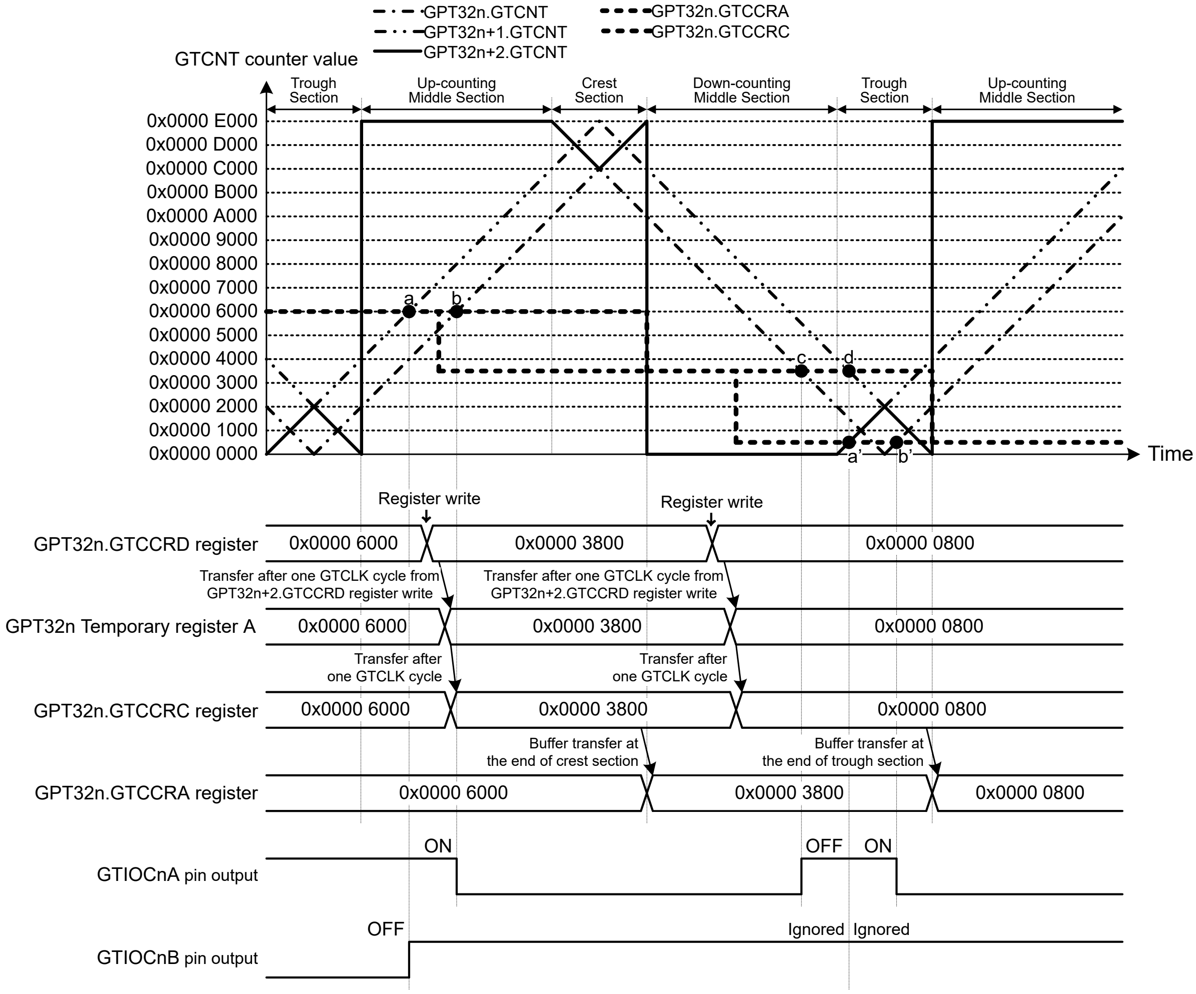


Figure 22.60 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: c → (d, a') → b') (n = 4, 7)

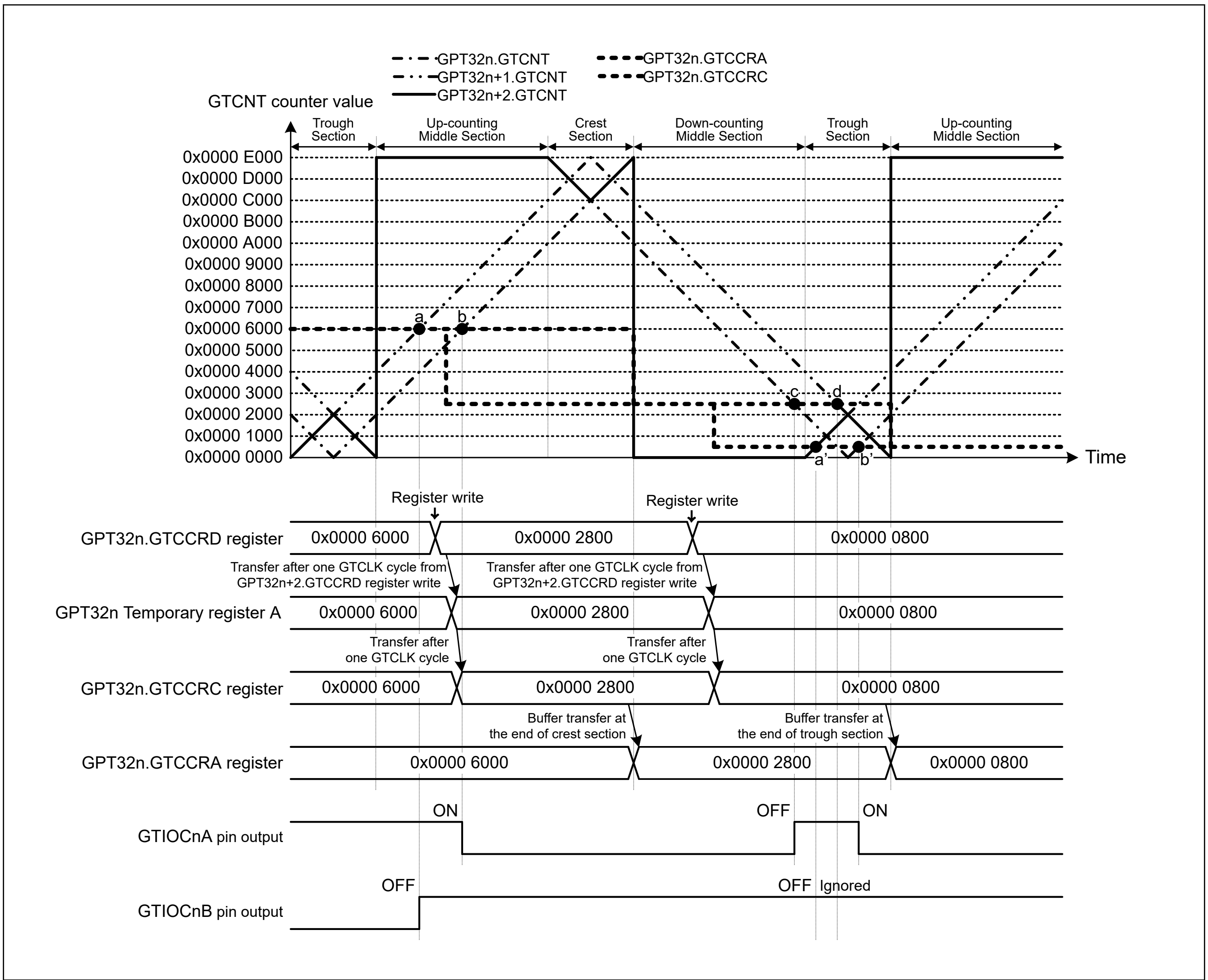


Figure 22.61 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: c → a' → d → b') (n = 4, 7)

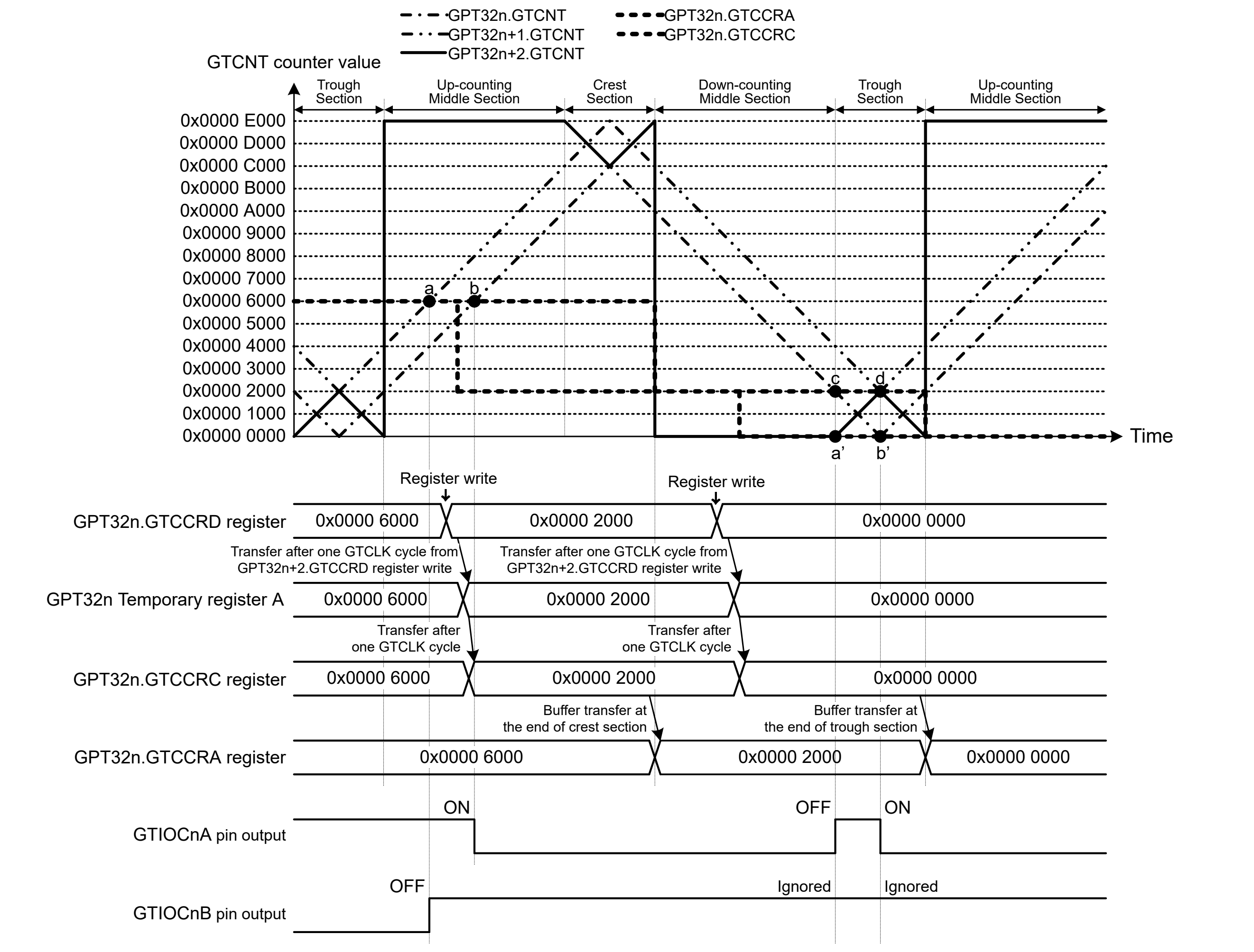


Figure 22.62 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: (c, a’) → (d, b’)) (n = 4, 7)

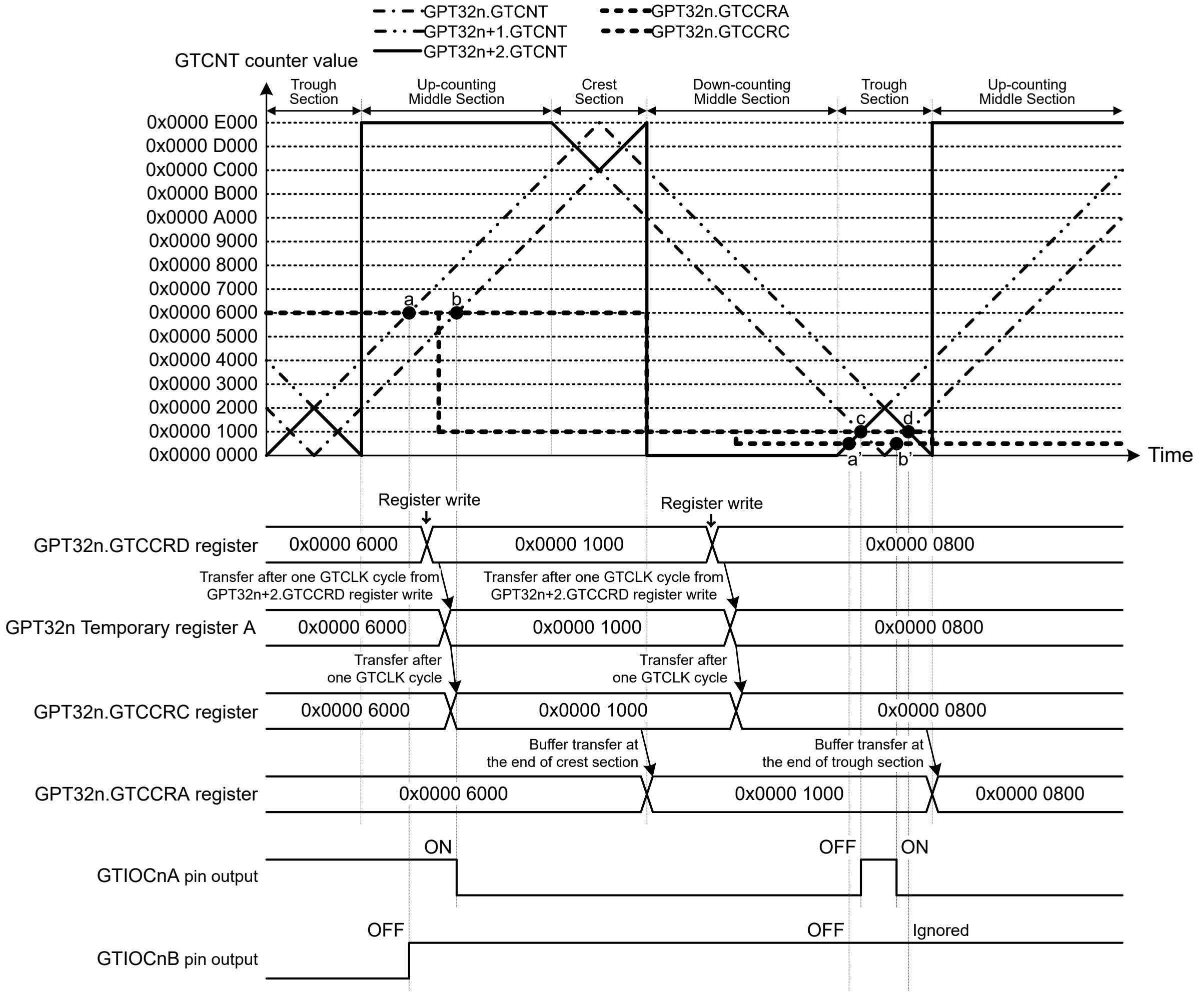


Figure 22.63 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: a’ → c → b’ → d) (n = 4, 7)

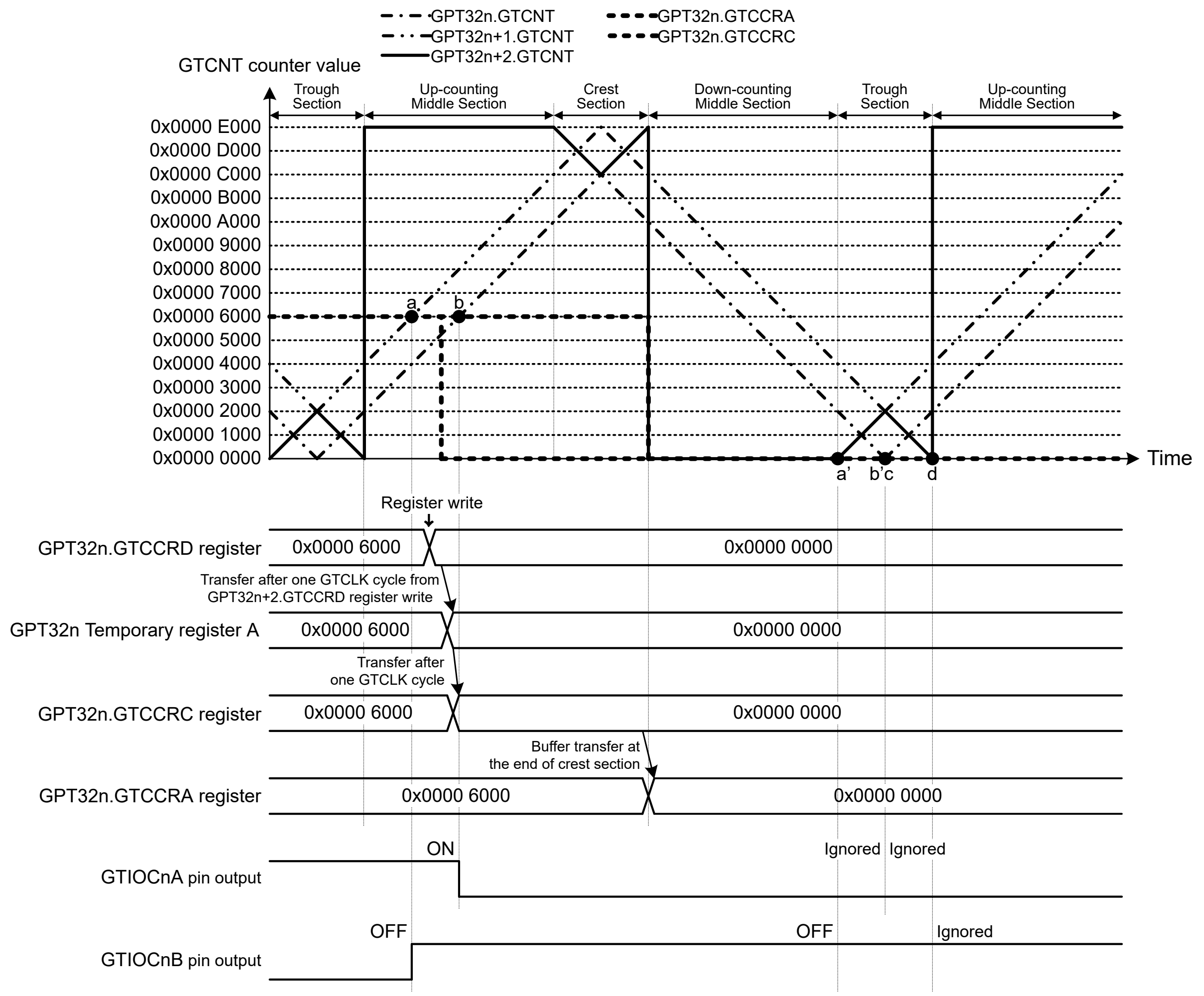


Figure 22.64 Example of Compare Match Generation Sequence Operation in Complementary PWM Mode (Complementary PWM mode 3 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: a' → (b', c) → d) (n = 4, 7)

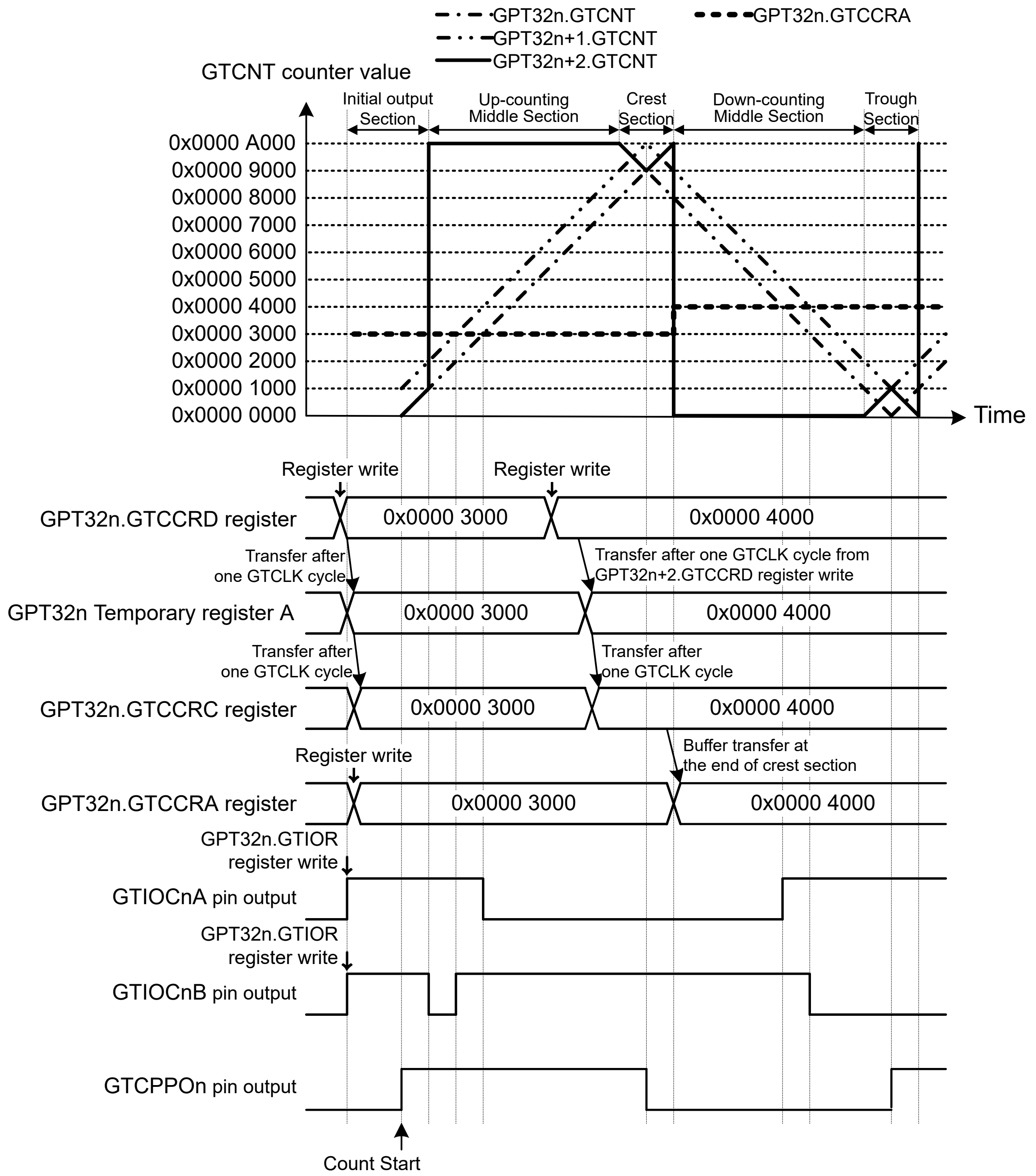


Figure 22.65 **Example of Complementary PWM Mode Initial Output Operation (Complementary PWM mode 1 operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000, in the case that the initial GTCCRA register value is larger than the dead time value) (n = 4, 7)**

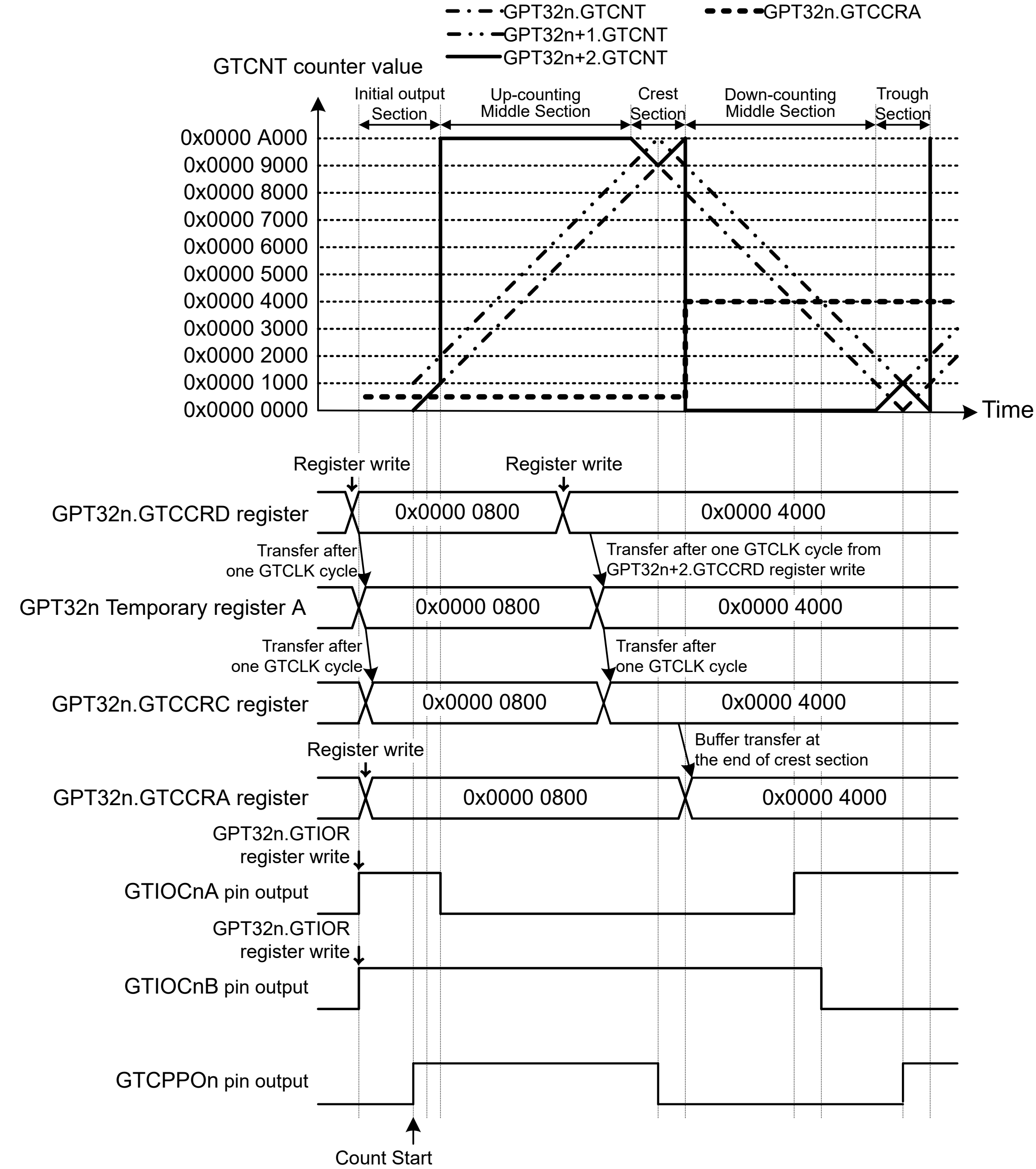


Figure 22.66 Example of Complementary PWM Mode Initial Output Operation (Complementary PWM mode 1 operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000, in the case that the initial GTCCRA register value is equal to or less than the dead time value) (n = 4, 7)

Table 22.35 Example for Setting Complementary PWM Mode 1, 2, 3 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[3:0] bits of GTP32n channel.
2	Select count clock	Select the count clock with GTCR.TPCS[3:0] bits of GPT32n channel.
3	Set cycle	Set the cycle in the GPTWn.GTPR, GPTWn.GTPBR and GPTWn.GTPDBR register.

Table 22.35 Example for Setting Complementary PWM Mode 1, 2, 3 (2 of 2)

No.	Step Name	Description
4	Set GTIOCnm /GTIOCn+1m / GTIOCn+2m pin function	Set the function of the GTIOCnm, GTIOCn+1m, and GTIOCn+2m pins with GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register of the GPT32n, GPT32n+1, and GPT32n+2 channels.
5	Enable GTCPPOn pin output	Set to enable or disable PWM synchronous output from the GTCPPOn pin with the PSYE bit in the GTIOR register of the GPT32n channel.
6	Enable GTIOCnm / GTIOCn+1m /GTIOCn+2m pin output	Set to enable the output from the GTIOCnm, GTIOCn+1m, and GTIOCn+2m pins with the OAE and OBE bits in the GTIOR register of the GPT32n, GPT32n+1, and GPT32n+2 channels.
7	Set buffer operation	In complementary PWM mode 3, set buffer operation with the GTBER2.CP3DB bit of the GPT32n, GPT32n+1, and GPT32n+2 channels.
8	Set compare match value	Set the output pin changing point during up-counting after count start in the GTCCRA register of the GPT32n, GPT32n+1, and GPT32n+2 channels.
9	Set buffer value	For single buffer operation, set data (to be transferred for the first buffer transfer to the GTCCRA register after count start) in the GTCCRD register. For buffer operation in complementary PWM mode 2, set the same value as the GTCCRA register in the GTCCRD register because the first buffer transfer does not take place. For double buffer operation in complementary PWM mode 3, set data to be transferred at the first end of crest in the GTCCRD register and set data to be transferred at the first end of trough in the GTCCRF register.
10	Set dead time value	Set the dead time value in GTDVU of GTP32n channel.
11	Start count operation	Set GTCR.CST of GPT32n channel to 1 to start count operation.
12	Set buffer value for each cycle	For single buffer operation, set data (to be transferred for the next buffer transfer to the GTCCRA register) in the GTCCRD register. For double buffer operation in complementary PWM mode 3, set data to be transferred for the next buffer transfer in the GTCCRF register. Make settings for the GPT32n+2.GTCCRD register finally. (Data is transferred to the temporary register.)

Note: n = 4, 7
m = A, B

22.3.3.8 Complementary PWM mode 4

In complementary PWM mode 4, the value written to the GTCCRD and GTCCRF registers is immediately applied to compare match operation by transferring data also to the GTCCRA register during buffer transfer to a temporary register before crest or trough transfer timing.

Figure 22.67 shows the block diagram in complementary PWM mode 4.

In the configuration of complementary PWM mode 4, a buffer transfer path from the GTCCRD register to the GTCCRC and GTCCRA registers and a buffer transfer path from the GTCCRF register to the GTCCRE and GTCCRA registers are added to other complementary PWM modes shown in Table 22.35.

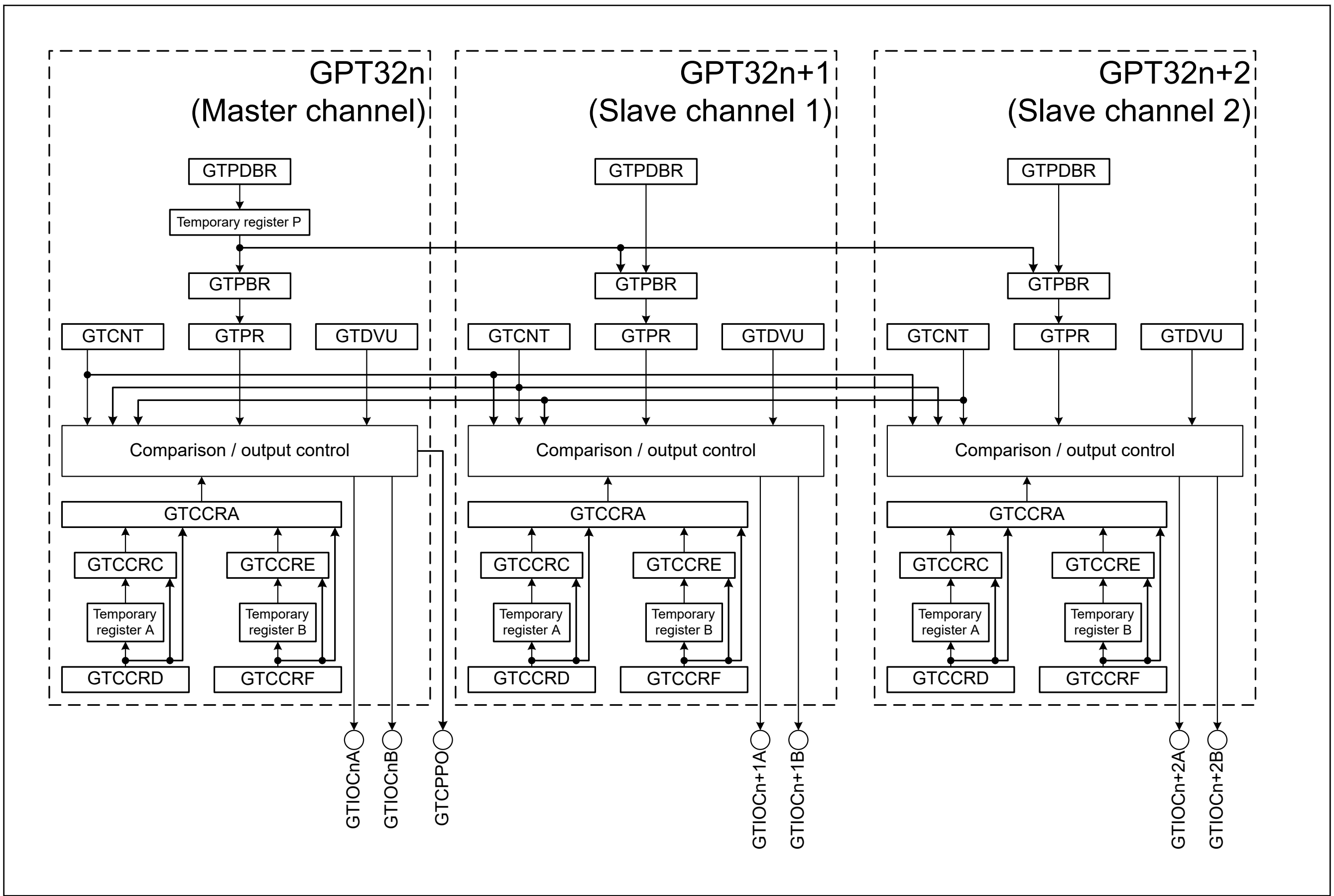


Figure 22.67 Block Diagram in Complementary PWM Mode 4 (n = 4, 7)

Count operation is performed in the same way as complementary PWM modes 1 to 3. See Table 22.30 and Table 22.31.

With respect to buffer operation and PWM waveform changes in complementary PWM mode 4, buffer transfer (shown in Figure 22.67) from the GTCCRD and GTCCRF registers is added based on the complementary PWM mode 3 operation. Buffer transfer and PWM waveform are controlled according to operation section, state of comparison with the GTCCRA register, and write value.

The double buffer function by writing to the GTCCRF register can be enabled or disabled with the GTBER2.CP3DB bit. In double buffer operation, the value written to the GTCCRD register is used as a compare match value for positive-phase OFF (negative-phase ON) during down-counting, and the value written to the GTCCRF register is used as a compare match value for negative-phase OFF (positive-phase ON) during up-counting. Transfer register, transfer value, and PWM output changes are controlled by operation section (where the value is written), state of comparison with the GTCCRA register, and write value.

Double buffer operation is guaranteed only in the up-counting middle section and down-counting middle section. Setting a value not larger than the dead time value and not less than the count cycle is prohibited.

In single buffer operation, a compare match value is written only to the GTCCRD register, and transfer register, transfer value, and PWM output changes are controlled by operation section (where the value is written), state of comparison with the GTCCRA register, and write value.

Transfer from the GTCCRD register to the temporary register A and transfer from the GTCCRF register to the temporary register B are performed in the same way as other complementary PWM modes. Transfers are concurrently performed in three channels by writing a value to the GPT32n+2.GTCCRD register. Transfer from the GTCCRD register to the GTCCRC register, GTCCRA register, temporary register B, and GTCCRE register and transfer from the GTCCRF register to the GTCCRE register and the GTCCRA register are performed at the same time as the above-mentioned transfer to the temporary register.

Table 22.36 and Table 22.37 show immediate buffer transfer (for transfer to the temporary register by writing a value to the GPT32n+2.GTCCRD register) to the GTCCRC and GTCCRA registers by writing a value to the GTCCRD register during single buffer operation in complementary PWM mode 4 for each compare match state in each operation section. Transfers (from the GTCCRD register to the temporary register A, from the temporary register A to the GTCCRC register, and from the GTCCRC register to the GTCCRA register) other than those shown in Table 22.36 and Table 22.37 are the same as single buffer transfer in complementary PWM mode 3 shown in Table 22.32.

Table 22.38 and Table 22.39 show immediate buffer transfer (for transfer to the temporary register by writing a value to the GPT32n+2.GTCCRD register) to each register by writing a value to the GTCCRD and GTCCRF registers during double buffer operation in complementary PWM mode 4 for each compare match state in each operation section. Transfers (from the GTCCRD register to the temporary register A, from the GTCCRF register to the temporary register B, from the temporary register A to the GTCCRC register, from the temporary register B to the GTCCRE register, and from the GTCCRC and GTCCRE registers to the GTCCRA register) other than those shown in Table 22.38 and Table 22.39 are the same as double buffer transfer in complementary PWM mode 3 shown in Table 22.33.

Table 22.36 Immediate Single Buffer Transfer from GTCCRD Register in Complementary PWM Mode 4 (1 of 2)

Operation Section	Compare Match State	Immediate Transfer Destination Register	
		GTCCRC	GTCCRA
Up- counting middle section	Before up-counting compare match	GTCCRD	(i) In the case of GTCCRD > GPT32n+1.GTCNT GTCCRD (ii) In the case of GTCCRD < = GPT32+1.GTCNT GPT32n+1.GTCNT Negative-phase OFF
	Up-counting dead time period	GTCCRD	No transfer
	After up-counting compare match	GTCCRD	No transfer
Up-counting crest section	Before up-counting compare match	Before down-counting compare match GTCCRD After down-counting dead time start No transfer	Before down-counting compare match (i) In the case of GTCCRD > GPT32n+1.GTCNT GTCCRD (ii) In the case of GTCCRD < = GPT32n+1.GTCNT GPT32n+1.GTCNT Negative-phase OFF After down-counting dead time start No transfer
	Up-counting dead time period	Before down-counting compare match GTCCRD After down-counting dead time start No transfer	No transfer
	After up-counting compare match	Before down-counting compare match (i) In the case of GTCCRD < GPT32n+2.GTCNT GTCCRD (ii) In the case of GTCCRD = > GPT32n+2.GTCNT GPT32n+2.GTCNT Positive-phase OFF After down-counting dead time start No transfer	No transfer

Table 22.36 Immediate Single Buffer Transfer from GTCCRD Register in Complementary PWM Mode 4 (2 of 2)

Operation Section	Compare Match State	Immediate Transfer Destination Register	
		GTCCRC	GTCCRA
Down-counting crest section	Before down-counting compare match	Up-counting dead time period (i) In the case of GTCCRD < GPT32n+1.GTCNT GTCCRD (ii) In the case of GTCCRD = > GPT32n+1.GTCNT GPT32n+1.GTCNT Negative-phase ON After up-counting compare match (i) In the case of GTCCRD < GPT32n.GTCNT GTCCRD (ii) In the case of GTCCRD = > GPT32n.GTCNT GPT32n.GTCNT Positive-phase OFF	No transfer
	Down-counting dead time period	No transfer	No transfer
	After down-counting compare match	No transfer	No transfer

Table 22.37 Immediate Single Buffer Transfer from GTCCRD Register in Complementary PWM Mode 4 (1 of 2)

Operation Section	Compare Match State	Immediate Transfer Destination Register	
		GTCCRC	GTCCRA
Down-counting middle section	Before down-counting compare match	GTCCRD	(i) In the case of GTCCRD < GPT32n.GTCNT GTCCRD (ii) In the case of GTCCRD = > GPT32n.GTCNT GPT32n.GTCNT Positive-phase OFF
	Down-counting dead time period	GTCCRD	No transfer
	After down-counting compare match	GTCCRD	No transfer

Table 22.37 Immediate Single Buffer Transfer from GTCCRD Register in Complementary PWM Mode 4 (2 of 2)

Operation Section	Compare Match State	Immediate Transfer Destination Register	
		GTCCRC	GTCCRA
Down-counting trough section	Before down-counting compare match	Before up-counting compare match GTCCRD After up-counting dead time start No transfer	Before up-counting compare match (i) In the case of GTCCRD < GPT32n.GTCNT GTCCRD (ii) In the case of GTCCRD = > GPT32n.GTCNT GPT32n.GTCNT Positive-phase OFF After up-counting dead time start No transfer
	Down-counting dead time period	Before up-counting compare match GTCCRD After up-counting dead time start No transfer	No transfer
	After down-counting compare match	Before up-counting compare match (i) In the case of GTCCRD > GPT32n+2.GTCNT GTCCRD (ii) In the case of GTCCRD < = GPT32n+2.GTCNT GPT32n+2.GTCNT Negative-phase OFF After up-counting dead time start No transfer	No transfer
Up- counting trough section	Before up-counting compare match	Down-counting dead time period (i) In the case of GTCCRD > GPT32n.GTCNT GTCCRD (ii) In the case of GTCCRD < = GPT32n.GTCNT GPT32n.GTCNT Positive-phase ON After down-counting compare match (i) In the case of GTCCRD > GPT32n+1.GTCNT GTCCRD (ii) In the case of GTCCRD < = GPT32n+1.GTCNT GPT32n+1.GTCNT Negative-phase OFF	No transfer
	Up-counting dead time period	No transfer	No transfer
	After up-counting compare match	No transfer	No transfer

Table 22.38 Immediate Double Buffer Transfer from GTCCRD and GTCCRF Registers in Complementary PWM Mode 4

Operation Section	Compare Match State	Immediate Transfer Destination Register		
		GTCCRC	GTCCRE	GTCCRA
Up-counting middle section	Before up-counting compare match	GTCCRD	GTCCRF	(i) In the case of GTCCRF > GPT32n+1.GTCNT GTCCRF (ii) In the case of GTCCRF < = GPT32+1.GTCNT GPT32n+1.GTCNT Negative-phase OFF
	Up-counting dead time period	GTCCRD	GTCCRF	No transfer
	After up-counting compare match	GTCCRD	GTCCRF	No transfer

Table 22.39 Immediate Double Buffer Transfer from GTCCRD and GTCCRF Registers in Complementary PWM Mode 4

Operation Section	Compare Match State	Immediate Transfer Destination Register		
		GTCCRC	GTCCRE	GTCCRA
Down-counting middle section	Before down-counting compare match	GTCCRD	GTCCRF	(i) In the case of GTCCRD < GPT32n.GTCNT GTCCRD (ii) In the case of GTCCRD => GPT32n.GTCNT GPT32n.GTCNT Positive-phase OFF
	Down-counting dead time period	GTCCRD	GTCCRF	No transfer
	After down-counting compare match	GTCCRD	GTCCRF	No transfer

Figure 22.68 to Figure 22.72 show examples of single buffer operation for each operation section in complementary PWM mode 4.

Figure 22.73 to Figure 22.76 show examples of double buffer operation for each operation section in complementary PWM mode 4.

Table 22.40 shows an example for setting complementary PWM mode 4.

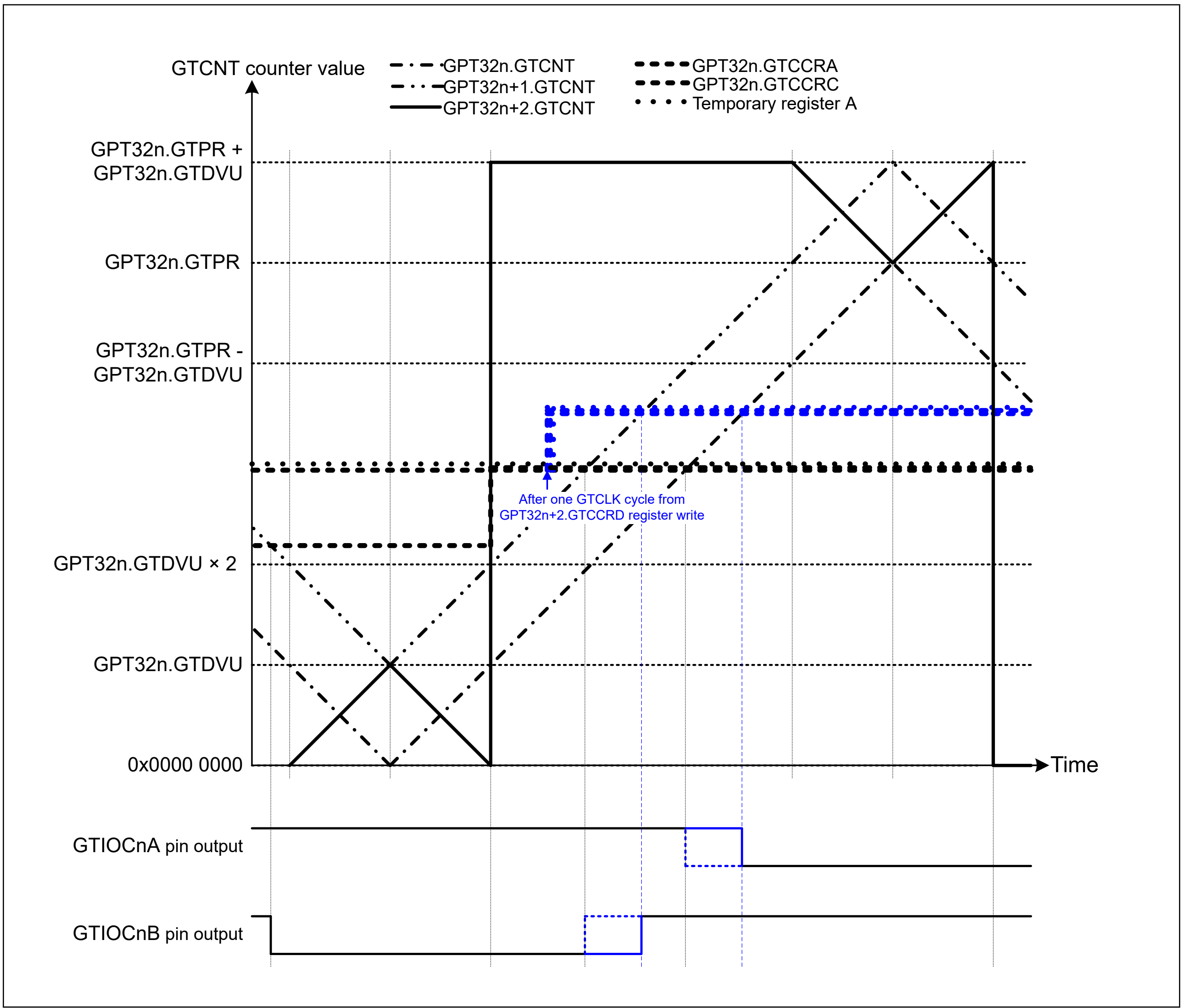


Figure 22.68 Example of Complementary PWM Mode 4 Single Buffer Operation (Up-Counting Middle Section) (Complementary PWM mode 4 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when a value larger than the GPT32n+1.GTCNT value is written to GTCCRD register before up-counting compare match) (n = 4, 7)

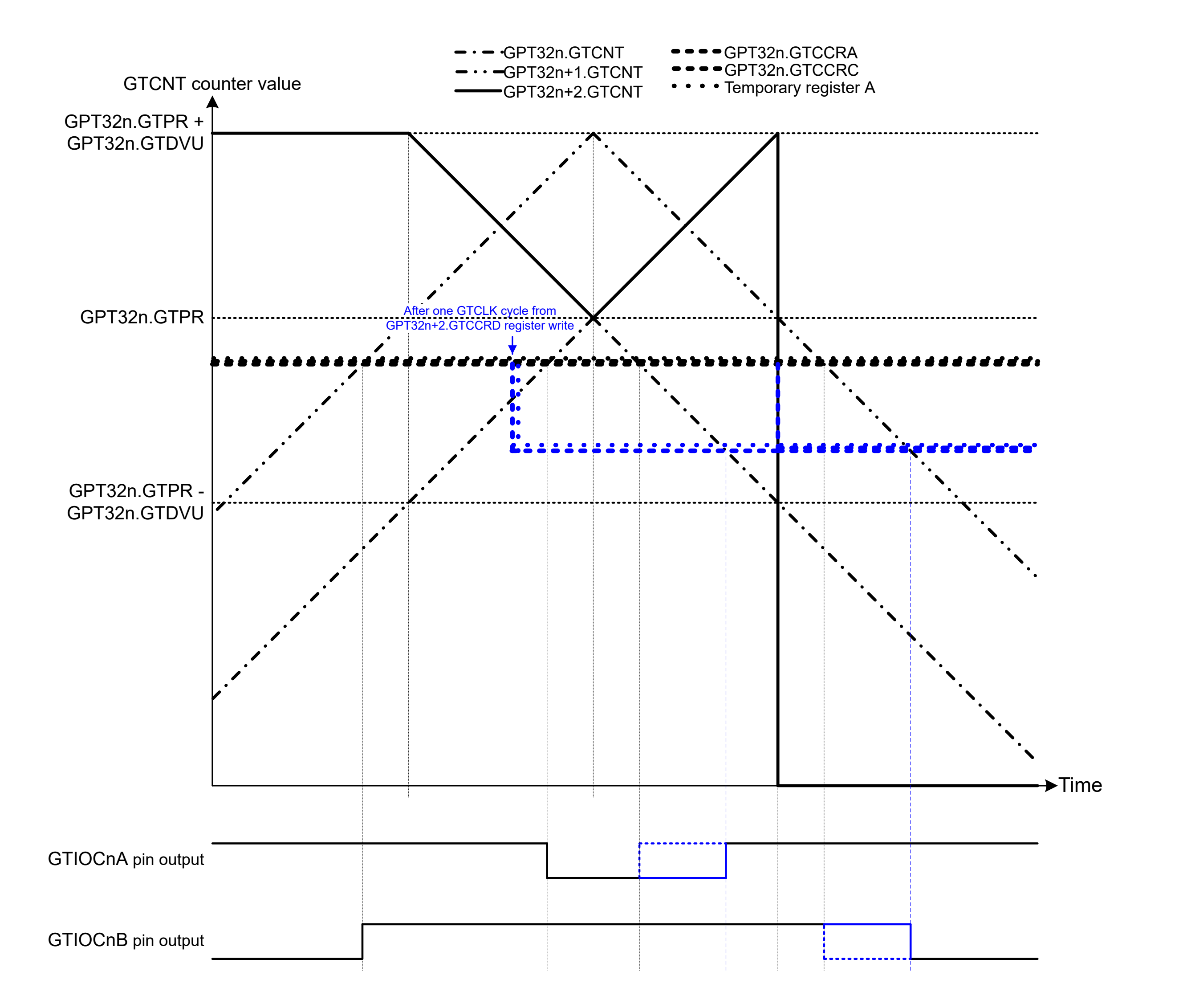


Figure 22.69 Example of Complementary PWM Mode 4 Single Buffer Operation (Up-Counting Crest Section) (Complementary PWM mode 4 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when a value is written to GTCCRD register during the up-counting dead time) (n = 4, 7)

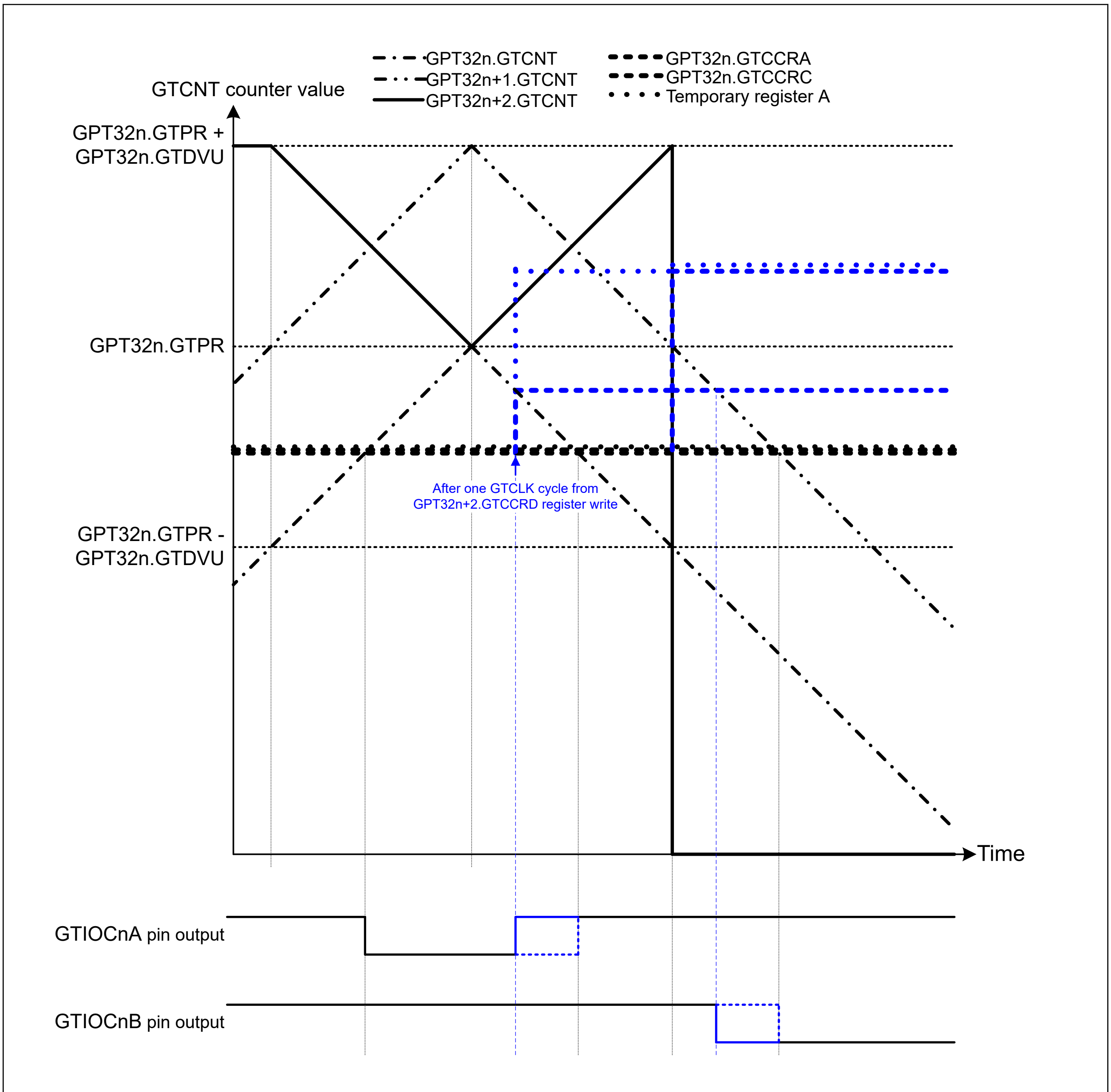


Figure 22.70 Example of Complementary PWM Mode 4 Single Buffer Operation (Down-Counting Crest Section) (Complementary PWM mode 4 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when a value not less than the GPT32n.GTCNT value is written to GTCCRD register after up-counting compare match before down-counting compare match) (n = 4, 7)

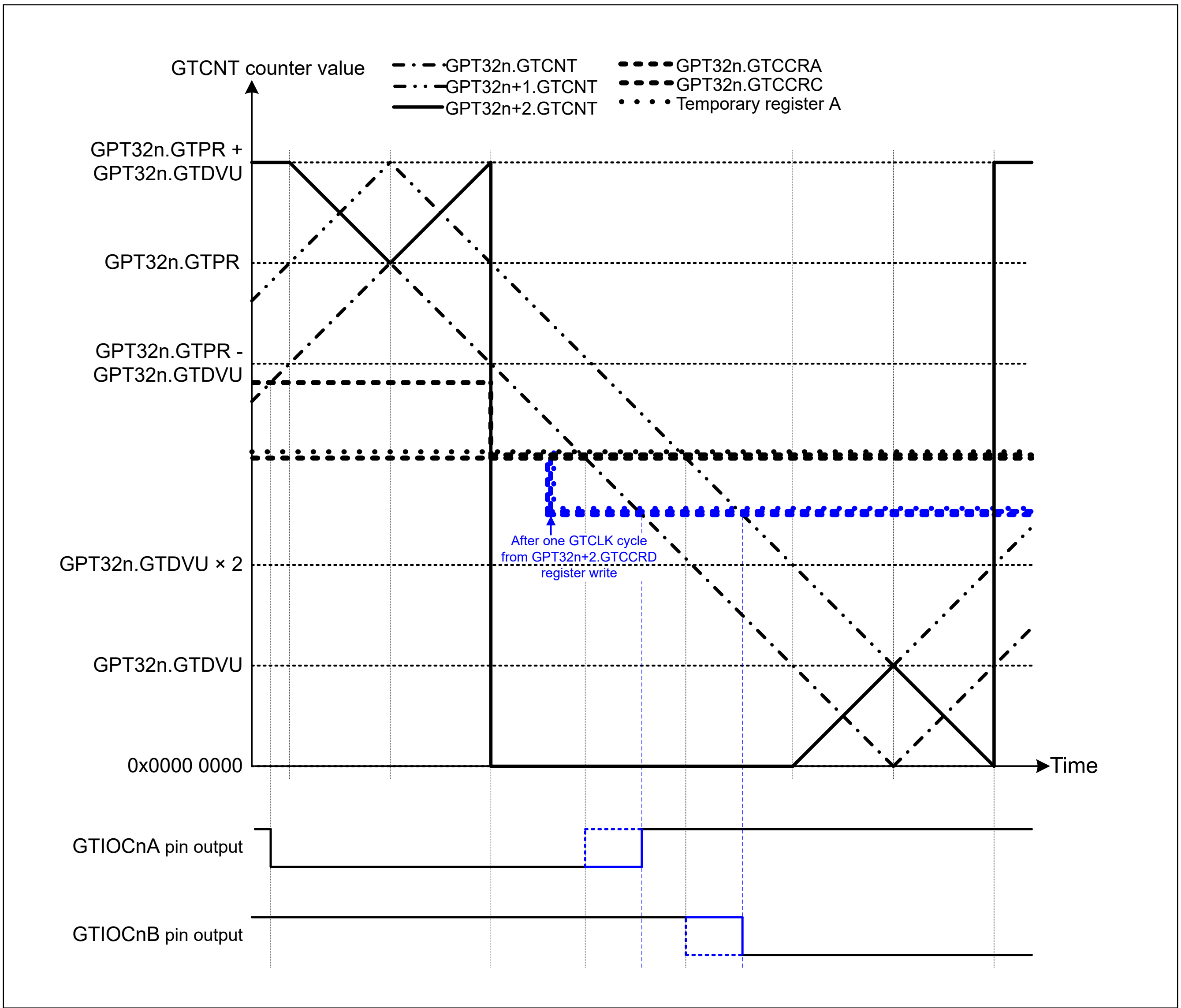


Figure 22.71 Example of Complementary PWM Mode 4 Single Buffer Operation (Down-Counting Middle Section) (Complementary PWM mode 4 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when a value smaller than the GPT32n.GTCNT value is written to GTCCRD register before down-counting compare match) (n = 4, 7)

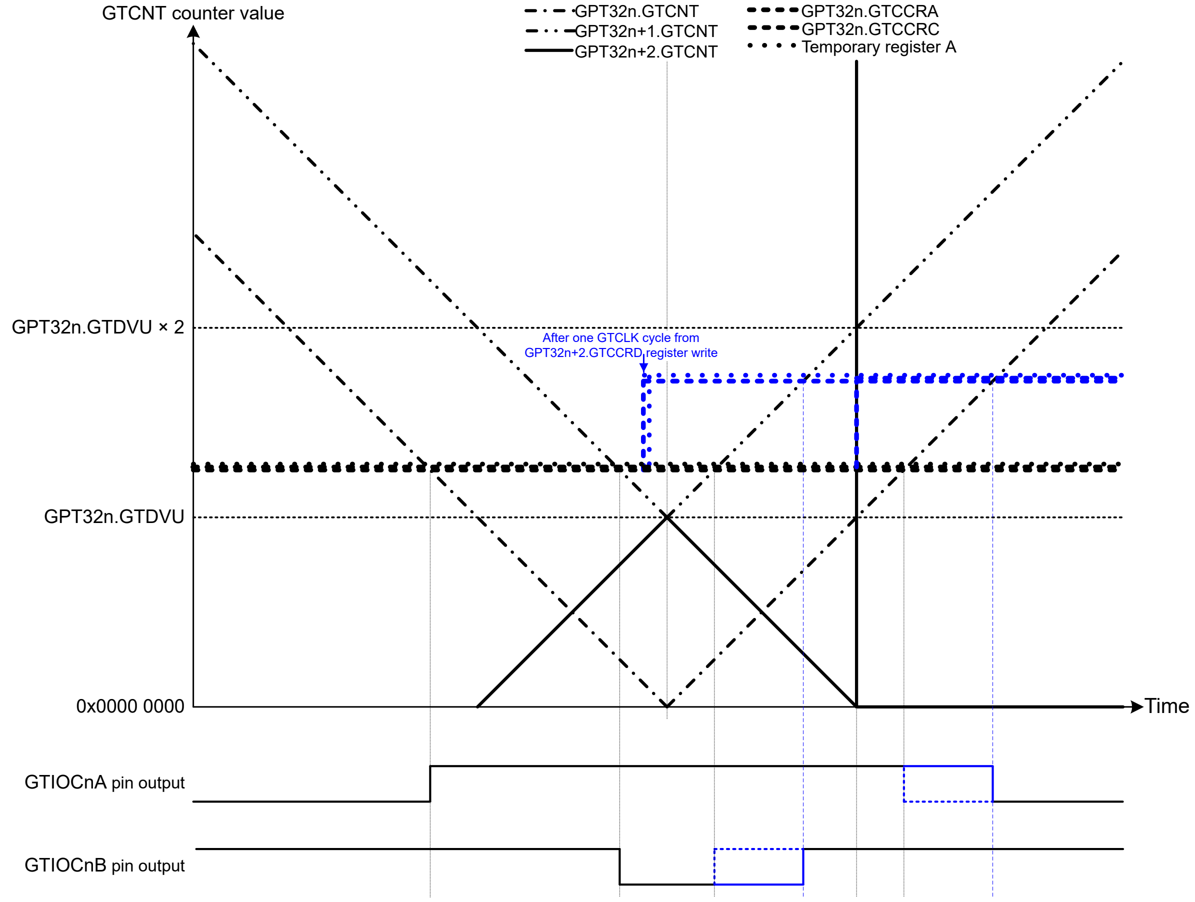


Figure 22.72 Example of Complementary PWM Mode 4 Single Buffer Operation (Down-Counting Trough Section) (Complementary PWM mode 4 single buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when a value larger than the GPT32n+2.GTCNT value is written to GTCCRD register after down-counting compare match) (n = 4, 7)

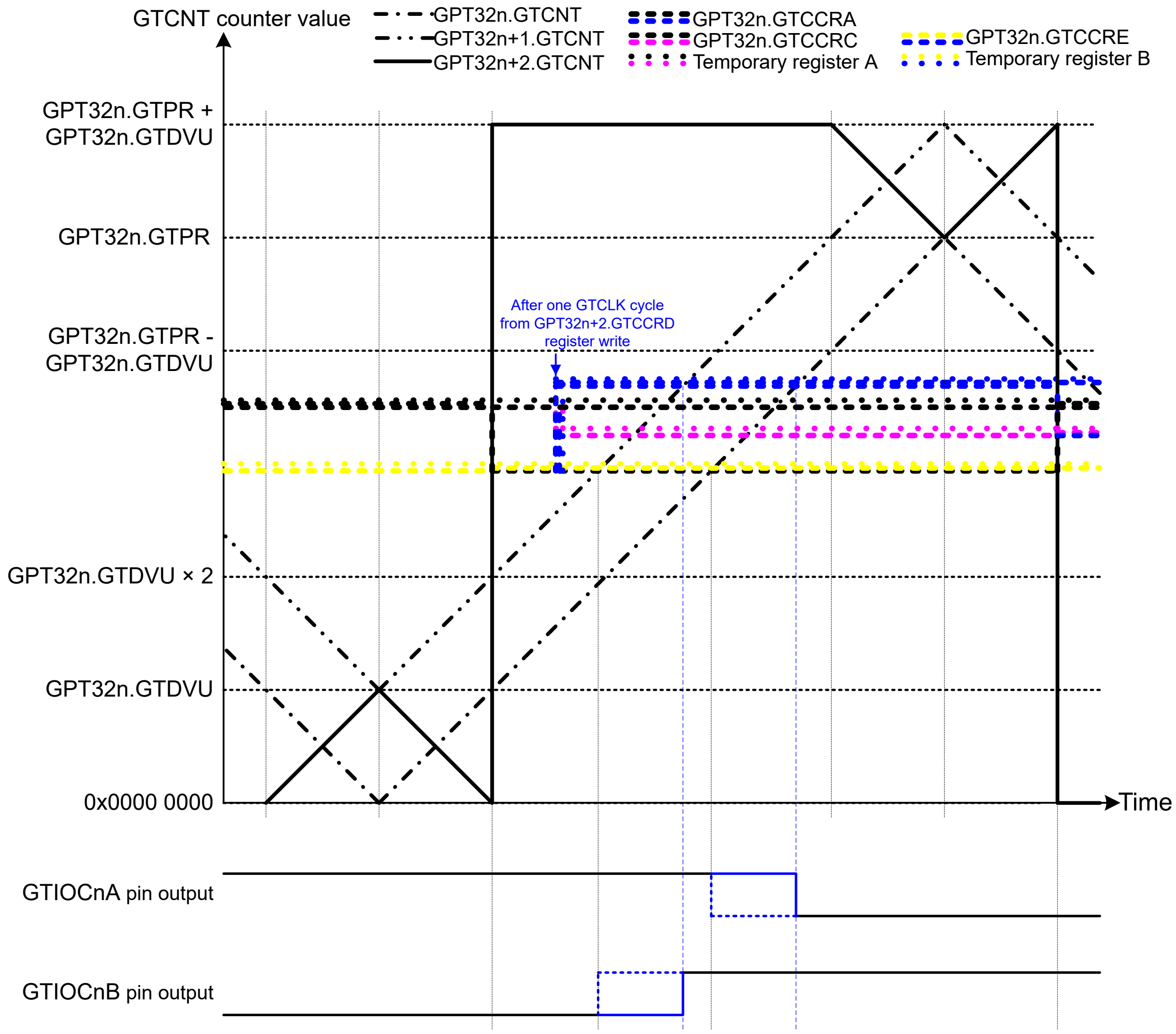


Figure 22.73 Example of Complementary PWM Mode 4 Double Buffer Operation (Up-Counting Middle Section) (Complementary PWM mode 4 double buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when a value larger than the GPT32n+1.GTCNT value is written to GTCCRF register before up-counting compare match) (n = 4, 7)

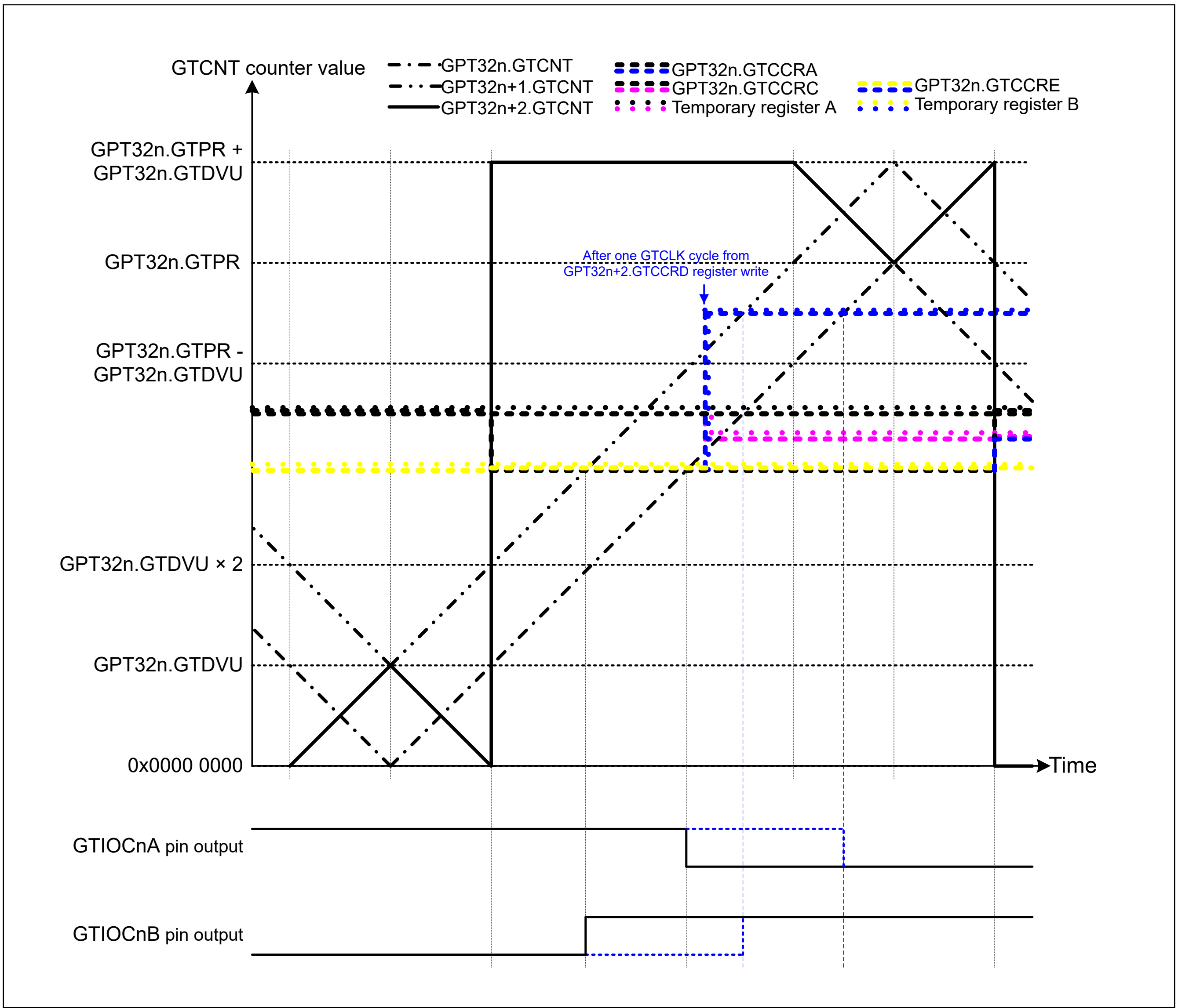


Figure 22.74 Example of Complementary PWM Mode 4 Double Buffer Operation (Up-Counting Middle Section) (Complementary PWM mode 4 double buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when a value is written to GTCCRF register after up-counting compare match) (n = 4, 7)

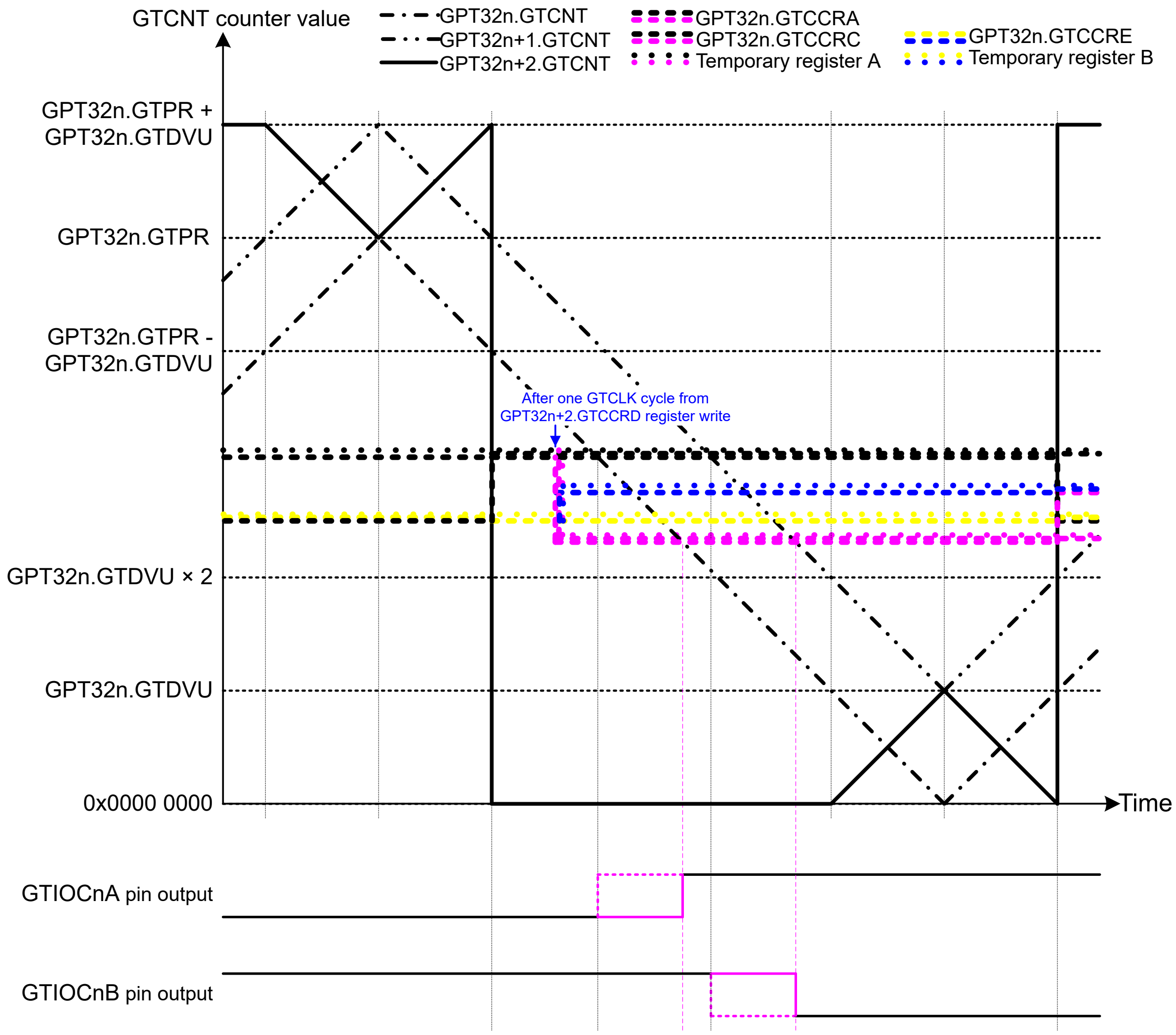


Figure 22.75 Example of Complementary PWM Mode 4 Double Buffer Operation (Down-Counting Middle Section) (Complementary PWM mode 4 double buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when a value smaller than the GPT32n.GTCNT value is written to GTCCRD register before down-counting compare match) (n = 4, 7)

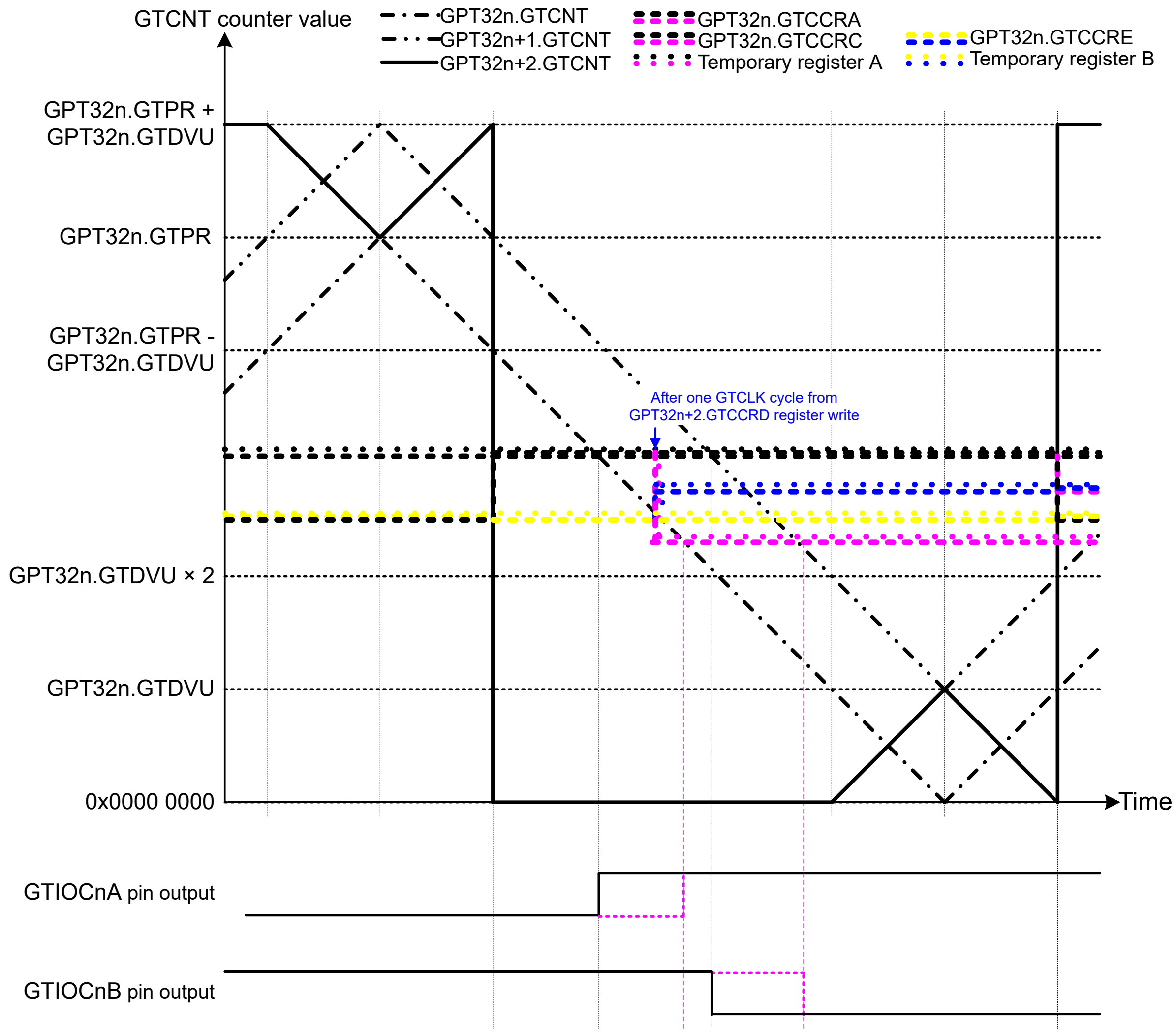


Figure 22.76 Example of Complementary PWM Mode 4 Double Buffer Operation (Down-Counting Middle Section) (Complementary PWM mode 4 double buffer operation, GTIOCnA pin = Low / GTIOCnB pin = High at GTCCRA register compare match during up-counting, GTIOCnA pin = High / GTIOCnB pin = Low at GTCCRA register compare match during down-counting, when a value is written to GTCCRD register during the down-counting dead time) (n = 4, 7)

Table 22.40 Example for Setting Complementary PWM Mode 4 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode (1111b) with GTCR.MD[3:0] of GTP32n channel.
2	Select count clock	Select the count clock with GTCR.TPCS[3:0] bits of GPT32n channel.
3	Set cycle	Set the cycle in the GPTWn.GTPR, GPTWn.GTPBR and GPTWn.GTPDBR register.
4	Set GTIOCnm /GTIOCn+1m / GTIOCn+2m pin function	Set the function of the GTIOCnm, GTIOCn+1m, and GTIOCn+2m pins with GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register of the GPT32n, GPT32n+1, and GPT32n+2 channels.
5	Enable GTCPPOn pin output	Set to enable or disable PWM synchronous output from the GTCPPOn pin with the PSYE bit in the GTIOR register of the GPTWn channel.
6	Enable GTIOCnm / GTIOCn+1m /GTIOCn+2m pin output	Set to enable the output from the GTIOCnm, GTIOCn+1m, and GTIOCn+2m pins with the OAE and OBE bits in the GTIOR register of the GPT32n, GPT32n+1, and GPT32n+2 channels.
7	Set buffer operation	Set buffer operation with the GTBER2.CP3DB bit of the GPT32n, GPT32n+1, and GPT32n+2 channels.

Table 22.40 Example for Setting Complementary PWM Mode 4 (2 of 2)

No.	Step Name	Description
8	Set compare match value	Set the output pin changing point during up-counting after count start in the GTCCRA register of the GPT32n, GPT32n+1, and GPT32n+2 channels.
9	Set buffer value	For single buffer operation, set data (to be transferred for the first buffer transfer to the GTCCRA register after count start) in the GTCCRD register. For double buffer operation, set data to be transferred at the first end of crest in the GTCCRD register and set data to be transferred at the first end of trough in the GTCCRF register.
10	Set dead time value	Set the dead time value in GTDVU of GTP32n channel.
11	Start count operation	Set GTCR.CST of GPT32n channel to 1 to start count operation.
12	Set compare match value to be transferred immediately	For single buffer operation, set data (to be transferred immediately to the GTCCRA register) in the GTCCRD register. For double buffer operation, set the compare match value for down-counting to be transferred immediately in the GTCCRD register and the compare match value for up-counting in the GTCCRF register. Make settings for the GPT32n+2.GTCCRD register finally. (Data is transferred to the temporary register.)

Note: n = 4, 7
m = A, B

22.3.4 Automatic Dead Time Setting Function

By setting GTDTCR, a compare match value for a negative waveform with dead time obtained by a compare match value for a positive waveform (GTCCRA value) and specified dead time value (GTDVU and GTDVD value) can automatically be set to GTCCRB.

The automatic dead time setting function can be used in saw-wave one-shot pulse mode and all the triangle PWM modes.

Dead time can be separately set for the first half and second half of a waveform. Dead time for the changing point in the first half of a negative waveform is set in the GTDVU register and that in the second half is set in GTDVD register. The same dead time can also be set for the first and second halves by setting the GTDTCR.TDFER bit to 1.

The GTDBU register can be used as a buffer register of the GTDVU register, and the GTDBD register can be used as a buffer register of the GTDVD registers. Buffer transfer is performed at the end of the cycle (in saw-wave mode: either of an overflow of the GTCNT counter (up-counting), an underflow (down-counting), or the GTCNT counter clearing; in triangle-wave mode: a trough).

The change point of the negative-phase waveform, which is automatically calculated, is obtained by reading the GTCCRB register. Writing to GTCCRB is prohibited when the automatic dead time setting function is used.

Do not set the dead time that makes the change point of the waveform exceeding the count period. When any dead-time setting which would generate a dead-time error is made, adjust the change points of the positive- and negative-phase waveforms to generate waveforms with secured dead-time as shown in Table 22.41. The adjusted change point of the negative-phase waveform is automatically set in the GTCCRB register. An internal signal is used to judge the change point of the positive-phase waveform, thus the value of the GTCCRA register is not updated by the adjusted value.

In saw-wave one-shot pulse mode, if the order of the change point becomes inconsistent by adjustment of the waveform change point due to occurrence of dead time errors, or if the change point exceeds the count period even after the adjustment, the complementary relation between the positive- and negative-phases cannot be guaranteed.

In triangle-wave PWM mode, if dead time exceeds the count period by setting 0x00000000 or a value greater than or equal to the setting value of the GTPR register is set in the GTCCRA register, output change is controlled by the output protection function (see section 22.8.4. Output Protection Function for GTIOCnm Pin Output (n = 0 to 13; m = A, B)). When the GTCCRA register is greater than or equal to [GTPR register + GTDVm (m = U, D) register], [GTPR register - 1] is set in the GTCCRB register as the upper limit.

Automatic setting for a dead time value to the GTCCRB register is performed at the next count clock after the register value for calculating the automatic setting value is updated. In triangle-wave mode, it also can be done at the next count clock from the current crest.

Table 22.41 Adjustment of the Waveform Change Point When a Dead-Time Error Occurs

PWM Output Operating Mode	Count Direction	Period	Condition for Dead Time Error	Change Point of the Positive-Phase Waveform after Adjustment	Change Point of the Negative-Phase Waveform after Adjustment
Saw-wave one-shot pulse mode	Up-counting	First half	GTCCRA - GTDVU < 0	GTDVU	0
		Second half	GTCCRA + GTDVD > GTPR (GTCCRA + GTDVU > GTPR)*1	GTPR - GTDVD (GTPR - GTDVU)*1	GTPR
	Down-counting	First half	GTCCRA + GTDVU > GTPR	GTPR - GTDVU	GTPR
		Second half	GTCCRA - GTDVD < 0 (GTCCRA - GTDVU < 0)*1	GTDVD (GTDVU)*1	0
Triangle-wave PWM mode 1/2/3	Up-counting	(First half)	GTCCRA - GTDVU ≤ 0	GTDVU + 1	1
	Down-counting	(Second half)	GTCCRA - GTDVD < 0 (GTCCRA - GTDVU < 0)*1	GTDVD (GTDVU)*1	0

Note 1. In the case of GTDTCR.TDFER = 1.

Figure 22.77 to Figure 22.80 show examples of automatic dead time setting function operation. Table 22.42 and Table 22.43 show the setting examples.

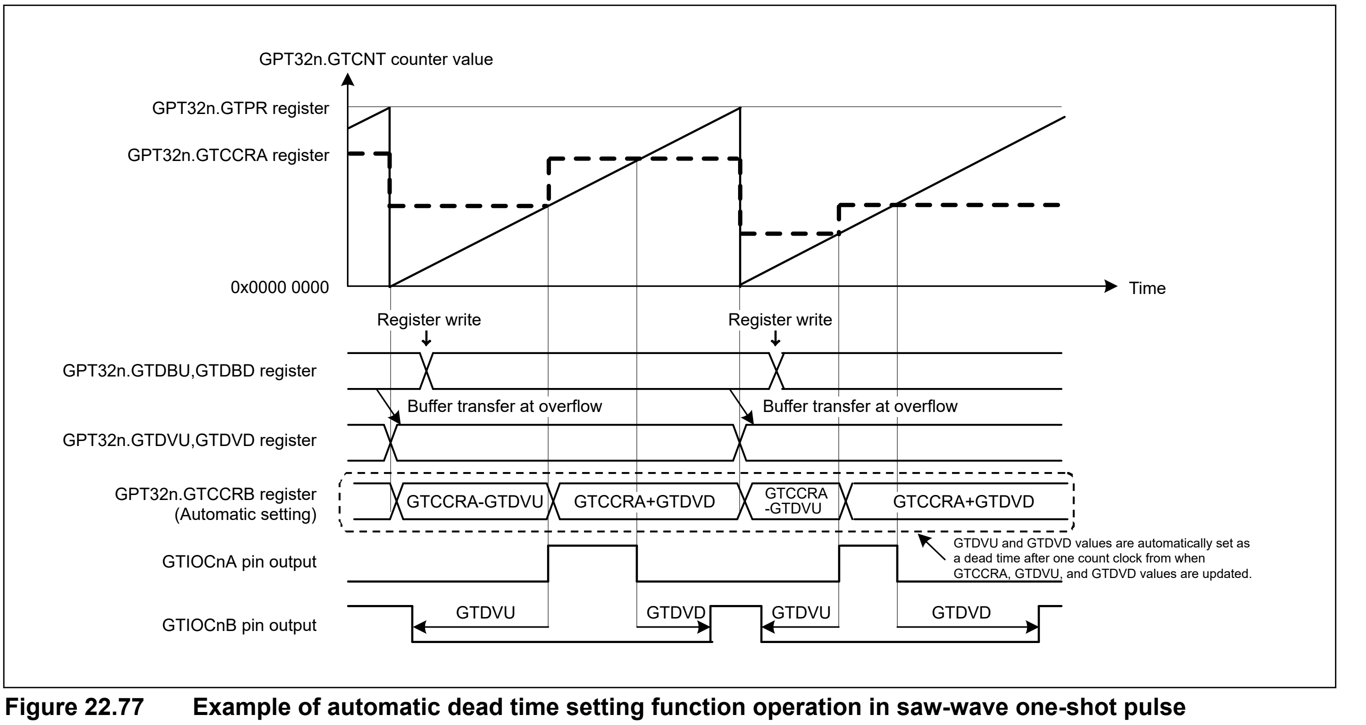


Figure 22.77 Example of automatic dead time setting function operation in saw-wave one-shot pulse mode, up-counting, GTDVU and GTDVD set to buffer operation, and active-high

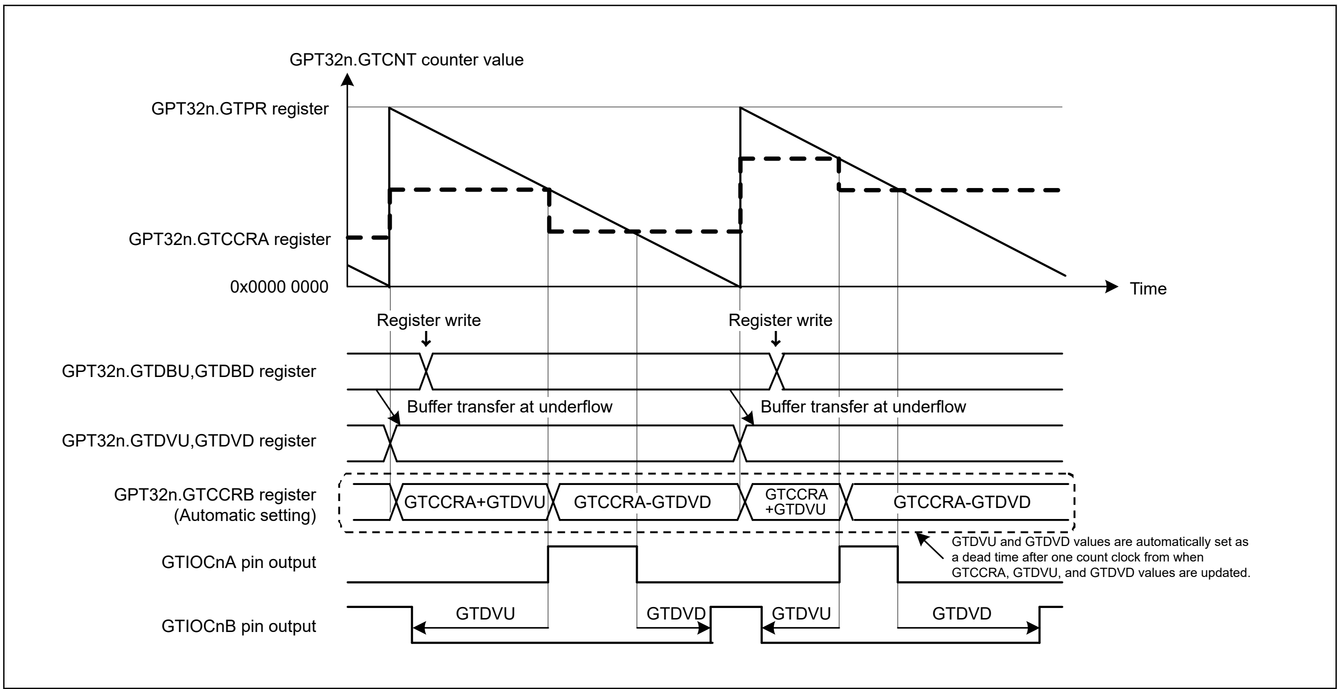


Figure 22.78 Example of automatic dead time setting function operation in saw-wave one-shot pulse mode, down-counting, GTDVU and GTDVD set to buffer operation, and active-high

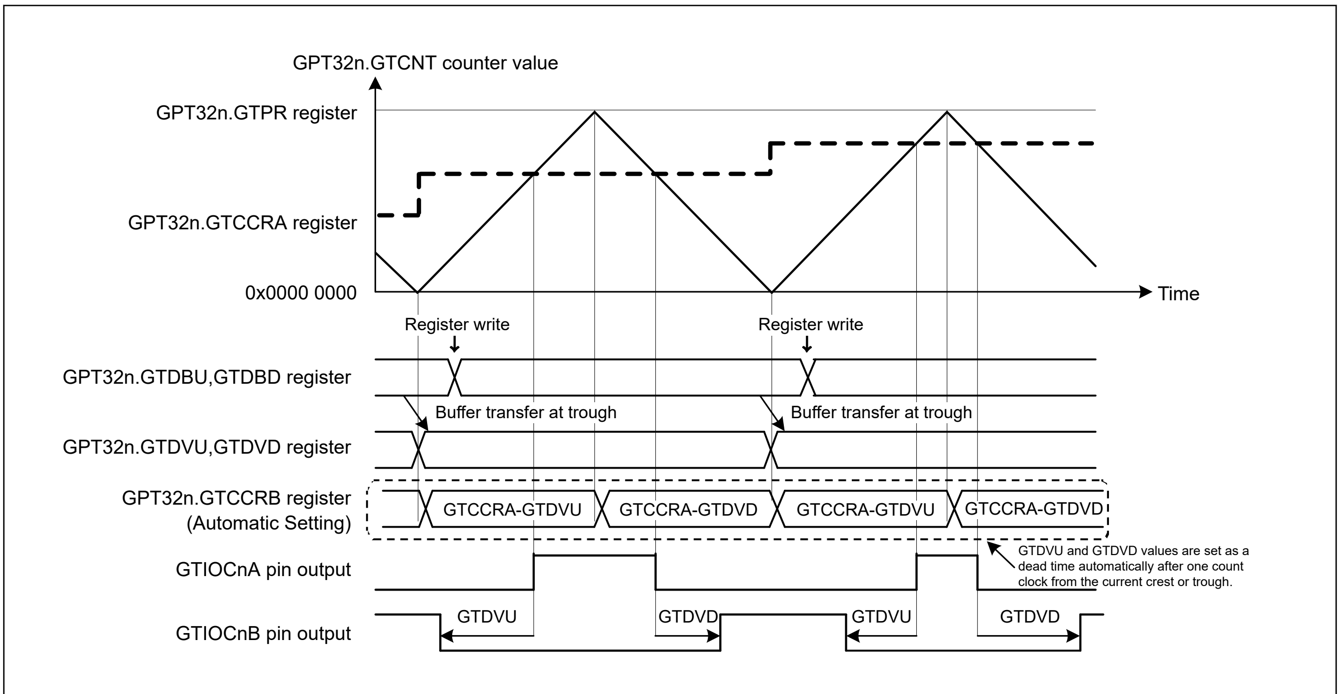


Figure 22.79 Example of automatic compare-match value setting function with dead time in triangle-wave PWM mode 1, GTDVU and GTDVD set to buffer operation, and active-high

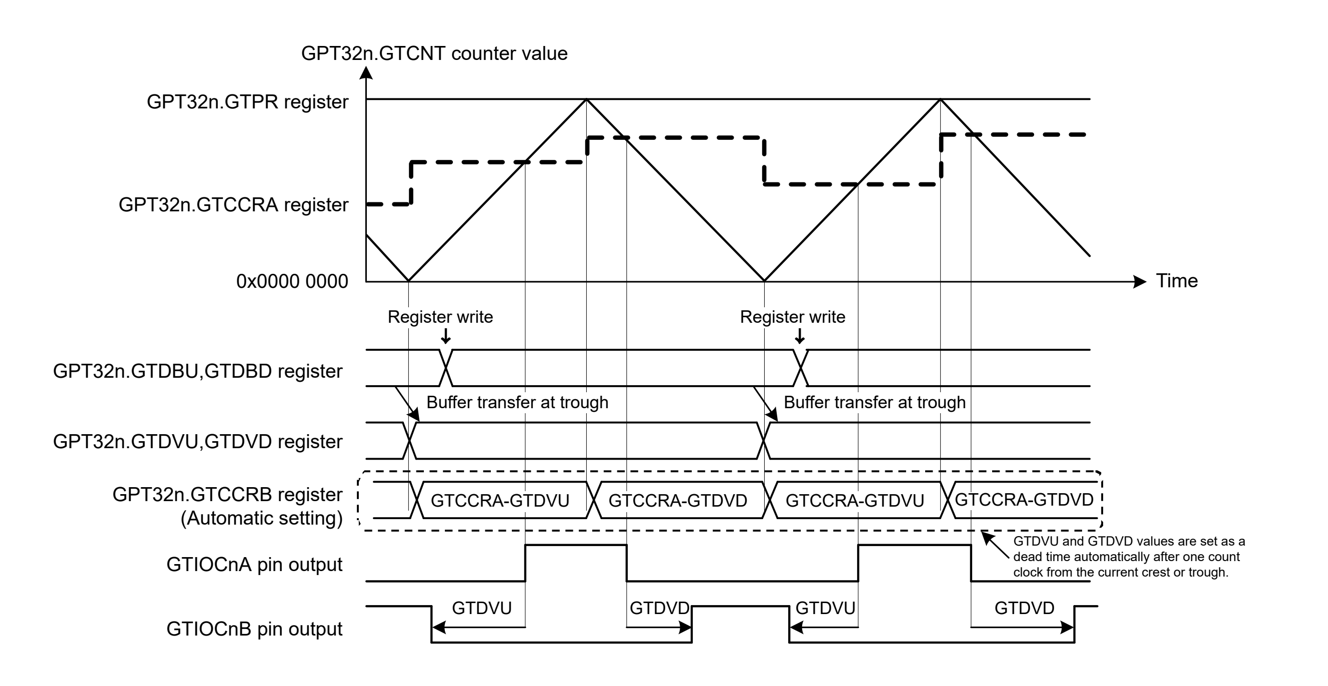


Figure 22.80 Example of automatic compare-match value setting function with dead time in triangle-wave PWM mode 2 or 3, GTDVU and GTDVD set to buffer operation, and active-high

Table 22.42 Example setting for automatic dead time setting function in saw-wave one-shot pulse mode, and triangle-wave PWM mode 3 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0] or GTCR.MD[3:0]. In Figure 22.77 and Figure 22.78, 001b or 0001b (saw-wave one-shot pulse mode) is set. In Figure 22.80, 110b or 0110b (triangle-wave PWM mode 3) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 22.77, 01b is set after 11b is set in GTUDDTYC[1:0] (up count). In Figure 22.78, 00b is set after 10b is set in GTUDDTYC[1:0] (down count).
3	Select count clock	Select the count clock with GTCR.TPCS[3:0].
4	Set cycle	Set the cycle in GTPR.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR. In Figure 22.77, Figure 22.78, and Figure 22.80, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
7	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with OAE and OBE in GTIOR.
8	Set buffer value for compare match*1	Set the GTIOCnA pin transition immediately after the count start in GTCCRC and GTCCRD.
9	Set forcible buffer transfer for compare match*1	Set GTBER.CCRSWT to 1 to transfer buffer register data forcibly to GTCCRA.
10	Set buffer value for compare match	Set the GTIOCnA pin transition in 1 cycle after the current cycle in GTCCRC and GTCCRD.
11	Set automatic dead time setting function	Set GTDTCR.TDE to 1 to enable the automatic dead time setting function.
12	Set buffer operation for dead time setting	Set buffer operation with TDBUE and TDBDE bits in GTDTCR.
13	Set dead time value	Set the first half dead time value in GTDVU and the second half dead time in GTDVD. When GTDVU is set with GTDTCR.TDFER bit set to 1, the same value is also set to GTDVD, the same dead time value can be set for the first and second halves.
14	Set buffer value for dead time	For buffer operation, set the first half dead time in one cycle after the current cycle in GTDBU and the second half dead time in GTDBD.

Table 22.42 Example setting for automatic dead time setting function in saw-wave one-shot pulse mode, and triangle-wave PWM mode 3 (2 of 2)

No.	Step Name	Description
15	Start count operation	Set GTCR.CST to 1 to start count operation.
16	Set buffer value for each cycle	Set the GTIOCnA pin transition in 1 cycle after the current cycle in GTCCRC and GTCCRD. When the dead time register is used for buffer operation, set the dead time value in the first half of the next cycle from the current cycle to GTDBU and the dead time value in the second half to GTDBD.

Note: n: 0 to 13
m: A, B

Note 1. When using PWM Delay Generation circuit, change the order of setting for Enable GTIOCnm pin output and setting for Set buffer value for compare match + Set forcible buffer transfer for compare match.

Table 22.43 Example setting for automatic dead time setting function in triangle-wave PWM mode 1 or 2

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0] or GTCR.MD[3:0]. In Figure 22.79, 100b or 0100b (triangle-wave PWM mode 1) is set. In Figure 22.80, 101b or 0101b (triangle-wave PWM mode 2) is set.
2	Select count clock	Select the count clock with GTCR.TPCS[3:0].
3	Set cycle	Set the cycle in GTPR.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set GTIOCnm pin function	Set the GTIOCnm pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR. In Figure 22.79 and Figure 22.80, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
6	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with OAE and OBE in GTIOR.
7	Set buffer operation for compare match	Set buffer operation with CCRA in GTBER.
8	Set compare match value*1	Set the GTIOCnA pin transition in GTCCRA.
9	Set buffer value for compare match	For buffer operation, set the GTIOCnA pin transition in 1 cycle after the current cycle (in triangle-wave PWM mode 1) or half cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRC. For double buffer operation, also set the GTIOCnA pin transition in 2 cycles after the current cycle (in triangle-wave PWM mode 1) or 1 cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRD.
10	Set automatic dead time setting function	Set GTDTCR.TDE to 1 to enable the automatic dead time setting function.
11	Set buffer operation for dead time setting	Set buffer operation with TDBUE and TDBDE bits in GTDTCR.
12	Set dead time value	Set the first half dead time value in GTDVU and the second half dead time in GTDVD. When GTDVU is set with GTDTCR.TDFER bit set to 1, the same value is also set to GTDVD, the same dead time value can be set for the first and second halves.
13	Set buffer value for dead time	For buffer operation, set the first half dead time in one cycle after the current cycle in GTDBU and the second half dead time in GTDBD.
14	Start count operation	Set GTCR.CST to 1 to start count operation.
15	Set buffer value for each cycle	When the compare match register is used for buffer operation, set the GTIOCnA pin transition in 1 cycle after the current cycle (in triangle-wave PWM mode 1) or half cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRC. When the compare match register is used for double-buffered operation, set the GTIOCnA pin changing point in two cycles after the current cycle (in triangle-wave PWM mode 1) or one cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRD. In the same way, set the dead time value in the first half of the cycle after current cycle in GTDBU and the dead time in the second half in GTDBD.

Note: n: 0 to 13
m: A, B

Note 1. When using PWM Delay Generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

22.3.5 Count Direction Changing Function

The count direction of the GTCNT counter can be changed by modifying the UD bit in GTUDDTYC.

In saw-wave mode, if the UD bit in GTUDDTYC is modified during count operation, the count direction is changed at an overflow (when modified during up-counting) or an underflow (when modified during down-counting). If the GTUDDTYC.UD bit is modified while the count operation stops and the GTUDDTYC.UDF bit is 0, the GTUDDTYC.UD bit modification is not reflected at the start of counting and the count direction is changed at an overflow or an underflow. If the UDF bit is set to 1 while the count operation stops, the GTUDDTYC.UD bit value at that time is reflected at the start of counting.

In triangle-wave mode, the count direction does not change even though the UD bit in GTUDDTYC is modified during the count operation. When the GTUDDTYC.OABDTYT bit is 0, the output duty setting changed when underflow occurs is reflected, and when the GTUDDTYC.OABDTYT bit is 1, the output duty setting changed when underflow occurs and at the top of the triangle-wave is reflected. Similarly, even though the GTUDDTYC.UD bit is modified while the count operation stops and GTUDDTYC.UDF bit is 0, the GTUDDTYC.UD bit value is not reflected to the count operation. If the GTUDDTYC.UDF bit is set to 1 while the count operation is stopped, the GTUDDTYC.UD bit value at that time is reflected at the start of counting.

If the count direction changes during a saw-wave count operation, the GTPR value after the start of up-counting is reflected in the count cycle during up-counting and the GTPR value after the start of down-counting is reflected in the count cycle during down-counting.

Figure 22.81 shows an example of count direction changing function operation.

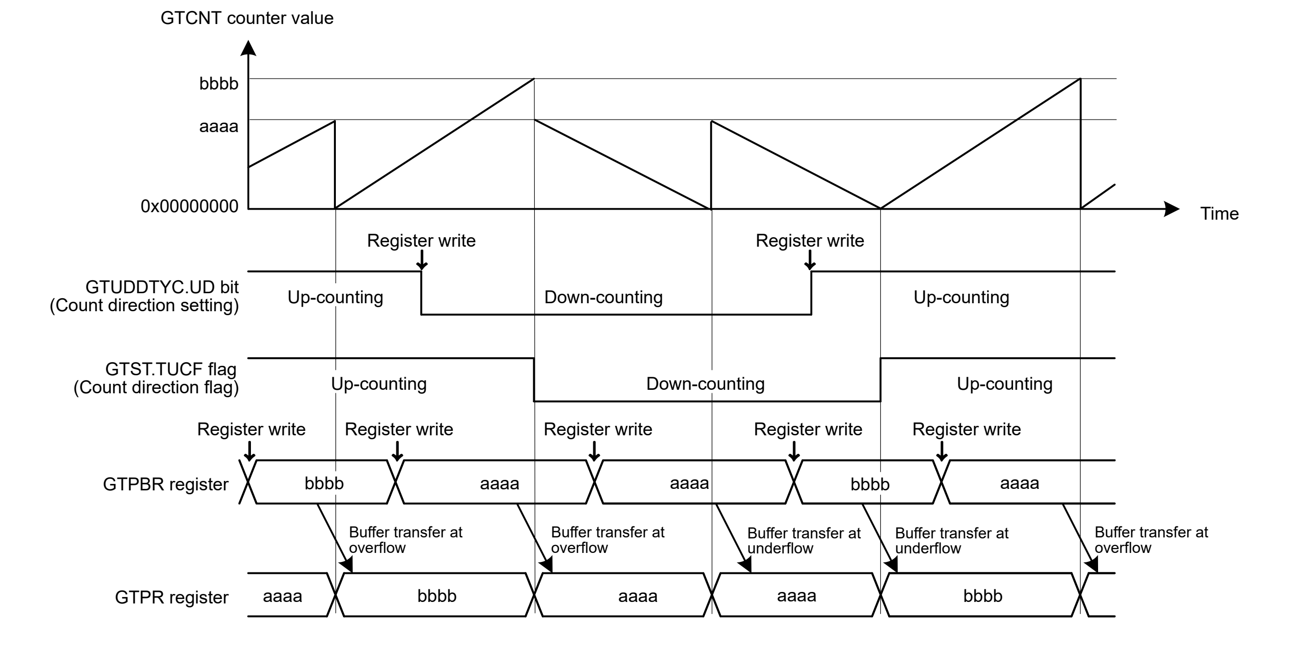


Figure 22.81 Example of a count direction changing function operation during buffer operation

22.3.6 Function of Output Duty 0% and 100%

The output duty of the GTIOCnA pin and the GTIOCnB pin (n = 0 to 13) are set to 0% or 100% by changing the GTUDDTYC.OADTY bit or GTUDDTYC.OBDTY bit.

This function is invalid in saw-wave PWM mode 2 or complementary PWM mode.

In saw-wave mode, if the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified during the count operation, the output duty setting is reflected at an overflow (when modified during up-counting) or an underflow (when modified during down-counting). If the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified while the count operation is stopped and the GTUDDTYC.OADTYF or the GTUDDTYC.OBDTYF bit is 0, the output duty modification is not reflected at the start of counting. The output duty changes at an overflow or an underflow. If the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified while the count operation is stopped and the

GTUDDTYC.OADTYF or the GTUDDTYC.OBDTYF bit is 1, the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit value at that time is reflected at the start of counting.

In triangle-wave mode, if the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified during the count operation, when the GTUDDTYC.OABDTYT bit is 0, the output duty setting changed when underflow occurs is reflected, and when the GTUDDTYC.OABDTYT bit is 1, the output duty setting changed when underflow occurs and at the top of the triangle-wave is reflected.

If the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified while the count operation is stopped and the GTUDDTYC.OADTYF or the GTUDDTYC.OBDTYF bit is 0, the output duty modification is not reflected at the start of counting. The output duty changes at an underflow. If the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified while the count operation stops and the GTUDDTYC.OADTYF or the GTUDDTYC.OBDTYF bit is 1, the output duty modification is reflected at the start of counting.

In performing 0% or 100% duty operation, GPT internally continues to:

- Perform compare match operation
- Set compare match flag
- Output interrupt
- Perform buffer operation.

When the control is changed from 0% or 100% duty setting to compare match, the output value of GTIOCnA pin at cycle end is decided by GTIOR.GTIOA[3:2] and GTUDDTYC.OADTYR. The output value of GTIOCnB pin at cycle end is decided by GTIOR.GTIOB[3:2] and GTUDDTYC.OBDTYR.

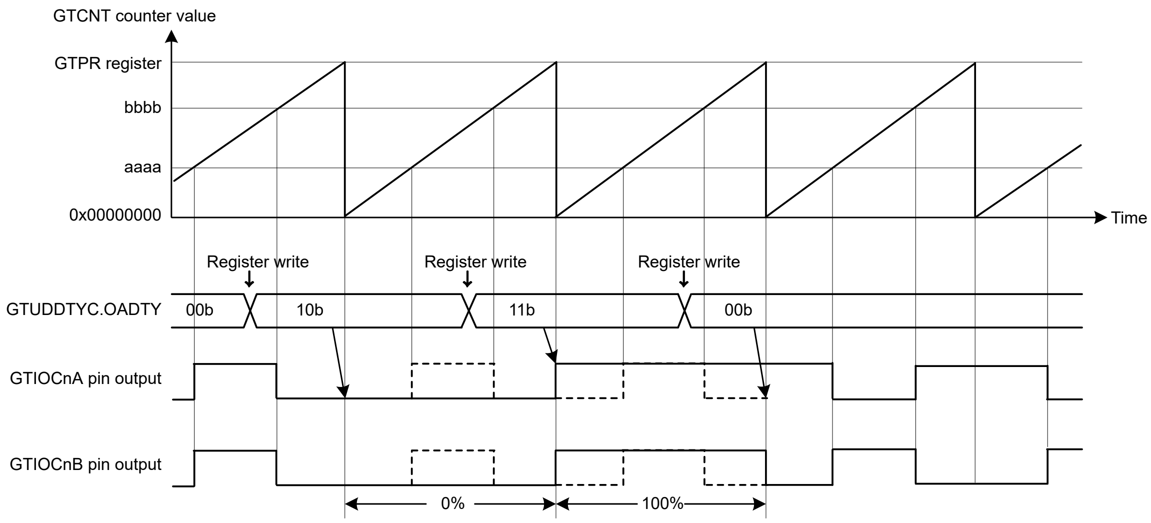
When GTIOR.GTIOA[3:2] and GTIOR.GTIOB[3:2] are set to 01b, the output pins output low at cycle end. When GTIOR.GTIOA[3:2] and GTIOR.GTIOB[3:2] are set to 10b, the output pins output high at cycle end.

GTUDDTYC.OADTYR selects the value that is the object of output retained/toggled at cycle end, when GTIOR.GTIOm[3:2] are set to 00b (output retained at cycle end) or when GTIOR.GTIOm[3:2] are set to 11b (output toggled at cycle end). Table 22.44 shows the values of GTIOCnA and GTIOCnB pin output at cycle end.

Table 22.44 Output values after releasing 0% or 100% duty setting (m = A, B)

GTIOR.GTIOm[3:2]	Compare match value at cycle end masked by 0% or 100% duty setting	GTUDDTYC.OmDTYR in duty 0% setting		GTUDDTYC.OmDTYR in duty 100% setting	
		0	1	0	1
00 (output retained at cycle end)	0	0	0	1	0
	1	0	1	1	1
01 (low output at cycle end)	—	0	0	0	0
10 (high output at cycle end)	—	1	1	1	1
11 (output toggled at cycle end)	0	1	1	0	1
	1	1	0	0	0

Figure 22.82 shows an example of output duty 0% and 100% function.



Note: The base address of registers are the same.

Note: [Setting examples]

GTIOR.GTIOA[4:0] bits:	00011b Initial low output, output toggled at compare match, output retained at cycle end
GTUDDTYC.OADTYR bit:	0b The function selected by GTIOA[3:2] is applied to the output value when the duty cycle is set after release from 0 or 100% duty-cycle setting.
GTIOR.GTIOB[4:0] bits:	00011b Initial low output, output toggled at compare match, output retained at cycle end
GTUDDTYC.OBDTYR bit:	1 The function of GTIOB[3:2] is applied to the compare match output value which is masked after 0 or 100% duty setting release.

Figure 22.82 Example of output duty 0% and 100% function

22.3.7 Hardware Count Start/Count Stop and Clear Operation

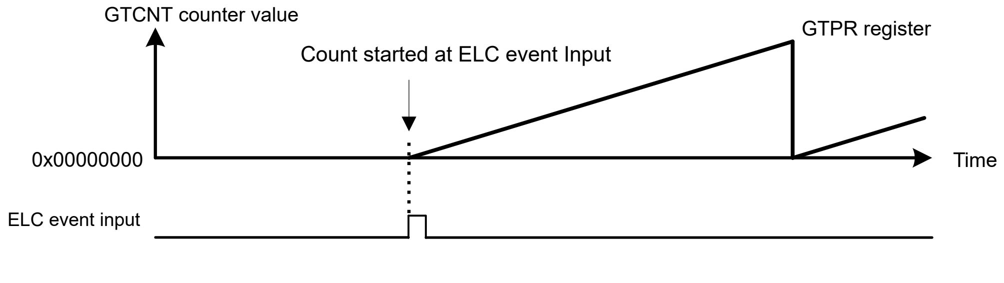
The GTCNT counter can be started, stopped, or cleared by the following hardware sources:

- External trigger input
- ELC event input
- GTIOCnA and GTIOCnB pin input (n = 0 to 13).

22.3.7.1 Hardware Start Operation

The GTCNT counter can be started by selecting a hardware source using GTSSR.

Figure 22.83 shows an example of a count start operation by a hardware source. Table 22.45 shows the setting example.



Note: ELC event input: ELCA event input

Figure 22.83 Example of count start operation by a hardware source started at the input of the signal from the ELCA event

Table 22.45 Example setting for count start operation by a hardware source

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 22.83, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 22.83, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 22.83, 0x00000000 is set.
6	Set hardware count start	Select a hardware source for starting count operation in the GTSSR register. In Figure 22.83, GTSSR.SSELCA = 1
7	Set hardware source operation	Set operation of the hardware source selected by the GTSSR register and start counting. In Figure 22.83, the ELCA event input operation is set.

Figure 22.84 shows an example of the timing of operations to start counting in response to ELC_GPTA event input.

This is an example of operations to start counting by the GPT321.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPT320.GTCCRA register. This is selected as a trigger for output to the GPT321 by the ELC as ELC_GPTA.

ELC passes the event signal output from GPT320 without delay to GPT321.

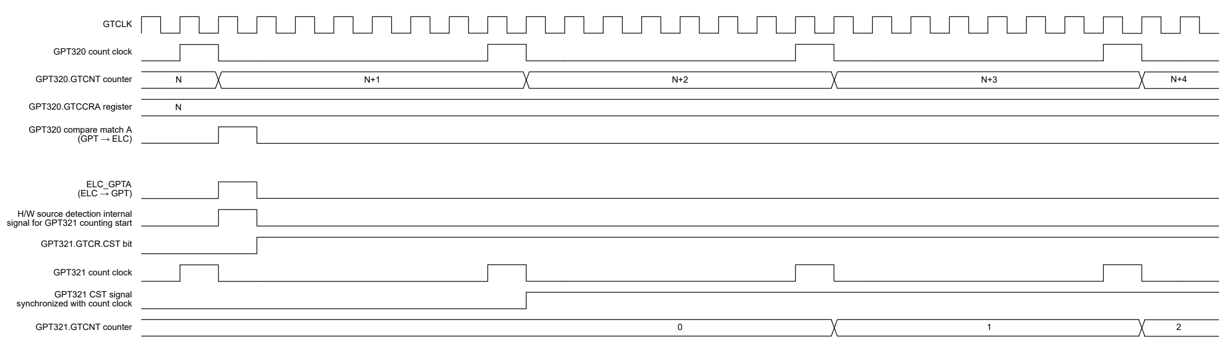


Figure 22.84 Example of Timing of Operations to Start Counting in Response to Event Input from ELC_GPTA

22.3.7.2 Hardware Stop Operation

The GTCNT counter can be stopped by selecting a hardware source using GTPSR.

Figure 22.85 shows an example of a count stop operation by a hardware source. Table 22.46 shows the setting example. In this example, the count operation stops at the ELCA event input and restarts at the ELCB event input.

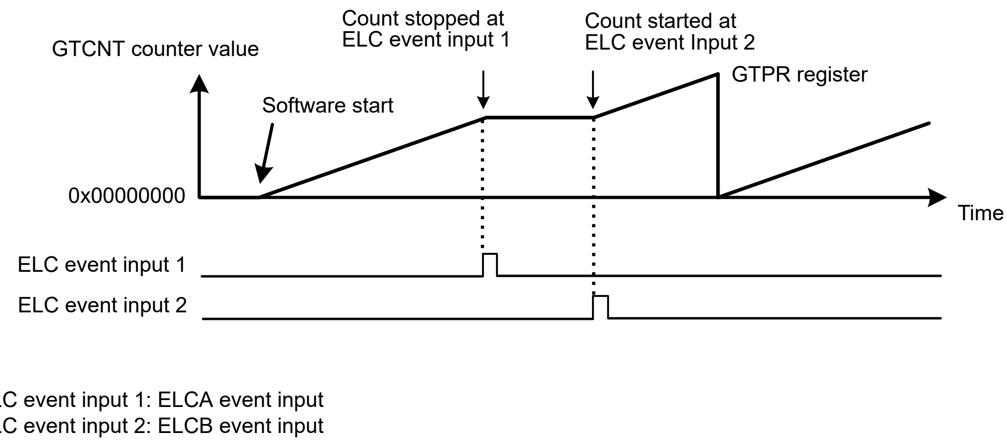


Figure 22.85 Example of count stop operation by hardware source started by software, stopped at ELCA input, and restarted at ELCB input

Table 22.46 Example setting for count stop operation by a hardware source

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0] or GTCR.MD[3:0] bits. In Figure 22.85, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 22.85, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).
3	Select count clock	Select the count clock with GTCR.TPCS[3:0].
4	Set cycle	Set the cycle in GTPR.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 22.85, 0x00000000 is set.
6	Set hardware count start	Select a hardware source for starting count operation in GTSSR register, and wait for count start by the hardware source. In Figure 22.85, GTSSR.SSELCB = 1.
7	Set hardware count stop	Select a hardware source for stopping count operation in GTPSR register and wait for count stop by the hardware source. In Figure 22.85, GTPSR.PSELCA = 1.
8	Set hardware source operation	Set operation of the hardware source selected in GTSSR register or GTPSR register, and start or stop counting. In Figure 22.85, ELCA input operation and ELCB input operation are set.

Figure 22.86 shows an example of a count start/stop operation by a hardware source. Table 22.47 shows the setting example. In this example, the counter operates during the high-level periods of the external trigger input GTETRGA.

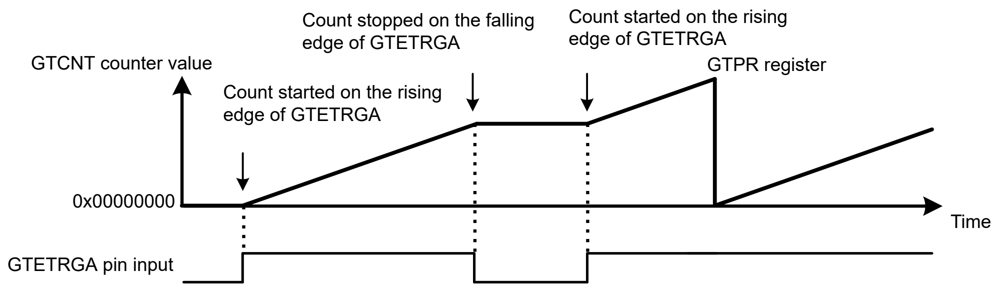


Figure 22.86 Example of count start/stop operation by a hardware source started on the rising edge of GTETRGA pin input, and stopped on the falling edge of GTETRGA pin input

Table 22.47 Example setting for count start/stop operation by a hardware source

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 22.86, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 22.86, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 22.86, 0x00000000 is set.
6	Set hardware count start	Select a hardware source for starting count operation with the GTSSR register, and wait for count start by the hardware source. In Figure 22.86, GTSSR.SSGTRGAR = 1.
7	Set hardware count stop	Select a hardware source for stopping count operation with the GTPSR register, and wait for count stop by the hardware source. In Figure 22.86, GTPSR.PSGTRGAF = 1.
8	Set hardware source operation	Set operation of the hardware source selected in the GTSSR register or GTPSR register and start or stop counting. In Figure 22.86, the GTETRGA pin operation is set.

Figure 22.87 shows an example of timing of operations to stop counting in response to event input from ELC_GPTA.

This is an example of operations to stop counting by the GPT321.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPT320.GTCCRA register. This is selected as a trigger for output to the GPT321 by the ELC as ELC_GPTA.

ELC passes the event signal output from GPT320 without delay to GPT321.

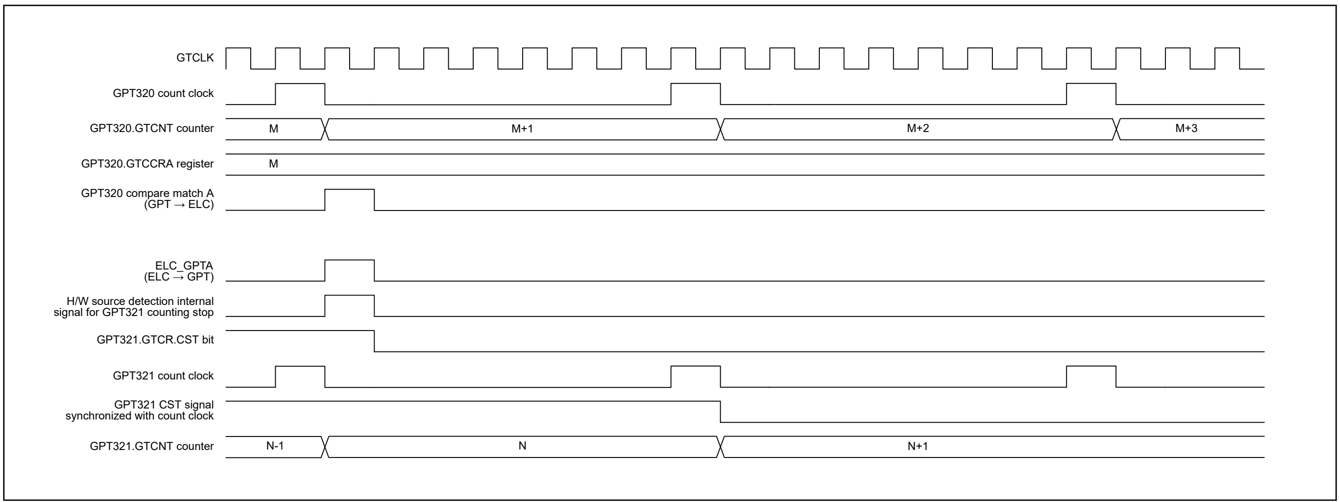


Figure 22.87 Example of Timing of Operations to Stop Counting in Response to Event Input from ELC_GPTA

22.3.7.3 Hardware Clear Operation

The GTCNT counter can be cleared by selecting a hardware source using GTCSR. The GPTn_OVF/GPTn_UDF (n = 0 to 13) interrupt (overflow/underflow interrupt) is not generated when the GTCNT counter is cleared by a hardware source or by software.

Figure 22.88 and Figure 22.89 show examples of the GTCNT counter clearing operation by a hardware source. Table 22.48 shows the setting example. In this example, the GTCNT counter starts at the ELCA input, and the counter stops and clears at the ELCB input.

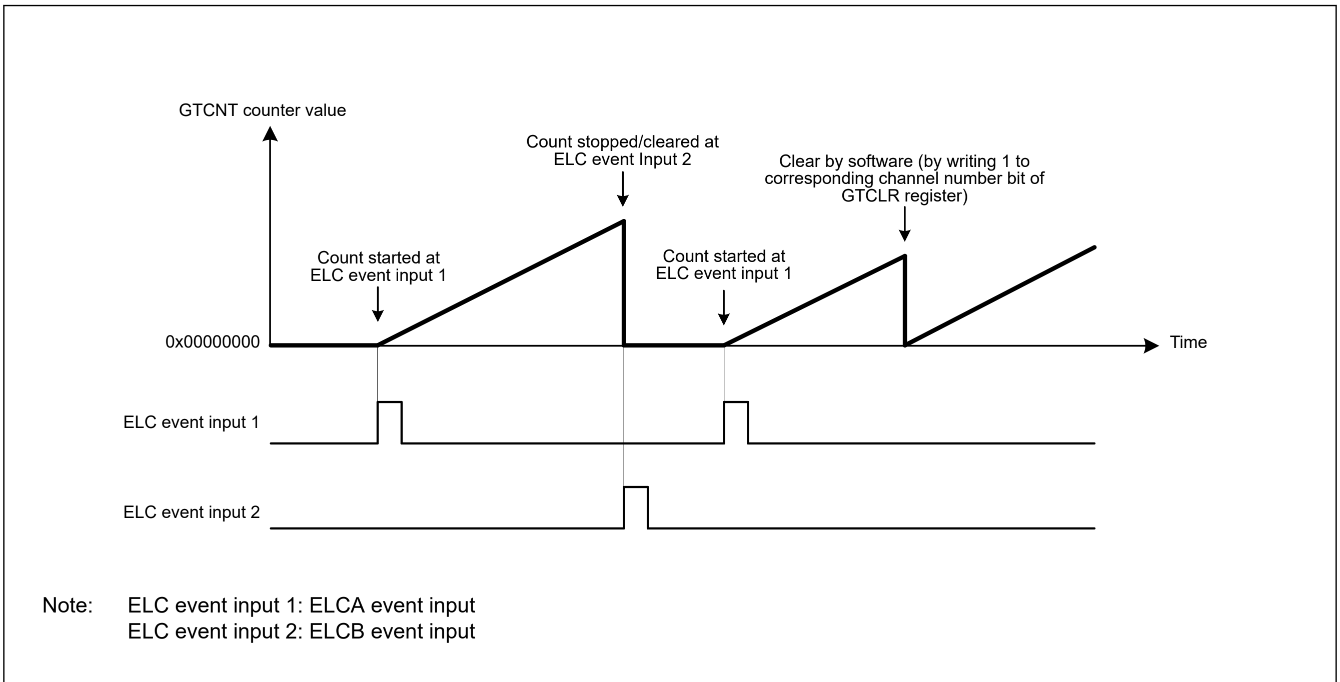
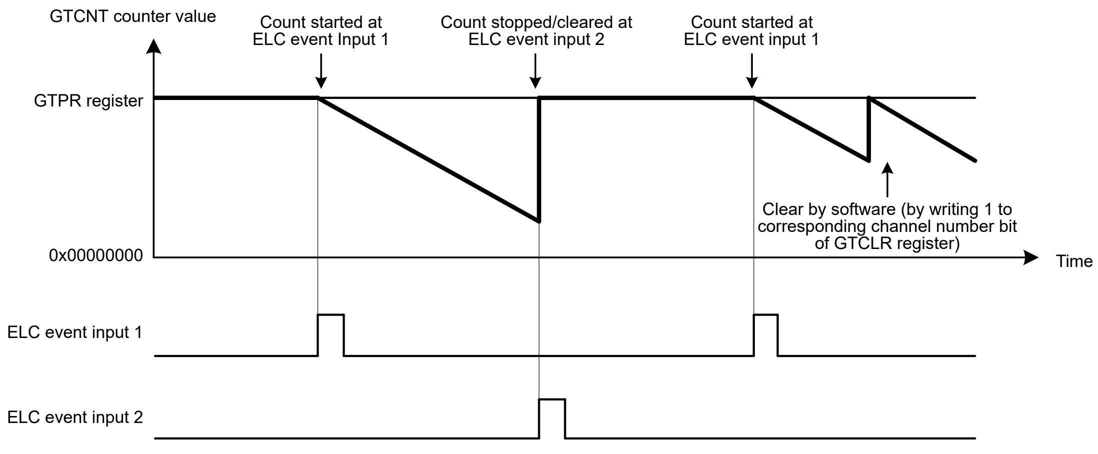


Figure 22.88 Examples of count clearing operation by hardware source in saw wave up-counting, started at ELCA input, and stopped/cleared at ELCB input



Note: ELC event input 1: ELCA event input
ELC event input 2: ELCB event input

Figure 22.89 Examples of count clearing operation by hardware source in saw wave down-counting, started at ELCA input, and stopped/cleared at ELCB input

Table 22.48 Example setting for count clearing operation by a hardware source

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 22.88 and Figure 22.89, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 22.88, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting). In Figure 22.89, after 10b is set in the GTUDDTYC[1:0] bits, 00b is set in the GTUDDTYC[1:0] bits (down-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 22.88, 0x00000000 is set. In Figure 22.89, the GTPR register value is set.
6	Set hardware count start	Select a hardware source for starting count operation in the GTSSR register, and wait for count start by the hardware source. In Figure 22.88 and Figure 22.89, GTSSR.SSELCA = 1.
7	Set hardware count stop	Select a hardware source for stopping count operation in the GTPSR register, and wait for count stop by the hardware source. In Figure 22.88 and Figure 22.89, GTPSR.PSELCB = 1.
8	Set hardware count clear	Select a hardware source for clearing count operation in the GTCSR register, and wait for count clear by the hardware source. In Figure 22.88 and Figure 22.89, GTCSR.CSELCB = 1.
9	Set hardware source operation	Set operation of the hardware source selected in the GTSSR register, GTPSR register or GTCSR register and start, stop or clear counting. In Figure 22.88 and Figure 22.89, the ELCA input and ELCB input are set.

The GPTn_OVF/GPTn_UDF (n = 0 to 13) interrupt (overflow/underflow interrupt) is not generated when the counter is cleared by a hardware source or by software.

Figure 22.90 shows the relationship between the counter clearing by a hardware source and the GPTn_OVF (n = 0 to 13) interrupt.

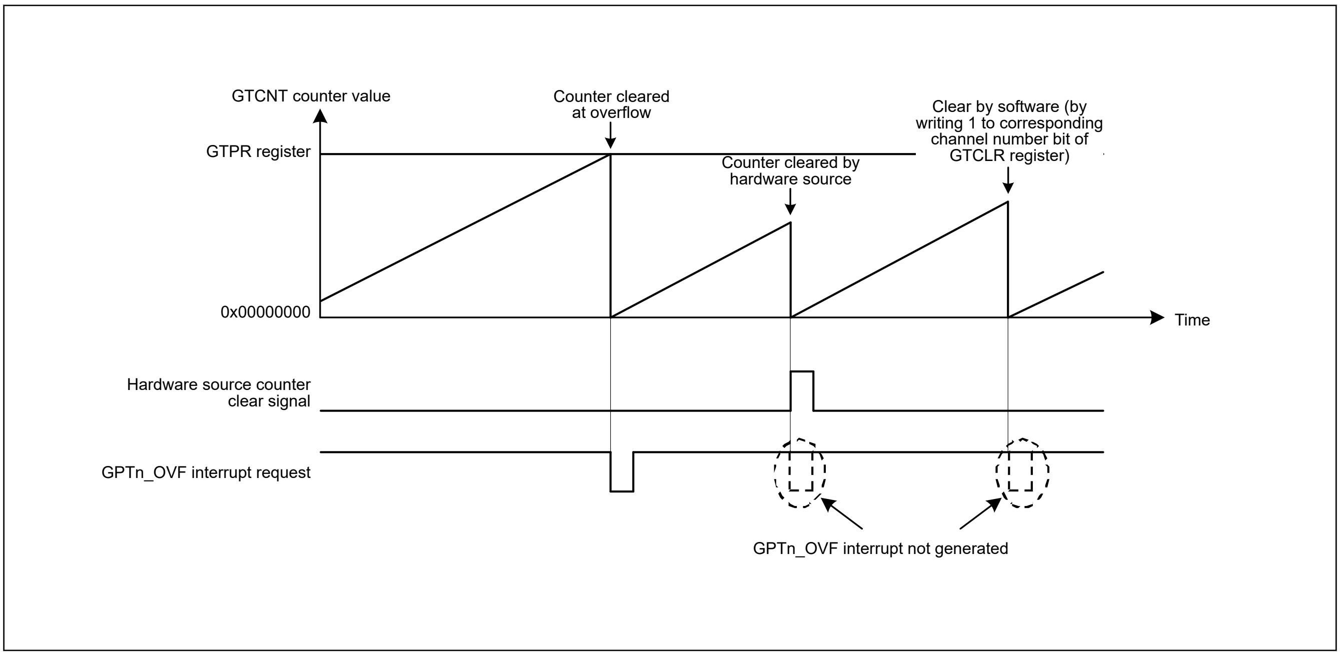


Figure 22.90 Relationship between counter clearing by hardware source and GPTn_OVF (n = 0 to 13) interrupt

Figure 22.91 shows an example of the timing of operations to clear the counter in response to event input from ELC_GPTA when a clock signal produced by frequency-dividing the GTCLK signal is used as the counter clock for the GTCNT counter.

This is an example of operations to clear counting by the GPT321.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPT320.GTCCRA register. This is selected as a trigger for output to the GPT321 by the ELC as ELC_GPTA.

ELC passes the event signal output from GPT320 without delay to GPT321.

The GTCNT counter is cleared when counting is in progress after the GPT32 has detected the internal clearing signal.

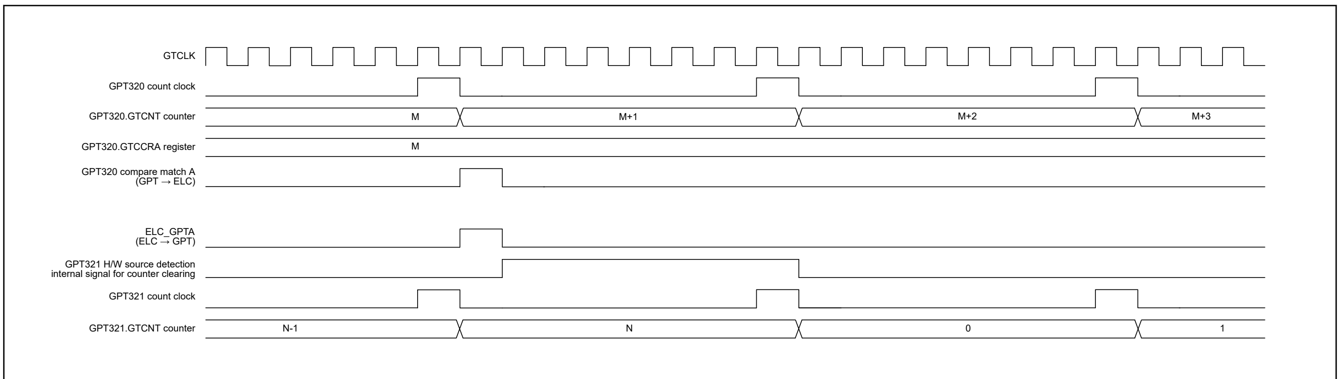


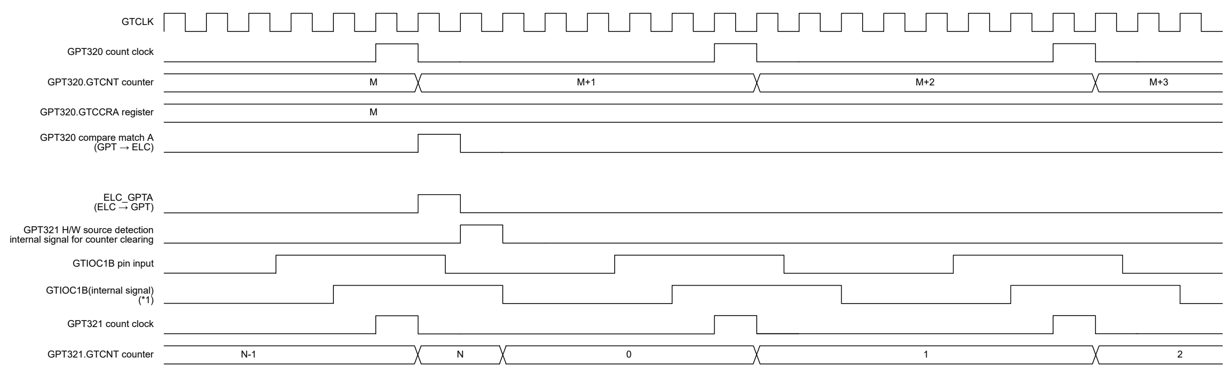
Figure 22.91 Examples of Timing of Operations for Counter Clearing in Response to Event Input from the ELC_GPTA (During the Counting of Cycles of Clock Signal Produced by Dividing the GTCLK Frequency)

Figure 22.92 shows an example of the timing of operations for clearing the counter in response to the input of an event signal from the ELC_GPTA when counting is triggered by a hardware source.

This is an example of operations to clear counting by the GPT321.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPT320.GTCCRA register. This is selected as a trigger for output to the GPT321 by the ELC as ELC_GPTA.

ELC passes the event signal output from GPT320 without delay to GPT321.

The GTCNT counter is cleared in synchronization with GTCLK after the GPT32 has detected the internal clearing signal.



Note 1. This is the case when the digital noise filter is not in use.

Figure 22.92 Examples of the Timing of Operations for Counter Clearing in Response to Event Input from the ELC_GPTA (During Counting Triggered by Hardware Source)

22.3.8 Synchronized Operation

Synchronized operation on channels such as a synchronized start, stop, and clear operation can be performed.

22.3.8.1 Synchronized Operation by Software

The GTCNT counters can be started, stopped, and cleared on multiple channels by setting the associated GTSTR, GTSTP, or GTCLR bits simultaneously to 1.

Count start with a phase difference is possible by setting the initial value in the GTCNT counter and setting the associated GTSTR bits simultaneously to 1.

When the GTCNT synchronous set/clear function is enabled on the GTCR.SSCEN bit, the GTCNT registers of the channel set in same group on the GTCR.SSCGRP [1:0] bits can be written at the same time.

The synchronous GTCNT write is invalid in complementary PWM mode.

When either the SSCE bit or the SSCD bit of the GTSECR register is set to 1, the GTCR.SSCEN bits of the channels that selected on the GTSECSR register are set to 0 or 1 and the GTCNT synchronous set/clear function of multiple channels are enabled or disabled at the same time.

Because the clock of count operation is selected by GTCR.TPCS[3:0] bits in respective channels, if the clock period of each channel that performs synchronous operation (count start/stop/clear) is different from others, the synchronous operation timings of every channel are not exact same.

Figure 22.93 shows an example of a simultaneous start, stop, and clear by software. Figure 22.94 shows an example of phase start operation by software. Figure 22.95, Figure 22.96, Figure 22.97 show an example of simultaneous start/stop/clearing with different count period.

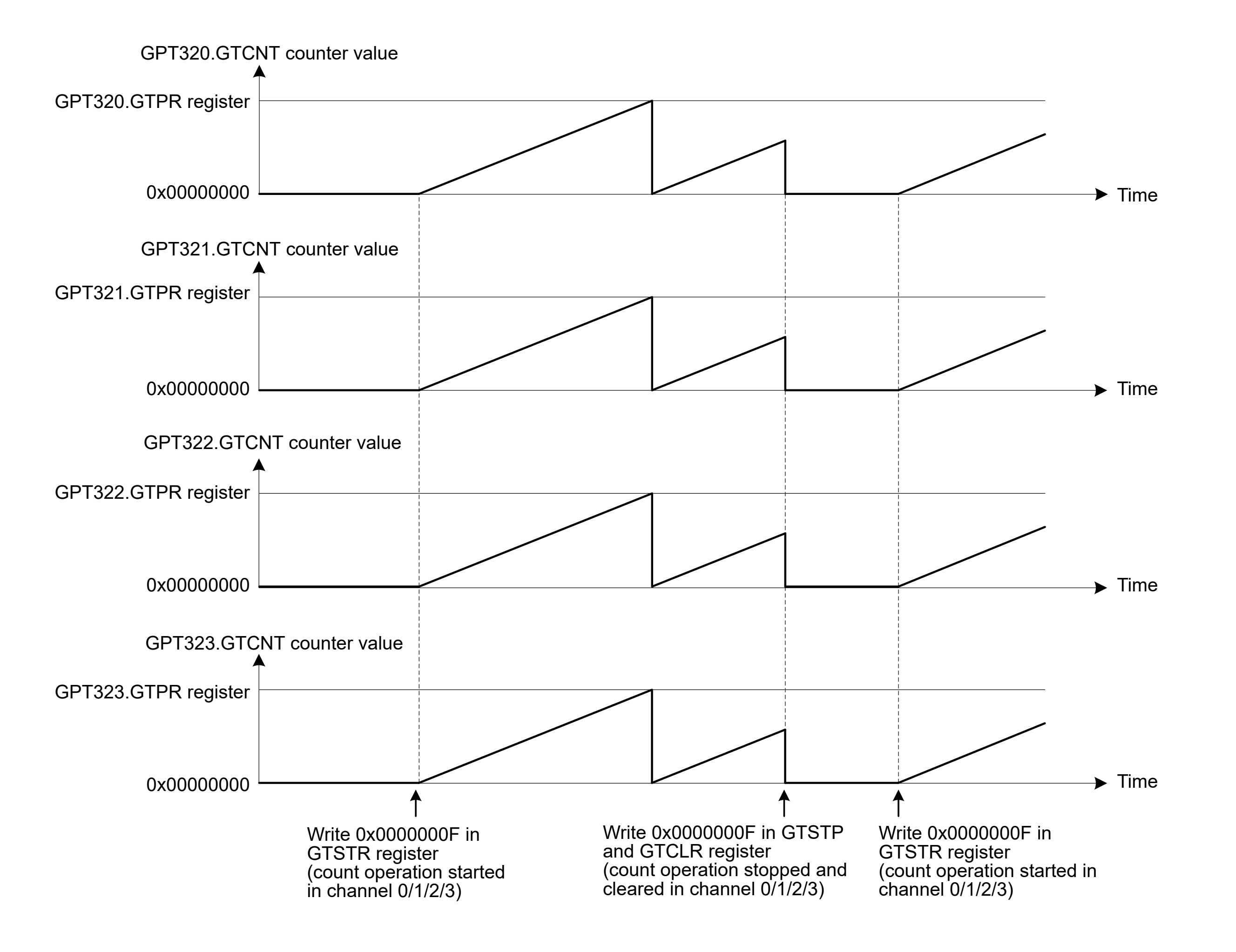


Figure 22.93 Example of a simultaneous start, stop, and clear by software with the same count cycle (GTPR register value)

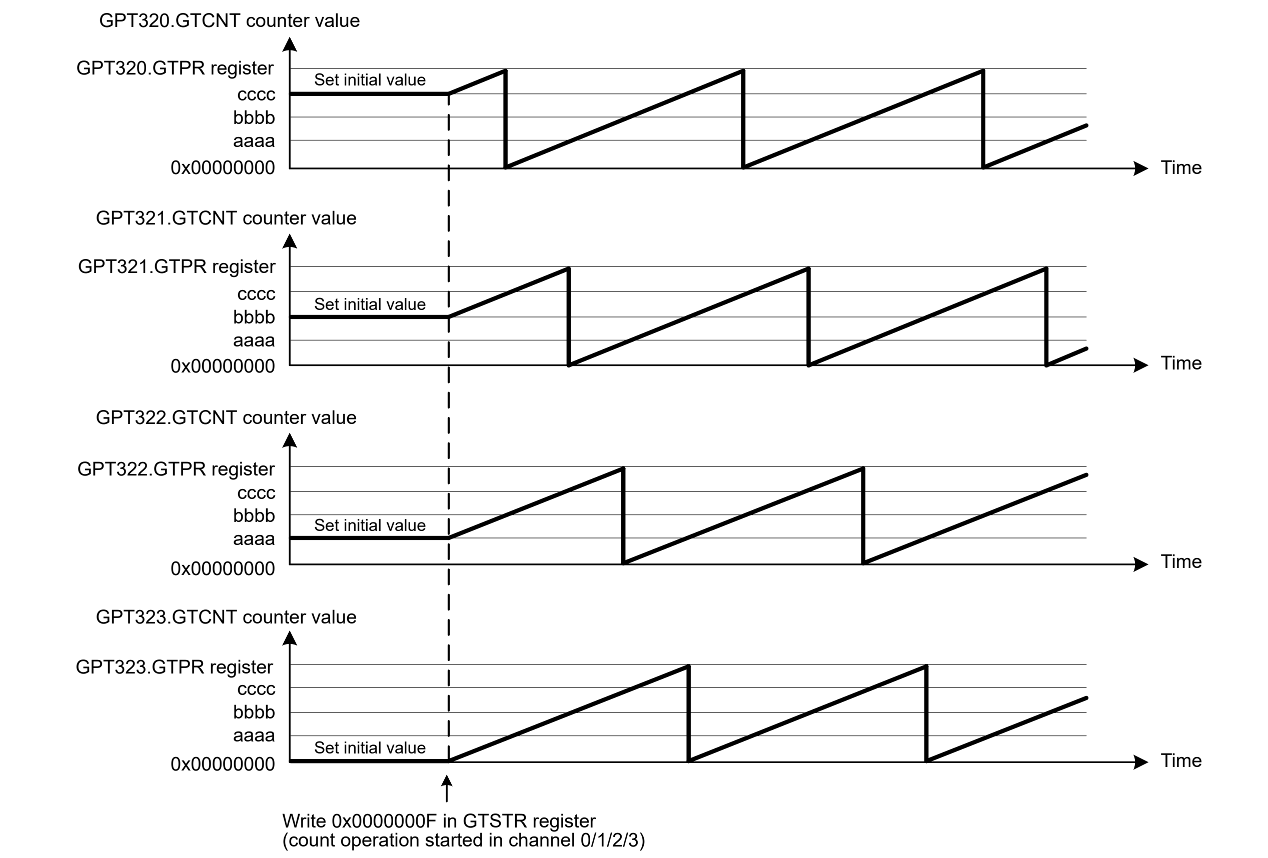


Figure 22.94 Example of software phase start with the same count cycle (GTPR register value)

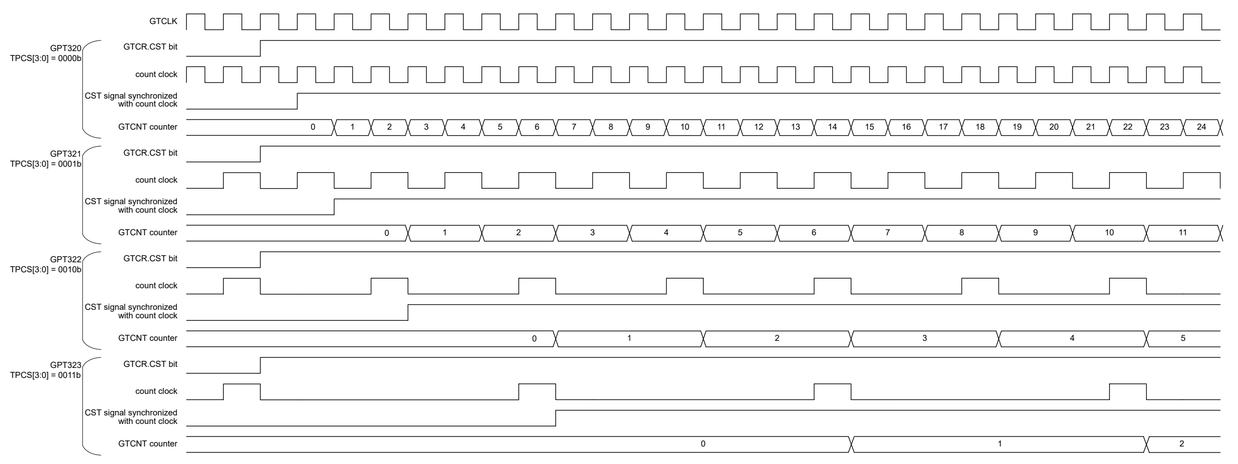


Figure 22.95 Example of Simultaneous Start Operation by Software (with Different Count Period)

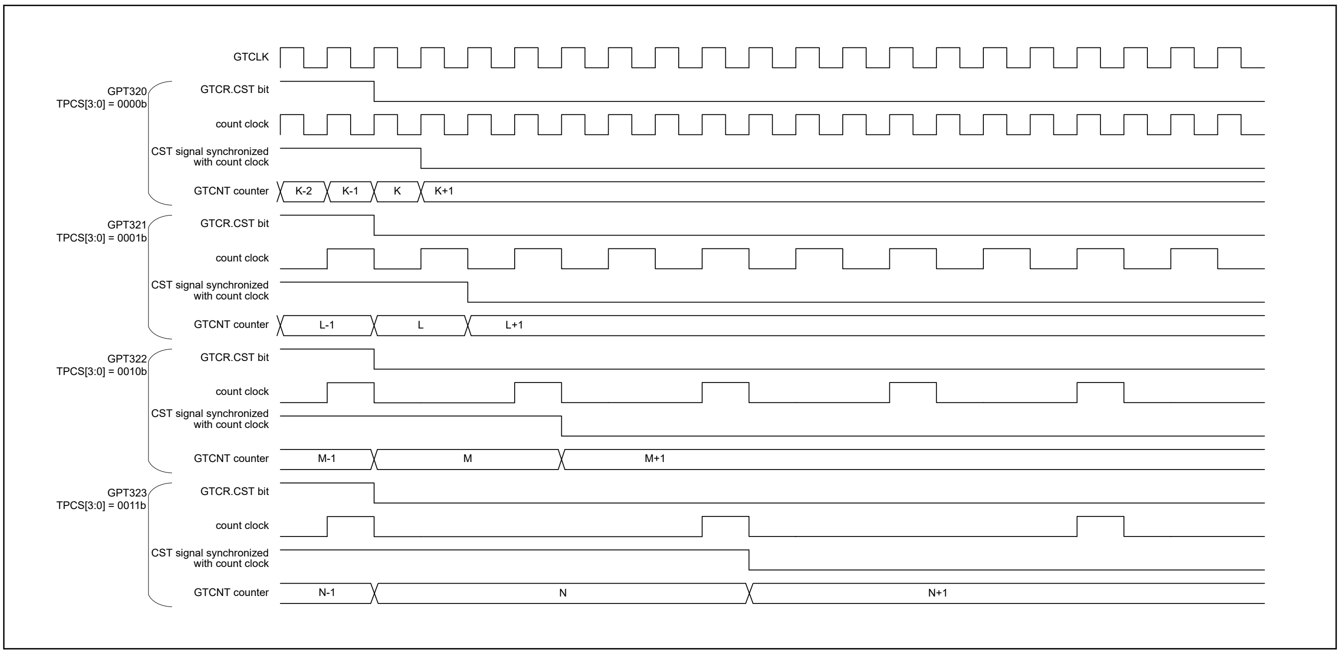


Figure 22.96 Example of Simultaneous Stop Operation by Software (with Different Count Period)

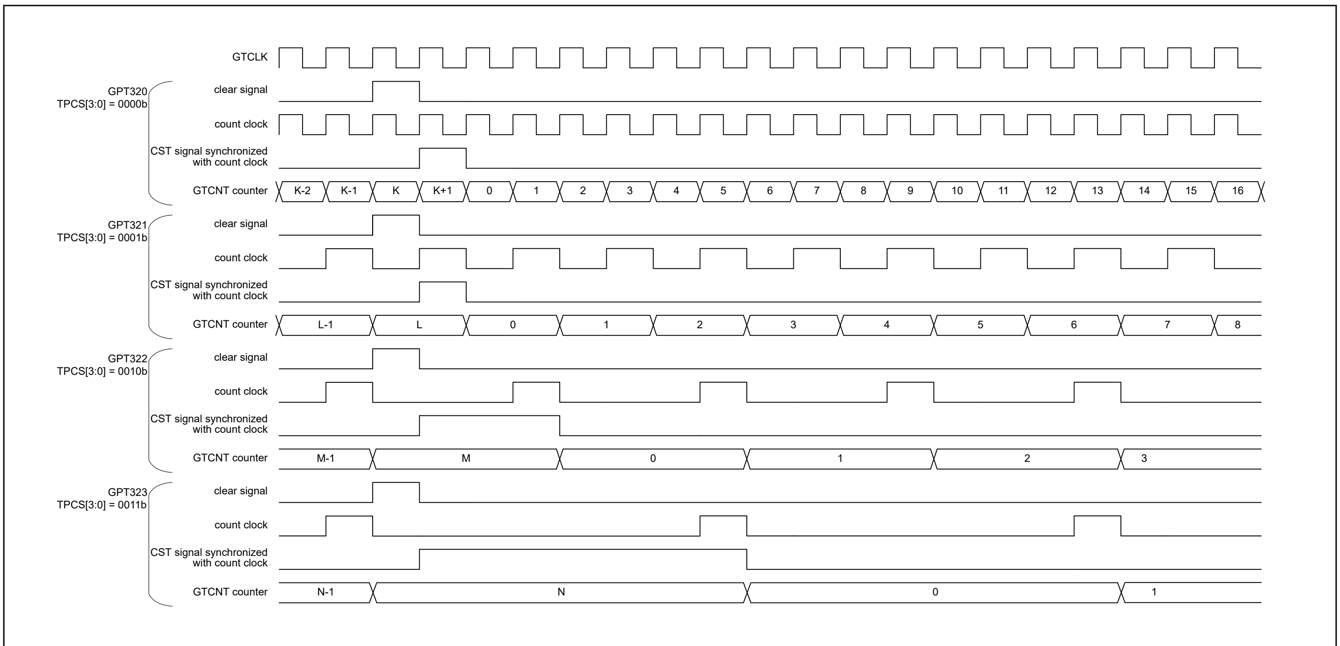
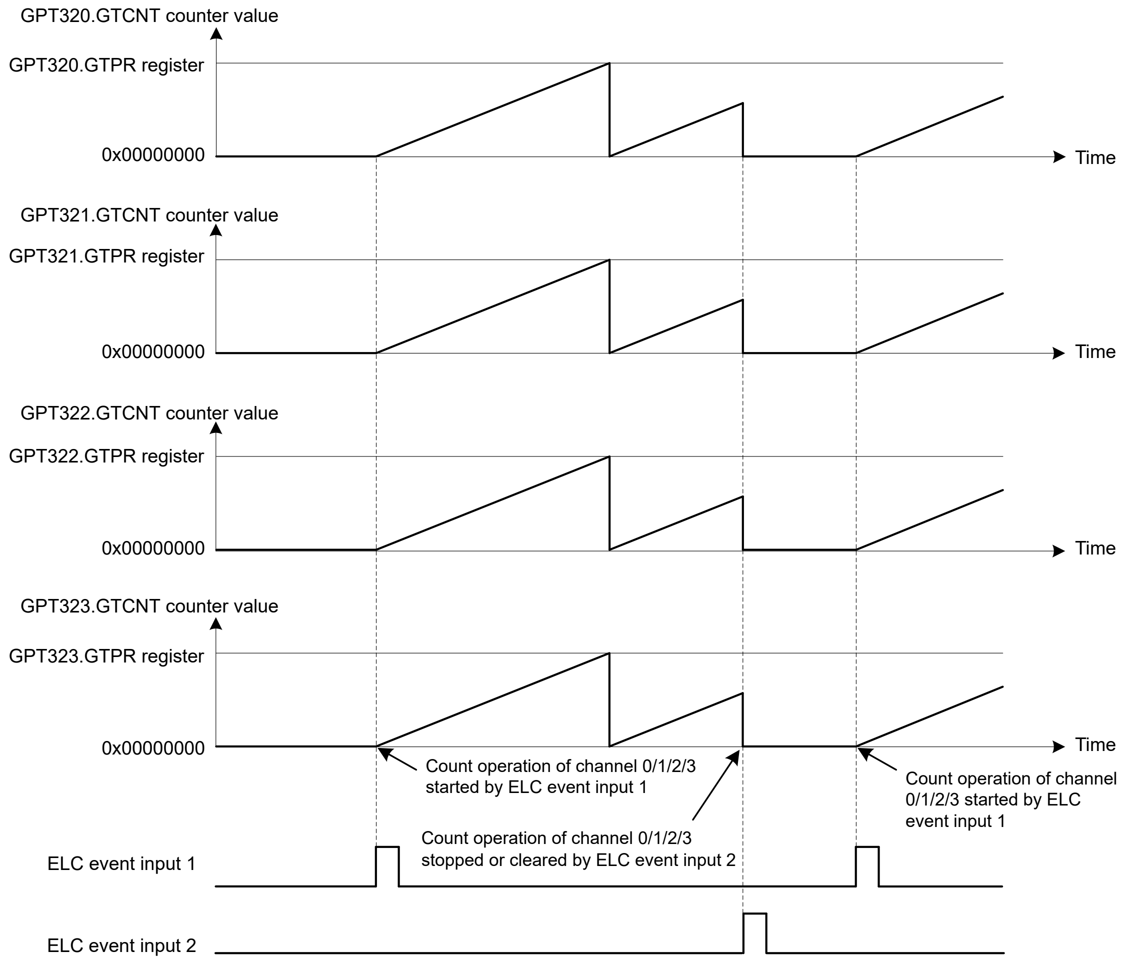


Figure 22.97 Example of Simultaneous Clearing Operation by Software (with Different Count Period)

22.3.8.2 Synchronized Operation by Hardware

The counters for multiple channels can be started, stopped, and cleared simultaneously by ELC event input.

Figure 22.98 shows an example of a simultaneous start, stop, and clear operation by a hardware source. Table 22.49 shows the setting example.



Note: ELC event input 1: ELCA event input
ELC event input 2: ELCB event input

Figure 22.98 Example of a simultaneous start, stop, and clear by a hardware source with the same count cycle (GTPR register value)

Table 22.49 Example setting for simultaneous start by a hardware source (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 22.98, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 22.98, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 22.98, 0x00000000 is set.
6	Set hardware count start	Select a hardware source for starting count operation with the GTSSR register, and wait for count start by the hardware source. In Figure 22.98, GTSSR.SSELCA = 1.

Table 22.49 Example setting for simultaneous start by a hardware source (2 of 2)

No.	Step Name	Description
7	Set hardware count stop	Select a hardware source for stopping count operation with the GTPSR register, and wait for count stop by the hardware source. In Figure 22.98, GTPSR.PSELCB = 1.
8	Set hardware count clear	Select a hardware source for clearing count operation with the GTCSR register, and wait for count clear by the hardware source. In Figure 22.98, GTCSR.CSELCB = 1.
9	Set hardware source operation	Set operation of the hardware source selected in the GTSSR, GTPSR, or GTCSR registers, and start, stop, or clear counting. In Figure 22.98, ELCA input and ELCB input are set.

22.3.8.3 Input Capture Operation by Inter Channel Cooperation

The events of compare match, input capture, sawtooth wave up-count overflow, sawtooth wave down-count underflow, the crest and trough of triangle wave and complementary PWM mode and the count clock can be used as the input capture factor for the GTCCRm register (m = A or B) of other channels. The input capture factors by inter channel cooperation can be set by the GTICCR register of the channel that generates the input capture factor. Input capture by other channel factors can be enabled by GTICmSR.mSOC bit (m = A or B) in the captured channels. The channels that generate input capture factors and the channels that are captured are set to the same input capture group by GTICCR.ICmGRP[1:0] bits (m = A or B).

Figure 22.99 shows an example of input capture operation by inter channel cooperation and Table 22.50 shows an example for setting input capture operation by inter channel cooperation.

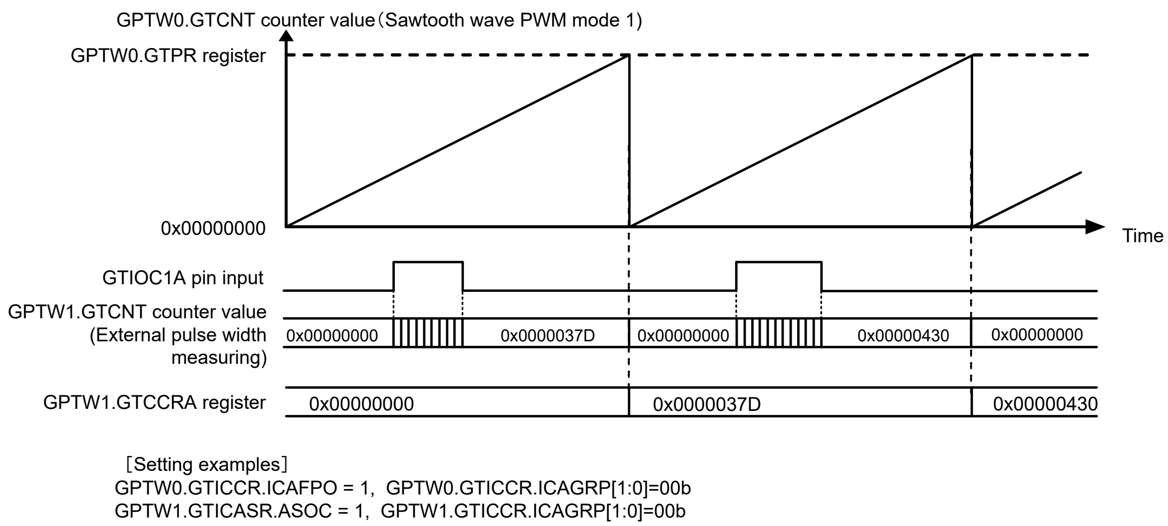


Figure 22.99 Example of input capture operation by inter channel cooperation (channel 1 is captured by the overflow of channel 0)

Table 22.50 Example for setting input capture operation by inter channel cooperation (1 of 2)

Step name	Description
(1) Set operating mode	Set the operating mode with the GTCR.MD[2:0] or the GTCR.MD [3:0] bits. In Figure 22.99,GTCR.MD[2:0] = 000b or GTCR.MD[3:0] = 0000b (sawtooth-wave PWM mode 1) is set to GPTW0.

Table 22.50 Example for setting input capture operation by inter channel cooperation (2 of 2)

Step name	Description
(2) Set external pulse width measuring function	Enable external pulse width measuring function with the GTUPSR.USILVL [3: 0] bits for up-counting operation and the GTDNSR.DSILVL [3: 0] bits for down-counting operation and select the input pin and level to measure. In Figure 22.99, GTUPSR.USILVL[3:0] = 0011b (count up when GTIOCnA pin is 1).
(3) Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 22.99, for GPTW0, lower 2 bits of the GTUDDTYC register is set to 11b, and then lower 2 bits of the GTUDDTYC register is set to 01b (up-counting).
(4) Select count clock	Select the count clock with GTCR.TPCS[2:0] bits or GTCR.TPCS [3:0] bits of the corresponding channel.
(5) Period setting	Set a period in the GTPR register of the corresponding channel.
(6) Set initial value for counter	Set the initial value in the GTCNT counter of the corresponding channel.
(7) Set counter clear factor to the inter channel cooperation Input capture (cleared channels)	Set the GTCSR.CSCMSC[2:0] bits in the channel to be cleared by input capture by inter channel cooperation to 001b, 010b, and select input capture as counter clear factor. In Figure 22.99, GPTW1.GTCSR.CSCMSC[2:0] bits are 001b.
(8) Inter channel cooperation input capture setting (source channel)	Set the input capture factor using the GTICCR register in the source channels of inter channel cooperation input capture. In Figure 22.99, GPTW0.GTICCR.ICAFPO bit is 1b.
(9) Inter channel cooperation input capture setting (captured channels)	Set the GTICmSR.mSOC bit (m = A or B) in the captured channels to allow input capture to the GTCCRm register (m = A or B) by other channel factors. In Figure 22.99, GPTW1.GTICASR.ASOC bit is 1b.
(10) Set group of inter channel cooperation input capture	Set the same value to the GTICCR.ICmGRP[1:0] bits (m = A or B) in the source channels and cleared channels of inter channel cooperation input capture and set them in the same input capture group.

22.3.8.4 Synchronous Clear Operation by Inter Channel Cooperation

The counters of other channels can be cleared at the same time with the counter clear of some channel caused by the compare match, input capture, saw wave up-count overflow, saw wave down-count underflow, and the GTIOCnA / GTIOCnB pin input selected in the GTCSR register.

The counter clear caused by GTCLR register, external trigger input and the ELC input can perform as synchronous clear when the same counter clear factor is set to target channels of synchronous clear. Therefore, these factors are not prepared for the synchronous clear operation by inter channel cooperation.

Set the channel that generates the synchronization clear source and the channel that is synchronized clear in the same synchronous set/clear group with the GTCR.SSCGRP[1:0] bits.

Similar to the synchronous operation in section 22.3.8.1. Synchronized Operation by Software, if the count clock selected by the GTCR.TPCS [3:0] bits is different for each channel, the synchronous clear operation cannot be performed at exact same timing. Similar to the example of simultaneous clearing operation by software in Figure 22.97, if the count clock is different for each channel, the synchronous clear factor is synchronized with the count clock of each channel before counter clearing.

When either the SSCE bit or the SSCD bit of the GTSECR register is set to 1, the GTCR.SSCEN bits of the channels that selected on the GTSECSR register are set to 0 or 1 and the GTCNT synchronous set/clear function of multiple channels are enabled or disabled at the same time.

Table 22.51 shows synchronous clear factor register setting and Figure 22.100 shows an example of synchronous clear operation by inter channel cooperation and Table 22.52 shows an example for setting synchronous clear operation by inter channel cooperation.

Table 22.51 Synchronous clear factor register setting

Synchronous clear factor by inter channel cooperation	Setting register	Setting value
GTCCRA register compare match/input capture	GTINTAD.SCFA	1b
	GTCSR.CSCMSC[2:0]	001b
GTCCRB register compare match/input capture	GTINTAD.SCFB	1b
	GTCSR.CSCMSC[2:0]	010b
GTCCRC register compare match	GTINTAD.SCFC	1b
	GTCSR.CSCMSC[2:0]	011b
GTCCRD register compare match	GTINTAD.SCFD	1b
	GTCSR.CSCMSC[2:0]	100b
GTCCRE register compare match	GTINTAD.SCFE	1b
	GTCSR.CSCMSC[2:0]	101b
GTCCRF register compare match	GTINTAD.SCFF	1b
	GTCSR.CSCMSC[2:0]	110b
Overflow of Sawtooth-Wave up-count	GTINTAD.SCFPO	1b
Underflow of Sawtooth-Wave down-count	GTINTAD.SCFPU	1b
Clear caused by GTIOCnA/GTIOCnB pin	GTCR.SCGTIOC	1b

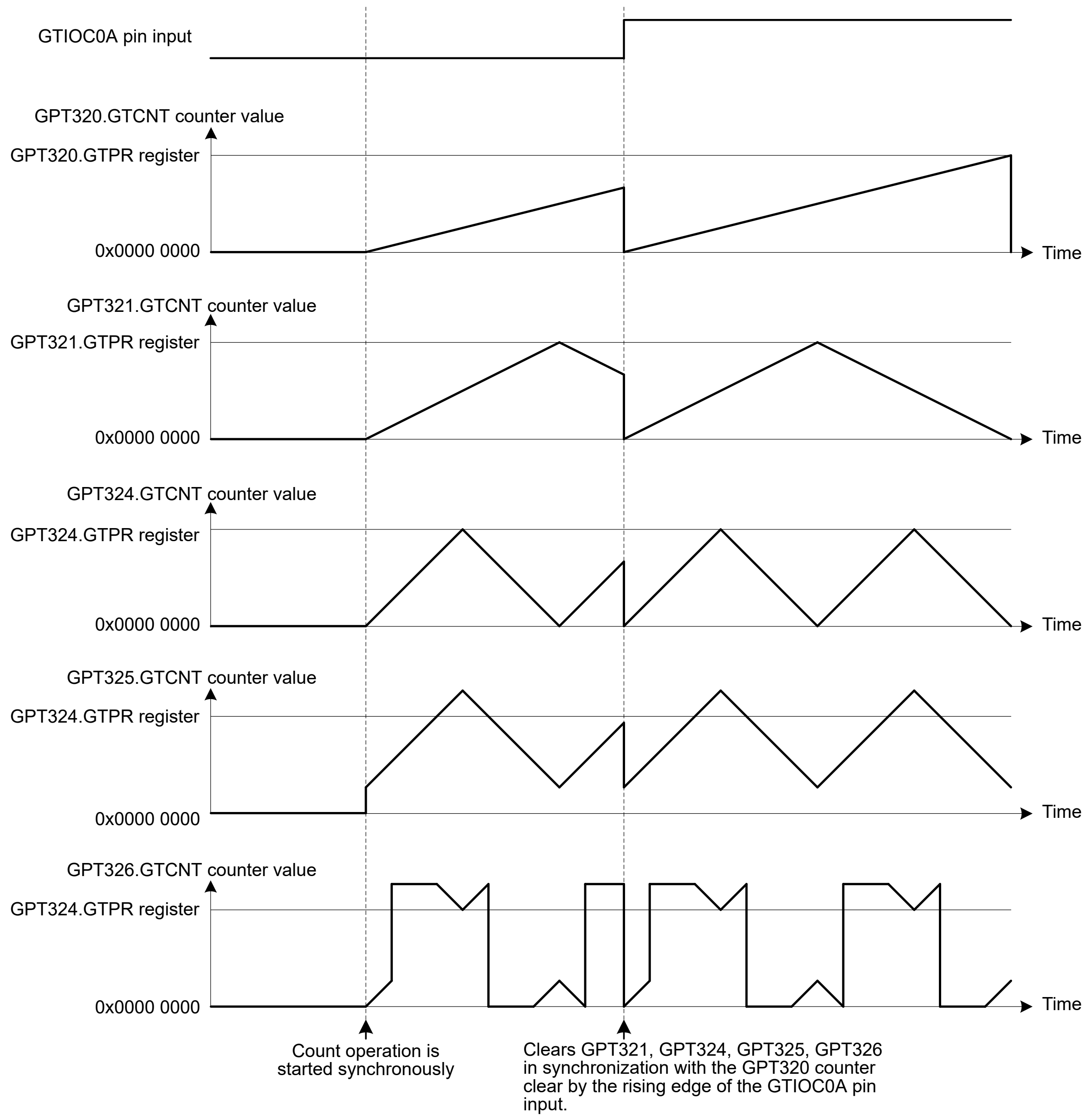


Figure 22.100 Example of Synchronous Clear Operation by Inter Channel Cooperation (GPT320 is saw-wave and counter is cleared by the rising edge of the GTIOC0A, GPT321 is triangle wave, GPT324,5,6 are complementary PWM mode. GPT320,1,4,5,6 are the same synchronous set/ clear group)

Table 22.52 Example for Setting Synchronous Clear Operation by Inter Channel Cooperation (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0] or GTCR.MD[3:0]. In Figure 22.100, 000b (saw-wave PWM mode 1) is set to GPT320. 100b (triangle-wave PWM mode 1) is set to GPT321. 1100b (complementary PWM mode 1) is set to GPT324(GPT325,6).
2	Set count direction	Select the count direction (up or down) with GTUDDTYC. In Figure 22.100, for GPT320 and GPT321, lower 2 bits of GTUDDTYC is set to 11b, and then lower 2 bits of GTUDDTYC is set to 01b (up-counting).
3	Select count clock	Select the count clock with GTCR.TPCS[3:0] bits of the corresponding channel.

Table 22.52 Example for Setting Synchronous Clear Operation by Inter Channel Cooperation (2 of 2)

No.	Step Name	Description
4	Set cycle	Set the cycle in GTPR of the corresponding channel. When complementary PWM mode, set the cycle in GTPR of the master channel.
5	Set initial value for counter	Set the initial value in the GTCNT counter of the corresponding channel.
6	Inter channel cooperation synchronous clear setting (Source channel)	Set the GTINTAD register and GTCR.SCGTIOC bit in the source channel of inter channel cooperation synchronous clear to enable synchronous clear. When complementary PWM mode, set GTINTAD of the master channel. In Figure 22.100, GPT320.GTCR.SCGTIOC bit is 1.
7	Inter channel cooperation synchronous clear setting (Cleared channels)	Set GTCSR.CSCMSC[2:0] bits in the cleared channels of inter channel cooperation synchronous clear to select the counter clear by synchronous counter clearing group. When complementary PWM mode, set GTCSR of the master channel. In Figure 22.100, GTCSR.CSCMSC[2:0] bits of GPT321, GPT324, GPT325 and GPT326 is 111b.
8	Set group of inter channel cooperation synchronous clear	Set the same value to GTCR.SSCGRP[1:0] bits in the source channel and cleared channels of inter channel cooperation synchronous clear and set them in the same synchronous set/clear group. When complementary PWM mode, set GTCR of the master channel.
9	Enable inter channel cooperation synchronous clear	Set GTCR.SSCEN bits in the source channel and cleared channels of inter channel cooperation synchronous clear to enable synchronous clear.

Table 22.53 Example for Setting Synchronous Clear Operation by Inter Channel Cooperation (Clear caused by the GTCCRA input capture using the GTIOCnA/GTIOCnB pin)

Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or the GTCR.MD[3:0] bits.
Set count direction	Select the count direction (up or down) with the GTUDDTYC register.
Select count clock	Select the count clock with GTCR.TPCS[2:0] bits of the corresponding channel, or Select the count clock with GTCR.TPCS [3:0] bits of the corresponding channel
Period setting	Set a period in the GTPR register of the corresponding channel. When complementary PWM mode, set a period in the GTPR register of the master channel.
Set initial value for counter	Set the initial value in the GTCNT counter of the corresponding channel.
Inter channel cooperation synchronous clear setting (Source channel)	Set the input capture factor using the GTIOCnA/B pin to the GTICASR register and set the same factor as the GTICASR register to the GTCSR register as the count clearing factor. In addition, Set GTINTAD.SCFA bit=1b and GTCSR.CSCMSC[2:0] bit=001b to enable synchronous clear source by the GTCCRA input capture. When complementary PWM mode, set the GTINTAD register of the master channel.
Inter channel cooperation synchronous clear setting (Cleared channels)	Set the same value to the GTCR.SSCGRP[1:0] bits in the source channel and cleared channels of inter channel cooperation synchronous clear and set them in the same synchronous set/clear group. When complementary PWM mode, set the GTCR register of the master channel.
Enable inter channel cooperation synchronous clear	Set the GTCR.SSCEN bits in the source channel and cleared channels of inter channel cooperation synchronous clear to enable synchronous clear.

22.3.9 PWM Output Operation Examples

(1) Synchronized PWM output

The GPT outputs 14 × 2 phases of linked PWM waveforms for a maximum of GPT × 14 channels.

Figure 22.101 shows an example in which four channels perform synchronized operation in saw-wave PWM mode 1 and eight phases of PWM waveforms are output. The GTIOCnA is set so that it outputs low as the initial value, high at a GTCCRA compare match, and low at the cycle end. The GTIOCnB is set so that it outputs low as the initial value, high at a GTCCRB compare match, and low at the cycle end.

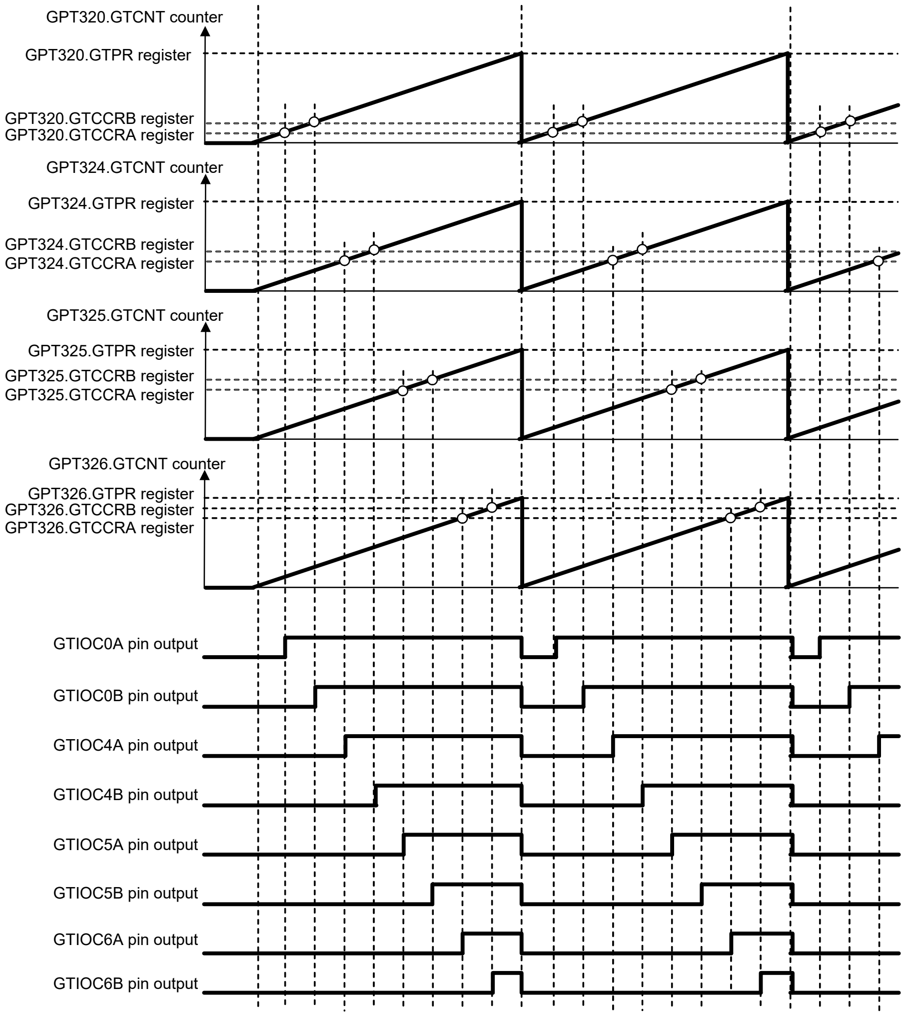


Figure 22.101 Example of synchronized PWM output

(2) 3-phase saw-wave complementary PWM output

Figure 22.102 shows an example in which three channels perform synchronized operation in saw-wave PWM mode 1 and 3-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it outputs low as the initial value, high at a GTCCRA compare match, and low at the cycle end. The GTIOCnB pin is set so that it outputs high as the initial value, low at a GTCCRB compare match, and high at the cycle end.

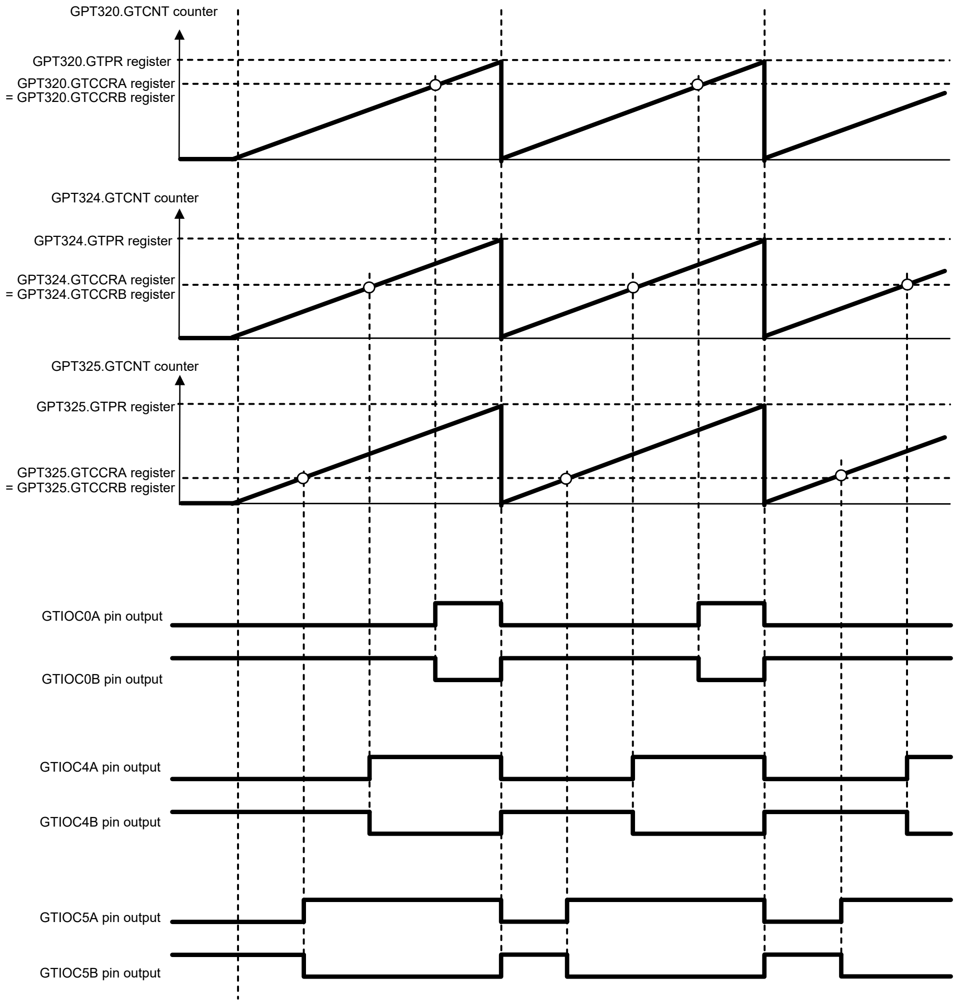


Figure 22.102 Example of 3-phase saw-wave complementary PWM output

(3) 3-phase saw-wave complementary PWM output with automatic dead time setting

Figure 22.103 shows an example in which three channels perform synchronized operation in saw-wave one-shot pulse mode with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCnB pin is set so that it outputs high as the initial value, toggles the output at a GTIOCnB compare match, and retains the output at the cycle end.

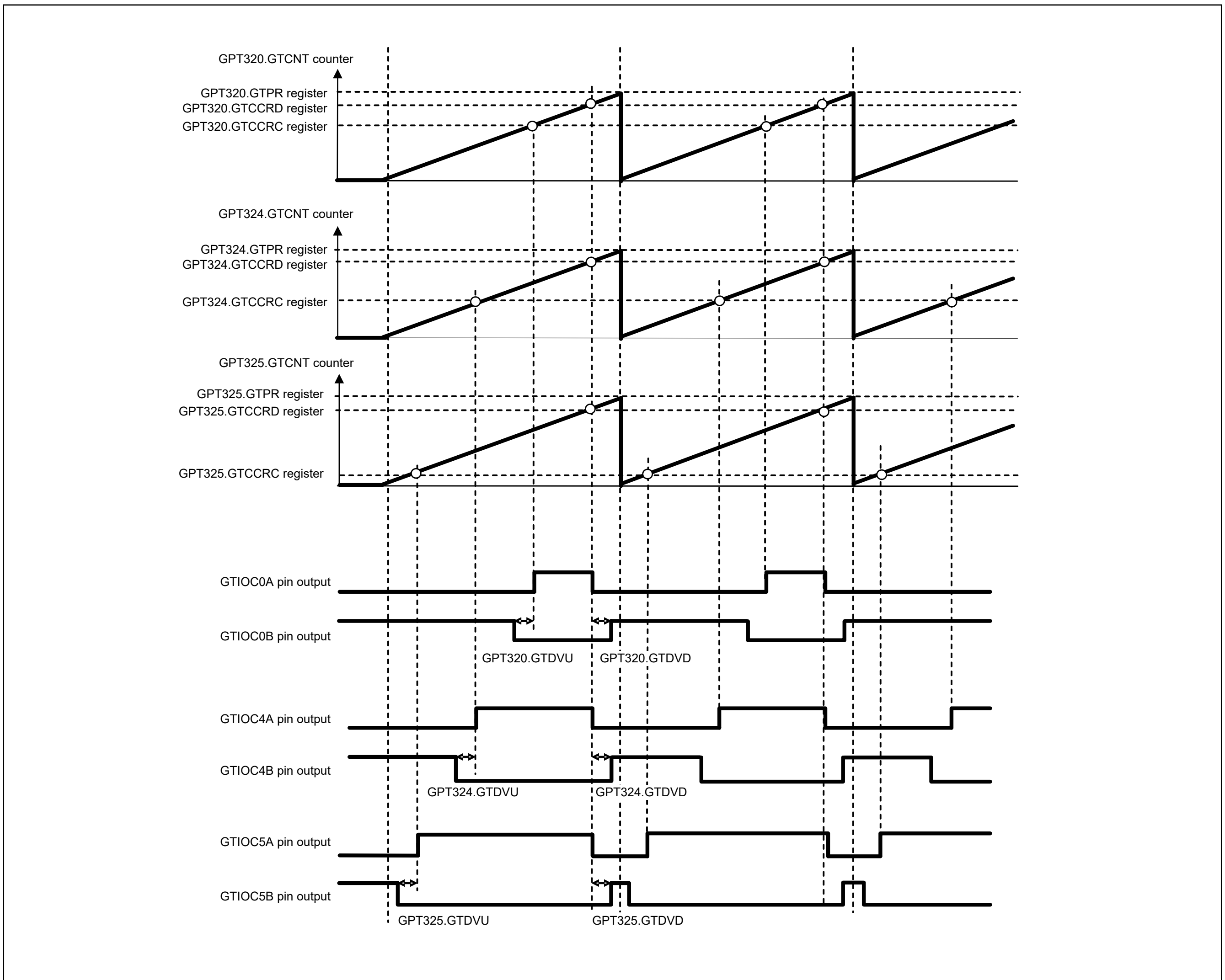


Figure 22.103 Example of 3-phase saw-wave complementary PWM output with automatic dead time setting

(4) 3-phase triangle-wave complementary PWM output

Figure 22.104 shows an example in which three channels perform synchronized operation in triangle-wave PWM mode 1 and 3-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCnB pin is set so that it outputs high as the initial value, toggles the output at a GTCCRB compare match, and retains the output at the cycle end.

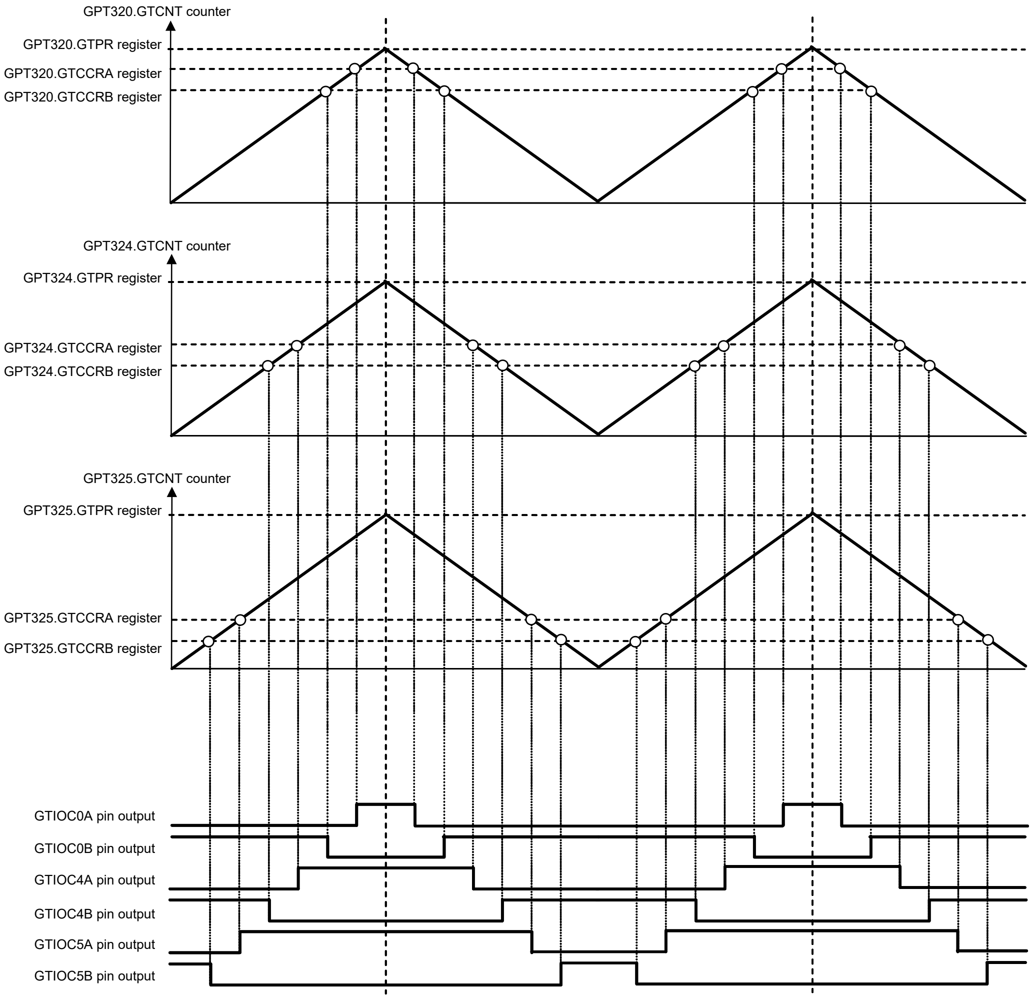


Figure 22.104 Example of 3-phase triangle-wave complementary PWM output

(5) 3-phase triangle-wave complementary PWM output with automatic dead time setting

Figure 22.105 shows an example in which three channels perform synchronized operation in triangle-wave PWM mode 1 with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCnB pin is set so that it outputs high as the initial value, toggles the output at a GTCCRB compare match, and retains the output at the cycle end.

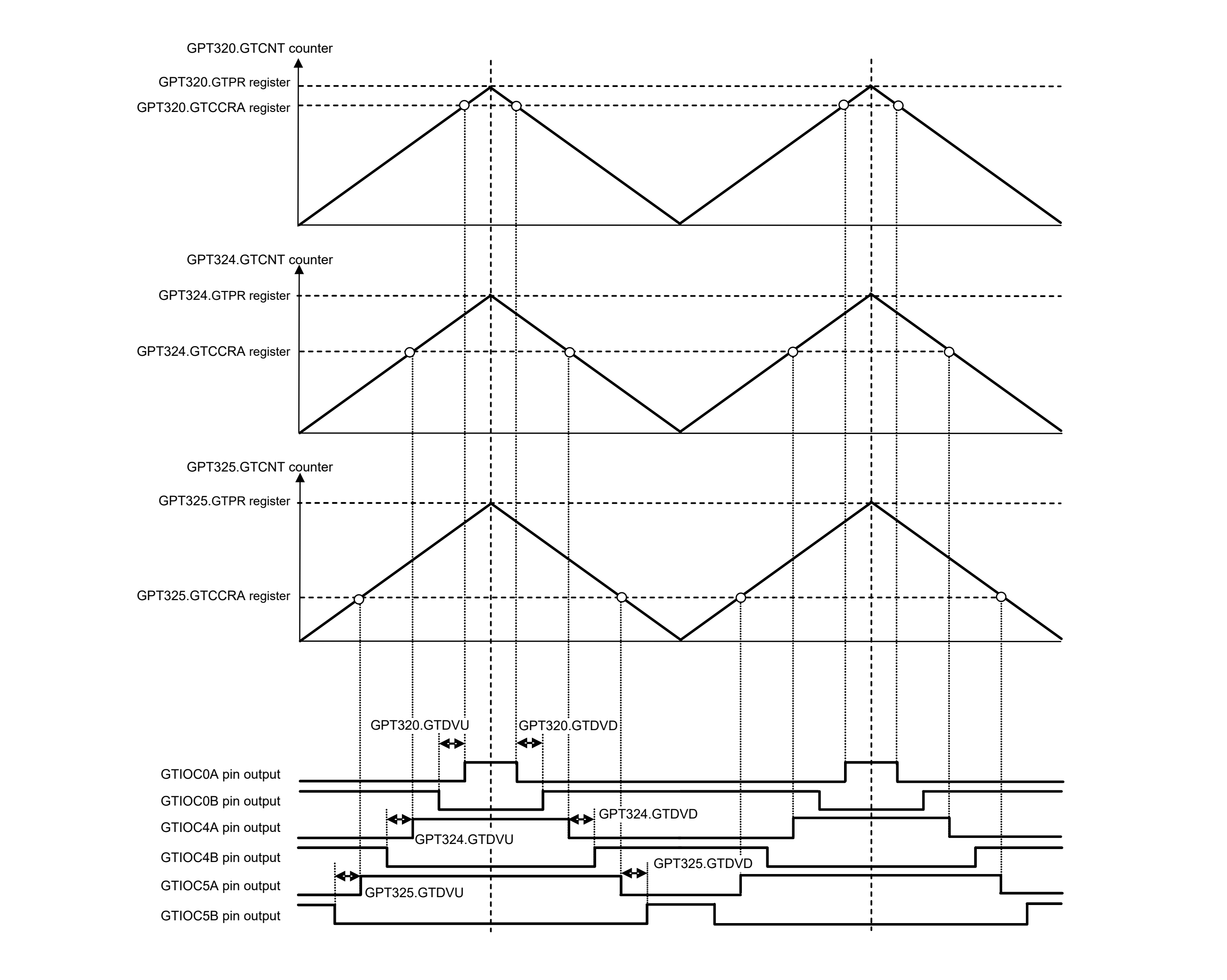


Figure 22.105 Example of 3-phase triangle-wave complementary PWM output with automatic dead time setting

(6) 3-phase asymmetric triangle-wave complementary PWM output with automatic dead time setting

Figure 22.106 shows an example in which three channels perform synchronized operation in triangle-wave PWM mode 3 with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCnA is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCnB is set so that it outputs high as the initial value, toggles the output at a GTCCRB compare match, and retains the output at the cycle end.

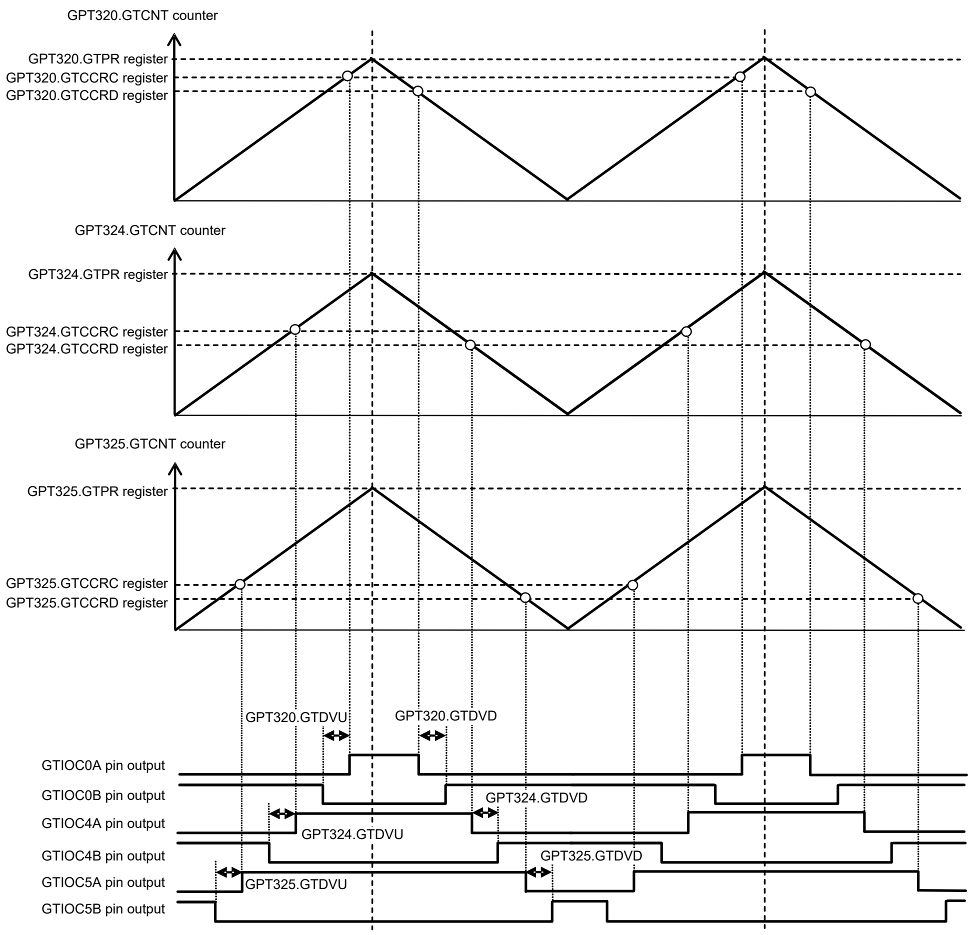


Figure 22.106 Example of 3-phase asymmetric triangle-wave complementary PWM output with automatic dead time setting

22.3.10 Period Count Function

By setting the GTPC register, the end of period can be counted.

The number of period to be counted should be set into the GTPC.PCNT counter when the GTPC.PCEN bit is 0. When the PCEN bit is 1, the PCNT counter can be read, but writing is disabled. When the PCEN bit is 1, down-counting is performed at the end of period. When the PCNT counter is 1 at the end of period, it becomes 0 and counting is stopped to finish the period count function. At that time, the GTST.PCF flag is set, and the period count function finish interrupt request GPTn_PC is generated. When the GTPC.ASTP bit is 1, the GTCNT counter is also stopped at the same time that the period count function is finished.

When the GTCNT counter is stopped while period count function is enabled, the PCNT counter keeps its value. When the GTCNT counter restarts counting and the PCEN bit is 1, the PCNT counter restarts down-counting from the hold value.

When the PCEN bit is changed from 0 to 1 while the PCNT counter is 0 and the ASTP bit is 1, the GTCNT counter is stopped at the count clock immediately after that.

When either GTSECR.SPCE bit or GTSECR.SPCD bit is set to 1, the PCEN bit in the channels set to 1 by the GTSECSR register is simultaneously set the value to enable or disable the period count function for multiple channels.

Figure 22.107 and Figure 22.108 show examples of PWM cycle count function.

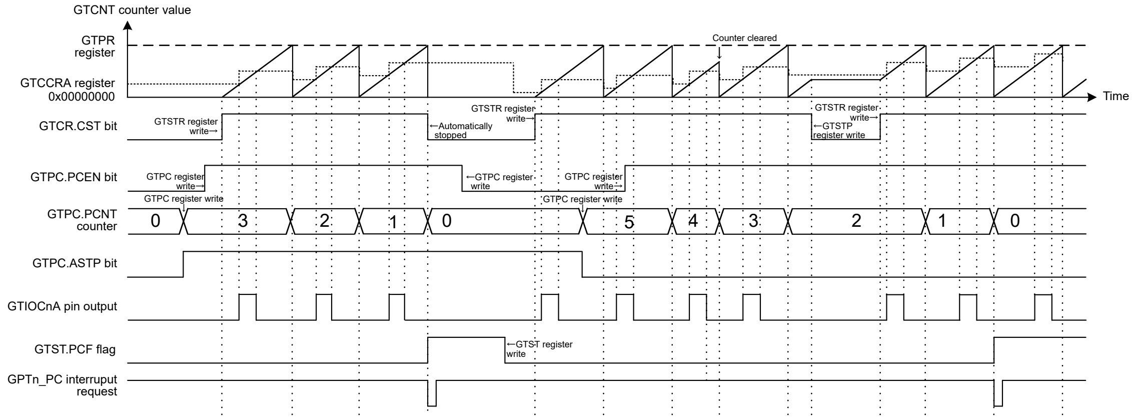
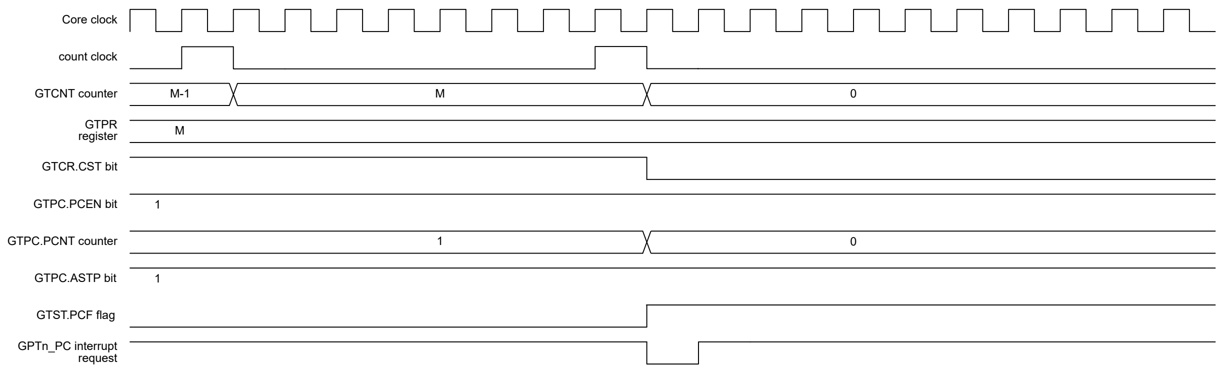


Figure 22.107 Example of PWM cycle count function (saw-wave one-shot pulse mode)



Note: Core clock: GTCLK

Figure 22.108 Example of the timing of operations for PWM cycle count function (saw-wave one-shot pulse mode, up-counting)

22.3.11 Phase Counting Function

The phase difference between the GTIOCnA and GTIOCnB pin (n = 0 to 13) inputs is detected and the associated GTCNT counts up or counts down. The detectable phase difference is available in any combination with the relationship between the edge and the level of GTIOCnA and GTIOCnB pin inputs being set in the GTUPSR and GTDNSR registers. For details on count operation, see section 22.3.1.1. Counter Operation.

Figure 22.109 to Figure 22.118 show an example of phase counting modes 1 to 5 operation when the GTIOCnA, GTIOCnB pins are used. Table 22.54 to Table 22.63 show conditions of up-counting or down-counting and list settings for the GTUPSR and GTDNSR registers which is corresponding to Figure 22.109 to Figure 22.118.

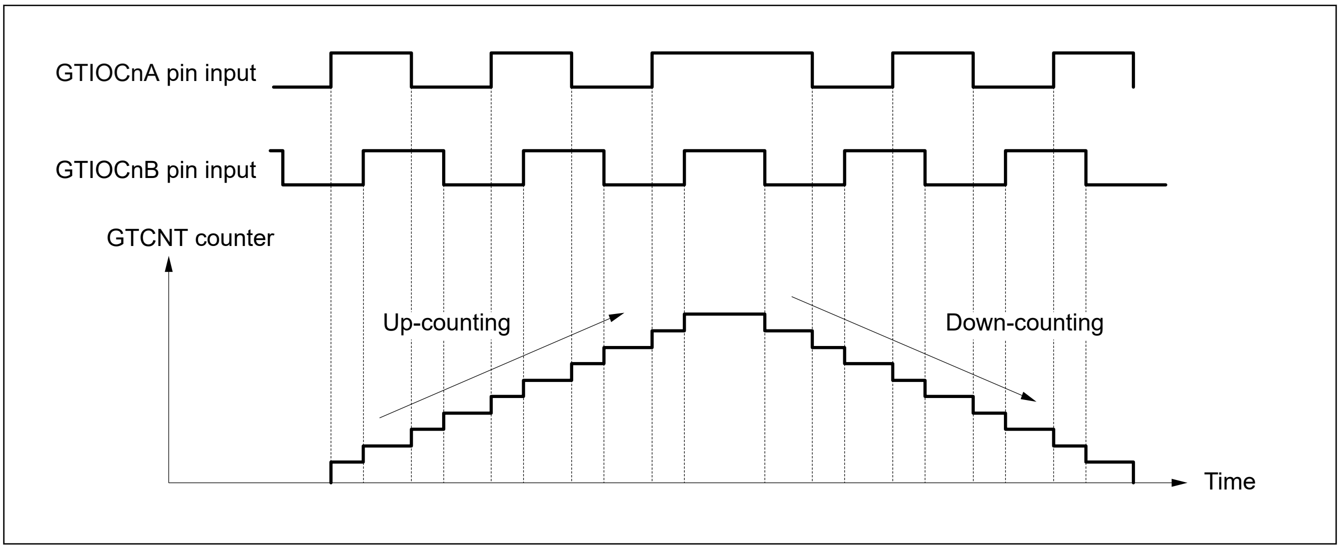


Figure 22.109 Example of phase counting mode 1

Table 22.54 Conditions of up-counting/down-counting in phase counting mode 1

↑ : Rising edge

↓ : Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Up-counting	GTUPSR = 0x00006900 GTDNSR = 0x00009600
Low	↓		
↑	Low		
↓	High		
High	↓	Down-counting	
Low	↑		
↑	High		
↓	Low		

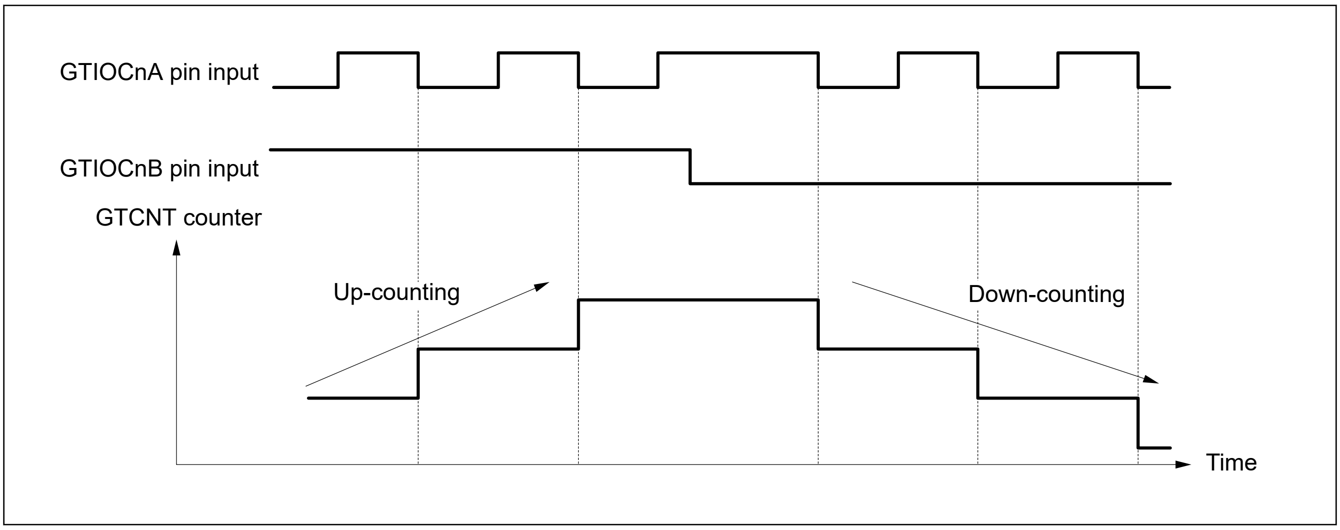


Figure 22.110 Example of phase counting mode 2 (A)

Table 22.55 Conditions of up-counting/down-counting in phase counting mode 2 (A)

↑ : Rising edge

↓ : Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Not counting	GTUPSR = 0x00000800 GTDNSR = 0x00000400
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Not counting	
Low	↑		
↑	High		
↓	Low	Down-counting	

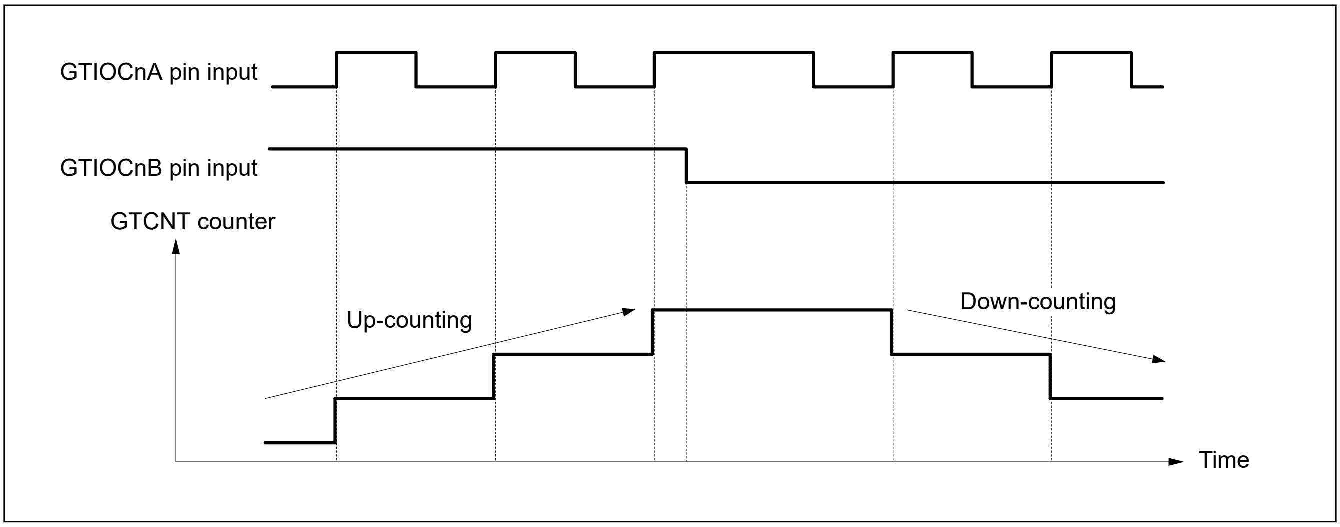


Figure 22.111 Example of phase counting mode 2 (B)

Table 22.56 Conditions of up-counting/down-counting in phase counting mode 2 (B)

↑ : Rising edge

↓ : Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Not counting	GTUPSR = 0x00000200 GTDNSR = 0x00000100
Low	↓		
↑	Low	Down-counting	
↓	High	Not counting	
High	↓		
Low	↑		
↑	High	Up-counting	
↓	Low	Not counting	

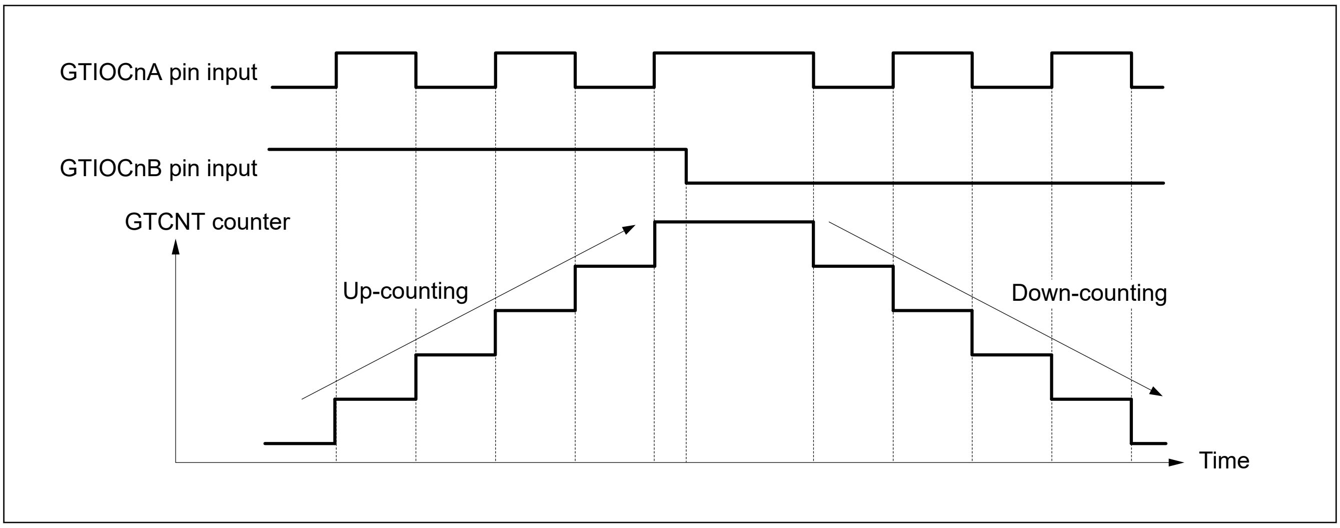


Figure 22.112 Example of phase counting mode 2 (C)

Table 22.57 Conditions of up-counting/down-counting in phase counting mode 2 (C)

↑ : Rising edge

↓ : Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Not counting	GTUPSR = 0x00000A00 GTDNSR = 0x00000500
Low	↓		
↑	Low	Down-counting	
↓	High	Up-counting	
High	↓	Not counting	
Low	↑		
↑	High	Up-counting	
↓	Low	Down-counting	

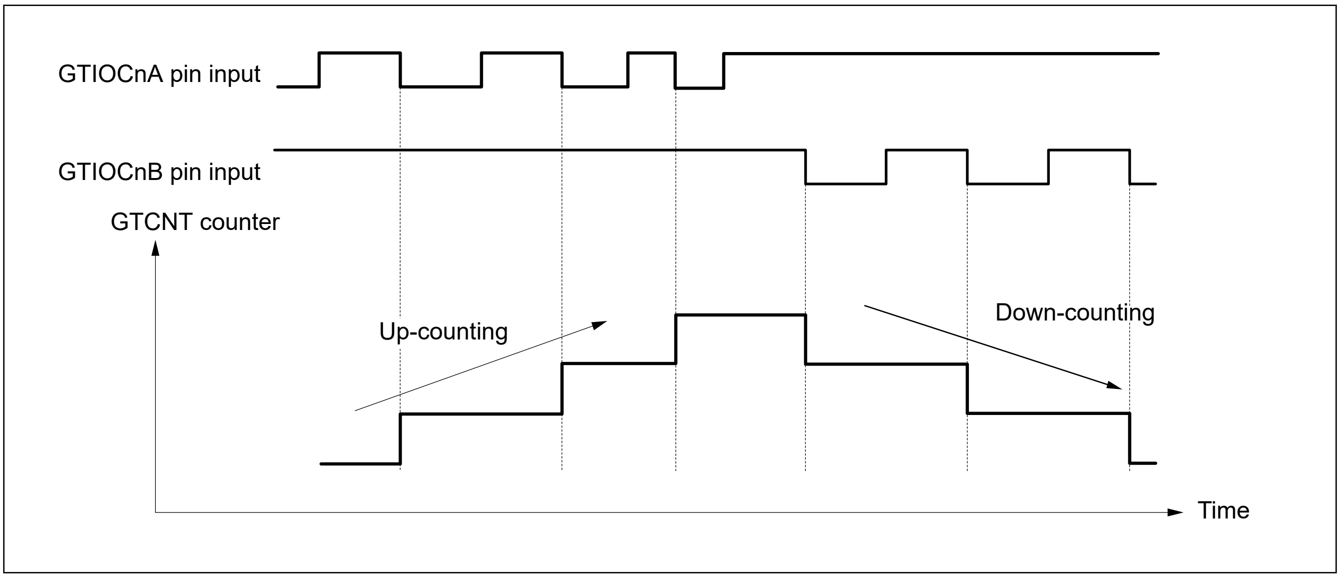


Figure 22.113 Example of phase counting mode 3 (A)

Table 22.58 Conditions of up-counting/down-counting in phase counting mode 3 (A)

↑ : Rising edge

↓ : Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Not counting	GTUPSR = 0x00000800 GTDNSR = 0x00008000
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Down-counting	
Low	↑	Not counting	
↑	High		
↓	Low		

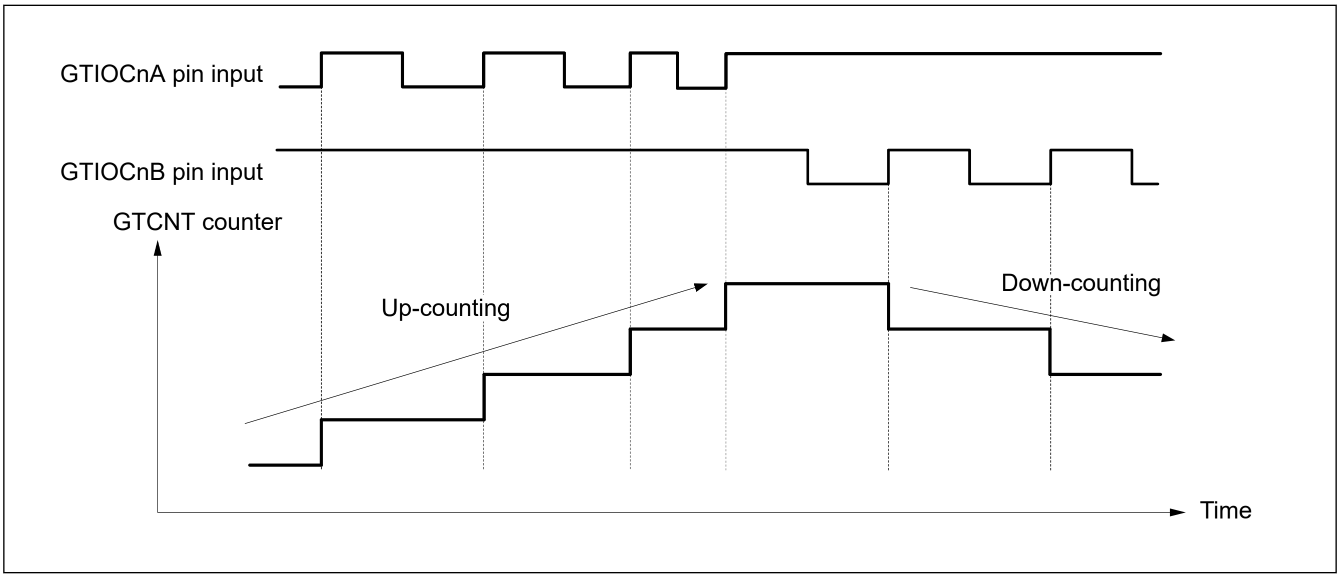


Figure 22.114 Example of phase counting mode 3 (B)

Table 22.59 Conditions of up-counting/down-counting in phase counting mode 3 (B)

↑ : Rising edge

↓ : Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Down-counting	GTUPSR = 0x00000200 GTDNSR = 0x00002000
Low	↓	Not counting	
↑	Low		
↓	High		
High	↓		
Low	↑		
↑	High	Up-counting	
↓	Low	Not counting	

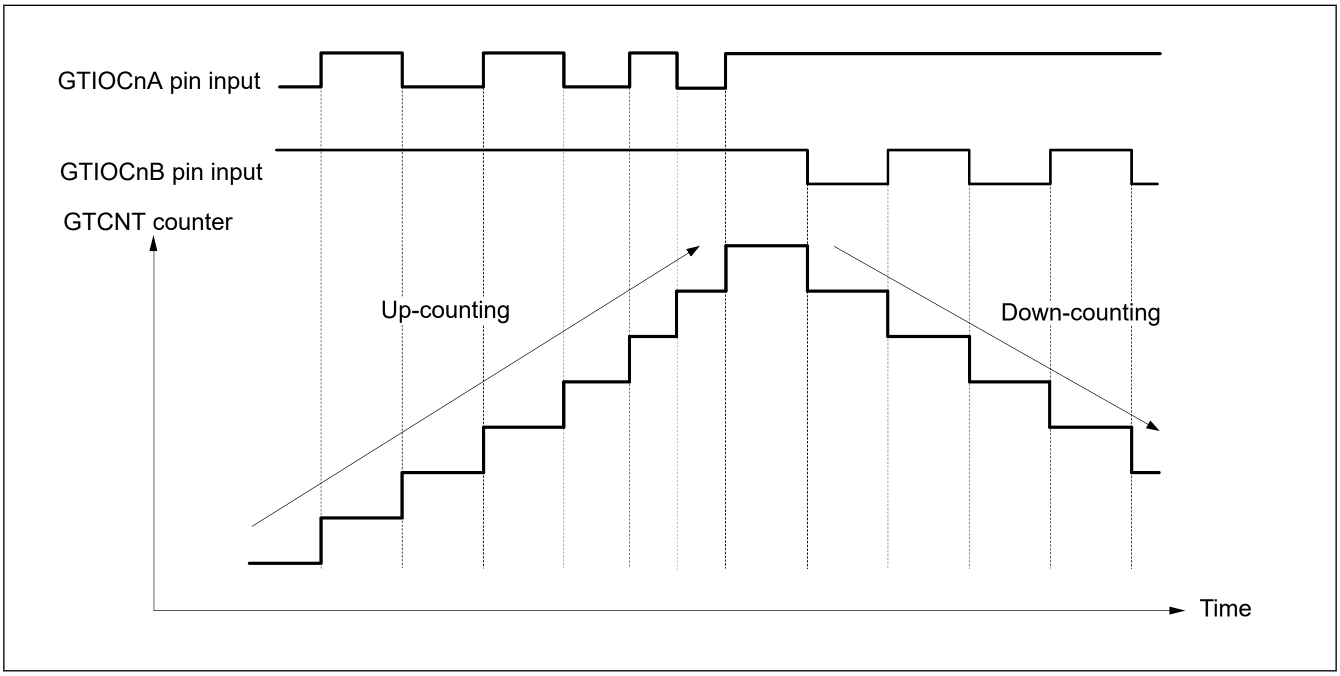


Figure 22.115 Example of phase counting mode 3 (C)

Table 22.60 Conditions of up-counting/down-counting in phase counting mode 3 (C)

↑ : Rising edge

↓ : Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Down-counting	GTUPSR = 0x00000A00 GTDNSR = 0x0000A000
Low	↓	Not counting	
↑	Low		
↓	High	Up-counting	
High	↓	Down-counting	
Low	↑	Not counting	
↑	High	Up-counting	
↓	Low	Not counting	

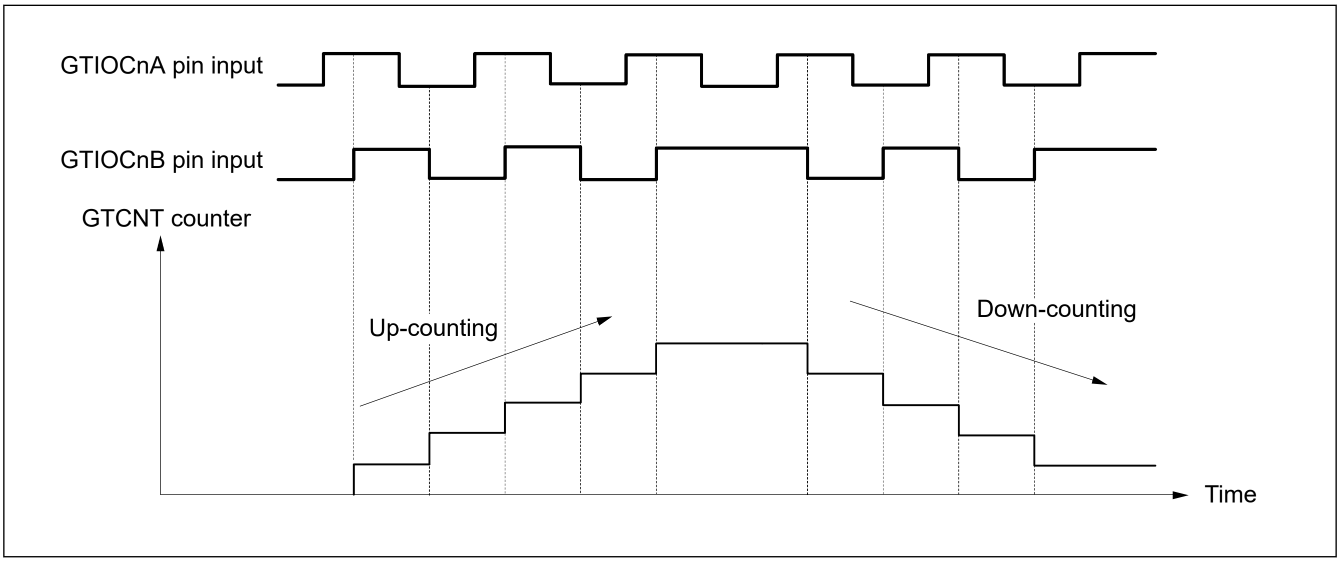


Figure 22.116 Example of phase counting mode 4

Table 22.61 Conditions of up-counting/down-counting in phase counting mode 4

↑ : Rising edge

↓ : Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Up-counting	GTUPSR = 0x00006000 GTDNSR = 0x00009000
Low	↓		
↑	Low	Not counting	
↓	High		
High	↓	Down-counting	
Low	↑		
↑	High	Not counting	
↓	Low		

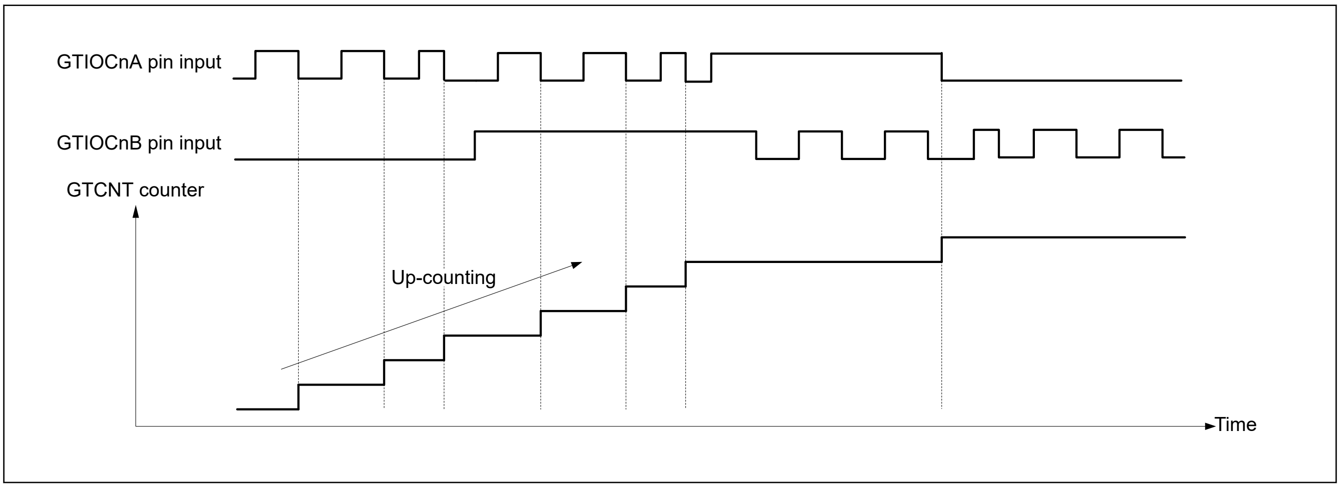


Figure 22.117 Example of phase counting mode 5 (A)

Table 22.62 Conditions of up-counting/down-counting in phase counting mode 5 (A)

↑ : Rising edge

↓ : Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Not counting	GTUPSR = 0x00000C00 GTDNSR = 0x00000000
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Not counting	
Low	↑		
↑	High		
↓	Low	Up-counting	

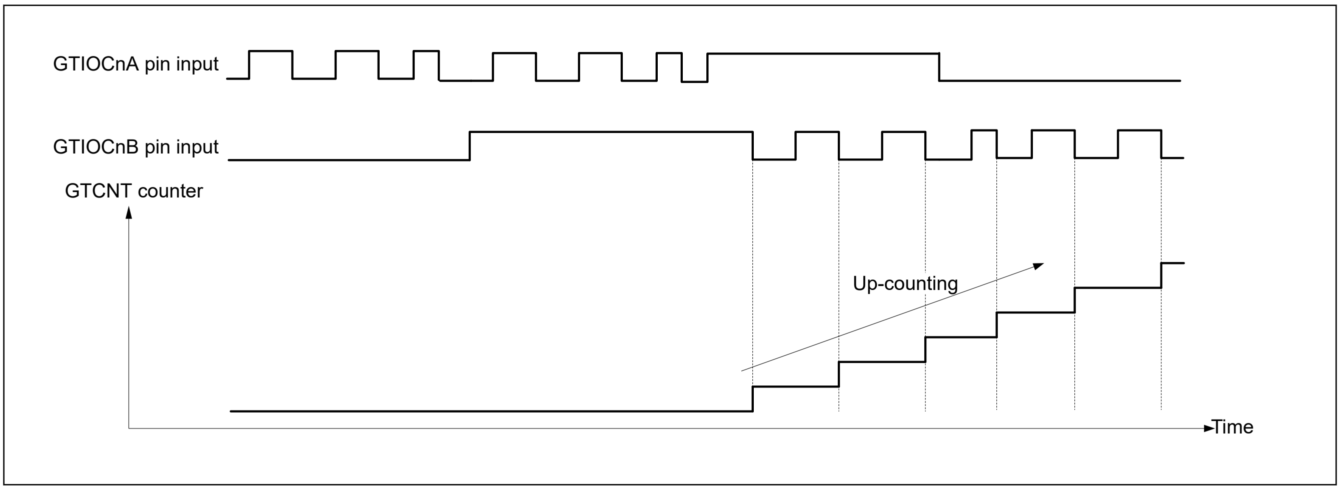


Figure 22.118 Example of phase counting mode 5 (B)

Table 22.63 Conditions of up-counting/down-counting in phase counting mode 5 (B)

↑ : Rising edge

↓ : Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Not counting	GTUPSR = 0x0000C000 GTDNSR = 0x00000000
Low	↓	Up-counting	
↑	Low	Not counting	
↓	High		
High	↓	Up-counting	
Low	↑	Not counting	
↑	High		
↓	Low		

22.3.12 External Pulse Width Measuring Function

The pulse width of GTIOCnA pin input (n = 0 to 3), GTIOCnB pin input, and GTETRGA / GTETRGB / GTETRGC / GTETRGD pin inputs can be measured.

The setting to enable or disable count-up of the GTCNT counter and the input pin and level which measured pulse width are selected by the USILVL[3:0] bits of the GTUPSR register.

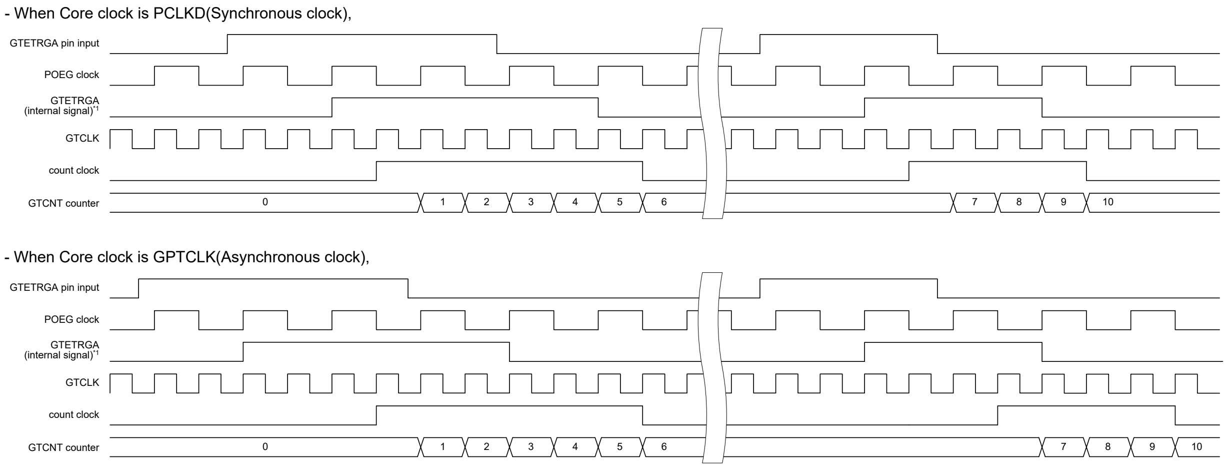
The setting to enable or disable count-down of the GTCNT counter and the input pin and level which measured pulse width are selected by the DSILVL[3:0] bits of the GTDNSR register.

The setting to enable both count-up and count-down of the GTCNT counter at the same time is prohibited.

The counting operation performs periodic counting with the period of the GTPR register.

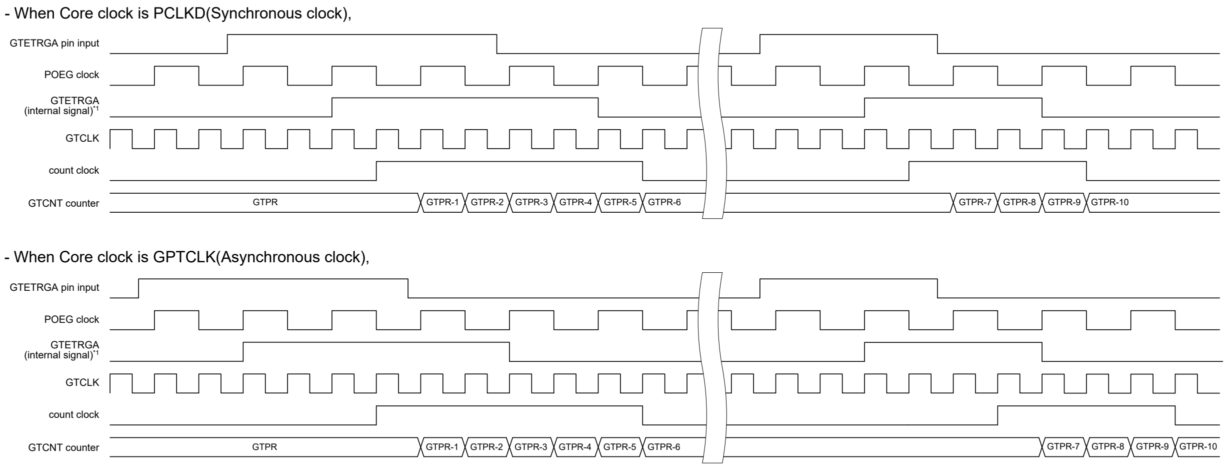
If the phase counting function and the pulse width measuring function are enabled at the same time, the pulse width measuring function does not work and the phase counting function works.

Figure 22.119, Figure 22.120 show examples of external pulse width measuring function and Table 22.64 shows example for setting external pulse width measuring function.



Note: POEG clock: PCLKB
Note 1. The input signal to GPT when the digital noise filter of GTETRGA pin is not used in POEG is shown.

Figure 22.119 Example of External pulse width measuring function(Up-counting)



Note: POEG clock: PCLKB
Note 1. The input signal to GPT when the digital noise filter of GTETRGA pin is not used in POEG is shown.

Figure 22.120 Example of External pulse width measuring function(Down-counting)

Table 22.64 Example for Setting External pulse width measuring function

No.	Step Name	Description
1	Set external pulse width measuring function	Enable external pulse width measuring function with the GTUPSR.USILVL[3:0] bits for up-counting operation and the GTDNSR.DSILVL[3:0] bits for down-counting operation and select the input pin and level to measure. In Figure 22.119, GTUPSR.USILVL[3:0] = 1001b (count up when GTETRGA pin is 1), In Figure 22.120, GTDNSR.DSILVL[3:0] = 1001b (count down when GTETRGA pin is 1)
2	Select count clock	Select the count clock with GTCR.TPCS[3:0] bits of the corresponding channel.
3	Set cycle	Set the cycle in GTPR of the corresponding channel.
4	Set initial value for counter	Set the initial value in the GTCNT counter of the corresponding channel. In Figure 22.119, 0000 0000h is set. In Figure 22.120, the GTPR register value is set.
5	Start count operation	Input a pulse to the input pin to be measured to start count operation.

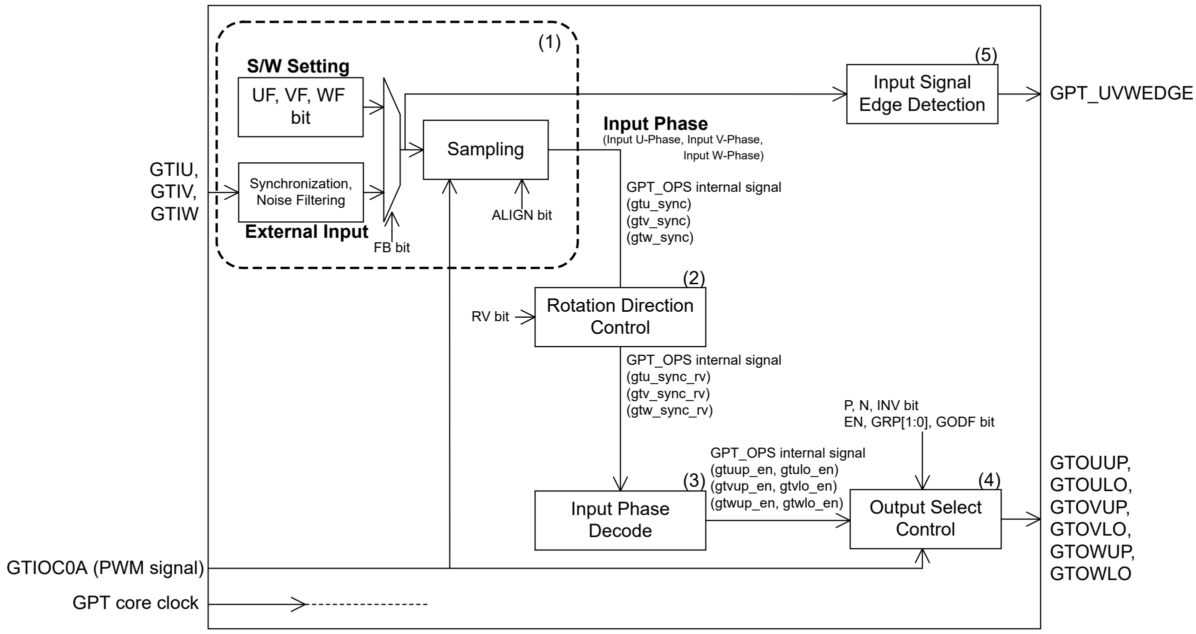
22.3.13 Output Phase Switching (GPT_OPS)

GPT_OPS can easily control Brushless DC motor using Output Phase Switching Control Register (OPSCR).

GPT_OPS uses S/W setting value (OPSCR.UF, VF, WF bits) or external signals detected by the Hall element as input signals. GPT_OPS outputs either level signals or chopped signals by GPT320's PWM as the 6-phase (U-positive phase / negative phase, V-positive phase / negative phase, W-positive phase / negative phase) signals to control motor.

Figure 22.121 shows the block diagram of GPT_OPS.

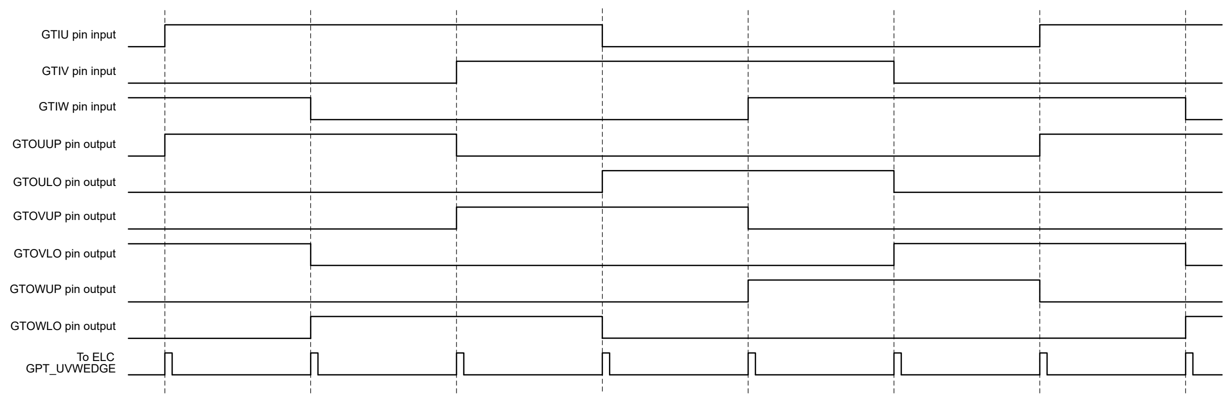
The GPT_UVWEDGE signal is output signal to ELC generated by detecting the edge of input signal.



Note: GPT core clock: GTCLK (PCLKD or GPTCLK)

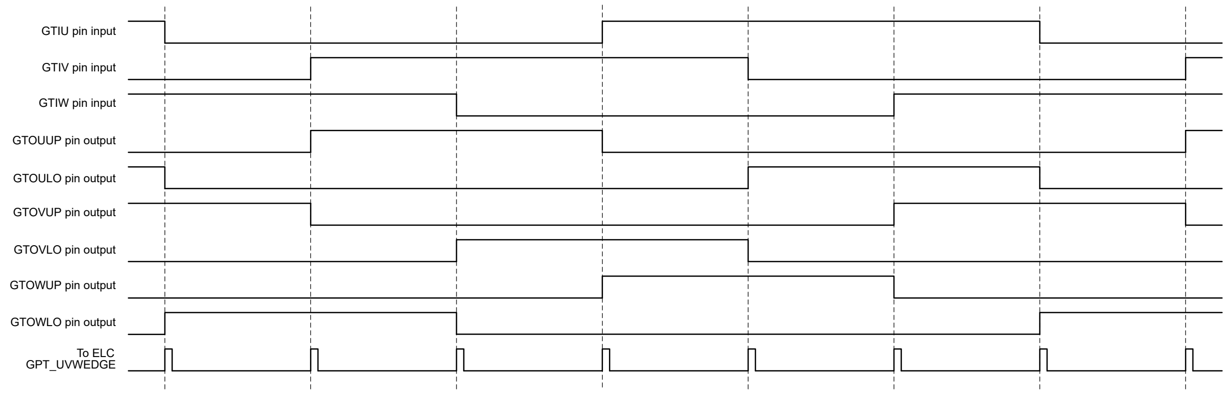
Figure 22.121 GPT_OPS Block Diagram

Figure 22.122 and Figure 22.123 show examples of GPT_OPS level output operation.



Note: Register settings: OPSCR.FB = 0, OPSCR.RV = 0, OPSCR.P = 0, OPSCR.N = 0, OPSCR.INV = 0

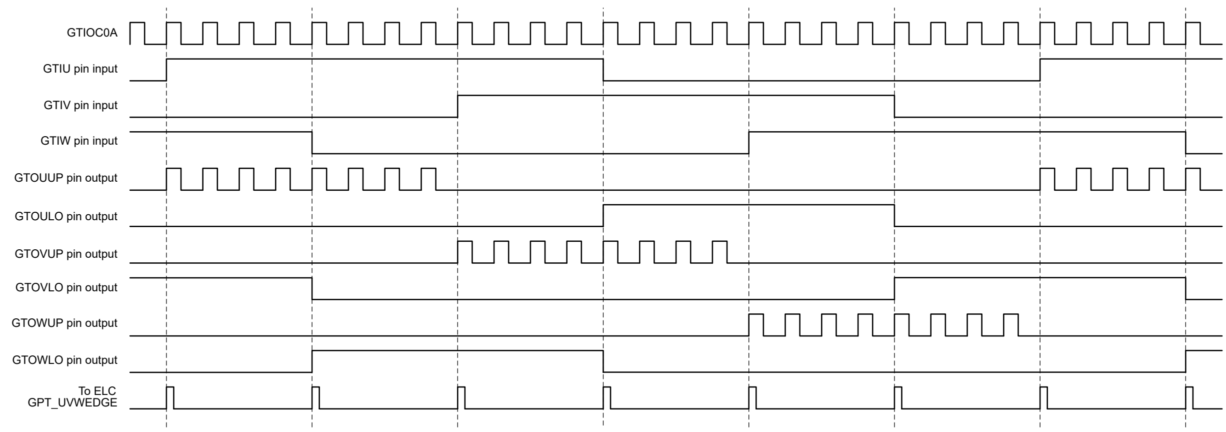
Figure 22.122 Example of GPT_OPS Level Output Operation (Forward Rotation)



Note: Register settings: OPSCR.FB = 0, OPSCR.RV = 1, OPSCR.P = 0, OPSCR.N = 0, OPSCR.INV = 0

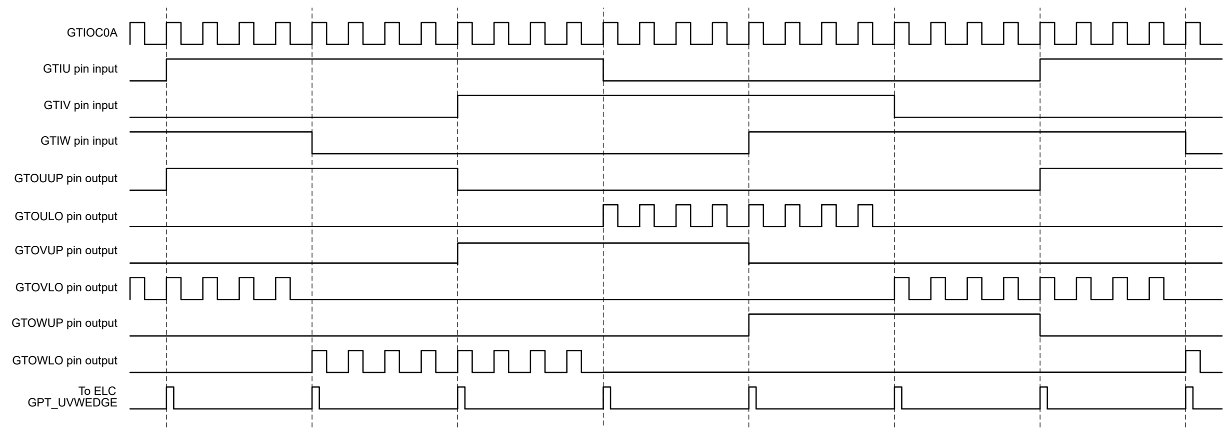
Figure 22.123 Example of GPT_OPS Level Output Operation (Reverse Rotation)

Figure 22.124 and Figure 22.125 show examples of GPT_OPS chopped output operation.



Note: Register settings: OPSCR.FB = 0, OPSCR.RV = 0, OPSCR.P = 1, OPSCR.N = 0, OPSCR.INV = 0

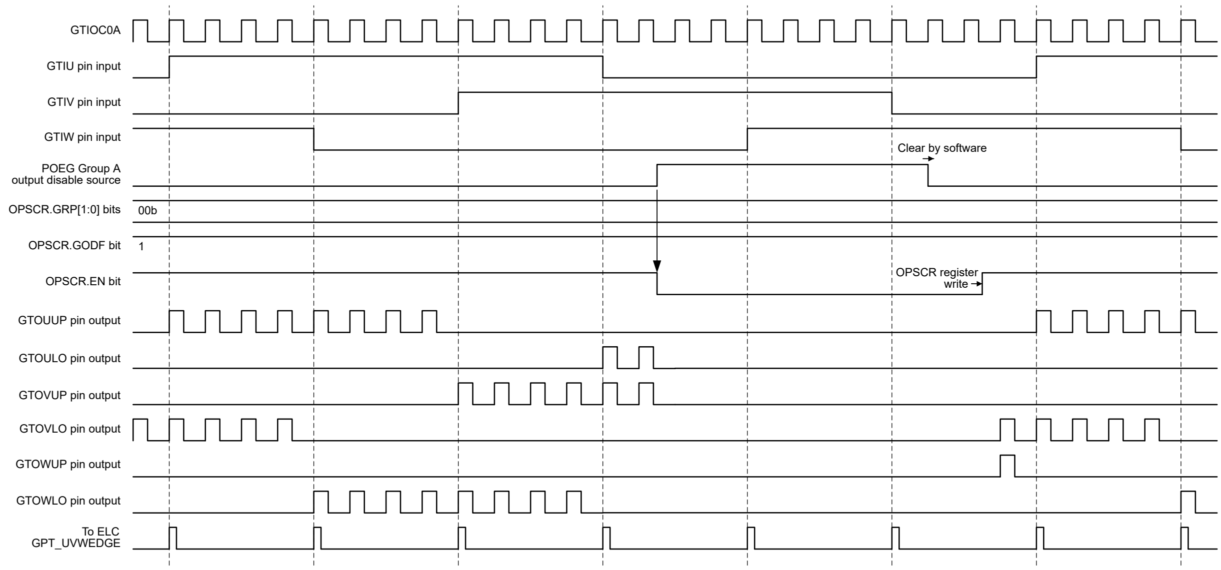
Figure 22.124 Example of GPT_OPS Chopped Output Operation (Positive Phase 120-degree)



Note: Register settings: OPSCR.FB = 0, OPSCR.RV = 0, OPSCR.P = 0, OPSCR.N = 1, OPSCR.INV = 0

Figure 22.125 Example of GPT_OPS Chopped Output Operation (Negative Phase 120-degree)

Figure 22.126 shows an example of GPT_OPS output disable control operation.



Note: Register settings: OPSCR.FB = 0, OPSCR.RV = 0, OPSCR.P = 1, OPSCR.N = 1, OPSCR.INV = 0

Figure 22.126 Example of GPT_OPS Output Disable Control Operation

22.3.13.1 Input Selection and Sampling

The FB bit selects either the software setting value or external input for the input signal.

When the FB bit is 0, the GTIU, GTIV, GTIW external input are selected for the input signal to GPT_OPS after synchronization with the GPT core clock (GTCLK) and noise filtering.

When the FB bit is 1, the software setting value (UF, VF, WF bits) are selected for the input signal to GPT_OPS.

The selected input signals are sampled by the method selected by the ALIGN bit, and they are treated as input phase of GPT_OPS.

When the ALIGN bit is 0, the input signals are sampled by GTCLK.

When the ALIGN bit is 1, the input signals are sampled by the falling edge of GTIOC0A pin output.

The signals after sampling can be read by the U, V, W bits.

Table 22.65 shows the input selection by the FB bit and sampling method by the ALIGN bit.

Table 22.65 Input Selection and Sampling Method

OPSCR Register		Input Selection Sampling Method	Input Phase (GPT_OPS internal signal)
FB bit	ALIGN bit		
0	0	GTIU, GTIV, GTIW external input GTCLK sampling	Input U-phase (gtu_sync) Input V-phase (gtv_sync) Input W-phase (gtw_sync)
	1	GTIU, GTIV, GTIW external input GTIOC0A falling edge sampling	
1	0	Software setting value UF, VF, WF bits GTCLK sampling	
	1	Software setting value UF, VF, WF bits GTIOC0A falling edge sampling	

22.3.13.2 Rotation Direction Control

When the rotation direction is reverse (RV bit = 1), the input phase is inverted.

22.3.13.3 Input Phase Decode

The 6-phase signals by decoding input phase after rotation direction control are generated.

Table 22.66 and Table 22.67 show the decode tables of input phase to rotate motor in forward (RV = 0) and reverse (RV = 1).

Table 22.66 The Decode Table of Input Phase (Forward Rotation)

	Input Phase			Input Phase after Rotation Direction Control			6-Phase Signals					
	U Phase	V Phase	W Phase	U Phase	V Phase	W Phase	U Positive Phase	U Negative Phase	V Positive Phase	V Negative Phase	W Positive Phase	W Negative Phase
	gtu_sync	gtv_sync	gtw_sync	gtu_sync_rv	gtv_sync_rv	gtw_sync_rv	gtuup_en	gtulo_en	gtvup_en	gtvlo_en	gtwup_en	gtwlo_en
	1	0	1	1	0	1	1	0	0	1	0	0
	1	0	0	1	0	0	1	0	0	0	0	1
	1	1	0	1	1	0	0	0	1	0	0	1
	0	1	0	0	1	0	0	1	1	0	0	0
	0	1	1	0	1	1	0	1	0	0	1	0
	0	0	1	0	0	1	0	0	0	1	1	0
	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	0	0	0	0	0	0

Table 22.67 The Decode Table of Input Phase (Reverse Rotation)

	Input Phase			Input Phase after Rotation Direction Control			6-Phase Signals					
	U Phase	V Phase	W Phase	U Phase	V Phase	W Phase	U Positive Phase	U Negative Phase	V Positive Phase	V Negative Phase	W Positive Phase	W Negative Phase
	gtu_sync	gtv_sync	gtw_sync	gtu_sync_rv	gtv_sync_rv	gtw_sync_rv	gtuup_en	gtulo_en	gtvup_en	gtvlo_en	gtwup_en	gtwlo_en
	1	0	1	0	1	0	0	1	1	0	0	0
	1	0	0	0	1	1	0	1	0	0	1	0
	1	1	0	0	0	1	0	0	0	1	1	0
	0	1	0	1	0	1	1	0	0	1	0	0
	0	1	1	1	0	0	1	0	0	0	0	1
	0	0	1	1	1	0	0	0	1	0	0	1
	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	0	0	0	0	0	0

22.3.13.4 Output Selection Control

The EN, P, N, INV bits select the output wave.

The EN bit enables output of the 6-phase output. When the EN bit is 1, output of the 6-phase output is enabled. When the EN bit is 0, the external pin output is Hi-Z.

The P, N bits select whether chopping positive and negative phase are performed or not. When P, N bits are 1, chopping is performed by GTIOC0A pin output.

When the chopping is performed, there are cases where the PWM width of output is shorter than the width of the PWM used to chop just before or after switching of output phase, depending on the phase difference between the phase output switch timing and the phase of PWM.

The INV bit selects the polarity (either positive logic or negative logic) of phase output.

Table 22.68 and Table 22.69, show the output selection control method for positive and negative phase output.

Table 22.68 Output Selection Control Method (Positive Phase)

EN bit	P bit	INV bit	GTOUUP / GTOVUP / GTOWUP
0	x	x	0 (External pin output is Hi-Z)
1	0	0	Positive logic level output (gtuup_en) (gtvup_en) (gtwup_en)
1	0	1	Negative logic level output (~gtuup_en) (~gtvup_en) (~gtwup_en)
1	1	0	Positive logic chopped output (GTIOC0A & gtuup_en) (GTIOC0A & gtvup_en) (GTIOC0A & gtwup_en)
1	1	1	Negative logic chopped output (~(GTIOC0A & gtuup_en)) (~(GTIOC0A & gtvup_en)) (~(GTIOC0A & gtwup_en))

Table 22.69 Output Selection Control Method (Negative Phase)

EN bit	N bit	INV bit	GTOULO / GTOVLO / GTOWLO
0	x	x	0 (External pin output is Hi-Z)
1	0	0	Positive logic level output (gtulo_en) (gtvlo_en) (gtwlo_en)
1	0	1	Negative logic level output (~gtulo_en) (~gtvlo_en) (~gtwlo_en)
1	1	0	Positive logic chopped output (GTIOC0A & gtulo_en) (GTIOC0A & gtvlo_en) (GTIOC0A & gtwlo_en)
1	1	1	Negative logic chopped output (~(GTIOC0A & gtulo_en)) (~(GTIOC0A & gtvlo_en)) (~(GTIOC0A & gtwlo_en))

22.3.13.5 Output Selection Control (Group Output Disable Function)

When GODF = 1 and signal value selected by the GRP bit is Hi (Output Disable Request), GPT_OPS's output pins are changed to Hi-Z asynchronously and the OPSCR.EN bit is cleared to 0 by the output disable request signal synchronized with GTCLK.

For the return, after clearing the Output Disable Request by software, set the EN bit to 1.

The timing of EN bit cleared to 0 is 3 GTCLK cycles after generating the output disable request. In order to perform the output disable control surely, the output disable request flag in POEG should be cleared in the timing that terminating the output disable request is at least 4 GTCLK cycles after generating the output disable request.

The example of the operation of the group output disable control, see the above-mentioned Figure 22.126.

22.3.13.6 Event Link Controller (ELC) Output

The logical sum of the pulse detected by rising and falling edge of U, V, W phase input is output to the event link controller (ELC). When the high level period of input phase is short, there are cases that the detected edge is not transmitted to the ELC correctly because of the logical sum.

22.3.13.7 GPT_OPS Start Operation Setting Flow

Table 22.70 Example for Setting of GPT_OPS Start Operation

No.	Step Name	Description
1	Set Operation Mode of GPT320	Set the PWM output operation mode of GPT320. See section 22.3.3. PWM Output Operating Mode
2	Start Count Operation of GTP320	Start count operation of GPT320, and outputs PWM waveform.
3	Set GPT_OPS Input Condition	● When the soft setting is selected, set the soft setting values in the UF, VF, WF bits. ● When the external input is selected, if necessary, use noise filtering. Select the sampling clock for external input into NFCS[1:0] bits, and set NFEN bit to 1.
4	Select GPT_OPS Input phase and Alignment	Select input phase by FB bit. Select alignment of input phase by ALIGN bit.
5	Set GPT_OPS Output phase Condition	Set the rotation direction by RV bit. Select whether chopping is performed or not by P, N bits. Select the output polarity by INV bit.
6	Set GPT_OPS Output Disable Condition	Select the error group by GRP[1:0] bits Set ON/OFF of the group output disable function by GODF bit.
7	Set GPT_OPS Operation	Set EN bit to 1 to output the 6-phase output to drive the brushless DC motor.

22.3.14 Inter Channel Logical Operation Function

The logical operation function between compare match outputs can be performed.

Figure 22.127 shows the block diagram of inter channel logical operation.

To prevent hazard to the GPT output, the signal after logical operation is latched with GTCLK. After latching, the output disable control is performed.

When the logical operation function which causes the delay of 1 GTCLK is selected, the output enable signal is also delayed with 1 GTCLK and input to the output disable control.

When the same signal (C = A or D = B) to operate logical function AND, OR, EXOR and NOR is selected, C or D is treated as 1. In the case of GTIOCnA pin output, when A of same channel is selected for C, the result of AND is A, the result of OR is 1, the result of EXOR is NOT A, and the result of NOR is 0.

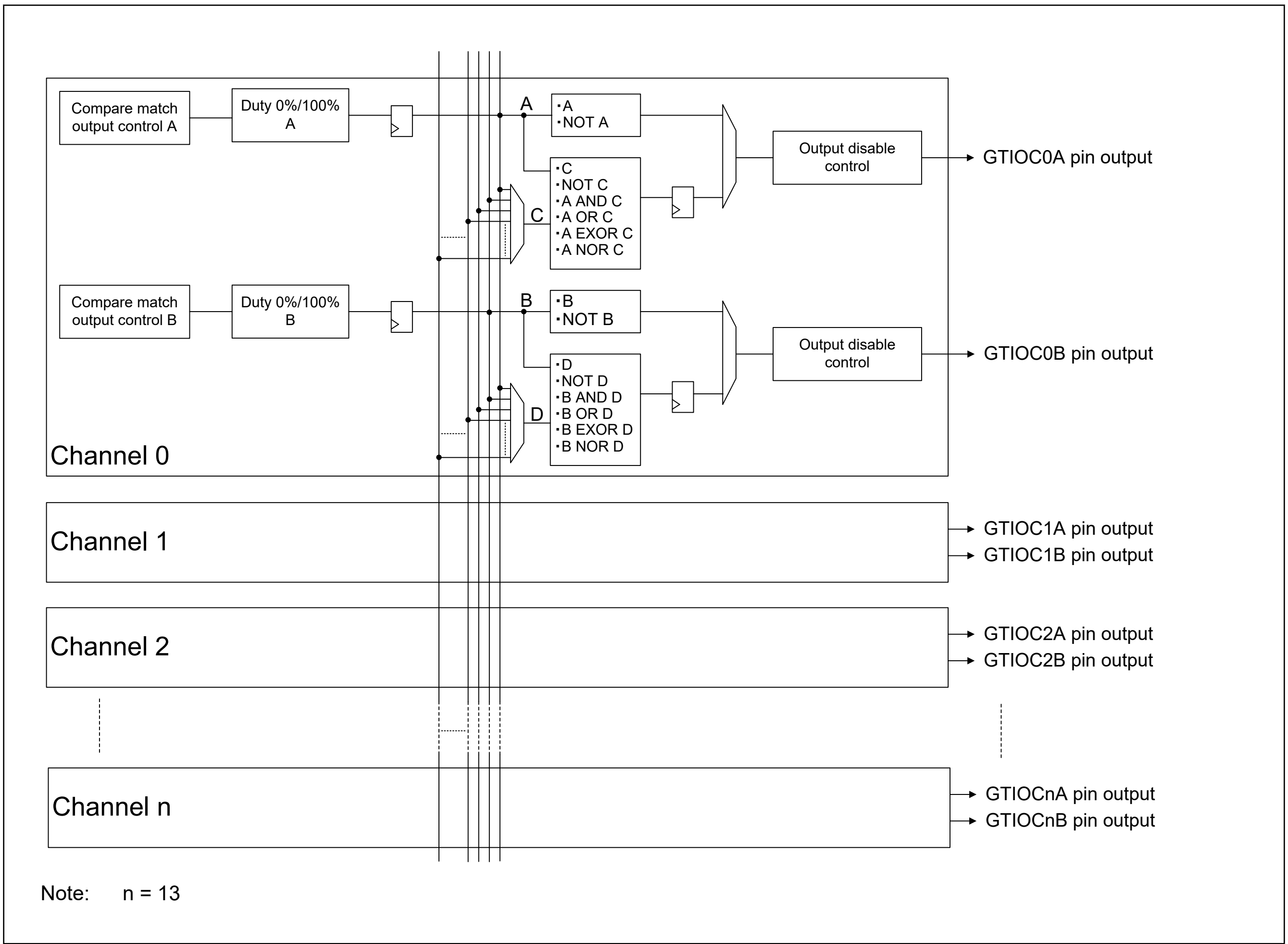


Figure 22.127 Block Diagram of Inter Channel Logical Operation

Figure 22.128 shows an example of inter channel logical operation.

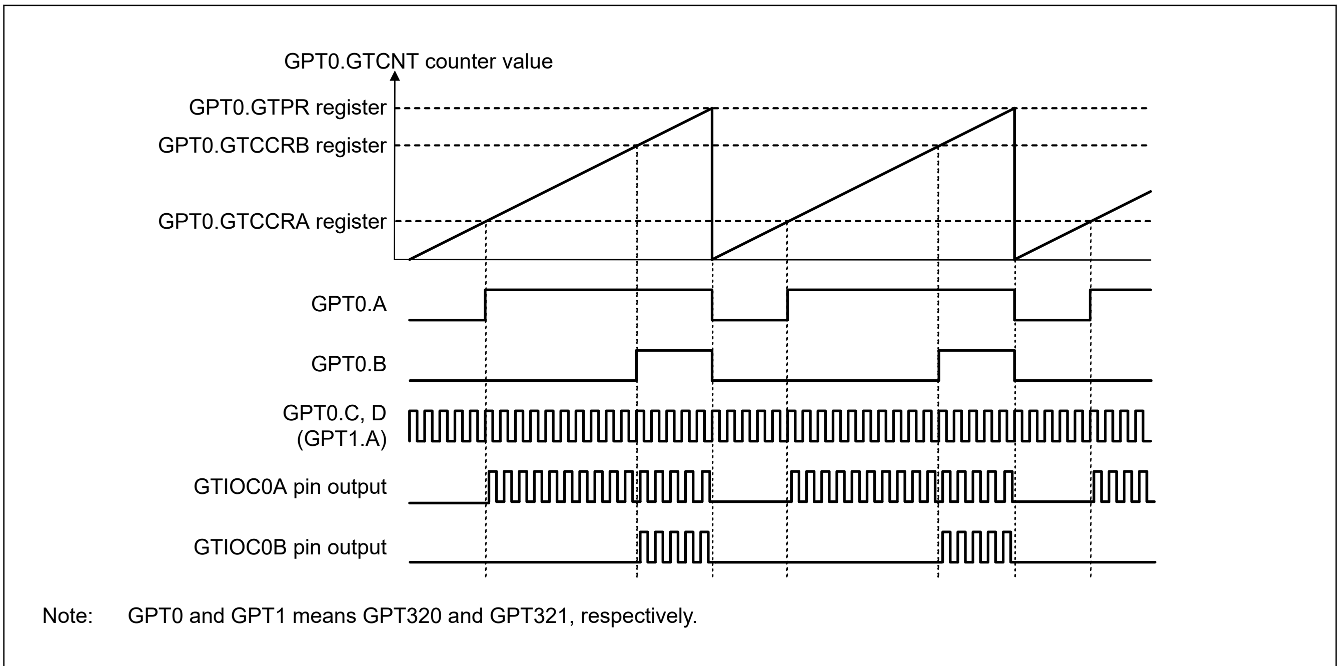


Figure 22.128 Example of Inter Channel Logical Operation

22.3.15 GTIOCnA/B Input/Output Pin Polarity Reversal Function

The polarity of the GTIOCnA, GTIOCnB input pins and the GTIOCnA, GTIOCnB output pins can be inverted according to the polarity of the external device.

When GTCR.AINV is "1", the input and output of GTIOCnA are inverted. When GTCR.BINV is "1", the input and output of GTIOCnB are inverted.

GTIOCnA and GTIOCnB are inverted at the final output stage immediately after input. When GTIOCnA and GTIOCnB are used as count start/stop/clear factors, countdown/up factors, and input capture factors, or when low/high/toggle output is used as PWM output, external input and external output are reversed. It is necessary to make settings that take this into consideration.

Set GTCR.AINV and BINV after setting the GTIOCnm pin functions during initial settings.

22.4 Interrupt Sources

22.4.1 Interrupt Sources

The GPT provides the following interrupt sources:

- GTCCR input capture/compare match
- GTADTR compare match
- GTCNT counter overflow (GTPR compare match)/underflow.
- period count function finish

Each interrupt source has its own status flag. When an interrupt source signal is generated, the associated status flag in GTST is set to 1, and an interrupt request is generated. The associated status flag in GTST can be cleared by writing 0. If flag set and flag clear occur at the same time, flag clear takes priority over flag set. These flags are automatically updated by the internal state. The Interrupt Controller Unit can change the relative channel priorities. However, the priority within a channel is fixed. For details, see section 14, Interrupt Controller Unit (ICU).

Table 22.71 lists the GPT interrupt sources.

Table 22.71 Interrupt sources

Channel	Name	Interrupt source	Interrupt flag	DTC activation
n = 0 to 13	GPTn_CCMPA	GPT32n.GTCCRA input capture/compare match	GTST[0] (TCFA)	Possible
	GPTn_CCMPB	GPT32n.GTCCRB input capture/compare match	GTST[1] (TCFB)	Possible
	GPTn_CMPC	GPT32n.GTCCRC compare match	GTST[2] (TCFC)	Possible
	GPTn_CMPD	GPT32n.GTCCRD compare match	GTST[3] (TCFD)	Possible
	GPTn_CMPE	GPT32n.GTCCRE compare match	GTST[4] (TCFE)	Possible
	GPTn_CMPF	GPT32n.GTCCRF compare match	GTST[5] (TCFF)	Possible
	GPTn_OVF	GPT32n.GTCNT overflow (GPT32n.GTPR compare match)	GTST[6] (TCFPO)	Possible
	GPTn_UDF	GPT32n.GTCNT underflow	GTST[7] (TCFPU)	Possible
	GPTn_PC	Period count function finish (n = 0 to 3, 10 to 13)	GTST[31] (PCF)	Possible

(1) GPTn_CCMPA interrupt (n = 0 to 13)

An interrupt request is generated under the following conditions:

- When the GTCCRA register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches with the GTCCRA register
- When the GTCCRA register functions as an input capture register, the input-capture signal causes transfer of the GTCNT counter value to the GTCCRA register. In complementary PWM mode, GTCCRA register does not function as an input capture register.

(2) GPTn_CCMPB interrupt (n = 0 to 13)

An interrupt request is generated under the following conditions:

- When the GTCCRB register functions as a compare match register, the GTCNT counter value matches with the GTCCRB register. In complementary PWM mode, GTCCRB register does not function as a compare match register.
- When the GTCCRB register functions as an input capture register, the input-capture signal causes transfer of the GTCNT counter value to the GTCCRB register. In complementary PWM mode, GTCCRB register does not function as an input capture register.

(3) GPTn_CMPC interrupt (n = 0 to 13)

An interrupt request is generated under the following condition:

- When the GTCCRC register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, GTCNT counter value of master channel) matches with the GTCCRC register.

A compare match is not performed and therefore, an interrupt is not requested in the following conditions:

- GTCR.MD[3:0] = 0001b (saw-wave one-shot pulse mode)
- GTCR.MD[3:0] = 0110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 01b, 10b, 11b (buffer operation with the GTCCRC register).

(4) GPTn_CMPD interrupt (n = 0 to 13)

An interrupt request is generated under the following condition:

- When the GTCCRD register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches with the GTCCRD register.

A compare match is not performed and therefore, an interrupt is not requested in the following conditions:

- GTCR.MD[3:0] = 0001b (saw-wave one-shot pulse mode)
- GTCR.MD[3:0] = 0110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 10b, 11b (buffer operation with the GTCCRD register).

(5) GPTn_CMPE interrupt (n = 0 to 13)

An interrupt request is generated under the following condition:

- When the GTCCRE register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches with the GTCCRE register.

A compare match is not performed and therefore, an interrupt is not requested in the following conditions:

- GTCR.MD[3:0] = 0001b (saw-wave one-shot pulse mode)
- GTCR.MD[3:0] = 0110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 01b, 10b, 11b (buffer operation with the GTCCRE register).

(6) GPTn_CMPF interrupt (n = 0 to 13)

An interrupt request is generated under the following condition:

- When the GTCCRF register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches with the GTCCRF register.

A compare match is not performed and therefore, an interrupt is not requested in the following conditions:

- GTCR.MD[3:0] = 0001b (saw-wave one-shot pulse mode)
- GTCR.MD[3:0] = 0110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 10b, 11b (buffer operation with the GTCCRF register).

(7) GPTn_OVF interrupt (n = 0 to 13)

An interrupt request is generated in the following conditions:

- In saw-wave PWM mode 1 and saw-wave one-shot pulse mode, interrupt requests are enabled at overflows (when the GTCNT counter value changes from GTPR to 0 during up-counting)
- In saw-wave PWM mode 2, interrupt requests are enabled at overflows (GTCNT counter value changes from GTCCRm (m = A to F) register value selected with GTCSR.CSCMSC[2:0] bits to 0) or the time when GTCNT counter value matches GTPR register value.
- In triangle-wave mode, interrupt requests are enabled at crests (the GTCNT changes from GTPR to GTPR-1)
- In complementary PWM mode, interrupt requests are enabled at crests (GTCNT counter value of master channel changes from GTPR register value to GTPR register value minus 1).
- In counting by hardware sources (include external pulse width measuring function), an overflow (GTCNT changes from GTPR to 0 in up count) has occurred.

(8) GPTn_UDF interrupt (n = 0 to 13)

An interrupt request is generated in the following conditions.

- In saw-wave PWM mode 1 and saw-wave one-shot pulse mode, interrupt requests are enabled at underflows (when the GTCNT counter value changes from 0 to GTPR during down-counting)
- In triangle-wave mode, interrupt requests are enabled at troughs (the GTCNT changes from 0 to 1)
- In complementary PWM mode, interrupt requests are enabled at troughs (GTCNT counter value of master channel changes from 0 to 1).
- In counting by hardware sources (include external pulse width measuring function), underflow (GTCNT changes from 0 to GTPR in down count) has occurred.

About Interrupt signals and interrupt status flags, see section 22.2.16. GTST : General PWM Timer Status Register.

(9) GPTn_PC Interrupt (n = 0 to 3, 10 to 13)

When the GTPC.PCEN bit is 1 and the GTPC.PCNT counter is 1, an interrupt request is generated at the end of cycle.

22.4.2 DMAC and DTC Activation

The DMAC and DTC can be activated by the interrupt in each channel. For details, see section 14, Interrupt Controller Unit (ICU), section 17, DMA Controller (DMAC), and section 18, Data Transfer Controller (DTC).

22.4.3 Interrupt and A/D Conversion Start Request Skipping Function

22.4.3.1 Interrupt Skipping Function by GTITC Register

By setting the GTITC register, the GTCNT counter overflow (GTPR register compare match) interrupt (GPTn_OVF) and underflow interrupt (GPTn_UDF) can be skipped. Other interrupts and A/D conversion start request signals can be skipped in coordination with the GPTn_OVF/GPTn_UDF skipping function. When the interrupt is skipped, the updating of relevant status flag is also skipped. The interrupt skipping is continued even if the status flag is set to 1.

The interrupt skipping function is related only to the GTITC register setting, and is not related to setting of the GTINTAD register interrupt enable bit. The interrupt skipping is continued even if the interrupt is disabled with GTINTAD register setting.

When both troughs and crests are counted and skipped in triangle-wave mode, if the number of times of skipping is odd, GPTn_OVF/GPTn_UDF interrupt requests cannot be generated at troughs only or at crests only depending on the skipping counter start timing. Therefore, in order to count both troughs and crests and generate the GPTn_OVF/GPTn_UDF interrupts at troughs only or crests only in triangle-wave mode, the number of times of skipping should be even.

Similarly, in saw-wave mode, when both overflows and underflows are counted and skipped with the count direction changed, GPTn_OVF/GPTn_UDF interrupt requests are sometimes not generated at overflows only or at underflows only. Therefore, in order to count both overflows and underflows with the count direction changed and generate the GPTn_OVF/GPTn_UDF interrupts at overflows only or underflows only in saw-wave mode, the skipping state should be carefully checked before using.

When changing the skipping count, be sure to release the skipping count setting (GTITC.IVTC[1:0] bits = 00b).

Figure 22.129 to Figure 22.134 show examples of skipping function operation.

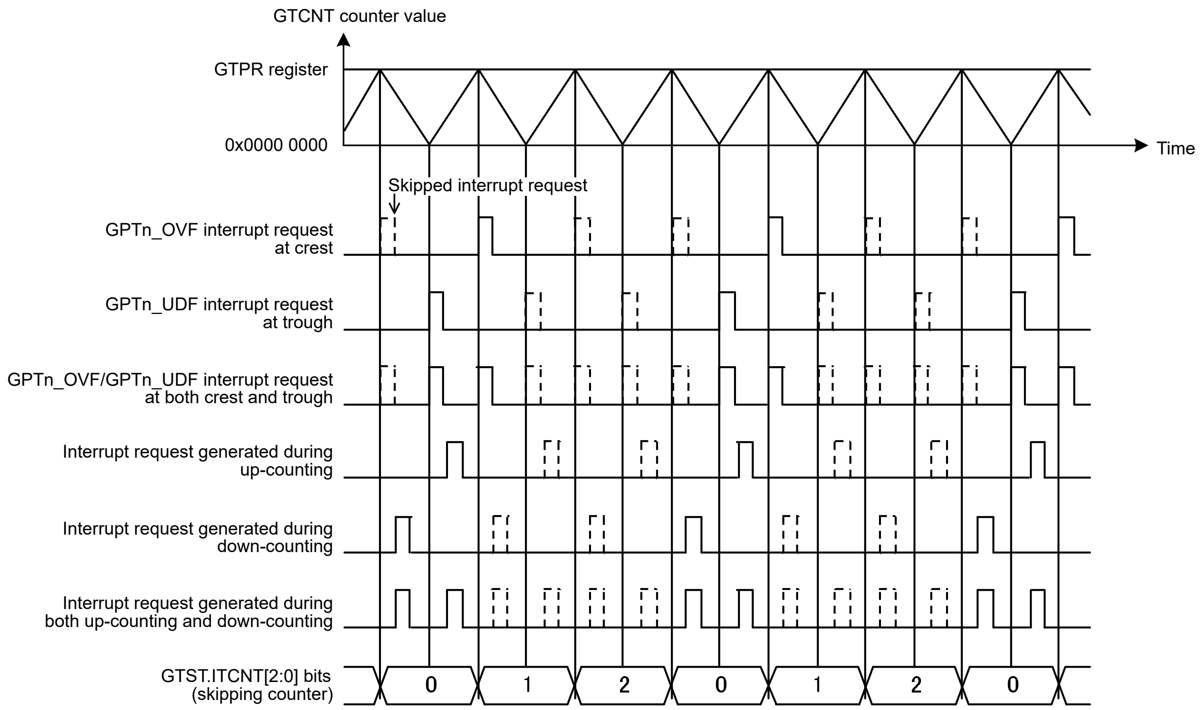


Figure 22.129 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Crests, Skipping Count: 2)

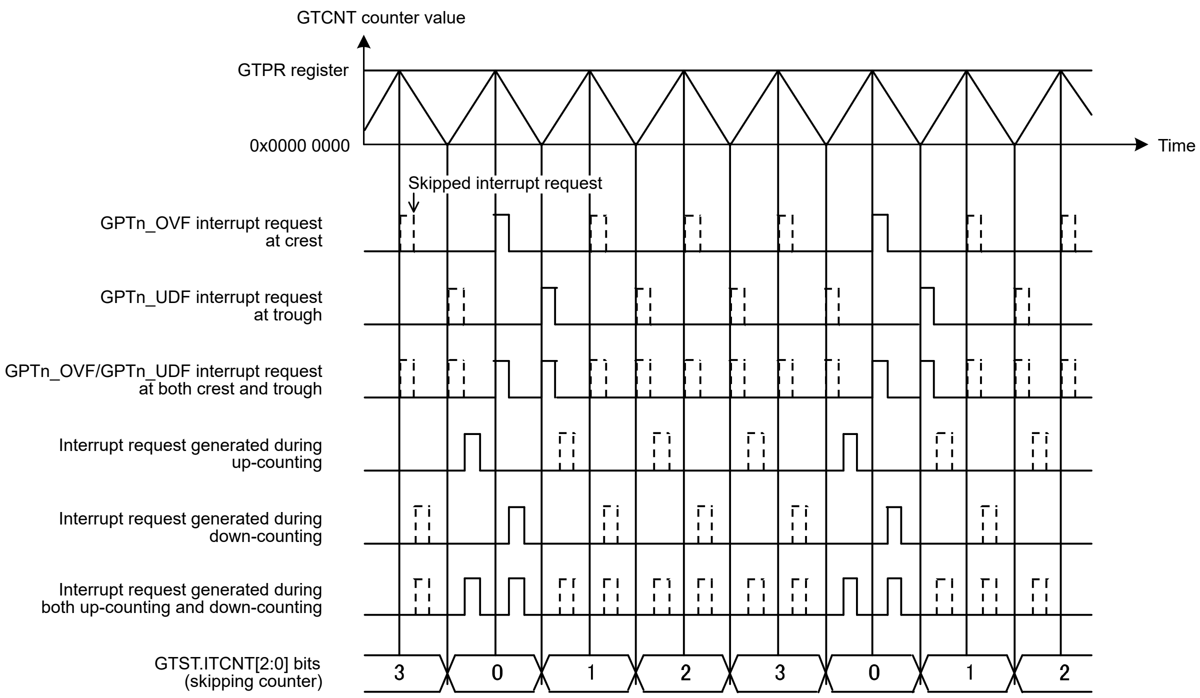


Figure 22.130 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Troughs, Skipping Count: 3)

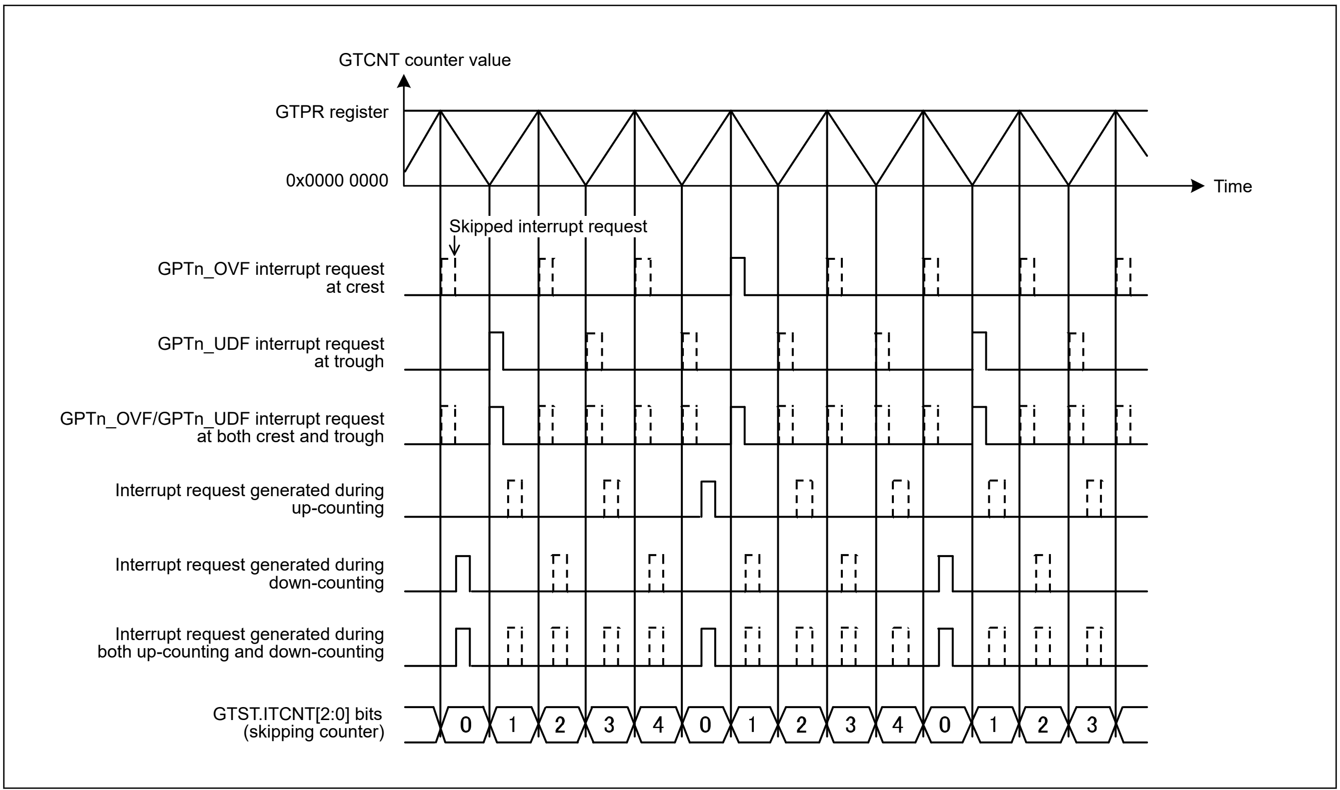


Figure 22.131 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Both Troughs and Crests, Skipping Count: 4)

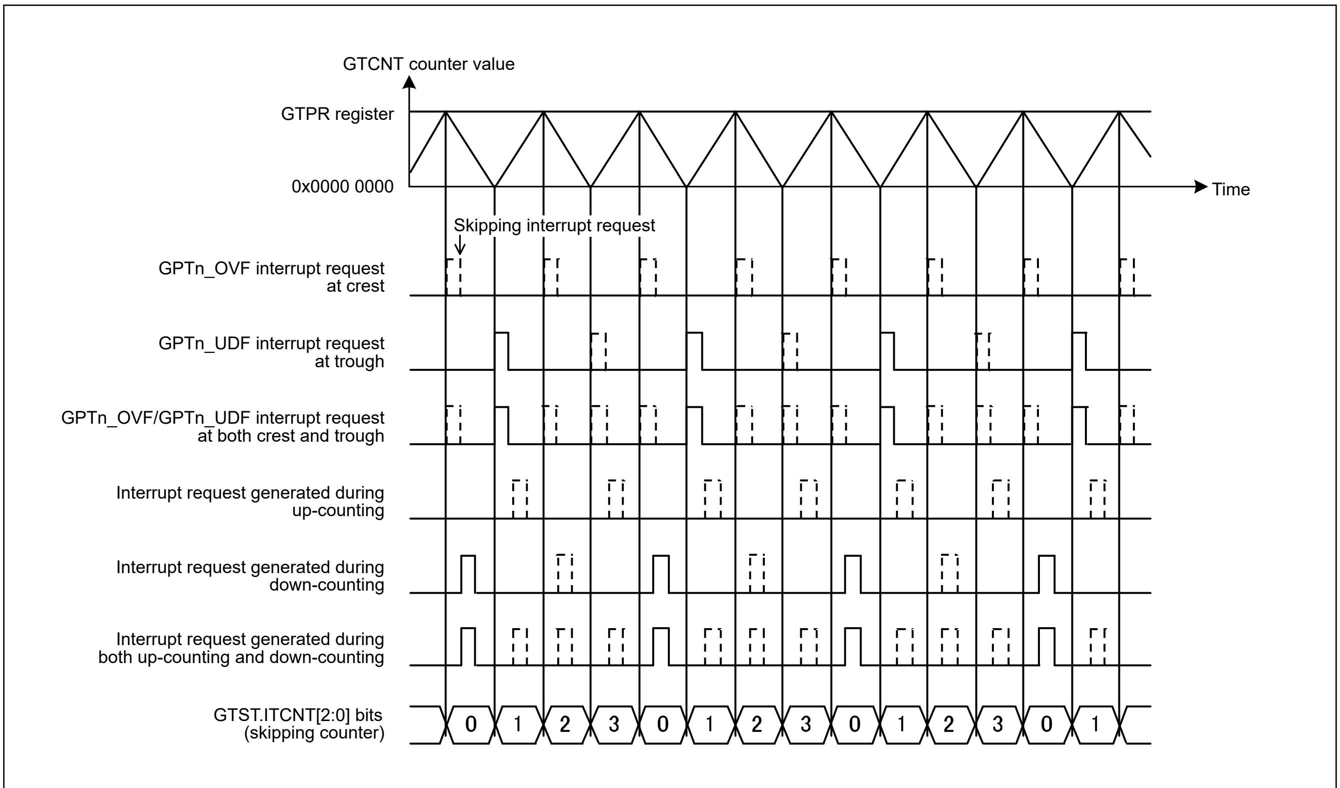


Figure 22.132 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Both Troughs and Crests, Skipping Count: 3, Skipping Started at Up-Counting)

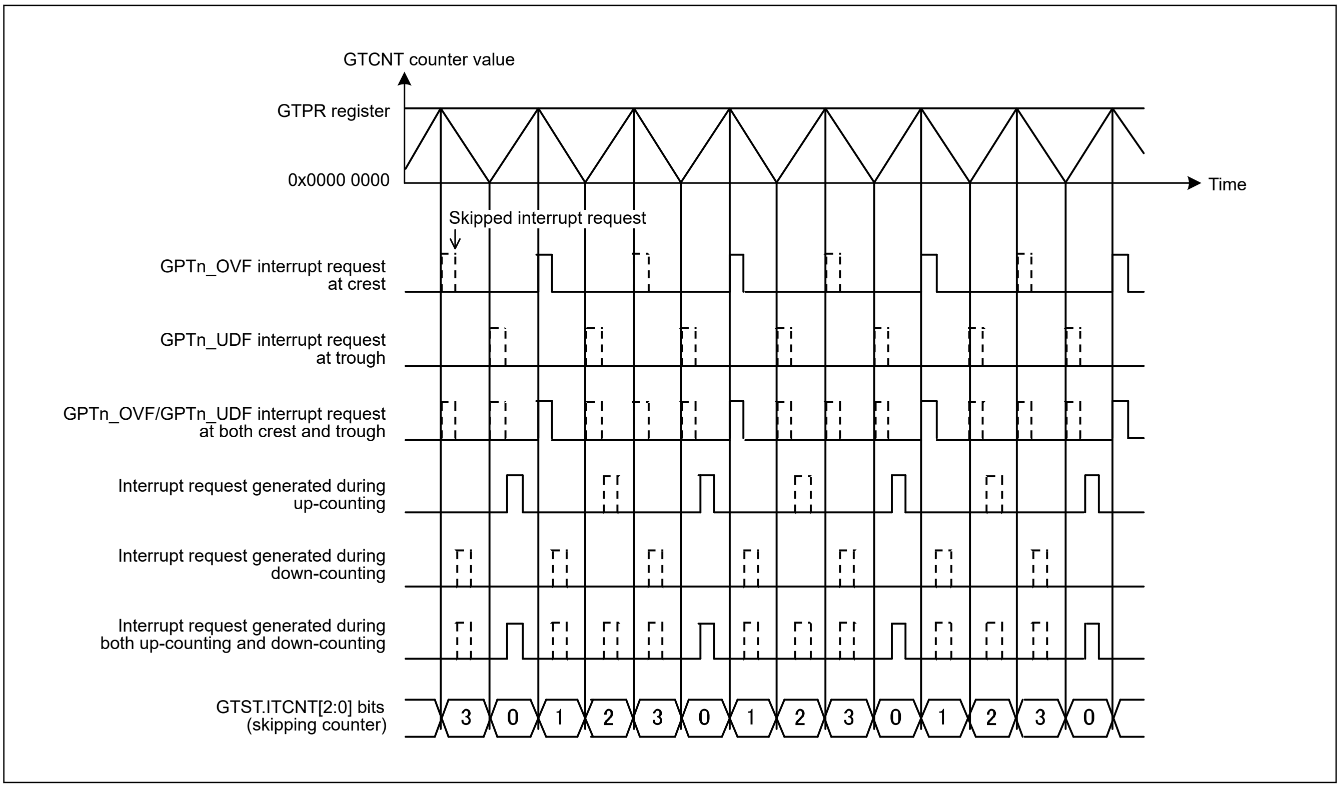


Figure 22.133 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Both Troughs and Crests, Skipping Count: 3, Skipping Started at Down-Counting)

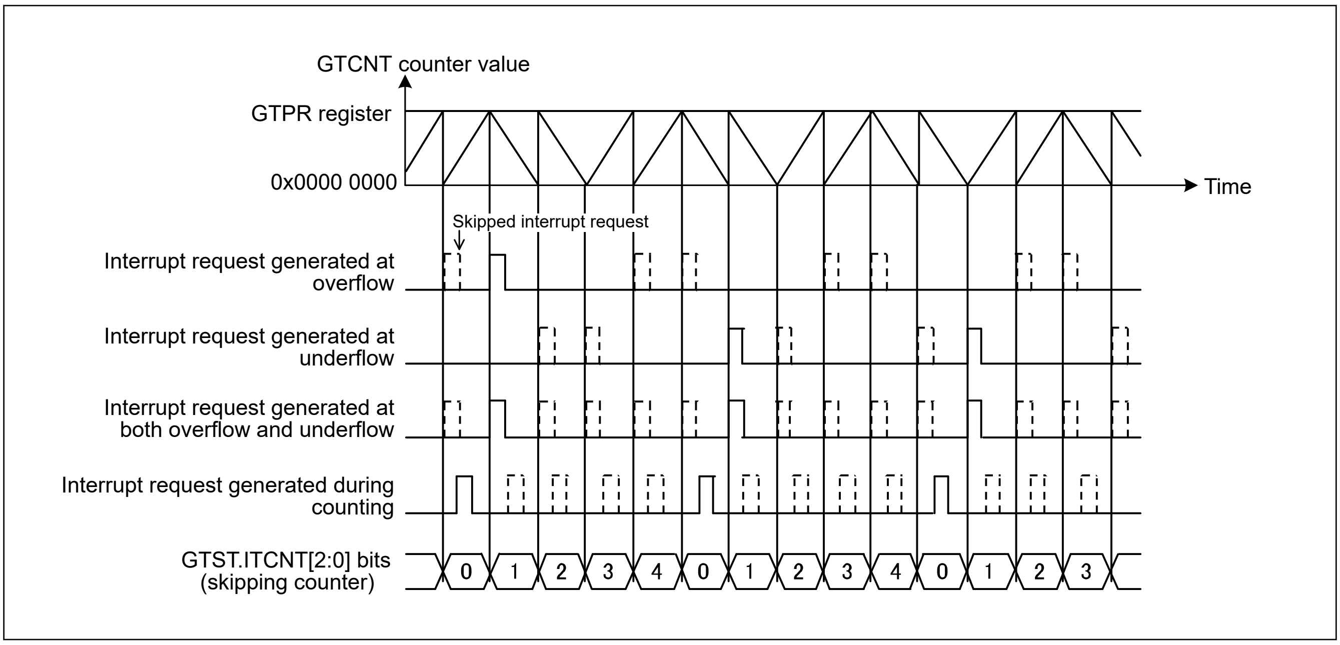


Figure 22.134 Example of Interrupt Skipping Function Operation (Saw Waves, Operation with Count Direction Changed, Counting and Skipping Both Overflows and Underflows, Skipping Count: 4)

22.4.3.2 Extended Interrupt Skipping Function

Overflow/underflow interrupt, compare match/input capture interrupt, A/D conversion start request, and buffer transfer can be skipped by counting the GTCNT counter overflow or underflow based on settings of the GTEITC, GTEITLI1, GTEITLI2 and GTEITLB registers.

Overflow/underflow interrupt, compare match/input capture interrupt, A/D conversion start request, and buffer transfer can individually be set for skipping and the skipping period using the GTEITLI1, GTEITLI2, and GTEITLB registers.

The skipping period is set, related to the operation of the two independent extended interrupt skipping counters (EITCNT1[3:0] and EITCNT2[3:0] bits in the GTEITC registers), as a period that either of the counter value for such skipping counters is other than 0 or other than the skipping count. The skipping period can also be set as a period that the both skipping counters are set as other than 0 or other than the skipping count for the counter value.

Figure 22.135 shows the counter operation for interrupt skipping by the GTITC register and extended interrupt skipping.

The counter operation for extended interrupt skipping is set by the GTEITC register.

The EITCNT1[3:0] bits are set for 0 as initial value and then they count the count source (in Figure 22.135, a crest is selected) selected by the extended interrupt counter 1 count source select bit (GTEITC.EIVTC1[1:0] bits), and repeat the count operation which returns to 0 when the skipping count (in Figure 22.135, count is 2) set by the extended interrupt skipping 1 skipping count setting bit (EIVTT1[3:0] bits) is achieved.

The EITCNT2[3:0] bits can be set for the initial value and they count the count source (in Figure 22.135, a trough is selected) selected by the extended interrupt counter 2 count source select bit (EIVTC2[1:0] bits), and repeat the count operation which return to 0 when the skipping count (in Figure 22.135 count is 2) set by the EIVTT2[3:0] bits is achieved. The initial value is set when the GTEITC register is written by the access of upper 16 bits or 32 bits while the setting for the extended interrupt skipping counter 2 is not counted (EIVTC2[1:0] bits are 00b) and when the write value to the EIVTC2[1:0] bits is other than 00b. When the initial value is set, the write value to the extended interrupt skipping counter 2 initial value bits (EITCNT2IV[3:0] bits) are set as the initial value for the EITCNT2[3:0] bits.

The extended interrupt skipping counter starts up-counting at the first count clock after the setting is modified from not counting to counting.

The interrupt skipping count counter (GTST.ITCNT[2:0] bits) for the interrupt skipping function set by the GTITC register is reset to 000b when the GTCNT counter operation is stopped: however, the EITCNT1[3:0] and EITCNT2[3:0] bits for the extended interrupt skipping function retain the value even after the GTCNT counter operation is stopped, and the counting can be resumed from the value before the counter is stopped. When values for the EITCNT1[3:0] and EITCNT2[3:0] bits are to be reset (0000b), set the EIVTC1[1:0] and EIVTC2[1:0] bits to the setting for not counting (not skipping) (00b).

When the skipping count is to be changed, change the count after stopping the skipping counter operation (set either of the EIVTC1[1:0] bit or the EIVTC2[1:0] bit to 00b).

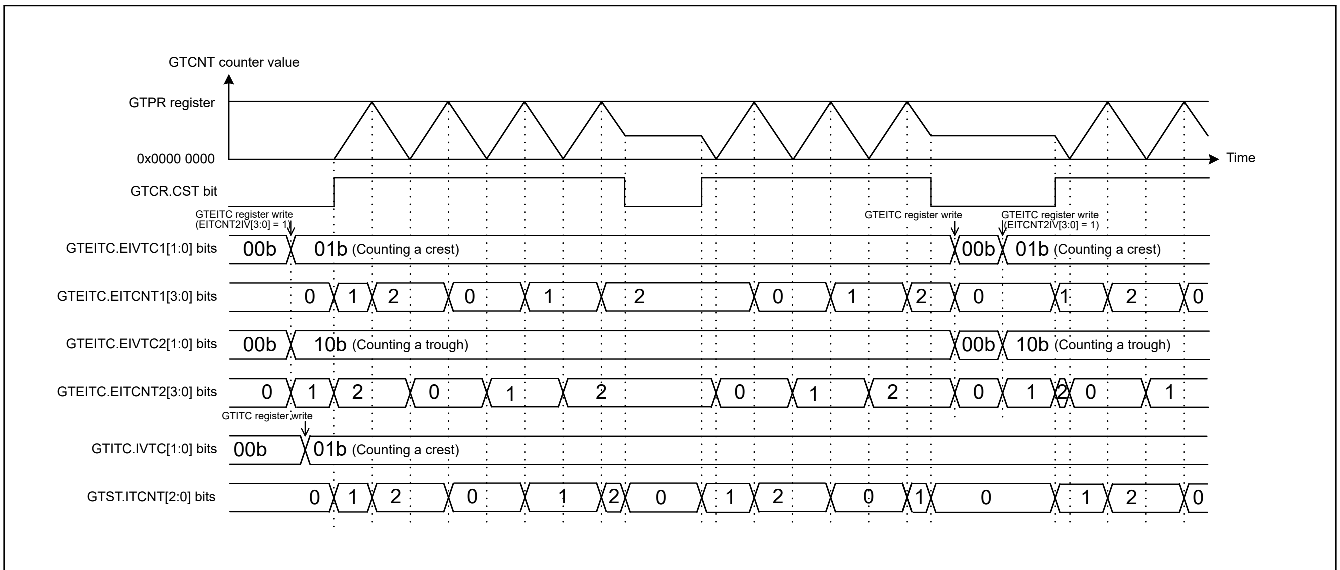


Figure 22.135 Example of Counter Operation for Interrupt Skipping

Interrupt skipping by the GTEITL1 register and A/D conversion start request skipping by the GTEITLI2 register can be performed simultaneously with skipping by the GTITC register or GTADCMSC register. A skipping period in this case is represented by OR-ed skipping periods of respective registers.

Figure 22.136 shows corresponding interrupt skipping operations by different registers are performed simultaneously.

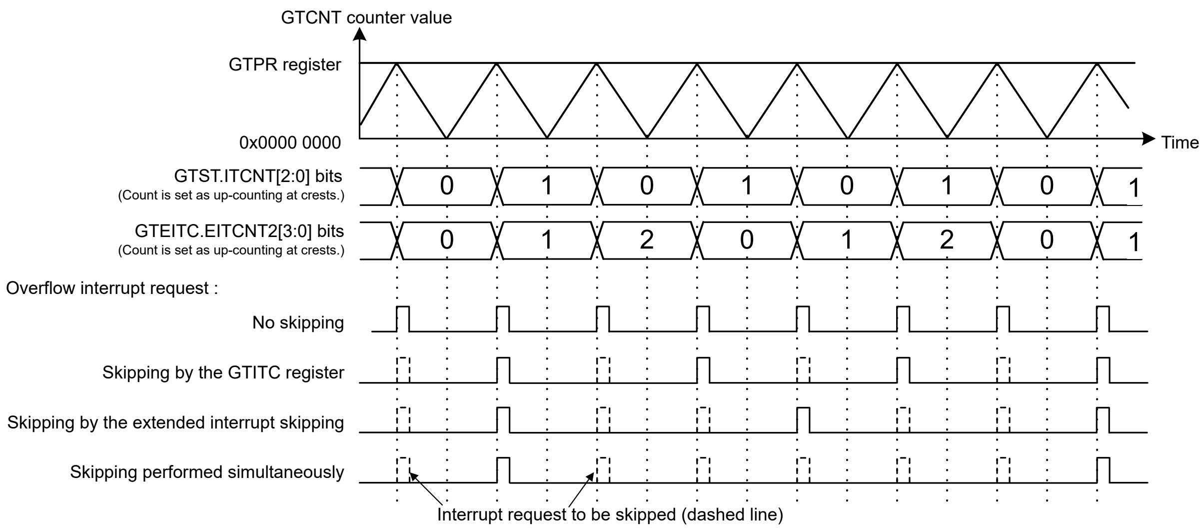


Figure 22.136 Example of Interrupt Skipping Operation (Skipping by the GTITC Register: Counting crests, Extended Interrupt Skipping: EIVTC1[1:0] bits = 00b, EIVTC2[1:0] bits = 01b, EITLV[2:0] bits = 010b)

When the extended interrupt skipping selected by GTEITLI1 is performed, the relevant status flag is also skipped. The skipping function is continued even if the status flag is set to 1.

When A/D conversion start request skipping selected by GTEITLI2 is performed, the relevant status flag is also skipped. The skipping function is continued even if the status flag is set to 1.

Skipping of updating of status flag and ELC event source outputs corresponding to A/D conversion start request for which skipping can be set up in the GTEITLI2 register is only based on the settings of the GTITC register and the extended interrupt skipping register, and has no connection with the setting of A/D conversion start request enable bit in the GTINTAD register. The A/D conversion start request enable bit is used only for A/D conversion start request output(ELC event source output) after skipping.

A buffer transfer skipping by the GTEITLB register is performed in all of the buffer operations which is enabled in the GTBER and GTDTCR registers and GTBER2 register, or all buffer operations performed by saw-wave one-shot pulse mode or triangle-wave PWM mode 3 or complementary PWM mode (excludes buffer transfer from GTCCRC, GTCCRE to GTCCRA).

An interrupt skipping and a buffer transfer skipping are operated individually. An interrupt output without performing a buffer transfer and performing a buffer transfer without an interrupt output can also be operated.

Figure 22.137 to Figure 22.144 show examples of extended skipping function operation.

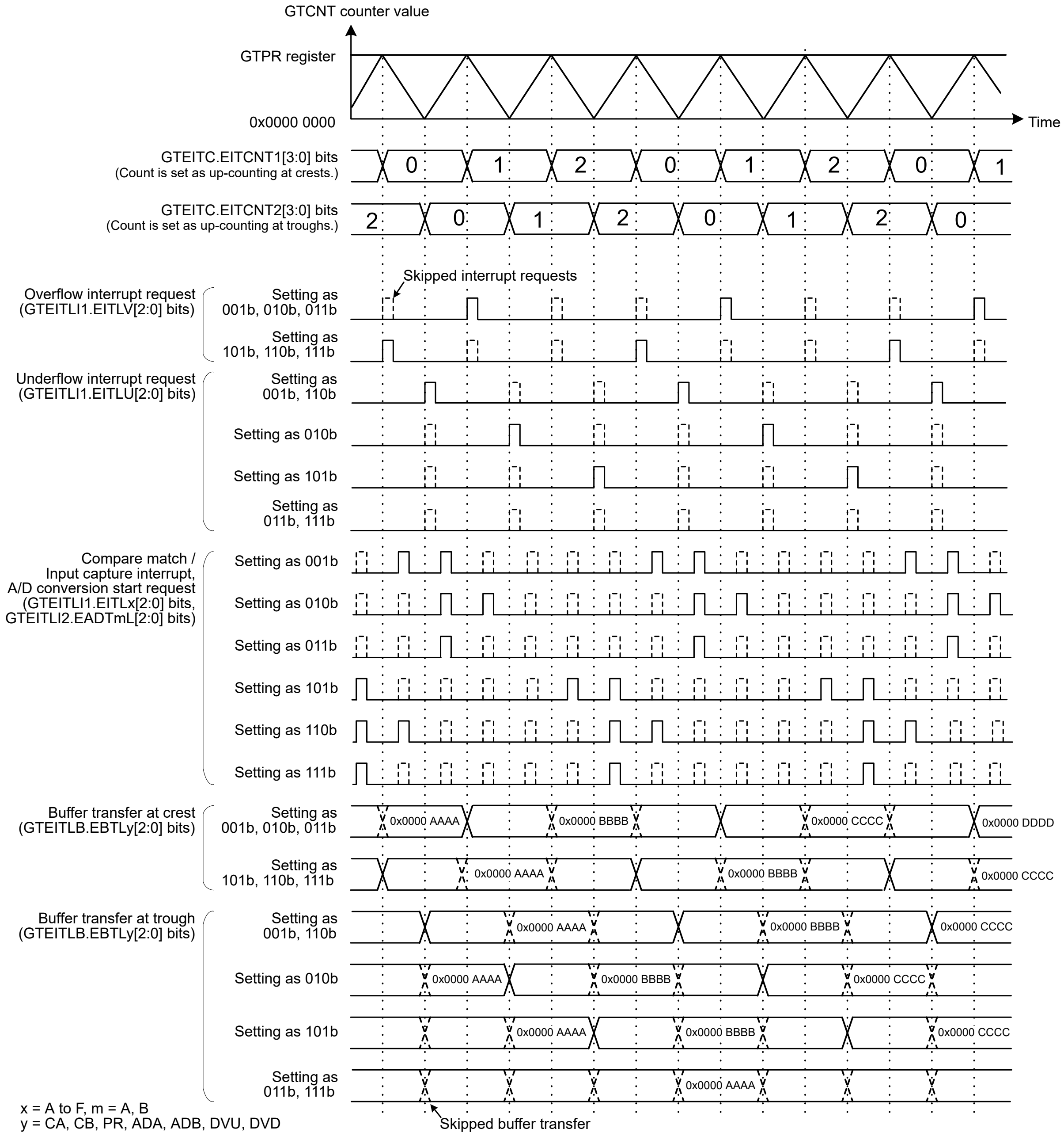


Figure 22.137 Example of Extended Interrupt Skipping Function (Triangle wave, Counting Crests, Extended Interrupt Skipping 1 Skipping Count: 2, Counting Troughs, Extended Interrupt Skipping 2 Skipping Count: 2, Extended Interrupt Skipping Counter 2 Initial Value: 0)

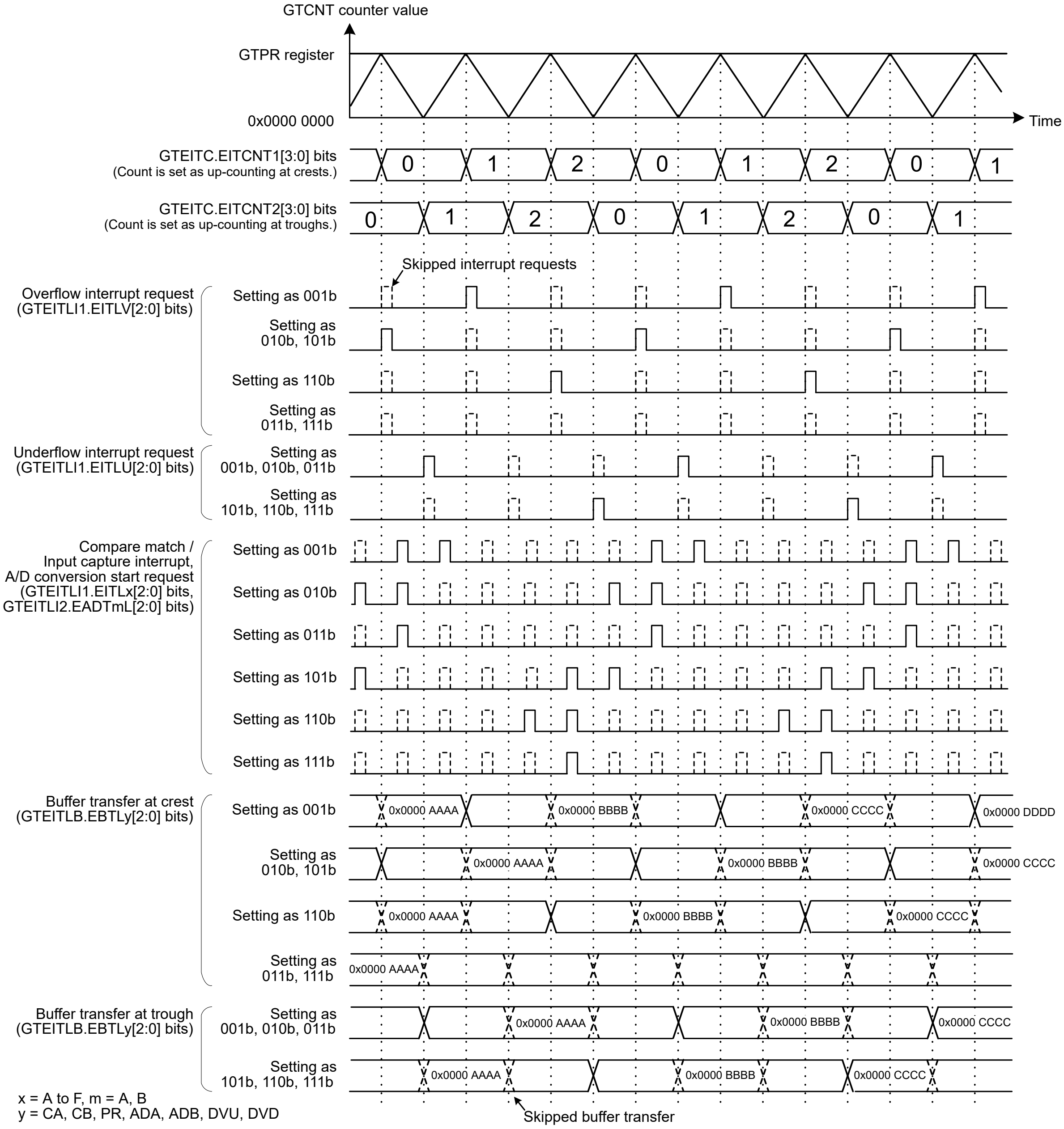


Figure 22.138 Example of Extended Interrupt Skipping Function (Triangle Waves, Count Crests by the Extended Interrupt Skipping 1 with Skipping Count: 2, Count Troughs by the Extended Interrupt Skipping 2 with Skipping Count: 2, Extended Interrupt Skipping Counter 2 Initial Value: 1)

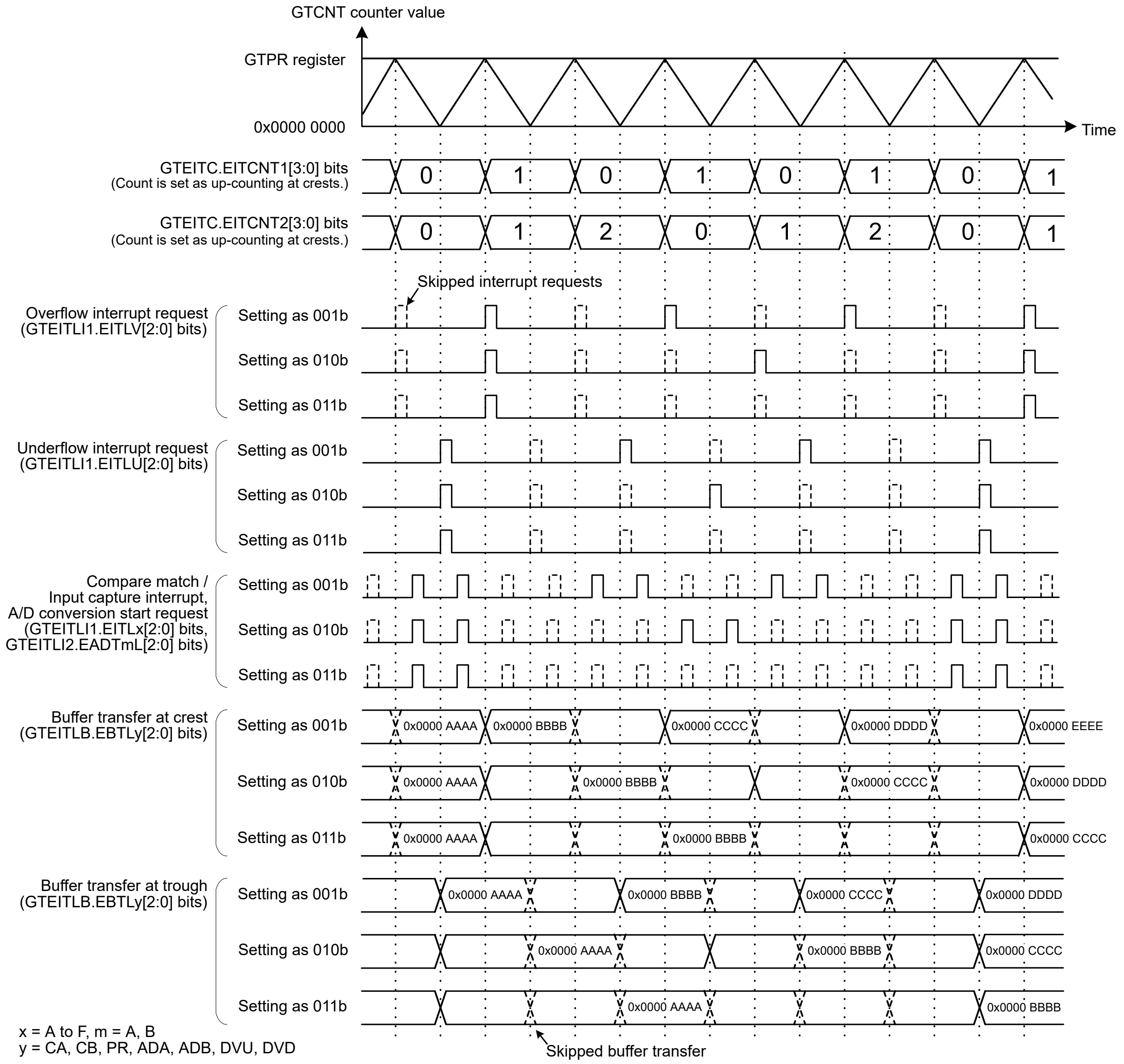


Figure 22.139 Example of Extended Interrupt Skipping Function Operation (Triangle Waves, Counting Crests, Extended Interrupt Skipping 1 Skipping Count: 1, Counting Crests, Extended Interrupt Skipping 2 Skipping Count: 2, Extended Interrupt Skipping Counter 2 Initial Value: 0, Skipped in the Period other than GTEITC.EITCNTk bits (k = 1, 2) as 0)

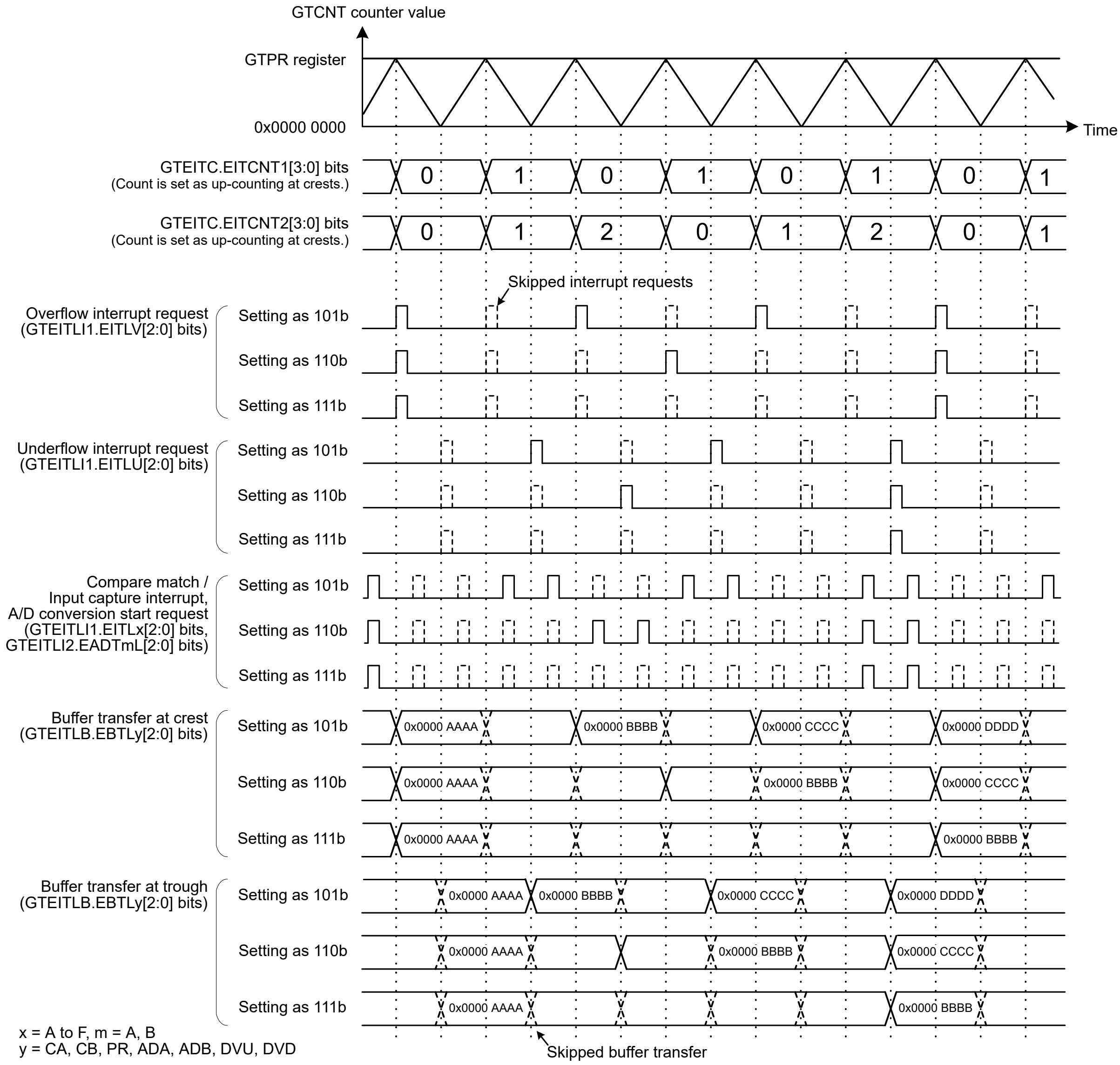


Figure 22.140 Example of Extended Interrupt Skipping Function Operation (Triangle Waves, Counting Crests, Extended Interrupt Skipping 1 Skipping Count: 1, Counting Crests, Extended Interrupt Skipping 2 Skipping Count: 2, Extended Interrupt Skipping Counter 2 Initial Value: 0, Skipped in the Period other than GTEITC.EITCNTk bits (k = 1, 2) as GTEITC.EIVTTk bits.)

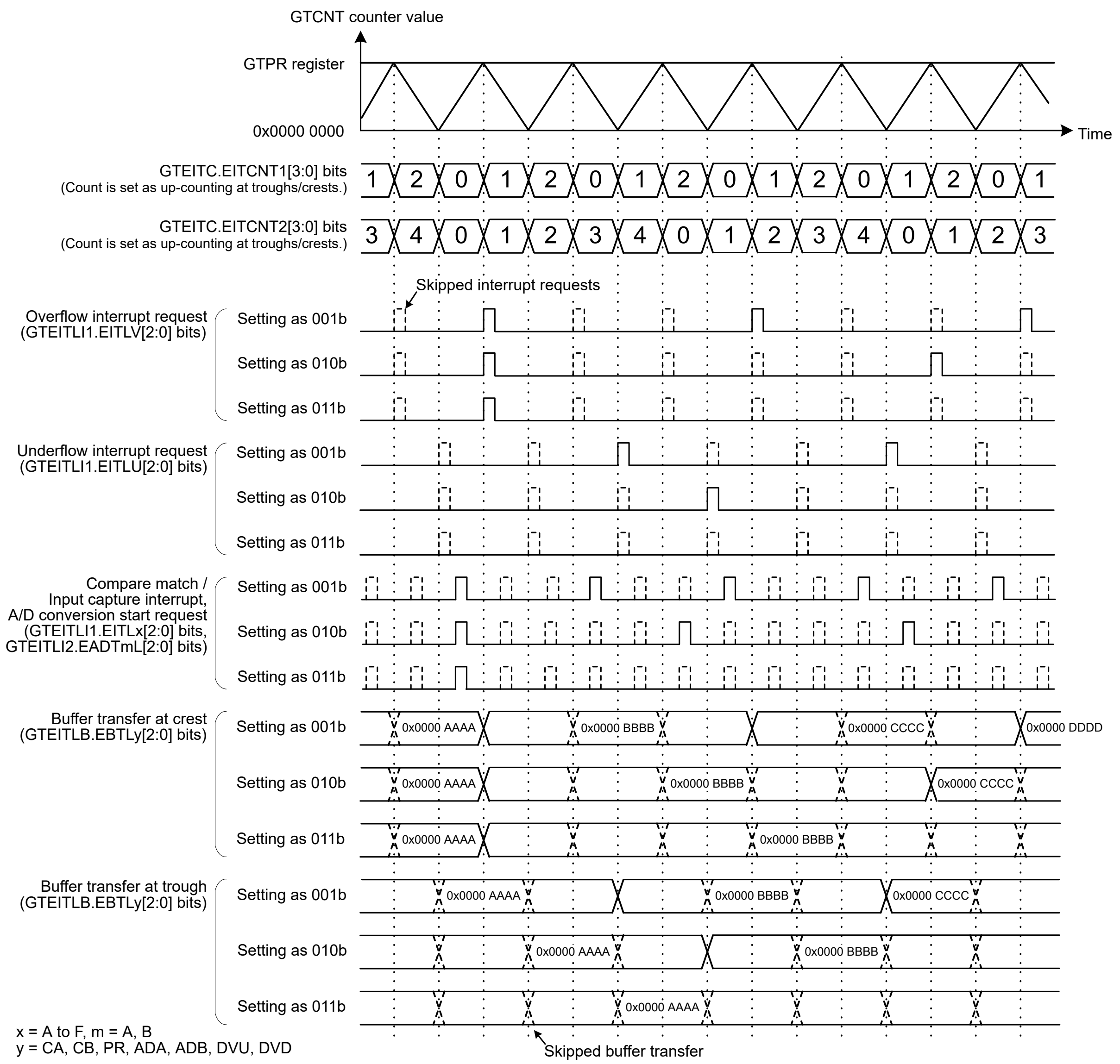


Figure 22.141 Example of Extended Interrupt Skipping Function Operation (Triangle Waves, Counting both Crests and Troughs, Extended Interrupt Skipping 1 Skipping Count: 2, Counting both Crests and Troughs, Extended Interrupt Skipping 2 Skipping Count: 4, Extended Interrupt Skipping Counter 2 Initial Value: 0, Skipped in the Period other than GTEITC.EITCNTk bits (k = 1, 2) as 0)

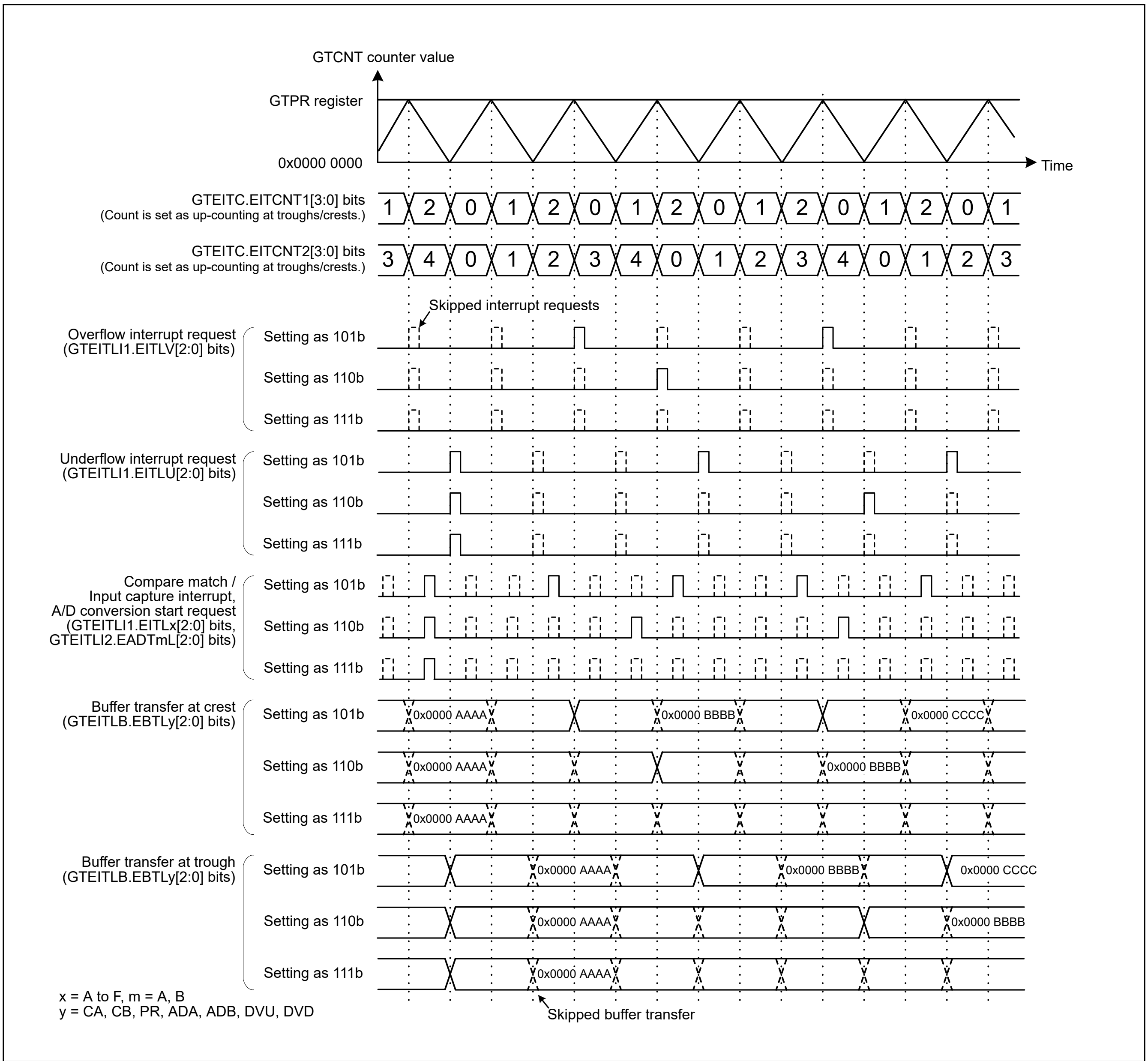


Figure 22.142 Example of Extended Interrupt Skipping Function Operation (Triangle Waves, Counting both Crests and Troughs, Extended Interrupt Skipping 1 Skipping Count: 2, Counting both Crests and Troughs, Extended Interrupt Skipping 2 Skipping Count: 4, Extended Interrupt Skipping Counter 2 Initial Value: 0, Skipped in the Period other than GTEITC.EITCNTk bits (k = 1, 2) as GTEITC.EIVTTk bit.)

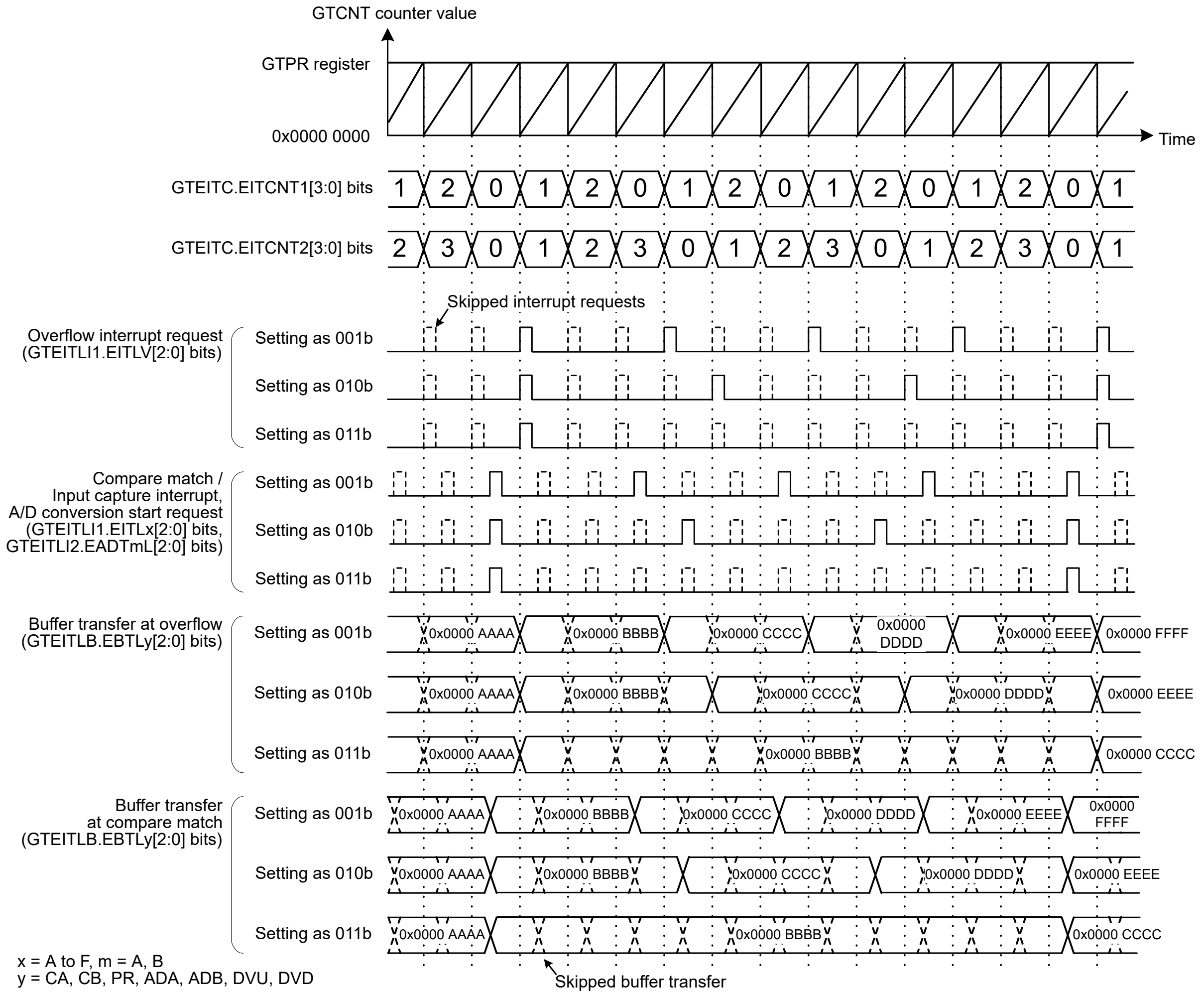


Figure 22.143 Example of Extended Interrupt Skipping Function Operation (Up-Counting in Saw Waves, Counting Overflow, Extended Interrupt Skipping 1 Skipping Count: 2, Counting Overflow, Extended Interrupt Skipping 2 Skipping Count: 3, Extended Interrupt Skipping Counter 2 Initial Value: 0, Skipped in the Period other than GTEITC.EITCNTk bits (k = 1, 2) as 0)

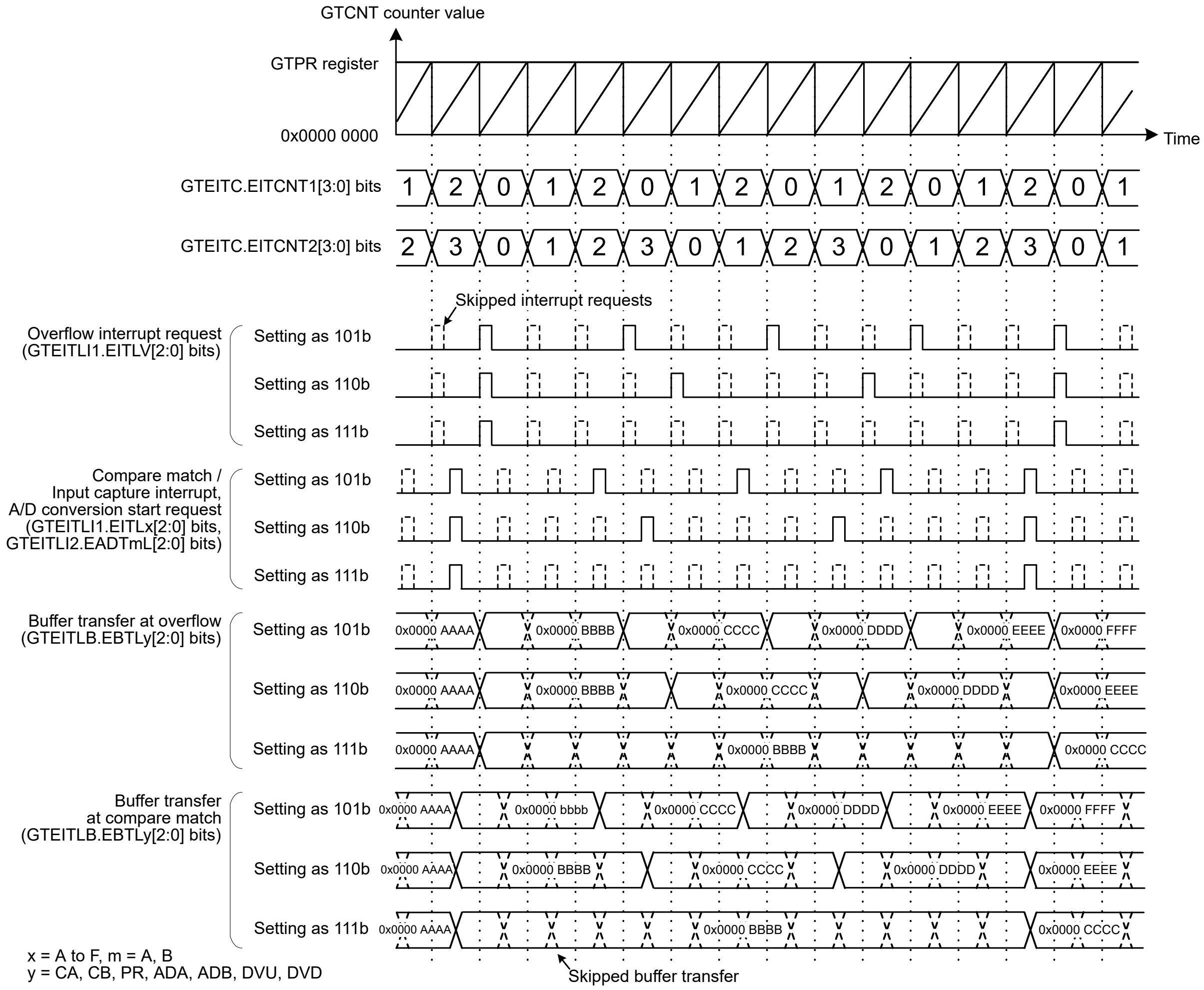


Figure 22.144 Example of Extended Interrupt Skipping Function Operation (Up-Counting Saw Waves, Counting Overflow, Extended Interrupt Skipping 1 Skipping Count: 2, Counting Overflow, Extended Interrupt Skipping 2 Skipping Count: 3, Extended Interrupt Skipping Counter 2 Initial Value: 0, Skipped in the Period other than GTEITC.EITCNTk bits (k = 1, 2) as GTEITC.EIVTTk bits.)

Figure 22.145 shows an example of the extended interrupt skipping operation on the input capture. When the setting is for the input capture operation setting (GTCR.ICDS bit = 0) at count stop of the GTCNT counter, the interrupt by the input capture and the extended skipping on buffer transfer is enabled even at count stop of the GTCNT counter.

When the input capture is generated at count stop of the GTCNT counter by setting the ICDS bit as 0, an interrupt and a buffer transfer are skipped if the skipping counter value is the same with the skipping period set in a corresponding interrupt skipping function select bit.

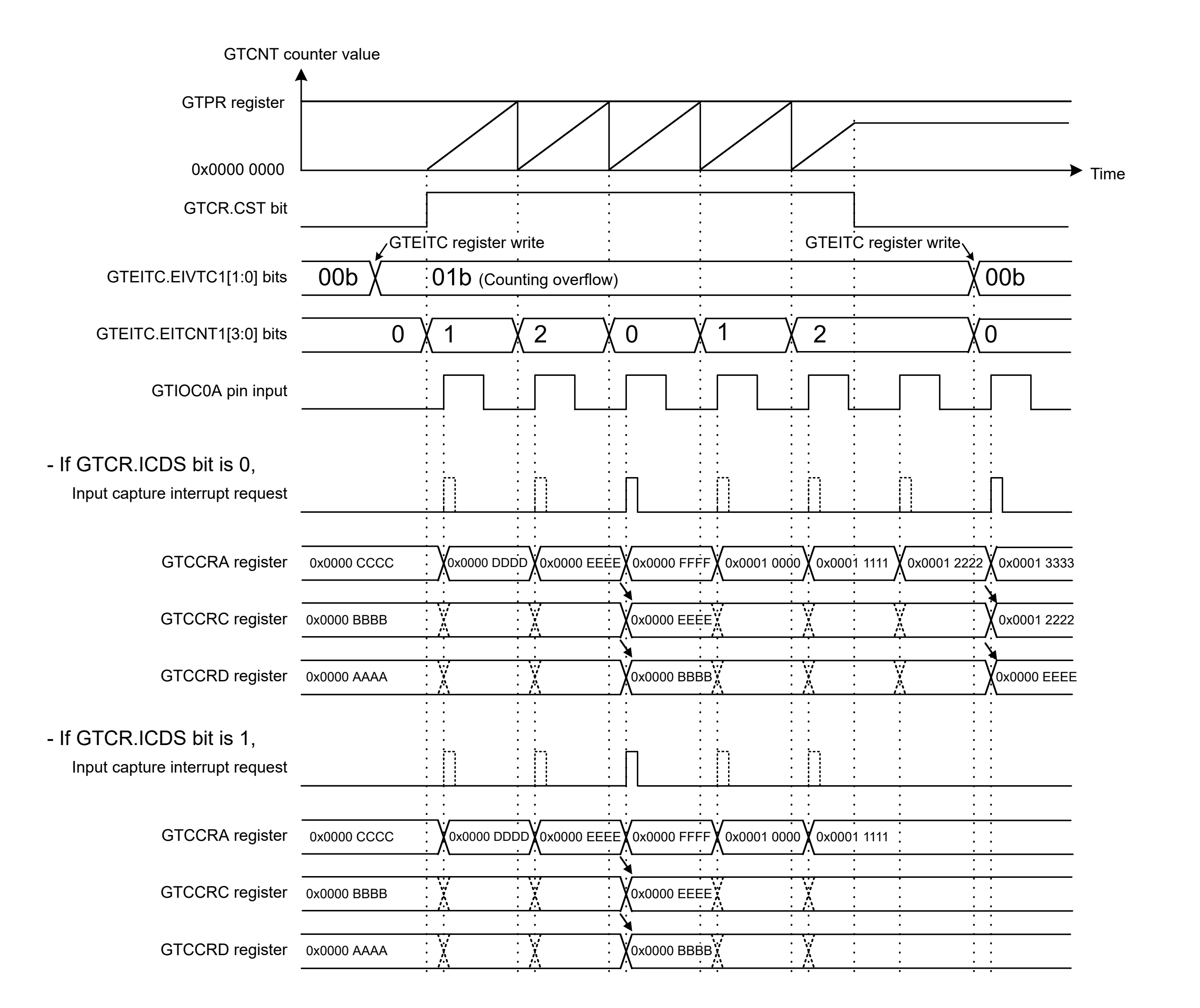


Figure 22.145 Example of Input Capture Operation in the Extended Interrupt Skipping Function Operation (Up-Counting Saw Waves, Counting Overflow, Extended Interrupt Skipping 1 Skipping Count: 2, Skipped in the Period other than the EITCNT1 as 0, Input Capture at the Input Rising)

Table 22.72 shows an example of setting the extended interrupt skipping.

The extended interrupt skipping counter 2 initial value is set by the written value to the ETICNT2IV[3:0] bits which is applied to change the setting from not counting the extended interrupt skipping counter 2 count source (GTEITC.EIVTC2[1:0] bits = 00b) to counting (EIVTC2[1:0] bits = other than 00b). Writing to the extended interrupt skipping counter 2 initial value bit (ETICNT2IV[3:0] bits) is performed only when the setting of the above-mentioned extended interrupt skipping counter 2 initial value is written.

Table 22.72 Example for Setting the Extended Interrupt Skipping (1 of 2)

No.	Step Name	Description
1	Set GTCNT counter operation, Set buffer operation, Set compare match value	See section 22.3.2. Buffer Operation, section 22.3.3. PWM Output Operating Mode, and so on.
2	Set the extended interrupt skipping function	Select the skipping counter used for skipping and the skipping period with the skipping function select bit for the interrupt, for the A/D conversion start request, and for a buffer transfer, all which are to be skipped, in the GTEITLI1, GTEITLI2, and GTEITLB registers respectively.

Table 22.72 Example for Setting the Extended Interrupt Skipping (2 of 2)

No.	Step Name	Description
3	Set the extended skipping counter	With the GTEITC register, set a count source of the skipping counter used for skipping, skipping count, and the skipping counter 2 initial value as the following order. Set EIVTCk[1:0] bits (k = 1,2) as value other than 00b, and set EIVTTk[3:0] bits as value other than 0000b. When the skipping counter 2 is used, change EIVTC2[1:0] bits from 00b to value other than 00b as well as set EITCNT2IV[3:0] bits to the skipping counter 2 initial value.
4	Start count operation	Set GTCR.CST to 1 to start count operation.
5	Set buffer value for each cycle	See section 22.3.2. Buffer Operation, section 22.3.3. PWM Output Operating Mode, and so on.

22.4.3.3 A/D conversion Start Request Compare Match Skipping Function

The A/D conversion start request and GTADTR register buffer transfer can be skipped by counting the compare matches of GTADTRA register and GTADTRB register based on settings of the GTADCMSC register and GTADCMSS register.

The skipping period is set, related to the operation of the two independent A/D conversion start request compare match skipping counters (ADCMSCNT1[3:0] and ADCMSCNT2[3:0] bits in the GTADCMSC register), as a period that either of the counter value for such skipping counters is other than 0 or other than the skipping count. The skipping period can also be set as a period that the both skipping counters are set as other than 0 or other than the skipping count for the counter value.

Figure 22.146 shows the counter operation for A/D conversion start request compare match skipping.

The counter operation for A/D conversion start request compare match skipping is set by the GTADCMSC register.

The ADCMSCNT1[3:0] bits count the count source (in Figure 22.146, a crest is selected) selected by the A/D conversion start request compare match skipping counter 1 count source select bit (GTADCMSC.ADCMSC1[1:0] bits), and repeat the count operation which returns to 0 when the skipping count (in Figure 22.146, count is 2) set by the A/D conversion start request compare match skipping 1 skipping count setting bit (ADCMST1[3:0] bits) is achieved.

The ADCMSCNT2[3:0] bits count the count source (in Figure 22.146, a trough is selected) selected by the A/D conversion start request compare match skipping counter 2 count source select bit (GTADCMSC.ADCMSC2[1:0] bits), and repeat the count operation which returns to 0 when the skipping count (in Figure 22.146, count is 2) set by the A/D conversion start request compare match skipping 2 skipping count setting bit (ADCMST2[3:0] bits) is achieved.

The ADCMSCNTk[3:0] bits (k = 1, 2) can be set for the initial value. The initial value is set when the GTADCMSC register is written by the access of 16 bits or 32 bits while the setting for the A/D conversion start request compare match skipping counter k is not counted (ADCMSCk [1:0] bits are 00b) and the write value to the ADCMSCk[1:0] bits is other than 00b. When the initial value is set, the write value to the A/D conversion start request compare match skipping counter k initial value bits (ADCMSCNTkIV[3:0] bits) are set as the initial value for the ADCMSCNT2[3:0] bits.

The A/D conversion start request compare match skipping counter starts up-counting at the first count clock after the setting is modified from not counting to counting.

The ADCMSCNT1[3:0] and ADCMSCNT2[3:0] bits for the A/D conversion start request compare match skipping function retain the value even after the GTCNT counter operation is stopped, and the counting can be resumed from the value before the counter is stopped. When values for the ADCMSCNT1[3:0] and ADCMSCNT2[3:0] bits are to be reset (0000b), set the ADCMSC1[1:0] and ADCMSC2[1:0] bits to the setting for not counting (not skipping) (00b).

When the skipping count is to be changed, change the count after stopping the skipping counter operation (set either of the ADCMSC1[1:0] bit or the ADCMSC 2[1:0] bit to 00b).

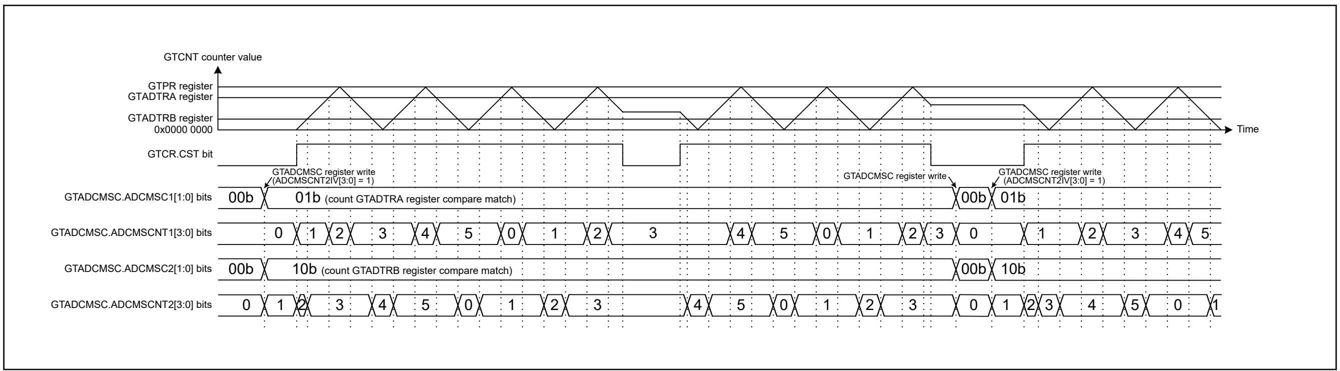


Figure 22.146 Example of A/D Conversion Start Request Compare Match Skipping Function Operation

A/D conversion start request skipping by the ADCMSC register can be performed simultaneously with skipping by the GTITC register or GTEITC register. A skipping period in this case is represented by OR-ed skipping periods of respective registers.

Figure 22.147 shows corresponding interrupt skipping operations and A/D conversion start request skipping operation are performed simultaneously.

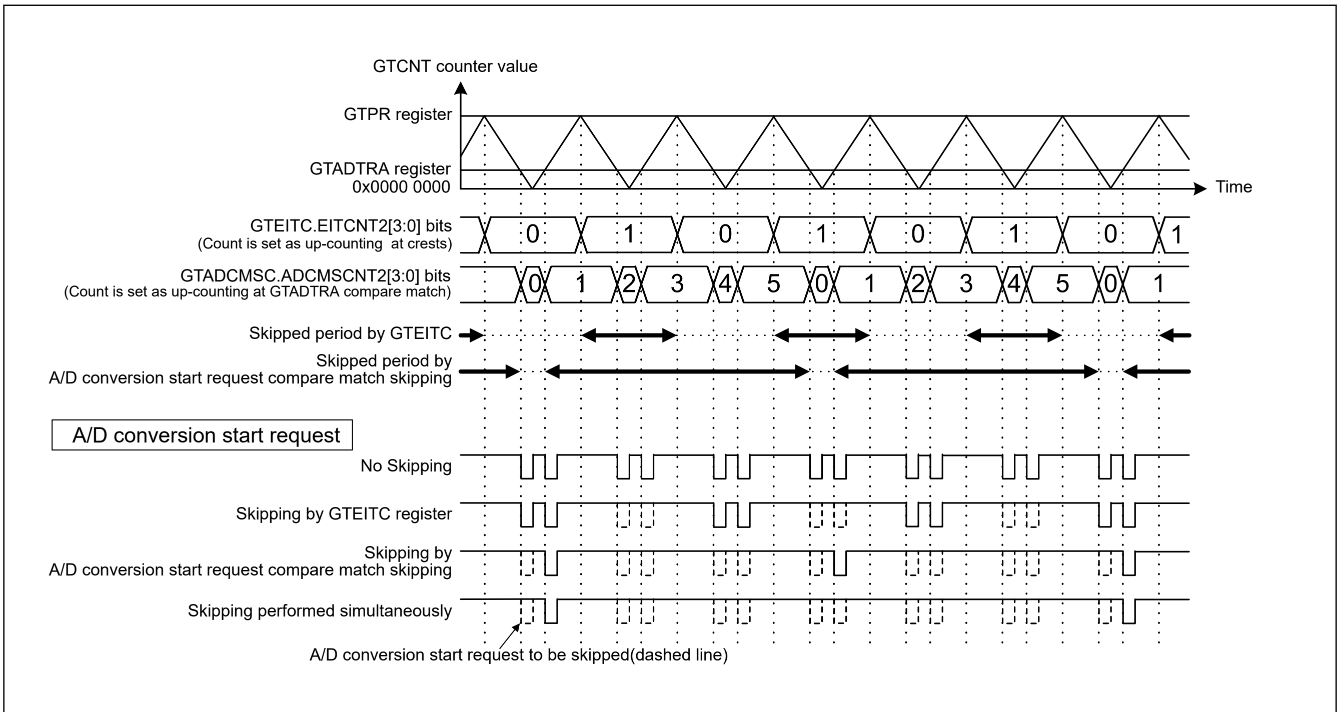


Figure 22.147 Example of A/D Conversion Start Request Compare Match Skipping Function Operation (Extended Interrupt Skipping: EIVTC2[1:0] = 01b, EADTAL[2:0] = 010b, A/D Conversion Start Request Compare Match Skipping: ADCMSC2[1:0] = 01b, ADCMSAL[2:0] = 010b)

When A/D conversion start request skipping which can be set by the GTADCMSS register is performed, updating of status flag and the ELC event output depend on the A/D conversion start request enable bit in the GTINTAD register. All the operations by A/D conversion start request which are set as disable by the GTINTAD register is not performed.

A buffer transfer skipping by the GTADCMSS register is performed in all of the buffer operations which is enabled in the GTBER and GTBER2 register, or all buffer operations performed by saw-wave one-shot pulse mode or triangle-wave PWM mode 3 or complementary PWM mode (excludes buffer transfer from GTCCRC, GTCCRE to GTCCRA).

An A/D conversion start request skipping and a buffer transfer skipping are operated individually.

Figure 22.148, Figure 22.149 show examples of extended skipping function operation.

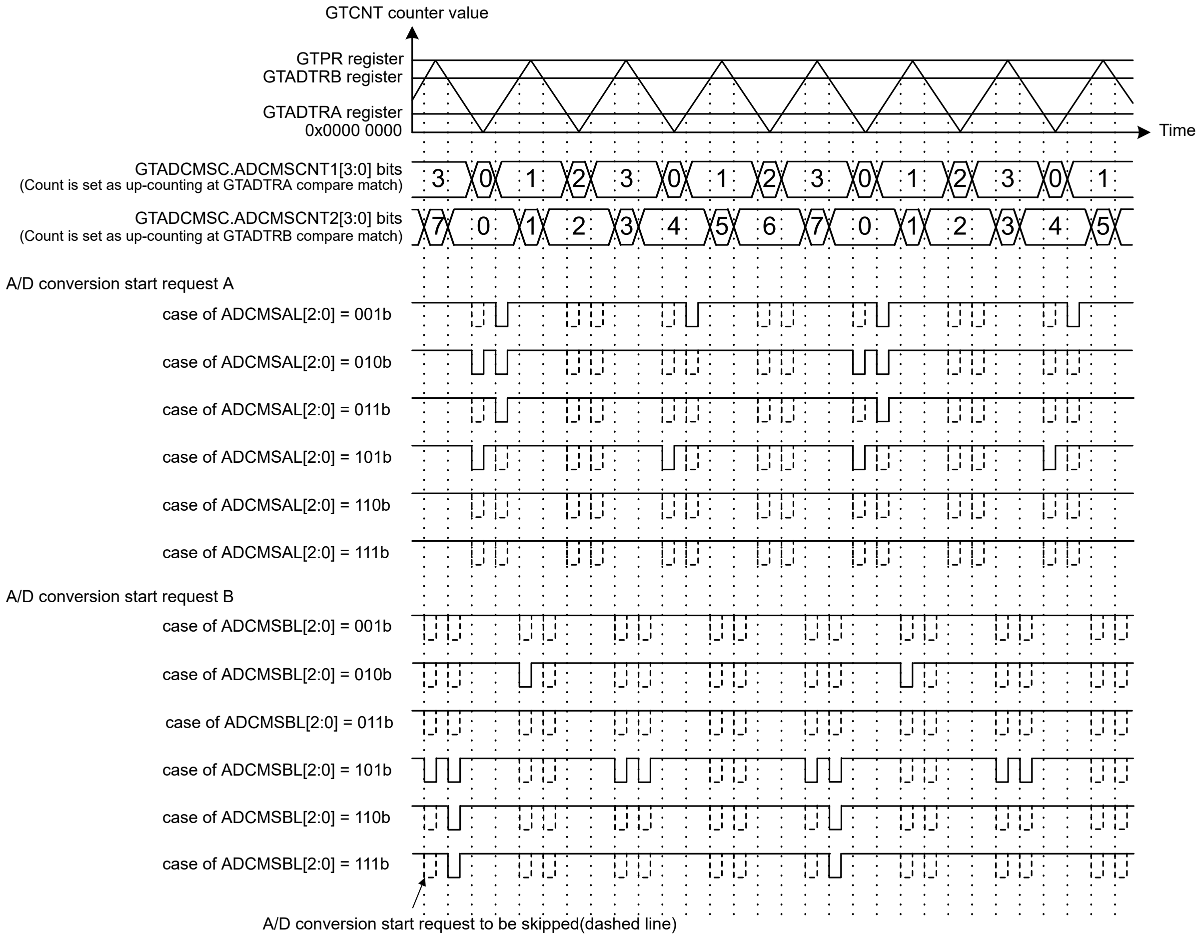


Figure 22.148 Example of A/D Conversion Start Request Compare Match Skipping Function Operation (Triangle wave, A/D Conversion Start Request Compare Match Skipping 1 Skipping Count: 3 Counting GTADTRA compare match, A/D Conversion Start Request Compare Match Skipping 2 Skipping Count: 7 Counting GTADTRB compare match)

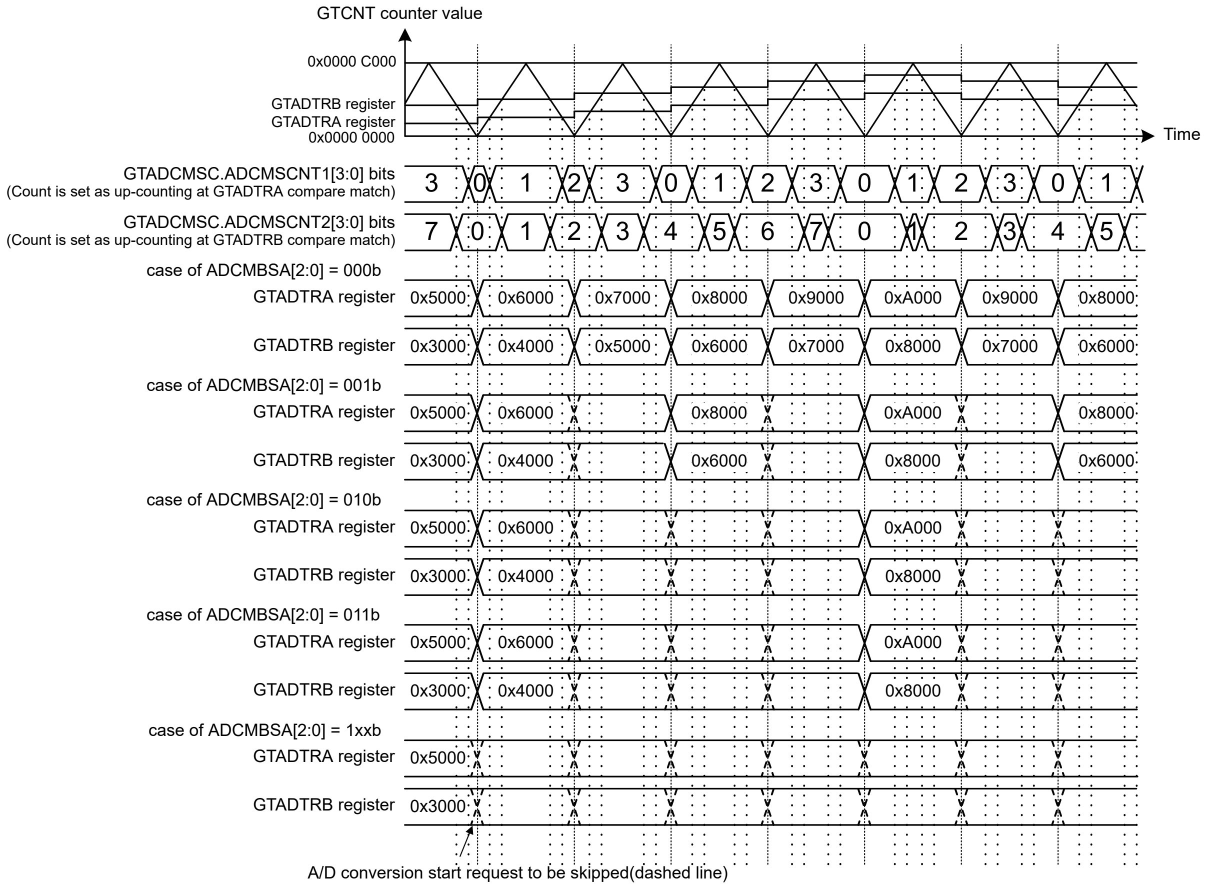


Figure 22.149 Example of A/D Conversion Start Request Compare Match Skipping Function Operation (Triangle wave, A/D Conversion Start Request Compare Match Skipping 1 Skipping Count: 3 Counting GTADTRA compare match, A/D Conversion Start Request Compare Match Skipping 2 Skipping Count: 7 Counting GTADTRB compare match, Buffer transfer of GTADTRA and GTADTRB at trough)

Table 22.73 shows an example of setting the A/D conversion start request compare match skipping.

The A/D conversion start request compare match skipping counter 2 initial value is set by the written value to the ADCMSCNT2IV[3:0] bits which is applied to change the setting from not counting the extended interrupt skipping counter 2 count source (GTADCMSC.ADCMSC2[1:0] bits = 00b) to counting (ADCMSC2[1:0] bits = other than 00b). Writing to the A/D conversion start request compare match skipping counter 2 initial value bit (ADCMSCNT2IV[3:0] bits) is performed only when the setting of the above-mentioned A/D conversion start request compare match skipping counter 2 initial value is written.

Table 22.73 Example for Setting the A/D Conversion Start Request Compare Match Skipping (1 of 2)

No.	Step Name	Description
1	Set GTCNT counter operation, Set buffer operation, Set compare match value	See section 22.3.2. Buffer Operation, section 22.3.3. PWM Output Operating Mode, section 22.5. A/D Conversion Start Request, and so on.
2	Set the A/D conversion start request compare match skipping function	Select the skipping counter and the skipping period with the skipping function select bit for the A/D conversion start request or the bit for the buffer transfer to be skipped in the GTADCMSS register.

Table 22.73 Example for Setting the A/D Conversion Start Request Compare Match Skipping (2 of 2)

No.	Step Name	Description
3	Set the A/D conversion start request compare match skipping counter k (k = 1, 2)	With the GTADCMSC register, set a count source of the skipping counter, skipping count, and the skipping counter k initial value as the following order. Set the GTADCMSC.ADCMSCk[1:0] bits as value other than 00b, and set the ADCMSTk[3:0] bits as value other than 0000b. Change the ADCMSCk[1:0] bits from 00b to value other than 00b as well as set the ADCMSCNTkIV[3:0] bits to the skipping counter k initial value.
4	Start count operation	Set GTCR.CST to 1 to start count operation.
5	Set buffer value for each cycle	See section 22.3.2. Buffer Operation, section 22.3.3. PWM Output Operating Mode, section 22.5. A/D Conversion Start Request, and so on.

22.5 A/D Conversion Start Request

The A/D conversion start request can be issued at a compare match between the GTCNT counter and the GTADTRA or GTADTRB register. Up-counting only, down-counting only, or both up-counting and down-counting can be specified by setting the GTINTAD register.

In complementary PWM mode, the A/D conversion start request can be issued at a compare match with the GTCNT counter of master channel.

During event count operation, the A/D conversion start request cannot be generated.

The A/D conversion start request is output directly to ADC16H.

The GTADTRA and GTADTRB registers each has two buffer registers. Buffer operation with the GTADTRA register used together with the GTADTBRA and GTADTDBRA registers, and buffer operation with the GTADTRB register used together with the GTADTBRB and GTADTDBRB registers can be performed.

The timing of the generation of requests to start A/D conversion can be monitored by an external pin. When the A/D conversion start request signal to be monitored is selected in the GTADSMR.ADSMSk bit (k = 0, 1) and when the output is enabled in the ADSMENk bit, a signal is output synchronized with a cycle frame of the timer used to generate the A/D conversion start request signal, of which the output is driven high at the generation of the A/D conversion start request signal by the GTADSMk pin, or at the end of the cycle of which the output is driven low. When a signal to request the start of A/D conversion is generated at the end of the cycle, the generation of this signal has priority in terms of monitoring output and the output remains at the high level till the end of the next cycle. The registers (GTADTRA and GTADTRB) that are sources of generating the A/D conversion start request signals and their counting directions can be checked by the A/D conversion start request flags (ADTRAUF, ADTRADF, ADTRBUF, and ADTRBDF) in the GTST register. When the output of the same A/D conversion start request signal monitoring output is enabled for multiple channels, ORed signals will be output from the GPT32.

Figure 22.150 shows an example of A/D conversion start request operation, Table 22.74 shows example for setting A/D conversion start request operation.

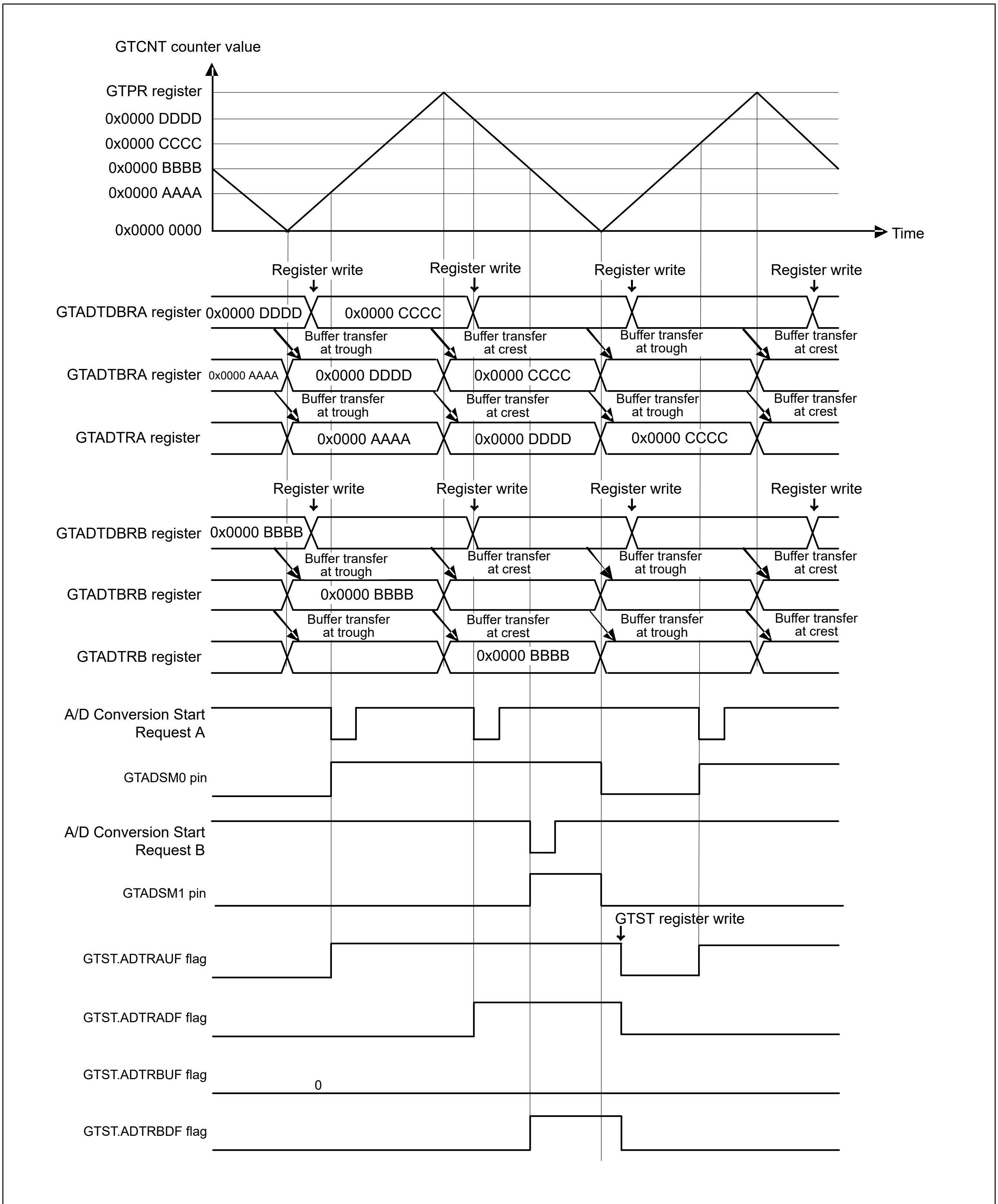


Figure 22.150 Example of A/D Conversion Start Request Timing Operation (Triangle Waves, Double Buffer Operation, Buffer Transfer at Both Troughs and Crests, A/D Conversion Start Request by GTADTRA Register at Both Up-Counting and Down-Counting, A/D Conversion Start Request by GTADTRB Register at Down-Counting, Monitoring of the GTADTRA Register Up-Counting by the GTADSM0 Pin, Monitoring of the GTADTRB Register Down-Counting by the GTADSM1 Pin)

Table 22.74 Example for Setting A/D Conversion Start Request Timing Operation

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 22.150, 100b, 101b, or 110b (0100b, 0101b, or 0110b) (triangle-wave PWM mode) is set.
2	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
3	Set cycle	Set the cycle in the GTPR register.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set buffer operation	Set buffer operation with the ADTTA[1:0], ADTTB[1:0], ADTDA, and ADTDB bits in the GTBER register. In Figure 22.150, ADTTA[1:0] = 11b, ADTTB[1:0] = 11b, ADTDA = 1, and ADTDB = 1.
6	Set compare match value	Set the A/D conversion start request point in the GTADTRA and GTADTRB registers.
7	Set buffer value	For buffer operation, set the A/D conversion start request point in one cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers. For double buffer operation, also set the A/D conversion start request point in two cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers.
8	Set A/D conversion start request for monitoring	Select the A/D conversion start request signal to be monitored with ADSMS0[1:0] and ADSMS1[1:0] bits in GTADSMR from GTADSM0 and GTADSM1 pins and enable output of the A/D conversion start request signal being monitored to ADSMEN0 and ADSMEN1 bits in GTADSMR. In Figure 22.150, ADSMS0[1:0] = 00b, ADSMS1[1:0] = 11b, ADSMEN0 = 1, and ADSMEN1 = 1.
9	Enable A/D conversion start request	Set to enable A/D conversion start request with the ADTRAUEN, ADTRADEN, ADTRBUEN, and ADTRBDEN bits in the GTINTAD register. In Figure 22.150, ADTRAUEN = 1, ADTRADEN = 1, ADTRBUEN = 0, and ADTRBDEN = 1.
10	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
11	Set buffer value for each cycle	For buffer operation, set the A/D conversion start request point in one cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers. For double buffer operation, also set the A/D conversion start request point in two cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers.

Figure 22.151 shows an example for A/D conversion start request timing operation.

This shows an example of the output of A/D conversion start request A as start source for the A/D converter. The A/D conversion start request A signal is output in response to a match in comparison with the GTADTRA register.

If GPT is operating with PCLKD and ADC is operating with PCLKA, A/D conversion start request A is passed to ADC on the next rising edge of PCLKA.

If GPT is operating with GPTCLK and ADC is also operating with GPTCLK, A/D conversion start request A is passed to ADC without delay.

For all other clock combinations, A/D conversion start request A is synchronized and passed to ADC.

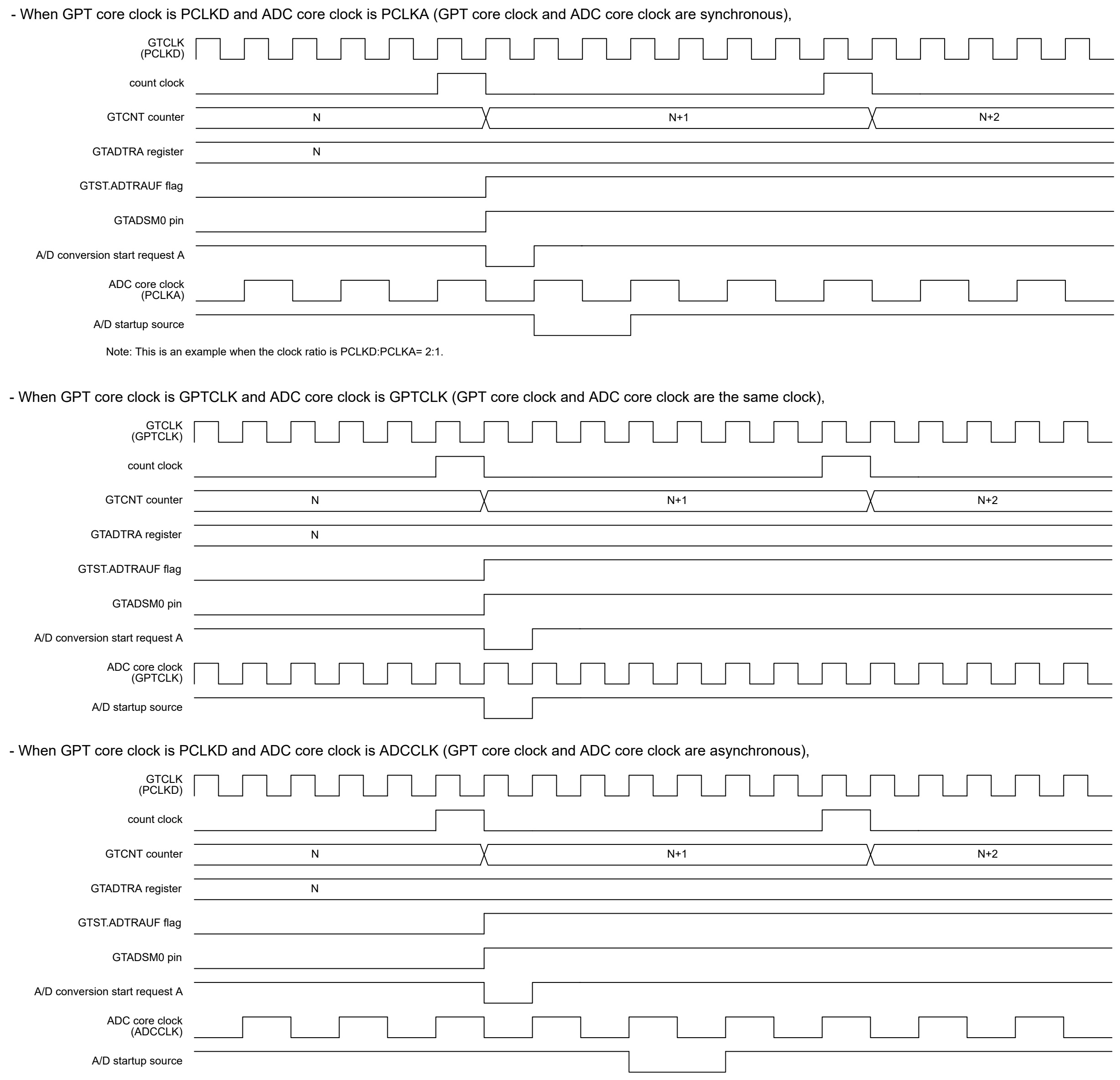


Figure 22.151 Example of A/D Conversion Start Request Timing Operation

For the restriction of A/D Conversion Start Request, see section 22.10.6. Interval of Interrupt Request.

22.6 Operations Linked by ELC

22.6.1 Event Signal Output to ELC

The GPT can perform operation linked with another module set in advance when its interrupt request signal is used as an event signal by the Event Link Controller (ELC).

The A/D conversion start request during up-counting/down-counting can be enabled/disabled individually with the A/D conversion start request enable bit to output events output to ELC.

The GPT has the following ELC event signals:

- Generation of compare match and input capture A interrupt (GPTn_CCMPA)
- Generation of compare match and input capture B interrupt (GPTn_CCMPB)
- Generation of compare match C interrupt (GPTn_CMPC)
- Generation of compare match D interrupt (GPTn_CMPD)

- Generation of compare match E interrupt (GPTn_CMPE)
- Generation of compare match F interrupt (GPTn_CMPF)
- Generation of overflow interrupt (GPTn_OVF)
- Generation of underflow interrupt (GPTn_UDF)
- Finish of period count function (GPTm_PC)

Note: n = 0 to 13

m = 0 to 3, 10 to 13

22.6.2 Event Signal Inputs from ELC

The GPT can perform the following operations in response to a maximum of 8 events from the ELC:

- Start counting, stop counting, clear counting
- Up-counting, down-counting
- Input capture.

See section 19, Event Link Controller (ELC) for the connection between the ELC and the event signal input.

22.7 Noise Filter Function

Each pin for use in input capture and Hall sensor input to the GPT is equipped with a noise filter. The noise filter samples input signals at the sampling clock and removes the pulses whose length is less than 3 sampling cycles.

The noise filter functionality includes enabling and disabling the noise filter for each pin and setting of the sampling clock for each channel.

Figure 22.152 shows the timing of noise filtering.

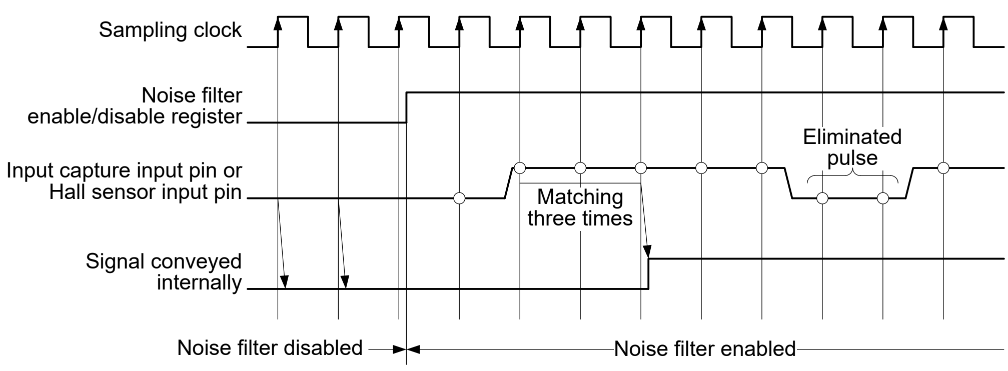


Figure 22.152 Timing of noise filtering

If noise filtering is enabled, the input capture operation or hall sensor input operation is performed on the edges of the noise filtered signal after a delay of (sampling interval × 2 + GTCLK) at the shortest. This is due to the noise filtering for the input capture input or hall sensor input.

22.8 Protection Function

22.8.1 Write-Protection for Registers

To prevent registers from being accidentally modified, registers can be write-protected in channel units by setting GTWP.WP. Write-protection can be set for the following registers:

GTSSR, GTPSR, GTCSR, GTUPSR, GTDNSR, GTICASR, GTICBSR, GTCR, GTUDDTYC, GTIOR, GTINTAD, GTST, GTBER, GTITC, GTCNT, GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, GTCCRF, GTPR, GTPBR, GTPDBR, GTADTRA, GTADTBRA, GTADTDBRA, GTADTRB, GTADTBRB, GTADTDBRB, GTDTCR, GTDVU,

GTDVD, GTDBU, GTDBD, GTSOS, GTSOTR, GTADSMR, GTEITC, GTEITLI1, GTEITLI2, GTEITLB, GTICLF, GTPC, GTADCMSC, GTADCMSS, GTBER2, GTOLBR, GTICCR.

Every bit in registers GTSTR, GTSTP and GTCLR which can update the corresponding registers in other channels and can be updated by any of the corresponding registers in other channels conversely, can be protected by setting the GTWP.STRWP, STPWP, and CLRWP bits, respectively, per channel.

Likewise, writing to the GTSECSR and GTSECR registers, which can control all channels by writing to the GTSECSR and GTSECR registers of a given channel, can by enabled or disabled by the setting of the GTWP.CMNWP bit.

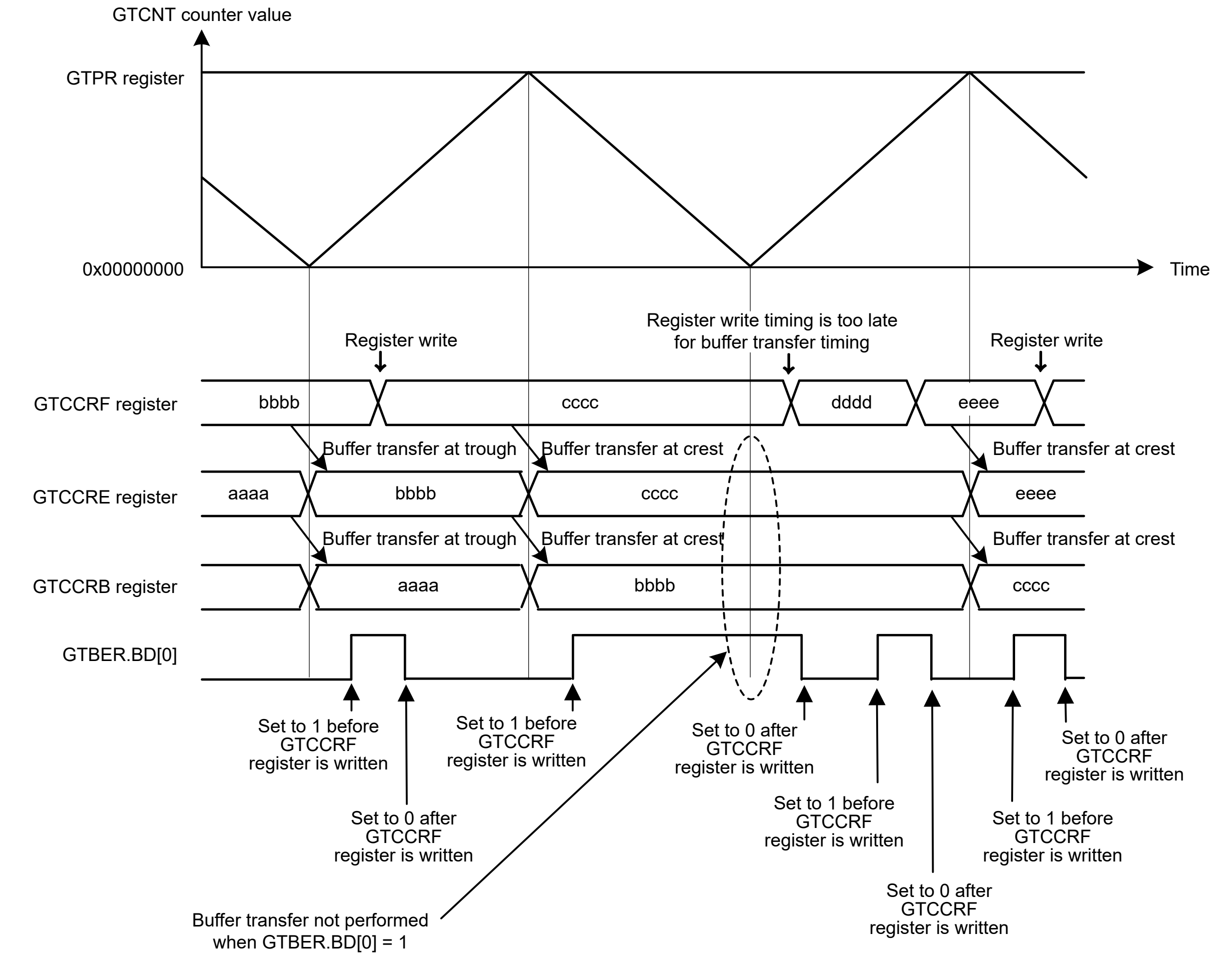
Protection using the GTWP register is only for write operations by the CPU. This protection does not cover updates to registers that occur in association with CPU writes.

22.8.2 Disabling of Buffer Operation

If the timing of the buffer register write is delayed relative to the timing for the buffer transfer, buffer operation can be suspended with the GTBER.BD[3], BD[2], BD[1] and BD[0] bits settings. Specifically, buffer transfer can be temporarily disabled even though a buffer transfer condition is generated during buffer register write, by setting the BD[3], BD[2], BD[1] and BD[0] bits to 1 (buffer operation disabled) before buffer register write, and setting the bits to 0 (buffer operation enabled) after completion of writing to all the buffer registers.

The BD[3], BD[2], BD[1] and BD[0] bits can be set on channel basis by writing directly to the GTBER register or it can be set to 0 simultaneously by setting the GTSECR register for multiple channels which were set by the GTSECSR register.

Figure 22.153 shows an example of operation for disabling buffer operation by writing to the GTBER register.



Note: The base address of registers are the same.

Figure 22.153 Example of operation for disabling buffer operation with triangle waves, double buffer operation, and buffer transfer at both troughs and crests

22.8.2.1 Simultaneous Control of Buffer Operations of Multiple Channels

The GTBER.BD bit can be set by writing directly to the GTBER register per channel or by making settings in the GTSECR register for multiple channels that have already set in the GTSECSR register.

Follow the procedure below to simultaneously set the GTBER.BD bits of multiple channels.

1. Select the channels for simultaneously setting by the GTSECSR register
   Set the GTSECSR register so that the values at the bit positions for the corresponding channels for simultaneously setting of the GTBER.BD bits become 1. All GTSECSR registers can be updated by writing to the GTSECSR register of any channel.
2. Simultaneously set the GTBER.BD bits by updating the GTSECR register
   In the GTSECR register, set the operation of the GTBER.BD bits (enabling or disabling of buffer operation) which are to be simultaneously set. Writing to a GTSECR register from any channel updates the GTBER.BD bits in all channels corresponding to the bits set as 1 in the GTSECSR register, in accordance with the value of the GTSECR register.

Figure 22.154 and Figure 22.155 show examples of simultaneously controlling the enabling or disabling of buffer operation for multiple channels.

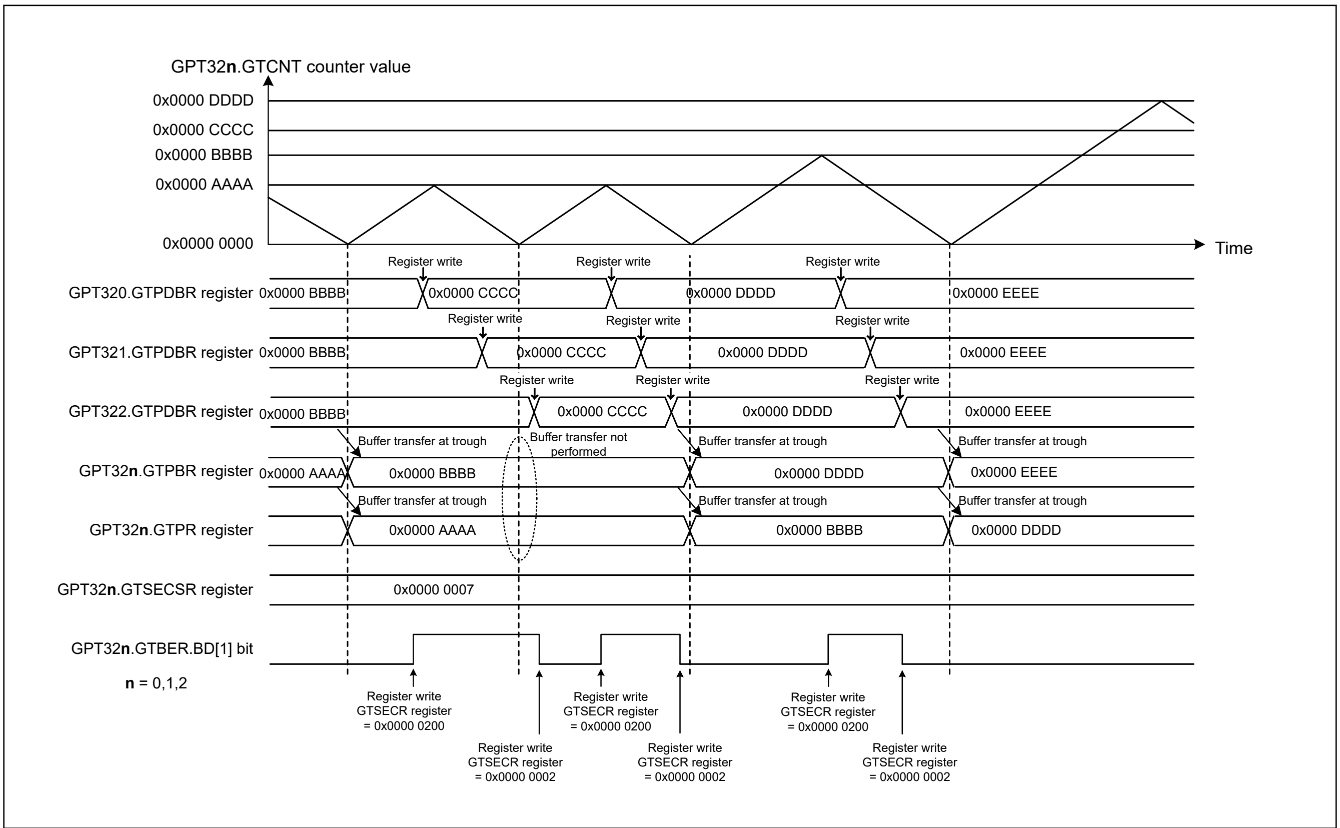


Figure 22.154 Example of Multiple Channel Operation for Disabling Buffer Operation (Triangle Waves, Double Buffer Operation)

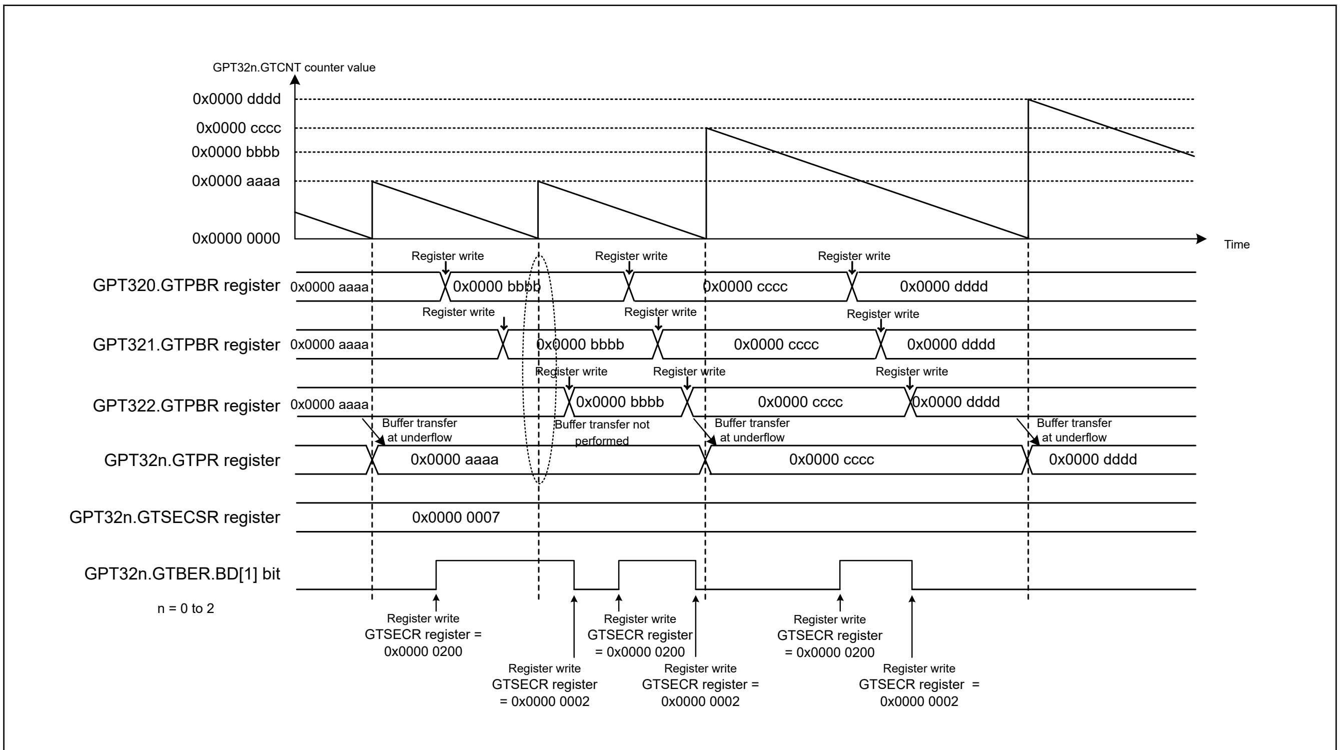


Figure 22.155 Example of Multiple Channel Operation for Disabling Buffer Operation (Saw Waves, Single Buffer Operation)

22.8.2.2 Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer

When a GTBER.DBRTECm (m = A, B) bit is set to 1 in saw-wave one-shot pulse mode or triangle-wave PWM mode 3, transfer from the intermediate buffer to the GTCCRm (m = A, B) register is repeated on a cyclic basis even while buffer transfer is disabled by the setting of the GTBER.BD[0] bit or by the buffer transfer extended skipping function.

(1) In saw-wave one-shot pulse mode

In saw-wave one-shot pulse mode, the compare match value for the first half of a waveform is stored in temporary register x (x = C, E) as the intermediate buffer for the GTCCRx (x = C, E) register and the compare match value for the second half of a waveform is stored in temporary register m (m = A, B) as the intermediate buffer for the GTCCRy (y = D, F) register, respectively, for compare match values during repeated buffer operation, and the given values are alternately transferred to the GTCCRm (m = A, B) registers.

Table 22.75 lists the types of buffer transfer of the GTCCR register during counting in saw-wave one-shot pulse mode.

While counting is stopped, the setting of the value in the temporary register is transferred through forcible buffer transfer. In forcible buffer transfer, the values of the GTCCRy (y = D, F) registers are transferred to temporary registers m (m = A, B), and the values of the GTCCRx (x = C, E) are transferred to temporary registers x (x = C, E) when the setting of the corresponding GTBER.DBRTECm (m = A, B) bit is 1, respectively.

When the setting of the GTBER.DBRTECm (m = A, B) bit is 1, values written by the CPU to the GTCCRm (m = A, B) registers are reflected as the values of temporary registers x (x = C, E).

Table 22.75 GTCCR Buffer Transfer Operation in Saw-Wave One Shot Pulse Mode during GTCNT Counting

GTBER.DBRTECm	Buffer Transfer	Timing of transfer				
		GTCCRx ↓ GTCCRm	GTCCRx ↓ Temporary register x	Temporary register x ↓ GTCCRm	GTCCRy ↓ Temporary register m	Temporary register m ↓ GTCCRm
0	Transfer enabled period	Overflow or Underflow	No transfer	No transfer	Overflow or Underflow	GTCCRm compare match
	Transfer disabled period	No transfer	No transfer	No transfer	No transfer	No transfer
1	Transfer enabled period	Overflow or Underflow	Overflow or Underflow	No transfer	Overflow or Underflow	GTCCRm compare match
	Transfer disabled period	No transfer	No transfer	Overflow or Underflow	No transfer	GTCCRm compare match

Note: m = A, B
x = C, E
y = D, F

Figure 22.156 shows the operation of generating transfer-disabled period by extended buffer transfer skipping as an example of repeated double buffer operations when the GTCCR buffer transfer disabled in saw-wave one-shot pulse mode.

Figure 22.157 shows the operation of generating transfer-disabled period by updating the GTBER.BD[0] bit as an example of repeated double-buffer operations when the GTCCR buffer transfer is disabled in saw-wave one-shot pulse mode.

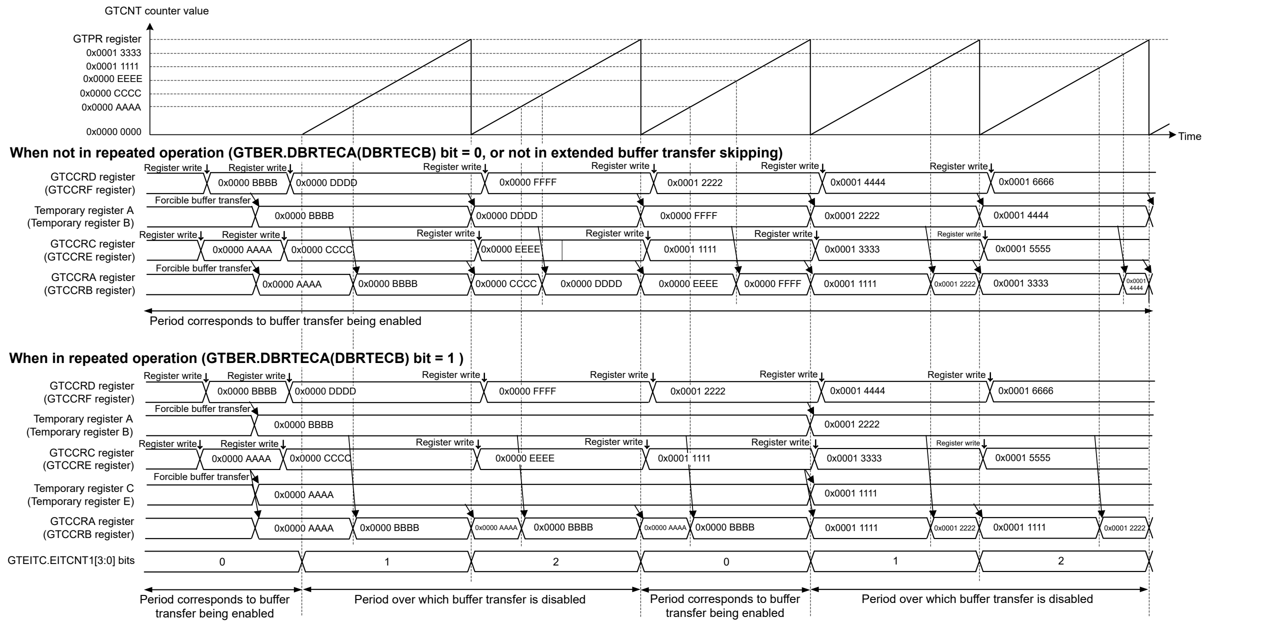


Figure 22.156 Example of Repeated Double-Buffer Operation When GTCCR Buffer Transfer is Disabled (Saw-Wave One-Shot Pulse Mode, Using Extended Buffer Transfer Skipping, GTBER.BD[0] is Constantly 0)

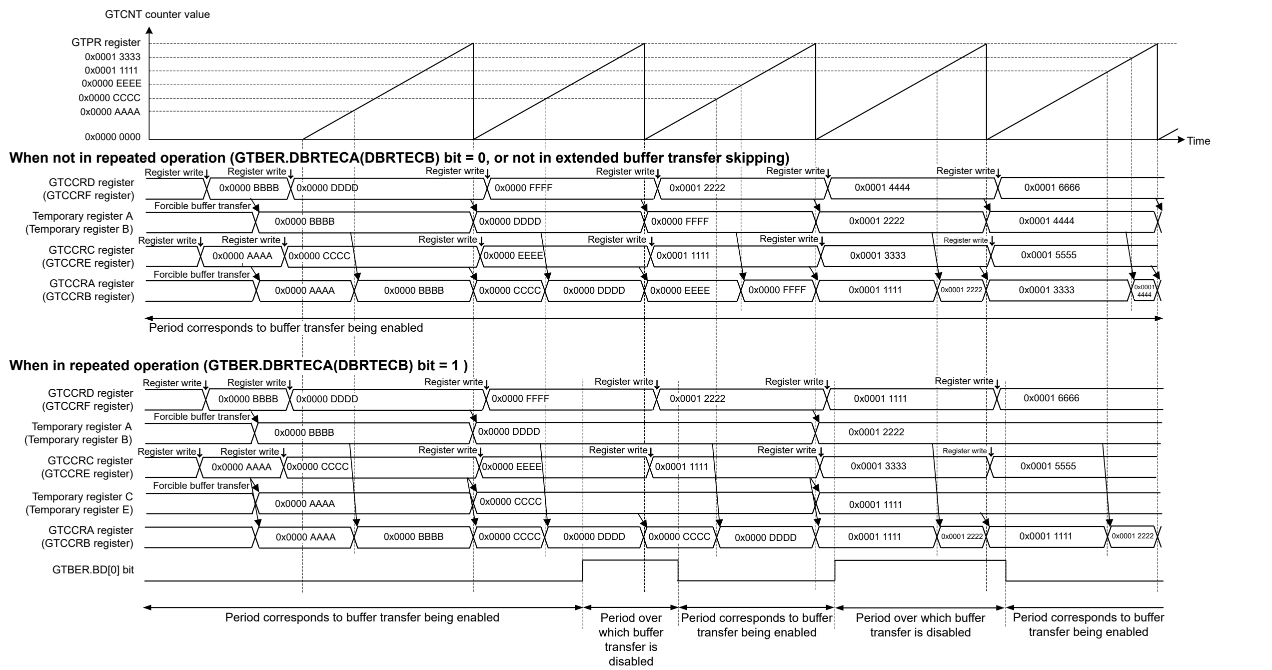


Figure 22.157 Example of Repeated Double-Buffer Operation When GTCCR Buffer Transfer is Disabled (Saw-Wave One-Shot Pulse Mode, Updating the GTBER.BD[0] Bit)

(2) In triangle-wave PWM mode 3

In triangle-wave PWM mode 3, the compare match value for the first half of a waveform is stored in the temporary register x (x = C, E) as the intermediate buffer for the GTCCRx (x = C, E) register and the compare match value for the second half of a waveform is stored in temporary register m (m = A, B) as the intermediate buffer for the GTCCRy (y = D, F) register, respectively, for compare match values during repeated buffer operation, and the given values are alternately transferred to the GTCCRm (m = A, B) register.

Table 22.76 lists the types of buffer transfer of the GTCCR register during counting operation in triangle-wave PWM mode 3.

While counting is stopped, the setting of the value in the temporary register is transferred through forcible buffer transfer. In forcible buffer transfer, the values of the GTCCRy (y = D, F) registers are transferred to temporary registers m (m = A, B), and the values of the GTCCRx (x = C, E) are transferred to temporary registers x (x = C, E), when the setting of the corresponding GTBER.DBRTECm (m = A, B) bit is 1, respectively.

When the setting of the GTBER.DBRTECm (m = A, B) bit is set to 1, values written by the CPU to the GTCCRm (m = A, B) registers are reflected as the values of temporary registers x (x = C, E).

Table 22.76 GTCCR Buffer Transfer Operation in Triangle-Wave PWM Mode 3 during GTCNT Counting

GTBER.DBRTECm	Buffer Transfer	Timing of transfer				
		GTCCRx ↓ GTCCRm	GTCCRx ↓ Temporary register x	Temporary register x ↓ GTCCRm	GTCCRy ↓ Temporary register m	Temporary register m ↓ GTCCRm
0	Transfer enabled period	Trough	No transfer	No transfer	Trough	Crest
	Transfer disabled period	No transfer	No transfer	No transfer	No transfer	No transfer
1	Transfer enabled period	Trough	Trough	No transfer	Trough	Crest
	Transfer disabled period	No transfer	No transfer	Trough	No transfer	Crest

Note: m = A, B
x = C, E
y = D, F

Figure 22.158 shows the operation of generating transfer-disabled period by extended buffer transfer skipping as an example of repeated double buffer operations when the GTCCR buffer transfer disabled in triangle-wave PWM mode 3.

Figure 22.159 shows the operation of generating transfer-disabled period by updating the GTBER.BD[0] bit as an example of repeated double-buffer operations when the GTCCR buffer transfer is disabled in triangle-wave PWM mode 3.

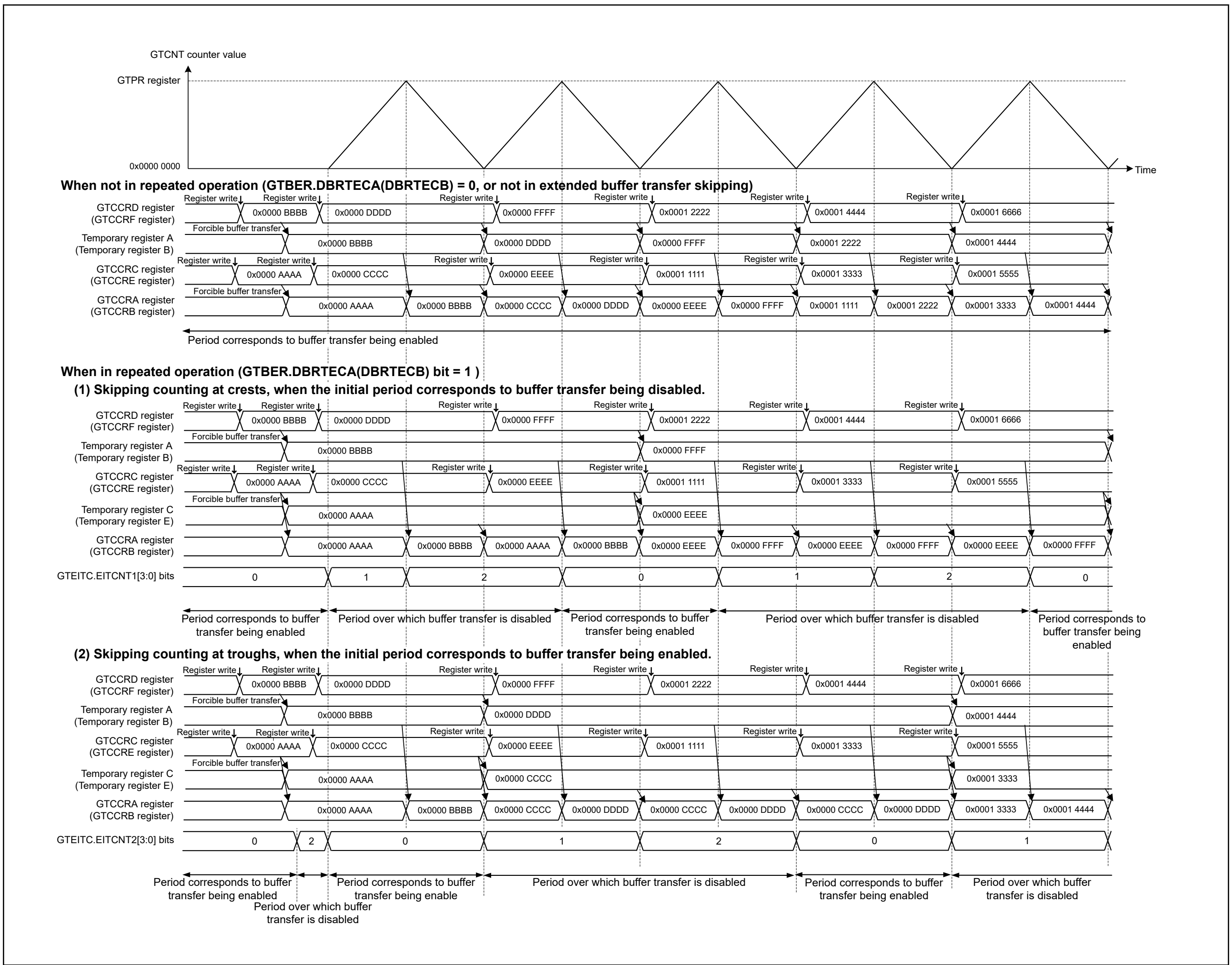


Figure 22.158 Example of Repeated Double-Buffer Operation When GTCCR Buffer Transfer is Disabled (Triangle-Wave PWM Mode 3, Using Extended Buffer Transfer Skipping, GTBER.BD[0] is Constantly 0)

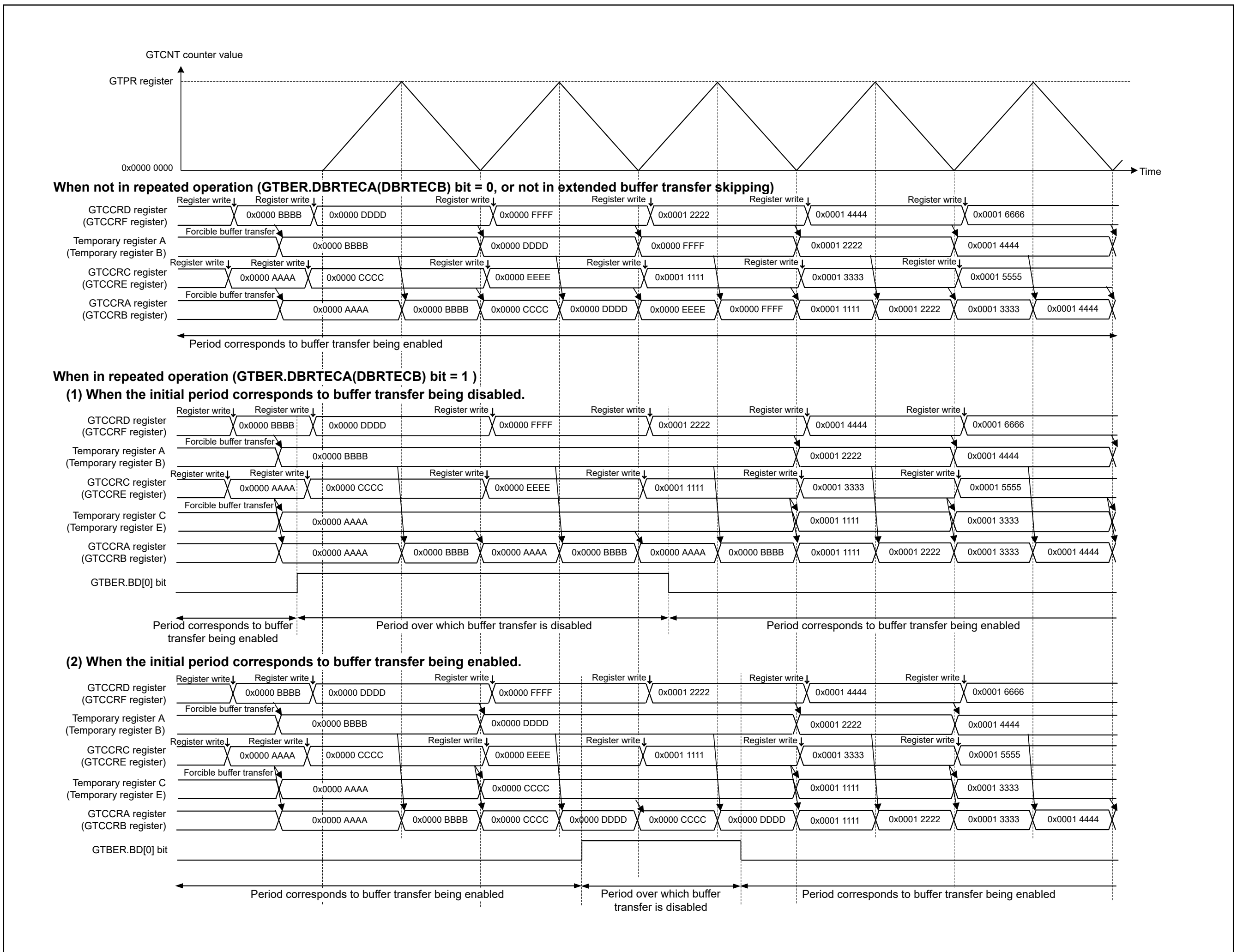


Figure 22.159 Example of Repeated Double-Buffer Operation When GTCCR Buffer Transfer is Disabled (Triangle-Wave PWM Mode 3, Updating the GTBER.BD[0] Bit)

22.8.3 GTIOCnm Pin Output Negate Control (n = 0 to 13, m = A, B)

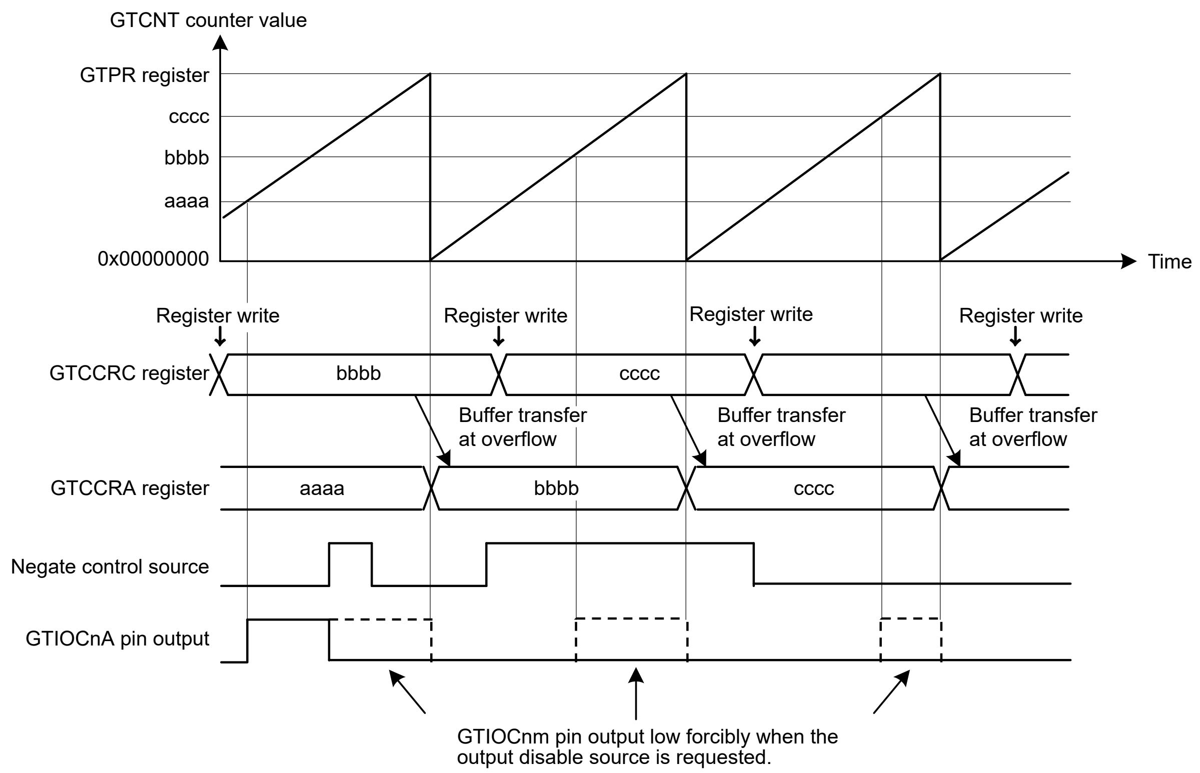
For protection from system failure, the output disable control that changes the GTIOCnm pin output value forcibly is provided for GTIOCnm pin output by the request of output disable from POEG. Output protection is required when a dead time error or the same output level being on the GTIOCnA and GTIOCnB pins is detected. GPT detects this condition and generates output disable requests to POEG according to the setting of the output disable request permission bits, such as GTINTAD.GRPDTE, GTINTAD.GRPABH, GTINTAD.GRPABL. After the POEG performs the logical OR of the output disable request from each channel and the output disable request from the external input, the POEG generates output disable requests to GPT.

One output disable signal (representing the shared output disable request signal of the GTIOCnA pin and the GTIOCnB pin) out of 4 output disable requests generated by the POEG is selected by setting GTINTAD.GRP[1:0]. The status of the selected disable output request is monitored by reading the GTST.ODF bit. The output level during output disable is set based on the GTIOR.OADF[1:0] bits for the GTIOCnA pin and the GTIOR.OBDF[1:0] setting for the GTIOCnB pin.

The change to the output disable state is performed asynchronously by generating the output disable request from the POEG. The release of the output disable state is performed at end of cycle by terminating the output disable request. It is after 3 GTCLK at shortest when the output disable condition is released after the output disable request becomes no longer satisfied. To reliably control output disabling, clear the flag of POEG for which the condition for the request to disable the output is no longer satisfied after 4 cycles of GTCLK.

When event count is performed or when the operating mode is saw-wave PWM mode 2 or when the output disable state should be released immediately without waiting for end of cycle, GTIOR.OADF[1:0] should be set to 00b (for GTIOCnA pin) or GTIOR.OBDF[1:0] should be set to 00b (for the GTIOCnB pin).

Figure 22.160 shows an example of the GTIOCnm pin output disable control operation. (n = 0 to 13, m = A, B)



Note: The base address of registers are the same.

Figure 22.160 Example of GTIOCnm pin output disable control operation in saw-wave up-counting, buffer operation, active level 1, high output at GTCCRA compare match, low output at cycle end, and low output at output disable (n = 0 to 13, m = A, B)

22.8.4 Output Protection Function for GTIOCnm Pin Output (n = 0 to 13; m = A, B)

To prepare for a case when an incorrect value (0x0000 0000 or a value greater than or equal to the GTPR register value) is set in the GTCCRA register, the output protection function for the GTIOCnm pin output (disabling function) is activated when the automatic dead time is set (GTDTCR.TDE bit = 1) in triangle-wave PWM mode.

The status of the output protection function can be read from the GTSOS.SOS[1:0] bits.

Figure 22.161 shows the output protection function state transition.

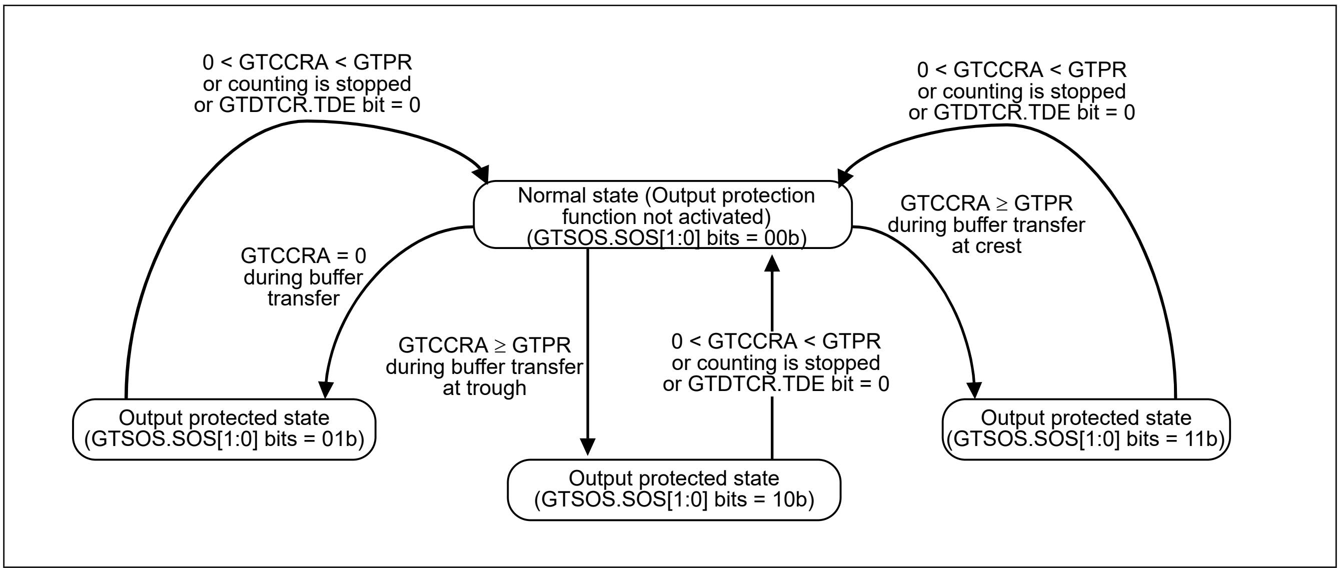


Figure 22.161 Output Protection Function

(1) Output Protection Function When the GTCCRA Register is Set to 0x0000 0000 during Buffer Transfer

Figure 22.162 and Figure 22.163 show examples of output protection function operation when the GTCCRA register is set to 0x0000 0000 during buffer transfer at troughs, and Figure 22.164 and Figure 22.165 show examples when the GTCCRA register is set to 0x0000 0000 during buffer transfer at crests.

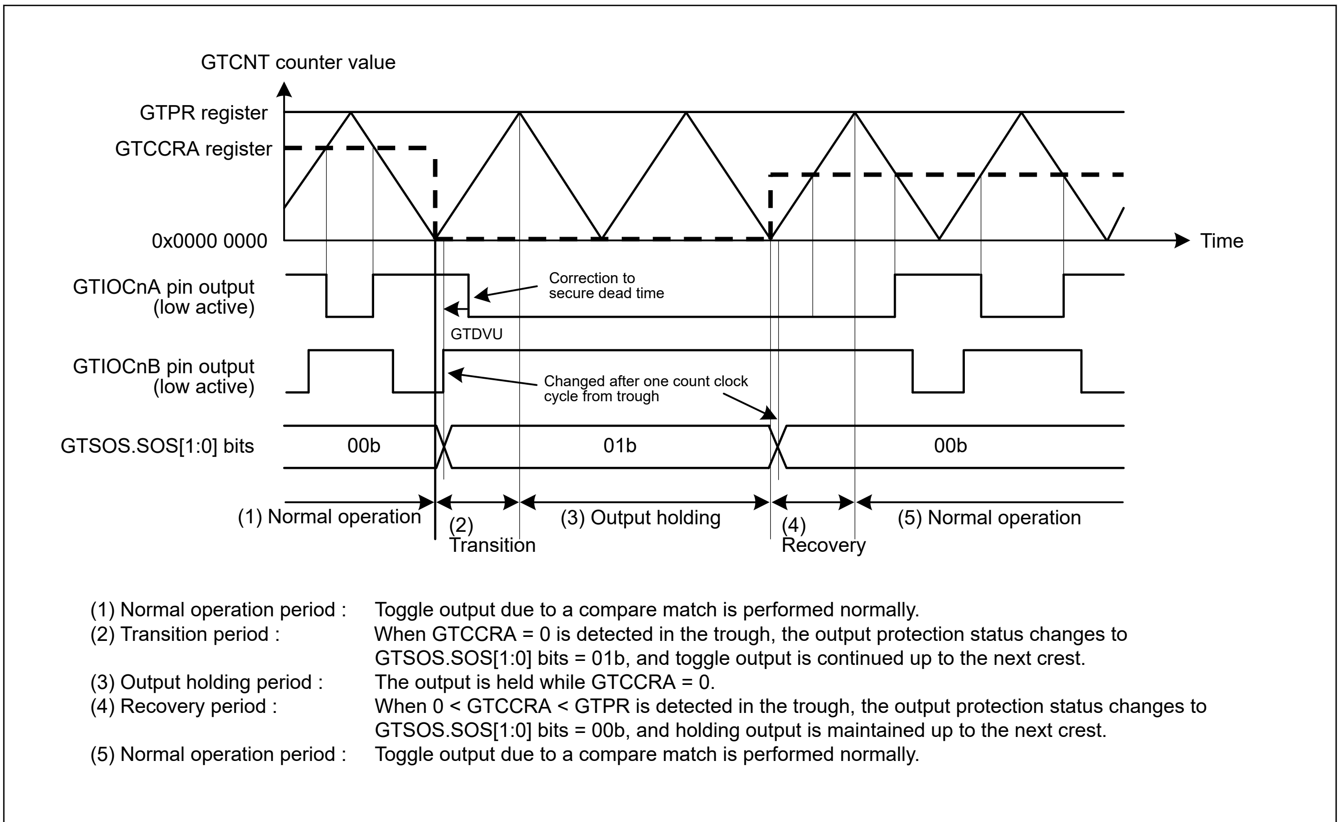
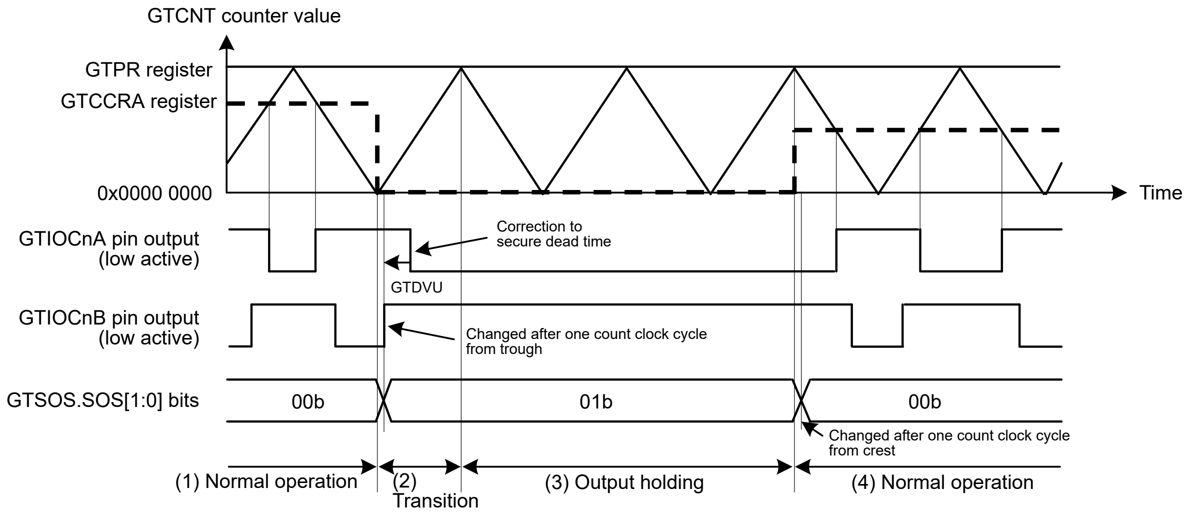
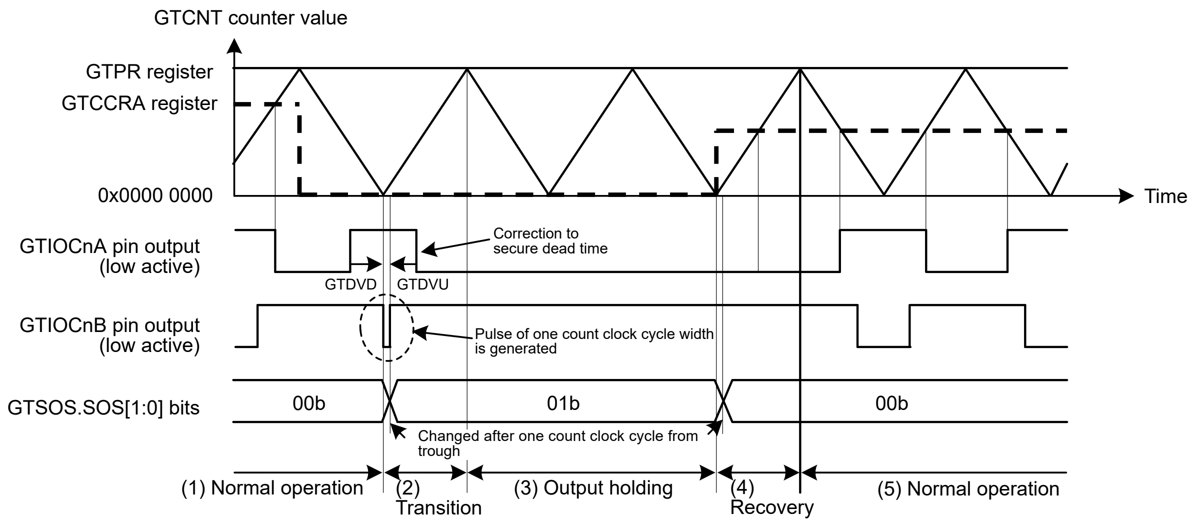


Figure 22.162 Example of Output Protection Function Operation When GTCCRA is Set to 0x0000 0000 during Buffer Transfer at Troughs (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Troughs, Active Level: Low) (n = 0 to 13)



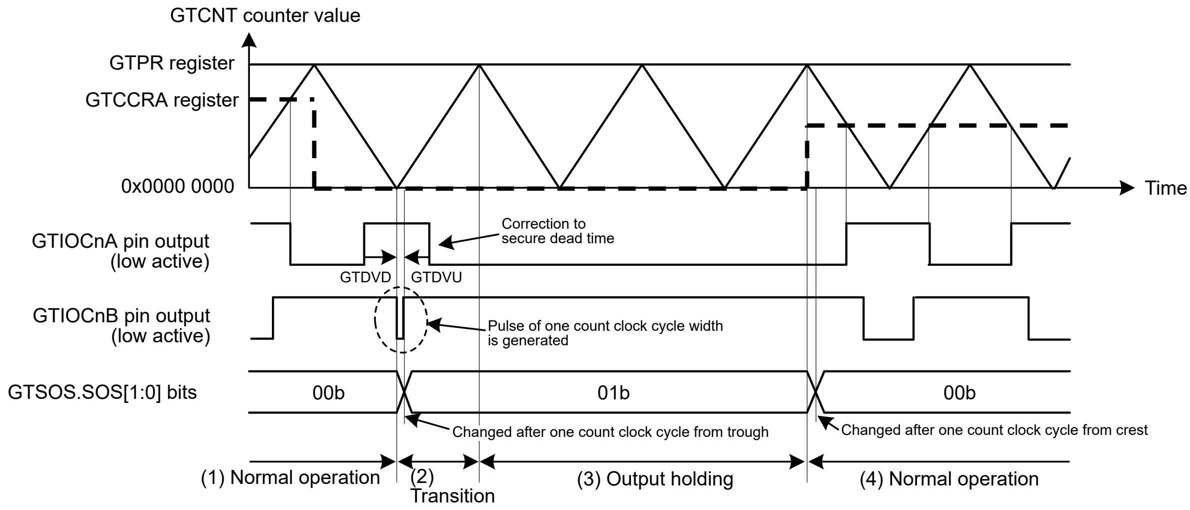
(1) Normal operation period : Toggle output due to a compare match is performed normally.
(2) Transition period : When GTCCRA = 0 is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 01b, and toggle output is continued up to the next crest.
(3) Output holding period : The output is held while GTCCRA = 0.
(4) Normal operation period : When 0 < GTCCRA < GTPR is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 22.163 Example of Output Protection Function Operation When GTCCRA is Set to 0x0000 0000 during Buffer Transfer at Troughs (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Crests, Active Level: Low) (n = 0 to 13)



(1) Normal operation period : Toggle output due to a compare match is performed normally.
(2) Transition period : When GTCCRA = 0 is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 01b, and toggle output is continued up to the next crest.
(3) Output holding : The output is held while GTCCRA = 0.
(4) Recovery period : When 0 < GTCCRA < GTPR is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and holding output is maintained up to the next crest.
(5) Normal operation period : Toggle output due to a compare match is performed normally.

Figure 22.164 Example of Output Protection Function Operation When GTCCRA is Set to 0x0000 0000 during Buffer Transfer at Crests (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Troughs, Active Level: Low) (n = 0 to 13)

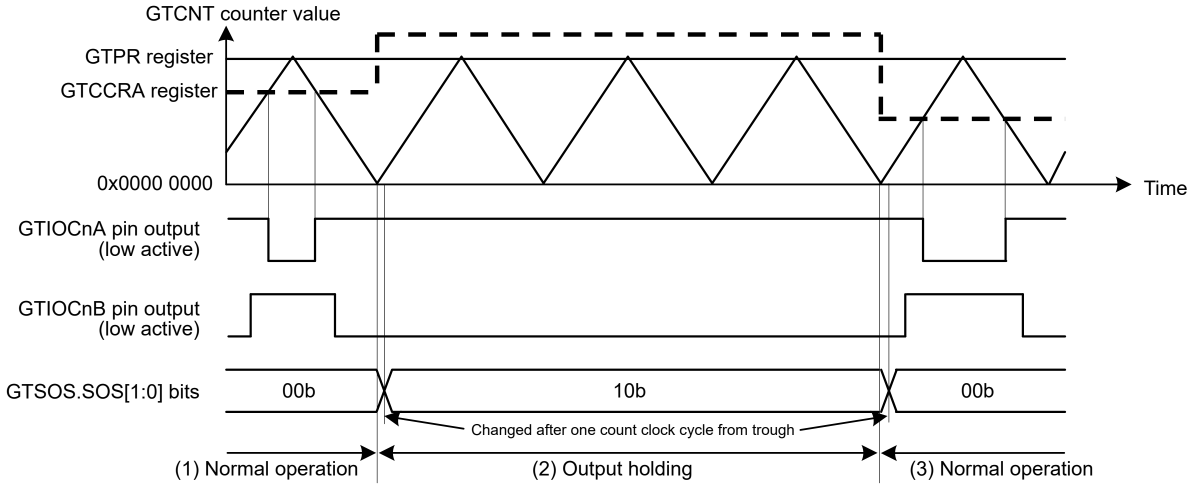


(1) Normal operation period :	Toggle output due to a compare match is performed normally.
(2) Transition period :	When GTCCRA = 0 is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 01b, and toggle output is continued up to the next crest.
(3) Output holding period :	The output is held while GTCCRA = 0.
(4) Normal operation period :	When 0 < GTCCRA < GTPR is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 22.165 Example of Output Protection Function Operation When the GTCCRA is Set to 0x0000 0000 during Buffer Transfer at Crests (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Crests, Active Level: Low) (n = 0 to 13)

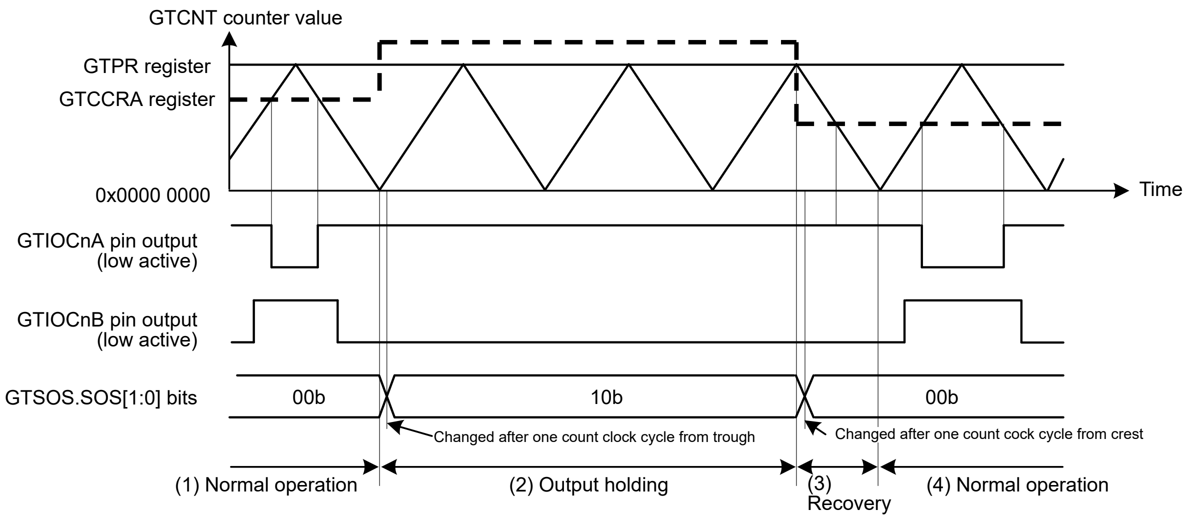
(2) Output Protection Function When GTCCRA Register ≥ GTPR Register is Set during Buffer Transfer at Troughs

Figure 22.166 and Figure 22.167 show examples of output protection function operation when GTCCRA register ≥ GTPR register is set during buffer transfer at troughs.



(1) Normal operation period :	Toggle output due to a compare match is performed normally.
(2) Output holding period :	When GTCCRA ≥ GTPR is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 10b, and the output is held while GTCCRA ≥ GTPR.
(3) Normal operation period :	When 0 < GTCCRA < GTPR is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 22.166 Example of Output Protection Function Operation When GTCCRA ≥ GTPR is set during Buffer Transfer at Troughs (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Troughs, Active Level: Low) (n = 0 to 13)

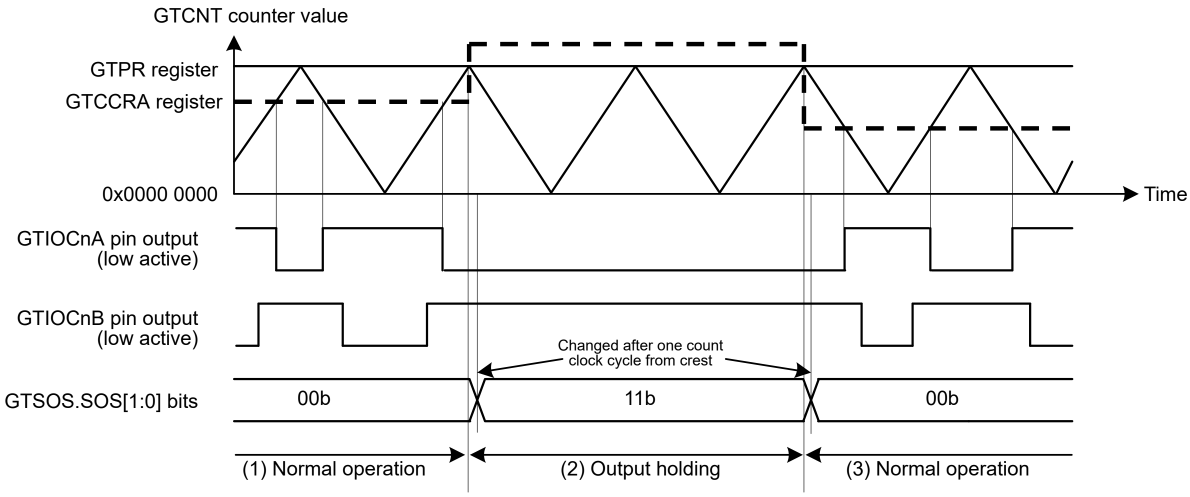


(1) Normal operation period :	Toggle output due to a compare match is performed normally.
(2) Output holding period :	When GTCCRA ≥ GTPR is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 10b, and the output is held while GTCCRA ≥ GTPR.
(3) Recovery period :	When 0 < GTCCRA < GTPR is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and holding output is maintained up to the next trough.
(4) Normal operation period :	Toggle output due to a compare match is performed normally.

Figure 22.167 Example of Output Protection Function Operation When GTCCRA ≥ GTPR is Set during Buffer Transfer at Troughs (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Crests, Active Level: Low) (n = 0 to 13)

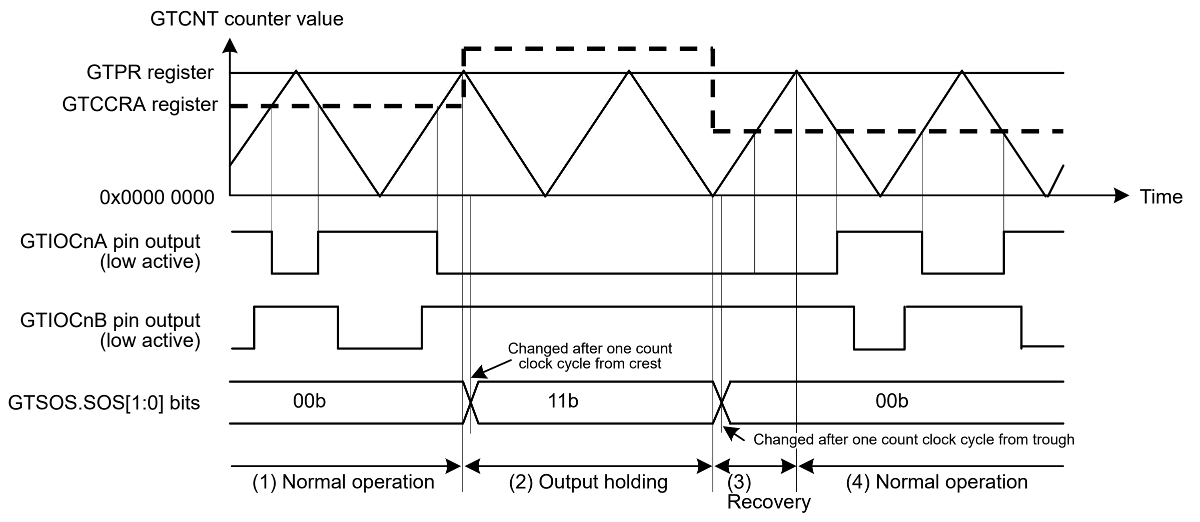
(3) Output Protection Function When GTCCRA ≥ GTPR is Set during Buffer Transfer at Crests

Figure 22.168 and Figure 22.169 show examples of output protection function operation when GTCCRA ≥ GTPR is set during buffer transfer at crests.



(1) Normal operation period : Toggle output due to a compare match is performed normally.
(2) Output holding period : When GTCCRA ≥ GTPR is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] bits = 11b, and the output is held while GTCCRA ≥ GTPR.
(3) Normal operation period : When 0 < GTCCRA < GTPR is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 22.168 Example of Output Protection Function Operation When GTCCRA ≥ GTPR is Set during Buffer Transfer at Crests (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Crests, Active Level: Low) (n = 0 to 13)



(1) Normal operation period :	Toggle output due to a compare match is performed normally.
(2) Output holding period :	When GTCCRA ≥ GTPR is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] = 11b, and the output is held while GTCCRA ≥ GTPR.
(3) Recovery period :	When 0 < GTCCRA < GTPR is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] = 00b, and holding output is maintained up to the next crest.
(4) Normal operation period :	Toggle output due to a compare match is performed normally.

Figure 22.169 Example of Output Protection Function Operation When GTCCRA ≥ GTPR is Set during Buffer Transfer at Crests (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Troughs, Active Level: Low) (n = 0 to 13)

(4) Restricted Specification of Output Protection Function

Even if an incorrect value (0x0000 0000 or a value greater than or equal to the GTPR register value) is set in the GTCCRA register during count operation, the output protection functions in a specific way such that one of the positive- and negative-phase outputs becomes non-active. However, if the following condition is not satisfied, the output protection does not operate normally.

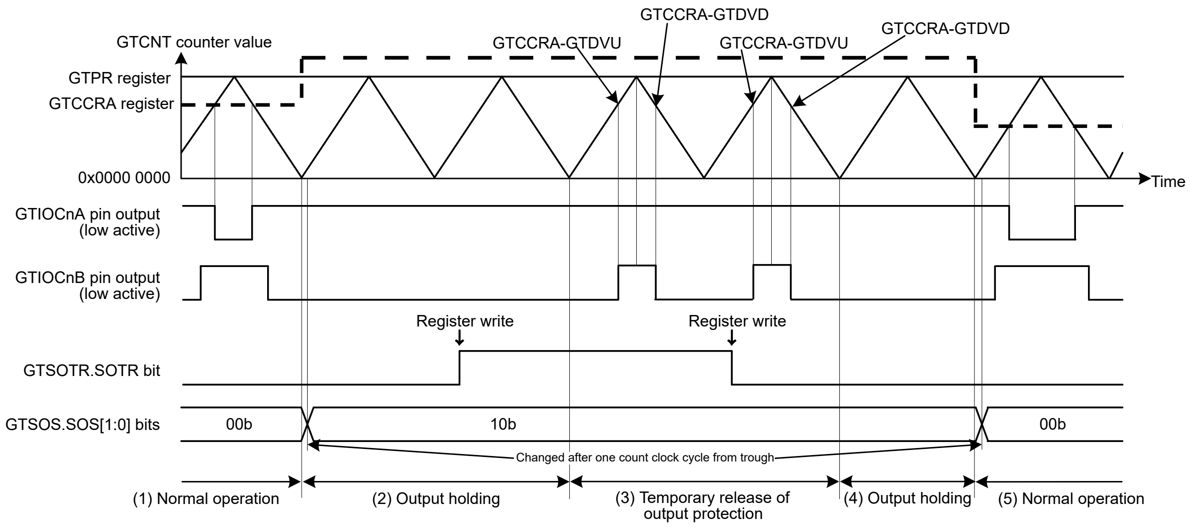
- When the GTCCRA register value at the start of count operation is greater than 0x0000 0000, and less than the setting value of the GTPR register

(5) Temporary Release of Output Protection Function

When the GTSOS.SOS[1:0] bits = 10b (protected state in which GTCCRA register ≥ GTPR register has occurred during transfer at trough), the protected state of the GTIOCnB pin output can be temporarily released by setting the GTSOTR.SOTR bit to 1. The SOS[1:0] bits retain 10b even if the output protection function is released.

When the SOTR bit is set to 0, the GTIOCnB pin output protection can be restarted.

Figure 22.170 shows examples of the operation of temporary release of output protection when the setting of the GTCCRA register ≥ GTPR register during buffer transfer at troughs.



(1) Normal operation period:	Toggle output due to a compare match is performed normally.
(2) Output holding period:	When GTCCRA ≥ GTPR is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 10b, and the output is held while GTCCRA ≥ GTPR. Even when the GTSOTR.SOTR bit is set to 1 during the output holding period, the GTIOCnB pin output is held until the first trough after the setting.
(3) Output protection temporary release period:	After GTSOTR.SOTR bit is set to 1, the GTIOCnB pin output is performed normally from the first trough after the setting. Even when the GTSOTR.SOTR bit is set to 0 during the output protection releasing period, the GTIOCnB pin output is held until the first trough after the setting.
(4) Output holding period:	The output is held until 0 < GTCCRA < GTPR is detected in the trough.
(5) Normal operation period:	When 0 < GTCCRA < GTPR is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 22.170 Example of Temporary Release of Output Protection When the Setting of the GTCCRA ≥ GTPR during Buffer Transfer at Trough s (Restored to 0 < GTCCRA < GTPR during Buffer Transfer at Troughs, Active Level: Low) (n = 0 to 13)

22.9 Initialization Method of Output Pins

22.9.1 Pin Settings After Reset

The GPT registers are initialized at a reset. Start counting after selecting the port pin function with the PmnPFS register, setting GTIOR.OAE and GTIOR.OBE bits, and outputting the GPT function to external pins.

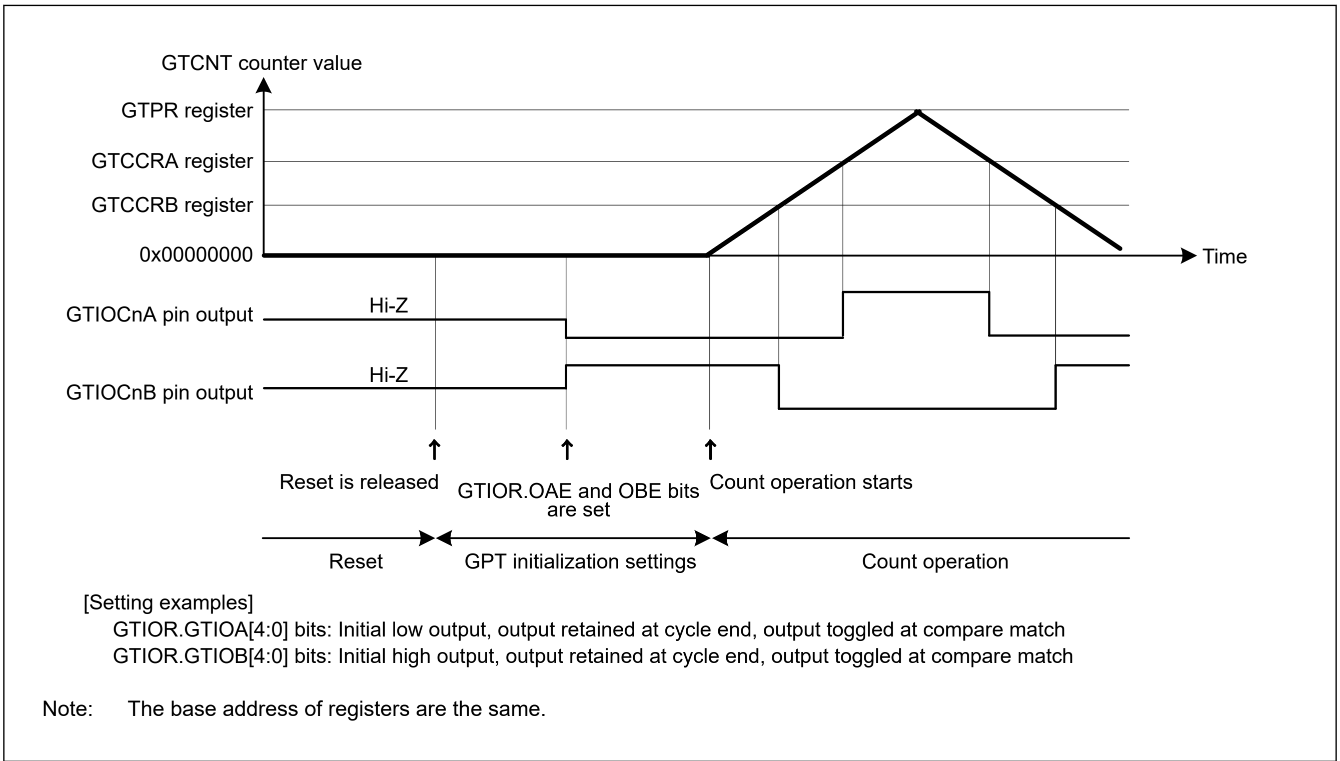


Figure 22.171 Example of pin settings after reset

22.9.2 Pin Initialization Due to Error During Operation

If an error occurs during GPT operation, the following four types of pin control can be performed before pin initialization:

- Set the OAHLD and OBHLD bits in GTIOR to 1 and retain the outputs at count stop
- Set the OAHLD and OBHLD bits in GTIOR to 0, specify arbitrary output values at OADFLT and OBDFLT in GTIOR, and output the arbitrary values at count stop
- Set the pin to output an arbitrary value as a general output port by setting the PDR, PODR registers and PmnPFS.PMR bit of the I/O port in advance. Set the OAE and OBE bits in GTIOR to 0, and the control bit associated with the pin in the PMR to 0 to allow arbitrary values to be output from the pin set as a general output port when an error occurs.
- Drive the output to a high impedance state using the POEG function.

If the automatic dead time setting is made, clear the GTDTCR.TDE bit to 0 after counting stops. When counting stops, only the values of registers that are changed by a GPT external source change. If counting is resumed, operation continues from where it stopped. If counting is stopped, the registers must be initialized before counting starts.

22.10 Usage Notes

22.10.1 Module-Stop Function Setting

The Module Stop Control Register can enable or disable GPT operation. The GPT is initially stopped after a reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Mode.

Set GTCLKCR register before releasing the module-stop state.

22.10.2 GTCCRn Settings During Compare Match Operation (n = A to F)

(1) When automatic dead time setting is made in triangle-wave PWM mode

The GTCCRA register must satisfy all of the following conditions:

- GTDVU < GTCCRA
- GTCCRA > GTDVD

- 0 < GTCCRA < GTPR

When the setting of GTCCRA = 0 or GTCCRA ≥ GTPR is made for the GTCCRA register during count operation, the output protection function is activated. However, if the following condition is not satisfied, the output protection function does not work normally:

- The value of the GTCCRA register at the start of counting is larger than 0 and less than GTPR.

For details, see section 22.8.4. Output Protection Function for GTIOCnm Pin Output (n = 0 to 13; m = A, B)

(2) When automatic dead time setting is not made in triangle-wave PWM mode

The GTCCRA register must be set within the range of 0 < GTCCRA < GTPR. If GTCCRA = 0 or GTCCRA = GTPR is set, a compare match occurs within the cycle only when GTCCRA = 0 or GTCCRA = GTPR is satisfied. When GTCCRA > GTPR, no compare match occurs.

Similarly, GTCCRB must be set within the range of 0 < GTCCRB < GTPR. If GTCCRB = 0 or GTCCRB = GTPR is set, a compare match occurs within the cycle only when GTCCRB is 0 or GTCCRB = GTPR is satisfied. When GTCCRB > GTPR, no compare match occurs.

(3) When automatic dead time setting is made in saw-wave one-shot pulse mode

The GTCCRC and GTCCRD registers must be set to satisfy the following restrictions. If the restrictions are not satisfied, the correct output waveforms with secured dead time may not be obtained.

- In up-counting: GTCCRC < GTCCRD, GTCCRC > GTDVU, GTCCRD < GTPR - GTDVD
- In down-counting: GTCCRC > GTCCRD, GTCCRC < GTPR - GTDVU, GTCCRD > GTDVD

(4) When automatic dead time setting is not made in saw-wave one-shot pulse mode

The GTCCRC and GTCCRD registers must be set to satisfy the following restrictions. If the restrictions are not satisfied, two compare matches do not occur and pulse output cannot be performed.

- In up-counting: 0 < GTCCRC < GTCCRD < GTPR
- In down-counting: GTPR > GTCCRC > GTCCRD > 0

Similarly, GTCCRE and GTCCRF must be set to satisfy the following restrictions. If the restrictions are not satisfied, two compare matches do not occur and pulse output cannot be performed.

- In up-counting: 0 < GTCCRE < GTCCRF < GTPR
- In down-counting: GTPR > GTCCRE > GTCCRF > 0.

(5) In saw-wave PWM mode

The GTCCRA register must be set with the range of 0 < GTCCRA < GTPR. If GTCCRA = 0 or GTCCRA = GTPR is set, a compare match occurs within the cycle only when GTCCRA = 0 or GTCCRA = GTPR is satisfied. If GTCCRA > GTPR is set, no compare match occurs.

Similarly, GTCCRB must be set with the range of 0 < GTCCRB < GTPR. If GTCCRB = 0 or GTCCRB = GTPR is set, a compare match occurs within the cycle only when GTCCRB = 0 or GTCCRB = GTPR is satisfied. If GTCCRB > GTPR is set, no compare match occurs.

(6) In Complementary PWM mode 1, 2, 3

The GTCCRn register must be set with the range of 0 ≤ GTCCRn ≤ GTPR + GTDVU.

(7) In Complementary PWM mode 4

In single buffer operation, the GTCCRn register must be set with the range of 0 ≤ GTCCRn ≤ GTPR + GTDVU.

In double buffer operation, the GTCCRn register must be set with the range of GTDVU < GTCCRn < GTPR.

22.10.3 Setting Range for GTCNT Counter

Other than the Saw-wave PWM mode 2 and Complementary PWM mode, the GTCNT counter register must be set with the range of 0 ≤ GTCNT ≤ GTPR.

22.10.4 Starting and Stopping the GTCNT Counter

The control timing of starting and stopping the GTCNT counter by the GTCR.CST bit synchronizes the count clock that is selected in GTCR.TPCS[3:0]. When GTCR.CST is updated, the GTCNT counter starts/stops after a count clock that is selected in GTCR.TPCS[3:0]. Therefore, an event generated before the GTCNT counter actually starts is ignored, resulting in situations in which an event is accepted or an interrupt occurs after GTCR.CST is set to 0.

22.10.5 Priority Order of Each Event

(1) GTCNT register

Table 22.77 shows a priority order of events updating the GTCNT register.

Table 22.77 Priority order of sources updating GTCNT

Source updating GTCNT	Priority order
Writing by CPU (writing to GTCNT/GTCLR)	High
Clear by hardware sources set in GTCSR	↑
Count up or down by hardware sources set in GTUPSR/GTDNSR	↑
Count operation	Low

If up-counting and down-counting by hardware sources occur at the same time, the GTCNT counter value does not change. When there is a conflict between updating the GTCNT register and reading by the CPU, pre-update data is read.

(2) GTCR.CST bit

When there is a conflict between starting/stopping by hardware sources set in the GTSSR/GTPSR registers and writing by the CPU (writing to GTCR/GTSTR/GTSTP registers), the writing by CPU has priority over the starting/stopping by hardware sources.

In case that stop by the period count function conflicts with start by the CPU writing (GTCR register writing/GTSTR register writing), the period count function is finished with setting the GTST.PCF flag. The CST bit is not changed and the GTCNT continues to count.

When there is a conflict between starting by hardware sources set in the GTSSR register and stopping by hardware sources set in GTPSR register, the GTCR.CST bit value does not change. When there is a conflict between updating the GTCR.CST bit and reading by the CPU (reading from GTCR/GTSTR/GTSTP registers), pre-update data is read.

(3) GTCCRm registers (m = A to F)

When there is a conflict between input capture/buffer transfer operation and writing to the GTCCRm registers, the writing to GTCCRm registers has priority over input capture/buffer transfer operation. When there is a conflict between input capture and writing to the counter register by the CPU or updating the counter register by hardware sources, the pre-update counter value is captured. When there is a conflict between updating the GTCCRm registers and reading by the CPU, pre-update data is read.

(4) GTPR register

When there is a conflict between buffer transfer operation and writing to the GTPR register, writing to GTPR register has priority over buffer transfer operation. When there is a conflict between updating GTPR register and reading by the CPU, pre-update data is read.

(5) GTADTRm registers (m = A, B)

When there is a conflict between buffer transfer operation and writing to GTADTRm register, writing to GTADTRm register has priority over buffer transfer operation.

When there is a conflict between updating the GTADTRm register and reading by the CPU, pre-update data is read.

(6) GTDVm registers (m = U, D)

When there is a conflict between buffer transfer operation and writing to GTDVm register, writing to GTDVm register has priority over buffer transfer operation.

When there is a conflict between updating the GTDVm register and reading by the CPU, pre-update data is read.

(7) GTIOR.GTIOm registers (m = A, B)

When there is a conflict between buffer transfer operation and writing to GTIOR.GTIOm register, writing to GTIOR.GTIOm register has priority over buffer transfer operation.

When there is a conflict between updating the GTIOR.GTIOm and reading by the CPU, pre-update data is read.

22.10.6 Interval of Interrupt Request

When the core clock of GPT is GPTCLK, interrupt may be lost if the interval between the same interrupt signal is shorter than the following value. However, this restriction does not apply to different interrupt signals.

Interrupt_Interval [ns] = Period_of_GPTCLK [ns] * 6 + Period_of_PCLKA [ns] * 4

Also, ADC can receive the A/D conversion start request from GPT without going through ELC by setting ADTRGGPTx register (x = 0 to 8).

Again, when the clocks of GPT and ADC are combined as shown in Table 22.78, the A/D conversion start request may be lost if the interval between one A/D conversion start request and the next is less than the following value for the same A/D conversion start request.

However, this restriction does not apply to different A/D conversion start request.

Event_Interval [ns] = Period_of_GPT_core_clock [ns] * 6 + Period_of_ADC_core_clock [ns] * 4

Table 22.78 Combination of clocks with the restricted Event Interval

GPT core clock	ADC core clock
GPTCLK	PCLKA or ADCCLK
PCLKD	ADCCLK or GPTCLK

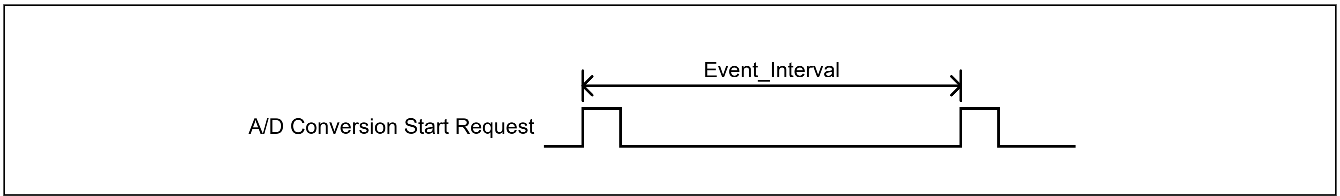


Figure 22.172 Example of A/D Conversion Start Request Event Interval

22.10.7 Notes on the GTIOCnm Signal Input to PWM Delay Generation Circuit (n = 0 to 3, m = A, B)

When controlling the delay of PWM waveform in PWM Delay Generation Circuit, the following limitations exist.

- In saw-wave mode, It is prohibited to change the GTIOCnm signal during the three clock cycles immediately before overflow or underflow.
- In saw-wave mode, it is prohibited to clear the GTCNT register by GTCSR during counting operation.
- In triangle-wave mode, It is prohibited to change the GTIOCnm signal during the three clock cycles immediately before trough.

If the above limitations are not followed, the edge of signal waveform output from PWM Delay Generation Circuit may disappear.

Figure 22.173 shows an example of the change timing of an unacceptable GTIOCnm signal.

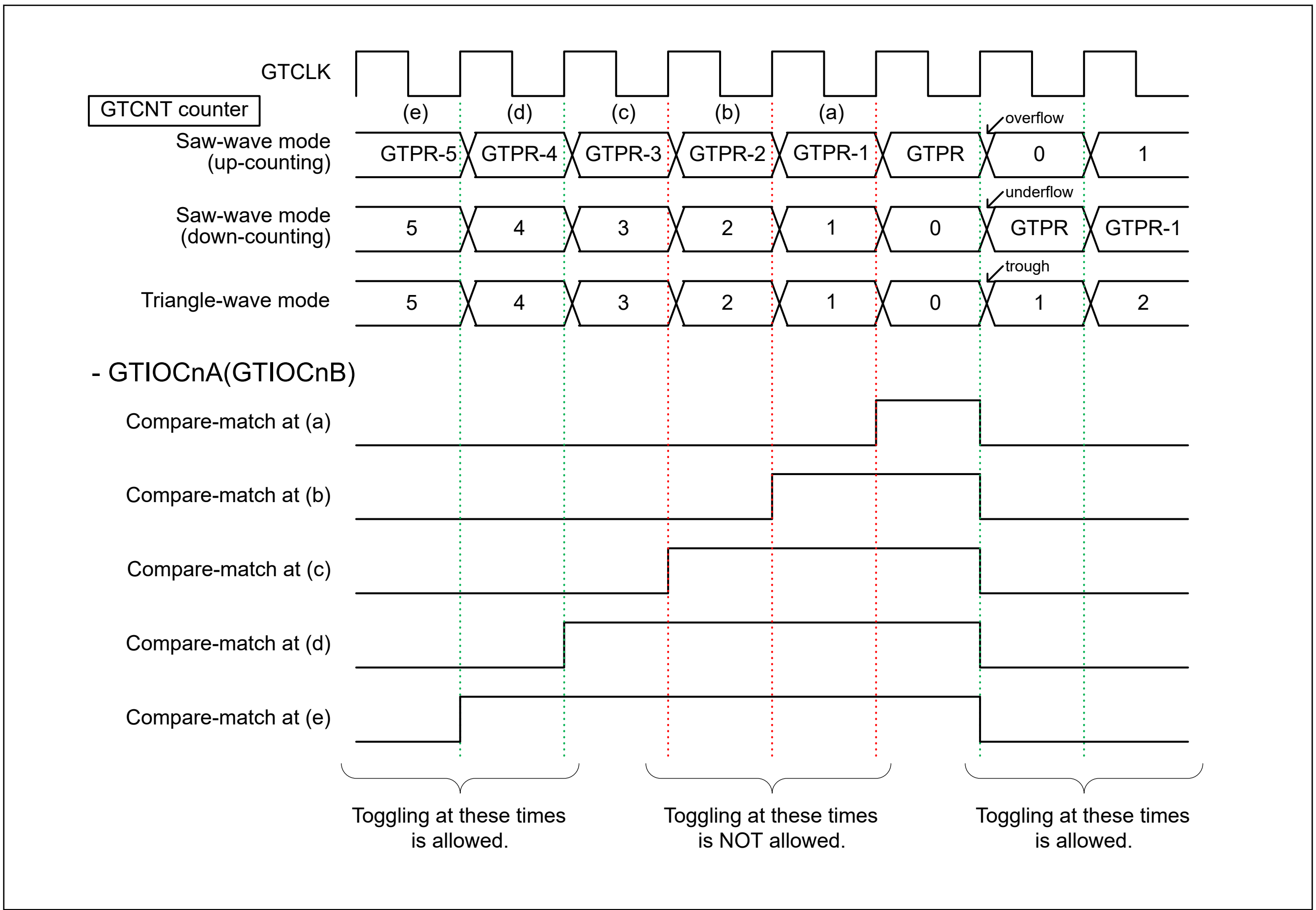


Figure 22.173 Example of unacceptable GTIOCnm signal timing (n = 0 to 3, m = A, B)

22.10.8 Setting Range of the Compare Match Registers when GTIOR.CPSCIR=1

In complementary PWM mode, when GTIOR.CPSCIR=1 and the initial output of the GTIOCnA/GTIOCnB pins after synchronization clearing is disabled, set the compare match registers (GTCCRA, GTCCRC, GTCCRD, GTCCRE, GTCCRF) to at least twice the GTDVU value.

22.10.9 Strobe Signal for Testing to PWM Delay Generation Circuit

Each channel output PWM output, PWM cycle synchronous signal (the cycle end), PWM output strobe signal for testing, register write protection signal (GTWP.WP bit) to the PWM Delay Generation circuit (PDG).

The strobe signals monitor each rising edge or falling edge of GTIOCxA output and GTIOCxB output. One GTCLK later from changing of PWM output, the strobe signal becomes 1. At the count clock of the cycle end, it becomes 0. The conditions that the strobe signal becomes 0 are reset and the cycle end.

When GTCNT counting is stopped, the status of strobe signal is retained. When GTCNT counting is restarted and PWM output is generated after GTCNT is stopped under the condition that strobe signal is 1, during first period after restarting to count, strobe signal does not change from 1. So the edge of PWM output is not monitored correctly. At the first cycle end after restarting, strobe signal returns to 0. The strobe signal is generated correctly from the second cycle.

In order to reset strobe signal keeping the status of 1 during GTCNT stopping, generate the cycle end by GTCNT counting as shown in Figure 22.174 or clear counter for sawtooth-wave operation as shown in Figure 22.175.

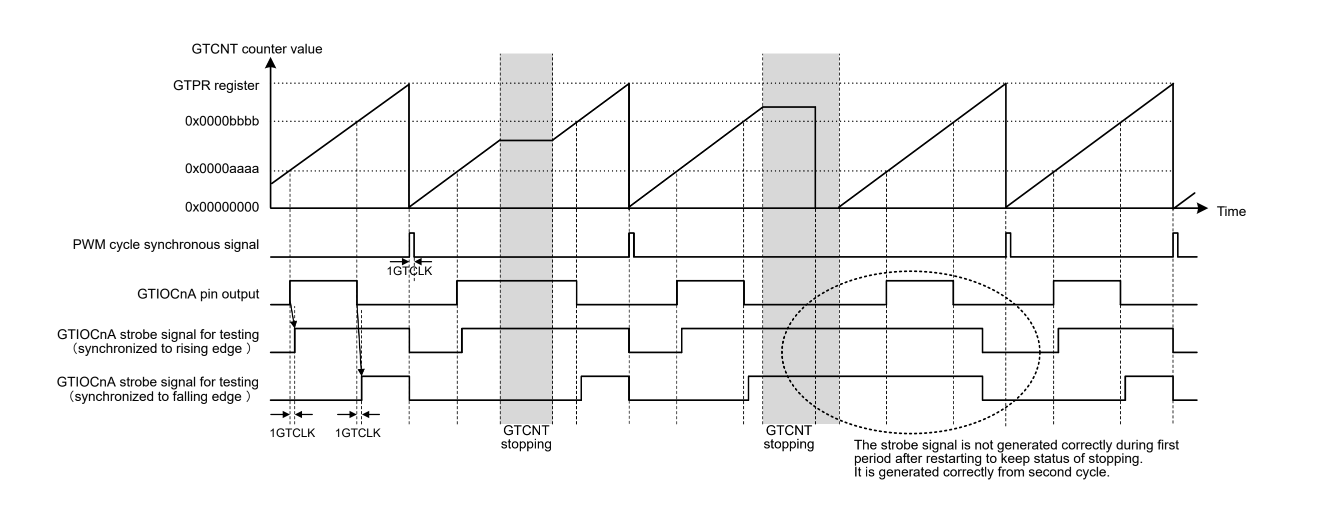


Figure 22.174 Example of output operation to PDG (1)

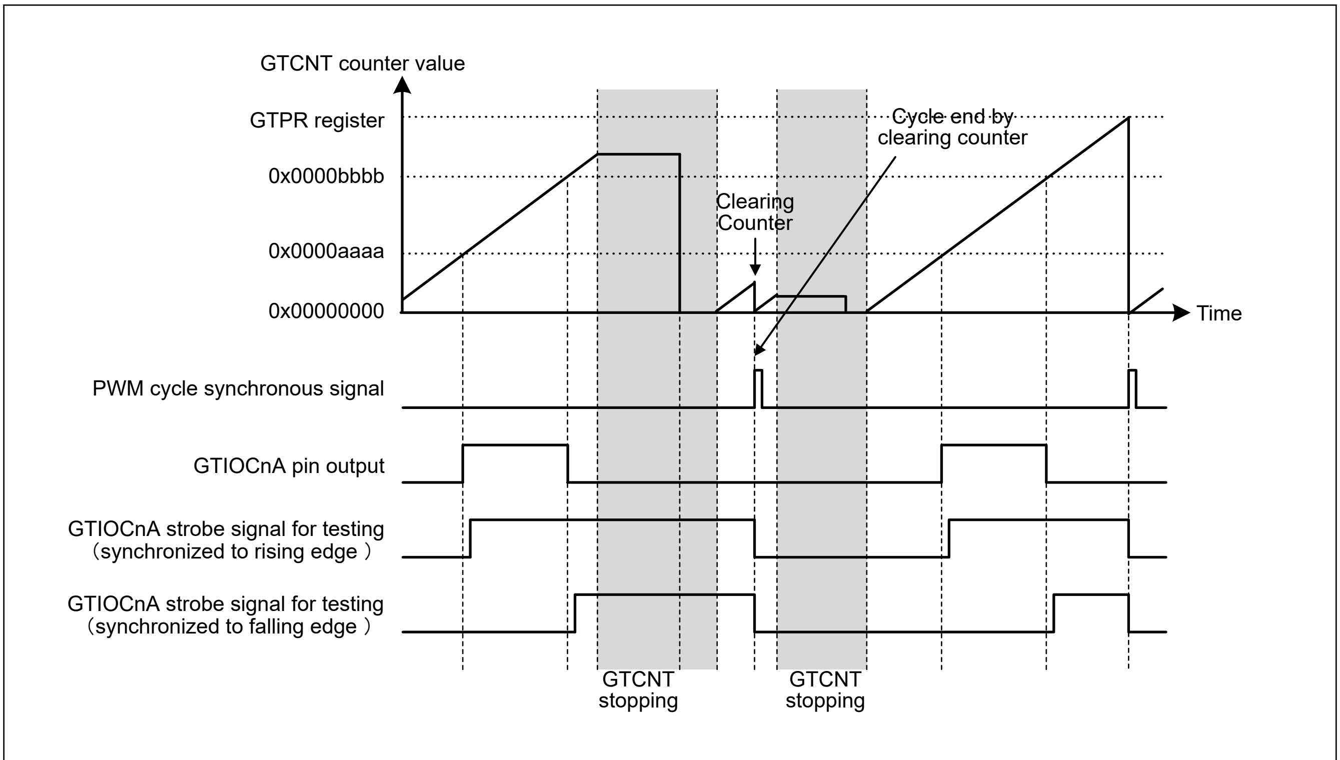


Figure 22.175 Example of output operation to PDG (2)

22.10.10 Notes on Transitioning to Standby Mode

Before transitioning to standby mode, it should be stopped counter operation (Set GTCR.CST = 0).

23. PWM Delay Generation Circuit (PDG)

23.1 Overview

The PWM Delay Generation circuit (PDG) has 4 channels delay circuits that can connect to the GPT. The PDG can control the rise and fall edge timing with which the PWM output for the GPT320 through the GPT323.

Table 23.1 lists the specifications for the PWM Delay Generation Circuit, Figure 23.1 shows a block diagram, and Table 23.2 lists the I/O pins.

Table 23.1 Specifications of the PWM Delay Generation Circuit

Parameter	Specifications
Function	The circuit can control the timing with which signals on the two PWM output pins for channel 0/1/2/3 rise and fall to an accuracy of up to 1/128 times the period of the GPT core clock (GTCLK). The GPT core clock (GTCLK) can be selected from PCLKD or GPTCLK.

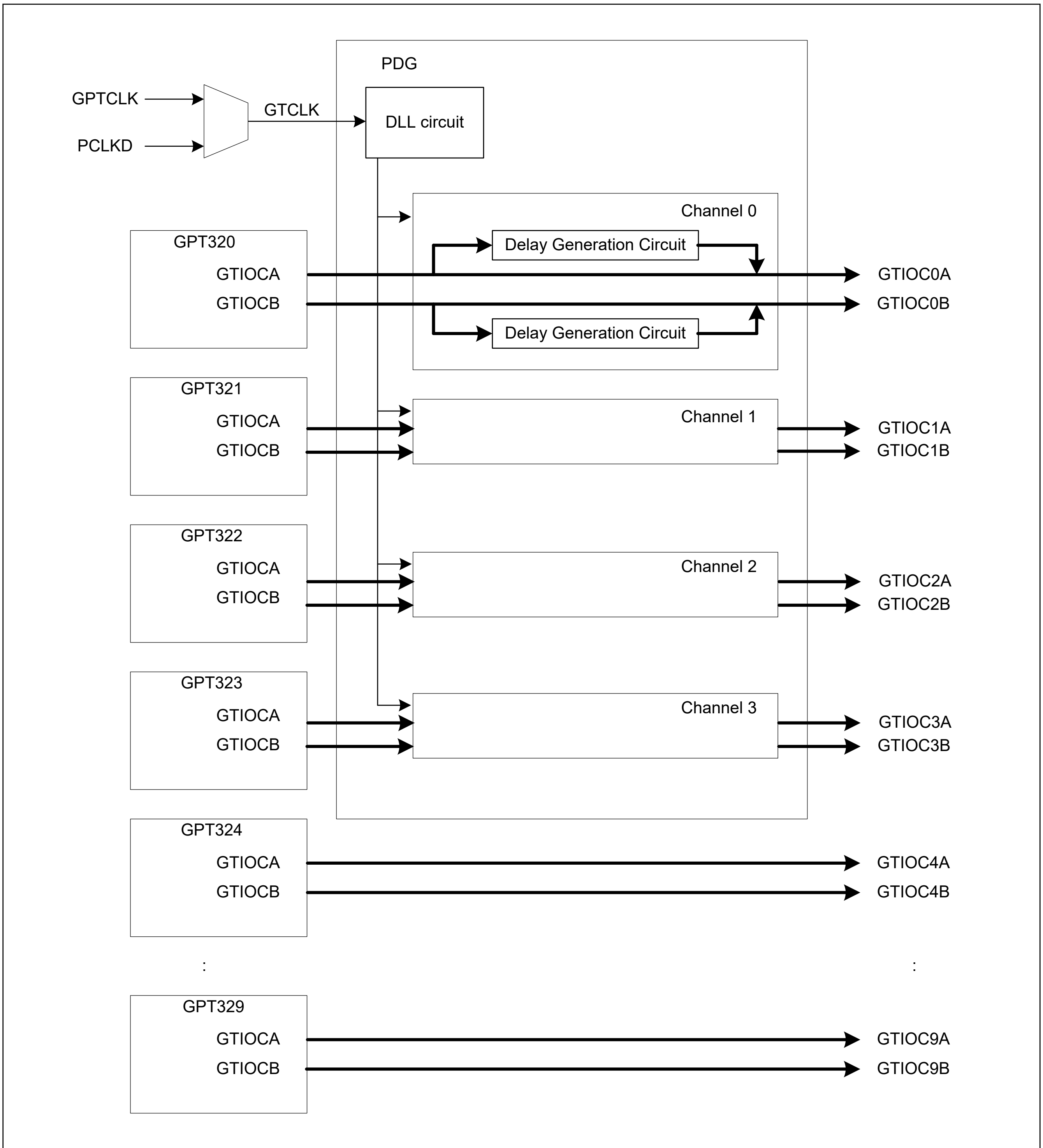


Figure 23.1 PWM delay generation circuit block diagram

Table 23.2 PWM delay generation circuit I/O pins (1 of 2)

I/O pin	I/O	Function
GTIOC0A	Output	Delayed output of GTIOCA pin of GPT channel 0
GTIOC0B	Output	Delayed output of GTIOCB pin of GPT channel 0
GTIOC1A	Output	Delayed output of GTIOCA pin of GPT channel 1
GTIOC1B	Output	Delayed output of GTIOCB pin of GPT channel 1
GTIOC2A	Output	Delayed output of GTIOCA pin of GPT channel 2

Table 23.2 PWM delay generation circuit I/O pins (2 of 2)

I/O pin	I/O	Function
GTIOC2B	Output	Delayed output of GTIOCB pin of GPT channel 2
GTIOC3A	Output	Delayed output of GTIOCA pin of GPT channel 3
GTIOC3B	Output	Delayed output of GTIOCB pin of GPT channel 3

23.2 Register Descriptions

23.2.1 GTDLYCR : PWM Output Delay Control Register

Base address: PDG = 0x4032_4000
PDG_NS = 0x5032_4000

Offset address: 0x0000

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

Bit	Symbol	Function	R/W
0	DLLEN	DLL Operation Enable 0: DLL operation disabled 1: DLL operation enabled	R/W
1	DLYRST	PWM Delay Generation Circuit Reset 0: Normal operation 1: Reset	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	FRANGE[1:0]	GPT core clock Frequency Range 0 0: GPT core clock frequency is 80 MHz to 160 MHz 0 1 GPT core clock frequency is 155 MHz to 300 MHz Others: Setting prohibited	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

The GTDLYCR register controls the PWM delay generation circuit, which applies delays to the PWM outputs. GTDLYCR register can be written when register write protection is disabled (GPT320.GTWP.WP = 0).

When changing GTDLYCR after changing the value of GPT320.GTWP.WP bit, be sure to read back the value of GTWP register before changing the value of GTDLYCR.

DLLEN bit (DLL Operation Enable)

The DLLEN bit selects whether the on-chip DLL in the PWM delay generation circuit is activated or not.

DLYRST bit (PWM Delay Generation Circuit Reset)

The DLYRST bit resets the internal state of the PWM delay generation circuit.

FRANGE[1:0] bit (GPT core clock Frequency Range)

The FRANGE[1:0] bits set the frequency range of the GPT core clock.

Set the FRANGE[1:0] bits only when the DLLEN bit is 0.

23.2.2 GTDLYCR2 : PWM Output Delay Control Register 2

Base address: PDG = 0x4032_4000
PDG_NS = 0x5032_4000

Offset address: 0x0002

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	DLYEN3	DLYEN2	DLYEN1	DLYEN0	—	—	—	—	DLYBS3	DLYBS2	DLYBS1	DLYBS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DLYBS0	PWM Delay Generation Circuit bypass for channel 0 0: Delay generation circuit of channel 0 bypassed 1: Delay generation circuit of channel 0 not bypassed	R/W
1	DLYBS1	PWM Delay Generation Circuit bypass for channel 1 0: Delay generation circuit of channel 1 bypassed 1: Delay generation circuit of channel 1 not bypassed	R/W
2	DLYBS2	PWM Delay Generation Circuit bypass for channel 2 0: Delay generation circuit of channel 2 bypassed 1: Delay generation circuit of channel 2 not bypassed	R/W
3	DLYBS3	PWM Delay Generation Circuit bypass for channel 3 0: Delay generation circuit of channel 3 bypassed 1: Delay generation circuit of channel 3 not bypassed	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	DLYEN0	PWM Delay Generation Circuit enable for channel 0 0: Delay generation circuit of channel 0 enabled 1: Delay generation circuit of channel 0 disabled	R/W
9	DLYEN1	PWM Delay Generation Circuit enable for channel 1 0: Delay generation circuit of channel 1 enabled 1: Delay generation circuit of channel 1 disabled	R/W
10	DLYEN2	PWM Delay Generation Circuit enable for channel 2 0: Delay generation circuit of channel 2 enabled 1: Delay generation circuit of channel 2 disabled	R/W
11	DLYEN3	PWM Delay Generation Circuit enable for channel 3 0: Delay generation circuit of channel 3 enabled 1: Delay generation circuit of channel 3 disabled	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W

The GTDLYCR2 register controls each channel of PWM delay generation circuit. GTDLYCR2 can be written when register write protection is disabled (GPT320.GTWP.WP = 0).

When changing GTDLYCR2 after changing the value of GPT320.GTWP.WP bit, be sure to read back the value of GTWP register before changing the value of GTDLYCR2.

DLYBSn (n = 0 to 3) bit (PWM Delay Generation Circuit Bypass for channel n)

The DLYBSn bit selects whether delays are applied to PWM output signals from the GTIOCnA and GTIOCnB pins (n = 0 to 3) by the PWM delay generation circuit or whether the circuit is bypassed.

A signal delayed in the PWM delay generation circuit is output 3 cycles of GPT core clock (GTCLK) later than if it bypasses the PWM delay generation circuit.

DLYENn (n = 0 to 3) bit (PWM Delay Generation Circuit Enable for channel n)

The DLYENn bit selects whether channel n (n = 0 to 3) of PWM delay generation circuit is power on or off. If channel n of the PWM delay generation circuit is not used, set to 1.

23.2.3 GTDLYRnA : GTIOCnA Rising Output Delay Register (n = 0 to 3)

Base address: PDG = 0x4032_4000
PDG_NS = 0x5032_4000

Offset address: 0x018 + 0x4 × n

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

Bit	Symbol	Function	R/W
6:0	DLY[6:0]	GTIOCnA Output Rising Edge Delay Setting (1) Case of GTDLYCR.FRANGE[1:0] = 00b 0x00: Delay on rising edges is not applied. 0x01: Delay of 1/128 times GTCLK period applied. 0x02: Delay of 2/128 times GTCLK period applied. 0x03: Delay of 3/128 times GTCLK period applied. 0x04: Delay of 4/128 times GTCLK period applied. 0x05: Delay of 5/128 times GTCLK period applied. 0x06: Delay of 6/128 times GTCLK period applied. ⋮ 0x7B: Delay of 123/128 times GTCLK period applied. 0x7C: Delay of 124/128 times GTCLK period applied. 0x7D: Delay of 125/128 times GTCLK period applied. 0x7E: Delay of 126/128 times GTCLK period applied. 0x7F: Delay of 127/128 times GTCLK period applied. (2) Case of GTDLYCR.FRANGE[1:0] = 01b 0x00: Delay on rising edges is not applied. 0x01: Delay on rising edges is not applied. 0x02: Delay of 1/64 times GTCLK period applied. 0x03: Delay of 1/64 times GTCLK period applied. 0x04: Delay of 2/64 times GTCLK period applied. 0x05: Delay of 2/64 times GTCLK period applied. 0x06: Delay of 3/64 times GTCLK period applied. ⋮ 0x7B: Delay of 61/64 times GTCLK period applied. 0x7C: Delay of 62/64 times GTCLK period applied. 0x7D: Delay of 62/64 times GTCLK period applied. 0x7E: Delay of 63/64 times GTCLK period applied. 0x7F: Delay of 63/64 times GTCLK period applied.	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W

The GTDLYRnA register sets a delay to be applied to rising edges of output signals on the GTIOCnA pin. On the timing for the transfer of settings, see section 23.3.2. Timing for Transfer of GTDLYRnA, GTDLYRnB, GTDLYFnA, and GTDLYFnB Register Settings.

GTDLYRnA can be written when register write protection is disabled (GPT32n.GTWP.WP = 0).

When changing GTDLYRnA after changing the value of GPT32n.GTWP.WP bit, be sure to read back the value of GTWP register before changing the value of GTDLYRnA.

23.2.4 GTDLYFnA : GTIOCnA Falling Output Delay Register (n = 0 to 3)

Base address: PDG = 0x4032_4000
PDG_NS = 0x5032_4000

Offset address: 0x028 + 0x4 × n

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

Bit	Symbol	Function	R/W
6:0	DLY[6:0]	GTIOCnA Output Falling Edge Delay Setting (1) Case of GTDLYCR.FRANGE[1:0] = 00b 0x00: Delay on rising edges is not applied. 0x01: Delay of 1/128 times GTCLK period applied. 0x02: Delay of 2/128 times GTCLK period applied. 0x03: Delay of 3/128 times GTCLK period applied. 0x04: Delay of 4/128 times GTCLK period applied. 0x05: Delay of 5/128 times GTCLK period applied. 0x06: Delay of 6/128 times GTCLK period applied. ⋮ 0x7B: Delay of 123/128 times GTCLK period applied. 0x7C: Delay of 124/128 times GTCLK period applied. 0x7D: Delay of 125/128 times GTCLK period applied. 0x7E: Delay of 126/128 times GTCLK period applied. 0x7F: Delay of 127/128 times GTCLK period applied. (2) Case of GTDLYCR.FRANGE[1:0] = 01b 0x00: Delay on rising edges is not applied. 0x01: Delay on rising edges is not applied. 0x02: Delay of 1/64 times GTCLK period applied. 0x03: Delay of 1/64 times GTCLK period applied. 0x04: Delay of 2/64 times GTCLK period applied. 0x05: Delay of 2/64 times GTCLK period applied. 0x06: Delay of 3/64 times GTCLK period applied. ⋮ 0x7B: Delay of 61/64 times GTCLK period applied. 0x7C: Delay of 62/64 times GTCLK period applied. 0x7D: Delay of 62/64 times GTCLK period applied. 0x7E: Delay of 63/64 times GTCLK period applied. 0x7F: Delay of 63/64 times GTCLK period applied.	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W

The GTDLYFnA register sets a delay to be applied to falling edges of output signals on the GTIOCnA pin. On the timing for the transfer of settings, see section 23.3.2. Timing for Transfer of GTDLYRnA, GTDLYRnB, GTDLYFnA, and GTDLYFnB Register Settings.

GTDLYFnA can be written when register write protection is disabled (GPT32n.GTWP.WP = 0).

When changing GTDLYFnA after changing the value of GPT32n.GTWP.WP bit, be sure to read back the value of GTWP register before changing the value of GTDLYFnA.

23.2.5 GTDLYRnB : GTIOCnB Rising Output Delay Register (n = 0 to 3)

Base address: PDG = 0x4032_4000
PDG_NS = 0x5032_4000

Offset address: 0x01A + 0x4 × n

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

Bit	Symbol	Function	R/W
6:0	DLY[6:0]	GTIOCnB Output Rising Edge Delay Setting (1) Case of GTDLYCR.FRANGE[1:0] = 00b 0x00: Delay on rising edges is not applied. 0x01: Delay of 1/128 times GTCLK period applied. 0x02: Delay of 2/128 times GTCLK period applied. 0x03: Delay of 3/128 times GTCLK period applied. 0x04: Delay of 4/128 times GTCLK period applied. 0x05: Delay of 5/128 times GTCLK period applied. 0x06: Delay of 6/128 times GTCLK period applied. ⋮ 0x7B: Delay of 123/128 times GTCLK period applied. 0x7C: Delay of 124/128 times GTCLK period applied. 0x7D: Delay of 125/128 times GTCLK period applied. 0x7E: Delay of 126/128 times GTCLK period applied. 0x7F: Delay of 127/128 times GTCLK period applied. (2) Case of GTDLYCR.FRANGE[1:0] = 01b 0x00: Delay on rising edges is not applied. 0x01: Delay on rising edges is not applied. 0x02: Delay of 1/64 times GTCLK period applied. 0x03: Delay of 1/64 times GTCLK period applied. 0x04: Delay of 2/64 times GTCLK period applied. 0x05: Delay of 2/64 times GTCLK period applied. 0x06: Delay of 3/64 times GTCLK period applied. ⋮ 0x7B: Delay of 61/64 times GTCLK period applied. 0x7C: Delay of 62/64 times GTCLK period applied. 0x7D: Delay of 62/64 times GTCLK period applied. 0x7E: Delay of 63/64 times GTCLK period applied. 0x7F: Delay of 63/64 times GTCLK period applied.	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W

The GTDLYRnB register sets a delay to be applied to rising edges of output signals on the GTIOCnB pin. On the timing for the transfer of settings, see section 23.3.2. Timing for Transfer of GTDLYRnA, GTDLYRnB, GTDLYFnA, and GTDLYFnB Register Settings.

GTDLYRnB can be written when register write protection is disabled (GPT32n.GTWP.WP = 0).

When changing GTDLYRnB after changing the value of GPT32n.GTWP.WP bit, be sure to read back the value of GTWP register before changing the value of GTDLYRnB.

23.2.6 GTDLYFnB : GTIOCnB Falling Output Delay Register (n = 0 to 3)

Base address: PDG = 0x4032_4000
PDG_NS = 0x5032_4000

Offset address: 0x02A + 0x4 × n

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

Bit	Symbol	Function	R/W
6:0	DLY[6:0]	GTIOCnB Output Falling Edge Delay Setting (1) Case of GTDLYCR.FRANGE[1:0] = 00b 0x00: Delay on rising edges is not applied. 0x01: Delay of 1/128 times GTCLK period applied. 0x02: Delay of 2/128 times GTCLK period applied. 0x03: Delay of 3/128 times GTCLK period applied. 0x04: Delay of 4/128 times GTCLK period applied. 0x05: Delay of 5/128 times GTCLK period applied. 0x06: Delay of 6/128 times GTCLK period applied. ⋮ 0x7B: Delay of 123/128 times GTCLK period applied. 0x7C: Delay of 124/128 times GTCLK period applied. 0x7D: Delay of 125/128 times GTCLK period applied. 0x7E: Delay of 126/128 times GTCLK period applied. 0x7F: Delay of 127/128 times GTCLK period applied. (2) Case of GTDLYCR.FRANGE[1:0] = 01b 0x00: Delay on rising edges is not applied. 0x01: Delay on rising edges is not applied. 0x02: Delay of 1/64 times GTCLK period applied. 0x03: Delay of 1/64 times GTCLK period applied. 0x04: Delay of 2/64 times GTCLK period applied. 0x05: Delay of 2/64 times GTCLK period applied. 0x06: Delay of 3/64 times GTCLK period applied. ⋮ 0x7B: Delay of 61/64 times GTCLK period applied. 0x7C: Delay of 62/64 times GTCLK period applied. 0x7D: Delay of 62/64 times GTCLK period applied. 0x7E: Delay of 63/64 times GTCLK period applied. 0x7F: Delay of 63/64 times GTCLK period applied.	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W

The GTDLYFnB register sets a delay to be applied to falling edges of output signals on the GTIOCnB pin. On the timing for the transfer of settings, see section 23.3.2. Timing for Transfer of GTDLYRnA, GTDLYRnB, GTDLYFnA, and GTDLYFnB Register Settings.

GTDLYFnB can be written when register write protection is disabled (GPT32n.GTWP.WP = 0).

When changing GTDLYFnB after changing the value of GPT32n.GTWP.WP bit, be sure to read back the value of GTWP register before changing the value of GTDLYFnB.

23.3 Operation

23.3.1 Adjustments to the Timing of Rising and Falling Edges in PWM Waveforms

The timing of rising and falling edges in PWM waveforms which are output from the GTIOCnA and GTIOCnB pins, where n = channel number, can be delayed to an accuracy of 1/64 or 1/128 of the GPT core clock (GTCLK) period.

If the timing of rising or falling edges in PWM waveforms output from the GTIOCnA and GTIOCnB pins must be adjusted, initialize the PWM generation circuit as shown in the procedure in Figure 23.2.

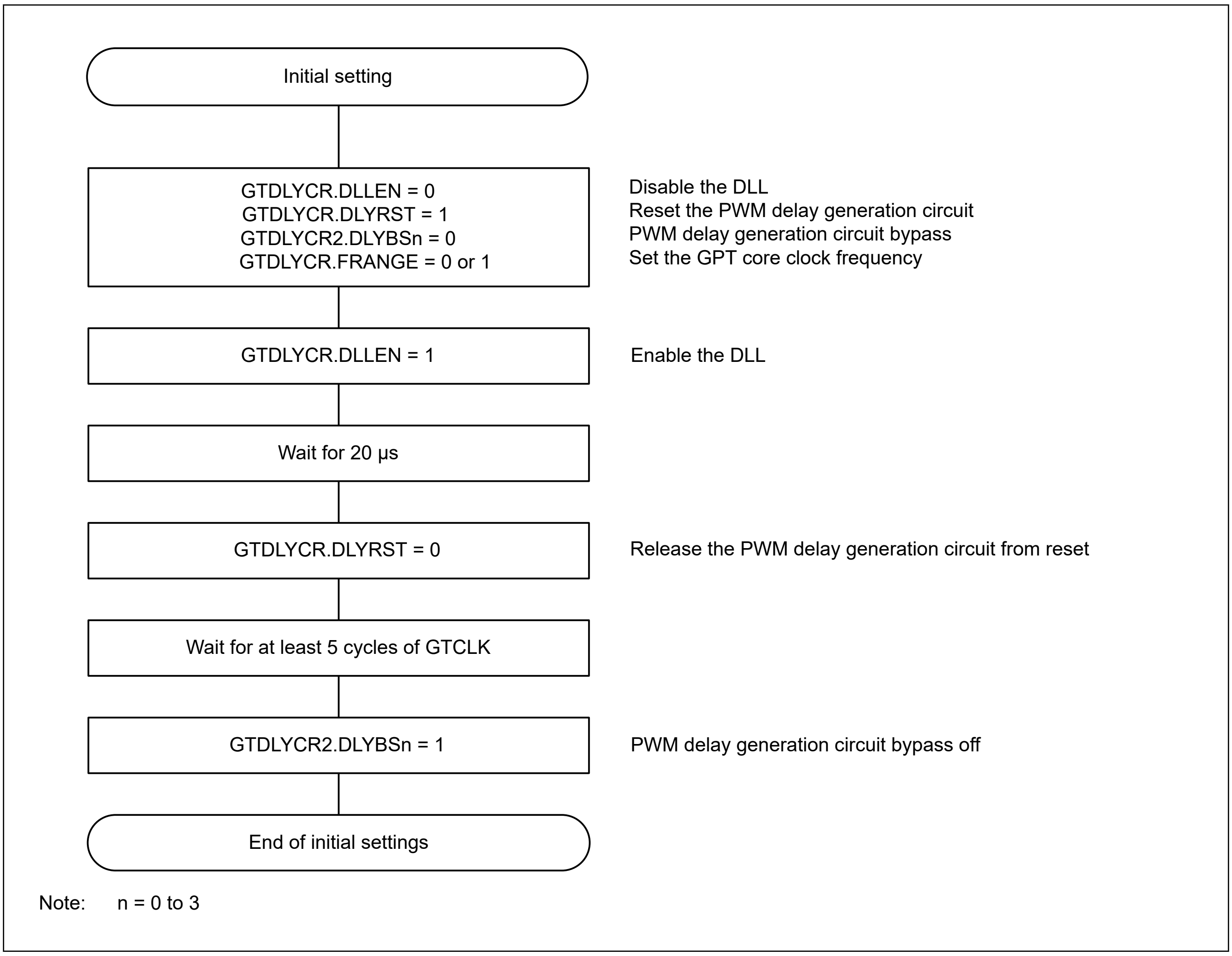


Figure 23.2 Example of initialization flow for the PWM delay generation circuit

In the PWM delay generation circuit, delay can be applied to rising and falling edges of the PWM output to an accuracy of 1/64 or 1/128 of the period of the GPT core clock (GTCLK). This is described in section 22.3.3. PWM Output Operating Mode. Delays associated with the settings are reflected in the PWM output with the timing described in section 23.3.2. Timing for Transfer of GTDLYRnA, GTDLYRnB, GTDLYFnA, and GTDLYFnB Register Settings. Table 23.3 shows the association between the GTDLYRnA, GTDLYRnB, GTDLYFnA, and GTDLYFnB registers and the PWM outputs.

Table 23.3 Association between PWM output pins and delay setting registers

PWM output pin	Rising-edge delay setting register	Falling-edge delay setting register
GTIOC0A	GTDLYR0A	GTDLYF0A
GTIOC0B	GTDLYR0B	GTDLYF0B
GTIOC1A	GTDLYR1A	GTDLYF1A
GTIOC1B	GTDLYR1B	GTDLYF1B
GTIOC2A	GTDLYR2A	GTDLYF2A
GTIOC2B	GTDLYR2B	GTDLYF2B
GTIOC3A	GTDLYR3A	GTDLYF3A
GTIOC3B	GTDLYR3B	GTDLYF3B

When the PWM delay generation circuit is in use, the timing with which a PWM output signal rises and falls can be controlled to an accuracy of 1/64 or 1/128 of the period of the GPT core clock (GTCLK). When this option is not in use, the period of the PWM output waveform is controlled to an accuracy of one period of the input clock for the timer counter, which is GTCLK. With the PWM delay generation circuit, the output can be controlled to an accuracy 32 times better.

Additionally, the delay settings also control the periods at high and low level for the PWM waveform to the given accuracy. PWM delay generation circuit channels can be individually enabled or disabled.

23.3.2 Timing for Transfer of GTDLYRnA, GTDLYRnB, GTDLYFnA, and GTDLYFnB Register Settings

Settings for the GTDLYRnA, GTDLYRnB, GTDLYFnA, and GTDLYFnB registers are initially transferred to temporary registers, and then reflected in the delay on the GTIOCnA and GTIOCnB (n = 0 to 3) outputs. Transfer of the settings takes place on overflows (in up-counting) or underflows (in down-counting) for saw waves, and in the troughs of triangle waves.

Figure 23.3 and Figure 23.4 show examples of the operation of the GTDLYR0A and GTDLYF0A registers.

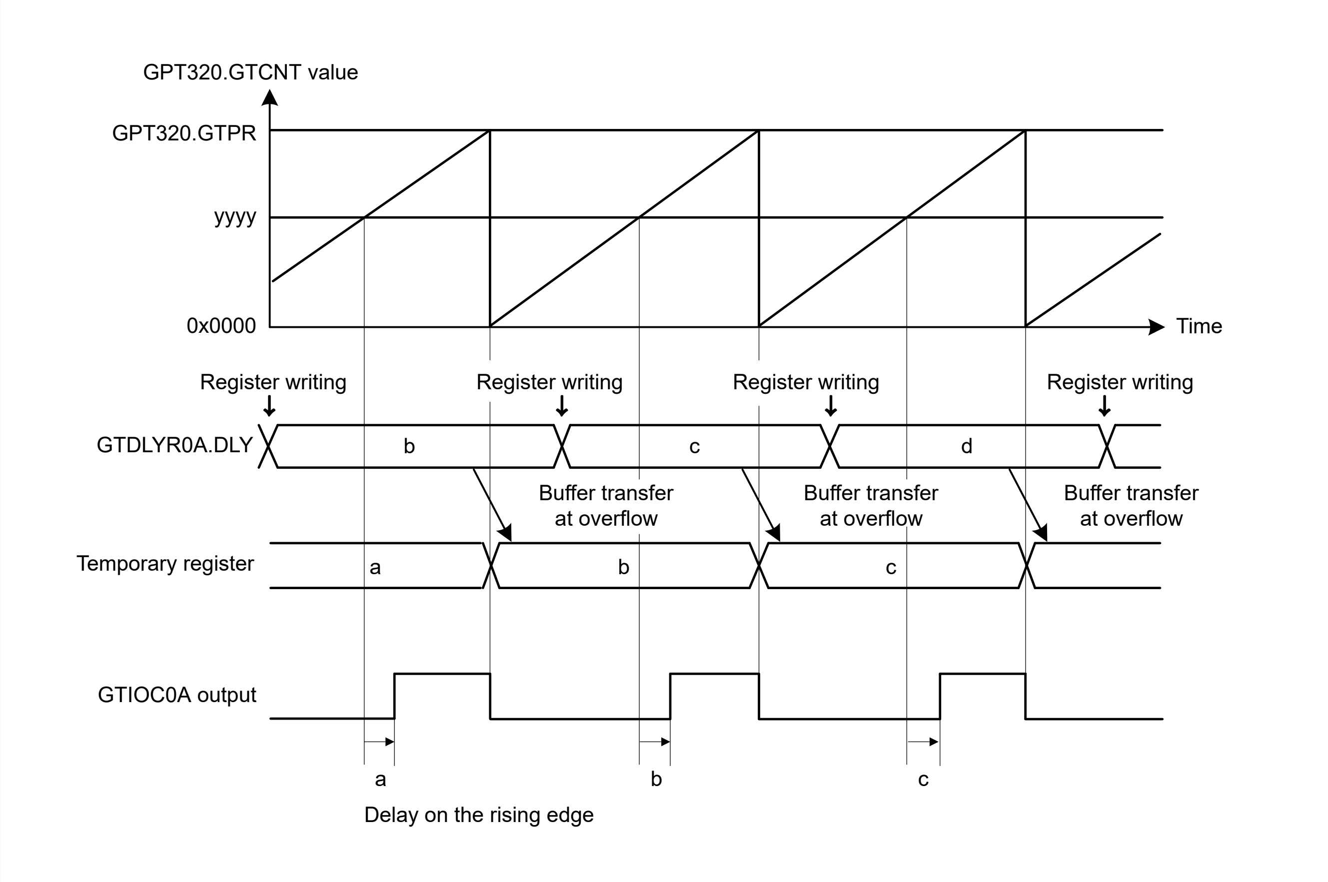


Figure 23.3 Example of GTDLYR0A register operation with PWM saw-wave generation

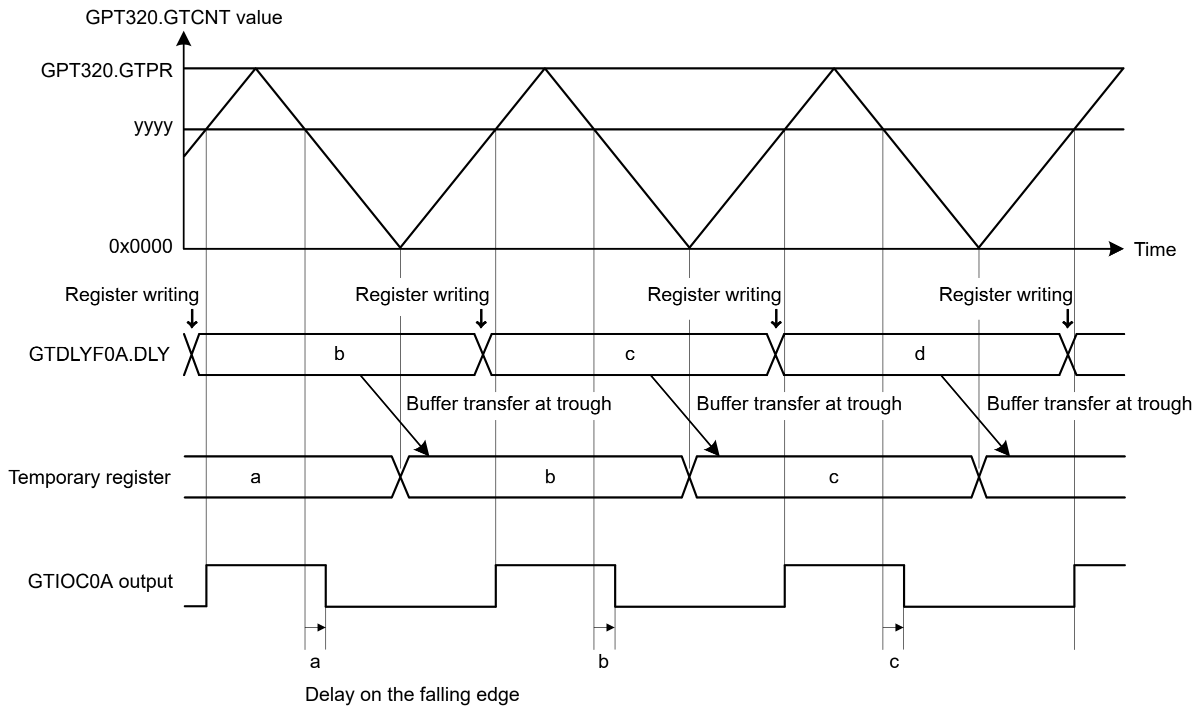


Figure 23.4 Example of GTDLYF0A register operation with PWM triangle-wave generation

23.4 Usage Notes

23.4.1 Settings for the Module-Stop Function

The Module Stop Control Register D (MSTPCRD) can enable or disable operation of the PWM delay generation circuit. The PWM delay generation circuit is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Mode.

23.4.2 Notes on Delay Settings for PWM Delay Generation Circuit

When the PWM delay generation circuit generates delays for a PWM output waveform and the waveform is toggled in response to compare-matches, do not change the settings for delay while the compare-match value is within the ranges listed in Table 23.4. This constraint applies to the GTDLYFnA, GTDLYRnA, GTDLYFnB, and GTDLYRnB registers.

Table 23.4 Constraints on delay settings

Mode	Direction of counting	Compare-match value
Saw-wave mode	Up	GTPR - 2 or above
	Down	2 or below
Triangle-wave mode	Down	2 or below

Figure 23.5 shows an example of how the constraints apply to the timing of setting GTDLYFnA in saw-wave waveform one-shot pulse mode (counting up). Do not change the value set in GTDLYFnA while GTCCRD ≥ GTPR - 2.

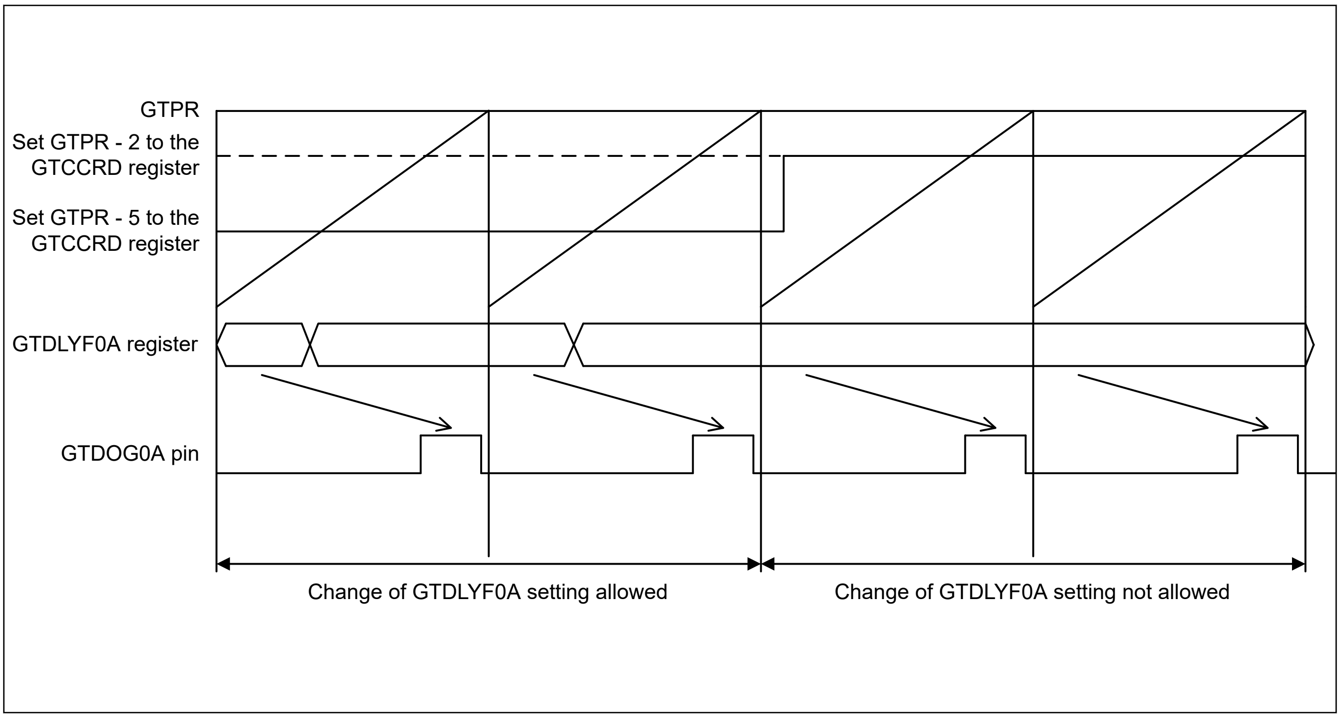


Figure 23.5 Constraints on the timing of GTDLYF0A register settings

Changing the values in the GTDLYFnA, GTDLYRnA, GTDLYFnB, and GTDLYRnB registers during periods where changes to settings are not allowed, might lead to faulty output waveforms such as shifts in the timing of output waveform transitions from the expected values.

23.4.3 Register Write Interval

When GPT core clock is GPTCLK, the writing value may not be reflected if the interval between writing to the GTDLYRnA/GTDLYFnA/GTDLYRnB/GTDLYFnA register is shorter than the following interval time. This restriction only applies to successive writes to the same register.

Write_Interval [ns] = Period_of_PCLKA [ns] × 6 + Period_of_GPTCLK [ns] × 4

24. Low Power Asynchronous General Purpose Timer (AGT)

24.1 Overview

The Low Power Asynchronous General Purpose Timer (AGT) is a 16-bit timer that can be used for pulse output, external pulse width or period measurement, and counting external events. This timer consists of a reload register and a down counter. The reload register and the down counter are allocated to the same address, and can be accessed with the AGT register.

Table 24.1 lists the AGT specifications, Figure 24.1 shows a block diagram, and Table 24.2 lists the I/O pins.

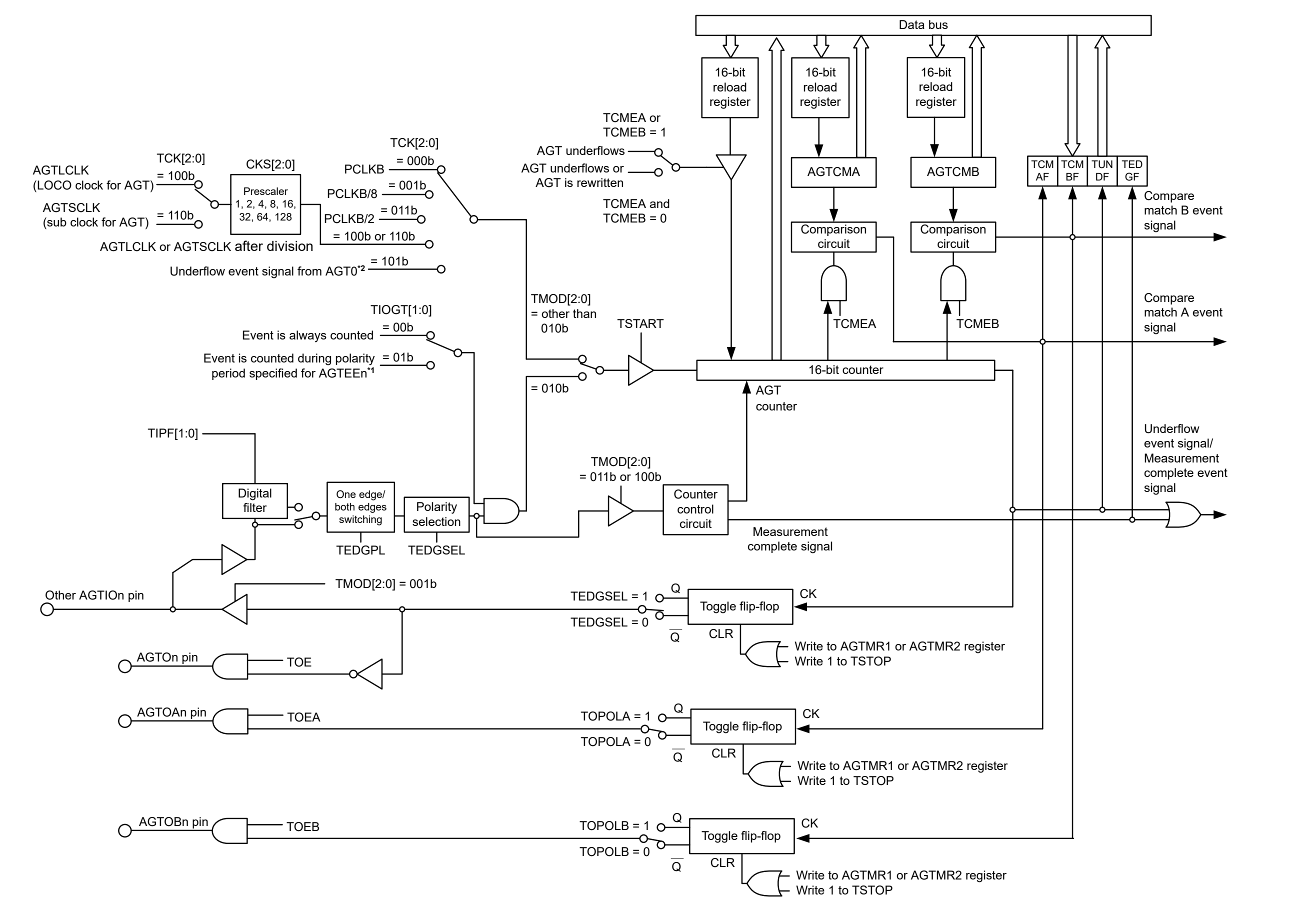
Table 24.1 AGT specifications

Parameter		Description
Operating modes	Timer mode	The count source is counted
	Pulse output mode	The count source is counted and the output is inverted at each timer underflow
	Event counter mode	An external event is counted
	Pulse width measurement mode	An external pulse width is measured
	Pulse period measurement mode	An external pulse period is measured
Number of Channels		16 bits × 2 channels (AGTn (n = 0, 1))
Count source (operating clock)[*2]	Timer mode	PCLKB, PCLKB/2, PCLKB/8, AGTLCLK/d, AGTSCLK/d (d = 1, 2, 4, 8, 16, 32, 64, or 128), or underflow signal of AGT0 selectable.[*1]
	Pulse output mode	
	Pulse width measurement mode	
	Pulse period measurement mode	
	Event counting mode	External event input
Interrupt and Event Link function		• Underflow event signal or measurement complete event signal – When the counter underflows – When the measurement of the active width of the external input pin (AGTIOn) completes in pulse width measurement mode – When the set edge of the external input pin (AGTIOn) is input in pulse period measurement mode. • Compare match A event signal – When the values of AGT register and AGTCMA register matched (compare match A function enabled). • Compare match B event signal – When the values of AGT and AGTCMB registers matched (compare match B function enabled). • Return from Software Standby mode can be performed with AGT1_AGTI, AGT1_AGTCMAI, or AGT1_AGTCMBI[*3]
Selectable functions		• Compare match function One or two of the AGT Compare Match A register and AGT Compare Match B register is selectable.
Module-stop function		Module-stop state can be set for each channel to reduce power consumption.
TrustZone Filter		Security and Privilege attribution can be set for each channel.

Note 1. AGT0 cannot use underflow signal. AGT1 connects directly with the underflow event signal from the AGT0 timer.

Note 2. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ the frequency of the count source clock. When selecting AGTLCLK/d or AGTSCLK/d as the count source clock, the frequency of PCLKB should be at least twice that of the count source clock.

Note 3. For details, see section 11, Low Power Mode.



Note 1. The polarity can be selected by the EEPS bit in the AGTISR register.
Note 2. AGT0 cannot use AGT0 underflow event. AGT1 uses the underflow of AGT0.

Figure 24.1 AGT block diagram

Table 24.2 AGT I/O pins

Pin name	I/O	Function
AGTEEn	Input	External event input enable for AGT
AGTIOn	Input/output	External event input and pulse output for AGT
AGTOn	Output	Pulse output for AGT
AGTOAn	Output	Compare match A output for AGT
AGTOBn	Output	Compare match B output for AGT

Note: Channel number: n = 0, 1

24.2 Register Descriptions

24.2.1 AGT : AGT Counter Register

Base address: AGTn = 0x4022_1000 + 0x0100 × n (n = 0, 1)
AGTn_NS = 0x5022_1000 + 0x0100 × n (n = 0, 1)

Offset address: 0x00

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

Bit	Symbol	Function	R/W
15:0	n/a	16-bit counter and reload register Setting range : 0x0000 to 0xFFFF	R/W

Note: S-TYPE-3, P-TYPE-3

AGTn.AGT is a 16-bit register. The write value is written to the reload register and the read value is read from the counter.

The states of the reload register and the counter change according to the TSTART bit in the AGTCR register and TCMEA/TCMEB bit in the AGTCMSR register. For details, see section 24.3.1. Reload Register and Counter Rewrite Operation.

When 1 is written to the TSTOP bit in the AGTCR register, AGT counter is forcibly stopped and set to 0xFFFF.

When the TCK[2:0] bits setting in the AGTMR1 register are a value other than 001b (PCLKB/8) or 011b (PCLKB/2), if the AGT register is set to 0x0000, a request signal to the ICU, the DTC, the DMAC, and the ELC is generated once immediately after the count starts. The AGTOn, AGTIOn pin output are toggled.

When the AGT register is set to 0x0000 in event counter mode, regardless of the value of TCK[2:0] bits, a request signal to the ICU, the DTC, the DMAC, and the ELC is generated once immediately after the count starts.

In addition, the AGTOn pin output is toggled even during a period other than the specified count period. When the AGT register is set to 0x0001 or more, a request signal is generated each time AGT underflows.

24.2.2 AGTCMA : AGT Compare Match A Register

Base address: AGTn = 0x4022_1000 + 0x0100 × n (n = 0, 1)
AGTn_NS = 0x5022_1000 + 0x0100 × n (n = 0, 1)

Offset address: 0x02

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

Bit	Symbol	Function	R/W
15:0	n/a	16-bit compare match A data is stored.*1 Setting range : 0x0000 to 0xFFFF	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Set the AGTCMA register to 0xFFFF when compare match A is not used.

The AGTCMA register is a read/write register to set a value for compare match with the AGT counter. The states of the reload register and compare register A change according to the TSTART bit in the AGTCR register. For details, see section 24.3.2. Reload Register and AGT Compare Match A/B Register Rewrite Operation.

24.2.3 AGTCMB : AGT Compare Match B Register

Base address: AGTn = 0x4022_1000 + 0x0100 × n (n = 0, 1)
AGTn_NS = 0x5022_1000 + 0x0100 × n (n = 0, 1)

Offset address: 0x04

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

Bit	Symbol	Function	R/W
15:0	n/a	16-bit compare match B data is stored.*1 Setting range : 0x0000 to 0xFFFF	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Set the AGTCMB register to 0xFFFF when compare match B is not used.

The AGTCMB register is a read/write register to set a value for compare match with the AGT counter. The states of the reload register and compare register B change according to the TSTART bit in the AGTCR register. For details, see section 24.3.2. Reload Register and AGT Compare Match A/B Register Rewrite Operation.

24.2.4 AGTCR : AGT Control Register

Base address: AGTn = 0x4022_1000 + 0x0100 × n (n = 0, 1)
AGTn_NS = 0x5022_1000 + 0x0100 × n (n = 0, 1)

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TCMBF	TCMAF	TUNDF	TEDGF	—	TSTOP	TCSTF	TSTART
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TSTART	AGT Count Start*2 0: Count stops 1: Count starts	R/W
1	TCSTF	AGT Count Status Flag*2 0: Count stopped 1: Count in progress	R
2	TSTOP	AGT Count Forced Stop*1 0: Writing is invalid 1: The count is forcibly stopped	W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	TEDGF	Active Edge Judgment Flag 0: No active edge received 1: Active edge received	R/(W)*3
5	TUNDF	Underflow Flag 0: No underflow 1: Underflow	R/(W)*3
6	TCMAF	Compare Match A Flag 0: No match 1: Match	R/(W)*3
7	TCMBF	Compare Match B Flag 0: No match 1: Match	R/(W)*3

Note: S-TYPE-3, P-TYPE-3
Note 1. When 1 (count is forcibly stopped) is written to the TSTOP bit, the TSTART bit and TCSTF flag are initialized at the same time. The pulse output level is also initialized. The read value is 0.
Note 2. For information on using the TSTART bit and TCSTF flag, see section 24.4.1. Count Operation Start and Stop Control.
Note 3. Only 0 can be written to clear the flag.

TSTART bit (AGT Count Start)

The count operation is started by writing 1 to the TSTART bit and stopped by writing 0. When the TSTART bit is set to 1 (count starts), the TCSTF flag is set to 1 (count in progress) in synchronization with the count source. Also, after 0 is written to the TSTART bit, the TCSTF flag is set to 0 (count stops) in synchronization with the count source. For details, see section 24.4.1. Count Operation Start and Stop Control.

TCSTF flag (AGT Count Status Flag)

The TCSTF flag indicates the AGT count status.

[Setting condition]

- When 1 is written to the TSTART bit (the TCSTF flag is set to 1 in synchronization with the count source).

[Clearing conditions]

- When 0 is written to the TSTART bit (the TCSTF flag is set to 0 in synchronization with the count source)
- When 1 is written to the TSTOP bit.

TSTOP bit (AGT Count Forced Stop)

When 1 is written to the TSTOP bit, the count is forcibly stopped. The read value is 0.

TEDGF flag (Active Edge Judgment Flag)

The TEDGF flag indicates that an active edge was detected.

[Setting condition]

- When the measurement of the active width of the external input pin (AGTIOn) is complete in pulse width measurement mode
- When the set edge of the external input pin (AGTIOn) is input in pulse period measurement mode.

[Clearing condition]

- When 0 is written to this flag by software.

TUNDF flag (Underflow Flag)

The TUNDF flag indicates that the counter underflowed.

[Setting condition]

- When the counter underflows.

[Clearing condition]

- When 0 is written to this flag by software.

TCMAF flag (Compare Match A Flag)

The TCMAF flag indicates that compare match A was detected.

[Setting condition]

- When the value in the AGT register matches the value in the AGTCMA register.

[Clearing condition]

- When 0 is written to this flag by software.

TCMBF flag (Compare Match B Flag)

The TCMBF flag indicates that compare match B was detected.

[Setting condition]

- When the value in the AGT register matches the value in the AGTCMB register.

[Clearing condition]

- When 0 is written to this flag by software.

24.2.5 AGTMR1 : AGT Mode Register 1

Base address: AGTn = 0x4022_1000 + 0x0100 × n (n = 0, 1)
AGTn_NS = 0x5022_1000 + 0x0100 × n (n = 0, 1)

Offset address: 0x09

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	TCK[2:0]			TEDGPL	TMOD[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TMOD[2:0]	Operating Mode*3 0 0 0: Timer mode 0 0 1: Pulse output mode 0 1 0: Event counter mode 0 1 1: Pulse width measurement mode 1 0 0: Pulse period measurement mode Others: Setting prohibited	R/W
3	TEDGPL	Edge Polarity*4 0: Single-edge 1: Both-edge	R/W
6:4	TCK[2:0]	Count Source*1 *2 *5 *7 0 0 0: PCLKB 0 0 1: PCLKB/8 0 1 1: PCLKB/2 1 0 0: Divided clock AGTLCLK specified by CKS[2:0] bits in the AGTMR2 register 1 0 1: Underflow event signal from AGT0*6 1 1 0: Divided clock AGTSCLK specified by CKS[2:0] bits in the AGTMR2 register Others: Setting prohibited	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Write access to the AGTMR1 register initializes the output from the AGTOn, AGTIOn, AGTOAn, and AGTOBn pins. For details on the output level at initialization, see section 24.2.7. AGTIOC : AGT I/O Control Register.
Note 1. When event counter mode is selected, the external input pin (AGTIOn) is selected as the count source regardless of the setting of TCK[2:0] bits.
Note 2. Do not switch count sources during count operation. Only switch count sources when both the TSTART bit and TCSTF flag in the AGTCR register are set to 0 (count stops).
Note 3. The operating mode can only be changed when the count is stopped while both the TSTART bit and TCSTF flag in the AGTCR register are set to 0 (count is stopped). Do not change the operating mode during count operation.
Note 4. The TEDGPL bit is enabled only in event counter mode.
Note 5. To run AGT in Software Standby mode, select AGTLCLK or AGTSCLK (TCK[2:0] = 100b, 110b).
Note 6. AGT0 cannot use AGT0 underflow (setting prohibited). AGT1 uses the AGT0 underflow.
Note 7. Do not change the TCK[2:0] bits when the CKS[2:0] bits in the AGTMR2 register is not 000b. First, change the CKS[2:0] bits in the AGTMR2 register to 000b. Then change the TCK[2:0] bits and wait for one cycle of the count source.

24.2.6 AGTMR2 : AGT Mode Register 2

Base address: AGTn = 0x4022_1000 + 0x0100 × n (n = 0, 1)
AGTn_NS = 0x5022_1000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LPM	—	—	—	—	CKS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CKS[2:0]	AGTLCLK or AGTSCLK Count Source Clock Frequency Division Ratio*1 *2 *3 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128	R/W
6:3	—	These bits are read as 0. The write value should be 0.	R/W
7	LPM	Low Power Mode 0: Normal mode 1: Low power mode	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Do not rewrite the CKS[2:0] bits during count operation. Only rewrite the CKS[2:0] bits when both the TSTART bit and TCSTF flag in the AGTCR register are set to 0 (count stops).
Note 2. When count source is AGTLCLK or AGTSCLK, the switch of CKS[2:0] bits is valid.
Note 3. Do not switch the TCK[2:0] bits in the AGTMR1 register when CKS[2:0] bits are not 000b. Switch the TCK[2:0] bits in the AGTMR1 register after CKS[2:0] bits are set to 000b, and wait for 1 cycle of the count source.

CKS[2:0] bit (AGTLCLK or AGTSCLK Count Source Clock Frequency Division Ratio)

CKS[2:0] bits select the Count Source Clock Frequency Division Ratio for AGTLCLK or AGTSCLK.

LPM bit (Low Power Mode)

The LPM bit sets the low power operation, which impacts access to certain AGT registers. Set this bit to 1 to operate in low power.

When this bit is 1, access to the following registers is prohibited:

- AGT/AGTCMA/AGTCMB/AGTCR.

After this bit is switched from 1 to 0, the first access to the register is constrained as follows:

- When reading from the AGT register, read AGT register twice. Only the second reading of data is valid.
- When writing to the AGT, AGTCMA, AGTCMB, and AGTCR register, allow at least 2 cycles of the count source clock when writing to the register.
- When confirm the value written to the AGT, AGTCMA, AGTCMB, and AGTCR registers.
  - When the count operation is stopped; after writing data, it can be read in the next cycle.
  - When the count operation is operating; after writing data, it can be read 4 cycles after the count source clock.

Figure 24.2 shows the flow of how to write LPM bit

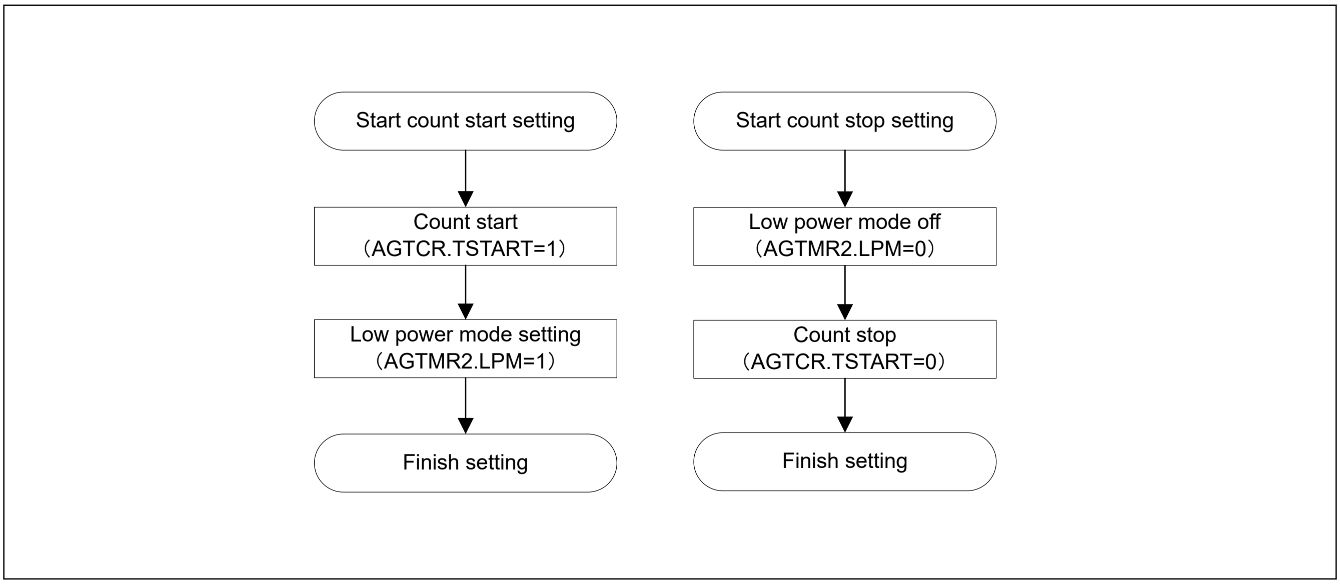


Figure 24.2 LPM how to write flow chart

24.2.7 AGTIOC : AGT I/O Control Register

Base address: AGTn = 0x4022_1000 + 0x0100 × n (n = 0, 1)
AGTn_NS = 0x5022_1000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TIOGT[1:0]		TIPF[1:0]		—	TOE	—	TEDGSEL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TEDGSEL	I/O Polarity Switch Function varies depending on the operating mode (see Table 24.3 and Table 24.4).	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	TOE	AGTOn pin Output Enable 0: AGTOn pin output disabled 1: AGTOn pin output enabled	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
5:4	TIPF[1:0]	Input Filter*3 These bits specify the sampling frequency of the filter for the AGTIOn input. If the input to the AGTIOn pin is sampled and the value matches three successive times, that value is taken as the input value. 0 0: No filter 0 1: Filter sampled at PCLKB 1 0: Filter sampled at PCLKB/8 1 1: Filter sampled at PCLKB/32	R/W
7:6	TIOGT[1:0]	Count Control*1 *2 0 0: Event is always counted 0 1: Event is counted during polarity period specified for AGTEEn pin Others: Setting prohibited	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. When AGTEEn pin is used, the polarity to count an event can be selected with the EEPS bit in the AGTISR register.
Note 2. TIOGT[1:0] bits are enabled only in event counter mode.
Note 3. When event counter mode operation is performed during Software Standby mode, the digital filter function cannot be used.

TEDGSEL bit (I/O Polarity Switch)

The TEDGSEL bit switches the AGTOn pin output polarity and the AGTIOn pin input/output edge and polarity.

In pulse output mode, it only controls polarity of the AGTOn pin output and AGTIOn pin output. AGTOn pin output and AGTIOn pin output are initialized when the AGTMR1 register is written or the TSTOP bit in the AGTCR register is written with 1.

TOE bit (AGTOn pin Output Enable)

The TOE bit selects whether the AGTOn pin output is disabled or enabled.

TIPF[1:0] bits (Input Filter)

The TIPF[1:0] bits specify the sampling frequency of the AGTIOn pin input filter. When the input to the AGTIOn pin is sampled and the values match three times in succession, the value is regarded as the input value.

TIOGT[1:0] bits (Count Control)

The TIOGT[1:0] bits control the event count.

Table 24.3 AGTIOn pin I/O edge and polarity switching

Operating mode	Function
Timer mode	Not used
Pulse output mode	0: Output is started at high (initialization level: high) i.e. inverted output 1: Output is started at low (initialization level: low). i.e. normal output
Event counter mode	0: Count on rising edge 1: Count on falling edge.
Pulse width measurement mode	0: Low-level width is measured 1: High-level width is measured.
Pulse period measurement mode	0: Measure from one rising edge to the next rising edge 1: Measure from one falling edge to the next falling edge.

Table 24.4 AGTOn pin output polarity switching

Operating mode	Function
All modes	0: Output is started at low (initial level: low): Normal output 1: Output is started at high (initial level: high): Inverted output

24.2.8 AGTISR : AGT Event Pin Select Register

Base address: AGTn = 0x4022_1000 + 0x0100 × n (n = 0, 1)
AGTn_NS = 0x5022_1000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0D

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

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
2	EEPS	AGTEEn Polarity Selection 0: An event is counted during the low-level period 1: An event is counted during the high-level period	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

EEPS bit (AGTEEn Polarity Selection)

The EEPS bit selects the polarity of events to be counted.

24.2.9 AGTCMSR : AGT Compare Match Function Select Register

Base address: AGTn = 0x4022_1000 + 0x0100 × n (n = 0, 1)
AGTn_NS = 0x5022_1000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	TOPOLB	TOEB	TCMEB	—	TOPOLA	TOEA	TCMEA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TCMEA	AGT Compare Match A Register Enable*1 *2 0: AGT Compare match A register disabled 1: AGT Compare match A register enabled	R/W
1	TOEA	AGTOAn Pin Output Enable*1 *2 0: AGTOAn pin output disabled 1: AGTOAn pin output enabled	R/W
2	TOPOLA	AGTOAn Pin Polarity Select*1 *2 0: AGTOAn pin output is started on low. i.e. normal output 1: AGTOAn pin output is started on high. i.e. inverted output	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	TCMEB	AGT Compare Match B Register Enable*1 *2 0: Compare match B register disabled 1: Compare match B register enabled	R/W
5	TOEB	AGTOBn Pin Output Enable*1 *2 0: AGTOBn pin output disabled 1: AGTOBn pin output enabled	R/W

Bit	Symbol	Function	R/W
6	TOPOLB	AGTOBn Pin Polarity Select[*1][*2] 0: AGTOBn pin output is started on low. i.e. normal output 1: AGTOBn pin output is started on high. i.e. inverted output	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Do not rewrite the AGTCMSR register during a count operation. Only rewrite the AGTCMSR register when both the TSTART bit and TCSTF flag in the AGTCR register are set to 0 (count stops).

Note 2. Do not set 1 when in pulse width measurement mode or pulse period measurement mode.

24.2.10 AGTIOSEL : AGT Pin Select Register

Base address: AGTn = 0x4022_1000 + 0x0100 × n (n = 0, 1)
AGTn_NS = 0x5022_1000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0F

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

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	TIES	AGTIOn Pin Input Enable 0: External event input is disabled during Software Standby mode 1: External event input is enabled during Software Standby mode	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The AGTIOSEL register sets the AGTIOn pin when using the AGTIOn pin in Software Standby mode.

TIES bit (AGTIOn Pin Input Enable)

The TIES bit enables or disables an external event input.

24.3 Operation

24.3.1 Reload Register and Counter Rewrite Operation

Regardless of the operating mode, the timing of the rewrite operation to the reload register and the counter differs depending on the value of the TSTART bit in the AGTCR register and of the TCMEA or TCMEB bit in the AGTCMSR register. When the TSTART bit is 0 (count stops), the count value is directly written to the reload register and the counter. When the TSTART bit is 1 (count starts) and the TCMEA bit and TCMEB bit are 0 (AGT compare match A/B register are invalid), the value is written to the reload register in synchronization with the count source, and then to the counter in synchronization with the next count source. When the TSTART bit is 1 (count starts) and the TCMEA bit or the TCMEB bit is 1 (AGT compare match A register or compare match B register is valid), the value is written to the reload register in synchronization with the count source, and then to the counter in synchronization with the underflow of the counter.

Figure 24.3 and Figure 24.4 show the timing of rewrite operation with TSTART bit value and TCMEA/TCMEB bit value.

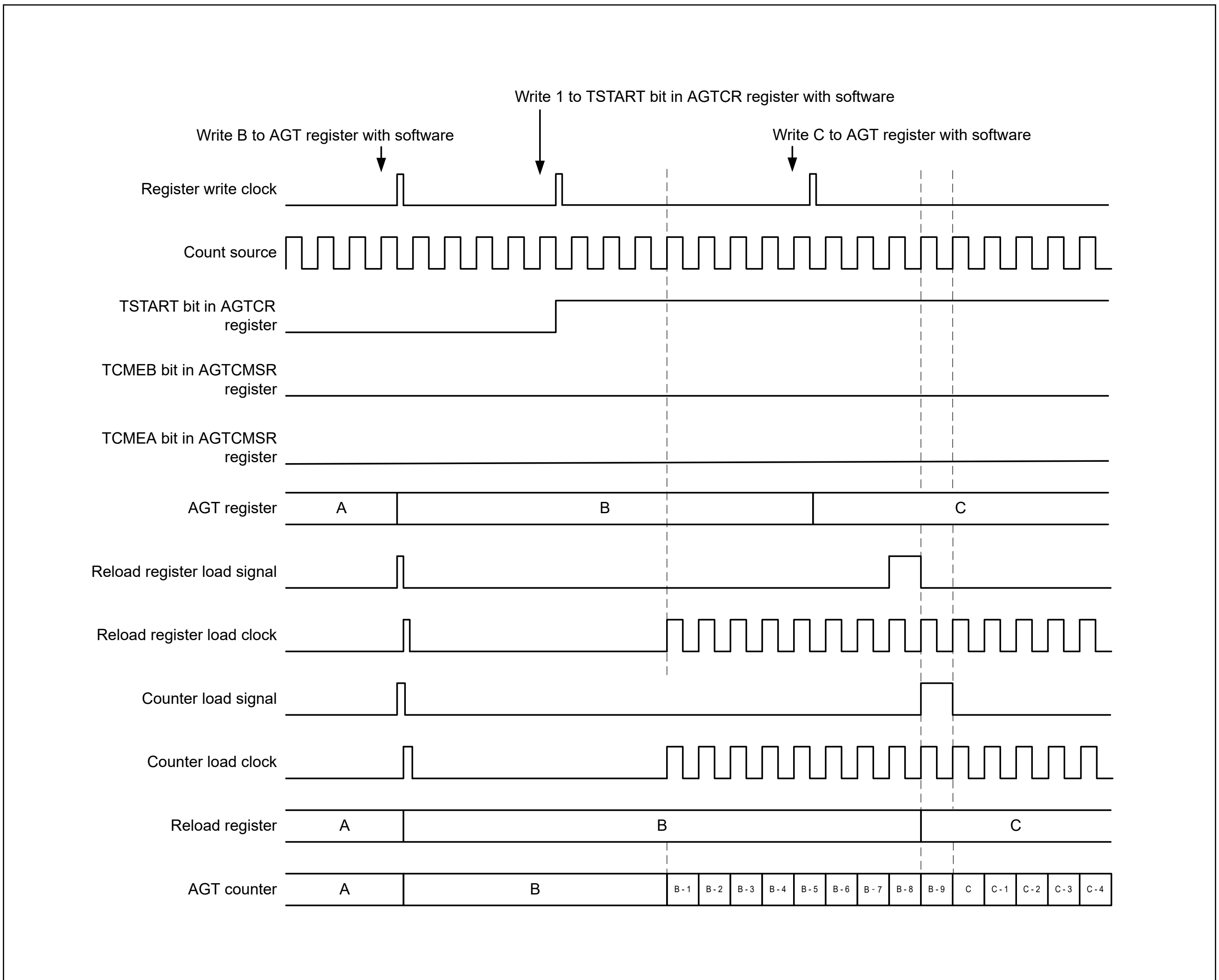


Figure 24.3 Timing of rewrite operation with TSTART, TCMEA, and TCMEB bit value when AGT compare match A register and AGT compare match B register is invalid

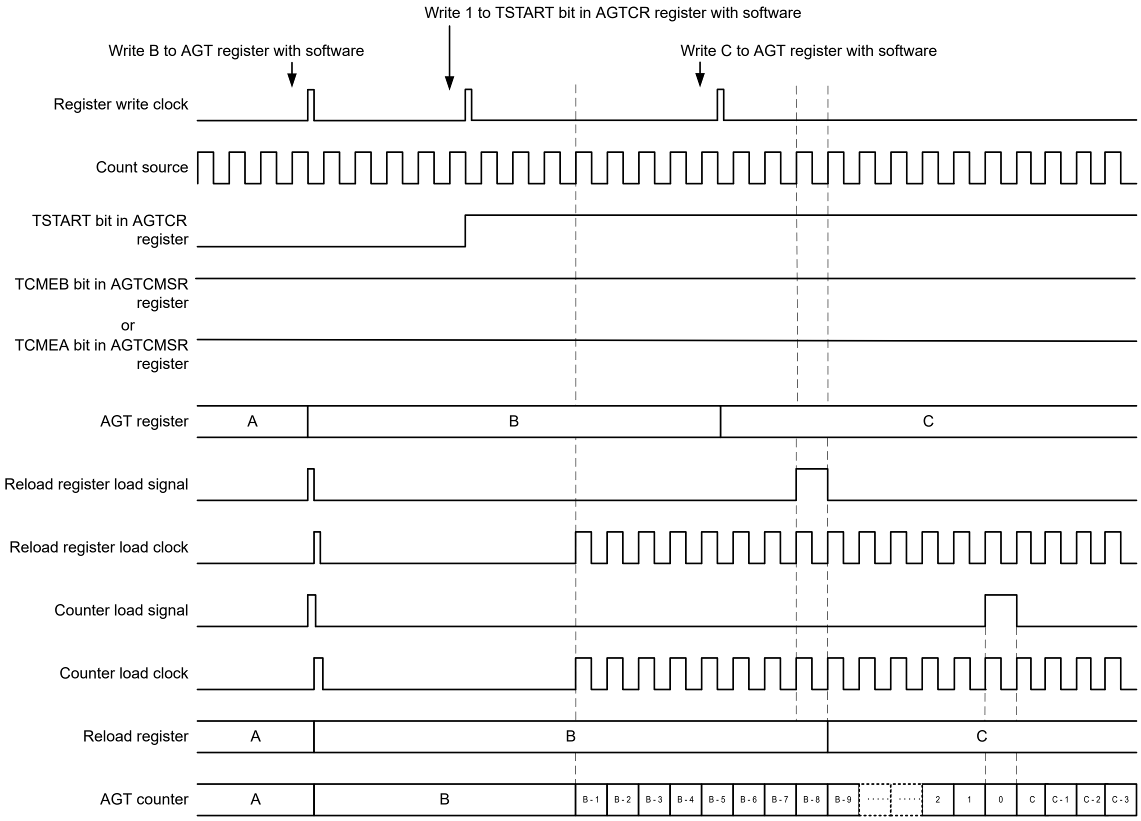


Figure 24.4 Timing of rewrite operation with TSTART bit value and TCMEA or TCMEB bit value when AGT compare match A register or AGT compare match B register is valid

24.3.2 Reload Register and AGT Compare Match A/B Register Rewrite Operation

Regardless of the operating mode, the timing of the rewrite operation to the reload register and AGT compare register A/B depends on the value of the TSTART bit in the AGTCR register. When the TSTART bit is 0 (count stops), the count value is directly written to the reload register and AGT compare register A/B. When the TSTART bit is 1 (count starts), the value is written to the reload register in synchronization with the count source, and then to the compare register in synchronization with the underflow of the counter.

Figure 24.5 shows the timing of rewrite operation with TSTART bit value for compare register A. AGT Compare register B is of the same timing as AGT compare register A.

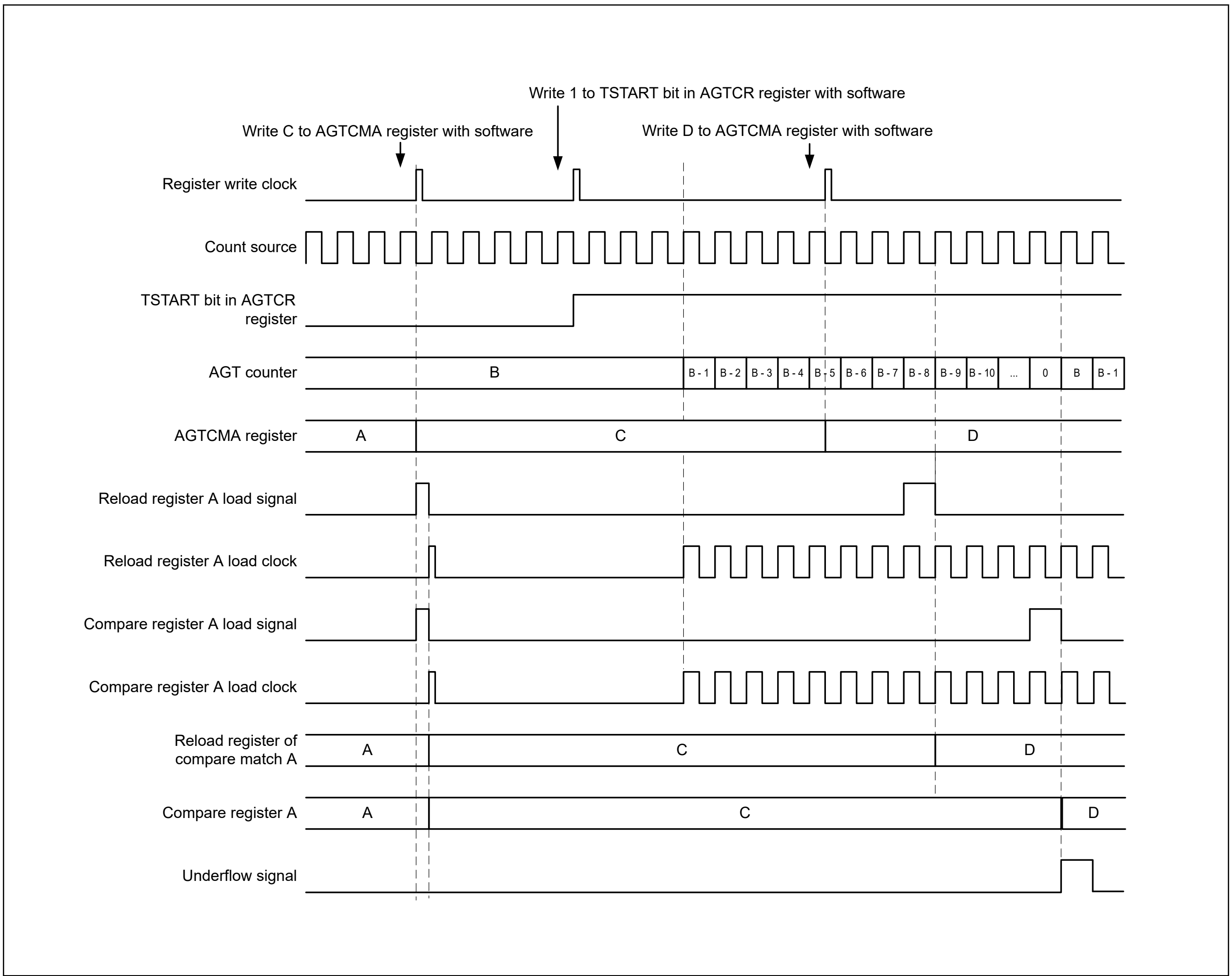


Figure 24.5 Timing of rewrite operation with the TSTART bit value for AGT compare register A

24.3.3 Timer Mode

In this mode, the AGT counter is decremented by the count source selected with the TCK[2:0] bits in the AGTMR1 register. In timer mode, the count value is decremented by 1 on each rising edge of the count source. When the count value reaches 0x0000 and the next count source is input, an underflow occurs, and an interrupt request is generated.

Figure 24.6 shows the operation example in timer mode.

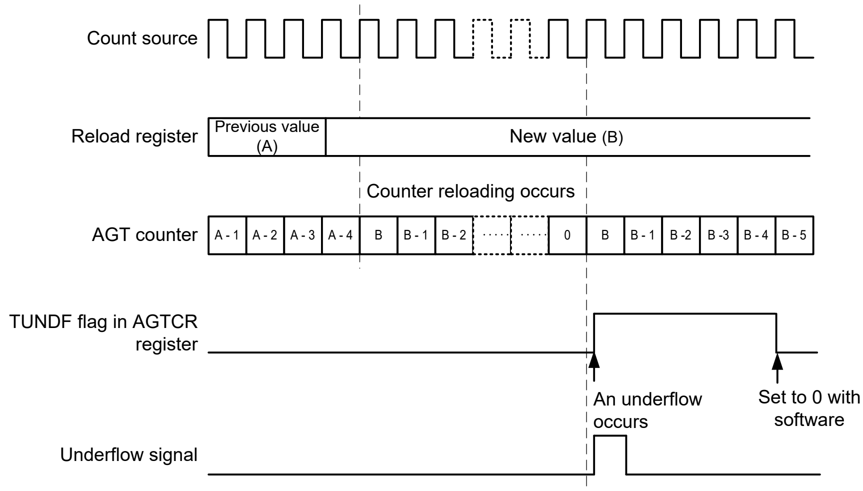


Figure 24.6 Operation example in timer mode

24.3.4 Pulse Output Mode

In pulse output mode, the counter is decremented by the count source selected with the TCK[2:0] bits in the AGTMR1 register, and the output level of the AGTIOn and AGTOn pins inverted each time an underflow occurs.

In pulse output mode, the count value is decremented by 1 on each rising edge of the count source. When the count value reaches 0x0000 and the next count source is input, an underflow occurs, and an interrupt request is generated. In addition, a pulse can be output from the AGTIOn and AGTOn pins. The output level is inverted each time an underflow occurs. The pulse output from the AGTOn pin can be stopped with the TOE bit in the AGTIOC register. The output level can be selected with the TEDGSEL bit in the AGTIOC register.

Figure 24.7 shows the operation example in pulse output mode.

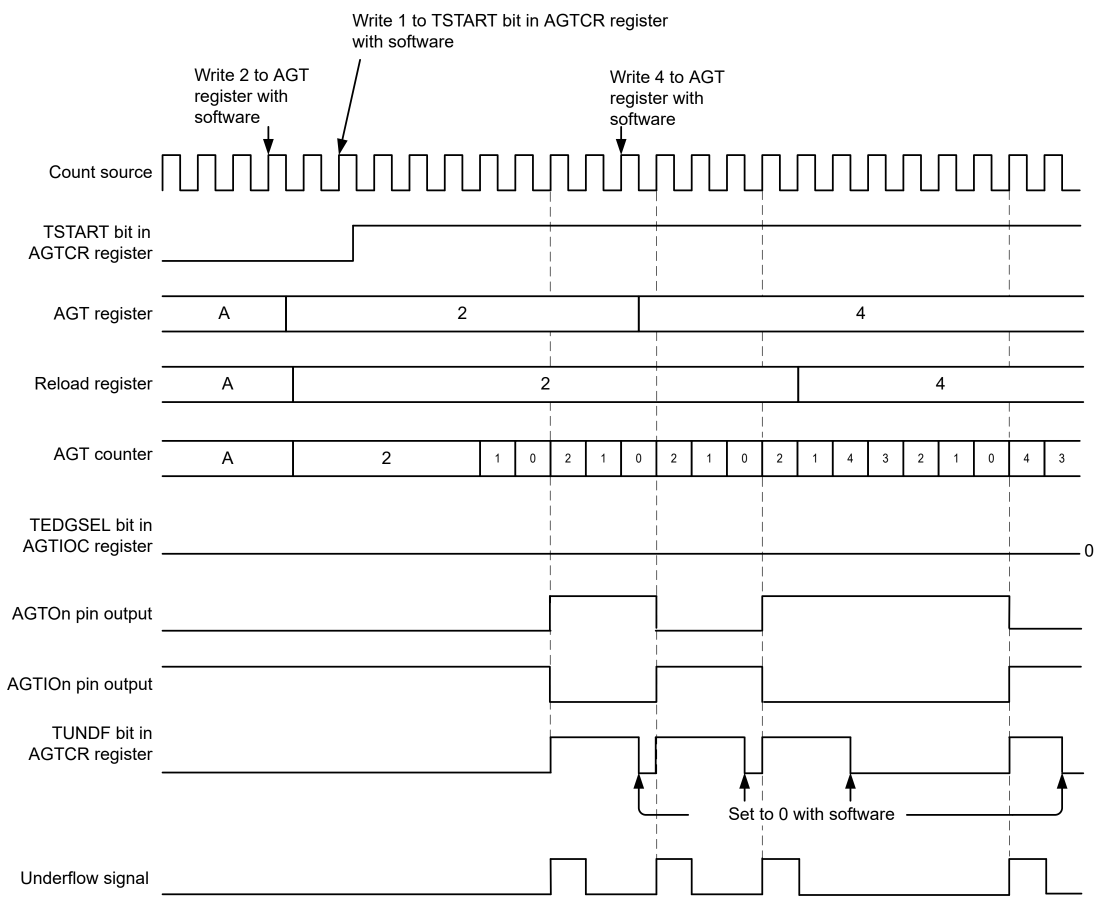


Figure 24.7 Operation example in pulse output mode

24.3.5 Event Counter Mode

In event counter mode, the counter is decremented by an external event signal (count source) input to the AGTIOn pin. Various periods for counting events can be set with the TIOGT[1:0] bits in the AGTIOC register and AGTISR registers. In addition, the filter function for the AGTIOn pin input can be specified with bits TIPF[1:0] in the AGTIOC register. The output from the AGTOn pin can be toggled even in event counter mode.

Figure 24.8 shows the operation example in event counter mode.

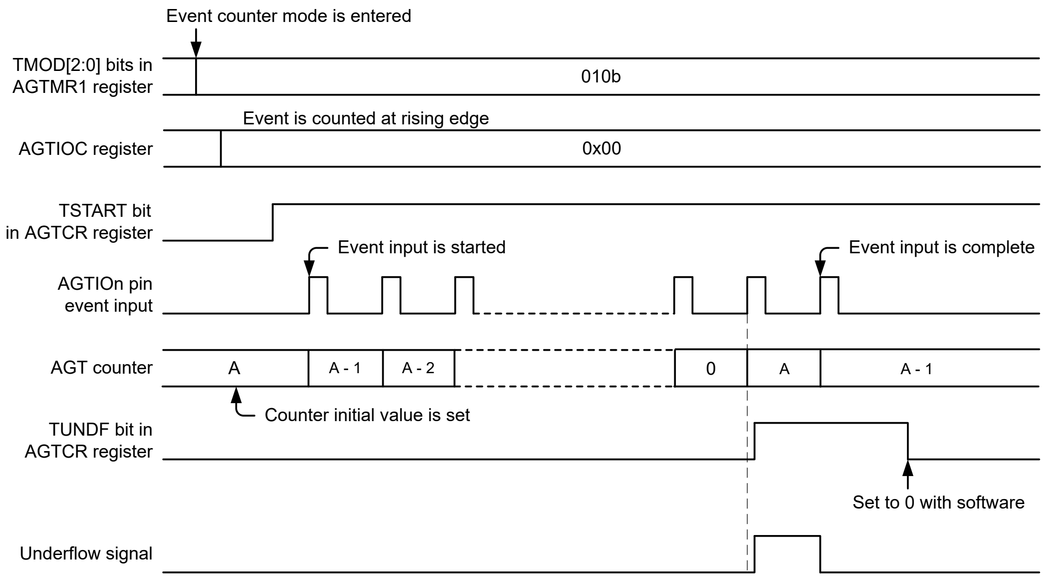
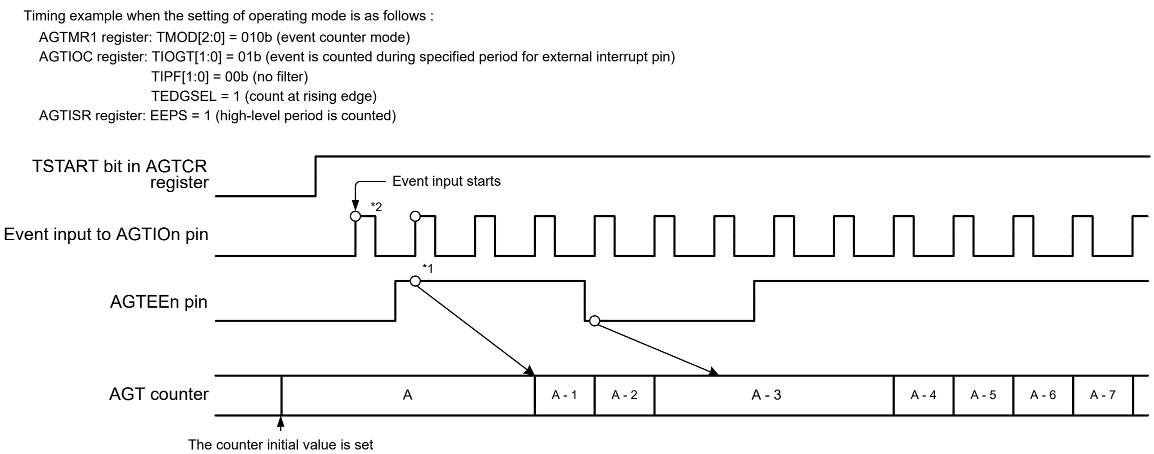


Figure 24.8 Operation example 1 in event counter mode

Figure 24.9 shows an operation example for counting during the specified period in event counter mode (TIOGT[1:0] bits in the AGTIOC register are set to 01b).



Note 1. To control synchronization, there is a delay of 2 cycles of the count source until the count operation is affected. It is also possible that the count start timing is shifted by 1 cycle because of the phase difference between the AGTEEn pin and the sampling clock.

Note 2. Count operation can be performed for 2 cycles of the count source immediately after the count starts, depending on the previous state before the count stops.
To disable the count for 2 cycles immediately after the count starts, write 1 to the TSTOP bit in the AGTCR register to initialize the internal circuit, and then complete the operation settings before starting the count operation.

Figure 24.9 Operation example 2 in event counter mode

24.3.6 Pulse Width Measurement Mode

In pulse width measurement mode, the pulse width of an external signal input to the AGTIOn pin is measured. When the level specified by the TEDGSEL bit in the AGTIOC register is input to the AGTIOn pin, the counter is decremented by the count source selected with the TCK[2:0] bits in the AGTMR1 register. When the specified level on the AGTIOn pin ends, the counter is stopped, the TEDGF flag in the AGTCR register is set to 1 (active edge received), and an interrupt request is generated. The measurement of pulse width data is performed by reading the count value while the counter is stopped. Also, when the counter underflows during measurement, the TUNDF flag in the AGTCR register is set to 1 and an interrupt request is generated.

Figure 24.10 shows the operation example in pulse width measurement mode.

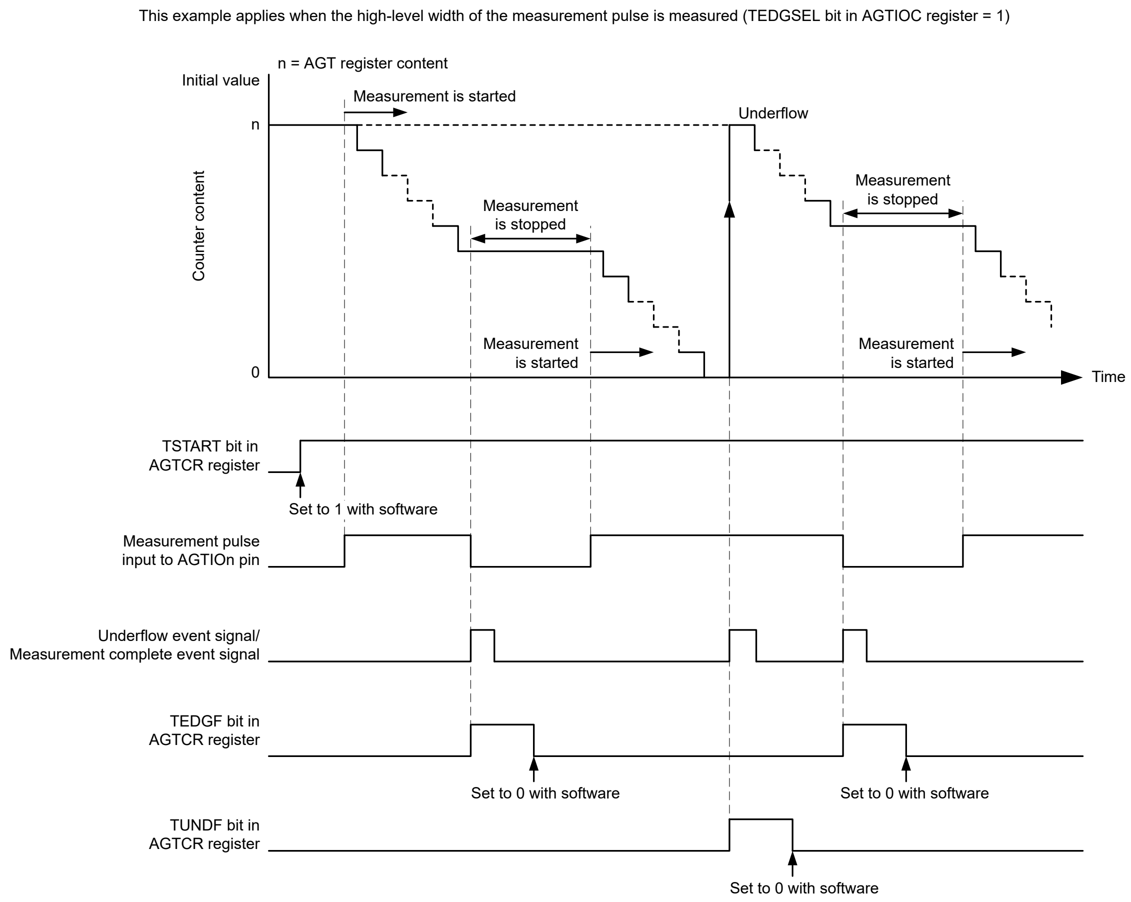


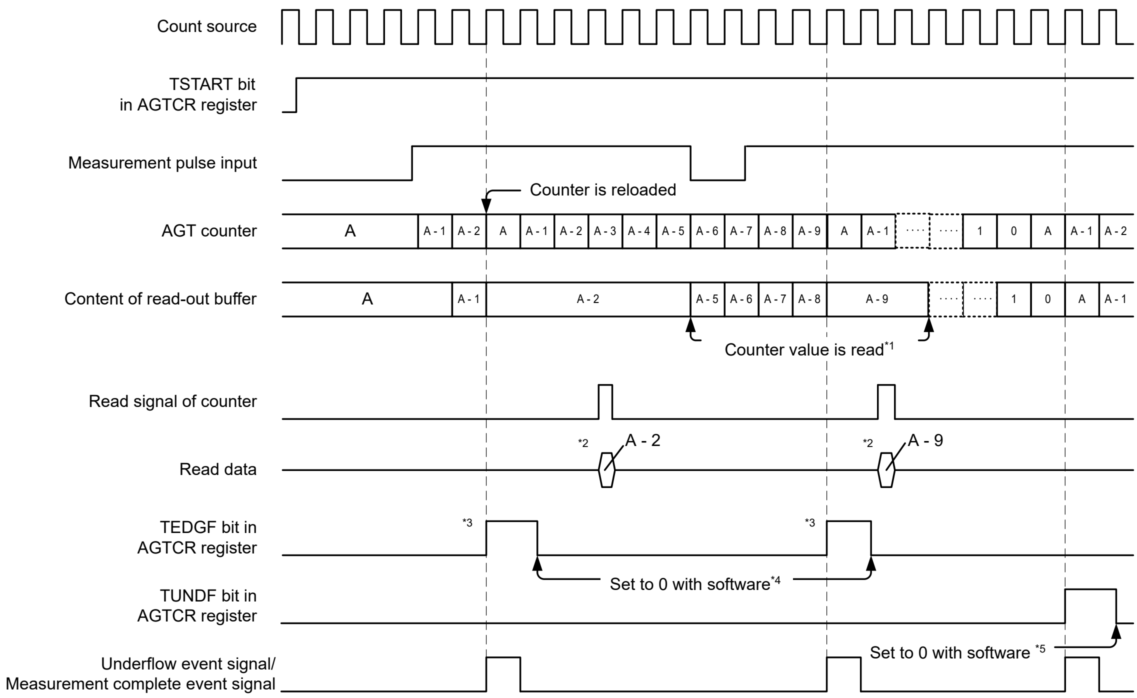
Figure 24.10 Operation example in pulse width measurement mode

24.3.7 Pulse Period Measurement Mode

In pulse period measurement mode, the pulse period of an external signal input to the AGTIOn pin is measured. The counter is decremented by the count source selected with TCK[2:0] bits in the AGTMR1 register. When a pulse with the period specified by the TEDGSEL bit in the AGTIOC register is input to the AGTIOn pin, the count value is transferred to the read-out buffer on the rising edge of the count source. The value in the reload register is loaded to the counter at the next rising edge. Simultaneously, the TEDGF flag in the AGTCR register is set to 1 (active edge received) and an interrupt request is generated. The read-out buffer (AGT register) is read at this time and the difference from the reload value (see section 24.4.6. How to Calculate Event Number, Pulse Width, and Pulse Period) is the period data of the input pulse. The period data is retained until the read-out buffer is read. When the counter underflows, the TUNDF flag in the AGTCR register is set to 1 (underflow) and an interrupt request is generated.

Figure 24.11 shows the operation example in pulse period measurement mode.

Only input pulses with a period longer than twice the period of the count source are measured. Also, the low-level and high-level widths must both be longer than the period of the count source. If a pulse period shorter than these conditions is input, the input might be ignored.



This example applies when the initial value of the AGT register is set to A, the TEDGSEL bit in the AGTIOC register is set to 0, and the period from one rising edge to the next edge of the measurement pulse is measured.

Note 1. Reading from the AGT register must be performed during the period from when the TEDGF flag is set to 1 (active edge received) until the next active edge is input. The content of the read-out buffer is retained until the AGT register is read. If it is not read before the active edge is input, the measurement result of the previous period is retained.

Note 2. When the AGT register is read in pulse period measurement mode, the content of the read-out buffer is read.

Note 3. When the active edge of the measurement pulse is input and then the set edge of an external pulse is input, the TEDGF flag in the AGTCR register is set to 1 (active edge received).

Note 4. To set to 0 with software, write 0 to the TEDGF flag in the AGTCR register with an 8-bit memory manipulation instruction.

Note 5. To set to 0 with software, write 0 to the TUNDF flag in the AGTCR register with an 8-bit memory manipulation instruction.

Figure 24.11 Operation example in pulse period measurement mode

24.3.8 Compare Match Function

The compare match function detects matches (compare match) between the content of the AGTCMA or AGTCMB register and the content of the AGT register. This function is enabled when the TCMEA or TCMEB bit in the AGTCMSR register is 1 (compare match A register or compare match B register is valid). The counter is decremented by the count source selected with the TCK[2:0] bits in the AGTMR1 register, and when the values of AGT and AGTCMA or AGTCMB match, the TCMAF/TCMBF flag in the AGTCR register is set to 1 (match), and an interrupt request is generated.

When the compare match function is enabled, the timing of the rewrite operation to the reload register and the counter differs. See section 24.3.1. Reload Register and Counter Rewrite Operation for details. In addition, the output level of the AGTOAn, AGTOBn pins is inverted by the match and by the underflow. The output level can be selected with the TOPOLA or TOPOLB bit in the AGTCMSR register.

Figure 24.12 shows the operation example in compare match function.

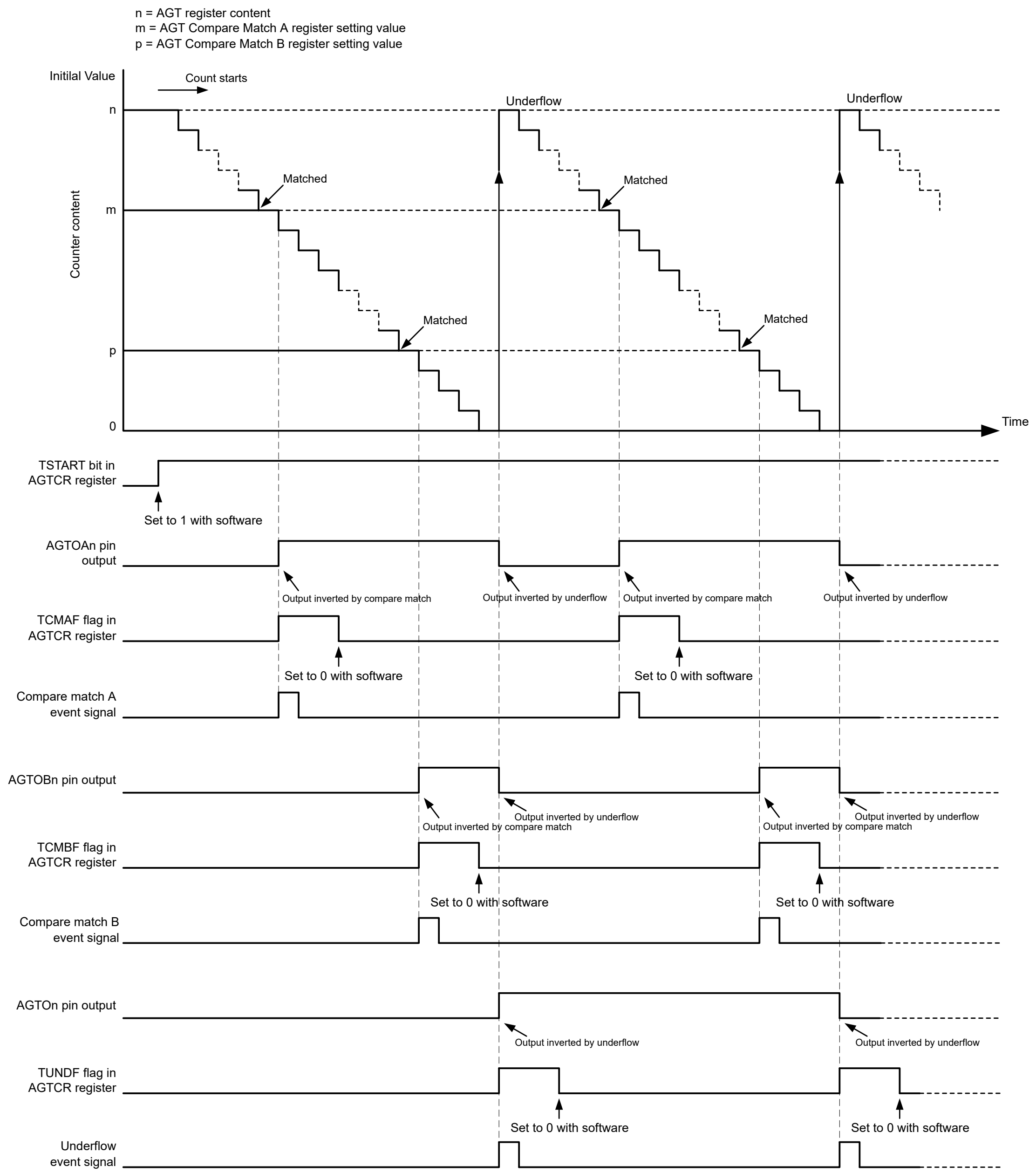


Figure 24.12 Operation example in compare match function (TOPOLA = 0, TOPOLB = 0)

24.3.9 Output Settings for Each Mode

Table 24.5 to Table 24.8 list the states of pins AGTOn, AGTIOn, AGTOAn, and AGTOBn pins in each mode.

Table 24.5 AGTOn pin setting

Operating mode	AGTIOC register		AGTOn pin output
	TOE bit	TEDGSEL bit	
All modes	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled

Table 24.6 AGTIOn pin setting

Operating mode	AGTIOC register	AGTIOn pin I/O
	TEDGSEL bit	
Timer mode	0 or 1	Input (not used)
Pulse output mode	1	Normal output
	0	Inverted output
Event counter mode	0 or 1	Input
Pulse width measurement mode		
Pulse period measurement mode		

Table 24.7 AGTOAn pin setting

Operating mode	AGTCMSR register		AGTOAn pin output
	TOEA bit	TOPOLA bit	
Timer mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Pulse output mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Event counter mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Pulse width measurement mode	0	0	Prohibited
Pulse period measurement mode			

Table 24.8 AGTOBn pin setting

Operating mode	AGTCMSR register		AGTOBn pin output
	TOEB bit	TOPOLB bit	
Timer mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Pulse output mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Event counter mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Pulse width measurement mode	0	0	Prohibited
Pulse period measurement mode			

24.3.10 Standby Mode

The AGT can operate in Software Standby mode. Set it to Software Standby mode with count operation start (TSTART = 1, TCSTF = 1).

Table 24.9 and Table 24.10 show the setting that can be used in Software Standby mode.

Table 24.9 Usable settings in Software Standby mode (AGT0)

Operating mode	AGTMR1.TCK[2:0]	Operating clock	Resurgence factor of CPU
Timer mode	100b or 110b	AGTLCLK or AGTSCLK	—
Pulse output mode	100b or 110b	AGTLCLK or AGTSCLK	—
Event counter mode*2 *3	—	AGTIO0*1	—
Pulse width measurement mode	100b or 110b	AGTLCLK or AGTSCLK	—
Pulse period measurement mode	100b or 110b	AGTLCLK or AGTSCLK	—

Note: —: invalid
Note 1. When using the AGTIO0 pin for external event input in Software Standby mode, set AGTIOSEL.TIES = 1.
Note 2. AGTEE pin is not available during Software Standby mode. External events are always enabled.
Note 3. When external event input is disabled during Software Standby mode (AGTIOSEL.TIES = 0), stop the count operation before the MCU enters Software Standby mode.
Restart the count operation after the MCU returns from Software Standby mode.

Table 24.10 Usable settings in Software Standby mode (AGT1)

Operating mode	AGTMR1.TCK[2:0]	Operating clock	Resurgence factor of CPU
Timer mode	100b or 110b or 101b*1	AGTLCLK or AGTSCLK or AGT0 underflow	• Underflow • Compare match A/B
Pulse output mode	100b or 110b or 101b*1	AGTLCLK or AGTSCLK or AGT0 underflow	• Underflow • Compare match A/B
Event counter mode*3 *4	—	AGTIO1*2	• Underflow • Compare match A/B
Pulse width measurement mode	100b or 110b or 101b*1	AGTLCLK or AGTSCLK or AGT0 underflow	• Underflow • Active edge
Pulse period measurement mode	100b or 110b or 101b*1	AGTLCLK or AGTSCLK or AGT0 underflow	• Underflow • Active edge

Note: —: invalid
Note: Release of Software Standby mode is only AGT1.
Note: Compare match A/B is resurgence factor of CPU from Software Standby mode.
Note 1. Only when AGT0 operates in Table 24.9
Note 2. When using the AGTIO1 pin for external event input in Software Standby mode, set AGTIOSEL.TIES = 1.
Note 3. AGTEE pin is not available during Software Standby mode. External events are always enabled
Note 4. When external event input is disabled during Software Standby mode (AGTIOSEL.TIES = 0), stop the count operation before the MCU enters Software Standby mode.
Restart the count operation after the MCU returns from Software Standby mode.

24.3.11 Interrupt Sources

The AGTn has three interrupt sources as listed in Table 24.11.

Table 24.11 AGT interrupt sources

Name	Interrupt source	DMAC/DTC activation
AGTn_AGTI	• When the counter underflows • When measurement of the active width of the external input pin (AGTIOn) is complete in pulse width measurement mode • When the set edge of the external input pin (AGTIOn) is input in pulse period measurement mode.	Possible
AGTn_AGTCMAI	• When the values of AGT register and AGTCMA register match	Possible
AGTn_AGTCMBI	• When the values of AGT register and AGTCMB register match	Possible

Note: Channel number (n = 0, 1)

24.3.12 Event Signal Output to ELC

The AGT uses the Event Link Controller (ELC) to perform a link operation to a specified module using the interrupt request signal as the event signal. The AGT outputs compare match A, compare match B, and underflow/measurement complete signals as event signals. For details, see section 19, Event Link Controller (ELC).

24.4 Usage Notes

24.4.1 Count Operation Start and Stop Control

- When the operating mode (see Table 24.1) is set to other than the event counter mode, or the count source is set to other than AGTn underflow event signal (TCK[2:0] = 101b):
  - After 1 (count starts) is written to the TSTART bit in the AGTCR register while the count is stopped, the TCSTF flag in the AGTCR register remains 0 (count stops) for 3 cycles of the count source. Do not access the registers associated with AGT other than the TCSTF flag until this flag is set to 1 (count in progress).
  - After 0 (count stops) is written to the TSTART bit during a count operation, the TCSTF flag remains 1 for 3 cycles of the count source. When the TCSTF flag is set to 0, the count is stopped. Do not access the registers associated with AGT other than the TCSTF flag until this flag is set to 0.
- When the operating mode (see Table 24.1) is set to event counter mode, or the count source is set to AGT1 underflow event signal (TCK[2:0] = 101b):
  - After 1 (count starts) is written to the TSTART bit in the AGTCR register while the count is stopped, the TCSTF flag in the AGTCR register remains 0 (count stops) for 2 PCLKB cycles. Do not access the registers associated with AGT other than the TCSTF flag until this flag is set to 1 (count in progress).
  - After 0 (count stops) is written to the TSTART bit during a count operation, the TCSTF flag remains 1 for 2 PCLKB cycles. When the TCSTF flag is set to 0, the count is stopped. Do not access the registers associated with AGT other than the TCSTF flag until this flag is set to 0.

24.4.2 Access to Counter Register

When the TSTART bit and TCSTF flag in the AGTCR register are both 1 (count starts), allow at least 3 cycles of the count source clock between writes when writing to the AGT register successively.

24.4.3 When Changing Mode

The registers associated with AGT operating mode (AGTMR1, AGTMR2, AGTIOC, AGTISR, and AGTCMSR) can be changed only when the count is stopped with both the TSTART bit and TCSTF flag set to 0 (count stops). Do not change these registers during count operation.

When the registers associated with AGT operating mode are changed, the values of TEDGF, TUNDF, TCMAF, and TCMBF flags are undefined. Before starting the count, write 0 to the following flags:

- TEDGF (no active edge received)
- TUNDF (no underflow)
- TCMAF (no match)
- TCMBF (no match).

24.4.4 Output Pin Setting

When using the AGTOn, AGTIOn, AGTOAn, or AGTOBn as an output pin, set up the Operation and determine the initial output values. Then set PmnPFS.PMR bit to 1.

When using the AGTIOn as an input pin in pulse width measurement mode or pulse period measurement mode, set up the Operation and start count operation. Then start to enter external events from the AGTIOn pin. Invalidate the first measurement and validate the second and later completed measurements.

24.4.5 Digital Filter

When using the digital filter, do not start the timer operation for 5 cycles of the digital filter clock after setting TIPF[1:0] bits and when the TEDGSEL bit in the AGTIOC register changes.

24.4.6 How to Calculate Event Number, Pulse Width, and Pulse Period

- In event counter mode, event number is expressed mathematically as follows:
  Event number = initial value of counter [AGT register] - counter value of active event end

- In pulse width measurement mode, pulse width is expressed mathematically as follows:
  Pulse width = counter value of stopping measurement - counter value of next stopping measurement
- In pulse period measurement mode, input pulse period is expressed mathematically as follows:
  Period of input pulse = (initial value of counter [AGT register] - reading value of the read-out buffer) + 1.

24.4.7 When Count is Forcibly Stopped by TSTOP Bit

After the counter is forcibly stopped by the TSTOP bit in the AGTCR register, do not access the following I/O registers for 1 cycle of the count source:

- AGT
- AGTCMA
- AGTCMB
- AGTCR
- AGTMR1
- AGTMR2.

24.4.8 When Selecting AGT0 Underflow as the Count Source

Operate according to the following procedures described in this section when selecting the underflow event signal as the count source.

(1) Procedure for starting operation

1. Set AGT.
2. Start the count operation of AGT1.
3. Start the count operation of AGT0.

(2) Procedure for stopping operation

1. Stop the count operation of AGT0.
2. Stop the count operation of AGT1.
3. Stop the count source clock of AGT1 (write 000b in the AGTMR1.TCK[2:0] bits).

24.4.9 Module-stop Function

AGT operation can be disabled or enabled using Module Stop Control Register D (MSTPCRD). The AGT module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Mode

24.4.10 When Switching Source Clock

When switching a clock source by changing SCKSCR.CKSEL[2:0], the clock output from the selector stops for 4 cycles of the switched clock. Therefore, when using the AGTIOn, AGTEEn, or both input as external event input, the clock source should not be switched. If switching the clock source while using the external event input, extend the input pulse width by 4 clock cycles of the switched source clock cycles.

25. Ultra-Low-Power Timer (ULPT)

25.1 Overview

This ultra-low power timer (ULPT) is a 32-bit timer that can output pulses or count external events.

This 32-bit timer consists of reload registers and a down-counter. The reload registers and the down-counter are allocated to the same address and can be accessed through the ULPTCNT register.

Table 25.1 lists the ULPT specifications.

Table 25.1 ULPT specifications

Item		Description
Operating mode	Timer mode	Count the internal clock
	Event counter mode	Count external events
Number of channels		32 bits × 2 channels (ULPTn (n = 0, 1))
Count source (operating clock)*1		The following signals can be selected: ● Internal clock: ULPTLCLK/d or ULPTSCLK/d (d = 1, 2, 4, 8, 16, 32, 64, 128) ● External event: ULPTEVIn pin.
Count operation		The following functions can be selected: ● Selection between continuous mode and one-shot mode ● Selection from count enable mode, count start mode, and count restart mode.
Interrupt/event link function (output)		● Underflow signal – When the counter underflows ● Compare match A signal – When the ULPTCNT value matches the ULPTCMA value The compare match A function is enabled. ● Compare match B signal – When the ULPTCNT value matches the ULPTCMB value The compare match B function is enabled.
Selectable functions		● Pulse output function – Inverts the output each time the counter underflows ● Compare match function – Either or both compare match A register and compare match B register are selectable ● Return from Software Standby mode can be performed with ULPTn_ULPTI, ULPTn_ULPTCMAI, or ULPTn_ULPTCMBI (n = 0, 1).*2 ● Return from Deep Software Standby mode 1 can be performed with ULPTn_ULPTI (n = 0, 1).*2
Module-stop function		Module-stop state can be set for each channel to reduce power consumption
TrustZone Filter		Security and privilege attribution can be set for each channel

Note 1. Make sure that the frequency of the peripheral module clock (PCLKB) is at least twice the frequency of the count source clock.
Note 2. For details, see section 11, Low Power Mode.

25.2 Configuration

Figure 25.1 shows the ULPT block diagram. Table 25.2 lists the I/O pins of the ULPT.

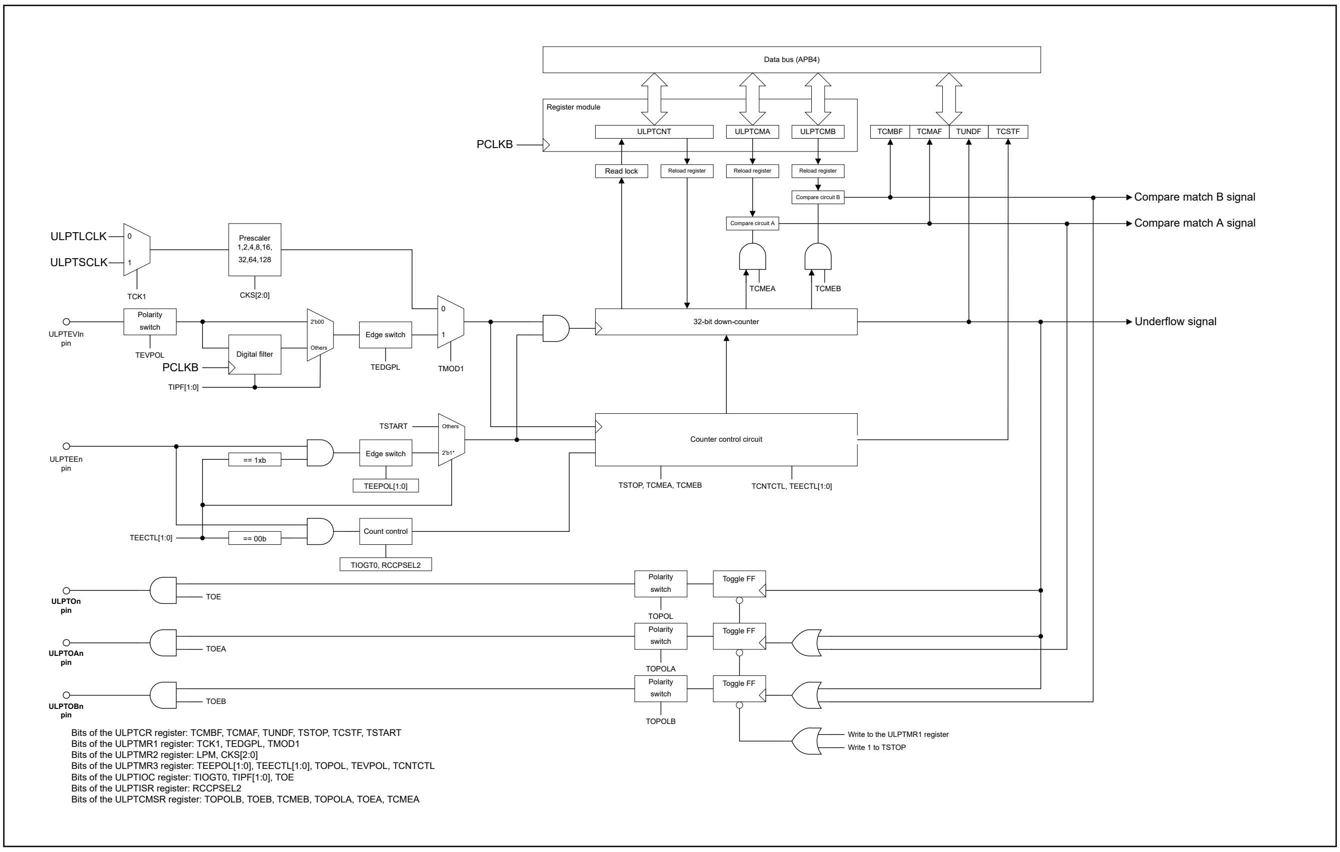


Figure 25.1 ULPT block diagram

Table 25.2 ULPT I/O pins

Pin name	I/O	Function
ULPTEEn*2	Input	External count control input*1
ULPTEVIn*2	Input	External event input
ULPTOn*2	Output	Pulse output
ULPTOAn*2	Output	Output compare match A output
ULPTOBn*2	Output	Output compare match B output

Note: Channel number: n = 0, 1

Note 1. The function of the ULPTEEn pin can be selected by the ULPTMR3.TEECTL[1:0] bits. For more details, see section 25.3.7. ULPTMR3 : ULPT Mode Register 3.

Note 2. Only pins marked "-DS" can be used in Deep Software Standby mode 1.

25.3 Register Descriptions

25.3.1 ULPTCNT : ULPT Counter Register

Base address: ULPTn = 0x4022_0000 + 0x0100 × n (n = 0, 1)
ULPTn_NS = 0x5022_0000 + 0x0100 × n (n = 0, 1)

Offset address: 0x00

Bit position:	31 … 0
Bit field:	ULPTCNT
Value after reset:	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Bit	Symbol	Function	R/W
31:0	ULPTCNT	32-bit counter and reload register*1*2 Setting range : 0x00000000 to 0xFFFFFFFF	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. When 1 is written to the ULPTCR.TSTOP bit, the counter is forcibly stopped and set to 0xFFFFFFFF. The reload register is set to 0xFFFFFFFF.
Note 2. When 0x00000000 is set in the ULPTCNT register, a request signal to the ICU, DTC, and ELC occurs only once immediately after the start of count operation (when the ULPTCR.TSTART bit is set to 1). However, the pulse output at the ULPTOn pin remains toggled.
The pulse output from the ULPTOn pin can be enabled or disabled by the ULPTIOC.TOE bit.
When the ULPTCNT register is set to a value equal to or greater than 0x00000001, a request signal is generated each time the counter underflows.

The ULPTCNT is a 32-bit register. The register value is written to the reload register and is read from the counter.

The states of the counter and reload register change according to the TSTART/TCSTF bit in the ULPTCR register and the TCMEA/TCMEB bit in the ULPTCMSR register. For details, see section 25.4.2. Rewriting the Counter and Reload Register.

The ULPTCNT register is only available for 32-bit read/write access.

25.3.2 ULPTCMA : ULPT Compare Match A Register

Base address: ULPTn = 0x4022_0000 + 0x0100 × n (n = 0, 1)
ULPTn_NS = 0x5022_0000 + 0x0100 × n (n = 0, 1)

Offset address: 0x04

Bit position:	31 ... 0
Bit field:	ULPTCMA
Value after reset:	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Bit	Symbol	Function	R/W
31:0	ULPTCMA	32-bit Compare Match A Data*1*2 Setting range : 0x00000000 to 0xFFFFFFFF	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. When 1 is written to the ULPTCR.TSTOP bit, 0xFFFFFFFF is set in the ULPTCMA register.
Note 2. Set the ULPTCMA register to 0xFFFFFFFF when compare match A is not used (ULPTCMSR.TCMEA bit = 0).

The ULPTCMA register is a readable/writable register to set a value for compare match with the counter.

The states of compare circuit A and the reload register change according to the TSTART and TCSTF bits in the ULPTCR register. For details, see section 25.4.3. Rewriting the Compare Circuits and Reload Registers for Compare Match A/B.

The ULPTCMA register is only available for 32-bit read/write access.

25.3.3 ULPTCMB : ULPT Compare Match B Register

Base address: ULPTn = 0x4022_0000 + 0x0100 × n (n = 0, 1)
ULPTn_NS = 0x5022_0000 + 0x0100 × n (n = 0, 1)

Offset address: 0x08

Bit position:	31 ... 0
Bit field:	ULPTCMB
Value after reset:	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Bit	Symbol	Function	R/W
31:0	ULPTCMB	32-bit Compare Match B Data*1*2 Setting range : 0x00000000 to 0xFFFFFFFF	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. When 1 is written to the ULPTCR.TSTOP bit, 0xFFFFFFFF is set in the ULPTCMB register.

Note 2. Set the ULPTCMB register to 0xFFFFFFFF when compare match B is not used (ULPTCMSR.TCMEB bit = 0).

The ULPTCMB register is a readable/writable register to set a value for compare match with the counter.

The states of compare circuit B and the reload register change according to the TSTART and TCSTF bits in the ULPTCR register. For details, see section 25.4.3. Rewriting the Compare Circuits and Reload Registers for Compare Match A/B.

The ULPTCMB register is only available for 32-bit read/write access.

25.3.4 ULPTCR : ULPT Control Register

Base address: ULPTn = 0x4022_0000 + 0x0100 × n (n = 0, 1)
ULPTn_NS = 0x5022_0000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TCMBF	TCMAF	TUNDF	—	—	TSTOP	TCSTF	TSTART
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TSTART	Counter Start 0: Stop the counter 1: Start the counter	R/W
1	TCSTF	Counter Status Flag*2 0: Counter stopped 1: Counter running	R
2	TSTOP	Counter Forcible Stop*1 0: Writing is invalid 1: Stop the counter forcibly	W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	TUNDF	Underflow Flag 0: No underflow occurred (counter ≠ 0x00000000) 1: Underflow occurred (counter = 0x00000000)	R/W*3
6	TCMAF	Compare Match A Flag 0: Not matched (counter ≠ ULPTCMA[31:0]) 1: Matched (counter = ULPTCMA[31:0])	R/W*3
7	TCMBF	Compare Match B Flag 0: Not matched (counter ≠ ULPTCMB[31:0]) 1: Matched (counter = ULPTCMB[31:0])	R/W*3

Note: S-TYPE-3, P-TYPE-3
Note 1. When 1 (stop the counter forcibly) is written to the TSTOP bit, the TSTOP, TSTART, and TCSTF bits are initialized. The pulse output, compare match A output and compare match B output are also initialized. This bit is read as 0.
Note 2. For precautions on using the TCSTF bit, see section 25.5.1. Start and Stop Control of the Counter.
Note 3. Only 0 can be written to clear the flag.

The ULPTCR register controls the start and stop of the counter and indicates the status of the ULPT.

The ULPTCR register is only available for 8-bit read/write access.

Set 1 to the TUNDF, TCMAF, and TCMBF bits when changing the TSTART or TSTOP bit to prevent clearing TUNDF, TCMAF, or TCMBF by mistake. At this time, setting 1 to TUNDF, TCMAF, and TCMBF bits is ignored.

TSTART bit (Counter Start)

When and how the counter is started depends on the setting of the ULPTMR3.TEECTL[1:0] bits.

The counter is stopped in the same way in any mode.

For details, see section 25.4.1. Count Operation.

[Starting the counter]

In count enable mode (ULPTMR3.TEECTL[1:0] bits = 00)

- When 1 is written to the TSTART bit.

In count start mode or count restart mode (ULPTMR3.TEECTL[1:0] bits = 10 or 11)

- When 1 is written to the TSTART bit, and then an edge at the ULPTEEn pin is input.

[Stopping the counter]

- When 0 is written to the TSTART bit
- When 1 is written to the TSTOP bit.

TCSTF bit (Counter Status Flag)

The operation of this bit depends on the settings of the TCNTCTL and TEECTL[1:0] bits of the ULPTMR3 register.

For details, see section 25.4.1. Count Operation.

[Setting conditions]

In timer mode (ULPTMR1.TMOD1 = 0) and count enable mode (ULPTMR3.TEECTL[1:0] bits = 00)

- When 1 is written to the TSTART bit (with the TCSTF bit set to 1 in synchronization with the count source).

In event counter mode (ULPTMR1.TMOD1 = 1) and count enable mode (ULPTMR3.TEECTL[1:0] bits = 00)

- When 1 is written to the TSTART bit (with the TCSTF bit set to 1 in synchronization with the count source).

In count start mode or count restart mode (ULPTMR3.TEECTL[1:0] bits = 10 or 11)

- When 1 is written to the TSTART bit, and then an edge at the ULPTEEn pin is input (with the TCSTF bit set to 1 in synchronization with the count source).

[Clearing conditions]

- When 0 is written to the TSTART bit (with the TCSTF bit set to 0 in synchronization with the count source)
- When 1 is written to the TSTOP bit
- When the counter underflows in one-shot mode (ULPTMR3.TCNTCTL = 1).
  Only under this condition, the TSTART bit does not change from 1.

TSTOP bit (Counter Forcible Stop)

When 1 is written to this bit, the counter is forcibly stopped. This bit is read as 0.

TUNDF bit (Underflow Flag)

[Setting condition]

- When the counter underflows (counter = 0x00000000).

[Clearing condition]

- When 0 is written to this bit.

TCMAF bit (Compare Match A Flag)

[Setting condition]

- When the ULPTCNT register value matches the ULPTCMA register value (counter = ULPTCMA[31:0]).

[Clearing condition]

- When 0 is written to this bit.

TCMBF bit (Compare Match B Flag)

[Setting condition]

- When the ULPTCNT register value matches the ULPTCMB register value (counter = ULPTCMB[31:0]).

[Clearing condition]

- When 0 is written to this bit.

25.3.5 ULPTMR1 : ULPT Mode Register 1

Base address: ULPTn = 0x4022_0000 + 0x0100 × n (n = 0, 1)
ULPTn_NS = 0x5022_0000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	TCK1	—	TEDGPL	—	TMOD1	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	TMOD1	Operating Mode*1 0: Timer mode 1: Event counter mode	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	TEDGPL	ULPTEVIn Edge Polarity 0: Either edge (rising) 1: Both edges This bit is valid only in event counter mode (TMOD1 = 1).	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	TCK1	Count Source*2 0: Divided clock specified by the ULPTMR2.CKS[2:0] bits (ULPTLCLK) 1: Divided clock specified by the ULPTMR2.CKS[2:0] bits (ULPTSCLK) This bit is valid only in timer mode (TMOD1 = 0).	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note: Do not rewrite this register while the counter is running. Rewrite this register only when the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0).

Note: A write access to this register initializes the output from pins ULPTOn, ULPTOAn, and ULPTOBn, For details on the output level at initialization, see section 25.3.7. ULPTMR3 : ULPT Mode Register 3 and section 25.3.8. ULPTIOC : ULPT I/O Control Register.

Note 1. Timer mode: The internal clock (ULPTLCLK or ULPTSCLK) is used as the count source. The counter is decremented by 1 on each rising edge of the count source.
Event counter mode: The external event input (ULPTEVIn pin) is used as the count source. The counter is decremented on each edge of the count source. The edge polarity can be selected by the TEDGPL bit and the ULPTMR3.TEVPOL bit.
For details, see section 25.4.1. Count Operation.

Note 2. The TCK1 bit can be changed only when the ULPTMR2.CKS[2:0] bits are 000. Otherwise, do not change the bit. Set the ULPTMR2.CKS[2:0] bits to 000, and then change the TCK1 bit after one cycle of the count source.

The ULPTMR1 register sets the operating mode of the ULPT.

The ULPTMR1 register is only available for 8-bit read/write access.

25.3.6 ULPTMR2 : ULPT Mode Register 2

Base address: ULPTn = 0x4022_0000 + 0x0100 × n (n = 0, 1)
ULPTn_NS = 0x5022_0000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LPM	—	—	—	—	CKS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CKS[2:0]	ULPTLCLK/ULPTSCLK Count Source Clock Division Ratio*1*2 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128 This bit is valid only in timer mode (ULPTMR1.TMOD1 = 0).	R/W
6:3	—	These bits are read as 0. The write value should be 0.	R/W
7	LPM	Low Power Mode 0: Normal mode 1: Low power mode	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Do not rewrite this register while the counter is running. Rewrite this register only when the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0).
Note 1. The CKS[2:0] bits can be changed only in timer mode (ULPTMR1.TMOD1 = 0). Set the CKS[2:0] bits to 000 in event counter mode (ULPTMR1.TMOD1 = 1).
Note 2. The ULPTMR1.TCK1 bit can be changed only when the CKS[2:0] bits are 000. Set the CKS[2:0] bits to 000, and then change the ULPTMR1.TCK1 bit after one cycle of the count source.

The ULPTMR2 register sets the operating mode of the ULPT.

The ULPTMR2 register is only available for 8-bit read/write access.

LPM bit (Low Power Mode)

This bit implements low power consumption by restricting register access. Set this bit to 1 for low-power mode operation.

When this bit is 1, access to the following registers is prohibited:

- ULPTCNT, ULPTCMA, ULPTCMB, and ULPTCR.

After this bit is switched from 1 to 0, the first access to an access-prohibited register is restricted as follows:

- ULPTCNT: The register must be read twice. Only the second reading of data is valid.
- ULPTCNT, ULPTCMA, ULPTCMB, ULPTCR: In write access, leave two cycles of the count source.

25.3.7 ULPTMR3 : ULPT Mode Register 3

Base address: ULPTn = 0x4022_0000 + 0x0100 × n (n = 0, 1)
ULPTn_NS = 0x5022_0000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TEEPOL[1:0]		TEECTL[1:0]		—	TOPOL	TEVPOL	TCNTCTL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TCNTCTL	Count Function Select*1*2 0: Continuous mode 1: One-shot mode	R/W
1	TEVPOL	ULPTEVIn Polarity Switch 0: External event input (ULPTEVIn pin) 1: External event input (ULPTEVIn pin) in reverse This bit is valid only in event counter mode (ULPTMR1.TMOD1 = 1).	R/W
2	TOPOL	ULPTOn Polarity Select*3 0: Start the ULPTOn output when in low level 1: Start the ULPTOn output when in high level	R/W

Bit	Symbol	Function	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
5:4	TEECTL[1:0]	ULPTEEn Function Select*1 0 0: Count enable mode 1 0: Count start mode 1 1: Count restart mode Other settings are prohibited. The count enable mode is valid only in event counter mode (ULPTMR1.TMOD1 = 1).	R/W
7:6	TEEPOL[1:0]	ULPTEEn Edge Polarity Select 0 0: Rising edge 0 1: Falling edge 1 0: Both edges Other settings are prohibited. This bit is valid only in count start mode or count restart mode. (TEECTL[1:0] = 10 or 11)	R/W

Note: S-TYPE-3, P-TYPE-3

Note: Do not rewrite this register while the counter is running. Rewrite this register only when the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0).

Note 1. For details, see section 25.4.1. Count Operation.

Note 2. Continuous mode: After starting counting, the counter continues to count down (reload value to underflow) repeatedly until the counter stops.
One-shot mode: After starting counting, the counter counts down once (reload value to underflow).

Note 3. When data is set to the ULPTMR1 register or 1 is set to the ULPTCR.TSTOP bit, the ULPTOn output is initialized.

The ULPTMR3 register sets the operating mode of the ULPT.

The ULPTMR3 register is only available for 8-bit read/write access.

TEECTL[1:0] bits (ULPTEEn Function Select)

The ULPTEEn pin functions differently depending on the mode. For details, see section 25.4.1. Count Operation.

In count enable mode, external events are counted while the ULPTEEn pin is valid. The duration can be selected by the ULPTIOC.TIOGT0 bit, and the polarity can be selected by the ULPTISR.RCCPSEL2 bit.

In count start mode, the counter starts counting at an edge trigger at the ULPTEEn pin. The operation of the ULPTEEn pin differs depending on the TSTART and TCSTF bits of the ULPTCR register.

In count restart mode, the counter starts counting at the first edge trigger at the ULPTEEn pin and resets the count at the second or following edge trigger. The operation of the ULPTEEn pin differs depending on the TSTART and TCSTF bits of the ULPTCR register.

Table 25.3 shows the operation of the ULPTEEn pin in count start or count restart mode.

Table 25.3 Operation of the ULPTEEn pin in count start mode or count restart mode

TEECTL[1:0]	ULPTCR.TSTART	ULPTCR.TCSTF	Function of the ULPTEEn pin and counter operation
10 (count start mode)	0 (stop the counter)	—	The edge input at the ULPTEEn pin is invalid. The counter remains stopped.
	1 (start the counter)	0 (counter stopped)	The counter starts counting from the reload value on the edge input at the ULPTEEn pin.
	1 (start the counter)	1 (counter running)	The edge input at the ULPTEEn pin is invalid. The counter continues to count.
11 (count restart mode)	0 (stop the counter)	—	The edge input at the ULPTEEn pin is invalid. The counter remains stopped.
	1 (start the counter)	0 (counter stopped)	The counter starts counting from the reload value on the edge input at the ULPTEEn pin.
	1 (start the counter)	1 (counter running)	The counter is reset on the edge input at the ULPTEEn pin, and it starts counting again from the reload value.

Note: The edge polarity of the ULPTEEn pin can be selected by the TEEPOL[1:0] bits.

25.3.8 ULPTIOC : ULPT I/O Control Register

Base address: ULPTn = 0x4022_0000 + 0x0100 × n (n = 0, 1)
ULPTn_NS = 0x5022_0000 + 0x0100 × n (n = 0, 1)

Offset address: 0x10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	TIOGT0	TIPF[1:0]		—	TOE	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
2	TOE	ULPTOn Output Enable 0: Disable the ULPTOn output 1: Enable the ULPTOn output	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
5:4	TIPF[1:0]	ULPTEVIn Input Filter*2 0 0: No filter 0 1: Filter sampling at PCLKB 1 0: Filter sampling at PCLKB/8 1 1: Filter sampling at PCLKB/32 This bit is valid only in event counter mode (ULPTMR1.TMOD1 = 1).	R/W
6	TIOGT0	ULPTEVIn Count Control*1 0: Always count external events 1: Count external events while the ULPTEVIn pin is valid This bit is valid only in event counter mode (ULPTMR1.TMOD1 = 1) and count enable mode (ULPTMR3.TEECTL[1:0] = 00).	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note: Do not rewrite this register while the counter is running. Rewrite this register only when the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0).

Note 1. The polarity for counting external events can be selected by the ULPTISR.RCCPSEL2 bit.

Note 2. Set the TIPF[1:0] bits to 00 before entering Software Standby mode or Deep Software Standby mode 1.

The ULPTIOC register controls the input/output of the ULPT.

The ULPTIOC register is only available for 8-bit read/write access.

TIPF[1:0] bits (ULPTEVIn Input Filter)

These bits are used to specify the sampling frequency of the ULPTEVIn input filter. Input to the ULPTEVIn pin is sampled. When a value matches three consecutive times, the value is regarded as the input value.

25.3.9 ULPTISR : ULPT Event Pin Select Register

Base address: ULPTn = 0x4022_0000 + 0x0100 × n (n = 0, 1)
ULPTn_NS = 0x5022_0000 + 0x0100 × n (n = 0, 1)

Offset address: 0x11

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	RCCPSEL2	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
2	RCCPSEL2	ULPTEEn Polarity Select 0: Count external events when low 1: Count external events when high This bit is valid only in event counter mode (ULPTMR1.TMOD1 = 1), count enable mode (ULPTMR3.TEECTL[1:0] = 00), and when ULPTIOC.TIOGT0 = 1.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Do not rewrite this register while the counter is running. Rewrite this register only when the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0).

The ULPTISR register sets the polarity of the ULPTEEn pin.

The ULPTISR register is only available for 8-bit read/write access.

25.3.10 ULPTCMSR : ULPT Compare Match Function Select Register

Base address: ULPTn = 0x4022_0000 + 0x0100 × n (n = 0, 1)
ULPTn_NS = 0x5022_0000 + 0x0100 × n (n = 0, 1)

Offset address: 0x12

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	TOPOLB	TOEB	TCMEB	—	TOPOLA	TOEA	TCMEA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TCMEA	Compare Match A Register Enable*1 0: Disable compare match A register 1: Enable compare match A register	R/W
1	TOEA	ULPTOAn Output Enable 0: Disable the ULPTOAn output 1: Enable the ULPTOAn output	R/W
2	TOPOLA	ULPTOAn Polarity Select*2 0: Start the ULPTOAn output when low 1: Start the ULPTOAn output when high	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	TCMEB	Compare Match B Register Enable*1 0: Disable compare match B register 1: Enable compare match B register	R/W
5	TOEB	ULPTOBn Output Enable 0: Disable the ULPTOBn output 1: Enable the ULPTOBn output	R/W
6	TOPOLB	ULPTOBn Polarity Select*2 0: Start the ULPTOBn output when low 1: Start the ULPTOBn output when high	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Do not rewrite this register while the counter is running. Rewrite this register only when the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0).
Note 1. When 1 (stop the counter forcibly) is set to the ULPTCR.TSTOP bit, the TCMEA, and TCMEB bits are set to 0.
Note 2. When data is set to the ULPTMR1 register or 1 is set to the ULPTCR.TSTOP bit, the ULPTOAn and ULPTOBn output is initialized.

The ULPTCMSR register controls the interrupt request signal and input/output of compare match A and compare match B.

The ULPTCMSR register is only available for 8-bit read/write access.

25.4 Operation

25.4.1 Count Operation

The counter is decremented by one on each edge of the count source. When the count value reaches 0x00000000 and the next edge of the count source is input, an interrupt request of underflow is generated. The count operation is set by the ULPTMR1 and the ULPTMR3 registers. Table 25.4 lists the count operations.

Table 25.4 Count operation

Section No.	Count operation			Count source	ULPTMR1 register		ULPTMR3 register	
	Mode 1	Mode 2	Mode 3		TMOD1	TCK1	TCNTCTL	TEECTL[1:0]
section 25.4.1.1. Timer and Continuous Modes	Timer	Continuous	—	ULPTLCLK/ ULPTSCLK	0	0 : ULPTLCLK 1 : ULPTSCLK	0	00
section 25.4.1.5. Common, Continuous, and Count Start Modes			Count start					10
section 25.4.1.6. Common, Continuous, and Count Restart Modes			Count restart					11
section 25.4.1.2. Timer and One-shot Modes		One-shot	—				1	00
section 25.4.1.7. Common, One-shot, and Count Start Modes			Count start					10
section 25.4.1.8. Common, One-shot, and Count Restart Modes			Count restart					11
section 25.4.1.3. Event Counter, Continuous, and Count Enable Modes	Event counter	Continuous	Count enable	ULPTEVIn pin	1	Don't care	0	00
section 25.4.1.5. Common, Continuous, and Count Start Modes			Count start					10
section 25.4.1.6. Common, Continuous, and Count Restart Modes			Count restart					11
section 25.4.1.4. Event Counter, One-shot, and Count Enable Modes		One-shot	Count enable				1	00
section 25.4.1.7. Common, One-shot, and Count Start Modes			Count start					10
section 25.4.1.8. Common, One-shot, and Count Restart Modes			Count restart					11

Note: For details about register settings, see section 25.3.5. ULPTMR1 : ULPT Mode Register 1 and section 25.3.7. ULPTMR3 : ULPT Mode Register 3.

[Mode 1]

The clock used as the count source is selected by the TMOD1 and TCK1 bits of the ULPTMR1 register.

Timer mode: The internal clock (ULPTLCLK or ULPTSCLK) is used as the count source.

The counter is decremented on each rising edge of the count source.

Event counter mode: The external event input (ULPTEVIn pin) is used as the count source.

The counter is decremented on each edge of the count source.

The edge polarity can be selected by the ULPTMR1.TEDGPL bit and the ULPTMR3.TEVPOL bit.

[Mode 2]

The counter behavior is selected by the TCNTCTL bit of the ULPTMR3 register.

Continuous mode: After starting counting, the counter continues to count down (reload value to underflow) repeatedly until the counter stops.

One-shot mode: After starting counting, the counter counts down once (reload value to underflow).

[Mode 3]

The function of the ULPTEEn pin is selected by the TEECTL[1:0] bits of the ULPTMR3 register.

Count enable mode: External events are counted while the ULPTEEn pin is valid.

(This bit is valid only in event counter mode.)

Note: Because the ULPTEEn pin is synchronized with the count source, external events are counted after three cycles of the count source.

Count start mode: The counter starts counting at an edge trigger of the ULPTEEn pin.

The edge polarity of the ULPTEEn pin can be selected by the ULPTMR3.TEEPOL[1:0] bits.

Note: Because the ULPTEEn pin is synchronized and held with the count source, external events are counted after four cycles of the count source.

Count restart mode: The counter starts counting at the first edge trigger of the ULPTEEn pin.

The counter resets the count at the second or following edge trigger.

The edge polarity of the ULPTEEn pin can be selected by the ULPTMR3.TEEPOL[1:0] bits.

Note: Because the ULPTEEn pin is synchronized and held with the count source, external events are counted after four cycles of the count source.

25.4.1.1 Timer and Continuous Modes

This mode operates in timer mode only.

The counter continues to count down (reload value to underflow) repeatedly in synchronization with the count source.

When 1 is set to the ULPTCR.TSTART bit, the counter is started (with the ULPTCR.TCSTF bit set to 1).

When an underflow occurs, the counter counts down again from the reload value.

When 0 is set to the ULPTCR.TSTART bit, the counter is stopped (with the ULPTCR.TCSTF bit set to 0).

Figure 25.2 shows an operation example in timer and continuous modes.

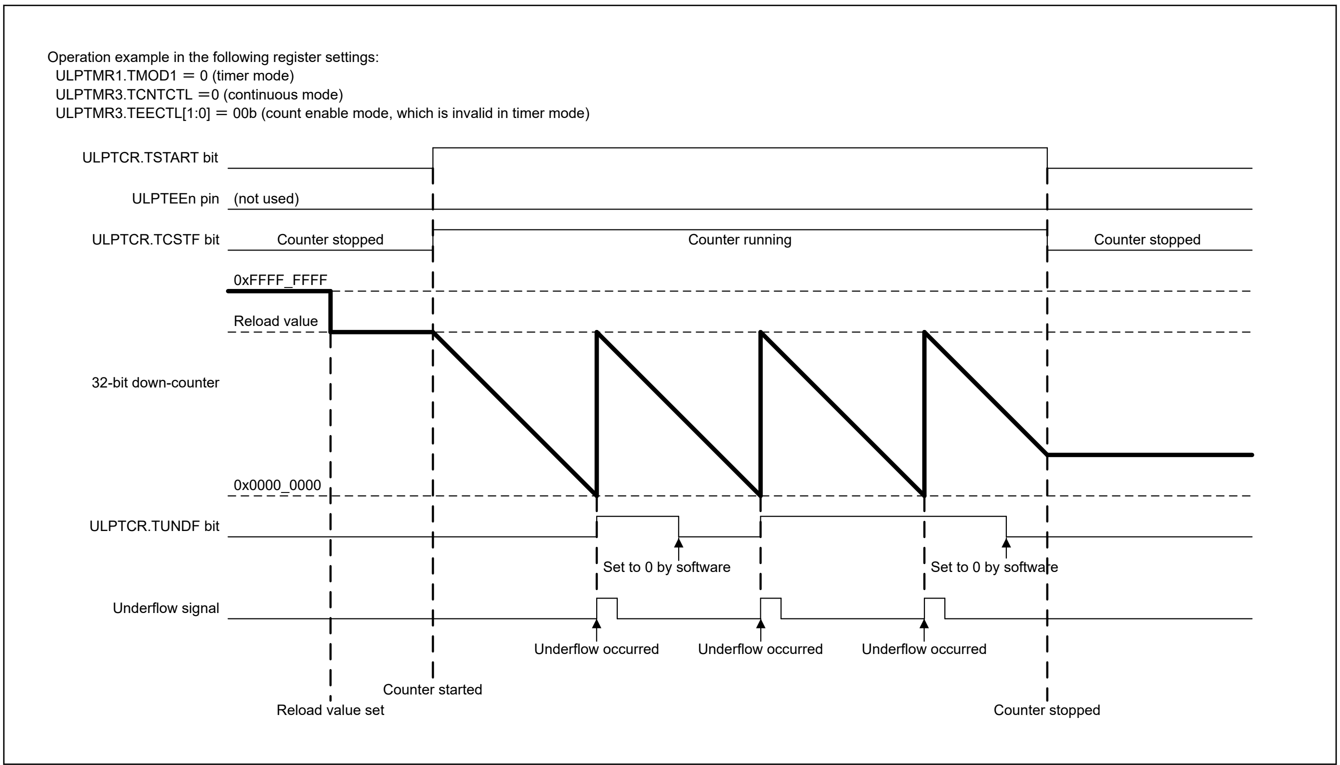


Figure 25.2 Operation example in timer and continuous modes

25.4.1.2 Timer and One-shot Modes

This mode operates in timer mode only.

The counter counts down (reload value to underflow) once in synchronization with the count source.

When 1 is set to the ULPTCR.TSTART bit, the counter is started (with the ULPTCR.TCSTF bit set to 1).

When an underflow occurs, the counter is stopped (with the ULPTCR.TSTART bit set to 1 and the ULPTCR.TCSTF bit set to 0).

To restart the counter after it is stopped, perform the following procedure:

1. Read the ULPTCR.TUNDF bit to confirm that it is set to 1 (underflow occurred).
   This is to make sure that counting down is finished.
2. Write 0 to the ULPTCR.TSTART bit to stop the counter.
3. Write 1 to the ULPTCR.TSTART bit to start the counter.

When 0 is set to the ULPTCR.TSTART bit, the counter is stopped (with the ULPTCR.TCSTF bit set to 0).

Figure 25.3 shows an operation example in timer and one-shot modes.

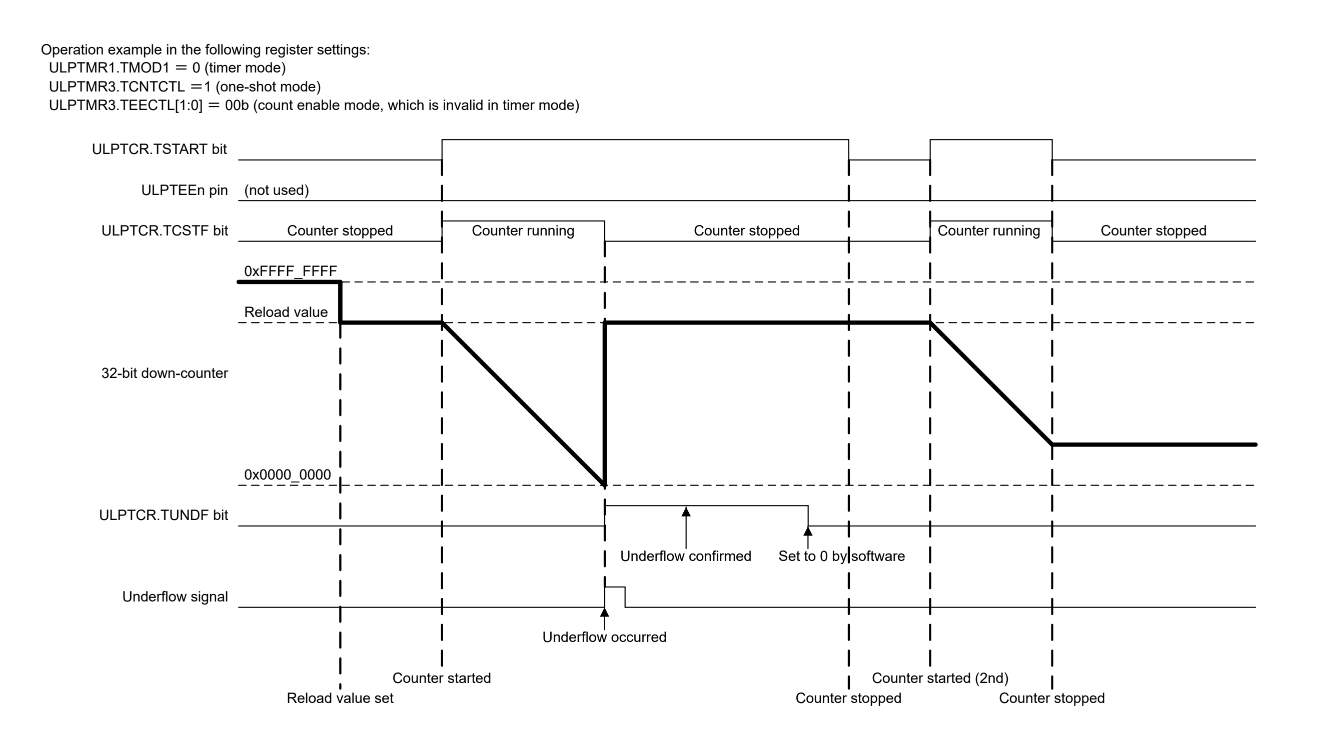


Figure 25.3 Operation example in timer and one-shot modes

25.4.1.3 Event Counter, Continuous, and Count Enable Modes

This mode operates in event counter mode only.

The counter continues to count down (reload value to underflow) repeatedly in synchronization with the count source. The duration in which the count source is counted can be specified by the ULPTIOC.TIOGT0 bit and ULPTISR.RCCPSEL2 bit.

When 1 is set to the ULPTCR.TSTART bit, the counter is started (with the ULPTCR.TCSTF bit set to 1).

When an underflow occurs, the counter counts down again from the reload value.

When 0 is set to the ULPTCR.TSTART bit, the counter is stopped (with the ULPTCR.TCSTF bit set to 0).

Figure 25.4 shows an operation example when the ULPTIOC.TIOGT0 bit is set to 0. Figure 25.5 and Figure 25.6 show operation examples when the ULPTIOC.TIOGT0 bit is set to 1.

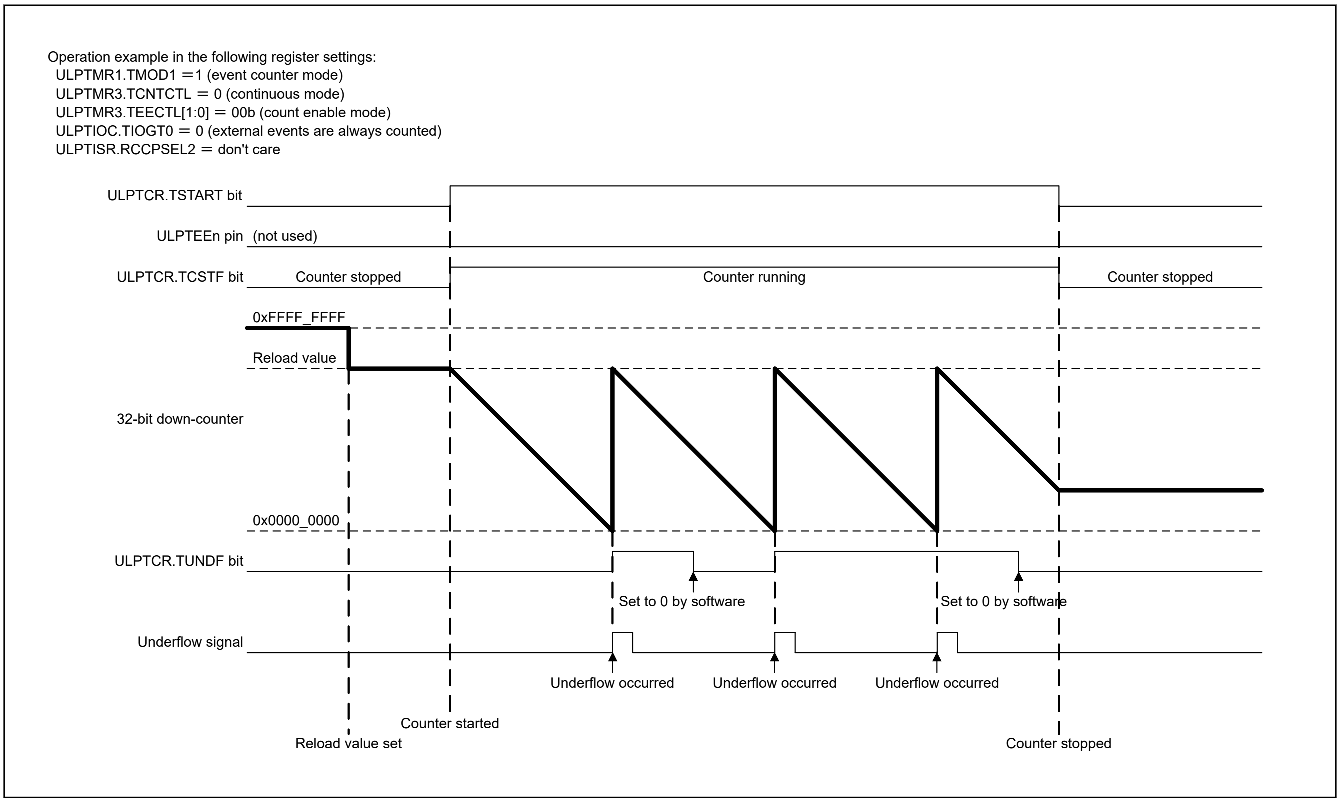


Figure 25.4 Operation example in event counter, continuous, and count enable modes (with the ULPTIOC.TIOGT0 bit set to 0)

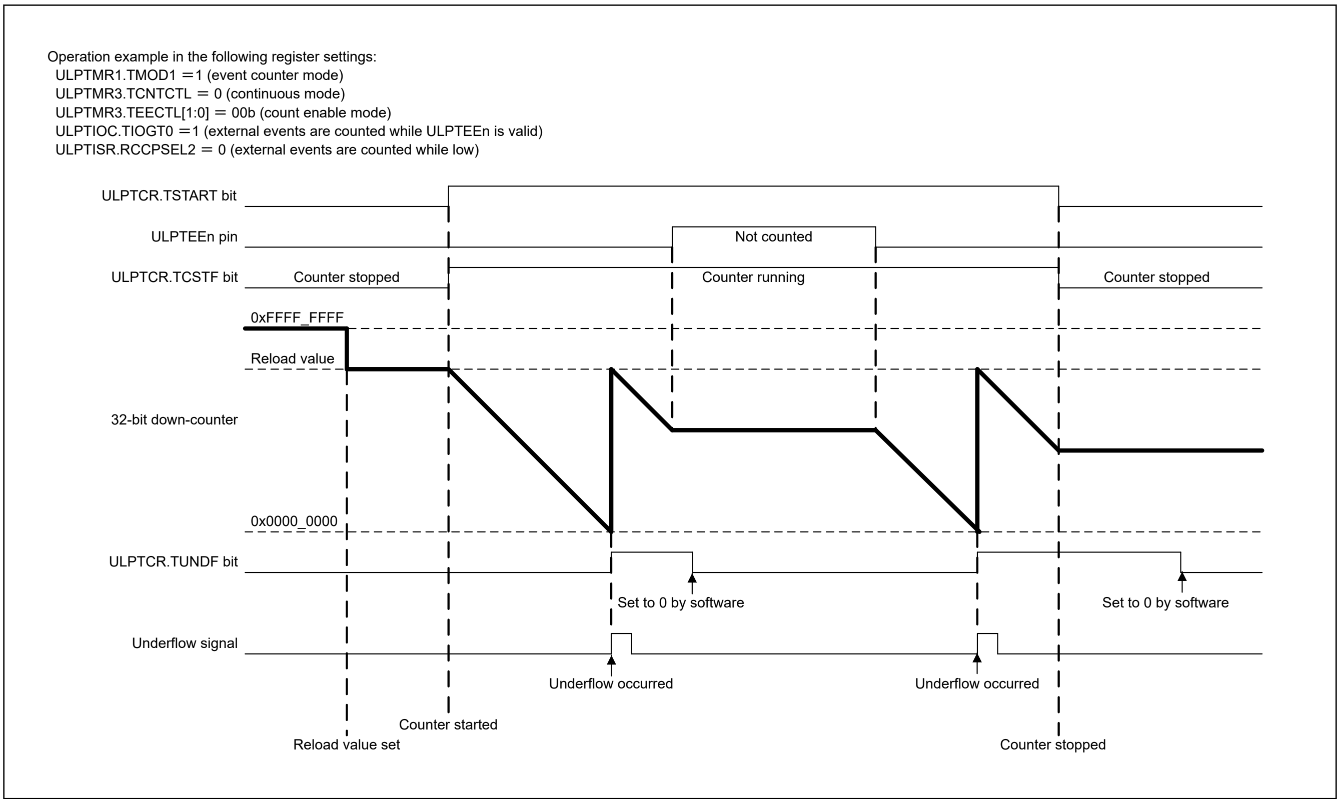
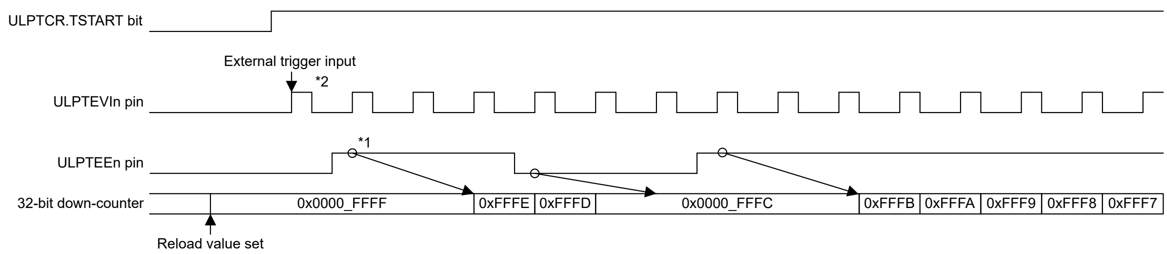


Figure 25.5 Operation example 1 in event counter, continuous, and count enable modes (with the ULPTIOC.TIOGT0 bit set to 1)

Operation example in the following register settings:
ULPTMR1.TMOD1 = 1 (event counter mode)
ULPTMR3.TCNTCTL = 0 (continuous mode)
ULPTMR3.TEECTL[1:0] = 00b (count enable mode)
ULPTIOC.TIOGT0 = 1 (external events are counted while ULPTEEn is valid)
ULPTISR.RCCPSEL2 = 1 (external events are counted while high)
ULPTMR3.TEVPOL = 0 (ULPTEVIn is inverted)



Note 1. When synchronization is controlled, there is a delay by two cycles of count source until external events are counted. In addition, the count start timing may be shifted by one cycle due to the phase difference between the ULPTEEn pin and the sampling clock.

Note 2. Depending on the state before the counter is stopped, counting may be performed during two cycles of the count source immediately after the counter is started.
To disable the counting operation during two cycles immediately after the counter is started, set 1 to the ULPTCR.TSTOP bit to initialize the internal circuits, and then finish the operation settings before starting the counter.

Figure 25.6 Operation example 2 in event counter, continuous, and count enable modes (with the ULPTIOC.TIOGT0 bit set to 1)

25.4.1.4 Event Counter, One-shot, and Count Enable Modes

This mode operates in event counter mode only.

The counter counts down (reload value to underflow) once in synchronization with the count source. The duration in which the count source is counted can be specified by the ULPTIOC.TIOGT0 bit and ULPTISR.RCCPSEL2 bit.

When 1 is set to the ULPTCR.TSTART bit, the counter is started (with the ULPTCR.TCSTF bit set to 1).

When an underflow occurs, the counter is stopped (with the ULPTCR.TSTART bit set to 1 and the ULPTCR.TCSTF bit set to 0).

To restart the counter after the counter is stopped, perform the following procedure:

1. Read the ULPTCR.TUNDF bit to confirm that it is set to 1 (underflow occurred).
   This is to make sure that counting down is finished.
2. Write 0 to the ULPTCR.TSTART bit to stop the counter.
3. Write 1 to the ULPTCR.TSTART bit to start the counter.

When 0 is set to the ULPTCR.TSTART bit, the counter is stopped (with the ULPTCR.TCSTF bit set to 0).

Figure 25.7 shows an operation example when the ULPTIOC.TIOGT0 bit is set to 00, and Figure 25.8 shows an operation example when the ULPTIOC.TIOGT0 bit is set to 01.

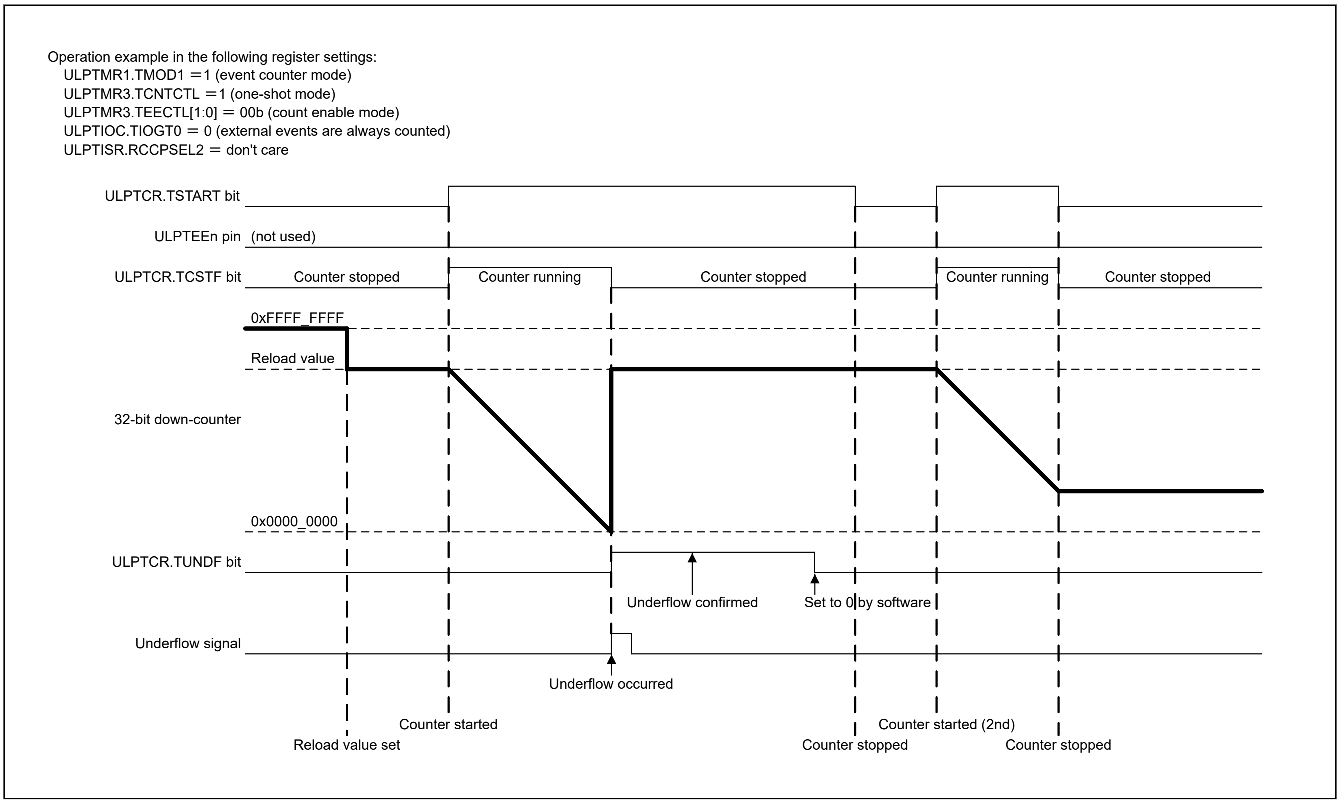


Figure 25.7 Operation example in event counter, one-shot, and count enable modes (with the ULPTIOC.TIOGT0 bit set to 0)

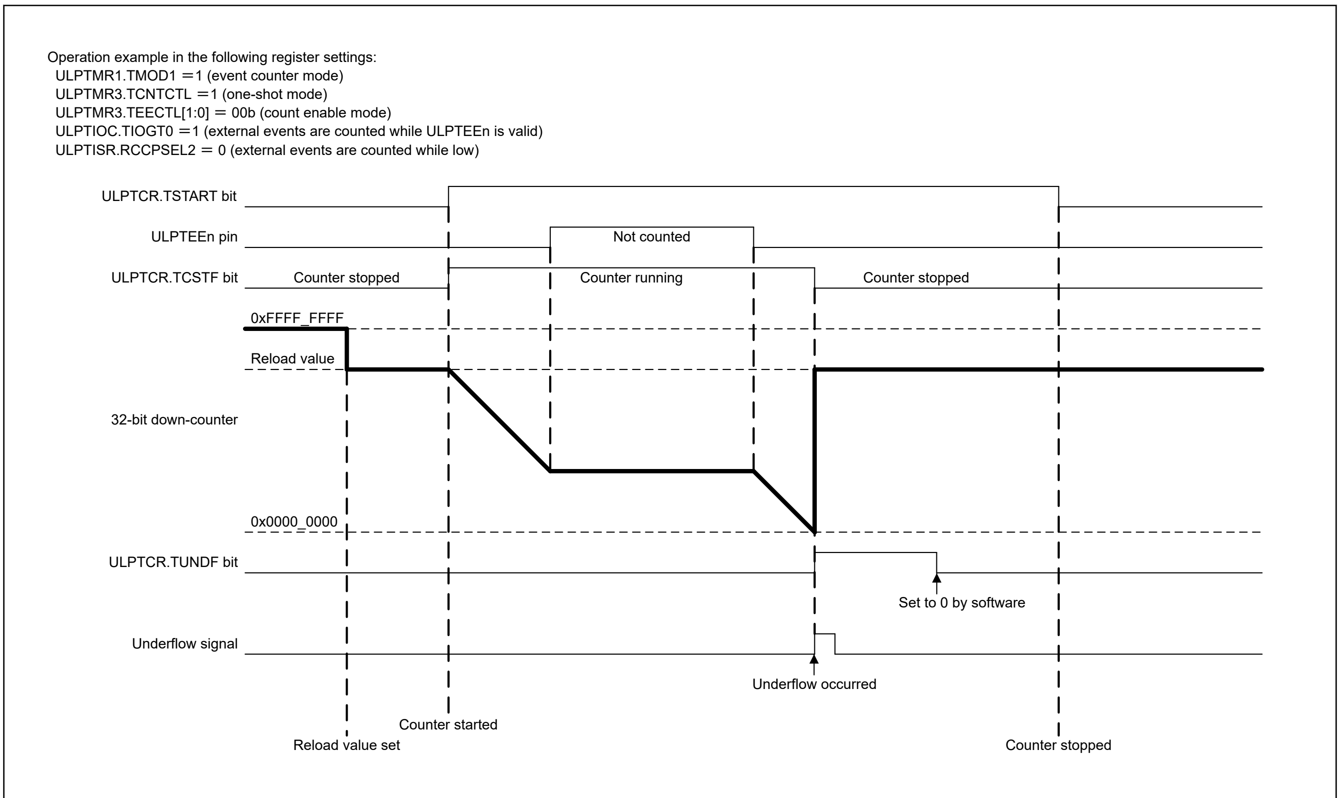


Figure 25.8 Operation example in event counter, one-shot, and count enable modes (with the ULPTIOC.TIOGT0 bit set to 1)

25.4.1.5 Common, Continuous, and Count Start Modes

This mode operates in both timer mode and event counter mode.

The counter continues to count down (reload value to underflow) repeatedly in synchronization with the count source.

After 1 is set to the ULPTCR.TSTART bit, the counter is started (with the ULPTCR.TCSTF bit set to 1) by an edge trigger at the ULPTEEn pin.

When an underflow occurs, the counter counts down again from the reload value.

When 0 is set to the ULPTCR.TSTART bit, the counter is stopped (with the ULPTCR.TCSTF bit set to 0).

Figure 25.9 shows an operation example in common, continuous, and count start modes.

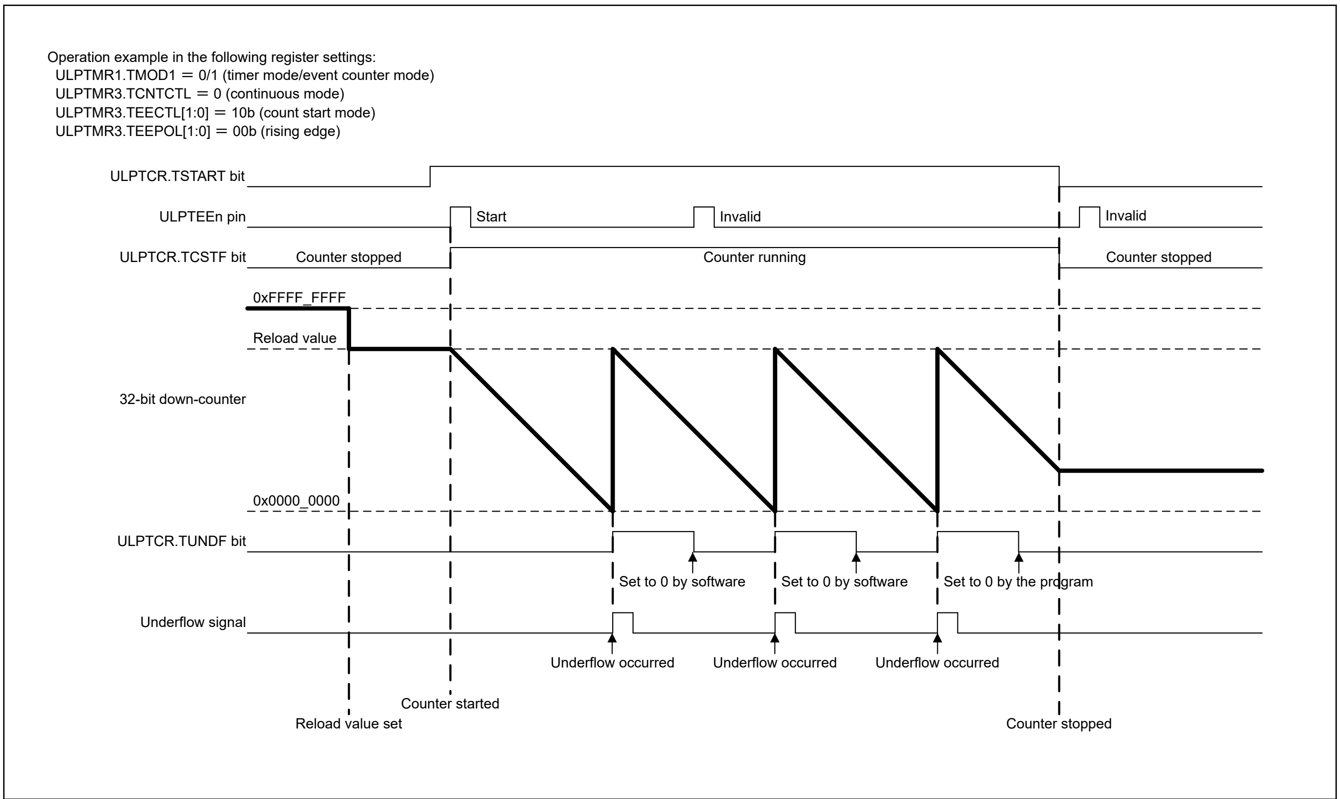


Figure 25.9 Operation example in common, continuous, and count start modes

25.4.1.6 Common, Continuous, and Count Restart Modes

This mode operates in both timer mode and event counter mode.

The counter continues to count down (reload value to underflow) repeatedly in synchronization with the count source.

After 1 is set to the ULPTCR.TSTART bit, the counter is started (with the ULPTCR.TCSTF bit set to 1) by an edge trigger at the ULPTEEn pin.

On an edge trigger input at the ULPTEEn pin when the counter is running (with the ULPTCR.TCSTF bit set to 1), the counter is reset and it starts counting again from the reload value.

If no edge trigger is input from the ULPTEEn pin and an overflow occurs, the counter starts counting again from the reload value.

When 0 is set to the ULPTCR.TSTART bit, the counter is stopped (with the ULPTCR.TCSTF bit set to 0).

Figure 25.10 shows an operation example in common, continuous, and count restart modes.

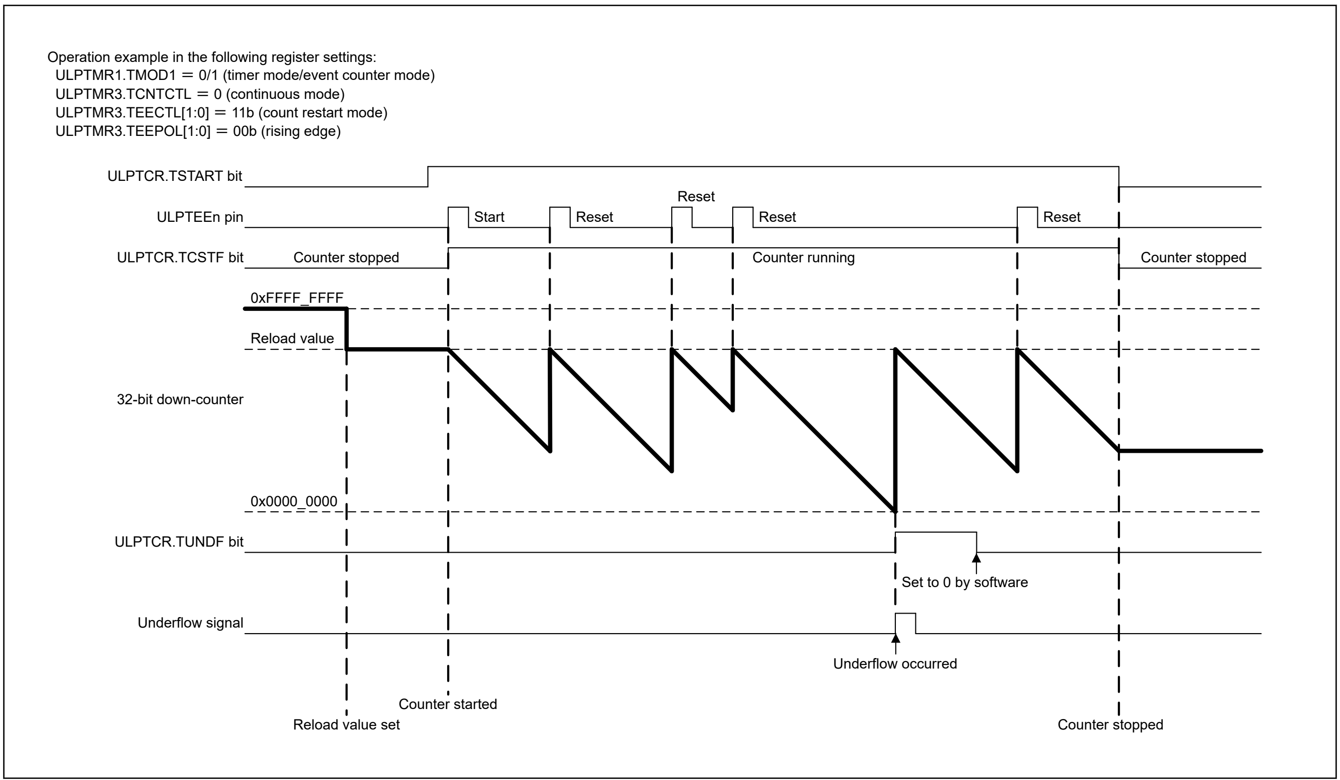


Figure 25.10 Operation example in common, continuous, and count restart modes

25.4.1.7 Common, One-shot, and Count Start Modes

This mode operates in both timer mode and event counter mode.

The counter counts down (reload value to underflow) once in synchronization with the count source.

After 1 is set to the ULPTCR.TSTART bit, the counter is started (with the ULPTCR.TCSTF bit set to 1) by an edge trigger at the ULPTEEn pin.

When an underflow occurs, the counter is stopped (with the ULPTCR.TSTART bit set to 1 and the ULPTCR.TCSTF bit set to 0).

On an edge trigger input at the ULPTEEn pin when the counter is running (with ULPTCR.TSTART bit set to 1 and the ULPTCR.TCSTF bit set to 0), the counter starts counting down again from the reload value.

When 0 is set to the ULPTCR.TSTART bit, the counter is stopped (with the ULPTCR.TCSTF bit set to 0).

Figure 25.11 shows an operation example in common, one-shot, and count start modes.

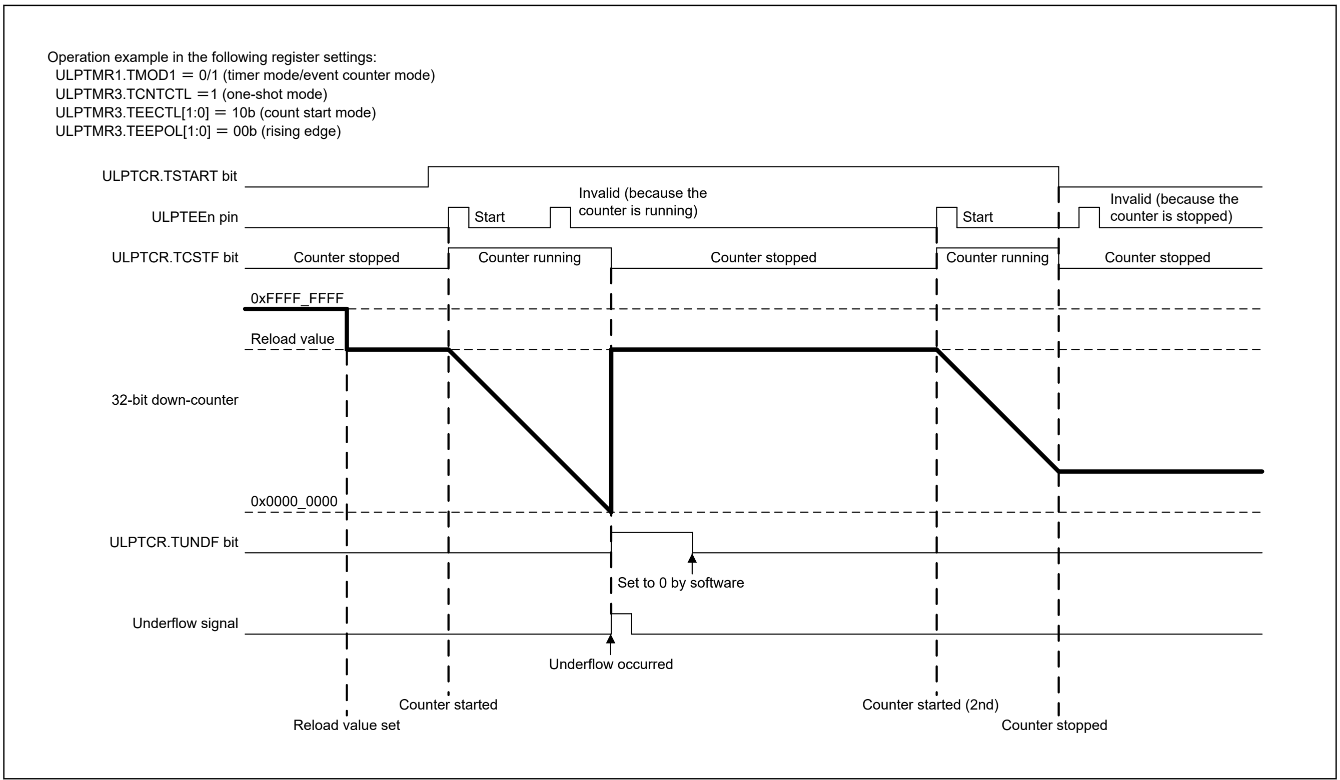


Figure 25.11 Operation example in common, one-shot, and count start modes

25.4.1.8 Common, One-shot, and Count Restart Modes

This mode operates in both timer mode and event counter mode.

The counter counts down (reload value to underflow) once in synchronization with the count source.

After 1 is set to the ULPTCR.TSTART bit, the counter is started (with the ULPTCR.TCSTF bit set to 1) by an edge trigger of the ULPTEEn pin.

On an edge trigger input at the ULPTEEn pin when the counter is running (with the ULPTCR.TCSTF bit set to 1), the counter is reset and it starts counting again from the reload value.

If no edge trigger is input at the ULPTEEn pin and an underflow occurs, the counter is stopped (with the ULPTCR.TSTART bit set to 1 and the ULPTCR.TCSTF bit set to 0).

On an edge trigger input at the ULPTEEn pin when the counter is running (with the ULPTCR.TSTART bit set to 1 and the ULPTCR.TCSTF bit set to 0), the counter starts counting down again from the reload value.

When 0 is set to the ULPTCR.TSTART bit, the counter is stopped (with the ULPTCR.TCSTF bit set to 0).

Figure 25.12 shows an operation example in common, one-shot, and count restart modes.

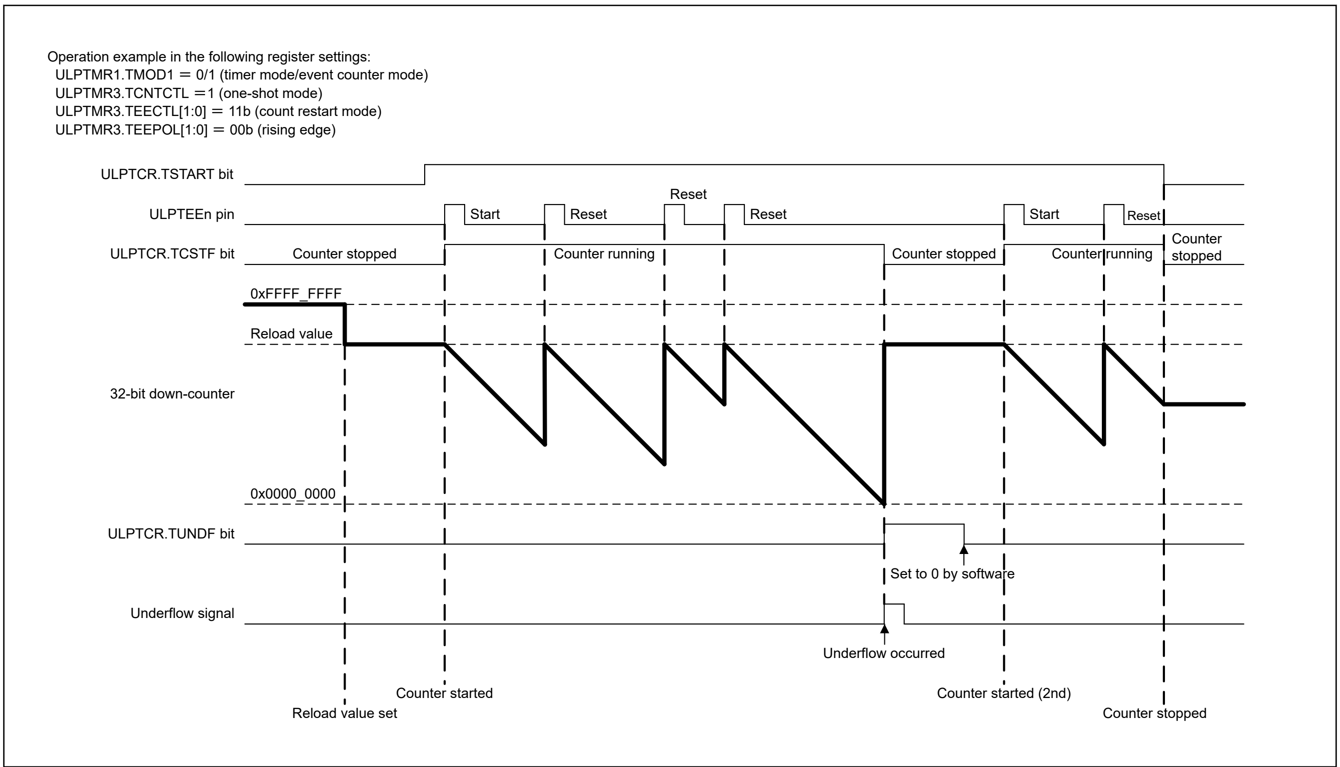


Figure 25.12 Operation example in common, one-shot, and count restart modes

25.4.2 Rewriting the Counter and Reload Register

Regardless of the operating mode, the counter and reload register rewrite timing changes depending on the following bits:

- TSTART and TCSTF bits of the ULPTCR register
- TCMEA and TCMEB bits of the ULPTCMSR register
- TEECTL[1:0] bits of the ULPTMR3 register.

Condition A: When the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0)

A reload value is written directly to the counter and the reload register.

Condition B: When the counter is running (with the ULPTCR.TCSTF bit set to 1) and the compare match A/B registers are disabled (with both the TCMEA and TCMEB bits of the ULPTCR register set to 0)

After a reload value is written to the ULPTCNT register, the reload value is first set to the reload register, then to the counter in synchronization with the count source.

Condition C: When the counter is running (with the ULPTCR.TCSTF bit set to 1) and the compare match A/B registers are enabled (with either the TCMEA bit or the TCMEB bit of the ULPTCR register set to 1)

After a reload value is written to the ULPTCNT register, the reload value is written to the reload register in synchronization with the count source. Then, the reload value is set to the counter in synchronization with an underflow.

Condition D: When the counter is running (with ULPTCR.TCSTF bit set to 1) and when the count restart mode is enabled (with the ULPTMR3.TEECTL[1:0] bits set to 11)

After an edge trigger is input from the ULPTEEn pin, the reload value set in the reload register is set to the counter in synchronization with the count source. In this condition, the reload register is not rewritten.

Figure 25.13 to Figure 25.15 show the counter and reload register rewrite timing.

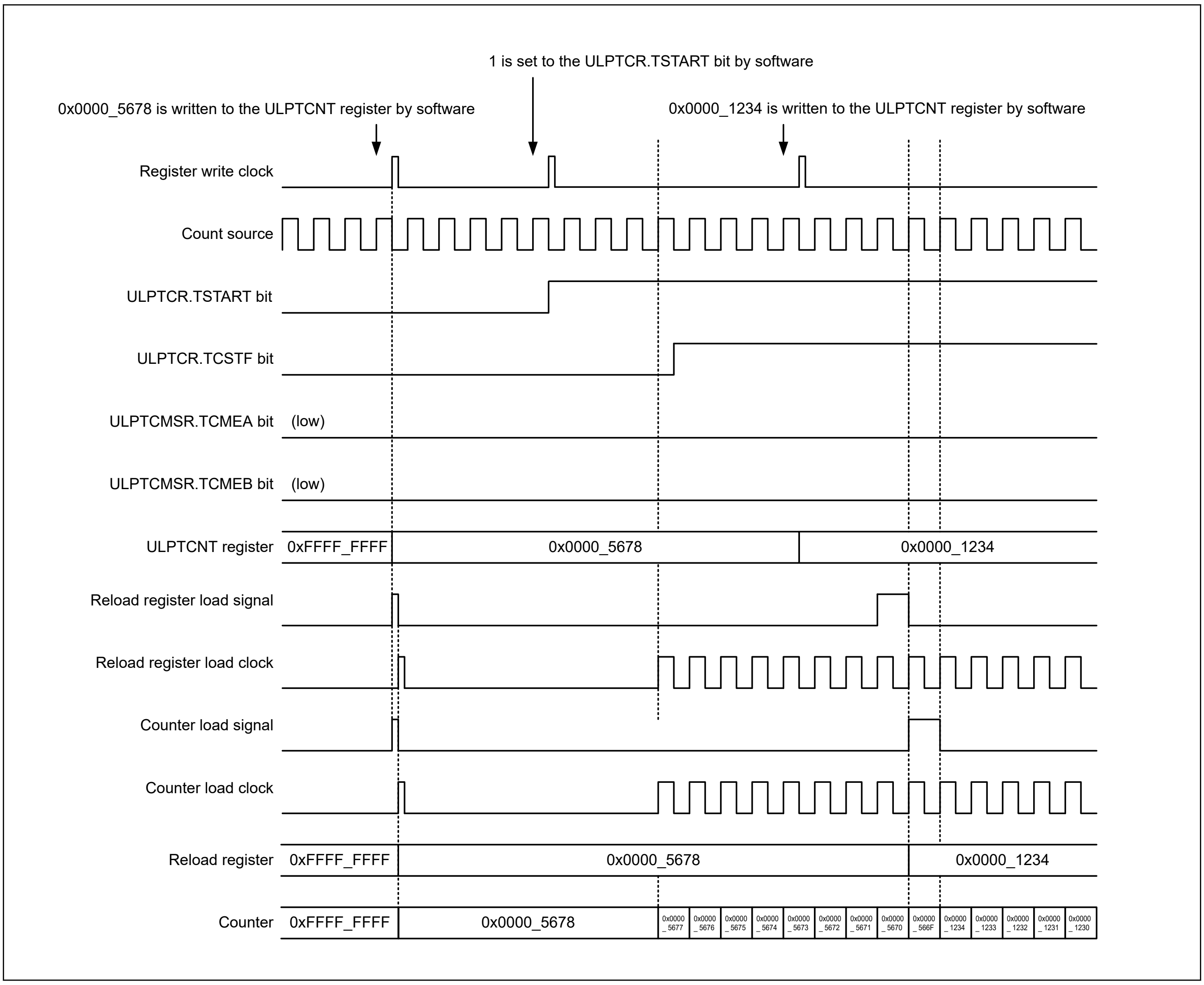


Figure 25.13 Counter and reload rewrite timing (under condition A or B)

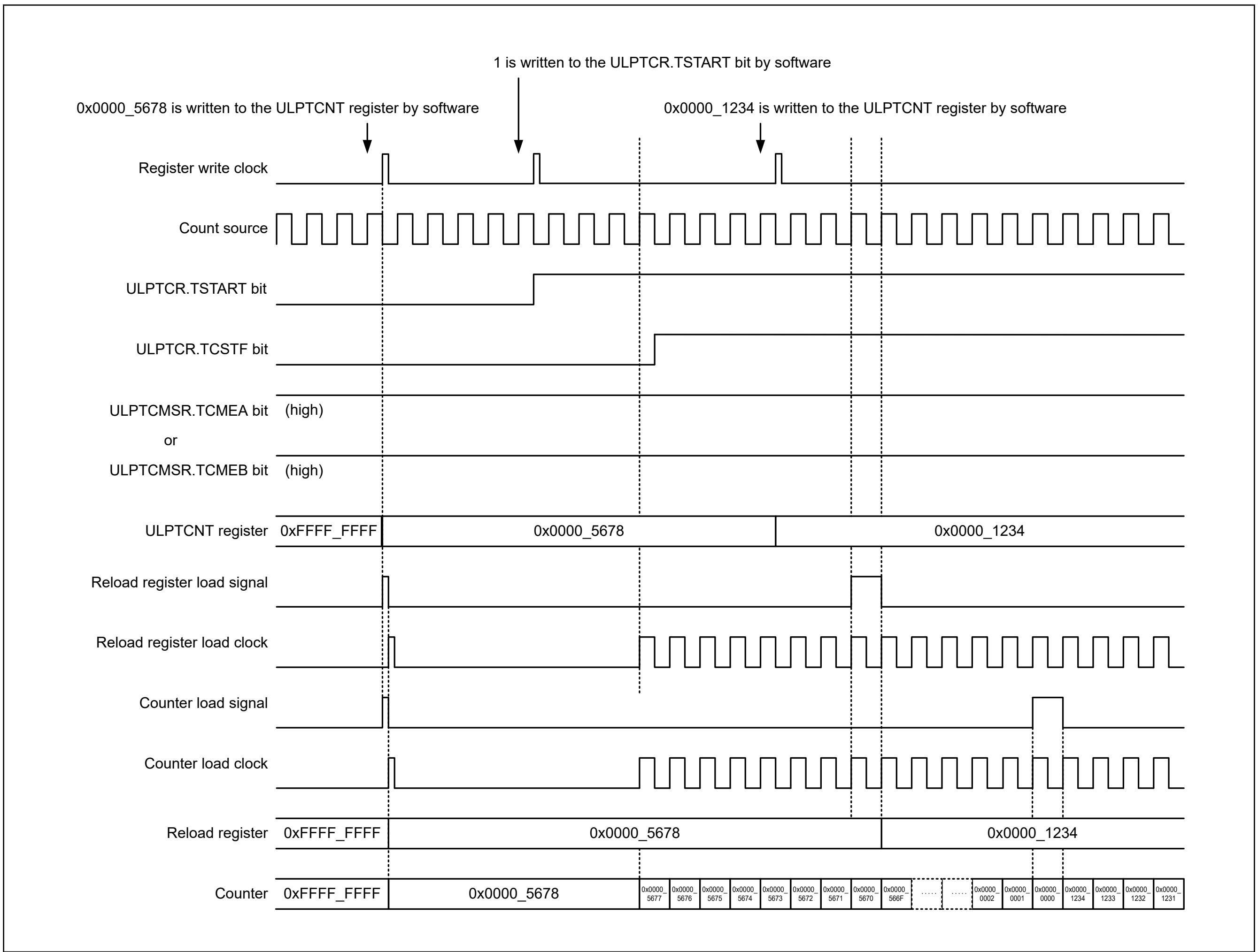


Figure 25.14 Counter and reload rewrite timing (under condition A or C)

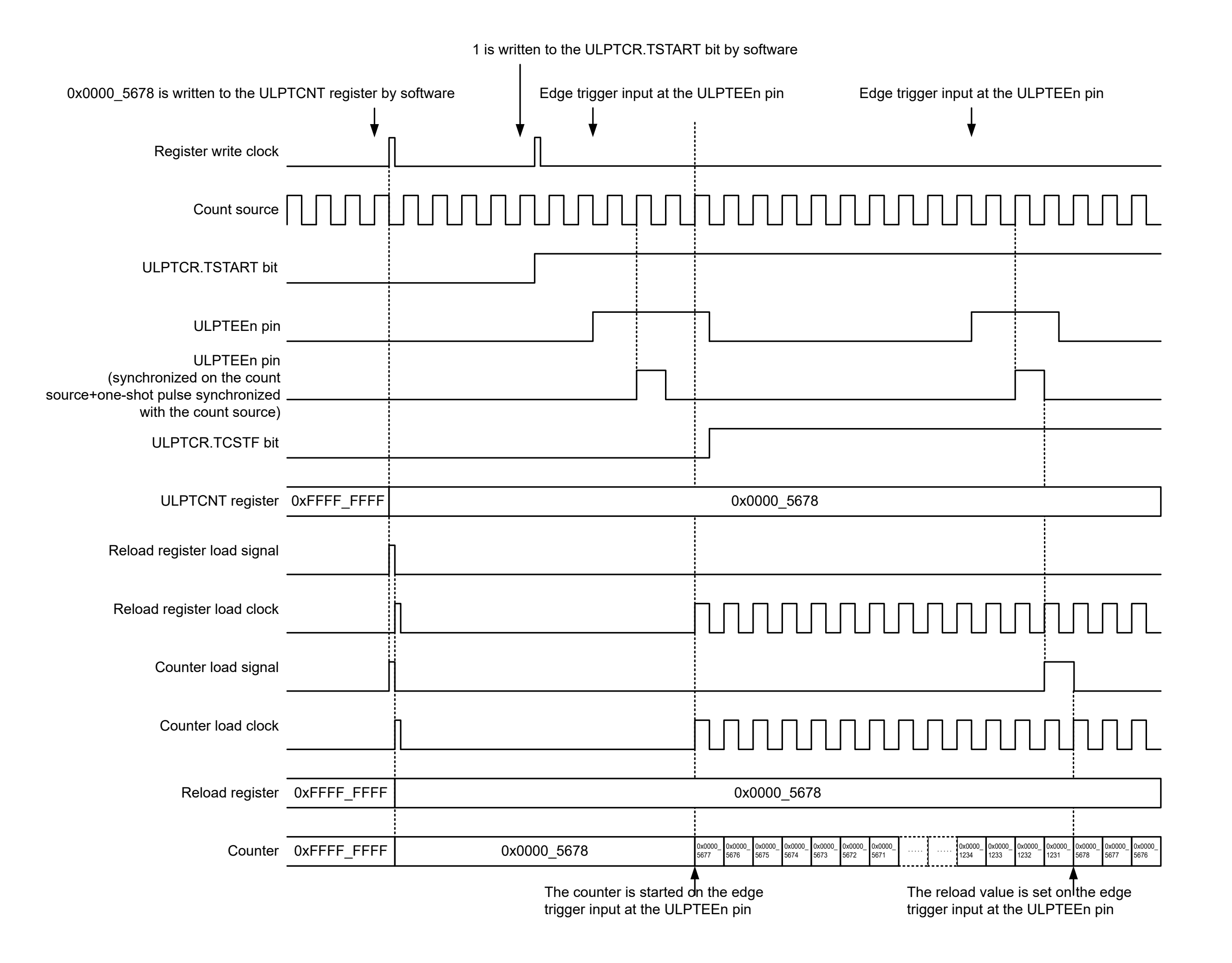


Figure 25.15 Counter and reload rewrite timing (under condition D)

25.4.3 Rewriting the Compare Circuits and Reload Registers for Compare Match A/B

Regardless of the operating mode, the rewrite timing for the compare circuits and reload registers for compare match A/B changes depending on the TSTART and TCSTF bits of the ULPTCR register.

Condition A: When the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0)

Compare match A/B values are written directly to the compare circuits and reload registers.

Condition B: When the counter is running (with the ULPTCR.TCSTF bit set to 1)

After compare match A/B values are written to the ULPTCMA and ULPTCMB registers, the compare match A/B values are written to the reload registers in synchronization with the count source. Then, the compare match A/B values are written to the compare circuits in synchronization with an underflow.

Figure 25.16 shows the rewrite timing for the compare circuit and reload register for compare match A.

The timing for compare match B is similar to that for compare match A.

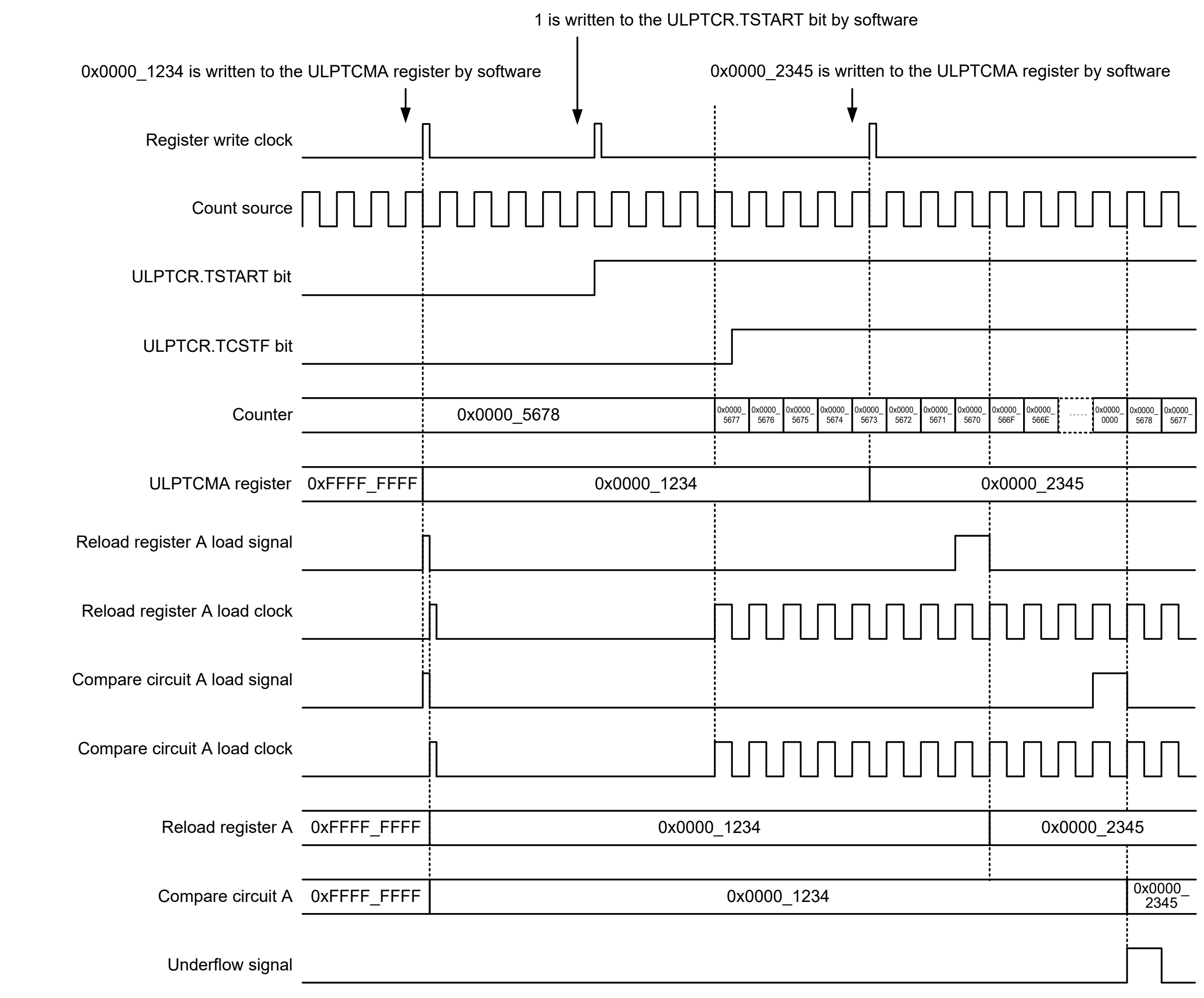


Figure 25.16 Rewrite timing for the compare circuit and reload register for compare match A

25.4.4 Pulse Output

Regardless of the operating mode, the ULPTOn pin can output pulses. The output level is inverted each time the counter underflows. The pulse output from the ULPTOn pin can be stopped by the ULPTIOC.TOE bit. The initial value of the output level can be selected by the ULPTMR3.TOPOL bit.

Figure 25.17 shows an operation example of a pulse output.

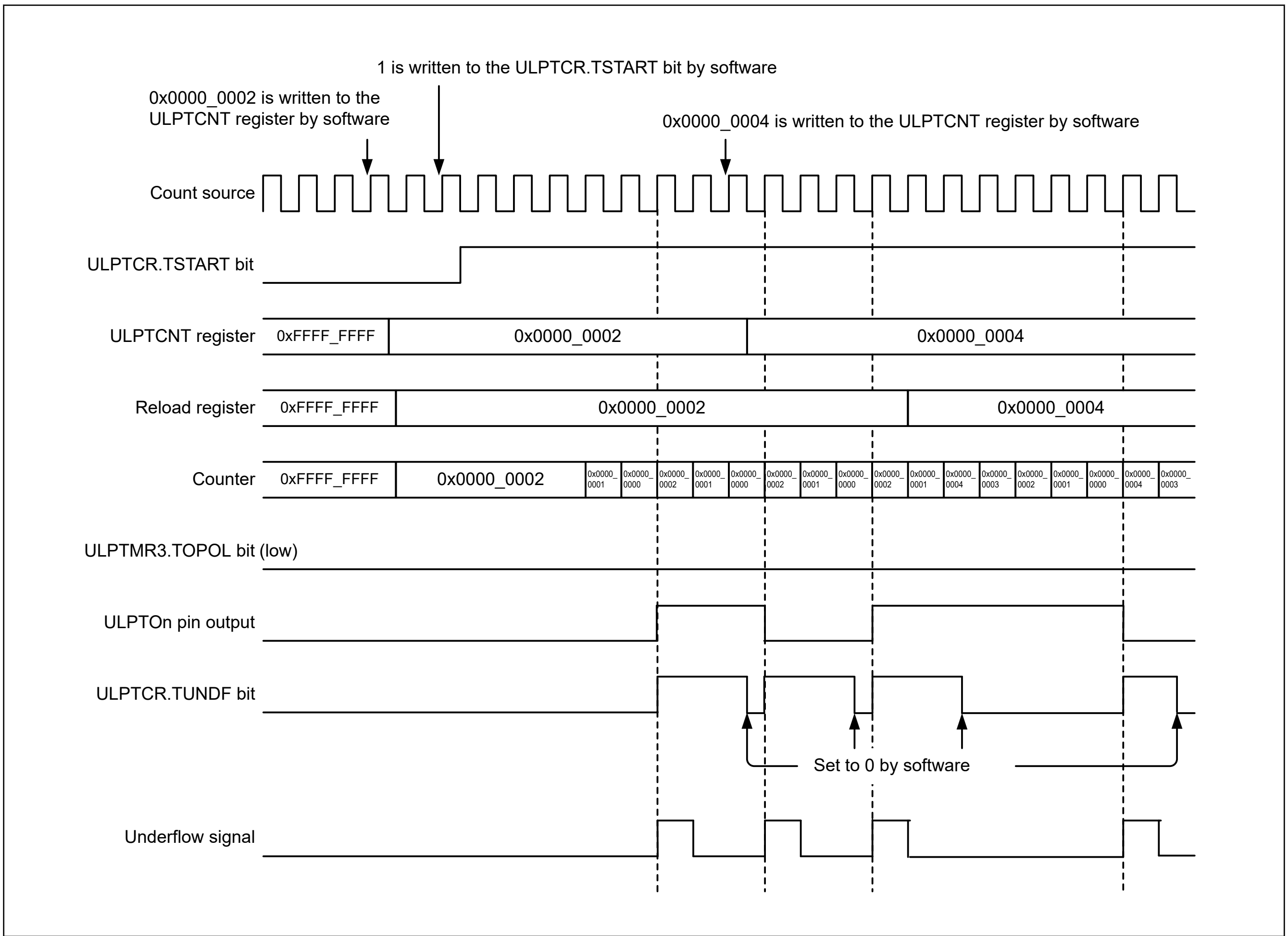


Figure 25.17 Operation example of pulse output

25.4.5 Compare Match Function

This function detects a match (compare match) between the ULPTCMA/ULPTCMB register contents and the ULPTCNT register contents. This function is enabled when the TCMEA or TCMEB bit of the ULPTCMSR register is set to 1 (the compare match A/B registers are enabled).

The counter is decremented in synchronization with the count source selected by the TMOD1/TCK1 bits of the ULPTMR1 register. When the ULPTCNT register value matches the ULPTCMA or ULPTCMB register value, the TCMAF or TCMBF bit of the ULPTCR register is set to 1 and an interrupt request is generated.

The counter and reload register rewrite timing differs depending on whether the compare match function is enabled. For details, see section 25.4.2. Rewriting the Counter and Reload Register.

The output level of the ULPTOAn/ULPTOBn pin is inverted on a match or underflow. The output level can be selected by the ULPTOAn/ULPTOBn bit of the ULPTCMSR register.

Figure 25.18 shows an operation example of the compare match function.

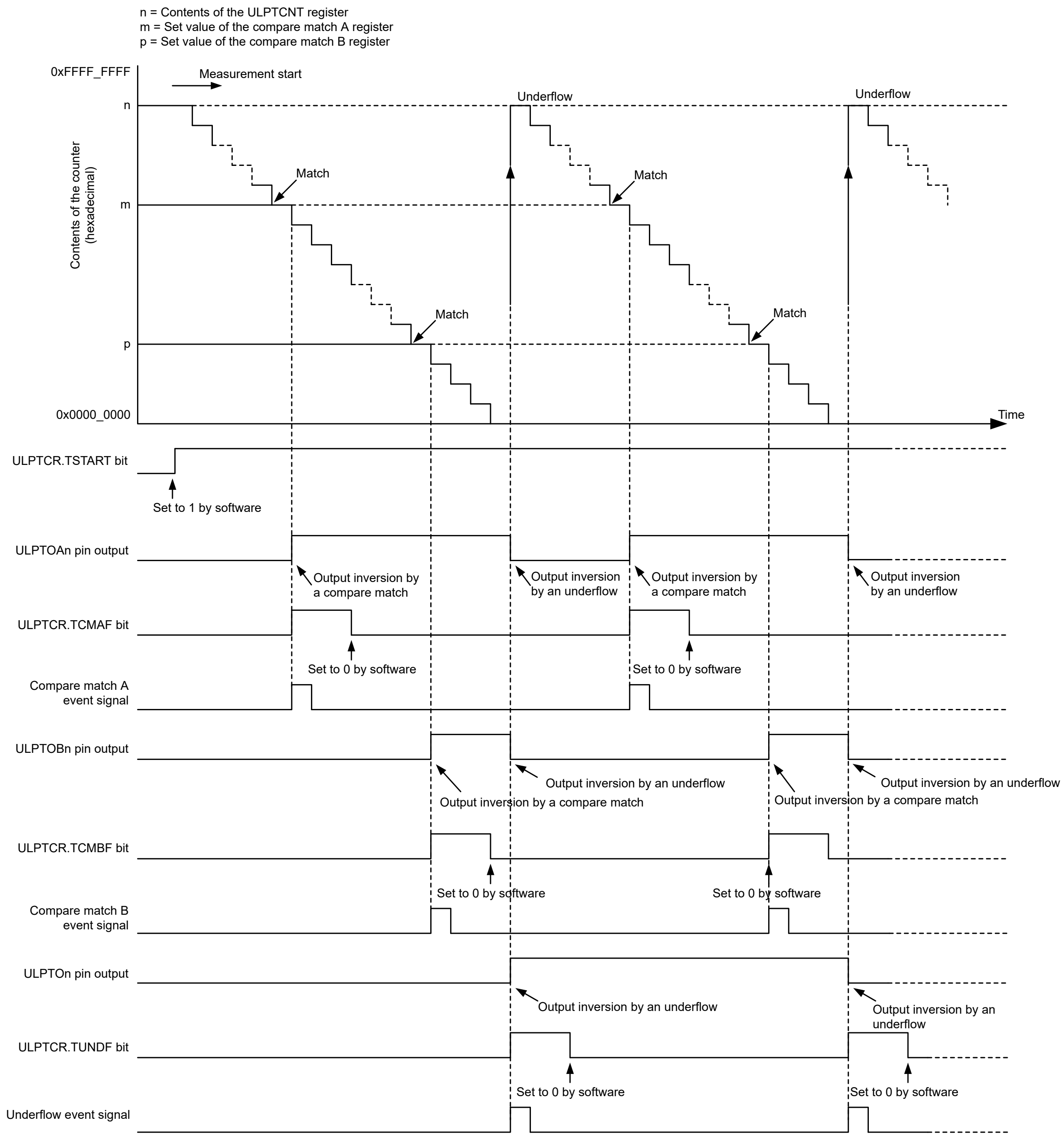


Figure 25.18 Operation example of the compare match function (both the TOPOLA and TOPOLB bits of the ULPTCMSR register are set to 0)

25.4.6 Input and Output Settings for each Mode

Table 25.5 to Table 25.9 show the states of the input pins (ULPTEEn and ULPTEVIn) and output pins (ULPTOn, ULPTOAn, and ULPTOBn) in each mode.

Table 25.5 ULPTEEn pin settings

Operating mode		ULPTMR1 register TMOD1 bit	ULPTMR3 register TEECTL[1:0] bit	ULPTEEn pin input
Timer mode	—	0	00	Input disabled
	Count start		10	Normal input
	Count restart		11	
Event counter mode	Count enable	1	00	
	Count start		10	
	Count restart		11	

Table 25.6 ULPTEVIn pin settings

Operating mode	ULPTMR1 register TMOD1 bit	ULPTMR3 register TEVPOL bit	ULPTEVIn pin input
Timer mode	0	0 or 1	Input disabled
Event counter mode	1	0	Normal input
		1	Inverted input

Table 25.7 ULPTOn pin settings

Operating mode	ULPTIOC register TOE bit	ULPTMR3 register TOPOL bit	ULPTOn pin output
All modes	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled

Table 25.8 ULPTOAn pin settings

Operating mode	ULPTCMSR register TOEA bit	 TOPOLA bit	ULPTOAn pin output
All modes	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled

Table 25.9 ULPTOBn pin settings

Operating mode	ULPTCMSR register TOEB bit	 TOPOLB bit	ULPTOBn pin output
All modes	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled

25.4.7 Standby Mode

The ULPT can operate in Software Standby mode or Deep Software Standby mode 1. Set it to each standby mode with count operation start (ULPTCR.TSTART = 1, ULPTCR.TCSTF = 1).

Table 25.10 lists the settings available in Software Standby mode or Deep Software Standby mode 1.

Table 25.10 Settings available in Software Standby mode or Deep Software Standby mode 1

Section No.	Count operation			Count source	CPU recovery source
	Mode 1	Mode 2	Mode 3		
section 25.4.1.1. Timer and Continuous Modes	Timer	Continuous	—	ULPTLCLK, ULPTSCLK	• Underflow • Compare match A/B
section 25.4.1.5. Common, Continuous, and Count Start Modes			Count start		
section 25.4.1.6. Common, Continuous, and Count Restart Modes			Count restart		
section 25.4.1.2. Timer and One-shot Modes		One-shot	—		
section 25.4.1.7. Common, One-shot, and Count Start Modes			Count start		
section 25.4.1.8. Common, One-shot, and Count Restart Modes			Count restart		
section 25.4.1.3. Event Counter, Continuous, and Count Enable Modes	Event counter	Continuous	Count enable	ULPTEVIn pin*1	• Underflow • Compare match A/B
section 25.4.1.5. Common, Continuous, and Count Start Modes			Count start		
section 25.4.1.6. Common, Continuous, and Count Restart Modes			Count restart		
section 25.4.1.4. Event Counter, One-shot, and Count Enable Modes		One-shot	Count enable		
section 25.4.1.7. Common, One-shot, and Count Start Modes			Count start		
section 25.4.1.8. Common, One-shot, and Count Restart Modes			Count restart		

Note: The ULPT can operate in all modes even in Software Standby mode or Deep Software Standby mode 1. For details about the count operations and register settings, see section 25.4.1. Count Operation.

Note 1. Only ULPTEVIn pin can be used in Deep Software Standby mode 1.

Only in event counter mode (ULPTMR1.TMOD1 = 1), set the ULPTIOC.TIPF[1:0] bits to 00 (no filter) before entering standby mode.

In Software Standby mode or Deep Software Standby mode 1, the digital filter clock (PCLKB) stops (the digital filter does not function).

It is prohibited to rewrite the ULPTCNT, ULPTCMA, and ULPTCMB registers immediately before setting each standby mode.

If the ULPTCNT, ULPTCMA, and ULPTCMB registers are rewritten while the counter is running, set each standby mode after four or more cycles of the count source.

25.4.8 Interrupt Sources

The ULPT has three interrupt sources:

- ULPTI
- ULPTCMAI
- ULPTCMBI.

Table 25.11 lists the interrupt sources.

Table 25.11 ULPT interrupt sources (1 of 2)

Name	Interrupt source	DMAC/DTC activation
ULPTn_ULPTI	When the counter underflows	Possible
ULPTn_ULPTCMAI	When the ULPTCNT value matches the ULPTCMA value	Possible

Table 25.11 ULPT interrupt sources (2 of 2)

Name	Interrupt source	DMAC/DTC activation
ULPTn_ULPTCMBI	When the ULPTCNT value matches the ULPTCMB value	Possible

25.4.9 Event Output to the ELC

The ULPT is capable of link operation for a specified module through the event link controller (ELC) using the interrupt request signal as an event signal.

The ULPT outputs an event signal when compare match A/compare match B or an underflow occurs. For details, see section 19, Event Link Controller (ELC).

25.5 Usage Notes

25.5.1 Start and Stop Control of the Counter

Condition A: In timer mode (ULPTMR1.TMOD1 = 0), continuous mode (ULPTMR3.TCNTCTL = 0), and count enable mode (ULPTMR3.TEECTL[1:0] = 00)

When the counter is stopped (with the TCSTF bit of the ULPTCR register set to 0), even if 1 (start the counter) is set to the TSTART bit of the ULPTCR register, the TCSTF bit remains at 0 (counter stopped) for five cycles of the count source.

Do not access the registers*1 related to the ULPT other than the TCSTF bit until the TCSTF bit is set to 1 (counter running).

When the counter is running (with the TCSTF bit set to 1), even if 0 (stop the counter) is set to the TSTART bit, the TCSTF bit remains at 1 (counter running) for five cycles of the count source.

Do not access the registers*1 related to the ULPT other than the TCSTF bit until the TCSTF bit is set to 0 (counter stopped).

Before changing the TSTART bit from 0 to 1, clear the interrupt register. For details, see section 14, Interrupt Controller Unit (ICU).

Note 1. Registers related to the ULPT: ULPTCNT, ULPTCMA, ULPTCMB, ULPTCR, ULPTMR1, ULPTMR2, ULPTMR3, ULPTIOC, ULPTISR, and ULPTCMSR.

Condition B: In event counter mode (ULPTMR1.TMOD1 = 1), continuous mode (ULPTMR3.TCNTCTL = 0), and count enable mode (ULPTMR3.TEECTL[1:0] = 00)

When the counter is stopped (with the TCSTF bit of the ULPTCR register set to 0), even if 1 (start the counter) is set to the TSTART bit of the ULPTCR register, the TCSTF bit remains at 0 (counter stopped) for two cycles of the count source.

Do not access the registers*1 related to the ULPT other than the TCSTF bit until the TCSTF bit is set to 1 (counter running).

When the counter is running (with the TCSTF bit set to 1), even if 0 (stop the counter) is set to the TSTART bit, the TCSTF bit remains at 1 (counter running) for two cycles of the count source.

Do not access the registers*1 related to the ULPT other than the TCSTF bit until the TCSTF bit is set to 0 (counter stopped).

Before changing the TSTART bit from 0 to 1, clear the interrupt register. For details, see section 14, Interrupt Controller Unit (ICU).

Note 1. Registers related to the ULPT: ULPTCNT, ULPTCMA, ULPTCMB, ULPTCR, ULPTMR1, ULPTMR2, ULPTMR3, ULPTIOC, ULPTISR, and ULPTCMSR.

Condition C: In count start mode or count restart mode (ULPTMR3.TEECTL[1:0] bits = 10 or 11)

When the counter is stopped (with the TCSTF bit of the ULPTCR register set to 0), even if 1 is set to the TSTART bit of the ULPTCR register and then the counter is started on an edge trigger of the ULPTEEn pin, the TCSTF bit remains at 0 (counter stopped) for five cycles of the count source.

Do not access the registers*1 related to the ULPT other than the TCSTF bit until the TCSTF bit is set to 1 (counter running).

When the counter is running (with the TCSTF bit set to 1), even if 0 (stop the counter) is set to the TSTART bit, the TCSTF bit remains at 1 (counter running) for five cycles of the count source.

Do not access the registers*1 related to the ULPT other than the TCSTF bit until the TCSTF bit is set to 0 (counter stopped).

Before changing the TSTART bit from 0 to 1, clear the interrupt register. For details, see section 14, Interrupt Controller Unit (ICU).

Note 1. Registers related to the ULPT: ULPTCNT, ULPTCMA, ULPTCMB, ULPTCR, ULPTMR1, ULPTMR2, ULPTMR3, ULPTIOC, ULPTISR, and ULPTCMSR.

25.5.2 Access to Flags (TUNDF, TCMAF, and TCMBF bits in ULPTCR Register)

Set 1 to the TUNDF, TCMAF, and TCMBF bits when changing the TSTART or TSTOP bits to prevent clearing TUNDF, TCMAF, or TCMBF by mistake. At this time, setting 1 to TUNDF, TCMAF, and TCMBF bits are ignored.

25.5.3 Access to the ULPTCNT, ULPTCMA, and ULPTCMB Registers

When setting to the same register sequentially while the counter is running (ULPTCR.TCSTF bit = 1), leave three or more cycles of count source between writes. The following registers are applicable:

- ULPTCNT
- ULPTCMA
- ULPTCMB

25.5.4 Mode Changes

Registers related to the operating mode of the ULPT (ULPTMR1, ULPTMR2, ULPTMR3, ULPTIOC, ULPTISR, and ULPTCMSR) can be changed only while the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0). Do not change these registers while the counter is running.

If a register related to the operating mode of the ULPT is changed, the values of the TUNDF, TCMAF, and TCMBF bits of the ULPTCR register are undefined. Before starting the counter, write 0 to the TUNDF, TCMAF, and TCMBF bits.

- TUNDF (no underflow)
- TCMAF (no match for compare match A)
- TCMBF (no match for compare match B)

25.5.5 Setting the ULPTOn, ULPTOAn, and ULPTOBn Pins

To use the ULPTOn, ULPTOAn, and ULPTOBn pins as output pins, set up the ULPT, and then set the output in the PORT register after determining the initial output values.

25.5.6 Calculating the Number of Events

In event counter mode, the number of events is calculated using the following formula:

Number of events = Initial counter value (ULPTCNT register write value) − Counter value at the end of a valid event

25.5.7 When the Counter is Stopped Forcibly by the TSTOP Bit

Do not access the ULPTCNT, ULPTCMA, ULPTCMB, ULPTCR, ULPTMR1, ULPTMR2, or ULPTMR3 register during one cycle of the count source after stopping the counter forcibly by the ULPTCR.TSTOP bit.

25.5.8 Digital Filter

When using a digital filter, do not start the timer within five cycles of the digital filter clock after setting the ULPTIOC.TIPF[1:0] bits.

Also, when ULPTMR3.TEVPOL bit is changed while a digital filter is in use, do not start the timer within five cycles of the digital filter clock after setting the bit.

In Software Standby mode or Deep Software Standby mode 1, the digital filter clock (PCLKB) stops and the digital filter does not function. Before entering Software standby mode or Deep Software Standby mode 1, set the ULPTIOC.TIPF[1:0] bits to 00 (no filter).

25.5.9 Module-Stop Function Setting

ULPT operation can be disabled or enabled using Module Stop Control Register E (MSTPCRE). The ULPT module is initially stopped after reset. Releasing the module-stop state enables access to the registers.

25.5.10 Setting the ULPTEEn and ULPTEVIn Pins

To use the ULPTEEn and ULPTEVIn pins as input pins, set up the ULPT, and then set PmnPFS.PMR bit to 1.

26. Realtime Clock (RTC)

26.1 Overview

The realtime clock (RTC) has two counting modes, calendar count mode and binary count mode, that are used by switching register settings. For calendar count mode, the RTC has a 100-year calendar from 2000 to 2099 and automatically adjusts dates for leap years. For binary count mode, the RTC counts seconds and retains the information as a serial value. Binary count mode can be used for calendars other than the Gregorian (Western) calendar.

The sub-clock oscillator or LOCO can be selected as the count source of the time counters. The RTC uses a 128-Hz clock acquired by dividing the count source by a prescaler. Year, month, date, day-of-week, a.m. /p.m. (in 12-hour mode), hour, minute, second, or 32-bit binary is counted by 1/128 second.

Table 26.1 lists the RTC specifications, Figure 26.1 shows a block diagram, and Table 26.2 lists the I/O pins.

Table 26.1 RTC specifications

Parameter	Specifications
Count mode	Calendar count mode/binary count mode
Count source*1	Sub-clock (XCIN), External clock input (EXCIN) or LOCO
Clock and calendar functions	● Calendar count mode – Year, month, date, day of week, hour, minute, second are counted, BCD display – 12 hours/24 hours mode switching function – 30 seconds adjustment function (a number less than 30 is rounded down to 00 seconds, and 30 seconds or more are rounded up to 1 minute) – Automatic adjustment function for leap years ● Binary count mode – Count seconds in 32 bits, binary display ● Shared by both modes – Start/stop function – The sub-second digit is displayed in binary units (1 Hz, 2 Hz, 4 Hz, 8 Hz, 16 Hz, 32 Hz, or 64 Hz) – Clock error correction function – Clock (1-Hz/64-Hz) output
Interrupts	● Alarm interrupt (RTC_ALM) – As an alarm interrupt condition, selectable for comparison with the following: • Calendar count mode: Year, month, date, day-of-week, hour, minute, or second can be selected • Binary count mode: Each bit of the 32-bit binary counter ● Periodic interrupt (RTC_PRD) – 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, 1/128 second, or 1/256 second can be selected as an interrupt period. ● Carry interrupt (RTC_CUP) – An interrupt is generated at either of the following conditions: • When a carry from the 64-Hz counter to the second counter is generated. • When the 64-Hz counter is changed and the R64CNT register is read at the same time. (32-KHz count mode is only for 64-Hz counter reading) ● Return from Software Standby or Deep Software Standby mode can be performed by the alarm interrupt or periodic interrupt
Time capture function	● Times can be captured when the edge of the time capture event input pin is detected. For every event input, month, date, hour, minute, and second are captured or the 32-bit binary counter value is captured. ● Interrupt can be generated when the edge of the time capture event input is detected. The time capture event input pin and IRQ are shared.
Event link function	Periodic event output (RTC_PRD)
TrustZone filter	Security and Privilege attribution can be set

Note 1. The frequency of the peripheral module clock (PCLKB) ≥ the frequency of the count source should be satisfied.

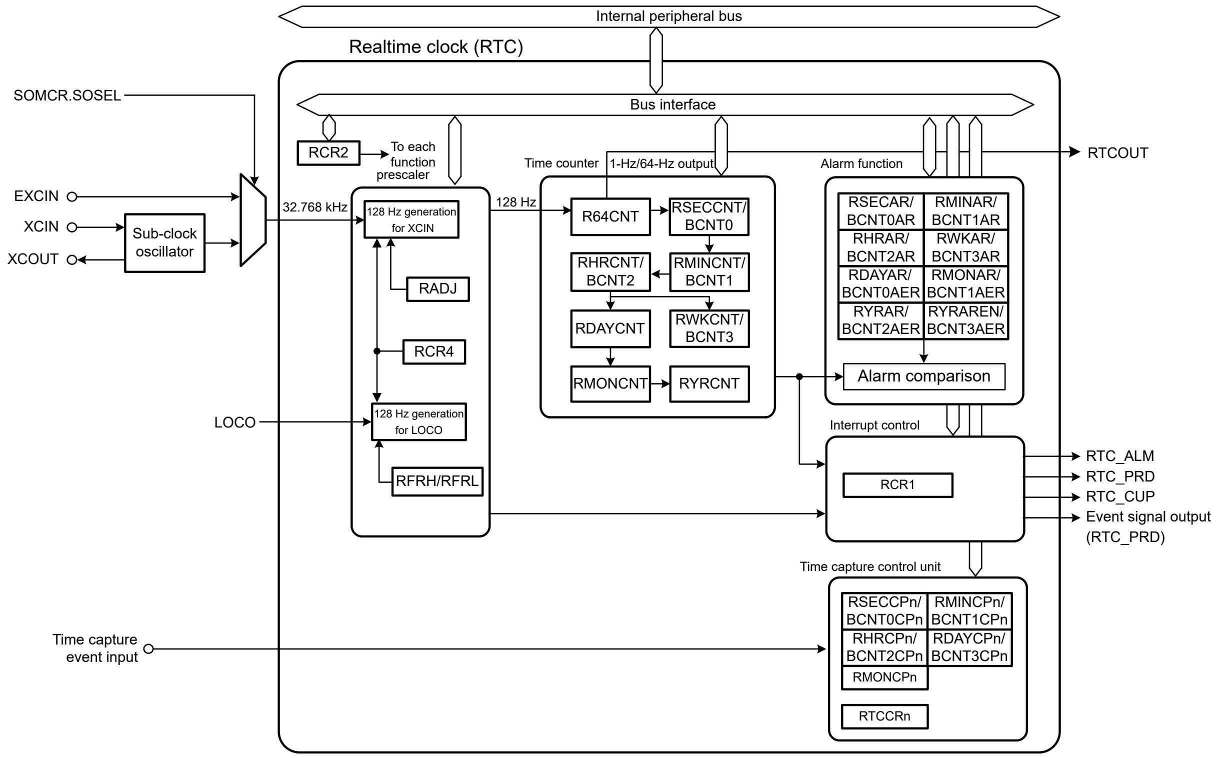


Figure 26.1 RTC block diagram

Table 26.2 RTC I/O pins

Pin name	I/O	Description
XCIN	Input	Connect a 32.768-kHz crystal to these pins
XCOUT	Output	
EXCIN	Input	External sub-clock input
RTCOUT	Output	This pin is used to output a 1-Hz/64-Hz waveform, but not in Deep Software Standby mode
RTCICn (n = 0 to 2)	Input	Time capture event input pins RTCICn can be controlled by the VBTICTLR register. For more information, see section 12, Battery Backup Function and section 20, I/O Ports.

26.2 Register Descriptions

Write or read from the RTC registers as described in section 26.6.5. Notes on Writing to and Reading from Registers.

If the value in an RTC register after a reset is given as x (undefined bits) in the list, it is not initialized by a reset. When RTC enters the reset state or a low power state during counting operations, for example, while the RCR2.START bit is 1, the year, month, day of the week, date, hours, minutes, seconds, and 64-Hz counters continue to operate.

Note: A reset generated while writing to a register might destroy the register value. In addition, do not allow the MCU to enter Software Standby mode, Deep Software Standby mode, or battery backup state immediately after setting any of these registers. For details, see section 26.6.4. Transitions to Low Power Modes After Setting Registers.

26.2.1 R64CNT : 64-Hz Counter

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	F1HZ	F2HZ	F4HZ	F8HZ	F16HZ	F32HZ	F64HZ
Value after reset:	0	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
0	F64HZ	64-Hz Flag This bit indicates the 64-Hz state of the sub-second digit.	R
1	F32HZ	32-Hz Flag This bit indicates the 32-Hz state of the sub-second digit.	R
2	F16HZ	16-Hz Flag This bit indicates the 16-Hz state of the sub-second digit.	R
3	F8HZ	8-Hz Flag This bit indicates the 8-Hz state of the sub-second digit.	R
4	F4HZ	4-Hz Flag This bit indicates the 4-Hz state of the sub-second digit.	R
5	F2HZ	2-Hz Flag This bit indicates the 2-Hz state of the sub-second digit.	R
6	F1HZ	1-Hz Flag This bit indicates the 1-Hz state of the sub-second digit.	R
7	—	This bit is read as 0.	R

Note: S-TYPE3, P-TYPE3

The R64CNT counter is used in both calendar count mode and binary count mode The 64-Hz counter (R64CNT) generates the period for a second by counting up periods of the 128-Hz clock. The state in the sub-second range can be confirmed by reading this counter.

This counter is set to 0x00 by an RTC software reset or an execution of a 30-second adjustment. To read this counter, follow the procedure in section 26.3.5. Reading 64-Hz Counter and Time.

26.2.2 RSECCNT : Second Counter (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	SEC10[2:0]			SEC1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	SEC1[3:0]	1-Second Count Counts from 0 to 9 every second. When a carry is generated, 1 is added to the tens place.	R/W
6:4	SEC10[2:0]	10-Second Count Counts from 0 to 5 for 60-second counting.	R/W
7	—	The read value is undefined. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RSECCNT counter sets and counts the BCD-coded second value. It counts the carries generated once per second in the 64-Hz counter.

The setting range is decimal 00 to 59. The RTC does not operate normally if any other value is set. Before writing to this register, you must stop the count operation using the START bit in RCR2.

To read this counter, follow the procedure in section 26.3.5. Reading 64-Hz Counter and Time.

26.2.3 RMINCNT : Minute Counter (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	MIN10[2:0]			MIN1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	MIN1[3:0]	1-Minute Count Counts from 0 to 9 every minute. When a carry is generated, 1 is added to the tens place.	R/W
6:4	MIN10[2:0]	10-Minute Count Counts from 0 to 5 for 60-minute counting.	R/W
7	—	The read value is undefined. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RMINCNT counter sets and counts the BCD-coded minute value. It counts the carries generated once every minute in the second counter.

A value from 00 through 59 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, you must stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 26.3.5. Reading 64-Hz Counter and Time.

26.2.4 RHRCNT : Hour Counter (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	PM	HR10[1:0]		HR1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	HR1[3:0]	1-Hour Count Counts from 0 to 9 once per hour. When a carry is generated, 1 is added to the tens place.	R/W
5:4	HR10[1:0]	10-Hour Count Counts from 0 to 2 once per carry from the ones place.	R/W
6	PM	AM/PM select for time counter setting. 0: AM 1: PM	R/W
7	—	The read value is undefined. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RHRCNT counter sets and counts the BCD-coded hour value. It counts the carries generated once per hour in the minute counter. The specifiable time differs based on the setting in the hours mode bit (RCR2.HR24):

- When the RCR2.HR24 bit is 0 – from 00 to 11 (in BCD).
- When the RCR2.HR24 bit is 1 – from 00 to 23 (in BCD).

If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, you must stop the count operation using the START bit in RCR2. The PM bit is only enabled when the RCR2.HR24 bit is 0.

Otherwise, the setting in the PM bit has no effect. To read this counter, follow the procedure in section 26.3.5. Reading 64-Hz Counter and Time.

26.2.5 RWKCNT : Day-of-Week Counter (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	DAYW[2:0]		
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
2:0	DAYW[2:0]	Day-of-Week Counting 0 0 0: Sunday 0 0 1: Monday 0 1 0: Tuesday 0 1 1: Wednesday 1 0 0: Thursday 1 0 1: Friday 1 1 0: Saturday 1 1 1: Setting prohibited	R/W
7:3	—	The read values are undefined. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RWKCNT counter sets and counts in the coded day-of-week value. It counts the carries generated once per day in the hour counter. A value from 0 through 6 can be specified. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, you must stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 26.3.5. Reading 64-Hz Counter and Time.

26.2.6 BCNTn : Binary Counter n (n = 0 to 3) (in Binary Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x02 + 0x02 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	BCNT[7:0]							
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
7:0	BCNT[7:0]	Binary Counter	R/W

Note: S-TYPE3, P-TYPE3

BCNTn is a read/write 8-bit register to access BCNT[31:0] that is a 32-bit binary counter. BCNT3 is assigned to the BCNT[31:24] bits, BCNT2 is assigned to the BCNT[23:16] bits, BCNT1 is assigned to the BCNT[15:8] bits, and BCNT0 is assigned to the BCNT[7:0] bits. BCNTn performs count operation by a carry generated for each second of the 64-Hz counter. Before writing to this register, you must stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 26.3.5. Reading 64-Hz Counter and Time.

26.2.7 RDAYCNT : Day Counter

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x0A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	DATE10[1:0]		DATE1[3:0]			
Value after reset:	0	0	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	DATE1[3:0]	1-Day Count Counts from 0 to 9 once per day. When a carry is generated, 1 is added to the tens place.	R/W
5:4	DATE10[1:0]	10-Day Count Counts from 0 to 3 once per carry from the ones place.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RDAYCNT counter is used in calendar count mode to set and count the BCD-coded date value. It counts the carries generated once per day in the hour counter. The count operation depends on the month and whether the year is a leap year. Leap years are determined according to whether the year counter (RYRCNT) value is divisible by 400, 100, and 4.

A value from 01 through 31 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. When specifying a value, the range of specifiable days depends on the month and whether the year is a leap year. Before writing to this register, you must stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 26.3.5. Reading 64-Hz Counter and Time.

26.2.8 RMONCNT : Month Counter

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x0C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	MON10	MON1[3:0]			
Value after reset:	0	0	0	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	MON1[3:0]	1-Month Count Counts from 0 to 9 once per month. When a carry is generated, 1 is added to the tens place.	R/W
4	MON10	10-Month Count Counts from 0 to 1 once per carry from the ones place.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RMONCNT counter is used in calendar count mode to set and count the BCD-coded month value. It counts the carries generated once per month in the date counter.

A value from 01 through 12 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, you must stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 26.3.5. Reading 64-Hz Counter and Time.

26.2.9 RYRCNT : Year Counter

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x0E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	YR10[3:0]				YR1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	YR1[3:0]	1-Year Count Counts from 0 to 9 once per year. When a carry is generated, 1 is added to the tens place.	R/W
7:4	YR10[3:0]	10-Year Count Counts from 0 to 9 once per carry from ones place. When a carry is generated in the tens place, 1 is added to the hundreds place.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RYRCNT counter is used in calendar count mode to set and count the BCD-coded year value. It counts the carries generated once per year in the month counter.

A value from 00 through 99 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, you must stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 26.3.5. Reading 64-Hz Counter and Time.

26.2.10 RSECAR : Second Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENB	SEC10[2:0]			SEC1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	SEC1[3:0]	1 Second Value for the ones place of seconds.	R/W
6:4	SEC10[2:0]	10 Seconds Value for the tens place of seconds.	R/W
7	ENB	ENB 0: Do not compare register value with RSECCNT counter value 1: Compare register value with RSECCNT counter value	R/W

Note: S-TYPE3, P-TYPE3

RSECAR is an alarm register associated with the BCD-coded second counter RSECCNT. When the ENB bit is set to 1, the RSECAR value is compared with the RSECCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR

- RMONAR
- RYRAREN

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. RSECAR values from 00 through 59 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 0x00 by an RTC software reset.

26.2.11 RMINAR : Minute Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x12

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENB	MIN10[2:0]			MIN1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	MIN1[3:0]	1 Minute Value for the ones place of minutes.	R/W
6:4	MIN10[2:0]	10 Minutes Value for the tens place of minutes.	R/W
7	ENB	ENB 0: Do not compare register value with RMINCNT counter value 1: Compare register value with RMINCNT counter value	R/W

Note: S-TYPE3, P-TYPE3

RMINAR is an alarm register associated with the BCD-coded minute counter RMINCNT. When the ENB bit is set to 1, the RMINAR value is compared with the RMINCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. RMINAR values from 00 through 59 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 0x00 by an RTC software reset.

26.2.12 RHRAR : Hour Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x14

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENB	PM	HR10[1:0]		HR1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	HR1[3:0]	1 Hour Value for the ones place of hours.	R/W
5:4	HR10[1:0]	10 Hours Value for the tens place of hours.	R/W
6	PM	AM/PM select for alarm setting. 0: AM 1: PM	R/W
7	ENB	ENB 0: Do not compare register value with RHRCNT counter value 1: Compare register value with RHRCNT counter value	R/W

Note: S-TYPE3, P-TYPE3

RHRAR is an alarm register associated with the BCD-coded hour counter RHRCNT. When the ENB bit is set to 1, the RHRAR value is compared with the RHRCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. The specifiable time differs according to the setting in the hours mode bit (RCR2.HR24):

- When the RCR2.HR24 bit is 0 – From 00 to 11 (in BCD).
- When the RCR2.HR24 bit is 1 – From 00 to 23 (in BCD).

If a value outside of this range is specified, the RTC does not operate correctly. When the RCR2.HR24 bit is 0, you must set the PM bit. When the RCR2.HR24 bit is 1, the setting in the PM bit has no effect. This register is set to 0x00 by an RTC software reset.

26.2.13 RWKAR : Day-of-Week Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x16

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENB	—	—	—	—	DAYW[2:0]		
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
2:0	DAYW[2:0]	Day-of-Week Setting 0 0 0: Sunday 0 0 1: Monday 0 1 0: Tuesday 0 1 1: Wednesday 1 0 0: Thursday 1 0 1: Friday 1 1 0: Saturday 1 1 1: Setting prohibited	R/W
6:3	—	The read values are undefined. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7	ENB	ENB 0: Do not compare register value with RWKCNT counter value 1: Compare register value with RWKCNT counter value	R/W

Note: S-TYPE3, P-TYPE3

RWKAR is an alarm register associated with the coded day-of-week counter RWKCNT. When the ENB bit is set to 1, the RWKAR value is compared with the RWKCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. RWKAR values from 0 through 6 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 0x00 by an RTC software reset.

26.2.14 BCNTnAR : Binary Counter n Alarm Register (n = 0 to 3) (in Binary Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x10 + 0x02 × n

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

Bit	Symbol	Function	R/W
7:0	BCNTAR	Alarm register associated with the 32-bit binary counter	R/W

Note: S-TYPE3, P-TYPE3

BCNTnAR is a read/write alarm register associated with the 32-bit binary counter. BCNT3AR is assigned to the BCNTAR[31:24] bits, BCNT2AR is assigned to the BCNTAR[23:16] bits, BCNT1AR is assigned to the BCNTAR[15:8] bits, and BCNT0AR is assigned to the BCNTAR[7:0]. This register is set to 0x00 by an RTC software reset.

26.2.15 RDAYAR : Date Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x18

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENB	—	DATE10[1:0]		DATE1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	DATE1[3:0]	1 Day Value for the ones place of days.	R/W

Bit	Symbol	Function	R/W
5:4	DATE10[1:0]	10 Days Value for the tens place of days.	R/W
6	—	The read value is undefined. The write value should be 0.	R/W
7	ENB	ENB 0: Do not compare register value with RDAYCNT counter value 1: Compare register value with RDAYCNT counter value	R/W

Note: S-TYPE3, P-TYPE3

RDAYAR is an alarm register associated with the BCD-coded date counter RDAYCNT. When the ENB bit is set to 1, the RDAYAR value is compared with the RDAYCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. The RDAYAR values from 01 through 31 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 0x00 by an RTC software reset.

26.2.16 RMONAR : Month Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x1A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENB	—	—	MON10	MON1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	MON1[3:0]	1 Month Value for the ones place of months.	R/W
4	MON10	10 Months Value for the tens place of months.	R/W
6:5	—	The read values are undefined. The write value should be 0.	R/W
7	ENB	ENB 0: Do not compare register value with RMONCNT counter value 1: Compare register value with RMONCNT counter value	R/W

Note: S-TYPE3, P-TYPE3

RMONAR is an alarm register associated with the BCD-coded month counter RMONCNT. When the ENB bit is set to 1, the RMONAR value is compared with the RMONCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR

- RDAYAR
- RMONAR
- RYRAREN

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. The RMONAR values from 01 through 12 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 0x00 by an RTC software reset.

26.2.17 RYRAR : Year Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x1C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	YR10[3:0]				YR1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	YR1[3:0]	1 Year Value for the ones place of years.	R/W
7:4	YR10[3:0]	10 Years Value for the tens place of years.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

RYRAR is an alarm register associated with the BCD-coded year counter RYRCNT. The RYRAR values from 00 through 99 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 0x0000 by an RTC software reset.

26.2.18 RYRAREN : Year Alarm Enable Register (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x1E

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

Bit	Symbol	Function	R/W
6:0	—	The read values are undefined. The write value should be 0.	R/W
7	ENB	ENB 0: Do not compare register value with the RYRCNT counter value 1: Compare register value with the RYRCNT counter value	R/W

Note: S-TYPE3, P-TYPE3

When the ENB bit in the RYRAREN register is set to 1, the RYRAR value is compared with the RYRCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR

- RDAYAR
- RMONAR
- RYRAREN

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. This register is set to 0x00 by an RTC software reset.

26.2.19 BCNTnAER : Binary Counter n Alarm Enable Register (n = 0 to 3) (in Binary Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x18 + 0x02 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENB[7:0]							
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
7:0	ENB[7:0]	Setting the alarm enable associated with the 32-bit binary counter	R/W

Note: S-TYPE3, P-TYPE3

BCNTnAER is a read/write register for setting the alarm enable (BCNTAER) associated with the 32-bit binary counter. BCNT3AER is assigned to the BCNTAER.ENB[31:24] bits, BCNT2AER register is assigned to the BCNTAER.ENB[23:16] bits, BCNT1AER is assigned to the BCNTAER.ENB[15:8] bits, and BCNT0AER is assigned to the BCNTAER.ENB[7:0] bits. The binary counter (BCNT[31:0]) associated with the BCNTAER.ENB[31:0] bits that are set to 1 is compared with the binary alarm register (BCNTAR) and, when all match, the IR flag associated with the RTC_ALM interrupt is set to 1. This register is set to 0x00 by an RTC software reset.

26.2.20 RCR1 : RTC Control Register 1

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x22

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PES[3:0]				RTCOS	PIE	CIE	AIE
Value after reset:	x	x	x	x	0	x	0	x

Bit	Symbol	Function	R/W
0	AIE	Alarm Interrupt Enable 0: Disable alarm interrupt requests 1: Enable alarm interrupt requests	R/W
1	CIE	Carry Interrupt Enable 0: Disable carry interrupt requests 1: Enable carry interrupt requests	R/W
2	PIE	Periodic Interrupt Enable 0: Disable periodic interrupt requests 1: Enable periodic interrupt requests	R/W
3	RTCOS	RTCOUT Output Select 0: Outputs 1 Hz on RTCOUT 1: Outputs 64 Hz RTCOUT	R/W

Bit	Symbol	Function	R/W
7:4	PES[3:0]	Periodic Interrupt Select 0x6: Generate periodic interrupt every 1/256 second*1 0x7: Generate periodic interrupt every 1/128 second 0x8: Generate periodic interrupt every 1/64 second 0x9: Generate periodic interrupt every 1/32 second 0xA: Generate periodic interrupt every 1/16 second 0xB: Generate periodic interrupt every 1/8 second 0xC: Generate periodic interrupt every 1/4 second 0xD: Generate periodic interrupt every 1/2 second 0xE: Generate periodic interrupt every 1 second 0xF: Generate periodic interrupt every 2 seconds Others: Do not generate periodic interrupts	R/W

Note: S-TYPE3, P-TYPE3

Note 1. When LOCO is selected (RCR4.RCKSEL = 1) while PES[3:0] = 0x6, a periodic interrupt is generated every 1/128 second.

The RCR1 register is used in both calendar count mode and binary count mode. Bits AIE, PIE, and PES[3:0] are updated synchronously with the count source. When the RCR1 register is modified, check that all the bits are updated before proceeding.

AIE bit (Alarm Interrupt Enable)

The AIE bit enables or disables alarm interrupt requests.

If the times indicated in the counters and alarm settings match in Deep Software Standby mode, the MCU returns from Deep Software Standby mode regardless of the AIE bit value.

CIE bit (Carry Interrupt Enable)

The CIE bit enables or disables interrupt requests when a carry to the RSECCNT/BCNT0 register occurs, or when a carry to the 64-Hz counter (R64CNT) occurs while reading the 64-Hz counter.

PIE bit (Periodic Interrupt Enable)

The PIE bit enables or disabled a periodic interrupt.

If the periods indicated in the counters and PES[3:0] settings match in Deep Software Standby mode, the MCU returns from Deep Software Standby mode regardless of the PIE bit value.

RTCOS bit (RTCOUT Output Select)

The RTCOS bit selects the RTCOUT output period. The RTCOS bit must be rewritten while the count operation is stopped (RCR2.START = 0) and the RTCOUT output is disabled (RCR2.RTCOE = 0). When RTCOUT is output to an external pin, the RCR2.RTCOE bit must be enabled.

PES[3:0] bits (Periodic Interrupt Select)

The PES[3:0] bits specify the period for the periodic interrupt. A periodic interrupt is generated with the period specified in these bits.

26.2.21 RCR2 : RTC Control Register 2 (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x24

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CNTMD	HR24	AADJP	AADJE	RTCOE	ADJ30	RESET	START
Value after reset:	x	x	x	x	0	0	0	x

Bit	Symbol	Function	R/W
0	START	Start 0: Stop prescaler and time counter 1: Operate prescaler and time counter normally	R/W

Bit	Symbol	Function	R/W
1	RESET	RTC Software Reset 0: In writing: Invalid (writing 0 has no effect). In reading: Normal time operation in progress, or an RTC software reset has completed. 1: In writing: Initialize the prescaler and target registers for RTC software reset*1. In reading: RTC software reset in progress.	R/W
2	ADJ30	30-Second Adjustment 0: In writing: Invalid (writing 0 has no effect). In reading: Normal time operation in progress, or 30-second adjustment has completed. 1: In writing: Execute 30-second adjustment. In reading: 30-second adjustment in progress.	R/W
3	RTCOE	RTCOUT Output Enable 0: Disable RTCOUT output 1: Enable RTCOUT output	R/W
4	AADJE	Automatic Adjustment Enable*2*3 0: Disable automatic adjustment 1: Enable automatic adjustment	R/W
5	AADJP	Automatic Adjustment Period Select*2*3 0: The RADJ.ADJ[5:0] setting from the count value of the prescaler every minute. 1: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every 10 seconds.	R/W
6	HR24	Hours Mode*3 0: Operate RTC in 12-hour mode 1: Operate RTC in 24-hour mode	R/W
7	CNTMD	Count Mode Select*4 0: Calendar count mode 1: Binary count mode	R/W

Note: S-TYPE3, P-TYPE3
Note 1. R64CNT, RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, RYRAR, RYRAREN, RADJ, RTCCRn, RSECCPn, RMINCPn, RHRCPn, RDAYCPn, RMONCPn, RCR2.ADJ30, RCR2.AADJE, RCR2.AADJP.
Note 2. When LOCO is selected, the setting of this bit is disabled.
Note 3. When rewriting this bit, confirm that the value has been rewritten before performing the following processing. See section 26.6.5. Notes on Writing to and Reading from Registers for notes on register writing/reading.
Note 4. When rewriting this bit, confirm that the value has been rewritten before performing the following processing.

The RCR2 register is related to hours mode, automatic adjustment function, enabling RTCOUT output, 30-second adjustment, RTC software reset, and controlling count operation.

START bit (Start)

The START bit stops or restarts the prescaler or time counter operation. This bit is updated in synchronization with the next cycle of the count source. When the START bit is modified, check that the bit is updated before proceeding.

RESET bit (RTC Software Reset)

The RESET bit initializes the prescaler and registers to be reset by RTC software. When 1 is written to this bit, initialization starts in synchronization with the count source. When the initialization is complete, the RESET bit is automatically set to 0. Check that this bit is 0 before proceeding.

ADJ30 bit (30-Second Adjustment)

The ADJ30 bit is for 30-second adjustment.

When 1 is written to the ADJ30 bit, the RSECCNT value of less than 30 seconds is rounded down to 00 second and the value of 30 seconds or more is rounded up to 1 minute.

The 30-second adjustment is performed in synchronization with the count source. When 1 is written to this bit, the ADJ30 bit is automatically set to 0 after the 30-second adjustment completes. If 1 is written to the ADJ30 bit, check that the bit is 0 before proceeding. When the 30-second adjustment is performed, the prescaler and R64CNT are also reset. The ADJ30 bit is set to 0 by an RTC software reset.

RTCOE bit (RTCOUT Output Enable)

The RTCOE bit enables output of a 1-Hz/64-Hz clock signal from the RTCOUT pin.

Use the START bit to stop counting before changing the value of the RTCOE bit. Do not stop counting (write 0 to the START bit) and change the value of the RTCOE bit at the same time.

When RTCOUT is to be output from an external pin, enable the RTCOE bit and set up the port control for the pin.

AADJE bit (Automatic Adjustment Enable)

The AADJE bit controls (enables or disables) automatic adjustment.

Set the plus-minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJE bit.

The AADJE bit is set to 0 by an RTC software reset.

AADJP bit (Automatic Adjustment Period Select)

The AADJP bit selects the automatic-adjustment period.

Set the plus-minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJP bit.

The AADJP bit is set to 0 by an RTC software reset.

HR24 bit (Hours Mode)

The HR24 bit specifies whether the RTC operates in 12- or 24-hour mode.

Use the START bit to stop counting before changing the value of the HR24 bit. Do not stop counting (write 0 to the START bit) and change the value of the HR24 bit at the same time.

CNTMD bit (Count Mode Select)

The CNTMD bit specifies whether the RTC count mode operates in calendar count mode or in binary count mode.

When setting the count mode, execute an RTC software reset and start again from the initial settings. This bit is updated in synchronization with the count source. However, the count mode switches only after the RTC software reset. (Bit switches before RTC reset, mode switches after RTC reset.)

For details on initial settings, see section 26.3.1. Outline of Initial Settings of Registers After Power On.

26.2.22 RCR2 : RTC Control Register 2 (in Binary Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x24

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CNTMD	—	AADJP	AADJE	RTCOE	—	RESET	START
Value after reset:	x	x	x	x	0	0	0	x

Bit	Symbol	Function	R/W
0	START	Start 0: Stop the 32-bit binary counter, 64-Hz counter, and prescaler 1: Operate the 32-bit binary counter, 64-Hz counter, and prescaler normally	R/W
1	RESET	RTC Software Reset 0: In writing: Invalid (writing 0 has no effect). In reading: Normal time operation in progress, or an RTC software reset has completed. 1: In writing: Initialize the prescaler and target registers for RTC software reset*1. In reading: RTC software reset in progress.	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
3	RTCOE	RTCOUT Output Enable 0: Disable RTCOUT output 1: Enable RTCOUT output	R/W
4	AADJE	Automatic Adjustment Enable*2*3 0: Disable automatic adjustment 1: Enable automatic adjustment	R/W
5	AADJP	Automatic Adjustment Period Select*2*3 0: Add or subtract RADJ.ADJ [5:0] bits from prescaler count value every 32 seconds 1: Add or subtract RADJ.ADJ [5:0] bits from prescaler count value every 8 seconds.	R/W
6	—	The read value is undefined. The write value should be 0.	R/W
7	CNTMD	Count Mode Select*4 0: Calendar count mode 1: Binary count mode	R/W

Note: S-TYPE3, P-TYPE3

Note 1. R64CNT, BCNTnAR, BCNTnAER, RADJ, RTCCRn, BCNTnCPm, RCR2.ADJ30, RCR2.AADJE, RCR2.AADJP.

Note 2. When LOCO is selected, the setting of this bit is disabled.

Note 3. When rewriting this bit, confirm that the value has been rewritten before performing the following processing. See section 26.6.5. Notes on Writing to and Reading from Registers for notes on register writing/reading.

Note 4. When rewriting this bit, confirm that the value has been rewritten before performing the following processing.

RCR2 in the binary count mode is a register related to the automatic correction function, RTCOUT output enable, RTC software reset, and count mode control.

START bit (Start)

The START bit stops or restarts the prescaler or counter (clock) operation. This bit is updated in synchronization with the count source. When the START bit is modified, check that the bit is updated before proceeding.

RESET bit (RTC Software Reset)

The RESET bit initializes the prescaler and registers to be reset by RTC software. When 1 is written to this bit, initialization starts in synchronization with the count source. When the initialization is complete, the RESET bit is automatically set to 0. When 1 is written to the RESET bit, check that the bit is 0 before proceeding.

RTCOE bit (RTCOUT Output Enable)

The RTCOE bit enables output of a 1-Hz/64-Hz clock signal from the RTCOUT pin.

Use the START bit to stop counting before changing the value of the RTCOE bit. Do not stop counting (write 0 to the START bit) and change the value of the RTCOE bit at the same time. When an RTCOUT signal is to be output from an external pin, enable the port control in addition to setting this bit.

AADJE bit (Automatic Adjustment Enable)

The AADJE bit controls (enables or disables) automatic adjustment.

Set the plus-minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJE bit. The AADJE bit is set to 0 by an RTC software reset.

AADJP bit (Automatic Adjustment Period Select)

The AADJP bit selects the automatic-adjustment period.

Correction period can be selected from 32 second units or 8 second units in binary count mode.

Set the plus-minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJP bit. The AADJP bit is set to 0 by an RTC software reset.

CNTMD bit (Count Mode Select)

The CNTMD bit specifies whether the RTC count mode operates in calendar count mode or in binary count mode.

When setting the count mode, execute an RTC software reset and start again from the initial settings. This bit is updated in synchronization with the count source. However, the count mode switches only after the RTC software reset. (Bit switches before RTC reset, mode switches after RTC reset.)

For details on initial settings, see section 26.3.1. Outline of Initial Settings of Registers After Power On .

26.2.23 RCR4 : RTC Control Register 4

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x28

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

Bit	Symbol	Function	R/W
0	RCKSEL	Count Source Select 0: Sub-clock oscillator is selected 1: LOCO is selected	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RCR4 register is used in both calendar count mode and binary count mode.

RCKSEL bit (Count Source Select)

The RCKSEL bit selects the count source from the sub-clock oscillator and LOCO.

The RCKSEL bit is only used in normal operation mode. When the RCKSEL bit is set to 0, the time is counted with the sub-clock oscillator. When the bit is set to 1, the time is counted with LOCO.

For details on count source setting, see section 26.3.1. Outline of Initial Settings of Registers After Power On and section 26.3.2. Clock and Count Mode Setting Procedure. The count source must be selected only once before specifying the initial settings of the RTC registers at power on.

26.2.24 RFRL : Frequency Register L

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x2C

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

Bit	Symbol	Function	R/W
15:0	RFC[15:0]	Frequency Comparison Value Write 0x00FF to this register when using the LOCO.	R/W

Note: S-TYPE3, P-TYPE3

RFRL is a register for controlling the prescaler when LOCO is selected.

The RTC time counter operates on a 128-Hz clock signal as the base clock. Therefore, when LOCO is selected, LOCO is divided by the prescaler to generate a 128-Hz clock signal. Set the frequency comparison value in the RFC[15:0] bits to generate a 128-Hz clock from the LOCO frequency. Before writing to RFC[15:0] after a cold start, write 0x0000 to the RFRH register.

A value from 0x0007 through 0x01FF can be specified as the frequency comparison value. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2. The operating frequency of the peripheral module clock and the LOCO should be such that the peripheral module clock is ≥ LOCO.

Calculation method of frequency comparison value:

RFC[15:0] = (LOCO clock frequency) / 128 - 1

When the LOCO frequency is 32.768 kHz, the RFRL register should be set to 0x00FF.

26.2.25 RFRH : Frequency Register H

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x2A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x

Bit	Symbol	Function	R/W
0	RFC16	Write 0 before writing to the RFRL register after a cold start.	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

Before writing to RFRHL.RFC[15:0] after a cold start, write 0x0000 to the RFRH register.

26.2.26 RADJ : Time Error Adjustment Register

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x2E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PMADJ[1:0]		ADJ[5:0]					
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
5:0	ADJ[5:0]	Adjustment Value These bits specify the adjustment value from the prescaler.	R/W
7:6	PMADJ[1:0]	Plus-Minus 0 0: Do not perform adjustment. 0 1: Adjustment is performed by the addition to the prescaler 1 0: Adjustment is performed by the subtraction from the prescaler 1 1: Setting prohibited.	R/W

Note: S-TYPE3, P-TYPE3

The RADJ register is used in both calendar count mode and binary count mode. Adjustment is performed by the addition to or subtraction from the prescaler or 64-Hz counter. If the Automatic Adjustment Enable (RCR2.AADJE) bit is 0, adjustment is performed when writing to the RADJ. If the RCR2.AADJE bit is 1, adjustment is performed in the interval specified in the Automatic Adjustment Period Select (RCR2.AADJP) bit.

The current adjustment by software (disabling automatic adjustment) may be invalid if the following adjustment value is specified within 320 cycles of the count source after the register setting. To perform adjustment consecutively, wait for 320 cycles or more of the count source after the register setting, then specify the next adjustment value.

RADJ is updated in synchronization with the count source. When RADJ is modified, check that all the bits are updated before continuing with more processing. This register is set to 0x00 by an RTC software reset. The setting of this register is enabled only when the sub-clock oscillator is selected. When LOCO is selected, adjustment is not performed.

ADJ[5:0] bits (Adjustment Value)

The ADJ[5:0] bits specify the adjustment value (number of sub-clock cycles) from the prescaler.

PMADJ[1:0] bits (Plus-Minus)

The PMADJ[1:0] bits select whether the clock is set ahead or back depending on the error-adjustment value set in the ADJ[5:0] bits.

26.2.27 RTCCRn : Time Capture Control Register n (n = 0 to 2)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x40 + 0x02 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TCEN	—	TCNF[1:0]		—	TCST	TCCT[1:0]	
Value after reset:	x	0	x	x	0	x	x	x

Bit	Symbol	Function	R/W
1:0	TCCT[1:0]	Time Capture Control 0 0: Do not detect events 0 1: Detect rising edge 1 0: Detect falling edge 1 1: Detect both edges	R/W
2	TCST	Time Capture Status 0: No event detected 1: Event detected*1	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
5:4	TCNF[1:0]	Time Capture Noise Filter Control 0 0: Turn noise filter off 0 1: Setting prohibited 1 0: Turn noise filter on (count source) 1 1: Turn noise filter on (count source by divided by 32)	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	TCEN	Time Capture Event Input Pin Enable 0: Disable the RTCICn pin as the time capture event input pin 1: Enable the RTCICn pin as the time capture event input pin	R/W

Note: S-TYPE3, P-TYPE3

Note 1. Indicates that an event is detected. Writing 1 to this bit has no effect. Writing 0 sets this bit to 0.

The RTCCRn register is used both in calendar count mode and in binary count mode. RTCCR0, RTCCR1, and RTCCR2 control the RTCIC0, RTCIC1, and RTCIC2 pins, respectively.

RTCCRn is updated in synchronization with the count source. When RTCCRn is modified, check that all the bits except the TCST bit are updated before continuing with additional processing. This register is cleared to 0x00 by an RTC software reset. When RTCICm is used as the time capture pin, VBTICTLR.VCHnIEN (n = 0 to 2) must be set to 1.

TCCT[1:0] bits (Time Capture Control)

The TCCT[1:0] bits control the edge detection of the time capture event input pins, RTCIC0, RTCIC1, and RTCIC2. The detection edge is selectable. The TCCT[1:0] bits must be set while the VBTICTLR.VCHnIEN bit is 1.

TCST bit (Time Capture Status)

The TCST bit indicates that an event on the time capture event input pins, RTCIC0, RTCIC1, and RTCIC2,was detected. When the TCST bit is 0, no event is detected. When the TCST bit is 1, this bit indicates that an event was detected on the associated pin and the capture register is valid. When multiple events are detected, the capture time for the first event is retained.

The event is detected only during count operation (RCR2.START bit = 1). Before reading the capture register, make sure that this bit is set to 1.

Set the TCST bit while the TCCT[1:0] bits are 00b (no event is detected). The TCST bit is set to 0 in synchronization with the count source. When the TCST bit is 0, check that the bit is updated before continuing with additional processing.

TCNF[1:0] bits (Time Capture Noise Filter Control)

The TCNF[1:0] bits control the noise filter of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2).

When the noise filter is on, the count source divided by 1 or divided by 32 is selectable. In this case, when the input level on the time capture event input pin matches three consecutive times at the set sampling period, the input level is determined.

Set the TCNF[1:0] bits while the TCCT[1:0] bits are 00b (no event is detected). When the noise filter is used, set the TCNF[1:0] bits, wait for 3 cycles of the specified sampling period, then set the TCCT[1:0] bits. Set the TCNF[1:0] bits when the VBTICTLR.VCHnIEN bit is 1.

TCEN bit (Time Capture Event Input Pin Enable)

The TCEN bit enables or disables the time capture event input pins RTCIC0, RTCIC1, and RTCIC2.When the functions of the time capture event input pins are multiplexed, set VBTICTLR first. If the TCEN bit is set to 0, also set the TCCT[1:0] bits to 00b.

26.2.28 RSECCPn : Second Capture Register n (n = 0 to 2) (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x52 + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	SEC10[2:0]			SEC1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	SEC1[3:0]	1-Second Capture Capture value for the ones place of seconds.	R
6:4	SEC10[2:0]	10-Second Capture Capture value for the tens place of seconds.	R
7	—	The read value is undefined.	R

Note: S-TYPE3, P-TYPE3

RSECCPn is a read-only register that captures the RSECCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RSECCP0, RSECCP1, and RSECCP2 registers, respectively. This register is cleared to 0x00 by an RTC software reset. Before reading from this register, the time capture event detection should be stopped using the RTCCRn.TCCT[1:0] bits.

26.2.29 RMINCPn : Minute Capture Register n (n = 0 to 2) (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x54 + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	MIN10[2:0]			MIN1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	MIN1[3:0]	1-Minute Capture Capture value for the ones place of minutes.	R
6:4	MIN10[2:0]	10-Minute Capture Capture value for the tens place of minutes.	R
7	—	The read value is undefined.	R

Note: S-TYPE3, P-TYPE3

RMINCPn is a read-only register that captures the RMINCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RMINCP0, RMINCP1, and RMINCP2 registers, respectively.

This register is cleared to 0x00 by an RTC software reset. Before reading from this register, the time capture event detection should be stopped using the RTCCRn.TCCT[1:0] bits.

26.2.30 RHRCPn : Hour Capture Register n (n = 0 to 2) (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x56 + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	PM	HR10[1:0]		HR1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	HR1[3:0]	1-Hour Capture Capture value for the ones place of hours	R
5:4	HR10[1:0]	10-Hour Capture Capture value for the tens place of hours	R
6	PM	PM 0: AM 1: PM	R
7	—	The read value is undefined.	R

Note: S-TYPE3, P-TYPE3

RHRCPn is a read-only register that captures the RHRCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RHRCP0, RHRCP1, and RHRCP2 registers, respectively.

The PM bit is only enabled when the RCR2.HR24 bit is 0 (in 12-hour mode).

This register is cleared to 0x00 by an RTC software reset. Before reading from this register, you must stop the time capture event detection using the RTCCRn.TCCT[1:0] bits.

26.2.31 RDAYCPn : Date Capture Register n (n = 0 to 2) (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x5A + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	DATE10[1:0]		DATE1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	DATE1[3:0]	1-Day Capture Capture value for the ones place of days.	R
5:4	DATE10[1:0]	10-Day Capture Capture value for the tens place of days.	R
7:6	—	The read value is undefined.	R

Note: S-TYPE3, P-TYPE3

RDAYCPn is a read-only register that captures the RDAYCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RDAYCP0, RDAYCP1, and RDAYCP2 registers, respectively.

This register is cleared to 0x00 by an RTC software reset. Before reading from this register, the time capture event detection should be stopped using the RTCCRn.TCCT[1:0] bits.

26.2.32 RMONCPn : Month Capture Register n (n = 0 to 2) (in Calendar Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x5C + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	MON10	MON1[3:0]			
Value after reset:	0	0	0	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	MON1[3:0]	1-Month Capture Capture value for the ones place of months.	R
4	MON10	10-Month Capture Capture value for the tens place of months.	R
7:5	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3

RMONCPn is a read-only register that captures the RMONCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RMONCP0, RMONCP1, and RMONCP2 registers, respectively.

This register is cleared to 0x00 by an RTC software reset. Before reading from this register, the time capture event detection should be stopped using the RTCCRn.TCCT[1:0] bits.

26.2.33 BCNTnCPm : BCNTn Capture Register m (n= 0 to 3, m = 0 to 2) (in Binary Count Mode)

Base address: RTC = 0x4020_2000
RTC_NS = 0x5020_2000

Offset address: 0x52 + 0x10 × m (BCNT0CPm)
0x54 + 0x10 × m (BCNT1CPm)
0x56 + 0x10 × m (BCNT2CPm)
0x5A + 0x10 × m (BCNT3CPm)

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

BCNTnCPm is a read-only register that captures the BCNTn value when a time capture event is detected. BCNT3CPm is assigned to the BCNTCPm[31:24] bits, BCNT2CPm is assigned to the BCNTCPm[23:16] bits, BCNT1CPm is assigned to the BCNTCPm[15:8] bits and BCNT0CPm is assigned to the BCNTCPm[7:0] bits. The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the BCNTnCP0, BCNTnCP1, and BCNTnCP2 registers, respectively.

This register is cleared to 0x00 by an RTC software reset. Before reading from this register, you must stop the time capture event detection using the RTCCRn.TCCT[1:0] bits.

26.3 Operation

26.3.1 Outline of Initial Settings of Registers After Power On

After the power is turned on, perform the initial settings for the clock, count mode, time error adjustment, time, alarm, interrupts, and time capture.

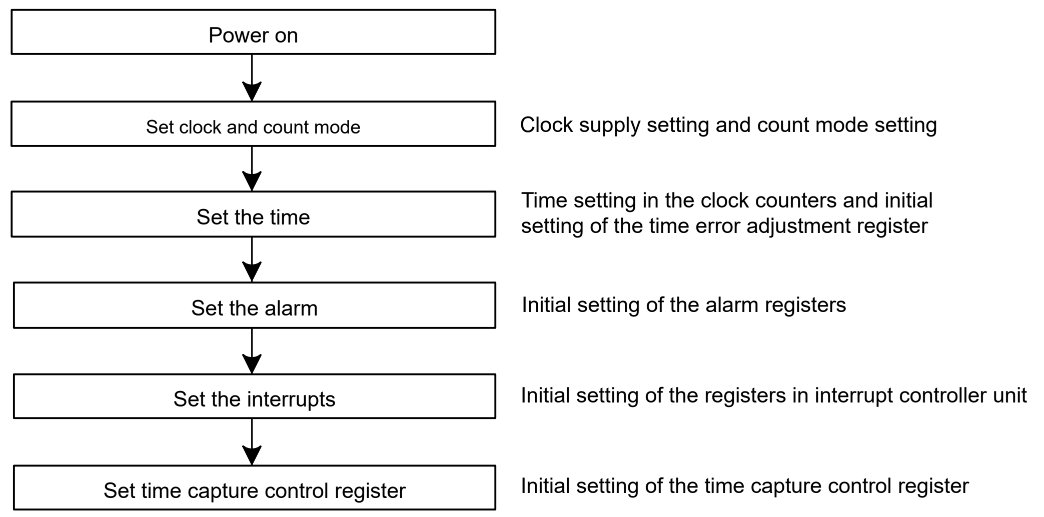
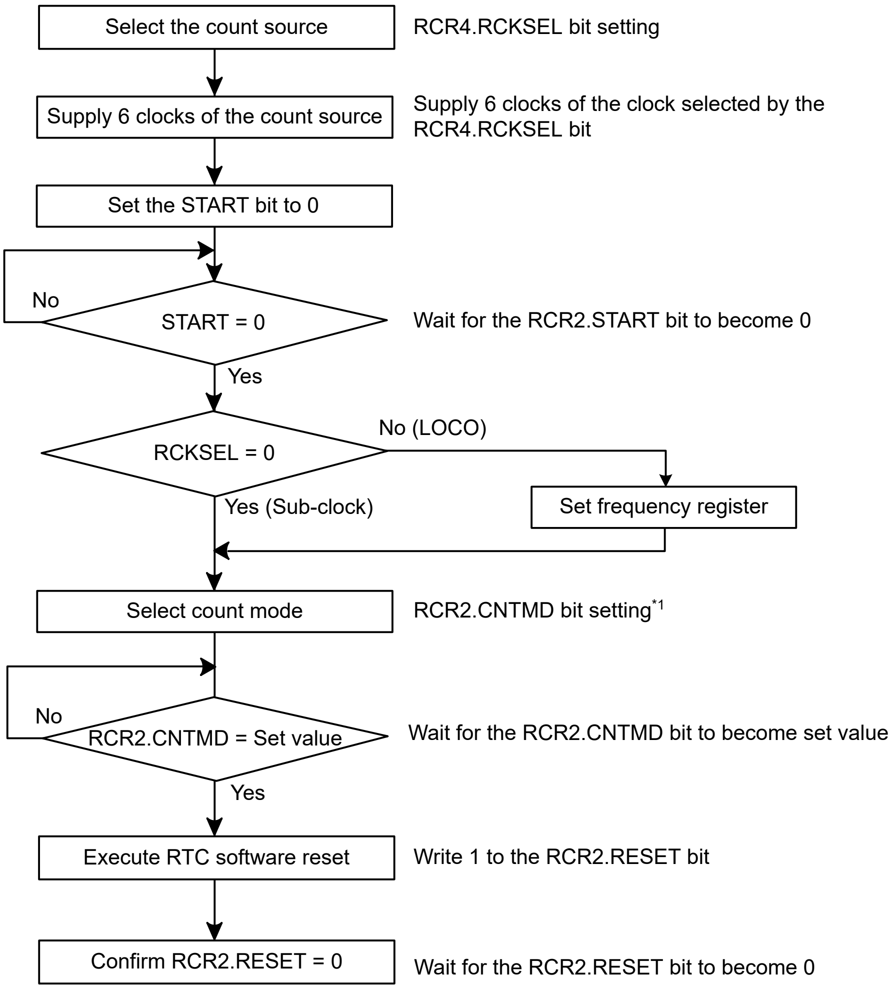


Figure 26.2 Outline of initial settings after a power on

26.3.2 Clock and Count Mode Setting Procedure

Figure 26.3 shows how to set the clock and the count mode.

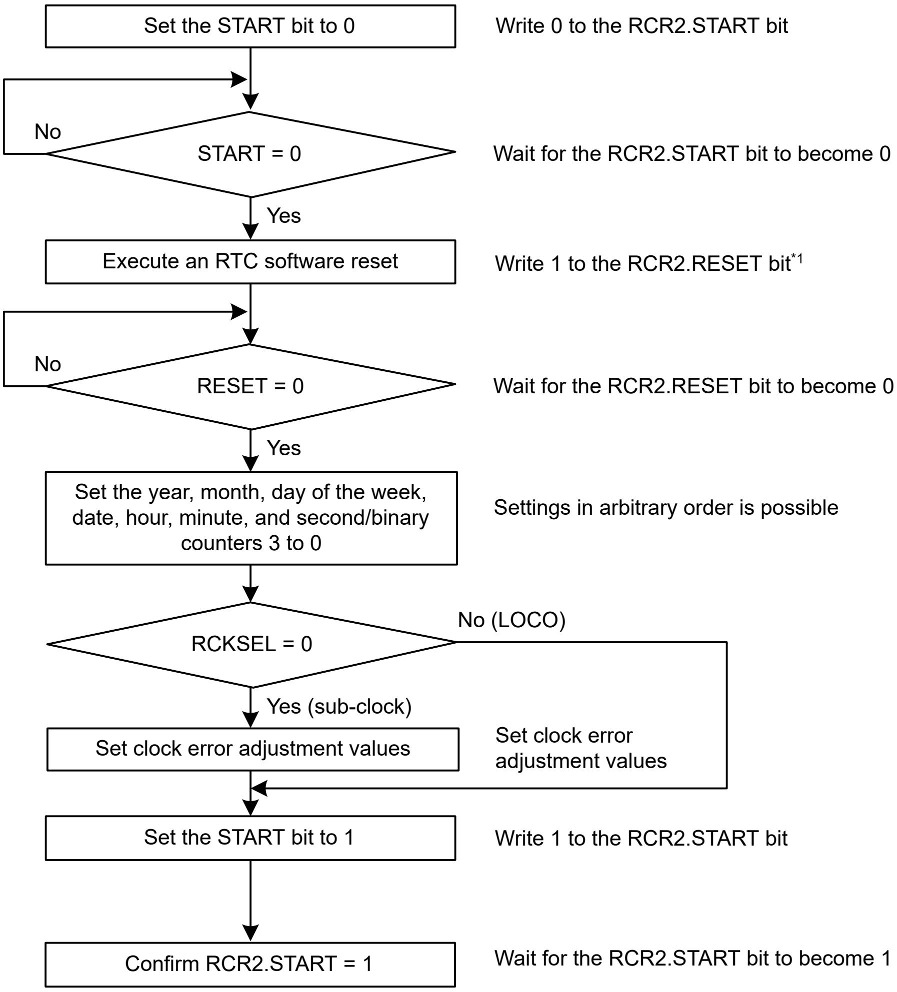


Note 1. This step is not required if the count mode is concurrently set by setting the START bit to 0. A value associated with the count mode setting must be written to the RCR2.CNTMD bit.

Figure 26.3 Clock and count mode setting procedure

26.3.3 Setting the Time

Figure 26.4 shows how to set the time.



Note 1. This step can be omitted in initial settings after power-on because an RTC software reset is executed in the clock setting procedure that is performed during initial settings at power-on.

Figure 26.4 Setting the time

26.3.4 30-Second Adjustment

Figure 26.5 shows how to execute a 30-second adjustment.

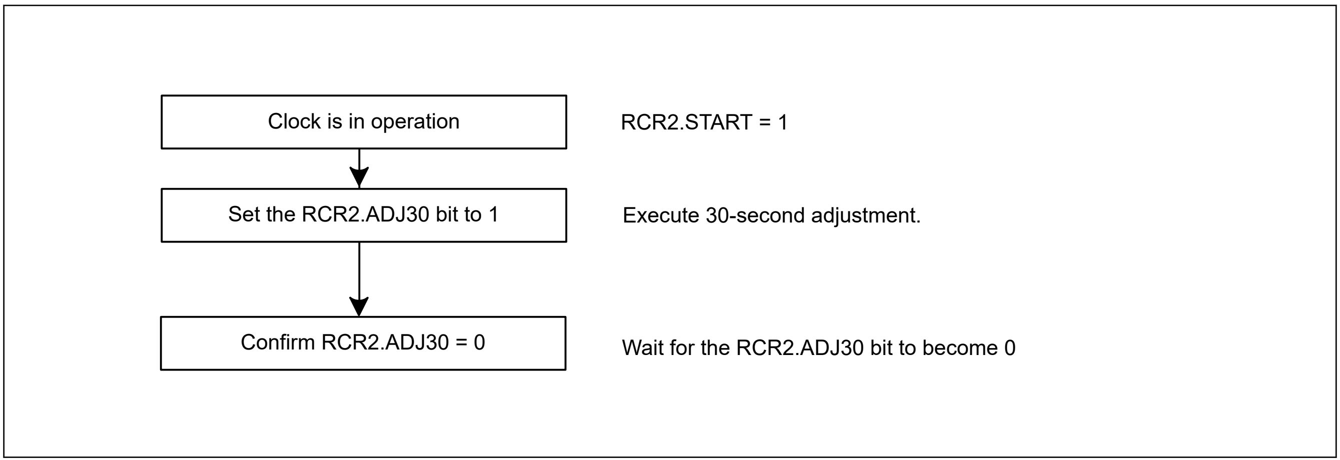
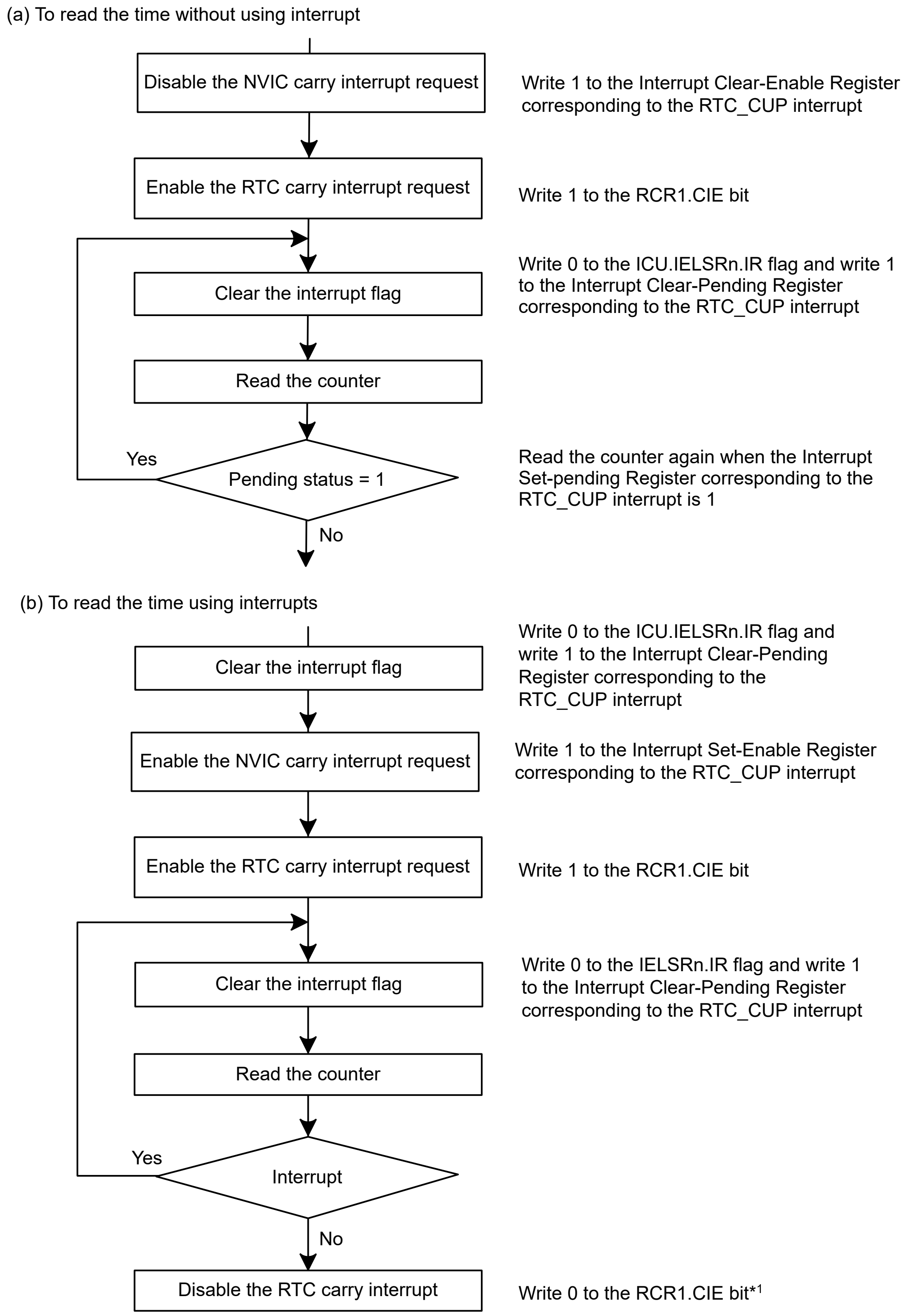


Figure 26.5 30-second adjustment

26.3.5 Reading 64-Hz Counter and Time

Figure 26.6 shows how to read a 64-Hz counter and time.



Note 1. Disable interrupts if required.

Figure 26.6 Reading time

If a carry occurs while the 64-Hz counter and time are read, the correct time is not obtained, therefore they must be read again. The procedure for reading the time without using interrupts is shown in (a) in Figure 26.6, and the procedure using carry interrupts is shown in (b). To keep the program simple, method (a) should be used in most cases.

26.3.6 Alarm Function

Figure 26.7 shows how to use the alarm function.

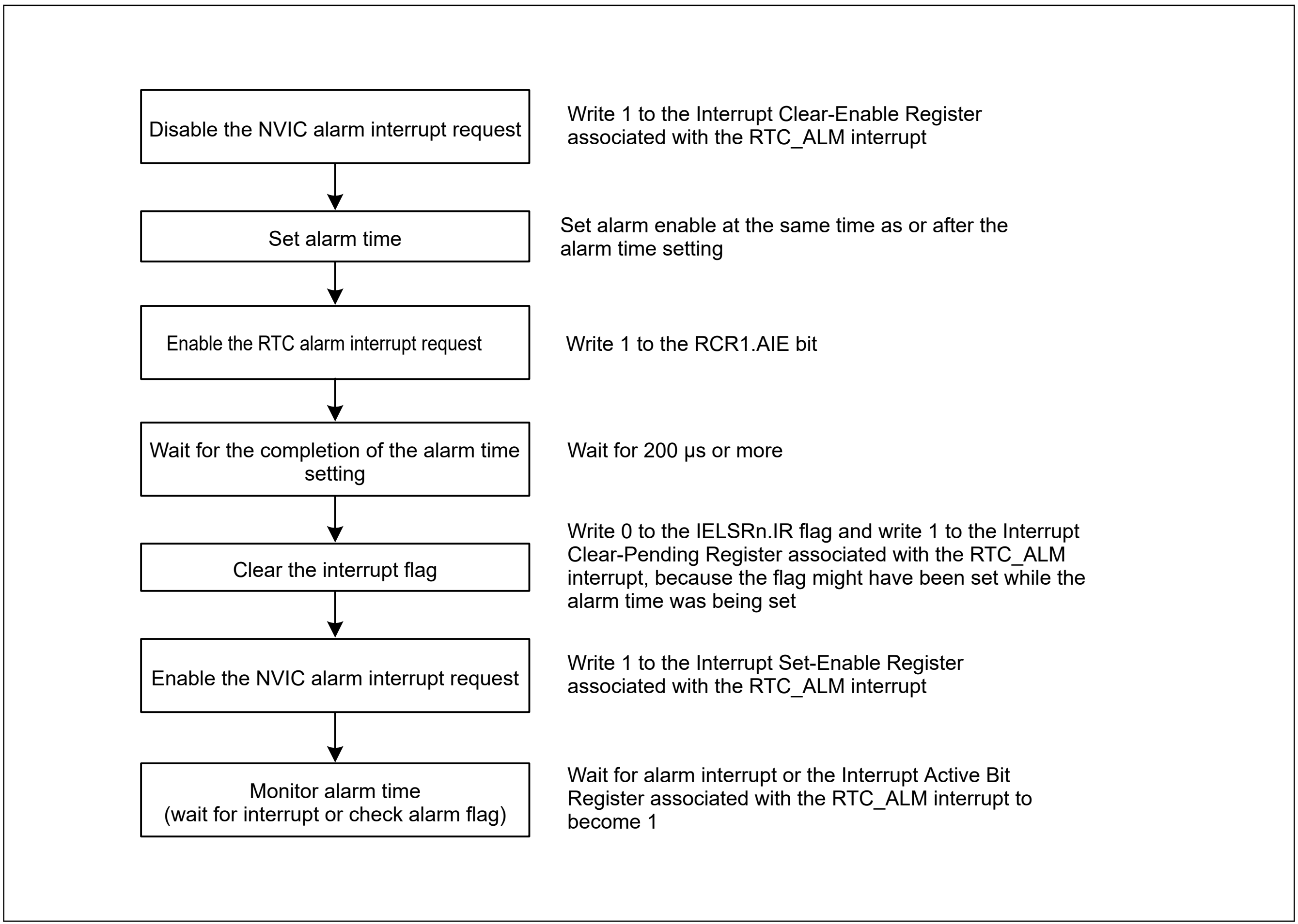


Figure 26.7 Using the alarm function

In calendar count mode, an alarm can be generated by any one of year, month, date, day-of-week, hour, minute or second, or any combination of those. Write 1 to the ENB bit in the alarm registers involved in the alarm setting, and set the alarm time in the lower bits. Write 0 to the ENB bit in registers not involved in the alarm setting.

In binary count mode, an alarm can be generated in any bit combination of 32 bits. Write 1 to the ENB bit of the Alarm Enable register associated with the target bit of the alarm, and set the alarm time in the alarm register. For bits that are not the target of the alarm, write 0 to the ENB bit of the Alarm Enable register.[*1]

For any of the ENB[31:0] bits that are set to 1, the bits in the corresponding positions in the binary counter (BCNT[31:0]) are compared with the values of the corresponding bits in the binary alarm registers[*1]. When all such bits match, the IR flag associated with the RTC_ALM interrupt is set to 1 and the corresponding bits in the Interrupt Set-Pending/Clear-Pending Registers are set to 1. Alarm detection can be confirmed by reading the Interrupt Set-Pending Register associated with the RTC_ALM interrupt, but an interrupt should be used in most cases. If 1 is set in the Interrupt Set-Enable Register associated with the RTC_ALM interrupt, an alarm interrupt is generated in the event of the alarm, enabling the alarm to be detected.

Writing 0 sets the IELSRn.IR flag associated with the RTC_ALM interrupt to 0. If interrupt is enabled, the Interrupt Set-Pending/Clear-Pending Register associated with the RTC_ALM interrupt is cleared automatically after exiting the interrupt handler. Otherwise, write 1 to the Interrupt Clear-Pending Register associated with the RTC_ALM interrupt to clear it.

When the counter and the alarm time match in a low power state, the MCU returns from the low power state.

Note 1. For any bits in the ENB bits that are set to 1, the values in the corresponding positions in the alarm registers from the following registers are compared with the corresponding bits of the counted values.

Counter registers: RSECCNT, RMINCNT, RHRCNT, RWKCNT, RDAYCNT, RMONCNT, RYRCNT

Alarm registers: RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, RYRAREN

26.3.7 Procedure for Disabling Alarm Interrupt

Figure 26.8 shows the procedure for disabling the enabled alarm interrupt request.

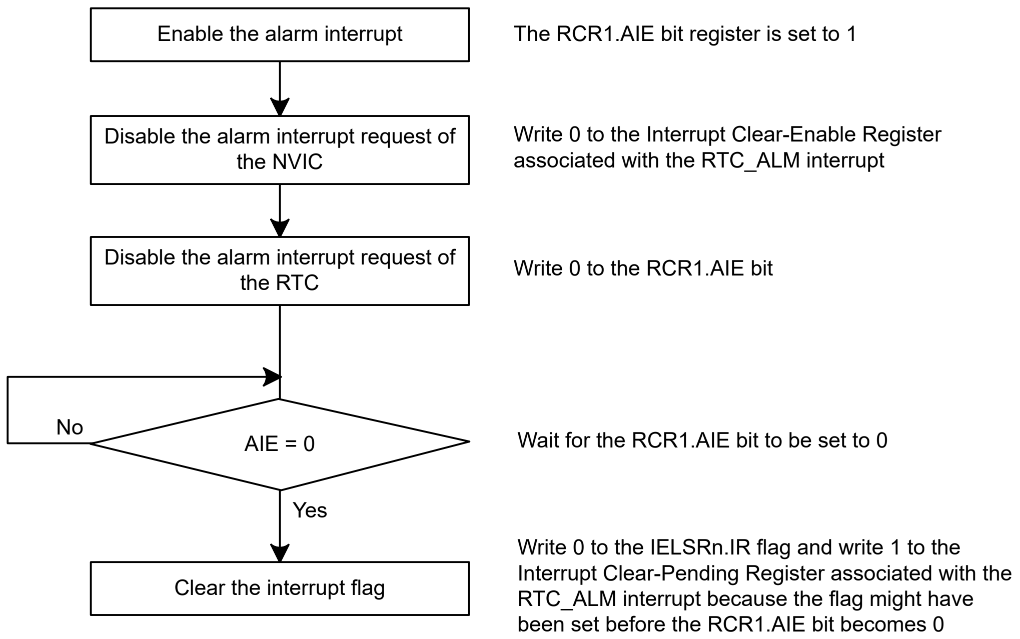


Figure 26.8 Procedure for disabling alarm interrupt request

26.3.8 Time Error Adjustment Function

The time error adjustment function is used to correct errors, running fast or slow, in the time caused by variation in the precision of oscillation by the sub-clock oscillator. Because 32768 cycles of the sub-clock oscillator constitute 1 second of operation when the sub-clock oscillator is selected, the clock runs fast if the sub-clock frequency is high and slow if the sub-clock frequency is low.

The time error adjustment functions include:

- Automatic adjustment
- Adjustment by software

Use the RCR2.AADJE bit to select automatic adjustment or adjustment by software.

26.3.8.1 Automatic Adjustment

Enable automatic adjustment by setting the RCR2.AADJE bit to 1.

Automatic adjustment is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register every time the adjustment period selected by the RCR2.AADJP bit elapses.

(1) Example 1: Sub-clock oscillator running at 32.769 kHz

Adjustment procedure

When the sub-clock oscillator is running at 32.769 kHz, 1 second elapses every 32769 clock cycles. The RTC is meant to run at 32768 clock cycles, so the clock runs fast by 1 clock cycle every second. The time on the clock is fast by 60 clock cycles per minute, so adjustment can take the form of setting the clock back by 60 cycles every minute.

Register settings when RCR2.CNTMD = 0:

- RCR2.AADJP = 0 (adjustment every minute)
- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler)
- RADJ.ADJ[5:0] = 60 (0x3C)

(2) Example 2: Sub-clock oscillator running at 32.766 kHz

Adjustment procedure

When the sub-clock oscillator is running at 32.766 kHz, 1 second elapses every 32766 clock cycles. The RTC is meant to run at 32768 clock cycles, so the clock runs slow by 2 clock cycles every second. The time on the clock is slow by 20 clock cycles every 10 seconds, so adjustment can take the form of setting the clock forward by 20 cycles every 10 seconds.

Register settings when RCR2.CNTMD = 0:

- RCR2.AADJP = 1 (adjustment every 10 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler)
- RADJ.ADJ[5:0] = 20 (0x14)

(3) Example 3: Sub-clock oscillator running at 32.764 kHz

Adjustment procedure

When the sub-clock oscillator is running at 32.764 kHz, 1 second elapses on 32764 clock cycles. Because the RTC operates for 32768 clock cycles as 1 second, the clock is delayed for 4 clock cycles per second. In 8 seconds, the delay is 32 clock cycles, therefore correction can be made by advancing the clock 32 clock cycles every 8 seconds.

Register settings when RCR2.CNTMD = 1:

- RCR2.AADJP = 1 (adjustment every 8 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler)
- RADJ.ADJ[5:0] = 32 (0x20)

26.3.8.2 Adjustment by Software

Enable adjustment by software by setting the RCR2.AADJE bit to 0. Adjustment by software is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register on execution of a write instruction to the RADJ register.

(1) Example 1: Sub-clock oscillator running at 32.769 kHz

Adjustment procedure

When the sub-clock oscillator is running at 32.769 kHz, 1 second elapses every 32769 clock cycles. The RTC is meant to run at 32768 clock cycles, so the clock runs fast by 1 clock cycle every second. The time on the clock is fast by 1 clock cycle per second, so adjustment can take the form of setting the clock back by 1 cycle every second.

Register settings

- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler)
- RADJ.ADJ[5:0] = 1 (0x01)
  This is written to the RADJ register once per 1-second interrupt.

26.3.8.3 Procedure to Change the Mode of Adjustment

When changing the mode of adjustment, change the value of the AADJE bit in RCR2 after setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

To change adjustment by software to automatic adjustment:

1. Set the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
2. Set the RCR2.AADJE bit to 1 (automatic adjustment is enabled).
3. Use the RCR2.AADJP bit to select the period of adjustment.
4. In RADJ, set the PMADJ[1:0] bits for addition or subtraction and the ADJ[5:0] bits to the value for use in time error adjustment.

To change automatic adjustment to adjustment by software:

1. Set the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
2. Set the RCR2.AADJE bit to 0 (adjustment by software is enabled).
3. Proceed with the adjustment by setting the RADJ.PMADJ[1:0] bits for addition or subtraction and the RADJ.ADJ[5:0] bits to the value for use in time error adjustment at the wanted time. After that, the time is adjusted every time a value is written to the RADJ register.

26.3.8.4 Procedure to Stop Adjustment

Stop the adjustment by setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

26.3.9 Capturing the Time

The RTC is capable of storing the month, date, hour, minute and second/binary counters 3 to 0 by detecting an edge of a signal on a time capture event input pin in calendar count mode or binary count mode.

A noise filter can also be used on a time capture event input pin. If the noise filter is enabled, the RTCCRn.TCST bit is set to 1 when the input level on the pin matches three times.

The noise filter can be switched on or off for each of the time capture event input pins.Set VBTICTLR.VCHnIEN (n = 0 to 2) to 1 to enable the RTCICn input. Operation when the noise filter is off is shown in Figure 26.9 and operation when the noise filter is on is shown in Figure 26.10.

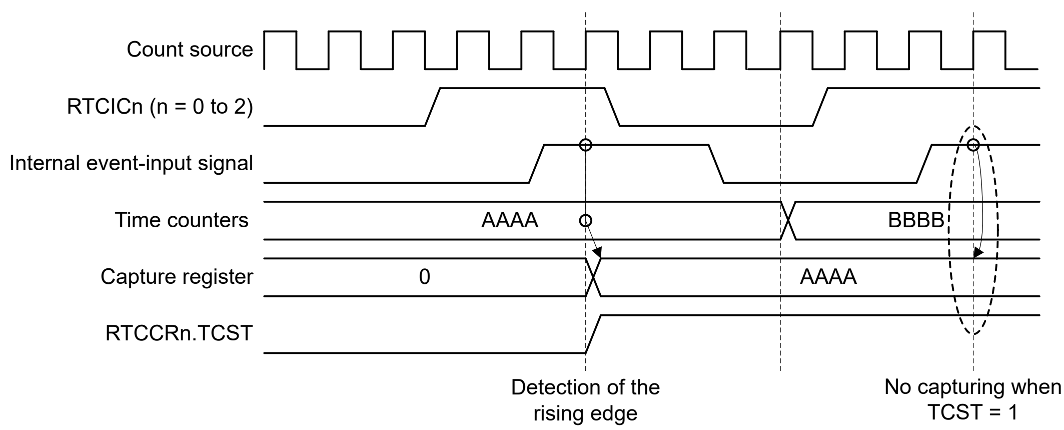


Figure 26.9 Timing of a time capture operation with the noise filter off

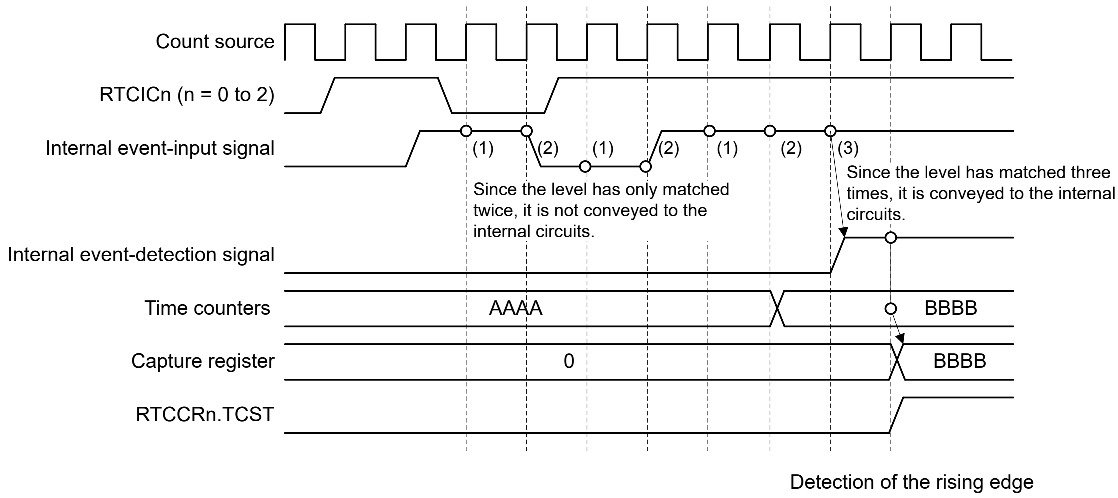


Figure 26.10 Timing of a time capture operation with the noise filter on

26.4 Interrupt Sources

The RTC has three interrupt sources, as listed in Table 26.3.

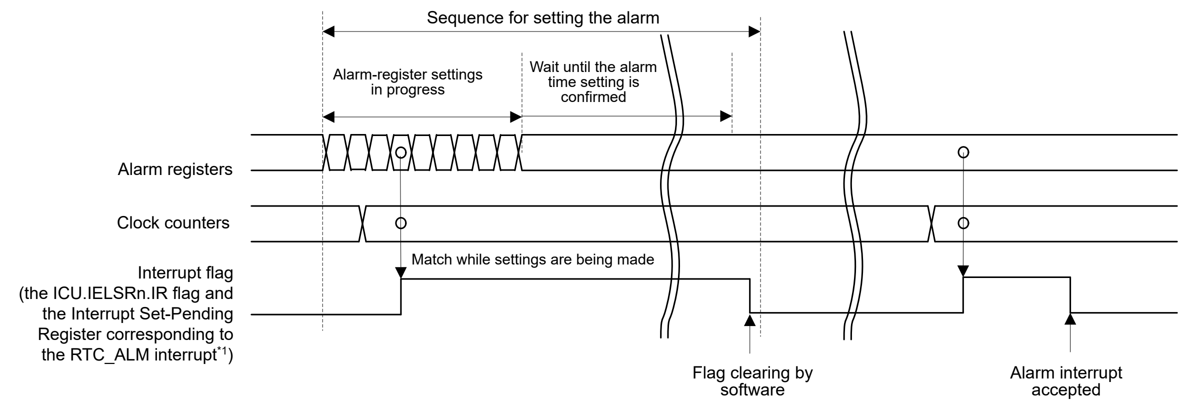
Table 26.3 RTC interrupt sources

Name	Interrupt source
RTC_ALM	Alarm interrupt
RTC_PRD	Periodic interrupt
RTC_CUP	Carry interrupt

(1) Alarm interrupt (RTC_ALM)

This interrupt is generated based on the comparison result between the alarm registers and RTC counters. For details, see section 26.3.6. Alarm Function.

Because there is a possibility that the interrupt flag might be set to 1 when the settings of the alarm registers match the clock counters, wait for the alarm time settings to be confirmed and clear the IELSRn.IR flag and the interrupt Set-Pending Register associated with the RTC_ALM interrupt to 0 again after modifying values of the alarm registers. After the interrupt flag for the alarm interrupt is set to 1 and the state is returned to mismatching of the alarm registers and clock counters, the flag is not 1 again until there is another match or the values of the alarm registers are modified again.



Note 1. See section 14, Interrupt Controller Unit (ICU) for details on the associated interrupt vector number.

Figure 26.11 Timing for the alarm interrupt (RTC_ALM)

(2) Periodic interrupt (RTC_PRD)

This interrupt is generated at intervals of 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, 1/128 second, or 1/256 second. The interrupt interval can be selected in the RCR1.PES[3:0] bits.

(3) Carry interrupt (RTC_CUP)

This interrupt is generated when a carry to the second counter/binary counter 0 occurred or a carry to the R64CNT counter occurred during read access to the 64-Hz counter.

Figure 26.12 shows the timing of the carry interrupt (RTC_CUP).

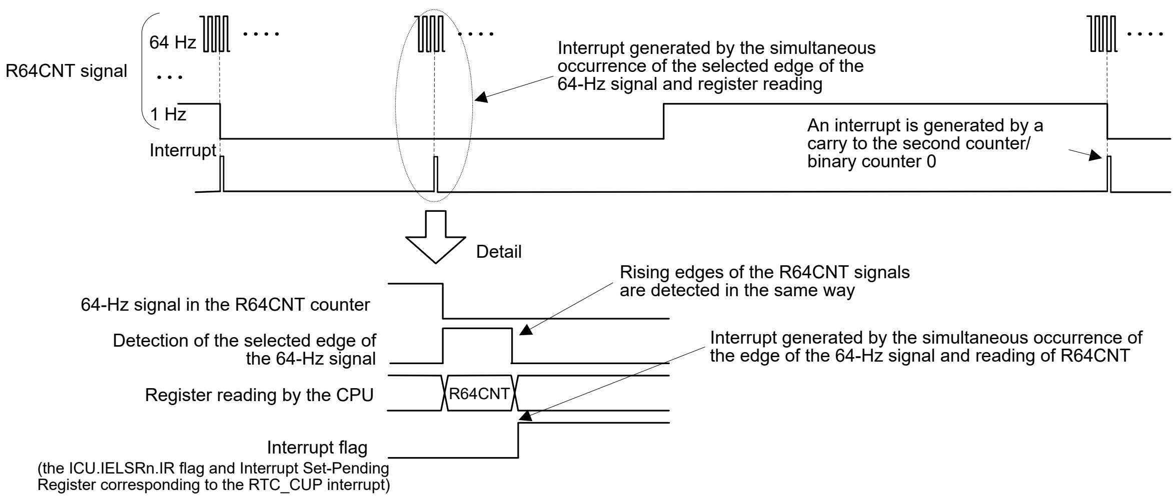


Figure 26.12 Timing for the carry interrupt (RTC_CUP)

26.5 Event Link Output

The RTC generates periodic event output (RTC_PRD) event signal for the ELC that can be used to initiate operations by other modules selected in advance.

The periodic event signal is output at the interval selected from 1/256, 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1, and 2 seconds by setting the RCR1.PES[3:0] bits.

The event generation period immediately after the event generation is selected is not guaranteed.

Note: If event linking from the RTC is used, only set the ELC after setting the RTC, for example initialization and time settings. Setting the RTC after the ELC can lead to output of unexpected event signals.

26.5.1 Interrupt Handling and Event Linking

The RTC has a bit to enable or disable periodic interrupts. An interrupt request signal is output to the CPU when an interrupt source is generated while the associated enable bit is enabled.

In contrast, an event link output signal is sent to other modules as an event signal through the ELC when an interrupt source is generated, regardless of the setting of the associated interrupt enable bit.

Note: Although alarm and periodic interrupts can still be output during Software Standby or Deep Software Standby mode, the periodic event signals for the ELC are not output.

26.6 Usage Notes

26.6.1 Register Writing During Counting

The following registers should not be written to during counting, that is, while the RCR2.START bit is 1:

- RSECCNT/BCNT0
- RMINCNT/BCNT1
- RHRCNT/BCNT2
- RDAYCNT
- RWKCNT/BCNT3
- RMONCNT
- RYRCNT
- RCR1.RTCOS
- RCR2.RTCOE
- RCR2.HR24
- RFRL

The counter should be stopped before writing to any of these registers.

26.6.2 Use of Periodic Interrupts

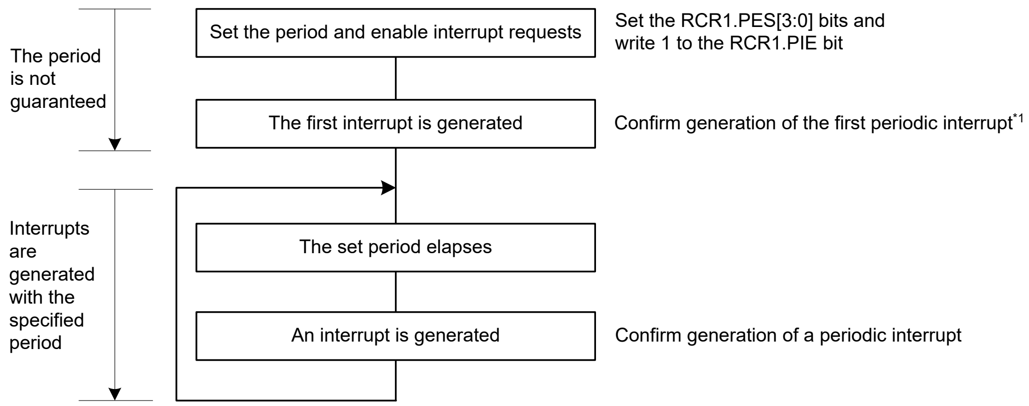
Figure 26.13 shows the procedure for using periodic interrupts.

The generation and period of the periodic interrupt can be changed by setting the RCR1.PES[3:0] bits. However, because the prescaler R64CNT and RSECCNT/BCNT0 are used to generate interrupts, the interrupt period is not guaranteed immediately after setting the RCR1.PES[3:0] bits.

In addition, any of the following operation can affect the interrupt period:

- Stopping/restarting or resetting counter operation
- Reset by RTC software
- 30-second adjustment by changing the RCR2 value

When the time error adjustment function is used, the interrupt generation period after adjustment is added or subtracted based on the adjustment value.



Note 1. When an interrupt generation period changes while the periodic interrupt is used, an interrupt might be generated at the completion of the setting. If the interrupt is generated immediately after the setting, the period is not guaranteed for two interrupts including the current interrupt.

Figure 26.13 Using the periodic interrupt function

26.6.3 RTCOUT (1-Hz/64-Hz) Clock Output

Stopping/restarting or resetting counter operation, reset by RTC software, and the 30-second adjustment by changing the RCR2 value affects the period of RTCOUT (1-Hz/64-Hz) output. When the time error adjustment function is used, the period of RTCOUT (1-Hz/64-Hz) output after adjustment is added or subtracted based on the adjustment value.

26.6.4 Transitions to Low Power Modes After Setting Registers

A transition to a low power state (Software Standby mode, Deep Software Standby mode, or battery backup state) during a write to an RTC register might corrupt the value of the register. After setting the register, confirm that the setting is in place before initiating a transition to a low power state.

26.6.5 Notes on Writing to and Reading from Registers

- When reading a counter register such as the second counter after writing to the counter register, follow the procedure in section 26.3.5. Reading 64-Hz Counter and Time.
- The value written to the count registers, alarm registers, year alarm enable register, bits RCR2.AADJE, AADJP, and HR24, RCR4 register, or frequency register is reflected when fourth read operations are performed after writing.
- The values written to the RCR1.CIE, RCR1.RTCOS, and RCR2.RTCOE bits can be read immediately after writing.
- To read the value from the timer counter after returning from a reset or a period in Software Standby mode, Deep Software Standby mode, or battery backup state, wait for 1/128 second while the clock is operating (RCR2.START bit = 1).
- After a reset is generated, write to the RTC register after 6 cycles of the count source clock have elapsed.

26.6.6 Changing the Count Mode

When changing the count mode (calendar count mode/binary count mode), set the RCR2.START bit to 0, stop the counting operation, then start it again from the initial setting. For details on the initial setting, see section 26.3.1. Outline of Initial Settings of Registers After Power On.

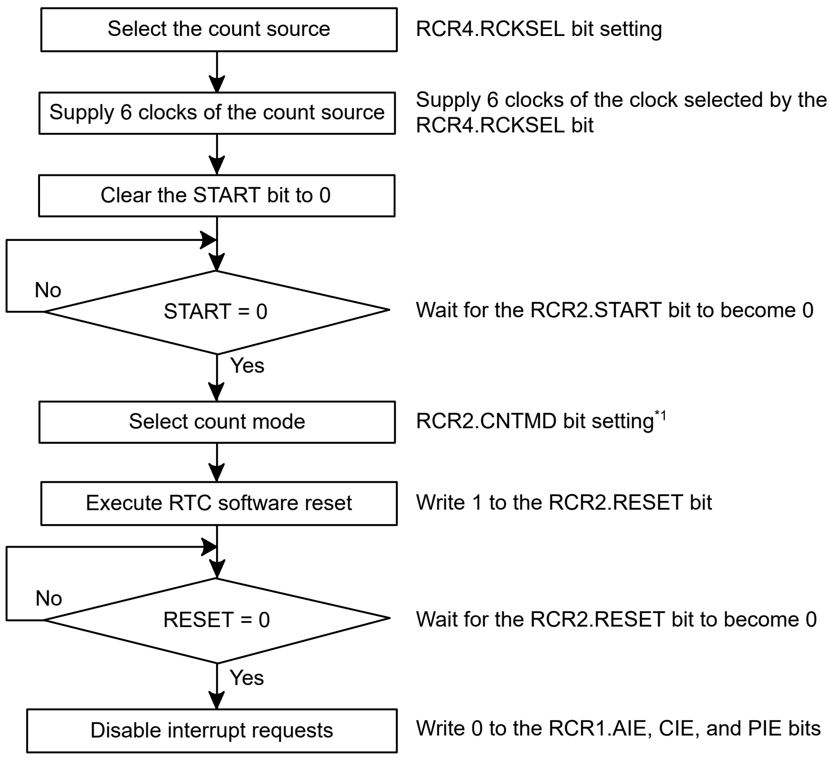
26.6.7 Initialization Procedure When the RTC Is Not to Be Used

Registers in the RTC are not initialized by a reset. Depending on the initial state, the generation of an unintentional interrupt request or operation of the counter might lead to increased power consumption.

For applications that do not require a realtime clock, initialize the registers by following the initialization procedure shown in Figure 26.14.

Alternatively, when the sub-clock oscillator is not used as the system clock source, realtime clock, operating clock of each peripheral modules or FLL function of HOCO, the counter can be stopped by writing 0 (sub-clock oscillator is selected) to the RCR4.RCKSEL bit and stopping the sub-clock oscillator. To stop the sub-clock oscillator, write 1 to the SOSCCR.SOSTP bit.

For details on the setting of the SOSCCR.SOSTP bit, see section 9, Clock Generation Circuit.



Note 1. This step is not necessary if the count mode is concurrently set by setting the START bit to 0.

Figure 26.14 Initialization procedure

26.6.8 When Switching Source Clock

When switching a clock source by changing SCKSCR.CKSEL[2:0], the clock output from the selector stops for 4 cycles of the switched clock. If the RTC periodical interrupt or RTC periodical event output was generated at this time, the interrupt or event is invalid.

26.6.9 When Writing to the RTC Registers

If VCC voltage of the MCU is lower than 1.8V and write access to the RTC registers is performed, read access should be performed at least once after one write access is performed, or for the consecutive write access, leave an interval of 167ns for each access.

27. Watchdog Timer (WDT)

27.1 Overview

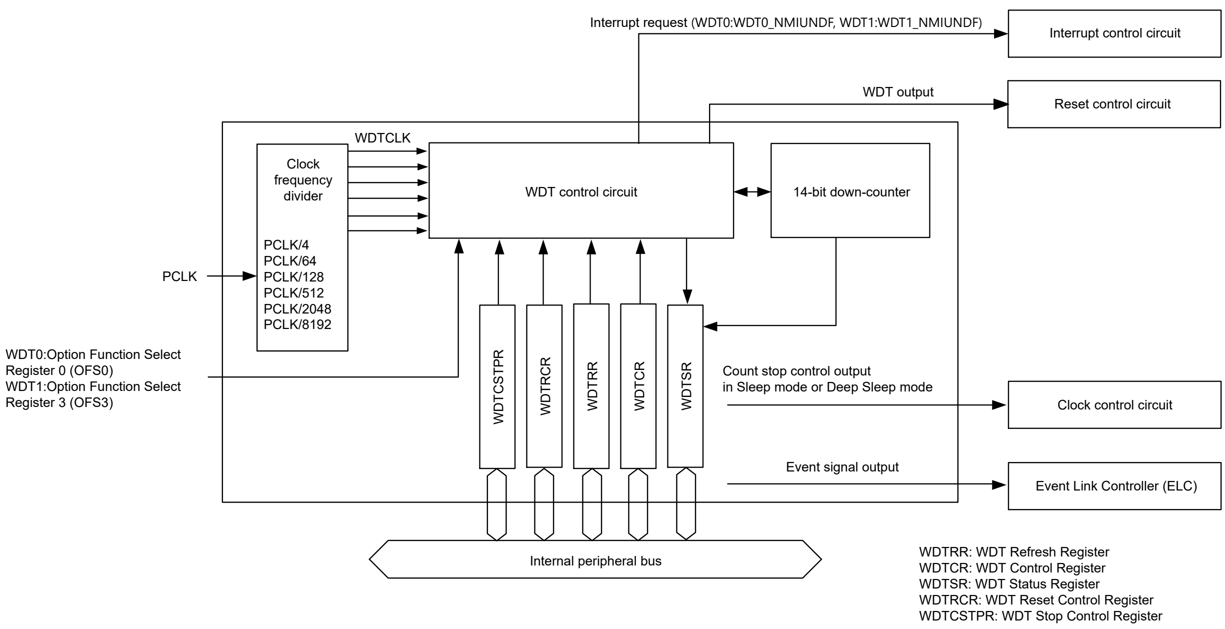
The Watchdog Timer (WDT) is a 14-bit down counter that can be used to reset the MCU when the counter underflows because the system has run out of control and is unable to refresh the WDT. In addition, the WDT can be used to generate a non-maskable interrupt or an underflow interrupt.

Table 27.1 lists the WDT specifications and Figure 27.1 shows a block diagram.

Table 27.1 WDT specifications

Parameter	Specifications
Number of units	2 (WDT0 for CPU0, WDT1 for CPU1)
Count source*1	Peripheral clock (PCLKB)
Clock division ratio	Division by 4, 64, 128, 512, 2048, or 8192
Counter operation	Counting down using a 14-bit down-counter
Condition for starting the counter	● Auto start mode: Counting automatically starts after a reset or after an underflow or refresh error occurs ● Register start mode: Counting is started with a refresh by writing to the WDTRR register ● Only secure developer can select Auto-start mode or Register-start mode
Conditions for stopping the counter	● Reset (the down-counter and other registers return to their initial values) ● A counter underflows or a refresh error is generated
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Watchdog timer reset sources	● Down-counter underflows ● Refreshing outside the refresh-permitted period (refresh error)
Non-maskable interrupt/interrupt sources	● Down-counter underflows ● Refreshing outside the refresh-permitted period (refresh error)
Reading of the counter value	The down-counter value can be read by the WDTSR register
Event link function (output)	● Down-counter underflow event output ● Refresh error event output
Output signal (internal signal)	● Reset output ● Interrupt request output ● Sleep mode or Deep Sleep mode count stop control output
TrustZone Filter	Security and Privilege attribution can be set

Note 1. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ 4 × (the frequency of the count clock source after division).



Note: PCLK = PCLKB

Figure 27.1 WDT block diagram

27.2 Register Descriptions

27.2.1 WDTRR : WDT Refresh Register

Base address: WDTn = 0x4020_2600 + 0x0100 × n (n = 0, 1)
WDTn_NS = 0x5020_2600 + 0x0100 × n (n = 0, 1)

Offset address: 0x00

Bit position:	7							0
Bit field:								
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	n/a	The down-counter is refreshed by writing 0x00 and then writing 0xFF to this register.	R/W

Note: S-TYPE-3, P-TYPE-3

The WDTRR register refreshes the down-counter of the WDT.

The down-counter of the WDT is refreshed by writing 0x00 and then writing 0xFF to WDTRR register (refresh operation) within the refresh-permitted period.

After the down-counter is refreshed, it starts counting down from the value selected by setting the WDT Timeout Period Select bits (OFS0.WDT0TOPS[1:0] or OFS3.WDT1TOPS[1:0]) in the Option Function Select Register m in auto start mode. In register start mode, counting down starts from the value selected by setting the Timeout Period Select bits (WDTCR.TOPS[1:0]) in the WDT Control Register.

When 0x00 is written, the read value is 0x00. When a value other than 0x00 is written, the read value is 0xFF. For details of the refresh operation, see section 27.3.3. Refresh Operation.

27.2.2 WDTCR : WDT Control Register

Base address: WDTn = 0x4020_2600 + 0x0100 × n (n = 0, 1)
WDTn_NS = 0x5020_2600 + 0x0100 × n (n = 0, 1)

Offset address: 0x02

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

Bit	Symbol	Function	R/W
1:0	TOPS[1:0]	Timeout Period Select 0 0: 1024 cycles (0x03FF) 0 1: 4096 cycles (0x0FFF) 1 0: 8192 cycles (0x1FFF) 1 1: 16384 cycles (0x3FFF)	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
7:4	CKS[3:0]	Clock Division Ratio Select 0 0 0 1: PCLKB/4 0 1 0 0: PCLKB/64 1 1 1 1: PCLKB/128 0 1 1 0: PCLKB/512 0 1 1 1: PCLKB/2048 1 0 0 0: PCLKB/8192 Others: Setting prohibited	R/W
9:8	RPES[1:0]	Window End Position Select 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (do not specify window end position).	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
13:12	RPSS[1:0]	Window Start Position Select 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (do not specify window start position).	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The WDTCR register is used to set the clock division ratio, and window start and end positions for refresh, and the timeout period until the down-counter underflows in register start mode.

Some constraints apply to writes to the WDTCR register. For details, see section 27.3.2. Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.

In auto start mode, the settings in the WDTCR register are disabled, and the settings in the Option Function Select Register m (OFSm) are enabled (m = 0, 3). The settings for the WDTCR register can also be made in the OFSm register (m = 0, 3). For details, see section 27.3.8. Association Between Option Function Select Register m (OFSm) and WDT Registers (m = 0, 3).

TOPS[1:0] bits (Timeout Period Select)

The TOPS[1:0] bits select the timeout period, the period until the down-counter underflows, from 1024, 4096, 8192, and 16384 cycles, taking the divided clock specified in the CKS[3:0] bits as 1 cycle. After the down-counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the number of PCLKB cycles until the counter underflows.

Table 27.2 lists the relationship between the CKS[3:0] and TOPS[1:0] bit settings, the timeout period, and the number of PCLKB cycles.

Table 27.2 Timeout period settings

CKS[3:0] bits	TOPS[1:0] bits	Clock division ratio	Timeout period (number of cycles)	PCLKB clock cycles
0x1	00b	PCLKB/4	1024	4096
	01b		4096	16384
	10b		8192	32768
	11b		16384	65536
0x4	00b	PCLKB/64	1024	65536
	01b		4096	262144
	10b		8192	524288
	11b		16384	1048576
0xF	00b	PCLKB/128	1024	131072
	01b		4096	524288
	10b		8192	1048576
	11b		16384	2097152
0x6	00b	PCLKB/512	1024	524288
	01b		4096	2097152
	10b		8192	4194304
	11b		16384	8388608
0x7	00b	PCLKB/2048	1024	2097152
	01b		4096	8388608
	10b		8192	16777216
	11b		16384	33554432
0x8	00b	PCLKB/8192	1024	8388608
	01b		4096	33554432
	10b		8192	67108864
	11b		16384	134217728

CKS[3:0] bits (Clock Division Ratio Select)

The CKS[3:0] bits specify the division ratio of the clock used for the down-counter. The division ratio can be selected from the PCLKB divided by 4, 64, 128, 512, 2048, and 8192. Combined with the TOPS[1:0] bit setting, this allows the WDT to be configured to a count period between 4096 and 134217728 PCLKB clock cycles.

RPES[1:0] bits (Window End Position Select)

The RPES[1:0] bits specify the window end position that indicates the refresh-permitted period. 75%, 50%, 25%, or 0% of the timeout period can be selected for the window end position. Set the window end position to a value less than the value for the window start position (window start position > window end position). If the window start position is set to a value less than or equal to the window end position, the window start position setting is enabled, and the window end position is set to 0%.

RPSS[1:0] bits (Window Start Position Select)

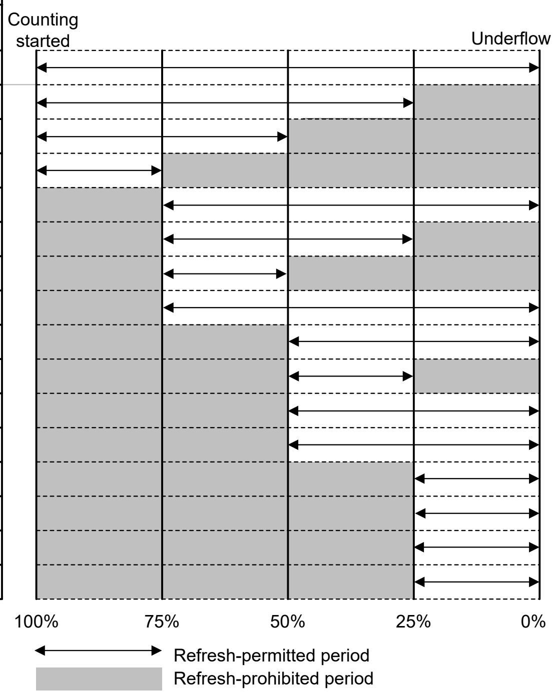
The RPSS[1:0] bits specify the window start position that indicates the refresh-permitted period. 100%, 75%, 50%, or 25% of the timeout period can be selected for the window start position. Set the window start position to a value greater than the value for the window end position. If the window start position is set to a value less than or equal to the window end position, the window start position setting is enabled, and the window end position is set to 0%.

Table 27.3 lists the counter values for the window start and end positions, and Figure 27.2 shows the refresh-permitted period set in the RPSS[1:0], RPES[1:0], and TOPS[1:0] bits.

Table 27.3 Relationship between the timeout period and window start and end counter values

TOPS[1:0]	Timeout period		Window start and end counter value			
	Cycles	Counter value	100%	75%	50%	25%
00b	1024	0x03FF	0x03FF	0x02FF	0x01FF	0x00FF
01b	4096	0x0FFF	0x0FFF	0x0BFF	0x07FF	0x03FF
10b	8192	0x1FFF	0x1FFF	0x17FF	0x0FFF	0x07FF
11b	16384	0x3FFF	0x3FFF	0x2FFF	0x1FFF	0x0FFF

RPSS[1:0] bits		RPES[1:0] bits		Window	
b13	b12	b9	b8	Start (%)	End (%)
1	1	1	1	100	0
		1	0		25
		0	1		50
		0	0		75
1	0	1	1	75	0
		1	0		25
		0	1		50
		0	0		75
0	1	1	1	50	0
		1	0		25
		0	1		50
		0	0		75
0	0	1	1	25	0
		1	0		25
		0	1		50
		0	0		75



Note: If window end position setting ≥ window start position setting, the window end position setting is set to 0%.

Figure 27.2 RPSS[1:0] and RPES[1:0] bits setting and refresh-permitted period

27.2.3 WDTSR : WDT Status Register

Base address: WDTn = 0x4020_2600 + 0x0100 × n (n = 0, 1)
WDTn_NS = 0x5020_2600 + 0x0100 × n (n = 0, 1)

Offset address: 0x04

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

Bit	Symbol	Function	R/W
13:0	CNTVAL[13:0]	Down-Counter Value Value counted by the down-counter	R
14	UNDFF	Underflow Flag 0: No underflow occurred 1: Underflow occurred	R/W*1

Bit	Symbol	Function	R/W
15	REFEF	Refresh Error Flag 0: No refresh error occurred 1: Refresh error occurred	R/W*1

Note: S-TYPE-3, P-TYPE-3
Note 1. Only 0 can be written to clear the flag.

The WDTSR register indicates the counter value of the down-counter and the status of whether an underflow or refresh error occurred in the down-counter.

CNTVAL[13:0] bits (Down-Counter Value)

Read the CNTVAL[13:0] bits to confirm the value of the down-counter. The read value might differ from the actual count by 1.

UNDFF flag (Underflow Flag)

Read the UNDFF flag to confirm whether an underflow occurred in the counter. A value of 1 indicates that the down counter underflowed. Write 0 to the flag to set the value to 0. Writing 1 has no effect.

Clearing of the UNDFF flag takes (N + 1) PCLKB cycles. In addition, clearing of the flag is ignored for (N + 2) PCLKB cycles after an underflow. N is specified as follows:

Auto start mode:

- When OFSm.WDTnCKS[3:0] = 0x1, N = 4
- When OFSm.WDTnCKS[3:0] = 0x4, N = 64
- When OFSm.WDTnCKS[3:0] = 0xF, N = 128
- When OFSm.WDTnCKS[3:0] = 0x6, N = 512
- When OFSm.WDTnCKS[3:0] = 0x7, N = 2048
- When OFSm.WDTnCKS[3:0] = 0x8, N = 8192

Register start mode:

- When WDTCR.CKS[3:0] = 0x1, N = 4
- When WDTCR.CKS[3:0] = 0x4, N = 64
- When WDTCR.CKS[3:0] = 0xF, N = 128
- When WDTCR.CKS[3:0] = 0x6, N = 512
- When WDTCR.CKS[3:0] = 0x7, N = 2048
- When WDTCR.CKS[3:0] = 0x8, N = 8192

REFEF flag (Refresh Error Flag)

Read the REFEF flag to confirm whether a refresh error occurred, indicating that a refresh operation was performed during a prohibited period. A value of 1 indicates that a refresh error occurred. Write 0 to the flag to set the value to 0. Writing 1 has no effect.

Clearing of the REFEF flag takes (N + 1) PCLKB cycles. In addition, clearing of the flag is ignored for (N + 2) PCLKB cycles after a refresh error. N is specified as follows:

Auto start mode:

- When OFSm.WDTnCKS[3:0] = 0x1, N = 4
- When OFSm.WDTnCKS[3:0] = 0x4, N = 64
- When OFSm.WDTnCKS[3:0] = 0xF, N = 128
- When OFSm.WDTnCKS[3:0] = 0x6, N = 512
- When OFSm.WDTnCKS[3:0] = 0x7, N = 2048
- When OFSm.WDTnCKS[3:0] = 0x8, N = 8192

Register start mode:

- When OFSm.WDTnCKS[3:0] = 0x1, N = 4
- When OFSm.WDTnCKS[3:0] = 0x4, N = 64
- When OFSm.WDTnCKS[3:0] = 0xF, N = 128
- When OFSm.WDTnCKS[3:0] = 0x6, N = 512
- When OFSm.WDTnCKS[3:0] = 0x7, N = 2048
- When OFSm.WDTnCKS[3:0] = 0x8, N = 8192

27.2.4 WDTRCR : WDT Reset Control Register

Base address: WDTn = 0x4020_2600 + 0x0100 × n (n = 0, 1)
WDTn_NS = 0x5020_2600 + 0x0100 × n (n = 0, 1)

Offset address: 0x06

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

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	RSTIRQS	WDT Behavior Selection 0: Interrupt 1: Reset	R/W

Note: S-TYPE-3, P-TYPE-3

The WDTRCR register controls reset output by a WDT down-counter underflow or interrupt request output.

Some constraints apply to writes to the WDTRCR register. For details, see section 27.3.2. Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.

In auto start mode, the WDTRCR register settings are disabled, and the settings in the Option Function Select register m (OFSm) are enabled (m = 0, 3). The settings for the WDTRCR register can also be made for the OFSm register (m = 0, 3). For details, see section 27.3.8. Association Between Option Function Select Register m (OFSm) and WDT Registers (m = 0, 3).

27.2.5 WDTCSTPR : WDT Count Stop Control Register

Base address: WDTn = 0x4020_2600 + 0x0100 × n (n = 0, 1)
WDTn_NS = 0x5020_2600 + 0x0100 × n (n = 0, 1)

Offset address: 0x08

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

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	SLCSTP	Sleep-Mode Count Stop Control Register 0: Disable count stop 1: Stop count on transition to Sleep mode or Deep Sleep mode	R/W

Note: S-TYPE-3, P-TYPE-3

The WDTCSTPR register controls whether to stop the WDT counter in Sleep mode or Deep Sleep mode. Some constraints apply to writes to the WDTCSTPR register. For details, see section 27.3.2. Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.

In auto start mode, the WDTCSTPR register settings are disabled, and the settings in the Option Function Select register m (OFSm) are enabled (m = 0, 3). The settings for the WDTCSTPR register can also be made for the OFSm register (m = 0, 3). For details, see section 27.3.8. Association Between Option Function Select Register m (OFSm) and WDT Registers (m = 0, 3).

SLCSTP bit (Sleep-Mode Count Stop Control Register)

The SLCSTP bit of WDT0 selects whether to stop counting on CPU0 transition to Sleep mode or Deep Sleep mode.

The SLCSTP bit of WDT1 selects whether to stop counting on CPU1 transition to Sleep mode or Deep Sleep mode.

27.2.6 Option Function Select Register

For information on the OFS0 register and the OFS3 register, see section x, Option-Setting Memory.

27.3 Operation

27.3.1 Count Operation in each Start Mode

The WDT has two start modes:

- Auto start mode, in which counting automatically starts after a release from the reset state
- Register start mode, in which counting starts with a refresh by writing to the register.

In auto start mode, counting automatically starts after a release from the reset state according to the settings in the Option Function Select register OFS0 or OFS3/OFS3_SEC in the flash.

In register start mode, counting starts with a refresh by writing to the WDTRR register after the respective registers are set after a release from the reset state.

Select auto start mode or register start mode by setting the Select bit (OFS0.WDT0STRT or OFS3.WDT1STRT).

When the auto start mode is selected, the settings in the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSTPR) are disabled while the settings in the OFSm register are enabled (m = 0, 3).

When the register start mode is selected, the setting for the OFS0 or OFS3/OFS3_SEC register is disabled while the settings for the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSTPR) are enabled.

27.3.1.1 Register Start Mode

When the WDT Start Mode Select bit (OFS0.WDT0STRT or OFS3.WDT1STRT) is 1, register start mode is selected and the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSTPR) are enabled.

After the reset state is released, set the following:

- Clock division ratio in the WDTCR register
- Window start and end positions in the WDTCR register
- Timeout period in the WDTCR register
- Reset output or interrupt request output in the WDTRCR register
- Counter stop control during transitions to Sleep mode or Deep Sleep mode in the WDTCSTPR register

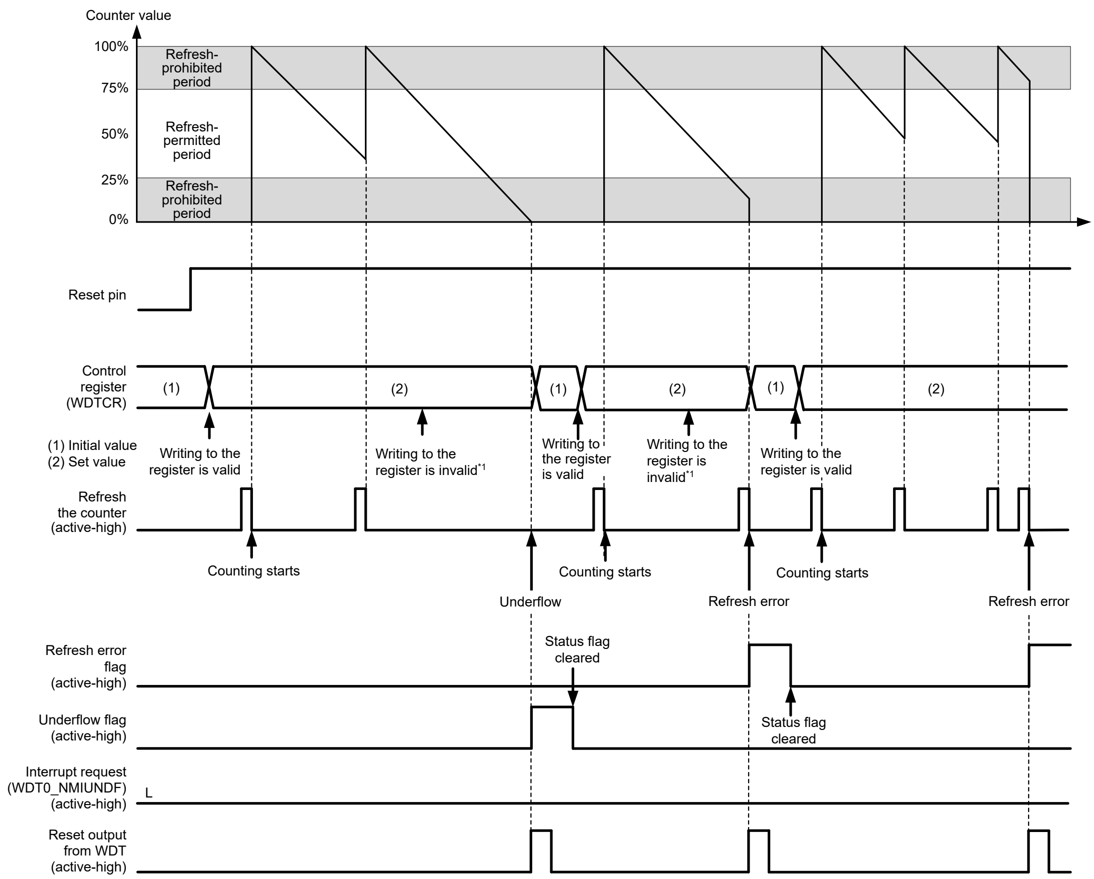
The WDT refresh register (WDTRR) refreshes the down counter. As a result, it starts counting down at the value set by the timeout period selection bit (WDTCR.TOPS[1:0]).

Thereafter, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is reset each time the counter is refreshed and counting down continues. The WDT does not output the reset signal or non-maskable interrupt request/interrupt request as long as counting continues. However, if the down-counter underflows because the down-counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the WDT outputs the reset signal or a non-maskable interrupt request/interrupt request (WDTn_NMIUNDF) (n = 0, 1). Reset output or interrupt request output can be selected in the WDT Reset Interrupt Request

Select bit (WDTRCR.RSTIRQS). The interrupt enabled for operating the NMI can be selected in the WDT Underflow/ Refresh Error Interrupt Enable bit (NMIER.WDTEN).

Figure 27.3 shows an example of operation under the following conditions:

- Register start mode (OFS0.WDT0STRT or OFS3.WDT1STRT = 1)
- WDT reset interrupt request selection (WDTRCR.RSTIRQS = 1)
- The window start position is 75% (WDTCR.RPSS[1:0] = 10b)
- The window end position is 25% (WDTCR.RPES[1:0] = 10b)



Note: Refreshing the counter requires up to 4 count signal cycles from refresh request to execution.

Note 1. See section 27.3.2. Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.
Reset pin = RES

Figure 27.3 Operation example in register start mode

27.3.1.2 Auto Start Mode

When the WDT Start Mode Select bit (OFS0.WDT0STRT or OFS3.WDT1STRT = 1) in the Option Function Select Register m (OFSm) is 0, auto start mode is selected, the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSTPR) are disabled, and the settings in the OFS0 or OFS3 register are enabled.

Within the reset state, the setting values for the following in the Option Function Select Register m (OFS0 or OFS3) are set in the WDT registers:

- Clock division ratio

- Window start and end positions
- Timeout period
- Reset output or interrupt request
- Counter stop control during transition to Sleep mode or Deep Sleep mode

When the reset state is released, the down-counter automatically starts counting down from the value set in the WDT Timeout Period Select bits (OFS0.WDT0STRT[1:0] or OFS3.WDT1STRT[1:0]).

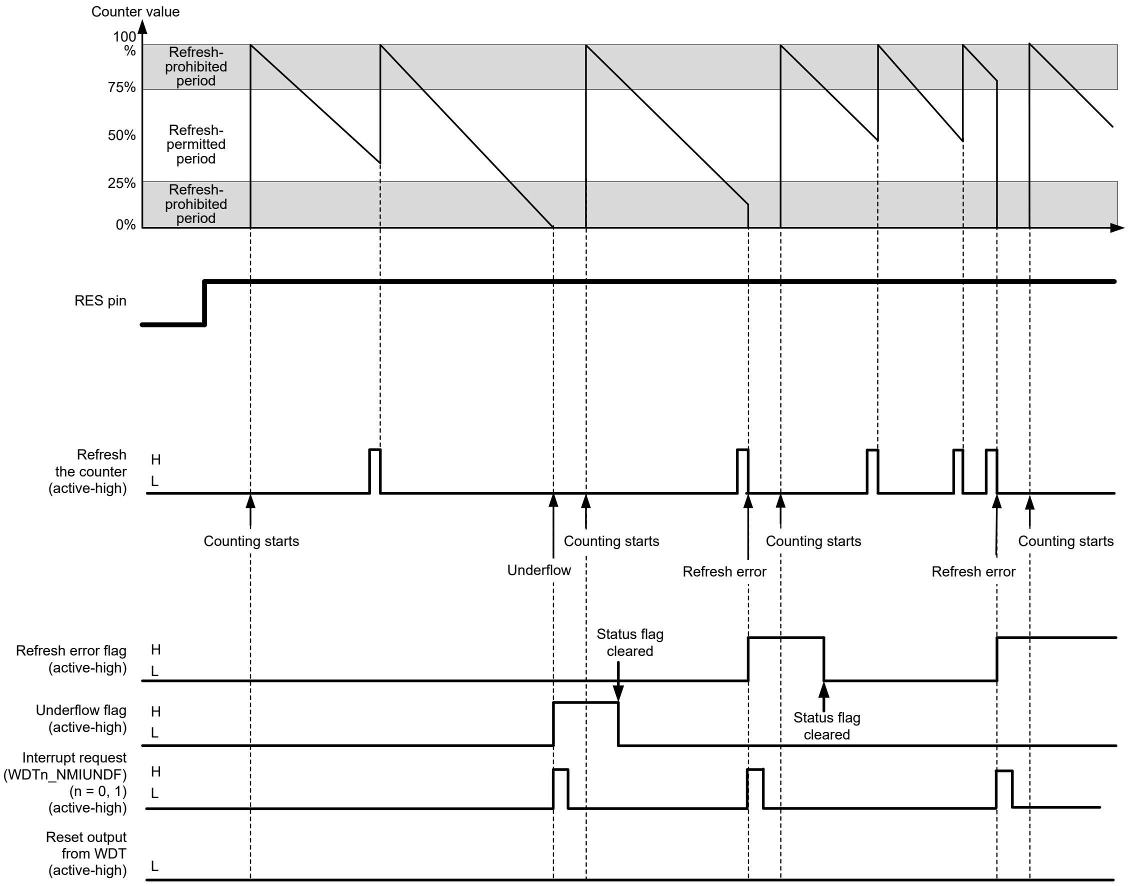
Thereafter, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is reset each time the counter is refreshed and counting down continues. The WDT does not output the reset signal or non-maskable interrupt request/interrupt request (WDT_NMIUNDF) as long as the counting continues. However, if the down-counter underflows because refreshing of the down-counter is not possible due to a runaway program or if a refresh error occurs due to refreshing outside the refresh-permitted period, the WDT outputs the reset signal or non-maskable interrupt request/interrupt request (WDTn_NMIUNDF) (n = 0, 1).

After the reset signal or non-maskable interrupt request/interrupt request is generated, the counter reloads the timeout period after counting for 1 cycle. The value of the timeout period is set in the down-counter and counting restarts.

Reset output or interrupt request output can be selected by setting the WDT Reset Interrupt Request Select bit (OFS0.WDT0STRT or OFS3.WDT1STRT = 1). Non-maskable interrupt request or interrupt request can be selected in the WDT Underflow/ Refresh Error Interrupt Enable bit (NMIER.WDTEN).

Figure 27.4 shows an example of operation (non-maskable interrupt) under the following conditions:

- Auto start mode (OFSm.WDTnSTRT = 0) ((m,n) = (0,0), (3,1))
- Non-maskable interrupt request output is enabled (OFSm.WDTnRSTIRQS = 0) ((m,n) = (0,0), (3,1))
- The window start position is 75% (OFSm.WDTnRPSS[1:0] = 10b) ((m,n) = (0,0), (3,1))
- The window end position is 25% (OFSm.WDTnRPES[1:0] = 10b) ((m,n) = (0,0), (3,1))



Note: Refreshing the counter requires up to 4 count signal cycles from refresh request to execution.
Note: Reset pin = RES

Figure 27.4 Operation example in auto start mode

27.3.2 Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers

Writing to the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), or WDT Count Stop Control Register (WDTCSTPR) is possible once each between the release from the reset state and the first refresh operation.

After a refresh (counting starts) or a write to WDTCR, WDTRCR or WDTCSTPR register, the protection signal in the WDT becomes 1 to protect WDTCR, WDTRCR and WDTCSTPR register against subsequent write attempts. This protection is released by the reset source of the WDT. With other reset sources, the protection is not released.

Figure 27.5 shows control waveforms produced in response to writing to the WDTCR.

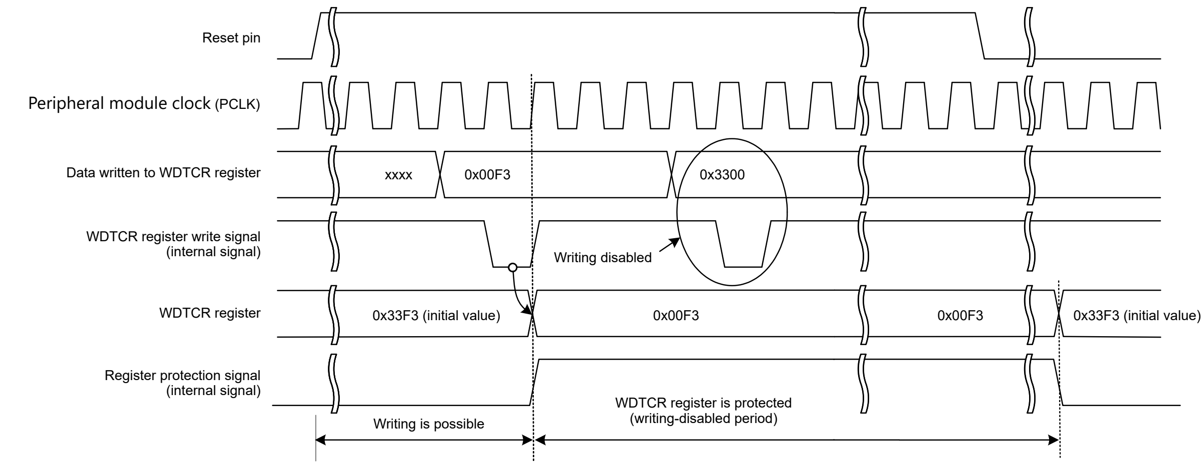


Figure 27.5 Control waveforms produced in response to writes to the WDTCR register

27.3.3 Refresh Operation

To refresh the down counter and start the counting operation, write to the WDT Refresh Register (WDTRR) in the order of values from 0x00 to 0xFF. If a value other than 0xFF is written after 0x00, the down-counter is not refreshed. If an invalid value is written, refreshing is performed normally by writing to the WDTRR register in the order of values from 0x00 to 0xFF.

Correct refreshing is also performed when a register other than WDTRR is accessed or WDTRR is read between writing 0x00 and writing 0xFF to WDTRR. Writes to refresh the counter must be made within the refresh-permitted period, and this is determined by the 0xFF write. For this reason, correct refreshing is performed even when 0x00 is written outside the refresh-permitted period.

[Example write sequences that are valid for refreshing the counter]

- 0x00 → 0xFF
- 0x00 ((n–1)th time) → 0x00 (nth time) → 0xFF
- 0x00 → access to another register or read from WDTRR → 0xFF

[Example write sequences that are invalid for refreshing the counter]

- 0x23 (a value other than 0x00) → 0xFF
- 0x00 → 0x54 (a value other than 0xFF)
- 0x00 → 0xAA (0x00 and a value other than 0xFF) → 0xFF

After 0xFF is written to the WDT Refresh Register (WDTRR), refreshing the down-counter requires up to 4 cycles of the signal for counting. To meet this requirement, complete writing 0xFF to WDTRR 4 count cycles before the down-counter underflows.

Figure 27.6 shows the WDT refresh-operation waveforms when the clock division ratio is PCLKB/64.

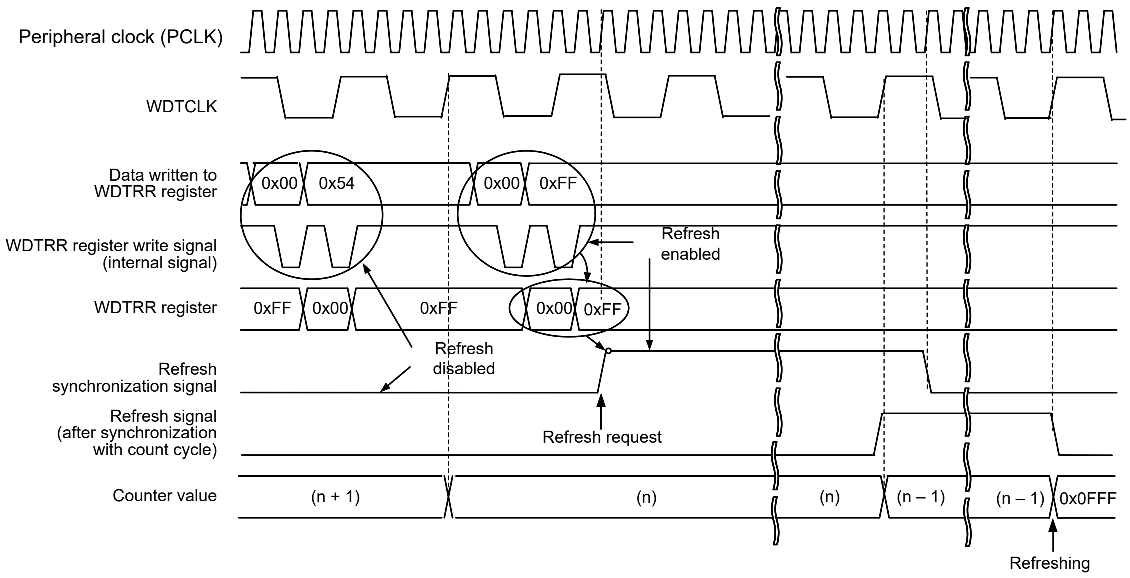


Figure 27.6 WDT refresh operation waveforms when WDTCR.CKS[3:0] = 0x4 and WDTCR.TOPS[1:0] = 01b

Note: When setting the refresh time, consider the oscillation accuracy of the clock sources of the PCLKB and WDTCLK. Set values which ensure that refreshing is possible even when the frequency varies in the range of error of the oscillation accuracy.

27.3.4 Status Flags

The refresh error (WDTSR.REFEF) and underflow (WDTSR.UNDFF) flags retain the source of the interrupt request from the WDT. After a release from the interrupt request generation, read the WDTSR.REFEF and WDTSR.UNDFF flags to check for the interrupt source. For each flag, writing 0 clears the bit. Writing 1 has no effect. Leaving the status flags unchanged does not affect operation. If the flags are not cleared at the next interrupt request from the WDT, the earlier interrupt source is cleared and the new interrupt source is written. For the time period between when 0 is written in each flag and when its value is reflected, see section 27.2.3. WDTSR : WDT Status Register.

27.3.5 Reset Output

When the Reset Interrupt Select bit (WDTRCR.RSTIRQS) is set to 1 in register start mode, or when the WDT Reset Interrupt Request Select bit (OFS0.WDT0RSTIRQS or OFS3.WDT1RSTIRQS) in the Option Function Select Register m (OFS0 or OFS3) is set to 1 in auto start mode, a reset signal is output for 1 cycle count when an underflow in the down-counter or a refresh error occurs.

In register start mode, the down-counter is initialized (all bits set to 0) and stopped in that state after output of a reset signal. After the reset state is released and the program is restarted, the counter is set up again and counting down starts again with a refresh. In auto start mode, counting down starts automatically after the reset state is released.

27.3.6 Interrupt Sources

When the Reset Interrupt Select bit (WDTRCR.RSTIRQS) is set to 0 in register start mode or when the WDT Reset Interrupt Request Select bit (OFS0.WDT0RSTIRQS or OFS3.WDT1RSTIRQS) in the Option Function Select Register m (OFS0 or OFS3) is set to 0 in auto start mode, an interrupt (WDTn_NMIUNDF) signal is generated when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt or an interrupt. For details, see section 14, Interrupt Controller Unit (ICU).

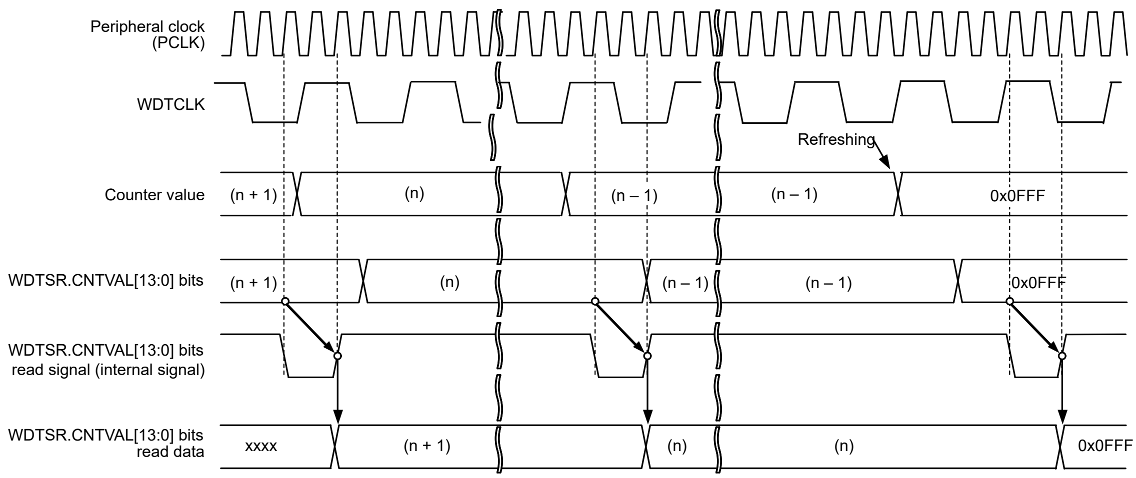
Table 27.4 WDT interrupt source

Name	Interrupt source	Interrupt to CPU	Start DMAC or DTC
WDTn_NMIUNDF (n = 0, 1)	• Down-counter underflow • Refresh error	Possible	Not possible

27.3.7 Reading the Down-Counter Value

The WDT stores the counter value in the down-counter value bits (WDTSR.CNTVAL[13:0]) of the WDT Status Register. Check these bits to obtain the counter value. The read value of the down-counter might differ from the actual count by one.

Figure 27.7 shows the processing for reading the WDT down-counter value when the clock division ratio is PCLKB/64.



Note: PCLK = PCLKB

Figure 27.7 Processing for reading WDT down-counter value when WDTCR.CKS[3:0] = 0x4 and WDTCR.TOPS[1:0] = 01b

27.3.8 Association Between Option Function Select Register m (OFSm) and WDT Registers (m = 0, 3)

Table 27.5 lists the association between the Option Function Select Register m (OFSm) used in auto start mode, and the registers used in register start mode. For details on the Option Function Select Register m (OFSm), see section 27.2.6. Option Function Select Register.

Table 27.5 Association between Option Function Select Register m (OFSm) and the WDT registers

Control target	Function	OFSm register (enabled in auto start mode) OFSm.WDTnSTRT = 0	WDT registers (enabled in register start mode) OFSm.WDTnSTRT = 1
Down-counter	Timeout period selection	OFSm.WDTnTOPS[1:0]	WDTCR.TOPS[1:0]
	Clock division ratio selection	OFSm.WDTnCKS[3:0]	WDTCR.CKS[3:0]
	Window start position selection	OFSm.WDTnRPSS[1:0]	WDTCR.RPSS[1:0]
	Window end position selection	OFSm.WDTnRPES[1:0]	WDTCR.RPES[1:0]
Reset output or interrupt request output	Select a reset interrupt request	OFSm.WDTnRSTIRQS	WDTRCR.RSTIRQS
Count stop	Sleep mode or Deep Sleep mode count stop control	OFSm.WDTnSTPCTL	WDTCSTPR.SLCSTP

27.4 Output to the Event Link Controller (ELC)

The WDT is capable of a link operation for the previously specified module when interrupt request signal is used as an event signal by the ELC. The event signal is output by the counter underflow and refresh error. An event signal is output regardless of the setting of the Reset Interrupt Request Select bit (WDTRCR.RSTIRQS) in register start mode or auto start mode. An event signal can also be output when the next interrupt source is generated while the Refresh Error flag (WDTSR.REFEF) or Underflow flag (WDTSR.UNDFF) is 1. For details, see section 19, Event Link Controller (ELC).

27.5 Usage Notes

27.5.1 ICU Event Link Setting Register n (IELSRn) Setting

Setting 0x53 to ICU Event Link Setting Register n (ICU.IELSRn) is prohibited when enabling the WDT reset assertion (OFS0.WDT0RSTIRQS or OFS3.WDT1RSTIRQS = 0 or WDTRCR.RSTIRQS = 0) or when enabling event link operation (ELSRn.ELS[8:0] = 0x93 or 0x94).

28. Independent Watchdog Timer (IWDT)

28.1 Overview

The Independent Watchdog Timer (IWDT) consists of a 14-bit down counter that must be serviced periodically to prevent counter underflow. The IWDT provides functionality to reset the MCU or to generate a non-maskable interrupt or an underflow interrupt. Because the timer operates with an independent clock from CPU clocks (CPUCLK0 and CPUCLK1) and System clock (ICLK), it is particularly useful in returning the MCU to a known state as a fail-safe mechanism when the system runs out of control. The IWDT can be triggered automatically by a reset, underflow, refresh error, or a refresh of the count value in the registers.

The IWDT functions differ from those of the WDT in the following respects:

The divided IWDT clock (IWDTCLK) is used as the count source (not affected by PCLKB)

Table 28.1 lists the IWDT specifications.

Table 28.1 IWDT specifications

Parameter	Specifications
Count source[*1]	IWDT clock (IWDTCLK)
Clock division ratio	Division by 1, 16, 32, 64, 128, or 256
Counter operation	Counting down using a 14-bit down-counter
Conditions for starting the counter	● Auto start mode: Counting automatically starts after a reset or after an underflow or refresh error occurs. ● Register start mode: Counting is started with a refresh by writing to the IWDTRR register. ● Only secure developer can select Auto-start mode or Register-start mode.
Conditions for stopping the counter	● Reset (the down-counter and other registers return to their initial values). ● A counter underflows or a refresh error is generated. ● The debug function of primary CPU is enabled.
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods).
Independent watchdog timer reset sources	● Down-counter underflows ● Refreshing outside the refresh-permitted period (refresh error)
Non-maskable interrupt/interrupt sources	● Down-counter underflows ● Refreshing outside the refresh-permitted period (refresh error)
Reading of the counter value	The down-counter value can be read by the IWDTSR register.
Event link function (output)	● Down-counter underflow event output ● Refresh error event output
Output signal (internal signal)	● Reset output ● Interrupt request output ● CPU Sleep mode, CPU Deep Sleep mode, Software Standby mode and Deep Software Standby mode count stop control output.
Trust Zone Filter	Security and Privilege attribution can be set.

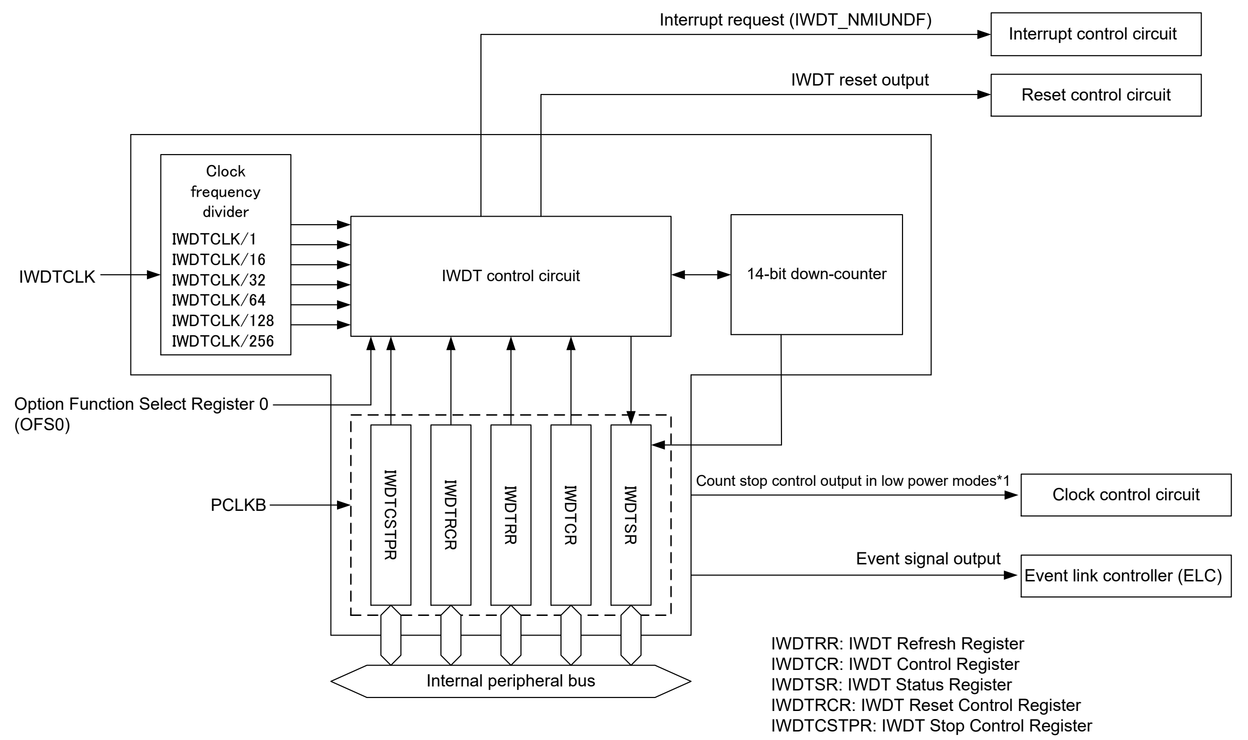
Note 1. Satisfy the frequency of the peripheral module clock (PCLKB) ≧ 32 × (the frequency of the count clock source after division). The bus interface and registers operate with PCLKB, and the 14-bit counter and control circuits operate with IWDTCLK.

28.1.1 Block Diagram

In addition to the peripheral clock (PCLKB), an MCU needs the IWDT clock (IWDTCLK) that does not stop even in low power modes so that the MCU runs even in low power modes in which the peripheral clock (PCLKB) stops. The bus interface and registers operate with the peripheral clock (PCLKB), and the 14-bit down-counter and control circuits operate with the IWDT clock (IWDTCLK).

Signals are connected between the block operating with the peripheral clock (PCLKB) and the block operating with the IWDT clock (IWDTCLK) via a synchronization circuit.

Figure 28.1 shows a block diagram.



Note 1. Low power modes indicate CPU Sleep mode, CPU Deep Sleep mode, Software Standby mode, Deep Software Standby mode 1

Figure 28.1 IWDT block diagram

28.2 Register Specifications

28.2.1 IWDTRR : IWDT Refresh Register

Base address: IWDT = 0x4020_2200
IWDT_NS = 0x5020_2200

Offset address: 0x0

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

Bit	Symbol	Function	R/W
7:0	REFRESH[7:0]	Refresh Register The counter is refreshed by writing 0x00 and then writing 0xFF to this register.	R/W

Note: S-TYPE-3, P-TYPE-3

The IWDTRR register refreshes the down-counter of the IWDT.

REFRESH[7:0] bits (Refresh Register)

The down-counter of the IWDT is refreshed by writing 0x00 and then writing 0xFF to IWDTRR(refresh operation) within the refresh-permitted period. After the down-counter is refreshed, it starts counting down from the value selected by the IWDT time-out period selection bits (OFS0.IWDTTOPS[1:0]) in the auto start mode.

In the register start mode, counting down starts from the value set by the TOPS[1:0] bits in the IWDT Control Register (IWDTCR).

Also in the register start mode, counting down starts from the value set by the IWDTCR.TOPS[1:0] bits by the first refresh operation after the release from the reset state.

When 0x00 is written, the read value is always 0x00. When a value other than 0x00 is written, the read value is always 0xFF.

For details of the refresh operation, see section 28.3.3. Refresh Operation.

28.2.2 IWDTCR : IWDT Control Register

Base address: IWDT = 0x4020_2200
IWDT_NS = 0x5020_2200

Offset address: 0x2

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

Bit	Symbol	Function	R/W
1:0	TOPS[1:0]	Timeout Period Select 0 0: 128 cycles (0x007F) 0 1: 512 cycles (0x01FF) 1 0: 1024 cycles (0x03FF) 1 1: 2048 cycles (0x07FF) Each value in parentheses () denotes a start value of down-counting.	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
7:4	CKS[3:0]	Clock Division Ratio Select 0 0 0 0: IWDTCLK 0 0 1 0: IWDTCLK/16 0 0 1 1: IWDTCLK/32 0 1 0 0: IWDTCLK/64 1 1 1 1: IWDTCLK/128 0 1 0 1: IWDTCLK/256 Other settings are prohibited.	R/W
9:8	RPES[1:0]	Window End Position Select 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (No window end position setting)	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
13:12	RPSS[1:0]	Window Start Position Select 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (No window start position setting)	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The IWDTCR register is used to specify the timeout period (time until the down-counter underflows in the register start mode), clock division ratio, and the window start and end positions of refresh-permitted period. Some restrictions apply to writes to this register. For details, see section 28.3.2. Controlling Writes to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.

In auto start mode, the settings of this register are disabled and the settings for the option function select register 0 (OFS0) are enabled. The same settings as in the bits in this register can be performed at the option function select register 0 (OFS0). For details, see section 28.3.8. Correspondence Between Option Function Select Register 0 (OFS0) and IWDT Registers.

TOPS[1:0] bits (Timeout Period Select)

The TOPS[1:0] bits select the timeout period, the period until the down-counter underflows, from 128, 512, 1024, and 2048 cycles, taking the divided clock specified in the CKS[3:0] bits as one cycle.

After the down-counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the time (number of IWDT clock (IWDTCLK) cycles) until the counter underflows.

section 28.2.2. IWDTCR : IWDT Control Register shows the relationship among the CKS[3:0] and TOPS[1:0] bits settings, the timeout period, and the number of IWDT clock (IWDTCLK) cycles.

Table 28.2 IWDT timeout period settings

CKS[3:0]				TOPS[1:0]		Clock division ratio	Timeout period (Number of cycles)	Number of IWDT clock (IWDTCLK) cycles
0	0	0	0	0	0	IWDTCLK	128	128
				0	1		512	512
				1	0		1024	1024
				1	1		2048	2048
0	0	1	0	0	0	IWDTCLK/16	128	2048
				0	1		512	8192
				1	0		1024	16384
				1	1		2048	32768
0	0	1	1	0	0	IWDTCLK/32	128	4096
				0	1		512	16384
				1	0		1024	32768
				1	1		2048	65536
0	1	0	0	0	0	IWDTCLK/64	128	8192
				0	1		512	32768
				1	0		1024	65536
				1	1		2048	131072
1	1	1	1	0	0	IWDTCLK/128	128	16384
				0	1		512	65536
				1	0		1024	131072
				1	1		2048	262144
0	1	0	1	0	0	IWDTCLK/256	128	32768
				0	1		512	131072
				1	0		1024	262144
				1	1		2048	524288

CKS[3:0] bits (Clock Division Ratio Select)

These bits select IWDT clock (IWDTCLK) division ratio from among division by 1, 16, 32, 64, 128, and 256.

Combined with the TOPS[1:0] bits setting, a count period between 128 and 524288 cycles of the IWDT clock (IWDTCLK) can be selected for the IWDT.

Note: In order to read the down-counter value correctly, the relationship between the peripheral clock (PCLKB) frequency and the count clock (IWDTCLK) frequency must be set appropriately. For the setting, see Note1 in Table 28.1

RPES[1:0] bits (Window End Position Select)

These bits select 75%, 50%, 25% or 0% of the count period for the window end position of the down-counter. The selected window end position should be a value smaller than the value for the window start position (that is, window start position > window end position). If the value for the window end position is greater than the value for the window start position, only the value for the window start position is effective.

The counter values for the window start and end positions selected by setting the RPES[1:0] and RPSS[1:0] bits change depending on the TOPS[1:0] bits setting.

Table 28.3 lists the counter values for the window start and end positions corresponding to TOPS[1:0] bits values.

Table 28.3 Relationship between timeout periods and window start and end counter values

TOPS[1:0]		Timeout period		Window start and end counter value			
		Number of cycles	Counter value	100%	75%	50%	25%
0	0	128	0x007F	0x007F	0x005F	0x003F	0x001F
0	1	512	0x01FF	0x01FF	0x017F	0x00FF	0x007F
1	0	1024	0x03FF	0x03FF	0x02FF	0x01FF	0x00FF
1	1	2048	0x07FF	0x07FF	0x05FF	0x03FF	0x01FF

RPSS[1:0] bits (Window Start Position Select)

These bits select a counter window start position from 100%, 75%, 50%, or 25% of the count period (100% when the count starts and 0% when the counter underflows). The interval between the window start position and window end position is the refresh-permitted period and the other periods are refresh-prohibited periods.

Figure 28.2 shows the relationship between the RPSS[1:0] and RPES[1:0] bits settings and the refresh-permitted and refresh-prohibited periods.

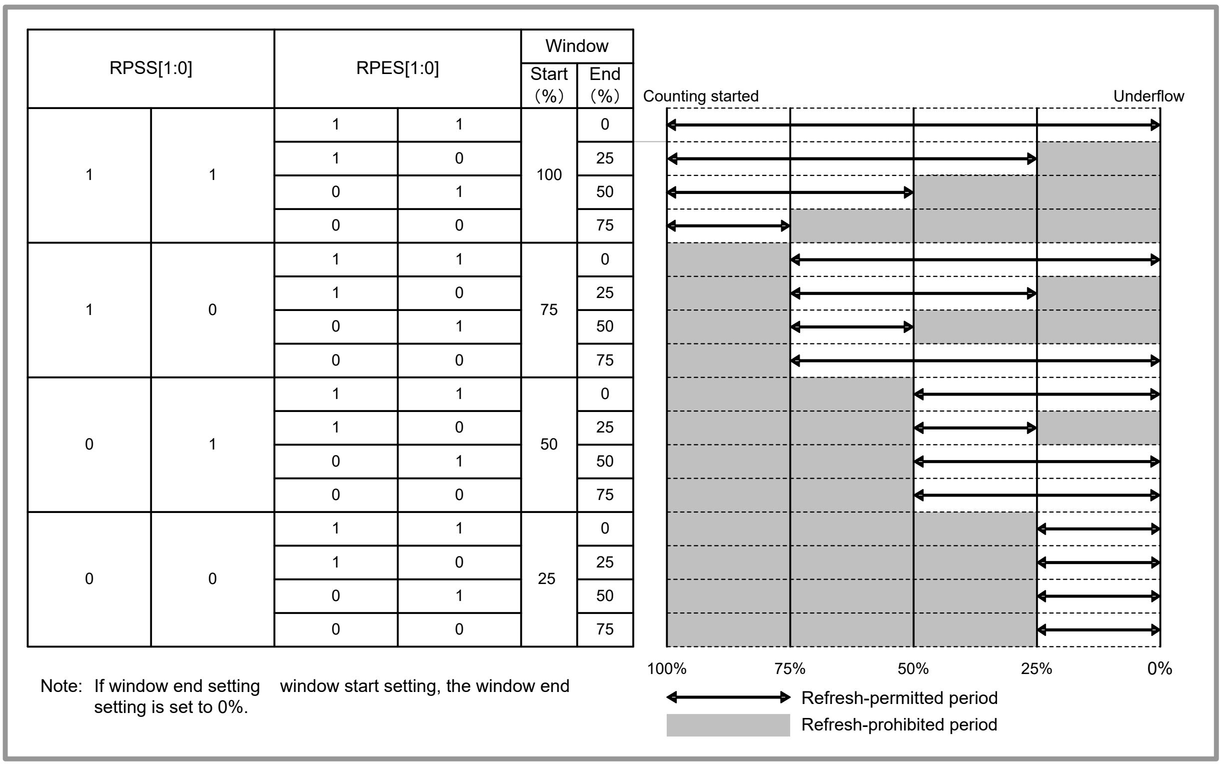


Figure 28.2 RPSS[1:0] and RPES[1:0] bits settings and refresh-permitted and refresh-prohibited periods

28.2.3 IWDTSR : IWDT Status Register

Base address: IWDT = 0x4020_2200
IWDT_NS = 0x5020_2200

Offset address: 0x4

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

Bit	Symbol	Function	R/W
13:0	CNTVAL[13:0]	Down-Counter Value Counter value of the down-counter	R
14	UNDFF	Underflow Flag 0: No underflow occurred. 1: Underflow occurred.	R/W
15	REFEF	Refresh Error Flag 0: No refresh error occurred. 1: Refresh error occurred.	R/W

Note: S-TYPE-3, P-TYPE-3

The IWDTSR register indicates the counter value of the down-counter and whether an underflow or refresh error occurred in the down-counter.

It is initialized by the reset source of the IWDT. IWDTSR is not initialized by other reset sources.

CNTVAL[13:0] bits (Down-Counter Value)

Read these bits to confirm the counter value of the down-counter.

Note: The read value might differ from the actual count by 1.

UNDFF flag (Underflow Flag)

Read the UNDFF flag to confirm whether an underflow occurred in the down-counter.

A value of 1 indicates that the down-counter underflowed. A value of 0 indicates that the down-counter has not underflowed.

Write 0 to the UNDFF flag to clear the value. Writing 1 has no effect.

Clearing of the UNDFF flag takes (N + 2) IWDTCLK cycles and 32 PCLKB cycles. In addition, clearing of this flag is ignored for (N + 2) IWDTCLK cycles after an underflow. N is specified as follows:

Auto start mode:

- When OFS0.IWDTCKS[3:0] = 0x0, N = 1
- When OFS0.IWDTCKS[3:0] = 0x2, N = 16
- When OFS0.IWDTCKS[3:0] = 0x3, N = 32
- When OFS0.IWDTCKS[3:0] = 0x4, N = 64
- When OFS0.IWDTCKS[3:0] = 0xF, N = 128
- When OFS0.IWDTCKS[3:0] = 0x5, N = 256

Register start mode:

- When IWDTCR.CKS[3:0] = 0x0, N = 1
- When IWDTCR.CKS[3:0] = 0x2, N = 16
- When IWDTCR.CKS[3:0] = 0x3, N = 32
- When IWDTCR.CKS[3:0] = 0x4, N = 64
- When IWDTCR.CKS[3:0] = 0xF, N = 128
- When IWDTCR.CKS[3:0] = 0x5, N = 256

REFEF flag (Refresh Error Flag)

Read the REFEF flag to confirm whether a refresh error (a refresh operation was performed during the prohibited period) occurred.

A value of 1 indicates that a refresh error occurred. A value of 0 indicates that no refresh error has occurred.

Write 0 to the REFEF flag to clear the value. Writing 1 has no effect.

Clearing of the REFEF flag takes (N + 2) IWDTCLK cycles and 32 PCLKB cycles. In addition, clearing of this flag is ignored for (N + 2) IWDTCLK cycles following a refresh error. N is specified as follows:

Auto start mode:

- When OFS0.IWDTCKS[3:0] = 0x0, N = 1
- When OFS0.IWDTCKS[3:0] = 0x2, N = 16
- When OFS0.IWDTCKS[3:0] = 0x3, N = 32
- When OFS0.IWDTCKS[3:0] = 0x4, N = 64
- When OFS0.IWDTCKS[3:0] = 0xF, N = 128
- When OFS0.IWDTCKS[3:0] = 0x5, N = 256

Register start mode:

- When IWDTCR.CKS[3:0] = 0x0, N = 1
- When IWDTCR.CKS[3:0] = 0x2, N = 16
- When IWDTCR.CKS[3:0] = 0x3, N = 32
- When IWDTCR.CKS[3:0] = 0x4, N = 64
- When IWDTCR.CKS[3:0] = 0xF, N = 128
- When IWDTCR.CKS[3:0] = 0x5, N = 256

28.2.4 IWDTRCR : IWDT Reset Control Register

Base address: IWDT = 0x4020_2200
IWDT_NS = 0x5020_2200

Offset address: 0x6

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

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	RSTIRQS	Reset Interrupt Request Select 0: Enable non-maskable interrupt requests or interrupt request output. 1: Enable reset output.	R/W

Note: S-TYPE-3, P-TYPE-3

IWDTRCR register controls the reset output or interrupt request output when the down-counter of the IWDT underflows. Some restrictions apply to writes to this register. For details, see section 28.3.2. Controlling Writes to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.

In auto start mode, the settings of this register are disabled and the settings for the input ports (OFS0) are enabled. The same settings as in the bits in this register can be performed at the input ports (OFS0). For details, see section 28.3.8. Correspondence Between Option Function Select Register 0 (OFS0) and IWDT Registers.

RSTIRQS bit (Reset Interrupt Request Select)

The RSTIRQS bit selects the reset output or interrupt request output when the down-counter underflows or a refresh error occurs.

28.2.5 IWDTCSTPR : IWDT Count Stop Control Register

Base address: IWDT = 0x4020_2200
IWDT_NS = 0x5020_2200

Offset address: 0x8

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

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	—	This bit is read as 0.	R
7	SLCSTP	CPU Sleep-Mode Count Stop Select 0: Disable count stop. 1: Stop the counter when the primary CPU enters CPU Sleep mode or CPU Deep Sleep mode, and when the MCU enters Software Standby mode or Deep Software Standby mode 1.	R/W

Note: S-TYPE-3, P-TYPE-3

The IWDTCSTPR register is used to stop the down-counter of the IWDT in case the primary CPU enters low-power modes. Some restrictions apply to writes to this register. For details, see section 28.3.2. Controlling Writes to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.

In auto start mode, the settings of this register are disabled and the settings for the option function select register 0 (OFS0) are enabled. The same settings as in the bits in this register can be performed at the option function select register 0 (OFS0). For details, see section 28.3.8. Correspondence Between Option Function Select Register 0 (OFS0) and IWDT Registers.

SLCSTP bit (CPU Sleep-Mode Count Stop Select)

The SLCSTP bit selects whether to stop the counter in case the primary CPU enters CPU Sleep mode or CPU Deep Sleep mode, and the MCU enters Software Standby mode or Deep Software Standby mode 1.

28.3 Operation

28.3.1 Count Operation in Each Start Mode

The start mode of the IWDT is selected by setting the IWDT start mode selection (OFS0.IWDTSTRT) bit during a reset.

When the IWDT start mode selection (OFS0.IWDTSTRT) bit is 1 (register start mode), the IWDT Control Register (IWDTCR), IWDT Reset Control Register (IWDTRCR), and IWDT Count Stop Control Register (IWDTCSTPR) are enabled, and counting is started by writing to the IWDT Refresh Register (IWDTRR) to refresh the counter. When the IWDT start mode selection (OFS0.IWDTSTRT) bit is 0 (auto start mode), the settings of the option function select register 0 (OFS0) are enabled, and counting automatically starts after a reset.

(1) Register start mode

When the register start mode is selected by setting the IWDT start mode selection (OFS0.IWDTSTRT) bit to 1 during a reset, the IWDT Control Register (IWDTCR), IWDT Reset Control Register (IWDTRCR), and IWDT Count Stop Control Register (IWDTCSTPR) are enabled.

After the reset state is released, set the clock division ratio, window start and end positions, and timeout period in the IWDTCR register, the reset output or interrupt request output in the IWDTRCR register, and the IWDT down-counter stop control at transitions to low power modes in the IWDTCSTPR register. Then, refresh the down-counter to set the value specified by the IWDTCR.TOPS[1:0] bits in the counter and start counting down.

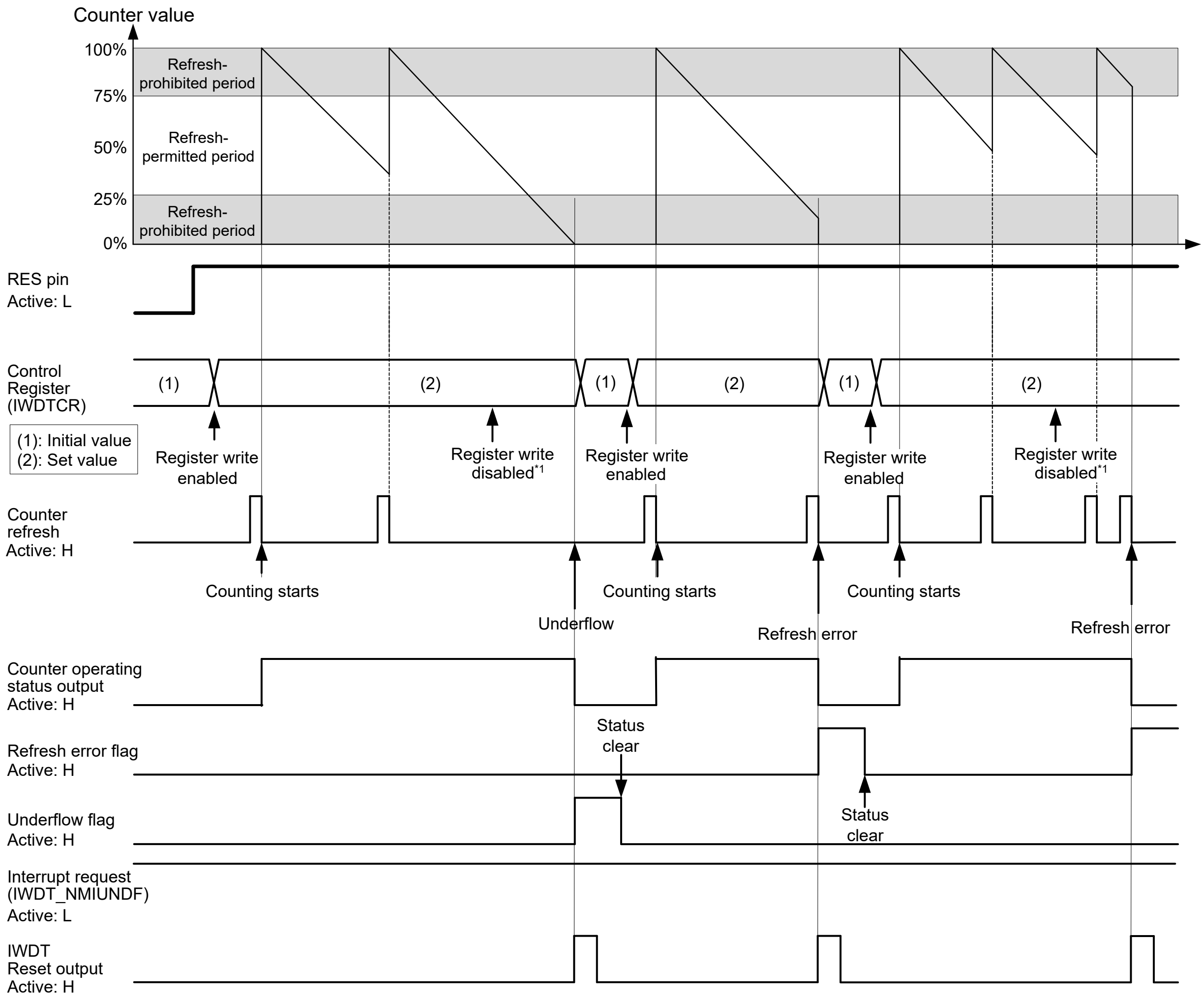
After that, as long as the program continues normal operation and the counter is refreshed in the refresh permitted period, the value in the counter is reset each time the counter is refreshed and counting down continues. The IWDT does not output a reset signal as long as this continues.

However, if the down counter underflows because the down-counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the IWDT outputs a reset signal or a non-maskable interrupt request (IWDT_NMIUNDF).

The reset output or interrupt request output is selected by setting the IWDTRCR.RSTIRQS bit.

Figure 28.3 shows an example of operation under the following conditions:

- Register start mode (OFS0.IWDTSTRT = 1)
- Reset output is enabled (IWDTRCR.RSTIRQS = 1)
- The window start position is 75% (IWDTCR.RPSS[1:0] = 10b)
- The window end position is 25% (IWDTCR.RPES[1:0] = 10b)



Note 1. See section 28.3.2. Controlling Writes to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers

Figure 28.3 Operation example in register start mode

(2) Auto start mode

When the auto start mode is selected by setting the IWDT start mode selection (OFS0.IWDTSTRT) bit to 0 during a reset, the IWDT Control Register (IWDTCR), IWDT Reset Control Register (IWDTRCR), and IWDT Count Stop Control Register (IWDTCSTPR) are disabled.

During a reset, the clock division ratio, window start and end positions, timeout period, reset output or interrupt request output, and counter stop control at transitions to low power modes should be set through the option function select register 0

(OFS0). Then, when the reset state is released, the timeout period set through the option function select register 0 (OFS0) is set in the down-counter, and then the down-counter automatically starts counting down.

After that, as long as the program continues normal operation and the counter is refreshed in the refresh permitted period, the value in the counter is reset each time the counter is refreshed and counting down continues. The IWDT does not output a reset signal as long as this continues.

However, if the down-counter underflows because the down counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the IWDT outputs a reset signal or a non-maskable interrupt request (IWDT_NMIUNDF).

The reset output or interrupt request output is selected by setting the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS).

Figure 28.4 shows an example of operation under the following conditions:

- Auto start mode (OFS0.IWDTSTRT = 0)
- Non-maskable interrupt request output is enabled (OFS0.IWDTRSTIRQS = 0)
- The window start position is 75% (OFS0.IWDTRPSS[1:0] = 10b)
- The window end position is 25% (OFS0.IWDTRPES[1:0] = 10b)

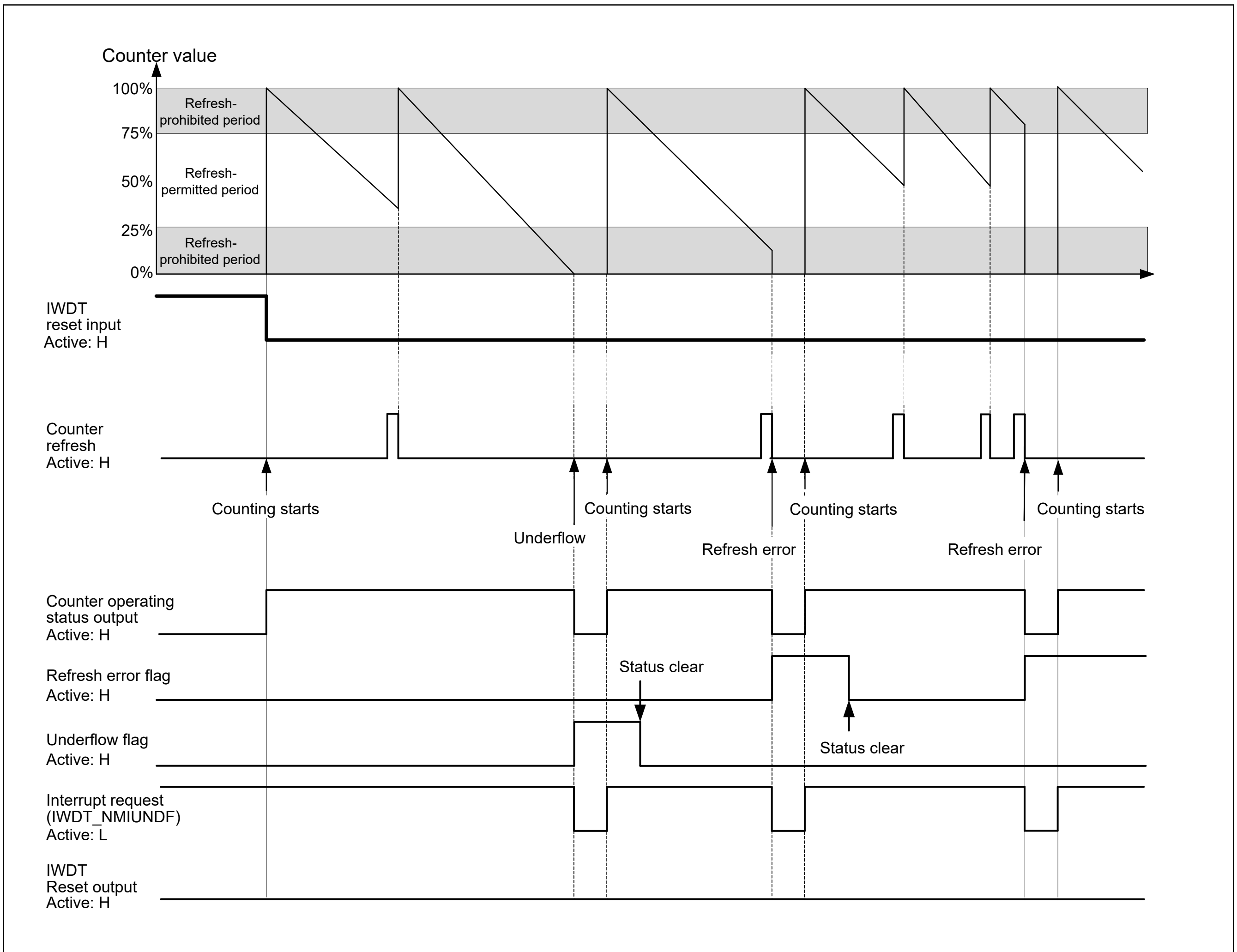


Figure 28.4 Operation example in auto start mode

28.3.2 Controlling Writes to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers

Writing to the IWDT Control Register (IWDTCR) is possible only once between the release from the reset state and the first refresh operation.

After a refresh operation (counting starts) or by writing to the IWDTCR register, the protection signal in the IWDT becomes 1 to protect the IWDTCR register against subsequent attempts of writing.

The IWDT Reset Control Register (IWDTRCR) and the IWDT Count Stop Control Register (IWDTCSTPR) are also controlled in the same way.

This protection is released by a reset source to the IWDT. With other reset sources, the protection is not released.

Figure 28.5 shows control waveforms produced in response to writing to the IWDTCR register.

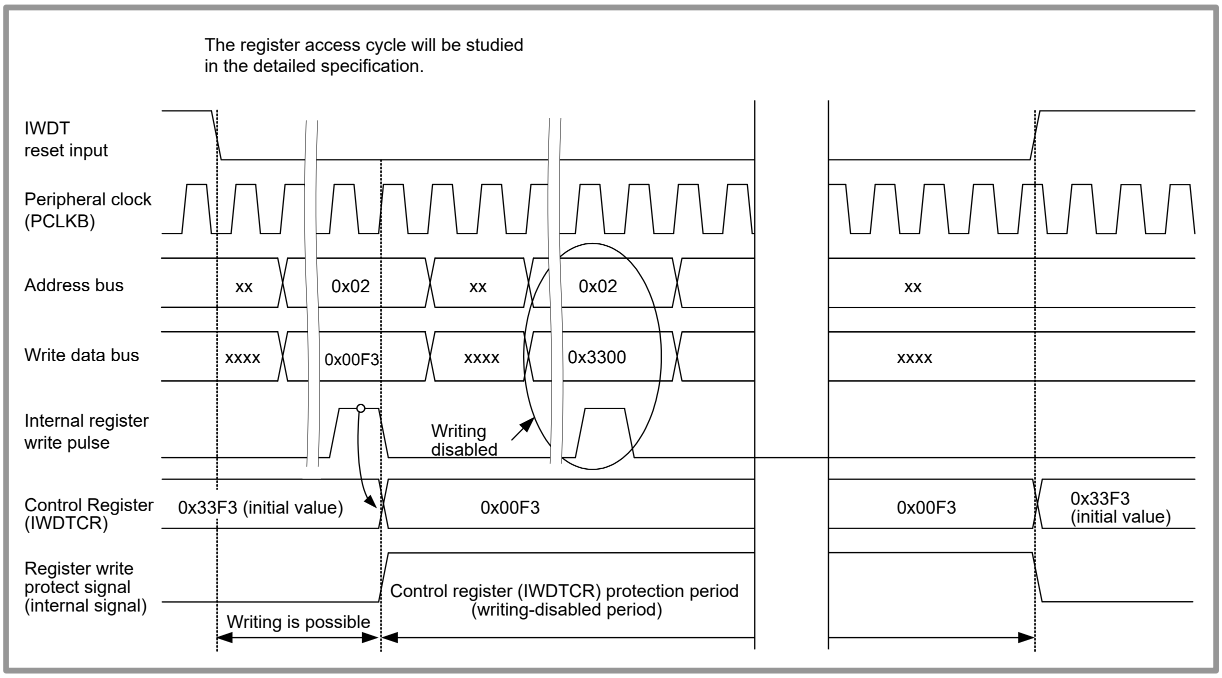


Figure 28.5 Control waveforms produced in response to writes to the IWDTCR register

28.3.3 Refresh Operation

The down-counter is refreshed and starts operation (counting is started by refreshing) by writing the values 0x00 and then 0xFF to the IWDT Refresh Register (IWDTRR). If a value other than 0xFF is written after 0x00, the counter is not refreshed. After such invalid writing, correct refreshing is performed by writing 0x00 and then 0xFF again to the IWDT Refresh Register (IWDTRR).

When writing is done in the order of 0x00 (first time) → 0x00 (second time), and if 0xFF is written after that, the writing order 0x00 → 0xFF is satisfied; writing 0x00 ((n–1) th time) → 0x00 (nth time) → 0xFF is valid and correct refreshing will be done. Even when the first value written before 0x00 is not 0x00, correct refreshing will be done if the operation contains the sequence of writing 0x00 → 0xFF. Correct refreshing is also performed when a register other than IWDTRR is accessed or IWDTRR is read between writing 0x00 and writing 0xFF to IWDTRR.

[Example write sequences that are valid for refreshing the counter]

- 0x00 → 0xFF
- 0x00 ((n-1) th time) → 0x00 (nth time) → 0xFF
- 0x00 → access to another register or read from IWDTRR → 0xFF

[Example write sequences that are invalid for refreshing the counter]

- 0x23 (a value other than 0x00) → 0xFF
- 0x00 → 0x54 (a value other than 0xFF)
- 0x00 → 0xAA (0x00 and a value other than 0xFF) → 0xFF

Even when 0x00 is written to IWDTRR outside the refresh-permitted period, if 0xFF is written to IWDTRR in the refresh-permitted period, the writing sequence is valid and refreshing will be done.

After 0xFF is written to the IWDTRR register, refreshing the counter requires up to four cycles of the signal for counting (the clock division ratio select bits (IWDTCR.CKS[3:0]) determine how many cycles of the IWDT clock (IWDTCLK) make up one cycle for counting).

Therefore, writing 0xFF to the IWDTRR register should be completed four count cycles before the end position of the refresh-permitted period or the down-counter underflow. The value of the down-counter can be checked using the down-counter value bits (IWDTSR.CNTVAL[13:0]).

[Example refreshing timings]

- When the window start position is set to 0x03FF, even if 0x00 is written to IWDTRR before 0x03FF is reached (at 0x0402, for example), refreshing occurs if 0xFF is written to the IWDTRR register after the value of the IWDTSR.CNTVAL[13:0] bits reaches 0x03FF.
- When the window end position is set to 0x03FF, refreshing is done if 0x0403 (four count cycles before 0x03FF) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 0x00 → 0xFF to the IWDTRR register.
- When the refresh-permitted period continues until count 0x0000, refreshing can be done immediately before an underflow. In this case, if 0x0003 (four count cycles before an underflow) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 0x00 → 0xFF to the IWDTRR register, no underflow occurs and refreshing is done.

Figure 28.6 shows the IWDT refresh operation waveforms when the peripheral clock (PCLKB) > IWDT clock (IWDTCLK) and the clock division ratio = IWDTCLK/1. Figure 28.7 shows the IWDT refresh operation waveforms when the peripheral clock (PCLKB) < IWDT clock (IWDTCLK) and the clock division ratio = IWDTCLK/16.

When transitioning to Software Standby mode or Deep Software Standby mode 1, the refresh procedure must be completed and the read value of the IWDTRR register must be 0xFF before the transitioning (Write 0xFF after writing 0x00 to the IWDTRR register, and confirm that the read value of the IWDTRR register is 0xFF).

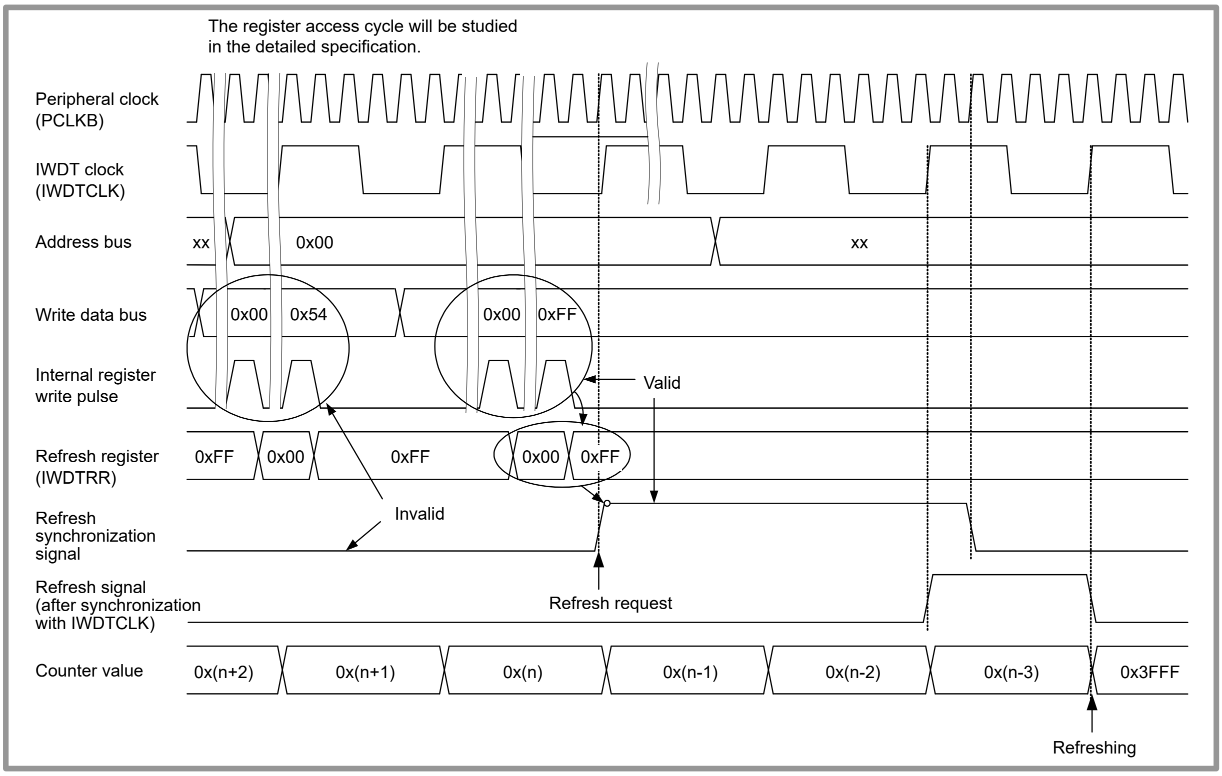


Figure 28.6 IWDT refresh operation waveforms (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

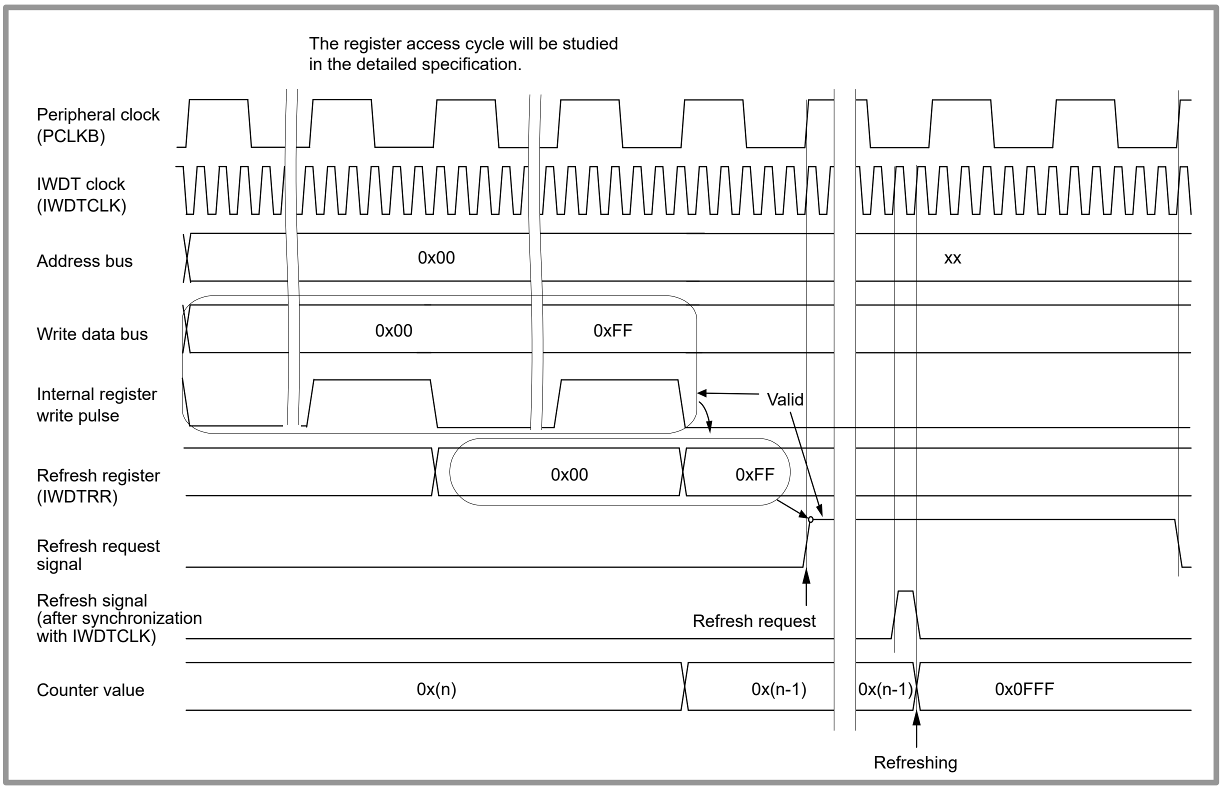


Figure 28.7 IWDT refresh operation waveforms (IWDTCR.CKS[3:0] = 0010b, IWDTCR.TOPS[1:0] = 01b)

28.3.4 Status Flags

The refresh error flag (IWDTSR.REFEF) and underflow flag (IWDTSR.UNDFF) retain the source of the reset output from the IWDT or the source of the interrupt request from the IWDT.

Thus, after release from the reset state or interrupt request generation, read the IWDTSR.REFEF and IWDTSR.UNDFF flags to check for the reset or interrupt source.

For each flag, writing 0 clears the bit and writing 1 has no effect.

Leaving each flag unchanged does not affect operation. If the flags are not cleared, the earlier reset or interrupt source is cleared and the new reset or interrupt source is written when the IWDT outputs a reset signal next.

28.3.5 Reset Output

When the reset interrupt select bit (IWDTRCR.RSTIRQS) is set to 1 in register start mode or when the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) is set to 1 in auto start mode, a reset signal is output for one count cycle when the down counter overflows or a refresh error occurs. In register start mode, the down counter is initialized (all bits set to 0) and kept in that state after assertion of the reset signal. After the reset is released and the program is restarted, the counter is set up again and counting down is started by refreshing. In auto start mode, counting down automatically starts after the reset output.

28.3.6 Interrupt Sources

When the reset interrupt select bit (IWDTRCR.RSTIRQS) is set to 0 in register start mode or when the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) is set to 0 in auto start mode, a non-maskable interrupt (IWDT_NMIUNDF) is generated for one-count cycle when the down-counter overflows or a refresh error occurs.

28.3.7 Reading the Down-counter Value

Because the down-counter in the IWDT operates on the IWDT clock (IWDTCLK), the counter value cannot be read directly.

Therefore, the IWDT synchronizes the counter value with the peripheral clock (PCLKB) and stores it in the down-counter value bits (CNTVAL[13:0]) of the IWDT Status Register (IWDTSR). The counter value can be checked indirectly by reading the value stored in the IWDTSR.CNTVAL[13:0] bits.

Reading the counter value requires multiple peripheral clock (PCLKB) cycles (up to four clock cycles), and the read counter value may differ from the actual counter value by a value of one count.

Figure 28.8 shows the processing for reading the IWDT counter value when the peripheral clock (PCLKB) > IWDT clock (IWDTCLK) and the clock division ratio = IWDTCLK/1. Figure 28.9 shows the processing for reading the IWDT counter value when the peripheral clock (PCLKB) < IWDT clock (IWDTCLK) and the clock division ratio = IWDTCLK/16.

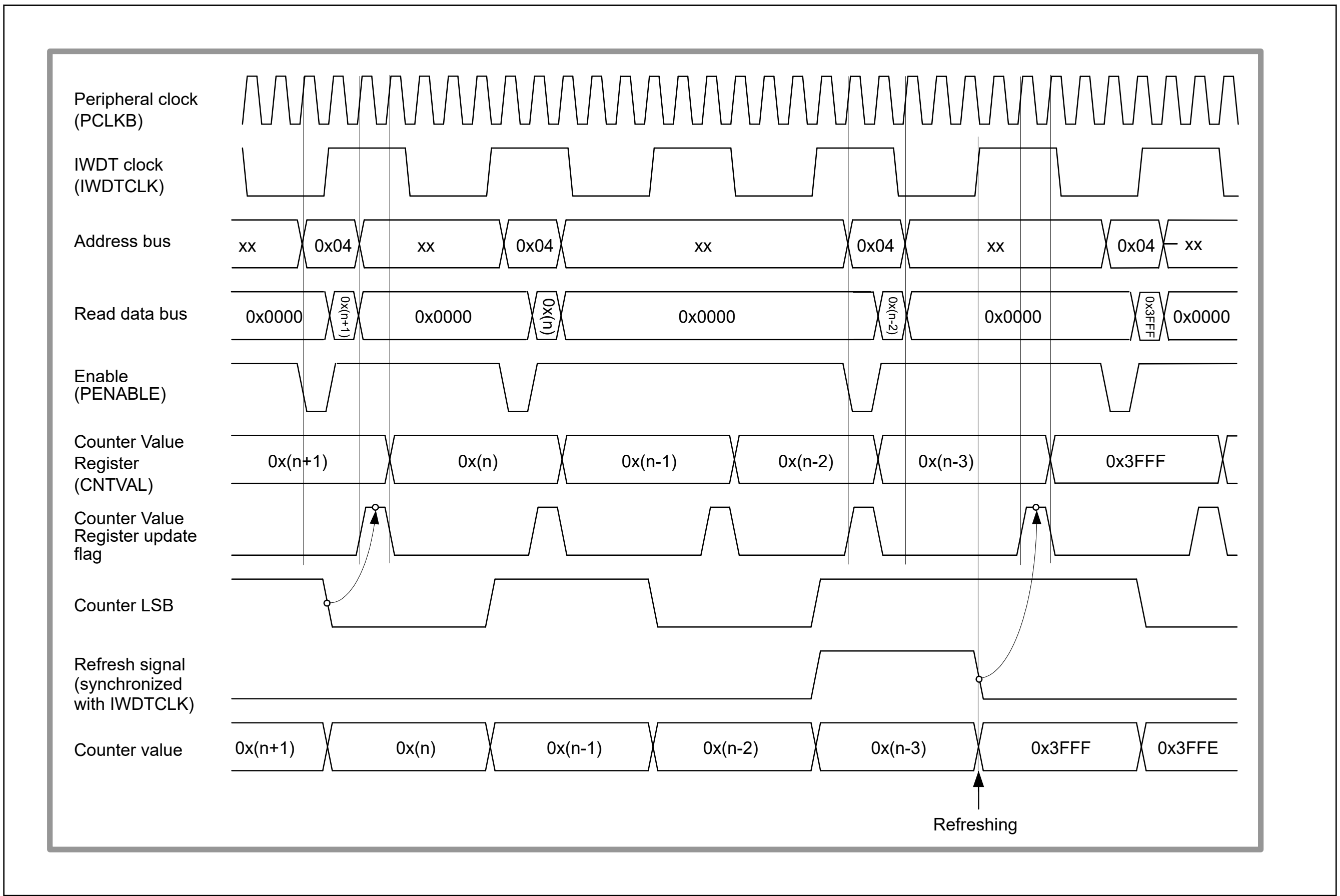


Figure 28.8 Processing for reading IWDT counter value when IWDTCR.CKS[3:0] = 0000b and IWDTCR.TOPS[1:0] = 11b

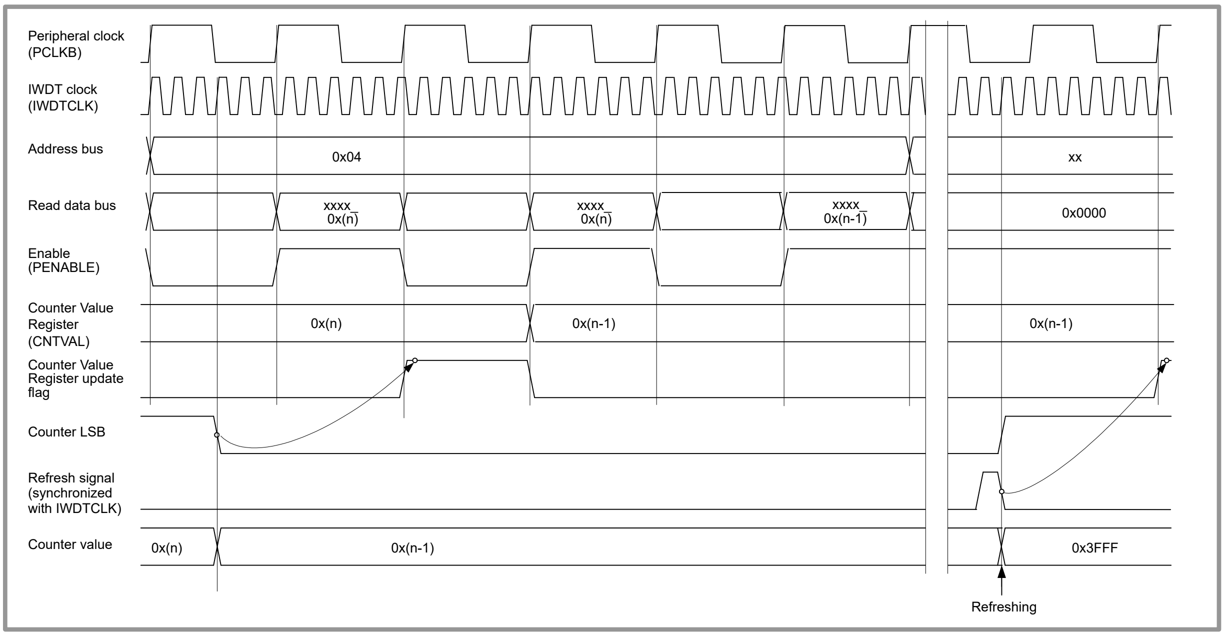


Figure 28.9 Processing for reading IWDT counter value when IWDTCR.CKS[3:0] = 0010b and IWDTCR.TOPS[1:0] = 11b

28.3.8 Correspondence Between Option Function Select Register 0 (OFS0) and IWDT Registers

Table 28.4 lists the correspondence between the option function select register 0 (OFS0) and the IWDT registers (IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), and IWDT count stop control register (IWDTCSTPR)) regarding control of the down-counter, reset or interrupt request output, and count stop function.

Control can be switched between the option function select register 0 (OFS0) and the IWDT registers (IWDTCR, IWDTRCR, and IWDTCSTPR) through the setting of the IWDT start mode selection (OFS0.IWDTSTRT) bit.

Note that the option function select register 0 (OFS0) setting should be kept unchanged during IWDT operation.

Table 28.4 Association between Option Function Select Register 0 (OFS0) and the IWDT register

Control	Function	OFS0 register (enabled in auto start mode) OFS0.IWDTSTRT = 0	IWDT register (enabled in register start mode) OFS0.IWDTSTRT = 1
Down-counter	Timeout period selection	OFS0.IWDTTOPS[1:0]	IWDTCR.TOPS[1:0]
	Clock division ratio selection	OFS0.IWDTCKS[3:0]	IWDTCR.CKS[3:0]
	Window start position selection	OFS0.IWDTRPSS[1:0]	IWDTCR.RPSS[1:0]
	Window end position selection	OFS0.IWDTRPES[1:0]	IWDTCR.RPES[1:0]
Reset output/ interrupt request output	Reset output/ interrupt request output select	OFS0.IWDTRSTIRQS	IWDTRCR.RSTIRQS
Stop the counter	CPU Sleep-mode count stop selection	OFS0.IWDTSTPCTL	IWDTCSTPR.SLCSTP

28.4 Link Operation by the Event Link Function

The IWDT is capable of link operation for a specified module when an interrupt request signal is used as an event signal. The IWDT outputs an event signal on an underflow of the down-counter or generation of a refresh error.

It outputs an event signal regardless of the setting of the reset interrupt request select bit (IWDTRCR.RSTIRQS) in register start mode or the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) in auto start mode. It can output an event signal at generation of the next interrupt source while retaining the refresh error flag (IWDTSR.REFEF) or underflow flag (IWDTSR.UNDFF).

28.5 Usage Notes

28.5.1 Transitioning to Low Power Mode

The refresh procedure must be completed before transitioning to low power mode. For details, see section 28.3.3. Refresh Operation.

28.5.2 Restriction on reading the counter value after returning from Deep Software Standby mode 1

After returning from Deep Software Standby mode 1, the counter value may not be read correctly and may read 0. To read the correct counter value after returning from Deep Software Standby mode 1, one of the following is required:

- Discard the read value until a value other than 0 is read.
- Wait one countdown period before reading the counter value.
- Refresh the counter when the refresh-permitted period is set to 100%.

29. Layer 3 Ethernet Switch Module (ESWM)

29.1 Overview

The Layer 3 Ethernet Switch Module (ESWM) consists of an Ethernet switch with higher-level routing capability and multi-protocol interface support. It allows autonomous frame routing within the same and between different network interface protocols (for now only Ethernet) for optimized gateway applications.

Table 29.1 lists the ESWM specifications and Figure 29.1 shows the ESWM block diagram.

Table 29.1 ESWM specifications

Item	Function		Specification
Basic function	PHY interface	Number or port	2 ports
		Type	MII, RMII, GMII, RGMII
		Speed	10 Mbps,100 Mbps, 1Gbps
		Mode	Full Duplex
		Management IF(MDIO)	Yes
		Energy Efficient Ethernet (low power idle)	Yes
		Magic Packet™	Yes
	Switch function	Layer 3 support	Yes
		Cut through	Yes
		VLAN	Yes
		Jumbo frame support	Max 64 KB
		MAC table	2048
		VLAN table	4096
		Stream (perfect filter, layer 3, layer 2) table	256
		Layer 2/Layer 3 update rule table	256
	gPTP	Number of channels	2
	PTP pulse generator	Number of channels	4
Ethernet AVB	IEEE802.1BA	The AVB profile	Yes
	IEEE1722-2011	Media transport protocol for audio and video	Yes
	IEEE 802.1AS-2011	gPTP (a profile of IEEE1588 for AVB systems)	Yes
	IEEE 802.1Qav-2009	Credit based shaper for bandwidth reservation	Yes
Ethernet TSN	IEEE802.1AS-Rev	Adds redundancy and enhancements to gPTP	Yes
	IEEE 802.1Qbv-2015	Time aware shaper	Yes
	IEEE 802.1Qbu-2016 (802.3br)	Frame preemption	Yes
	IEEE 802.1Qch-2017	Cyclic queuing and forwarding	Yes
	IEEE 802.1Qcr	Asynchronous traffic shaping	Yes
	IEEE 802.1Qci-2017	Stream based filtering and policing	Yes (without gate filter)
	IEEE 802.1CB-2017	Frame replication and elimination for reliability	Yes
TrustZone Filter	Security attribution and privilege attribution can be set		

Note: Please contact a Renesas Electronics sales office for network support.

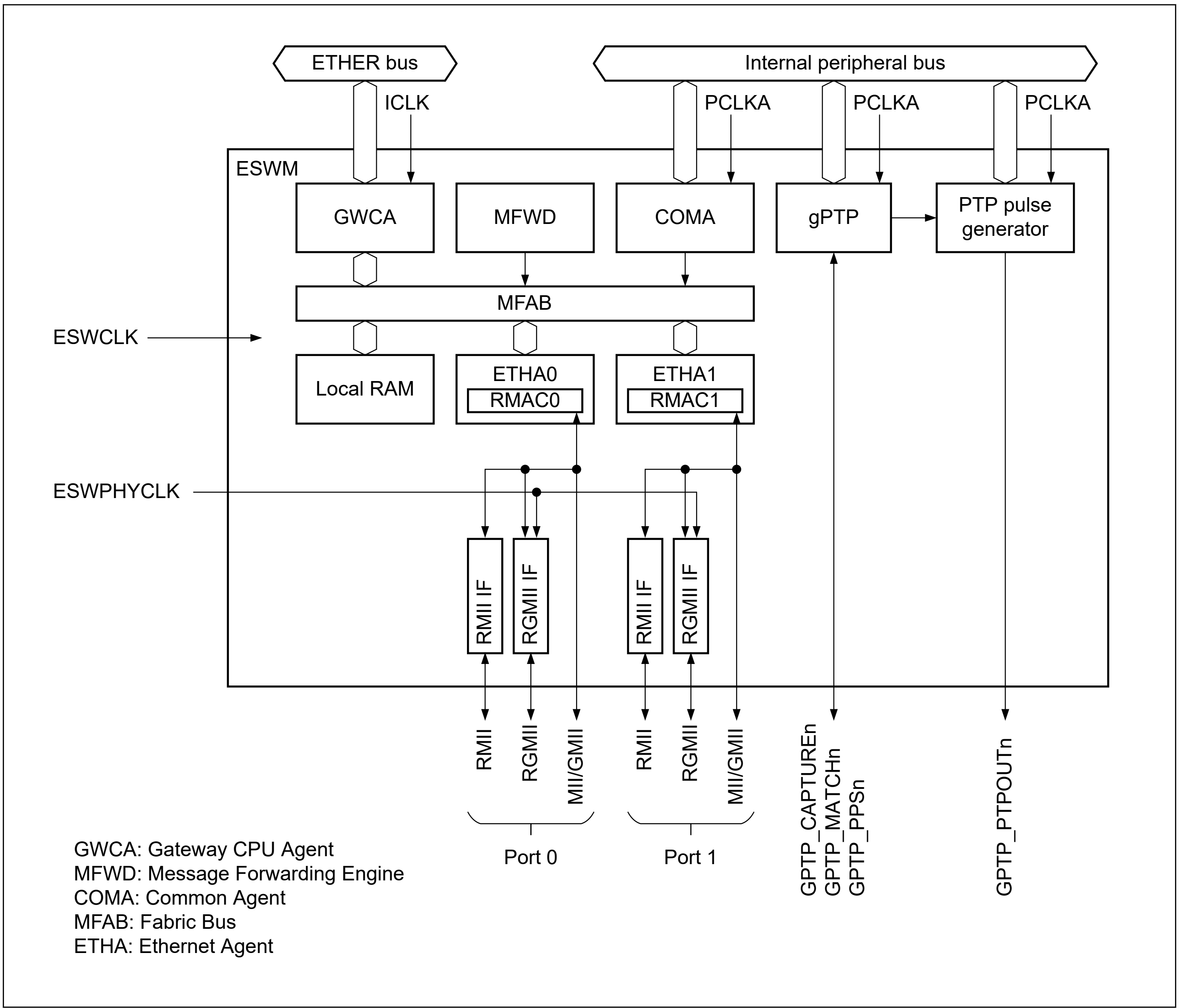


Figure 29.1 ESWM block diagram

Table 29.2 ESWM functional blocks (1 of 2)

Block name	Function
MFWD	Ethernet Message Forwarding Engine (MFWD) is part of ESWM and aims to filter, forward, and route the switch frames. MFWD snoops the frame information from fabric to redistribute them to the agent after filtering, forwarding, and routing mechanisms.
MFAB	Fabric Bus (MFAB) is an internal bus used to exchange data between Ethernet Agent [ETHA], Ethernet Common Agent [COMA], Ethernet CPU Agent [GWCA], local RAM, TAG RAM, pointer RAM, and Ethernet Message Forwarding Engine [MFWD].
ETHA	Ethernet Agent (ETHA) consists of an agent interface module to allow communication within ESWM. ETHA handles data exchange between ESWM and an Ethernet PHY.
RMAC	Ethernet MAC (RMAC) is an Ethernet controller that conforms to the definition of the MAC (Media Access Control) layer for Ethernet in the IEEE 802.3 standard. By connecting with a physical-layer LSI chip (PHY-LSI) that complies with the standard, RMAC can transmit and receive Ethernet (IEEE 802.3) frames. RMAC has a single MAC layer interface.
GWCA	Ethernet CPU Agent (GWCA) is a CPU interface that consists of an agent interface module allowing communication within ESWM. GWCA handles data exchange between ESWM and the GWCPU subsystem. Software is required for the configuration of Gateway AXI bus interface.
COMA	Ethernet Common Agent (COMA) is an agent of ESWM which aims at gathering common functionalities for other agents. COMA handles the APB to SFR bus conversion, the buffer pointer release for rejected frames and the pointer handling for local RAM. Software is required for the configuration of Ethernet Common Agent.

Table 29.2 ESWM functional blocks (2 of 2)

Block name	Function
gPTP	Ethernet Generic PTP timer (GPTP) enables accurate synchronization of the clock in the control system.
plsotim	The pulse output timer module (plsotim), a submodule of gPTP, produces a pulsed output according to the base period of a gPTP timer.

29.2 Register Descriptions

29.2.1 ESWM Interface Control Registers

29.2.1.1 MIIRR : Media-independent Interface Reset Register

Base address: ESWM = 0x403C_8000
ESWM_NS = 0x503C_8000

Offset address: 0x19400

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	RMRST1	RMRST0	—	—	—	—	—	—	RGRST1	RGRST0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RGRST0	RGMII0 Reset 0: Reset 1: Enable	R/W
1	RGRST1	RGMII1 Reset 0: Reset 1: Enable	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	RMRST0	RMII0 Reset 0: Reset 1: Enable	R/W
9	RMRST1	RMII1 Reset 0: Reset 1: Enable	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

29.2.1.2 MIICR0 : Media-independent Interface Control Register 0

Base address: ESWM = 0x403C_8000
ESWM_NS = 0x503C_8000

Offset address: 0x19404

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

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

Bit	Symbol	Function	R/W
1:0	MIISEL[1:0]	Media-independent Interface Select 0 0: MII/GMII 0 1: RGMII 1 0: RMII 1 1: Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	DIVSTP	Clock Divider Stop 0: Enable 1: Stop	R/W
11:9	—	These bits are read as 0. The write value should be 0.	R/W
12	TXCIDE	RGMII TXC Internal Delay Enable 0: TXC internal delay disable 1: TXC internal delay enable	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

29.2.1.3 MIICR1 : Media-independent Interface Control Register 1

Base address: ESWM = 0x403C_8000
ESWM_NS = 0x503C_8000

Offset address: 0x19408

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

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

Bit	Symbol	Function	R/W
1:0	MIISEL[1:0]	Media-independent Interface Select 0 0: MII/GMII 0 1: RGMII 1 0: RMII 1 1: Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
8	DIVSTP	Clock Divider Stop 0: Enable 1: Stop	R/W
11:9	—	These bits are read as 0. The write value should be 0.	R/W
12	TXCIDE	RGMII TXC Internal Delay Enable 0: TXC Internal delay disable 1: TXC Internal delay enable	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

29.2.1.4 MCCESR : Media Clock Capture Event Select Register

Base address: ESWM = 0x403C_8000
ESWM_NS = 0x503C_8000

Offset address: 0x19410

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCCES1	MCCES0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MCCES0	Media Clock 0 Capture Event Select 0: I/O Port 1: ELC	R/W
1	MCCES1	Media Clock 1 Capture Event Select 0: I/O Port 1: ELC	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

29.2.1.5 TASSTSR : TAS Status Monitor Signal Select Register

Base address: ESWM = 0x403C_8000
ESWM_NS = 0x503C_8000

Offset address: 0x19420

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	MSS3[4:0]					—	—	—	MSS2[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	MSS1[4:0]					—	—	—	MSS0[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	MSS0[4:0]	ET_TAS_STA0 Monitor Signal Select Select signal to output ET_TAS_STA0 pin from race_etha0_tas_gate_state[8:0] and race_etha1_tas_gate_state[8:0]	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
12:8	MSS1[4:0]	ET_TAS_STA1 Monitor Signal Select Select signal to output ET_TAS_STA1 pin from race_etha0_tas_gate_state[8:0] and race_etha1_tas_gate_state[8:0]	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
20:16	MSS2[4:0]	ET_TAS_STA2 Monitor Signal Select Select signal to output ET_TAS_STA2 pin from race_etha0_tas_gate_state[8:0] and race_etha1_tas_gate_state[8:0]	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
28:24	MSS3[4:0]	ET_TAS_STA3 Monitor Signal Select Select signal to output ET_TAS_STA3 pin from race_etha0_tas_gate_state[8:0] and race_etha1_tas_gate_state[8:0]	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

29.2.2 Register Descriptions of Related ESWM Interrupts

29.2.2.1 TPEMIMC1 : Error and Monitoring Interrupt Mapping Configuration Register 1

Base address: ESWM = 0x403C_8000
ESWM_NS = 0x503C_8000

Offset address: 0x0004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	CSICM[2:0]			—	—	CSIGM	CSIM	—	CEICM[2:0]			—	—	CEIGM	CEIM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	FSICM[2:0]			—	—	FSIGM	FSIM	—	FEICM[2:0]			—	—	FEIGM	FEIM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FEIM	MFWD Error Interrupt Mapping MFWD Error Interrupt contains the following interrupts: • FWEIS0 • FWEIS1 • FWEIS2 • FWEIS3 • FWEIS4 • FWEIS5 • FWEIS6 • FWEIS7 • FWEIS8 0: MFWD Error Interrupt is mapped to ETHER_FWEI 1: MFWD Error Interrupt is mapped to a specific core depending on TPEMIMC1.FEIGM register	R/W
1	FEIGM	MFWD Error Interrupt GWCA Mapping 0: When TPEMIMC1.FEIM is set to 1, MFWD Error Interrupt is mapped to ETHER_GWDI[TPEMIMC1.FEICM] 1: Setting prohibited	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
6:4	FEICM[2:0]	MFWD Error Interrupt Core Mapping Determine to which pin MFWD Error Interrupt is mapped when mapped to a GWCA	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
8	FSIM	MFWD Status Interrupt Mapping MFWD Status Interrupt contains the following interrupt: ● FWMIS0 [MFWD] 0: MFWD Status Interrupt is mapped to ETHER_FWSI 1: MFWD Status Interrupt is mapped to a specific core depending on TPEMIMC1.FSIGM register	R/W
9	FSIGM	MFWD Status Interrupt GWCA Mapping 0: When TPEMIMC1.FSIM is set to 1, MFWD Status Interrupt is mapped to ETHER_GWDI[TPEMIMC1.FSICM] 1: Setting prohibited	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
14:12	FSICM[2:0]	MFWD Status Interrupt Core Mapping Determine to which pin MFWD Status Interrupt is mapped when mapped to a GWCA	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
16	CEIM	COMA Error Interrupt Mapping COMA Error Interrupt contains the following interrupts: ● CAEIS0 [COMA] ● CAEIS1 [COMA] 0: COMA Error Interrupt is mapped to ETHER_CAEI 1: COMA Error Interrupt is mapped to a specific core depending on TPEMIMC1.CEIGM register	R/W
17	CEIGM	COMA Error Interrupt GWCA Mapping 0: When TPEMIMC1.CEIM is set to 1, COMA Error Interrupt is mapped to ETHER_GWDI[TPEMIMC1.CEICM] 1: Setting prohibited	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
22:20	CEICM[2:0]	COMA Error Interrupt Core Mapping Determine to which pin COMA Error Interrupt is mapped when mapped to a GWCA	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
24	CSIM	COMA Status Interrupt Mapping COMA Status Interrupt contains the following interrupts: ● CAMIS0 [COMA] ● CAMIS1 [COMA] 0: COMA Status Interrupt is mapped to ETHER_CAMI 1: COMA Status Interrupt is mapped to a specific core depending on TPEMIMC1.CSIGM register	R/W
25	CSIGM	COMA Status Interrupt GWCA Mapping 0: When TPEMIMC1.CSIM is set to 1, COMA Status Interrupt is mapped to ETHER_GWDI[TPEMIMC1.CSICM] 1: Setting prohibited	R/W
27:26	—	These bits are read as 0. The write value should be 0.	R/W
30:28	CSICM[2:0]	COMA Status Interrupt Core Mapping Used to decide to which pin COMA Status Interrupt are mapped when mapped to a GWCA.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

29.2.2.2 TPEMIMC2 : Error and Monitoring Interrupt Mapping Configuration Register 2

Base address: ESWM = 0x403C_8000
ESWM_NS = 0x503C_8000

Offset address: 0x0008

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	GEICM0[2:0]			—	—	GEIGM0	GEIM0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	GEIM0	GWCA0 Error Interrupt Mapping GWCA0 Error Interrupt contains the following interrupts: ● GWEIS0 except GWEIS0.TDFES for GWCA0 [GWCA] ● GWEIS1 for GWCA0 [GWCA] ● GWEIS4 for GWCA0 [GWCA] ● GWEIS5 for GWCA0 [GWCA] 0: GWCA0 Error Interrupt is mapped to ETHER_GWEI 1: GWCA0 Error Interrupt is mapped to a specific core depending on TPEMIMC2.GEIGM0 register	R/W
1	GEIGM0	GWCA0 Error Interrupt GWCA Mapping 0: When TPEMIMC2.GEIM0 is set to 1, GWCA0 Error Interrupt is mapped to ETHER_GWDI[TPEMIMC2.GEICM0] 1: Setting prohibited	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
6:4	GEICM0[2:0]	GWCA0 Error Interrupt Core Mapping Determine to which pin GWCA0 Error Interrupt is mapped when mapped to a GWCA.	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

29.2.2.3 TPEMIMC3 : Error and Monitoring Interrupt Mapping Configuration Register 3

Base address: ESWM = 0x403C_8000
ESWM_NS = 0x503C_8000

Offset address: 0x000C

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	ESICM0[2:0]			—	—	ESIGM0	ESIM0	—	EEICM0[2:0]			—	—	EEIGM0	EEIM0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EEIM0	ETHA0 Error Interrupt Mapping ETHA0 Error Interrupt contains the following interrupts: ● EAEIS0 for ETHA0 [ETHA] ● EAEIS1 for ETHA0 [ETHA] ● EAEIS2 for ETHA0 [ETHA] ● MEIS for RMAC0 [RMAC] 0: ETHA0 Error Interrupt is mapped to ETHER_EAEI0 1: ETHA0 Error Interrupt is mapped to a specific core depending on TPEMIMC3.EEIGM0 register	R/W
1	EEIGM0	ETHA0 Error Interrupt GWCA Mapping 0: When TPEMIMC3.EEIM0 is set to 1, ETHA0 Error Interrupt is mapped to ETHER_GWDI[TPEMIMC3.EEICM0] 1: Setting prohibited	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
6:4	EEICM0[2:0]	ETHA0 Error Interrupt Core Mapping Determine to which pin ETHA0 Error Interrupt is mapped when mapped to a GWCA.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	ESIM0	ETHA0 Status Interrupt Mapping ETHA0 Status Interrupt contains the following interrupt: ● MMIS0 for RMAC 0 [RMAC] 0: ETHA0 Status Interrupt is mapped to ETHER_EASI0 1: ETHA0 Status Interrupt is mapped to a specific core depending on TPEMIMC3.ESIGM0 register	R/W
9	ESIGM0	ETHA0 Status Interrupt GWCA Mapping 0: When TPEMIMC3.ESIM0 is set to 1, ETHA0 Status Interrupt is mapped to ETHER_GWDI[TPEMIMC3.ESICM0] 1: Setting prohibited	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
14:12	ESICM0[2:0]	ETHA0 Status Interrupt Core Mapping Determine to which pin ETHA0 Status Interrupt is mapped when mapped to a GWCA.	R/W
31:15	—	These bits are read as 0. The write value should be 0.	R/W

29.2.2.4 TPEMIMC4 : Error and Monitoring Interrupt Mapping Configuration Register 4

Base address: ESWM = 0x403C_8000
ESWM_NS = 0x503C_8000

Offset address: 0x0010

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	ESICM1[2:0]			—	—	ESIGM1	ESIM1	—	EEICM1[2:0]			—	—	EEIGM1	EEIM1
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EEIM1	ETHA1 Error Interrupt Mapping ETHA1 Error Interrupt contains the following interrupt: ● EAEIS0 for ETHA1 [ETHA] ● EAEIS1 for ETHA1 [ETHA] ● EAEIS2 for ETHA1 [ETHA] ● MEIS for RMAC1 [RMAC] 0: ETHA1 Error Interrupt is mapped to ETHER_EAEI1 1: ETHA1 Error Interrupt is mapped to a specific core depending on TPEMIMC4.EEIGM1 register	R/W

Bit	Symbol	Function	R/W
1	EEIGM1	ETHA1 Error Interrupt GWCA Mapping 0: When TPEMIMC4.EEIM1 is set to 1, ETHA1 Error Interrupt is mapped to ETHER_GWDI[TPEMIMC4.EEICM1] 1: Setting prohibited	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
6:4	EEICM1[2:0]	ETHA1 Error Interrupt Core Mapping Determine to which pin ETHA1 Error Interrupt is mapped when mapped to a GWCA.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	ESIM1	ETHA1 Status Interrupt Mapping ETHA1 Status Interrupt contains the following interrupt: ● MMIS0 for RMAC 1 [RMAC] 0: ETHA1 Status Interrupt is mapped to ETHER_EASI1 1: ETHA1 Status Interrupt is mapped to a specific core depending on TPEMIMC4.ESIGM1 register	R/W
9	ESIGM1	ETHA1 Status Interrupt GWCA Mapping 0: When TPEMIMC4.ESIM1 is set to 1, ETHA1 Status Interrupt is mapped to ETHER_GWDI[TPEMIMC4.ESICM1] 1: Setting prohibited	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
14:12	ESICM1[2:0]	ETHA1 Status Interrupt Core Mapping Determine to which pin ETHA1 Status Interrupt is mapped when mapped to a GWCA.	R/W
31:15	—	These bits are read as 0. The write value should be 0.	R/W

29.2.2.5 TPEMIMC6t : Error and Monitoring Interrupt Mapping Configuration Register 6t (t = 0 to 1)

Base address: ESWM = 0x403C_8000
ESWM_NS = 0x503C_8000

Offset address: 0x0080 + 0x04 × t

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	GTSICM0[2:0]			GTSIM0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	GTSIM0	GWCA0 Timestamp Interrupt Mapping GWCA0 Timestamp Interrupt contains the following interrupts: ● GWTDIS.TSDISt for GWCA0 [GWCA] ● GWEIS0.TDFESt for GWCA0 [GWCA] 0: GWCA0 Timestamp Interrupt t is mapped to ETHER_TSDI[t] 1: GWCA0 Timestamp Interrupt t is mapped to ETHER_GWDIn depending on TPEMIMC6t.GTSICM0 setting	R/W
3:1	GTSICM0[2:0]	GWCA0 Timestamp Interrupt Core Mapping Map GWCA0 Timestamp Interrupt t to ETHER_GWDI[TPEMIMC6t.GTSICM0] when TPEMIMC6t.GTSIM0 is set.	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

29.2.2.6 TPEMIMC7t : Error and Monitoring Interrupt Mapping Configuration Register 7t (t = 0 to 63)

Base address: ESWM = 0x403C_8000
ESWM_NS = 0x503C_8000

Offset address: 0x0100 + 0x04 × t

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	GDICM0[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	GDICM0[2:0]	GWCA0 Data Interrupt Core Mapping Map GWCA0 Data Interrupts t to ETHER_GWDI[TPEMIMC7t.GDICM0]. RMAC2 Status Interrupt contains the following interrupts: ● GWDISi.DIS for GWCA0 [GWCA] ● GWEIS2i.DFES for GWCA0 [GWCA] ● GWEIS3.IAOESt for t = 0 to 4 for GWCA0 [GWCA]	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

29.2.2.7 TSIM : Summarized Interrupt Mirroring Register

Base address: ESWM = 0x403C_8000
ESWM_NS = 0x503C_8000

Offset address: 0x0700

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	EIM1	EIM0	—	GIM0	CIM	FIM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FIM	MFWD Interrupt Mirroring 0: No interrupt is set in MFWD 1: An interrupt is set in MFWD	R
1	CIM	COMA Interrupt Mirroring 0: No interrupt is set in COMA agent 1: An interrupt is set in COMA agent	R
2	GIM0	GWCA0 Interrupt Monitoring 0: No interrupt is set in GWCA0 1: An interrupt is set in GWCA0	R
3	—	This bit is read as 0.	R
4	EIM0	ETHA0 Interrupt Monitoring 0: No interrupt is set in ETHA0 1: An interrupt is set in ETHA0	R

Bit	Symbol	Function	R/W
5	EIM1	ETHA1 Interrupt Monitoring 0: No interrupt is set in ETHA1 1: An interrupt is set in ETHA1	R
31:6	—	These bits are read as 0.	R

29.2.2.8 TFIM : MFWD Interrupt Mirroring Register

Base address: ESWM = 0x403C_8000
ESWM_NS = 0x503C_8000

Offset address: 0x0704

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	FWMISIM0	FWEISIM8	FWEISIM7	FWEISIM6	FWEISIM5	FWEISIM4	FWEISIM3	FWEISIM2	FWEISIM1	FWEISIM0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FWEISIM0	FWEIS0 Interrupt Mirroring 0: No interrupt is set in forwarding engine FWEIS0 register 1: An interrupt is set in forwarding engine FWEIS0 register	R
1	FWEISIM1	FWEIS1 Interrupt Mirroring 0: No interrupt is set in forwarding engine FWEIS1 register 1: An interrupt is set in forwarding engine FWEIS1 register	R
2	FWEISIM2	FWEIS2 Interrupt Mirroring 0: No interrupt is set in forwarding engine FWEIS2 register 1: An interrupt is set in forwarding engine FWEIS2 register	R
3	FWEISIM3	FWEIS3 Interrupt Mirroring 0: No interrupt is set in forwarding engine FWEIS3 register 1: An interrupt is set in forwarding engine FWEIS3 register	R
4	FWEISIM4	FWEIS4 Interrupt Mirroring 0: No interrupt is set in forwarding engine FWEIS4 register 1: An interrupt is set in forwarding engine FWEIS4 register	R
5	FWEISIM5	FWEIS5 Interrupt Mirroring 0: No interrupt is set in forwarding engine FWEIS5 register 1: An interrupt is set in forwarding engine FWEIS5 register	R
6	FWEISIM6	FWEIS6 Interrupt Mirroring 0: No interrupt is set in forwarding engine FWEIS6 register 1: An interrupt is set in forwarding engine FWEIS6 register	R
7	FWEISIM7	FWEIS7 Interrupt Mirroring 0: No interrupt is set in forwarding engine FWEIS7 register 1: An interrupt is set in forwarding engine FWEIS7 register	R
8	FWEISIM8	FWEIS8 Interrupt Mirroring 0: No interrupt is set in forwarding engine FWEIS8 register 1: An interrupt is set in forwarding engine FWEIS8 register	R
9	FWMISIM0	FWMIS0 Interrupt Mirroring 0: No interrupt is set in forwarding engine FWMIS0 register 1: An interrupt is set in forwarding engine FWMIS0 register	R
31:10	—	These bits are read as 0.	R

29.2.2.9 TCIM : COMA Interrupt Mirroring Register

Base address: ESWM = 0x403C_8000
ESWM_NS = 0x503C_8000

Offset address: 0x0708

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	CAMISIM1	CAMISIM0	CAEISIM1	CAEISIM0	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0.	R
1	CAEISIM0	CAEIS0 Interrupt Mirroring 0: No interrupt is set in COMA agent CAEIS0 register 1: An interrupt is set in COMA agent CAEIS0 register	R
2	CAEISIM1	CAEIS1 Interrupt Mirroring 0: No interrupt is set in COMA agent CAEIS1 register 1: An interrupt is set in COMA agent CAEIS1 register	R
3	CAMISIM0	CAMIS0 Interrupt Mirroring 0: No interrupt is set in COMA agent CAMIS0 register 1: An interrupt is set in COMA agent CAMIS0 register	R
4	CAMISIM1	CAMIS1 Interrupt Mirroring 0: No interrupt is set in COMA agent CAMIS1 register 1: An interrupt is set in COMA agent CAMIS1 register	R
31:5	—	These bits are read as 0.	R

29.2.2.10 TGIM0 : GWCA0 Interrupt Mirroring Register

Base address: ESWM = 0x403C_8000
ESWM_NS = 0x503C_8000

Offset address: 0x0710

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	GWEISIM5	GWEISIM4	GWEISIM3	GWEISIM2	GWEISIM1	GWEISIM0	GWTSDISIM	GWDISIM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	GWDISIM	GWDIS Interrupt Mirroring 0: No interrupt is set in GWCA0 GWDIS register 1: An interrupt is set in GWCA0 GWDIS register	R
1	GWTSDISIM	GWTSDIS Interrupt Mirroring 0: No interrupt is set in GWCA0 GWTSDIS register 1: An interrupt is set in GWCA0 GWTSDIS register	R

Bit	Symbol	Function	R/W
2	GWEISIM0	GWEIS0 Interrupt Mirroring 0: No interrupt is set in GWCA0 GWEIS0 register 1: An interrupt is set in GWCA0 GWEIS0 register	R
3	GWEISIM1	GWEIS1 Interrupt Mirroring 0: No interrupt is set in GWCA0 GWEIS1 register 1: An interrupt is set in GWCA0 GWEIS1 register	R
4	GWEISIM2	GWEIS2 Interrupt Mirroring 0: No interrupt is set in GWCA0 GWEIS2 register 1: An interrupt is set in GWCA0 GWEIS2 register	R
5	GWEISIM3	GWEIS3 Interrupt Mirroring 0: No interrupt is set in GWCA0 GWEIS3 register 1: An interrupt is set in GWCA0 GWEIS3 register	R
6	GWEISIM4	GWEIS4 Interrupt Mirroring 0: No interrupt is set in GWCA0 GWEIS4 register 1: An interrupt is set in GWCA0 GWEIS4 register	R
7	GWEISIM5	GWEIS5 Interrupt Mirroring 0: No interrupt is set in GWCA0 GWEIS5 register. 1: An interrupt is set in GWCA0 GWEIS5 register.	R
31:8	—	These bits are read as 0.	R

29.2.2.11 TEIM0 : ETHA0 Interrupt Mirroring Register

Base address: ESWM = 0x403C_8000
ESWM_NS = 0x503C_8000

Offset address: 0x0720

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	MMISIM	MEISIM	EAEISIM2	EAEISIM1	EAEISIM0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EAEISIM0	EAEIS0 Interrupt Mirroring 0: No interrupt is set in ETHA0 EAEIS0 register 1: An interrupt is set in ETHA0 EAEIS0 register	R
1	EAEISIM1	EAEIS1 Interrupt Mirroring 0: No interrupt is set in ETHA0 EAEIS1 register 1: An interrupt is set in ETHA0 EAEIS1 register	R
2	EAEISIM2	EAEIS2 Interrupt Mirroring 0: No interrupt is set in ETHA0 EAEIS2 register 1: An interrupt is set in ETHA0 EAEIS2 register	R
3	MEISIM	MEIS Interrupt Mirroring 0: No interrupt is set in ETHA0 RMAC MEIS register 1: An interrupt is set in ETHA0 RMAC MEIS register	R
4	MMISIM	MMIS0 Interrupt Mirroring 0: No interrupt is set in ETHA0 RMAC MMIS0 register 1: An interrupt is set in ETHA0 RMAC MMIS0 register	R
31:5	—	These bits are read as 0.	R

29.2.2.12 TEIM1 : ETHA1 Interrupt Mirroring Register

Base address: ESWM = 0x403C_8000
ESWM_NS = 0x503C_8000

Offset address: 0x0724

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	MMISIM	MEISIM	EAEISIM2	EAEISIM1	EAEISIM0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EAEISIM0	EAEIS0 Interrupt Mirroring 0: No interrupt is set in ETHA1 EAEIS0 register 1: An interrupt is set in ETHA1 EAEIS0 register	R
1	EAEISIM1	EAEIS1 Interrupt Mirroring 0: No interrupt is set in ETHA1 EAEIS1 register 1: An interrupt is set in ETHA1 EAEIS1 register	R
2	EAEISIM2	EAEIS2 Interrupt Mirroring 0: No interrupt is set in ETHA1 EAEIS2 register 1: An interrupt is set in ETHA1 EAEIS2 register	R
3	MEISIM	MEIS Interrupt Mirroring 0: No interrupt is set in ETHA1 RMAC MEIS register 1: An interrupt is set in ETHA1 RMAC MEIS register	R
4	MMISIM	MMIS0 Interrupt Mirroring 0: No interrupt is set in ETHA1 RMAC MMIS0 register 1: An interrupt is set in ETHA1 RMAC MMIS0 register	R
31:5	—	These bits are read as 0.	R

29.2.3 ESWM Core Registers

See the following sections:

- section 30.3. MFWD Register Descriptions
- section 31.3. COMA Register Descriptions
- section 32.3. ETHA Register Descriptions
- section 33.4. RMAC Register Descriptions
- section 34.3. GWCA Register Descriptions

29.2.4 gPTP and PTP Pulse Generator Registers

See the following sections:

- section 35.3. gPTP Register Descriptions
- section 35.6.3. Plsotim Register Descriptions

29.3 Register Utilization

29.3.1 Software Flows

Restrictions on software flows are as follows:

- SW: ESWM registers can only be written using the flows specified in this section

- SW: All the software flows described in this section can only be used sequentially.

29.3.1.1 Software Flow Legend

Figure 29.2 shows the software flow legend.

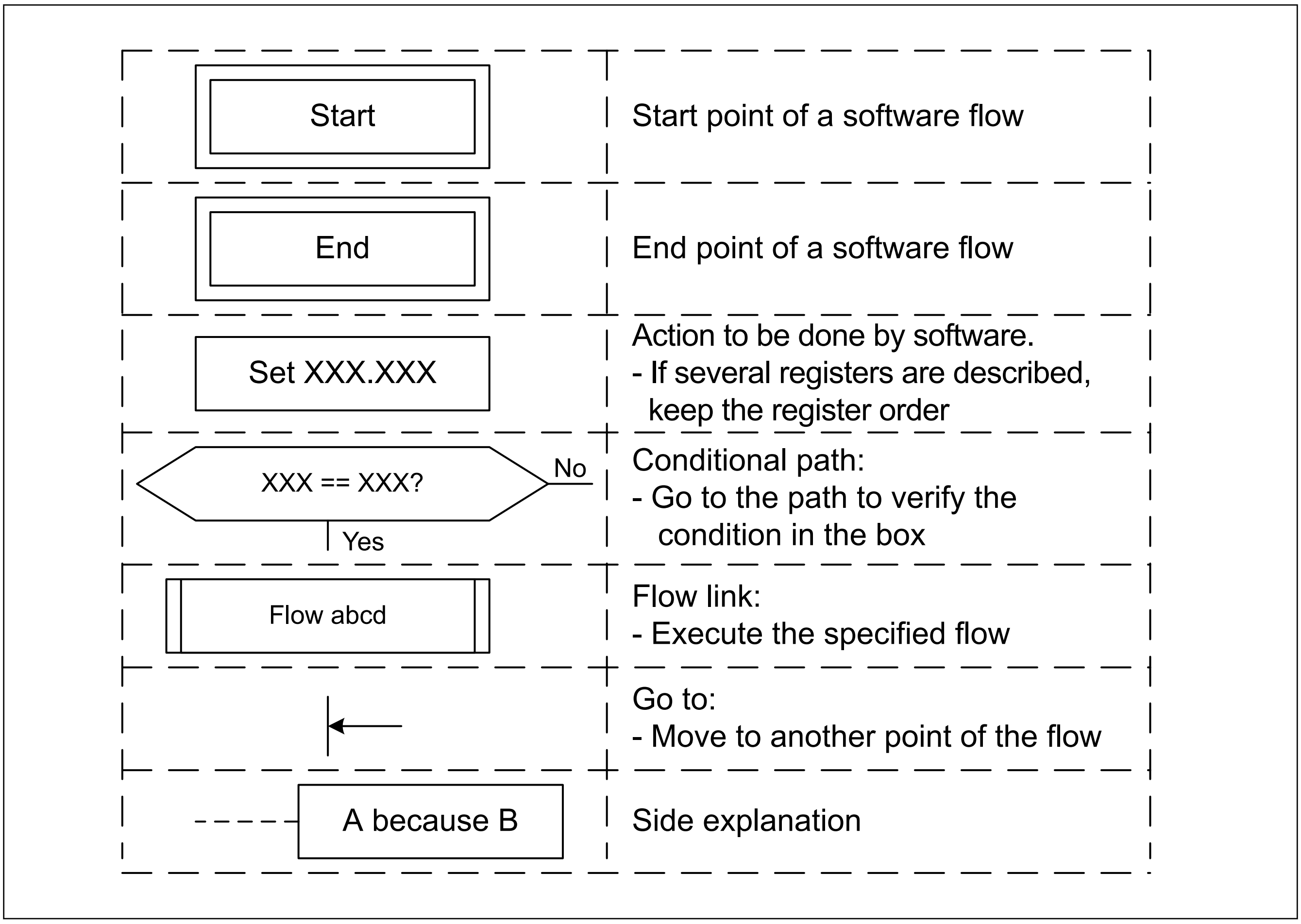


Figure 29.2 Software flow legend

29.3.1.2 Switch Initialization Flow

Figure 29.3 shows the switch initialization flow.

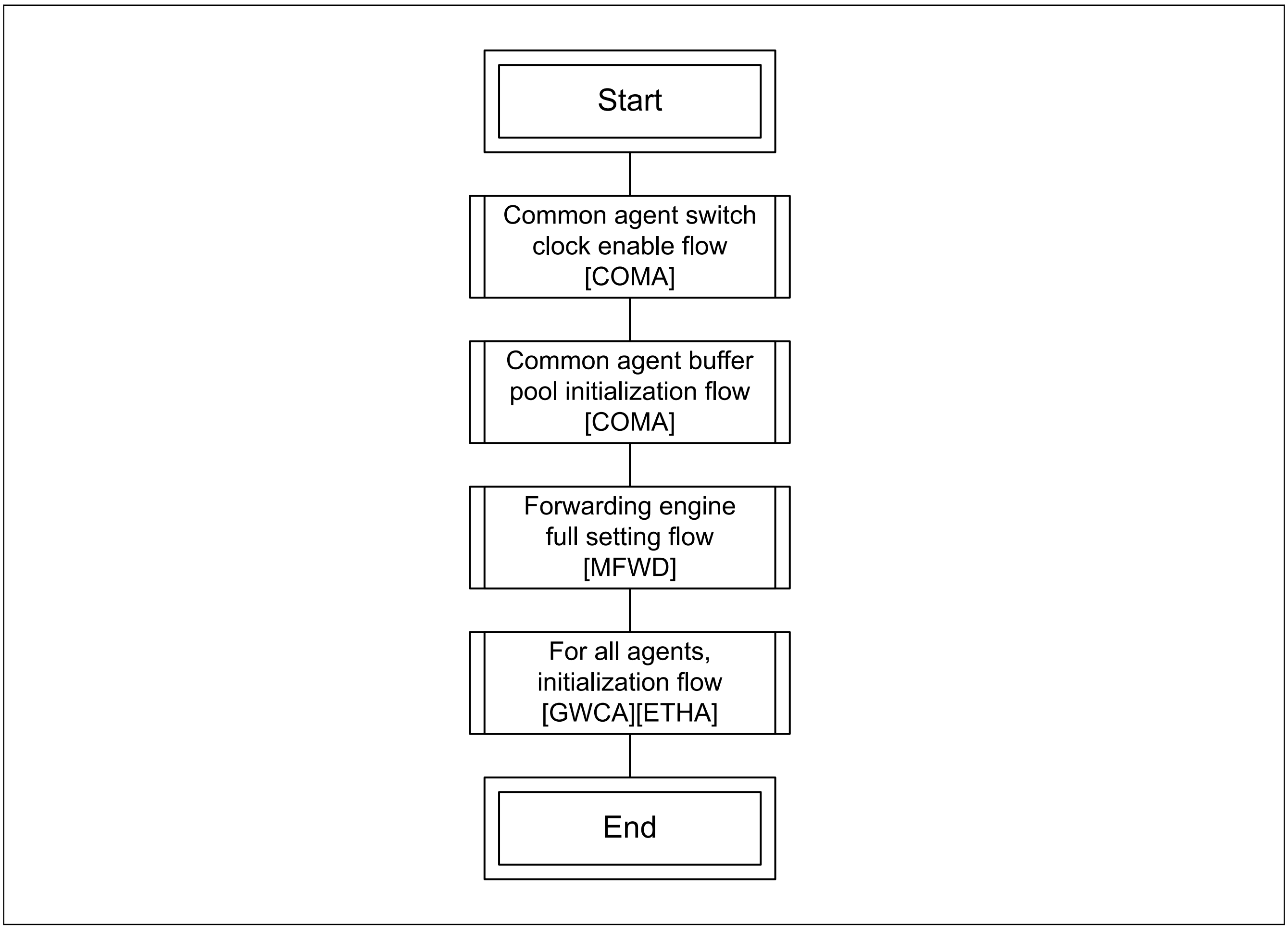


Figure 29.3 Switch initialization flow

29.3.1.3 Switch Disable Flow

Figure 29.4 shows the switch disable flow.

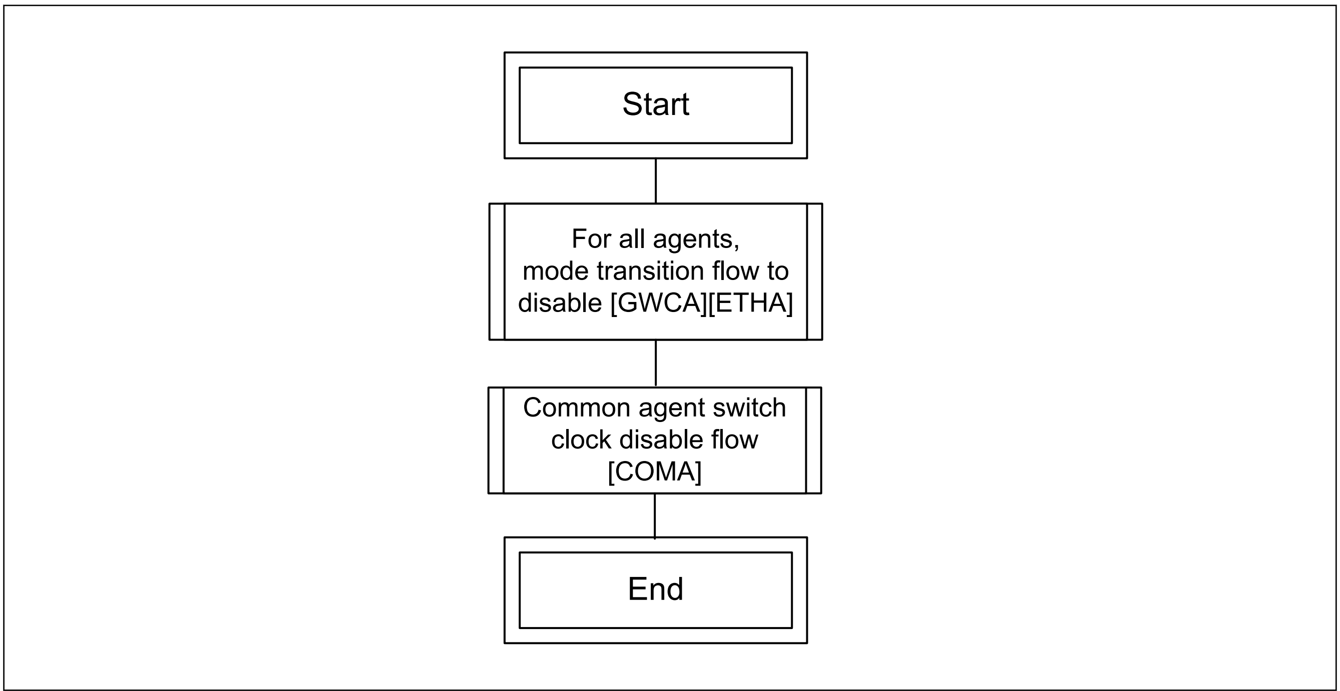


Figure 29.4 Switch disable flow

29.3.1.4 Switch Reset Flow

Figure 29.5 shows the switch reset flow.

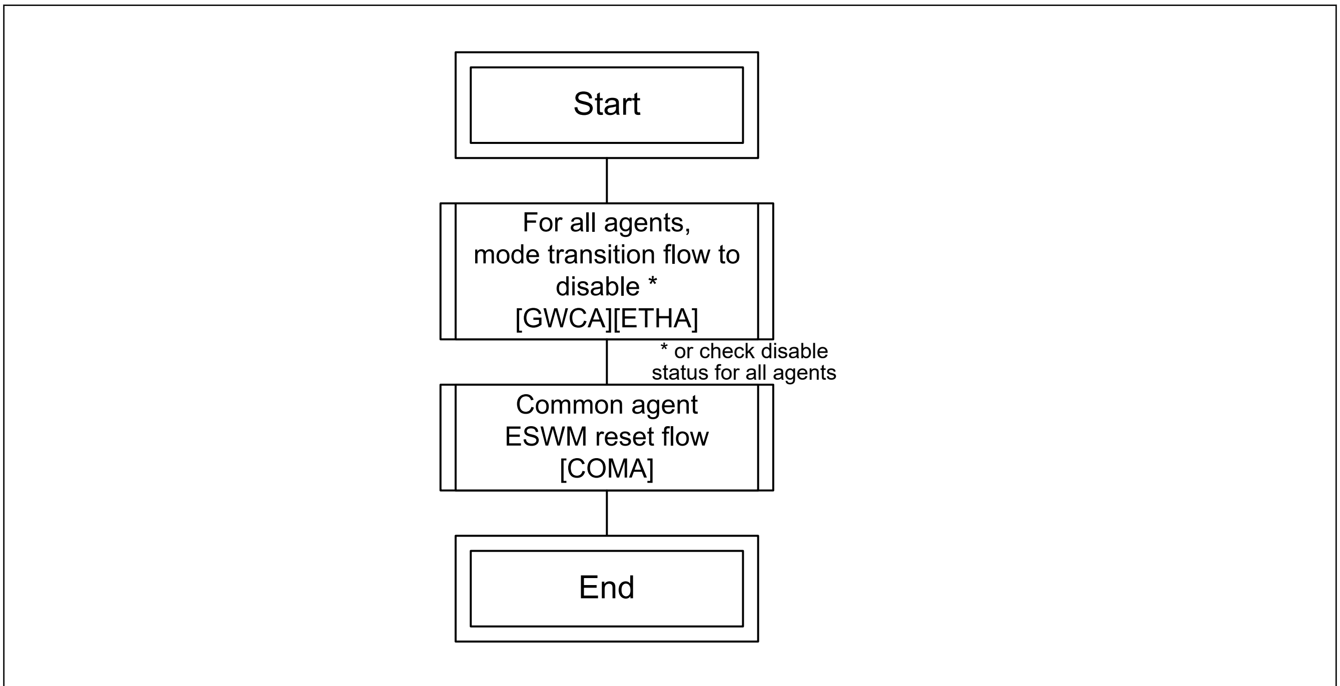


Figure 29.5 Switch reset flow

29.3.1.5 Switch Emergency Reset Flow

Figure 29.6 shows the switch emergency reset flow used when [ETHA] is locked.

For example, in the case of MII, if an error occurs in the PHY and the MII PHY TX clock stops, the switch cannot send data and cannot release the pointer (locked on [ETHA] Operation mode). The state clears when the MII PHY TX clock is restarted, but use emergency reset if required.

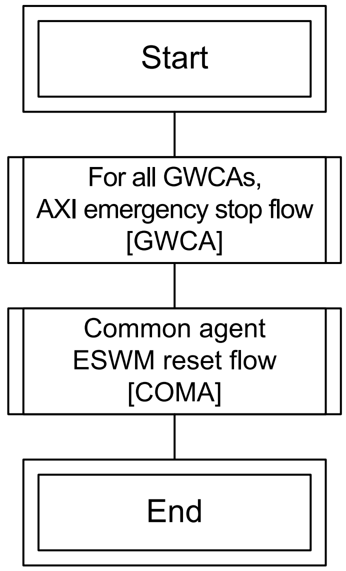


Figure 29.6 Switch emergency reset flow

29.3.1.6 Register Writable Without Software Flow

Table 29.3 describes the registers writable without software flow and in which mode they can be written.

Table 29.3 Writable without software flow registers (✓ = writable, × = not writable)

Register name	Writable anytime
TPEMIMC0	✓
TPEMIMC1	✓
TPEMIMC2	✓
TPEMIMC3	✓
TPEMIMC4	✓
TPEMIMC5	✓
TPEMIMC6t	✓
TPEMIMC7t	✓
TPICVC0	✓
TPICVC1	✓
TPICVC2	✓
TPRCVC	✓
TCVUC	✓

29.4 Functional Descriptions

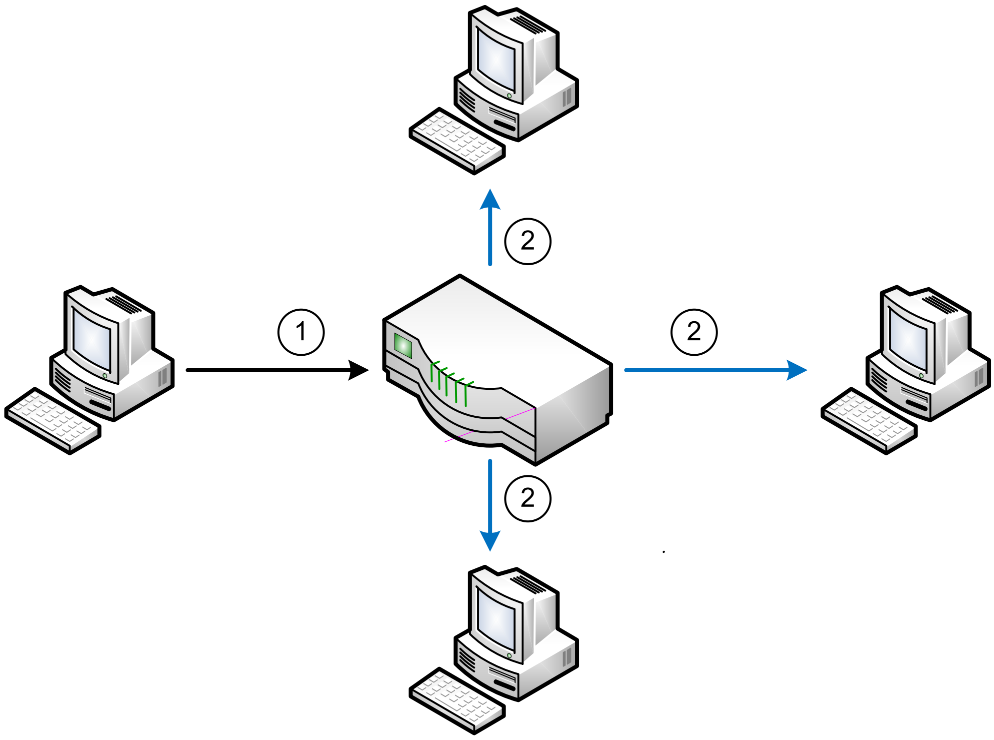
29.4.1 ESWM Forwarding Modes

29.4.1.1 Hub

29.4.1.1.1 Background

A hub is an ethernet network interconnection system (OSI) which only works on layer 1. It does not consider any kind of addressing and, consequently, is not aware of the kind of traffic that goes through it and can create security issues.

Figure 29.7 shows the hub operation.



1	The hub receives a packet from a port
2	The hub copies the packet to all ports except the ingress port

Figure 29.7 Hub operation

Note: In Hub mode, the internal port [GWCA] behaves like an ethernet port. In the following configuration, the internal port receives all frames by AXI descriptor queue number 0 and transmits all frames by AXI descriptor queue number 1.

29.4.1.1.2 Settings

Table 29.4 describes the setting required for ESWM in Hub mode. Specify this setting using the switch initialization flow described in section 29.3.1.2. Switch Initialization Flow. Any register not described in the table should be kept to its initial value, so it does not need to be set. After using the switch initialization flow, any used port should be moved to Operation mode when required.

Table 29.4 Hub settings (1 of 2)

Register name	Setting value	Description
Message Forwarding Engine [MFWD]		
FWPBFCi.PBDV	All 1s except for FWPBFCi.PBDV which is set to 0	All ports are set to send their incoming frames to all other ports
Ethernet Agent [ETHA]		
EAIRC	0	All frames are directed to descriptor queue 0
EATDQDCq.DQD	If q = 0, set to 512 Else set to 0	There is no QoS, so all frames are stored in priority 0 queue. As a result, all the output descriptor queue memory is reserved for queue 0.
EAVCC.VEM	011b	Allows SC-tagged frames and C-tagged frames to go through transparently
EATTFC.UT	0	Allows any TAG formats to go through even the unsupported ones

Table 29.4 Hub settings (2 of 2)

Register name	Setting value	Description
Ethernet MAC [RMAC]		
MPIC.PIS	User-defined	Set the PHY interface type
MPIC.LSC	User-defined	Set the PHY speed
MRAFC	03C703C7H	Set MAC reception in promiscuous mode
Gateway CPU Agent [GWCA]		
GWIRC	0	All frames are directed to descriptor queue 0
GWRDQDCq.DQD	If q = 0, set to 512 Else set to 0	There is no QoS, so all frames are stored in priority 0 queue. As a result, all the output descriptor queue memory is reserved for queue 0.
GWVCC.VEM	011	Allows SC-tagged frames and C-tagged frames to go through transparently
GWTTFC.UT	0	Allows any TAG formats to go through even the unsupported ones
GWDCBAC0/1	User-defined	Set the LINKFIX table base address
GWDCCi	For i = 1: ● Set DQT to 1 ● Set OSID to user-defined. Else: ● Set OSID to user-defined.	Queue 0 is used for data reception and queue 1 is used for data transmission. All other queues are not used.

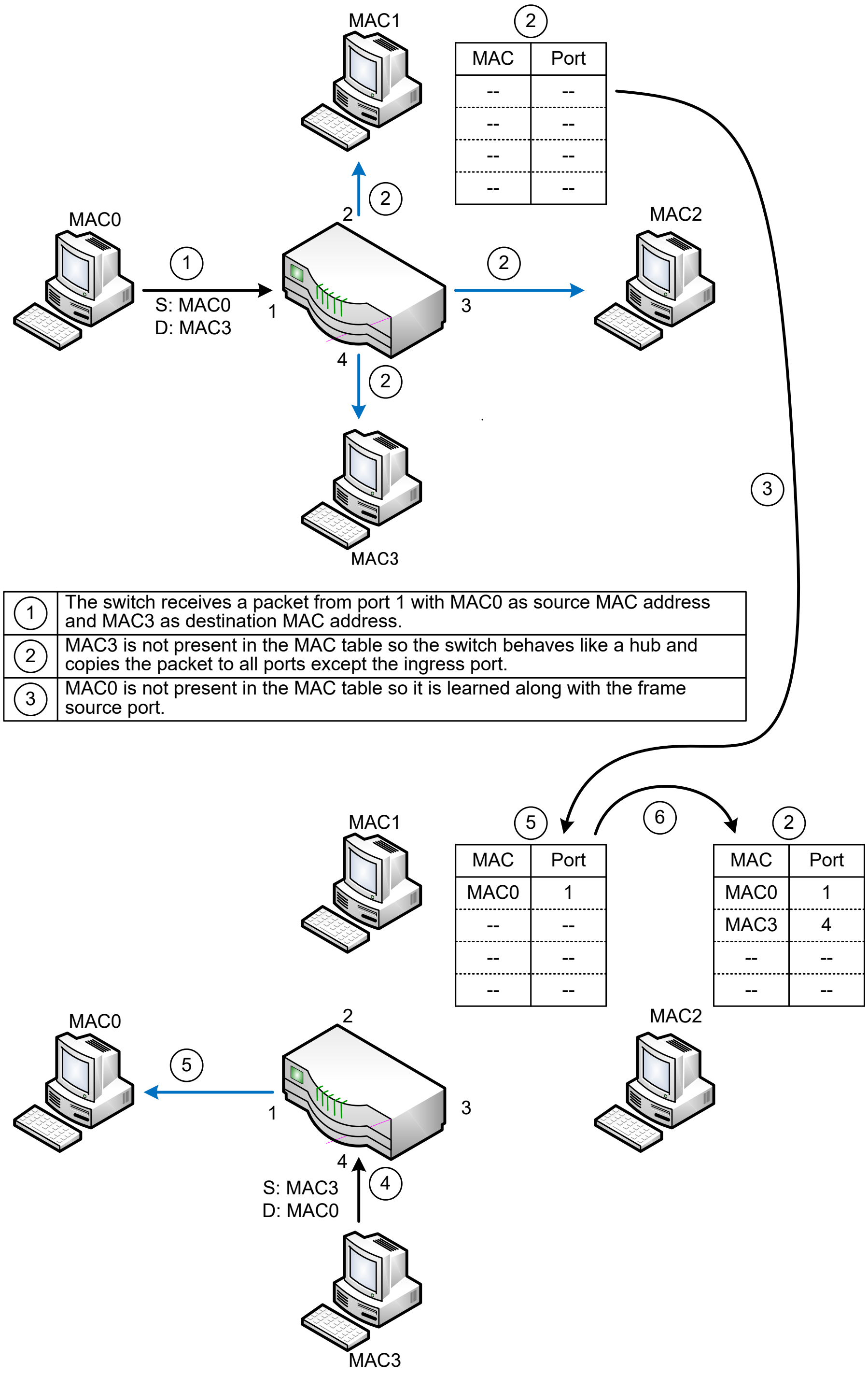
29.4.1.2 Layer 2 Switch

29.4.1.2.1 Background

A layer 2 switch is an ethernet network interconnection system (OSI) which only works on layer 2 using MAC addresses. MAC addresses are used to filter and forward messages. A layer 2 switch has three main functions:

- Address learning: A switch can learn MAC addresses by looking at received frame source MAC addresses
- Address aging: A switch can suppress a MAC address if a frame from the corresponding source has not been received since a given period (usually 5 minutes)
- Frame forward/filter: A switch can forward or filter a frame based on its destination MAC address.

Figure 29.8 shows the layer 2 switch operation.



1	The switch receives a packet from port 1 with MAC0 as source MAC address and MAC3 as destination MAC address.
2	MAC3 is not present in the MAC table so the switch behaves like a hub and copies the packet to all ports except the ingress port.
3	MAC0 is not present in the MAC table so it is learned along with the frame source port.

4	The switch receives a packet from port 4 with MAC3 as source MAC address and MAC0 as destination MAC address.
5	MAC0 is present in the MAC table along with port number 1 so the frame is only forwarded to port 1.
6	MAC3 is not present in the MAC table so it is learned along with the frame source port.

Figure 29.8 Layer 2 switch operation

29.4.1.2.2 Settings

Table 29.5 describes the setting required for ESWM in layer 2 switch mode. Specify this setting using the switch initialization flow described in section 29.3.1.2. Switch Initialization Flow. Any register not described in the table should be kept to its initial value, so it does not need to be set. After using the switch initialization flow, any used port should be moved to Operation mode when required.

Table 29.5 Layer 2 switch settings

Register name	Setting value	Explanation
Message Forwarding Engine [MFWD]		
FWPBFCi.PBDV	All 1s except for FWPBFCi.PBDV which is set to 0	All ports are set to send their incoming frames to all other ports
FWPCi0.MACDSA	1	MAC destination search occurs for all ports
FWPCi0.MACSSA	1	MAC source search occurs for all ports
FWPCi0.MACHLA	1	MAC source hardware learning is enabled for all ports
FWPCi0.MACHMA	1	MAC source hardware migration is enabled for all ports
FWCLPRC.USMACLF	User-defined	Unknown source MAC addresses can be sent to CPU for learning monitoring
FWCLPTC.LPCSD	2	Set the learning path queue number to 2
FWCLPTC.LPCS	User-defined	Select which CPU to be used for learning monitoring
Ethernet Agent [ETHA]		
EAIRC	0	All frames are directed to descriptor queue 0
EATDQDCq.DQD	If q = 0, set to 512 Else set to 0	There is no QoS, so all frames are stored in priority 0 queue. As a result, all the output descriptor queue memory is reserved for queue 0.
EAVCC.VEM	011b	Allows SC-tagged frames and C-tagged frames to go through transparently
EATTFC.UT	0	Allows any TAG formats to go through even the unsupported ones
Ethernet MAC [RMAC]		
MPIC.PIS	User-defined	Set the PHY interface type
MPIC.LSC	User-defined	Set the PHY speed
MRAFC	0x03C703C7	Set MAC reception in promiscuous mode
Gateway CPU Agent [GWCA]		
GWIRC	0	All frames are directed to descriptor queue 0
GWRDQDCq.DQD	If q = 0, set to 512 Else set to 0	There is no QoS, so all frames are stored in priority 0 queue. As a result, all the output descriptor queue memory is reserved for queue 0.
GWVCC.VEM	011b	Allows SC-tagged frames and C-tagged frames to go through transparently
GWTTFC.UT	0	Allows any TAG formats to go through even the unsupported ones
GWDCBAC0/1	User-defined	Set the LINKFIX table base address
GWDCCi	For i = 1: ● Set DQT to 1 ● Set OSID to user-defined. Else: ● Set OSID to user-defined.	Queue 0 is used for data reception, queue 1 is used for data transmission and queue 2 is used for learning monitoring. All other queues are not used. Note: For learning monitoring, use extended descriptors.

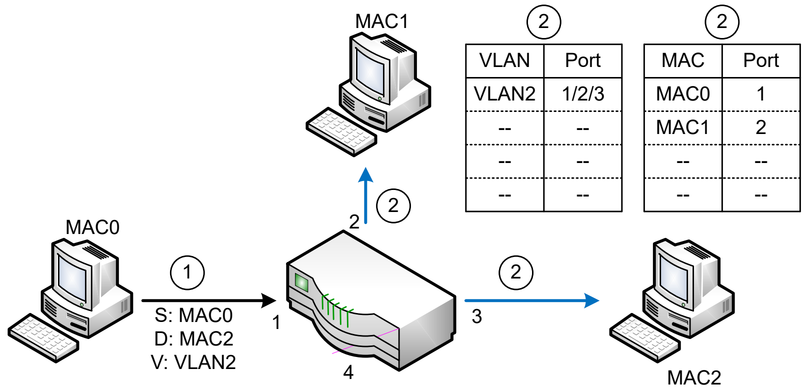
Note: Any MAC address learned by hardware coming from an internal port [GWCA] is always learned along with AXI descriptor queue 0. As a result, always reserve AXI descriptor queue 0 for hardware learned MAC addresses that correspond to frames reception queue.

29.4.1.3 VLAN Aware Layer 2 Switch

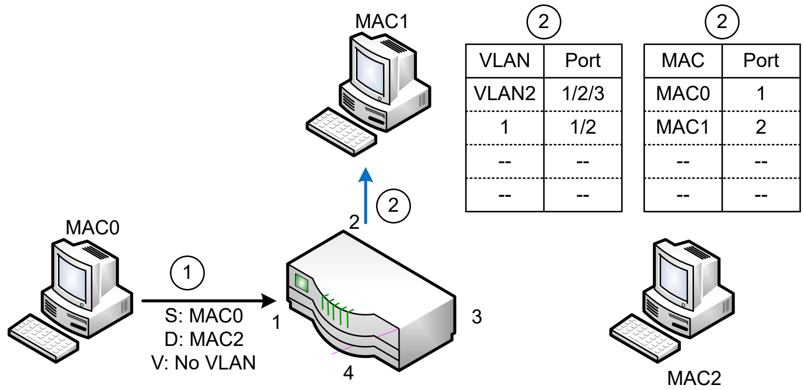
29.4.1.3.1 Background

A VLAN aware layer 2 switch is an ethernet network interconnection system (OSI) which works on layer 2 using MAC addresses and can reduce its broadcast domains in subdomains using VLAN IDs. A VLAN aware layer 2 switch has the same functions as a layer 2 switch but can also handle VLANs.

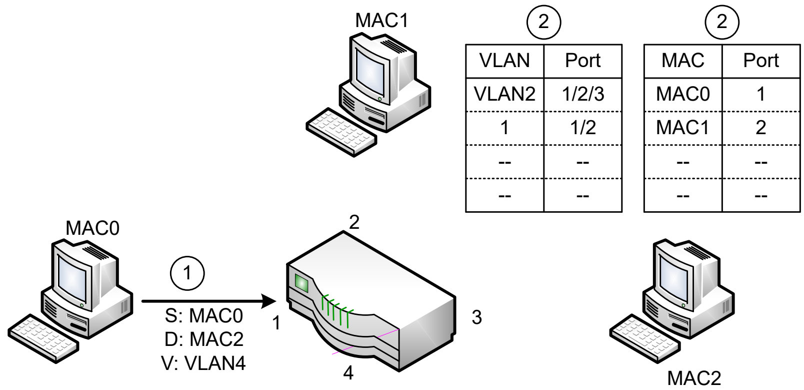
Figure 29.9 shows the VLAN aware layer 2 switch operation.



1	The switch receives a packet from port 1 with MAC0 as source MAC address, MAC2 as destination MAC address and VLAN2 as VLAN.
2	MAC2 is not present in the MAC table so the switch copies the packet to all ports present in VLAN2 domain (port 1/2/3) except the ingress port.



1	The switch receives a packet from port 1 with MAC0 as source MAC address, MAC2 as destination MAC address and no VLAN.
2	MAC2 is not present in the MAC table so the switch copies the packet to all ports present in default VLAN (1) domain (port 1/2) except the ingress port.



1	The switch receives a packet from port 1 with MAC0 as source MAC address, MAC2 as destination MAC address and VLAN4 as VLAN.
2	VLAN4 is unknown so the packet is rejected.

Figure 29.9 VLAN aware layer 2 switch operation

29.4.1.3.2 Settings

Table 29.6 describes the setting required for ESWM in VLAN aware layer 2 switch mode. Specify this setting using the switch initialization flow described in section 29.3.1.2. Switch Initialization Flow. Any register not described in the table should be kept to its initial value, so it does not need to be set. After using the switch initialization flow, any used port should be moved to Operation mode when required.

Table 29.6 VLAN aware layer 2 switch settings

Register name	Setting value	Explanation
Message forwarding engine [MFWD]		
FWGC.SVM	01	Set ESWM in C-TAG mode
FWPBFCi.PBDV	All 1s except for FWPBFCi.PBDV which is set to 0	All ports are set to send their incoming frames to all other ports
FWPCi0.MACDSA	1	MAC destination search occurs for all ports
FWPCi0.MACSSA	1	MAC source search occurs for all ports
FWPCi0.MACHLA	1	MAC source hardware learning is enabled for all ports
FWPCi0.MACHMA	1	MAC source hardware migration is enabled for all ports
FWPCi0.VLANSA	1	VLAN search occurs for all ports
FWPCi0.VLANRU	1	Unknown VLANs are rejected
FWCLPRC.USMACLF	User-defined	Unknown source MAC addresses can be sent to CPU for learning monitoring
FWCLPRC.UVLANLF	User-defined	Unknown VLANs can be sent to CPU for learning
FWCLPTC.LPCSD	2	Set the learning path queue number to 2
FWCLPTC.LPCS	User-defined	Select which CPU to be used for learning monitoring
FWVLANTL0/1/2/30/4	User-defined	New VLANs can be added to the VLAN table. For details, see Table 29.7.
Ethernet agent [ETHA]		
EATDQDCq.DQD	User-defined	The number of descriptors in descriptor queues can be set
EAVCC.VEM	011	Allows SC-tagged frames and C-tagged frames to go through transparently
EAVTC.CTV	1	Set the default VLAN value to 1
EATTFC.UT	0	Allows any TAG formats to go through even the unsupported ones
RMAC [RMAC]		
MPIC.PIS	User-defined	Set the PHY interface type
MPIC.LSC	User-defined	Set the PHY speed
MRAFC	03C703C7H	Set MAC reception in promiscuous mode
GWCA [GWCA]		
GWRDQDCq.DQD	User-defined	The number of descriptors in descriptor queues can be set
GWVCC.VEM	011	Allows SC-tagged frames and C-tagged frames to go through transparently
GWVTC.CTV	1	Set the default VLAN value to 1
GWTTFC.UT	0	Allows any TAG formats to go through even the unsupported ones
GWDCBAC0/1	User-defined	Set the LINKFIX table base address
GWDCCi	For i = 1: ● Set DQT to 1 ● Set OSID to user-defined. Else: ● Set OSID to user-defined.	Queue 0 is used for data reception, queue 1 is used for data transmission and queue 2 is used for learning monitoring. All other queues are not used. Note: For learning monitoring, use extended descriptors.

Table 29.7 VLAN learning in VLAN table

Field name VLAN	Setting value	Explanation
SL	0	No secure entry used
HLD	0	Hardware learning for MAC addresses not disabled for any VLAN
SLV	Set to 1 for ports included in VLAN	Only authorize to receive corresponding VLAN from ports included in this VLAN
DV	Set to 1 for ports included in VLAN	Only authorize to send corresponding VLAN to ports included in this VLAN
CSD	0	Frames received by CPUs on sub-destination 0
CME	0	No mirroring
EME	0	No mirroring
IPU	0	No frame priority update
IPV	0	No frame priority update

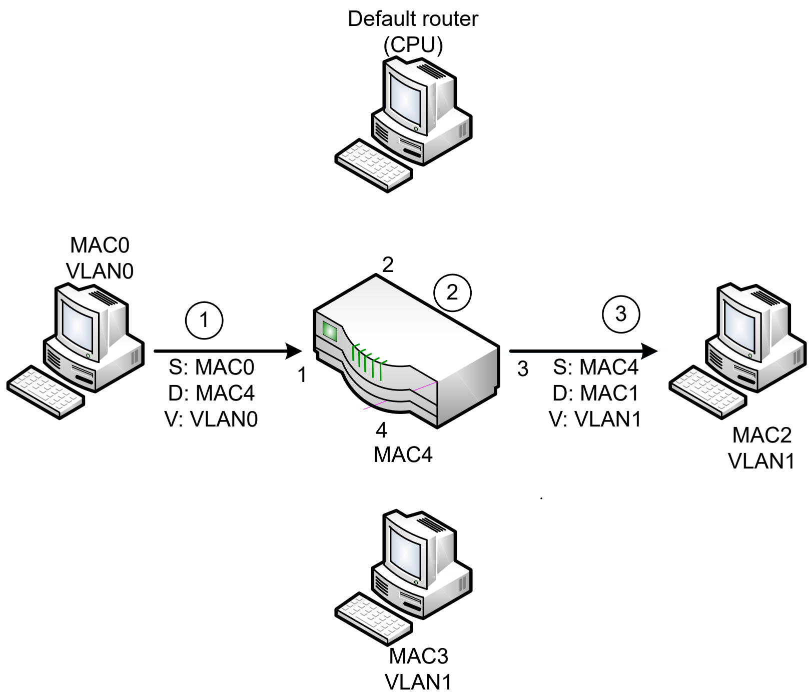
Note: When both VLAN and destination MAC addresses are found in forwarding tables and corresponding frame is forwarded to an internal port, the AXI descriptor queue (CPU sub-destination) chosen for forwarding is the one set in the MAC table. In this example, because both are set to 0, it raises no issue but, when AXI descriptor queues are used for virtualization, it should be taken in account during system design.

29.4.1.4 Layer 3 Switch

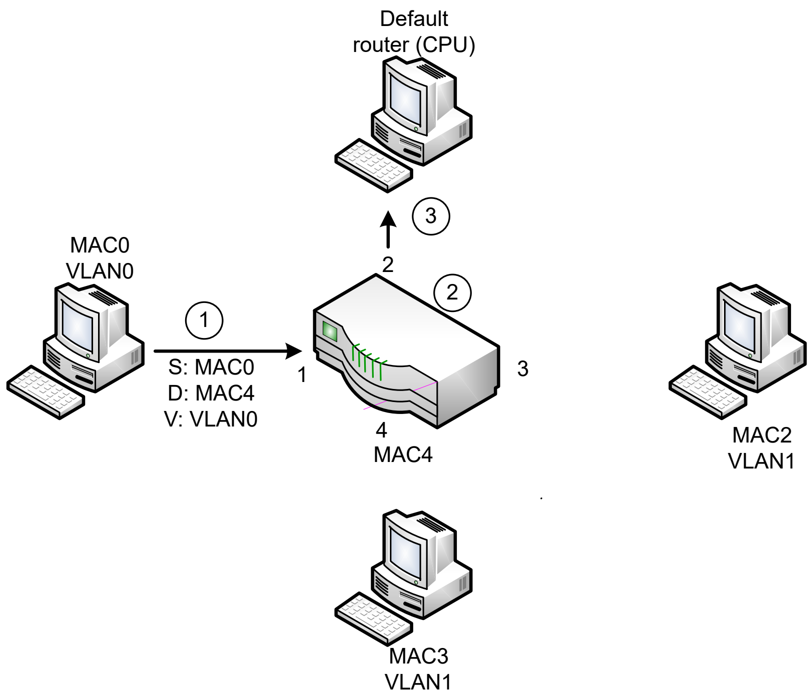
29.4.1.4.1 Background

A layer 3 switch is an ethernet network interconnection system (OSI) which works on layer 3 using IP addresses. Because ESWM can only perform perfect matches by stream forwarding, when developing a layer 3 switch, it is recommended to use perfect filters [FWD].

Figure 29.10 shows the layer 3 switch operation.



1	The switch receives a packet from port 1 with MAC0 as source MAC address, MAC4 as destination MAC address and VLAN0 as VLAN.
2	The packet IP address matches the cascade filter and should be routed to port 3.
3	The switch updates MAC destination to MAC1, MAC source to MAC4 and VLAN to VLAN1 and sends the frame to port 3.



1	The switch receives a packet from port 1 with MAC0 as source MAC address, MAC4 as destination MAC address and VLAN0 as VLAN.
2	The packet IP address is unknown so routing cannot happen.
3	The switch forwards the frame to the default router (CPU) without update for routing.

Figure 29.10 Layer 3 switch operation

29.4.1.4.2 Settings

Table 29.8 describes the setting required for ESWM in layer 3 switch mode. Specify this setting using the switch initialization flow described in section 29.3.1.2. Switch Initialization Flow. Any register not described in the table should be kept to its initial value, so it does not need to be set. After using the switch initialization flow, any used port should be moved to Operation mode when required.

Table 29.8 Layer 3 switch settings

Register name	Setting value	Explanation
Forwarding engine [MFWD]		
FWGC.SVM	01	Set ESWM in C-TAG mode
FWPBFCi.PBDV	All 1s except for FWPBFCi.PBDV which is set to 0	All ports are set to send their incoming frames to all other ports
FWPCi0.MACDSA	1	MAC destination search occurs for all ports
FWPCi0.MACSSA	1	MAC source search occurs for all ports
FWPCi0.MACHLA	1	MAC source hardware learning is enabled for all ports
FWPCi0.MACHMA	1	MAC source hardware migration is enabled for all ports
FWPCi0.VLANSA	1	VLAN search occurs for all ports
FWCLPRC.USMACLF	User-defined	Unknown source MAC addresses can be sent to CPU for learning monitoring
FWCLPRC.UVLANLF	1	Unknown VLANs are sent to CPU for learning
FWCLPTC.LPCSD	2	Set the learning path queue number to 2
FWCLPTC.LPCS	User-defined	Select which CPU to be used for learning monitoring
FWVLANTL0/1/2/30/4	User-defined	New VLANs can be added to the VLAN table. For details, see Table 29.7.
Ethernet agent [ETHA]		
EATDQDCq.DQD	User-defined	The number of descriptors in descriptor queues can be set
EAVCC.VEM	011	Allows SC-tagged frames and C-tagged frames to go through transparently
EAVTC.CTV	1	Set the default VLAN value to 1
EATTFC.UT	0	Allows any TAG formats to go through even the unsupported ones
RMAC [RMAC]		
MPIC.PIS	User-defined	Set the PHY interface type
MPIC.LSC	User-defined	Set the PHY speed
MRAFC	03C703C7H	Set MAC reception in promiscuous mode
GWCA [GWCA]		
GWRDQDCq.DQD	User-defined	The number of descriptors in descriptor queues can be set
GWVCC.VEM	011	Allows SC-tagged frames and C-tagged frames to go through transparently
GWVTC.CTV	1	Set the default VLAN value to 1
GWTTFC.UT	0	Allows any TAG formats to go through even the unsupported ones
GWDCBAC0/1	User-defined	Set the LINKFIX table base address
GWDCCi	For i = 1: ● Set DQT to 1 ● Set OSID to user-defined. Else: ● Set OSID to user-defined.	Queue 0 is used for data reception, queue 1 is used for data transmission and queue 2 is used for learning monitoring. All other queues are not used. Note: For learning monitoring, use extended descriptors.

Table 29.9 VLAN learning in VLAN table

Field name VLAN	Setting value	Explanation
SL	0	No secure entry used
HLD	0	Hardware learning for MAC addresses not disabled for any VLAN
SLV	Set to 1 for ports included in VLAN	Only authorize to receive corresponding VLAN from ports included in this VLAN
DV	Set to 1 for ports included in VLAN	Only authorize to send corresponding VLAN to ports included in this VLAN
CSD	0	Frames received by CPUs on sub-destination 0
CME	0	No mirroring
EME	0	No mirroring
IPU	0	No frame priority update
IPV	0	No frame priority update

29.4.1.5 AVB scheduling

29.4.1.5.1 Forwarding and Queuing Enhancements for Time-Sensitive Streams [802.1Qav]

29.4.2 Precautions

- ESWM can support max (([Unused pointer number] × 128) / 4) bytes frame (supported maximum frame size). A buffer overflow occurs when the transmit frame size exceeds the supported maximum size. [Unused pointer number] = 512 − (Activated port number × 6).
  - Set to [RMAC] MRFSCE.EMXS smaller than maximum frame size for Ethernet ports
  - Do not transmit larger than maximum frame size for CPU ports.
- GWCA can only transmit/receive frames with maximum size of 63488 bytes for the first frame. For the upcoming frames, the maximum frame size can be 65535 bytes provided LINK/LINKFIX descriptors were not used in between the queue.
- When using Cut through, use it so as not to cause a buffer overflow. The following usage is recommended:
  - Only use from-one-to-one communication.
    (Example of prohibition : Multicast forwarding, to Ethernet mirroring, Loopback forwarding).
  - Prohibition of the function to stop transmission. Use flow control on reception port.
    (Example of prohibition : TX queue pause, CBS, TAS).
- The overall specifications of ESWM are determined on the frequency of the operating clock.
  See the following table for details to select the operating clock.

Selection			Definition	
Local RAM size*1	Clk*1	Internal bandwidth	Ethernet & CPU total bandwidth	Throughput (Ether2Ether)
64 KB	250 MHz	32 Gbps	MAX 1G bps (Exp 100M Ether 2 port + 100M CPU 1 port) × TX/RX	800 to 2000 ns
64 KB	50 MHz	6.4 Gbps	MAX 1G bps (Exp 100M Ether 2 port + 100M CPU 1 port) × TX/RX	2000 to 6000 ns
64 KB	250 MHz	32 Gbps	MAX 3G bps (Exp 1G Ether 2 port + 1G CPU 1 port) × TX/RX	400 to 600 ns
64 KB	150 MHz	19.2 Gbps	MAX 3G bps (Exp 1G Ether 2 port + 1G CPU 1 port) × TX/RX	500 to 800 ns

Note: At the frame end (forwarding source port) - frame start (forwarding destination port). When measuring, there is no dependency on the payload length of the frame.
Note: The frame start (forwarding destination port) is up to the point where the preamble starts to output.
Note 1. Specifications less than these are prohibited.
Throughput measurement conditions:

- From frame end (forwarding source port) to frame start (forwarding destination port)
- The forwarding (destination port) is not busy
- L2 forwarding unicast (1:1 forwarding).

29.5 Operation

29.5.1 Internal Clocks

Table 29.10 lists ESWM internal clocks. These clocks are supplied by the Clock Generation Circuit (CGC). Set the clock before starting the ESWM operation.

Table 29.10 Configurable range of clocks

VDD Voltage [V]	Speed [Mbps]	ICLK [MHz]	PCLKA [MHz]	ESWCLK [MHz]	ESWPHYCLK [MHz]
VSCR_1, voltage range 1	1000	150 to 250	up to 125	150 to 250	500
	100/10	50 to 250	up to 125	50 to 250*1	—
VSCR_2, voltage range 2	1000	100 to 150	up to 75	150 to 225	500
	100/10	50 to 150	up to 75	50 to 225*1	—

Note 1. When using products in the HLQFP176 package, set ESWCLK to 125 MHz or less.

Clock restrictions are as follows:

- ICLK × 1.5 ≥ ESWCLK
- ESWCLK > PCLKA
- ICLK > PCLKA.

29.5.2 PHY Interface

29.5.2.1 PHY Interface Select

Table 29.11 lists the configuration settings for the ESWM PHY interface. Set the MIICRn.MIISEL[1:0] bits first before setting the MPIC register in RMACn.

Table 29.11 Configuration settings for ESWM PHY interface

Type	Speed	RMACn.MPIC		MIICRn
		LSC[2:0]	PIS[2:0]	MIISEL[1:0]
MII	10 Mbps	000b	000b (MII)	00b
MII	100 Mbps	001b	000b (MII)	00b
GMII	1 Gbps	010b	010b (GMII)	00b
RGMII	10 Mbps	000b	000b (MII)	01b
RGMII	100 Mbps	001b	000b (MII)	01b
RGMII	1 Gbps	010b	010b (GMII)	01b
RMII	10 Mbps	000b	000b (MII)	10b
RMII	100 Mbps	001b	000b (MII)	10b

29.5.2.2 MII/GMII Interface

When MIICRn.MIISEL[1:0] = 00b, MII/GMII interface is selected. GTX_CLK is generated by dividing ESWPHYCLK by 4.

29.5.2.3 RGMII Interface

ESWM can be converted from an external RGMII to an internal GMII/MII. The RGMII interface is enabled when MIICRn.MIISEL[1:0] = 01b. Since the initial state of the ESWM PHY interface is MII/GMII, it becomes operational after the reset is released with the MIIRR.RGRST bit. At this time, TXC is generated by dividing ESWPHYCLK. MIICRn. Set MIISEL[1:0] to 01b before a release reset.

29.5.2.3.1 RGMII Internal Delay Mode

The RGMII interface supports internal delay mode for RGMIIn_TXC.

29.5.2.4 RMII Interface

ESWM can be converted from an external RMII to an internal MII. The RMII interface is enabled when MIICRn.MIISEL[1:0] = 10b. Since the initial state of the ESWM PHY interface is MII/GMII, it becomes operational after the reset is released with the MIIRR.RMRST bit. Set MIICRn.MIISEL[1:0] to 10b before a release reset.

29.5.3 ESWM Core Function

See the following chapters:

- section 30, Ethernet Message Forwarding Engine (MFWD)
- section 30.5.4. Fabric (MFAB)
- section 31, Ethernet Common Agent (COMA)
- section 32, Ethernet Agent (ETHA)
- section 33, Ethernet MAC (RMAC)
- section 34, Ethernet CPU Agent (GWCA).

29.5.4 gPTP and PTP Pulse Generator

See the following chapters:

- section 35, Ethernet Generic PTP Timer (GPTP)
- section 35.6. Pulse Output Timer Module (plsotim).

29.5.5 TAS Gate Status Monitor

The signal to be output to the ET_TAS_STAn pin can be selected using the TASSTMS register. Table 29.12 lists the TASSTMS register select table.

Table 29.12

MSSn[4:0]	Module	Signal
00000b to 01000b	ETHA0	eha_race_tas_gate_state[0] to eha_race_tas_gate_state[8]
10000b to 11000b	ETHA1	eha_race_tas_gate_state[0] to eha_race_tas_gate_state[8]

For details of eha_race_tas_gate_state[8:0], see section 32, Ethernet Agent (ETHA).

29.6 Interrupt and Event

Table 29.13 Interrupt and event output (1 of 2)

Symbol	Source flag	Description	Type
ETHER_FWEI	MFWD.FWEIS	Forwarding Error Interrupt	Level
ETHER_CAEI	COMA.CAEIS	Common Error Interrupt	Level
ETHER_GWEI	GWCA0.GWEIS0 (other than GWEIS0.TDFES), GWCA0.GWEIS1, GWCA0.GWEIS4, GWCA0.GWEIS5	GWCA0 Error Interrupt	Level
ETHER_EAEI0	ETHA0.EAEIS0, ETHA0.EAEIS1, ETHA0.EAEIS2, RMAC0.MEIS	ETHA0 Error Interrupt	Level
ETHER_EAEI1	ETHA1.EAEIS0, ETHA1.EAEIS1, ETHA1.EAEIS2, RMAC1.MEIS	ETHA1 Error Interrupt	Level

Table 29.13 Interrupt and event output (2 of 2)

Symbol	Source flag	Description	Type
ETHER_PTPSI0	GPTP.PTPIS0	gPTP Status Interrupt 0	Level
ETHER_PTPSI1	GPTP.PTPIS1	gPTP Status Interrupt 1	Level
ETHER_FWSI	MFWD.FWMIS	Forwarding Status Interrupt	Level
ETHER_CAMI	COMA.CAMIS	Common Status Interrupt	Level
ETHER_EASI0	RMAC0.MMIS	ETHA0 Status Interrupt	Level
ETHER_EASI1	RMAC1.MMIS	ETHA1 Status Interrupt	Level
ETHER_GWDI7	GWCA0.GWDIDS, GWCA0.GWEIS2, GWCA0.GWEIS3 with mapped by TOP.TPEMIMC7t	GWCA Data Interrupt	Level
ETHER_GWDI6			
ETHER_GWDI5			
ETHER_GWDI4			
ETHER_GWDI3			
ETHER_GWDI2			
ETHER_GWDI1			
ETHER_GWDI0			
ETHER_TSDI1	GWCA0.GWTSDIS, GWCA0.GWEIS0.TDFES with mapped by TOP.TPMIMC6t	Time Stamp Data Interrupt	Level
ETHER_TSDI0		Time Stamp Data Interrupt	Level
ETHER_MDIO0	RMAC0.MMIS1	MDIO Interrupt 0	Level
ETHER_MDIO1	RMAC1.MMIS1	MDIO Interrupt 1	Level
ETHER_RMPI0	RMAC0.MMIS.LPIAIS, RMAC0.MMIS.LPIDIS, RMAC0.MMIS.MPDIS	RMAC0 PHY Interrupt	Level
ETHER_RMPI1	RMAC1.MMIS.LPIAIS, RMAC1.MMIS.LPIDIS, RMAC1.MMIS.MPDIS	RMAC1 PHY Interrupt	Level
GPTP_MATCH0	—	Media Clock Recovery Match 0	Pulse
GPTP_MATCH1	—	Media Clock Recovery Match 1	Pulse
GPTP_PTPOUT0	—	PTP Pulse Out 0	Pulse
GPTP_PTPOUT1	—	PTP Pulse Out 1	Pulse
GPTP_PTPOUT2	—	PTP Pulse Out 2	Pulse
GPTP_PTPOUT3	—	PTP Pulse Out 3	Pulse

Table 29.14 Event input

Symbol	Event source	Description
ELC_GPTP0	ELSR31	gPTP Media clock capture event 0
ELC_GPTP1	ELSR32	gPTP Media clock capture event 1

The gPTP media clock capture can select event source using the PTP Capture Event Select register.

29.7 Usage Notes

29.7.1 Settings for the Module-Stop Function

The MSTPC30 bit in Module Stop Control Register C (MSTPCRC) can enable or disable the ESWM module operation. The modules are initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Mode.

29.7.2 Transition to Software Standby Mode

ESWM should run Switch disable flow to disable it before transitioning to software standby.

If RGMII interface or RMII interface is selected, these interfaces need to be reset after Switch disable flow by the MIIRR register.

29.7.3 Return from Software Standby Mode

After returning from Software Standby mode, ESWM should run the switch initialization flow to initialize it.

29.7.4 Transition to ESWM Power Domain off by Power Gating

ESWM should run Switch disable flow to disable it before transitioning to the ESWM power domain off.

If RGMII interface or RMII interface is selected, these interfaces need to be reset after switch disable flow by the MIIRR register.

29.7.5 Transition to ESWM Power Domain on by Power Gating

After the ESWM power domain is on, ESWM should run the switch initialization flow to initialize it.

30. Ethernet Message Forwarding Engine (MFWD)

30.1 Overview

Ethernet Message Forwarding Engine (MFWD) is part of ESWM and aims to filter, forward, and route the switch frames. MFWD snoops the frame information from fabric to redistribute them to the agent after filtering, forwarding, and routing mechanisms.

Table 30.1 lists the MFWD specifications and Figure 30.1 shows a block diagram.

Table 30.1 MFWD specifications

Function		Details
Overall function	Cut-through	Forwarding frames while receiving them depending on Layer 2
	Direct forwarding	Forwarding frames coming from GWCAs by bypassing the forwarding engine [GWCA]
	Layer 3 forwarding	Forwarding frames based on a stream ID
	Layer 2 forwarding	Forwarding frames based on destination MAC addresses and VLAN
	Port based forwarding	Forwarding frames depending on source port
	L23U	Update of Layer 2 and Layer 3 depending on Layer 3 forwarding
	PSFP	PSFP filtering [PSFP] depending on Layer 3 forwarding
	ATS	ATS traffic shaping [ATS]
	FRER	FRER [FRER] depending on Layer 3 forwarding
Data provision	Ethernet Frames	Corresponds to the IEEE 802.3, 802.1Q and 802.1CB standards [802.3] [802.1Q] [802.1CB].
	Descriptors	Local Ethernet Descriptor, Local Direct Descriptor (Agents to MFWD) Forwarding descriptor (MFWD to Agents)

Figure 30.1 shows MFWD block diagram.

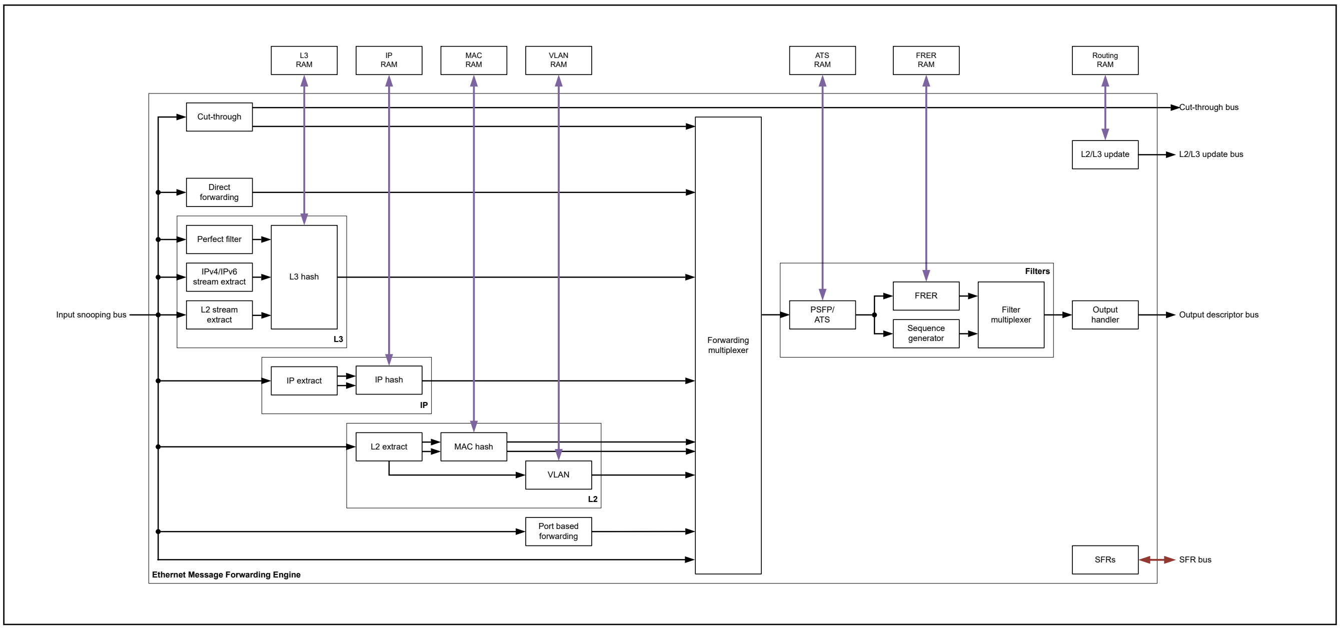


Figure 30.1 MFWD block diagram

Table 30.2 MFWD functional blocks (1 of 2)

Block name	Function
Cut-through	Filter frame L2 and send them to while receiving them.

Table 30.2 MFWD functional blocks (2 of 2)

Block name	Function
Direct forwarding	Forward frames coming from a CPU with the forwarding information already present.
L3	L2 to L4 Stream based forwarding/routing engine
L2	L2 switch
Port based forwarding	Source port-based forwarding
Forwarding multiplexer	Arbitrate between all the forwarding/routing mechanisms
PSFP/ATS	Per stream filtering and policing [802.1Qci] and asynchronous traffic shaping [802.1Qcr]
FRER	Frame Replication and Elimination for Reliability [802.1CB]
Sequence generator	Sequence number generation for FRER proxy functionality [802.1CB]
Filter multiplexer	Arbitrate between forwarding results and filtering results.
Output handler	Copy and send descriptors to targets. Manage overflow and security error recovery.

30.2 MFWD Register List

Table 30.3 MFWD registers (1 of 7)

Offset address	Register name	Register symbol
0x0000	General Configuration Register	FWGC
0x0010	TAG TPID Configuration Register 0	FWTTC0
0x0014	TAG TPID Configuration Register 1	FWTTC1
0x0020	CPU Exceptional Path Target Configuration Register	FWCEPTC
0x0024	CPU Exceptional Path Reason Configuration Register 0	FWCEPRC0
0x0028	CPU Exceptional Path Reason Configuration Register 1	FWCEPRC1
0x002C	CPU Exceptional Path Reason Configuration Register 2	FWCEPRC2
0x0030	CPU Learning Path Target Configuration Register	FWCLPTC
0x0034	CPU Learning Path Reason Configuration Register	FWCLPRC
0x0040	CPU Mirroring Path Target Configuration Register	FWCMPTC
0x0044	Ethernet Mirroring Path Target Configuration Register	FWEMPTC
0x0050	Source-Destination Mirroring Path Target Configuration Register	FWSDMPTC
0x0054	Source-Destination Mirroring Path Vector Configuration Register	FWSDMPVC
0x0080 + 0x04 × i	Port i Level Based Watermark Configuration Register (i = 0 to 2)	FWLBWMCi
0x0100 + 0x10 × i	Port i Configuration Register 0 (i = 0 to 2)	FWPCi0
0x0104 + 0x10 × i	Port i Configuration Register 1 (i = 0 to 2)	FWPCi1
0x0108 + 0x10 × i	Port i Configuration Register 2 (i = 0 to 2)	FWPCi2
0x0400 + 0x40 × i	Cut-Through General Configuration Register i0 (i = 0 to 7)	FWCTGCi0
0x0404 + 0x40 × i	Cut-Through General Configuration Register i1 (i = 0 to 7)	FWCTGCi1
0x0408 + 0x40 × i	Cut-Through Target Configuration Register i0 (i = 0 to 7)	FWCTTCi0
0x040C + 0x40 × i	Cut-Through Target Configuration Register i1 (i = 0 to 7)	FWCTTCi1
0x0410 + 0x40 × i	Cut-Through Target Configuration Register i20 (i = 0 to 7)	FWCTTCi20
0x0420 + 0x40 × i	Cut-Through Separation Configuration Register i0 (i = 0 to 7)	FWCTSCi0
0x0424 + 0x40 × i	Cut-Through Separation Configuration Register i1 (i = 0 to 7)	FWCTSCi1
0x0428 + 0x40 × i	Cut-Through Separation Configuration Register i2 (i = 0 to 7)	FWCTSCi2
0x042C + 0x40 × i	Cut-Through Separation Configuration Register i3 (i = 0 to 7)	FWCTSCi3
0x0430 + 0x40 × i	Cut-Through Separation Configuration Register i4 (i = 0 to 7)	FWCTSCi4

Table 30.3 MFWD registers (2 of 7)

Offset address	Register name	Register symbol
0x1000 + 0x10 × i	Two-Byte Filter Configuration Register i (i = 0 to 15)	FWTWBFCi
0x1004 + 0x10 × i	Two-Byte Filter Value Configuration Register i (i = 0 to 15)	FWTWBFVCi
0x1400 + 0x10 × i	Three-Byte Filter Configuration Register i (i = 0 to 15)	FWTHBFCi
0x1404 + 0x10 × i	Three-Byte Filter Value 0 Configuration Register i (i = 0 to 15)	FWTHBFV0Ci
0x1408 + 0x10 × i	Three-Byte Filter Value 1 Configuration Register i (i = 0 to 15)	FWTHBFV1Ci
0x1800 + 0x10 × i	Four-Byte Filter Configuration Register i (i = 0 to 15)	FWFOBFCi
0x1804 + 0x10 × i	Four-Byte Filter Value 0 Configuration Register i (i = 0 to 15)	FWFOBFV0Ci
0x1808 + 0x10 × i	Four-Byte Filter Value 1 Configuration Register i (i = 0 to 15)	FWFOBFV1Ci
0x1C00 + 0x10 × i	Range Filter Configuration Register i (i = 0, 1)	FWRFCi
0x1C04 + 0x10 × i	Range Filter Value Configuration Register i (i = 0, 1)	FWRFVCi
0x2000 + 0x40 × i	Cascade Filter Configuration Register i (i = 0 to 15)	FWCFCi
0x2004 + 0x40 × i + 0x04 × j	Cascade Filter Mapping Configuration Register ij (i = 0 to 15) (j = 0 to 6)	FWCFMCij
0x4008	IPv4 Stream Configuration Register	FWIP4SC
0x4018	IPv6 Stream Configuration Register	FWIP6SC
0x401C	IPv6 Offset Configuration Register	FWIP6OC
0x4020	Layer 2 Stream Configuration Register	FWL2SC
0x4030	Stream Filter Hash Equation Configuration Register	FWSFHEC
0x4040	Software Hash Calculation Request Register 0	FWSHCR0
0x4044	Software Hash Calculation Request Register 1	FWSHCR1
0x4048	Software Hash Calculation Request Register 2	FWSHCR2
0x404C	Software Hash Calculation Request Register 3	FWSHCR3
0x4050	Software Hash Calculation Request Register 4	FWSHCR4
0x4054	Software Hash Calculation Request Register 5	FWSHCR5
0x4058	Software Hash Calculation Request Register 6	FWSHCR6
0x405C	Software Hash Calculation Request Register 7	FWSHCR7
0x4060	Software Hash Calculation Request Register 8	FWSHCR8
0x4064	Software Hash Calculation Request Register 9	FWSHCR9
0x4068	Software Hash Calculation Request Register 10	FWSHCR10
0x406C	Software Hash Calculation Request Register 11	FWSHCR11
0x4070	Software Hash Calculation Request Register 12	FWSHCR12
0x4074	Software Hash Calculation Request Register 13	FWSHCR13
0x4078	Software Hash Calculation Request Result Register	FWSHCRR
0x4090	L3 Hash Entry Configuration Register	FWLTHHEC
0x4094	L3 Hash Configuration Register	FWLTHHC
0x40A0	L3 Table Learn Register 0	FWLTHTL0
0x40A4	L3 Table Learn Register 1	FWLTHTL1
0x40A8	L3 Table Learn Register 2	FWLTHTL2
0x40AC	L3 Table Learn Register 3	FWLTHTL3
0x40B0	L3 Table Learn Register 4	FWLTHTL4
0x40B4	L3 Table Learn Register 5	FWLTHTL5
0x40B8	L3 Table Learn Register 6	FWLTHTL6
0x40BC	L3 Table Learn Register 7	FWLTHTL7

Table 30.3 MFWD registers (3 of 7)

Offset address	Register name	Register symbol
0x40C0	L3 Table Learn Register 80	FWLTHTL80
0x40D0	L3 Table Learn Register 9	FWLTHTL9
0x40D4	L3 Table Learn Result Register	FWLTHTLR
0x40E0	L3 Table Initialization Monitoring Register	FWLTHTIM
0x40E4	L3 Table Entry Monitoring Register	FWLTHTEM
0x4100	L3 Table Search Register 0	FWLTHTS0
0x4104	L3 Table Search Register 1	FWLTHTS1
0x4108	L3 Table Search Register 2	FWLTHTS2
0x410C	L3 Table Search Register 3	FWLTHTS3
0x4110	L3 Table Search Register 4	FWLTHTS4
0x4120	L3 Table Search Result Register 0	FWLTHTSR0
0x4124	L3 Table Search Result Register 1	FWLTHTSR1
0x4128	L3 Table Search Result Register 2	FWLTHTSR2
0x412C	L3 Table Search Result Register 3	FWLTHTSR3
0x4130	L3 Table Search Result Register 40	FWLTHTSR40
0x4140	L3 Table Search Result Register 5	FWLTHTSR5
0x4150	L3 Table Read Register	FWLTHTR
0x4154	L3 Table Read Result Register 0	FWLTHTRR0
0x4158	L3 Table Read Result Register 1	FWLTHTRR1
0x415C	L3 Table Read Result Register 2	FWLTHTRR2
0x4160	L3 Table Read Result Register 3	FWLTHTRR3
0x4164	L3 Table Read Result Register 4	FWLTHTRR4
0x4168	L3 Table Read Result Register 5	FWLTHTRR5
0x416C	L3 Table Read Result Register 6	FWLTHTRR6
0x4170	L3 Table Read Result Register 7	FWLTHTRR7
0x4174	L3 Table Read Result Register 8	FWLTHTRR8
0x4180	L3 Table Read Result Register 90	FWLTHTRR90
0x4190	L3 Table Read Result Register 10	FWLTHTRR10
0x4620	MAC Hash Entry Configuration Register	FWMACHEC
0x4624	MAC Hash Configuration Register	FWMACHC
0x4630	MAC Table Learn Register 0	FWMACTL0
0x4634	MAC Table Learn Register 1	FWMACTL1
0x4638	MAC Table Learn Register 2	FWMACTL2
0x463C	MAC Table Learn Register 3	FWMACTL3
0x4640	MAC Table Learn Register 40	FWMACTL40
0x4650	MAC Table Learn Register 5	FWMACTL5
0x4654	MAC Table Learn Result Register	FWMACTLR
0x4660	MAC Table Initialization Monitoring Register	FWMACTIM
0x4664	MAC Table Entry Monitoring Register	FWMACTEM
0x4670	MAC Table Search Register 0	FWMACTS0
0x4674	MAC Table Search Register 1	FWMACTS1
0x4678	MAC Table Search Result Register 0	FWMACTSR0

Table 30.3 MFWD registers (4 of 7)

Offset address	Register name	Register symbol
0x467C	MAC Table Search Result Register 1	FWMACTSR1
0x4680	MAC Table Search Result Register 20	FWMACTSR20
0x4690	MAC Table Search Result Register 3	FWMACTSR3
0x46A0	MAC Table Read Register	FWMACTR
0x46A4	MAC Table Read Result Register 0	FWMACTRR0
0x46A8	MAC Table Read Result Register 1	FWMACTRR1
0x46AC	MAC Table Read Result Register 2	FWMACTRR2
0x46B0	MAC Table Read Result Register 3	FWMACTRR3
0x46B4	MAC Table Read Result Register 4	FWMACTRR4
0x46C0	MAC Table Read Result Register 50	FWMACTRR50
0x46D0	MAC Table Read Result Register 6	FWMACTRR6
0x4880	MAC Aging US Prescaler Configuration Register	FWMACAGUSPC
0x4884	MAC Aging Configuration Register	FWMACAGC
0x4888	MAC Aging Monitoring Register 0	FWMACAGM0
0x488C	MAC Aging Monitoring Register 1	FWMACAGM1
0x4900	VLAN Table Entry Configuration Register	FWVLANTEC
0x4910	VLAN Table Learn Register 0	FWVLANTL0
0x4914	VLAN Table Learn Register 1	FWVLANTL1
0x4918	VLAN Table Learn Register 2	FWVLANTL2
0x4920	VLAN Table Learn Register 30	FWVLANTL30
0x4930	VLAN Table Learn Register 4	FWVLANTL4
0x4934	VLAN Table Learn Result Register	FWVLANTLR
0x4940	VLAN Table Initialization Monitoring Register	FWVLANTIM
0x4944	VLAN Table Entry Monitoring Register	FWVLANTEM
0x4950	VLAN Table Search Register	FWVLANTS
0x4954	VLAN Table Search Result Register 0	FWVLANTSR0
0x4958	VLAN Table Search Result Register 1	FWVLANTSR1
0x4960	VLAN Table Search Result Register 20	FWVLANTSR20
0x4970	VLAN Table Search Result Register 3	FWVLANTSR3
0x4A00 + 0x10 × i	Port i Port Based Forwarding Configuration Register (i = 0 to 2)	FWPBFCi
0x4A04 + 0x10 × i	Port i Port Based Forwarding CSD Configuration Register 0 (i = 0 to 2)	FWPBFCSDCi0
0x4E00	Layer 2/Layer 3 Update Rule Learn Register 0	FWL23URL0
0x4E04	Layer 2/Layer 3 Update Rule Learn Register 1	FWL23URL1
0x4E08	Layer 2/Layer 3 Update Rule Learn Register 2	FWL23URL2
0x4E0C	Layer 2/Layer 3 Update Rule Learn Register 3	FWL23URL3
0x4E10	Layer 2/Layer 3 Update Rule Learn Result Register	FWL23URLR
0x4E20	Layer 2/Layer 3 Update Table Initialization Monitoring Register	FWL23UTIM
0x4E30	Layer 2/Layer 3 Update Rule Read Register	FWL23URR
0x4E34	Layer 2/Layer 3 Update Rule Read Result Register 0	FWL23URRR0
0x4E38	Layer 2/Layer 3 Update Rule Read Result Register 1	FWL23URRR1
0x4E3C	Layer 2/Layer 3 Update Rule Read Result Register 2	FWL23URRR2
0x4E40	Layer 2/Layer 3 Update Rule Read Result Register 3	FWL23URRR3

Table 30.3 MFWD registers (5 of 7)

Offset address	Register name	Register symbol
0x4F00 + 0x04 × i	Layer 2/Layer 3 Update Remapping Configuration Register i (i = 0 to 31)	FWL23URMCi
0x5000 + 0x04 × i	PSFP MSDU Filter Global Configuration Register i (i = 0 to 15)	FWPMFGCi
0x5600 + 0x20 × i	PSFP Meter i Filter Configuration Register (i = 0 to 31)	FWPMTRFCi
0x5604 + 0x20 × i	PSFP Meter i CBS Configuration Register (i = 0 to 31)	FWPMTRCBSCi
0x5608 + 0x20 × i	PSFP Meter i CIR Configuration Register (i = 0 to 31)	FWPMTRCIRCi
0x560C + 0x20 × i	PSFP Meter i EBS Configuration Register (i = 0 to 7)	FWPMTREBSCi
0x5610 + 0x20 × i	PSFP Meter i EIR Configuration Register (i = 0 to 7)	FWPMTREIRCi
0x5614 + 0x20 × i	PSFP Meter i Filter Monitoring Register (i = 0 to 31)	FWPMTRFMi
0x6000	FRER Table Learn Register 0	FWFTL0
0x6004	FRER Table Learn Register 1	FWFTL1
0x6008	FRER Table Learn Result Register	FWFTLR
0x6010	FRER Timeout Configuration Register	FWFTOC
0x6014	FRER Timeout Prescaler Configuration Register 0	FWFTOPC
0x6020	FRER Table Initialization Monitoring Register	FWFTIM
0x6030	FRER Table Read Register	FWFTR
0x6034	FRER Table Read Result Register 0	FWFTRR0
0x6038	FRER Table Read Result Register 1	FWFTRR1
0x603C	FRER Table Read Result Register 2	FWFTRR2
0x6100 + 0x08 × i	Sequence Number Generation i Configuration Register (i = 0 to 31)	FWSEQNGCi
0x6104 + 0x08 × i	Sequence Number Generation i Monitoring Register (i = 0 to 31)	FWSEQNGMi
0x6200	Sequence Number Reset Configuration Register	FWSEQNRC
0x6300 + 0x20 × i E: 0x6E00 + 0x20 × i	Port i Cut-Through Forwarded Descriptor Counter Register (i = 0, 1)	FWCTFDCNi
0x6340 E: 0x6E40	Port 2 Direct Descriptor Forwarded Descriptor Counter Register	FWDDFDCN2
0x6304 + 0x20 × i E: 0x6E04 + 0x20 × i	Port i Layer 3 Forwarded Descriptor Counter Register (i = 0 to 2)	FWLTHFDCNi
0x630C + 0x20 × i E: 0x6E0C + 0x20 × i	Port i Layer 2 Forwarded Descriptor Counter Register (i = 0 to 2)	FWLTWFDCNi
0x6310 + 0x20 × i E: 0x6E10 + 0x20 × i	Port i Port Based Forwarded Descriptor Counter Register (i = 0 to 2)	FWPBFDCNi
0x6314 + 0x20 × i E: 0x6E14 + 0x20 × i	Port i MAC Hardware Learn Counter Register (i = 0 to 2)	FWMHLCNi
0x6504 + 0x20 × i E: 0x7004 + 0x20 × i	Port i Watermark Rejected Descriptor Counter Register (i = 0 to 2)	FWWMRDCNi
0x6508 + 0x20 × i E: 0x7008 + 0x20 × i	Port i Cut-Through Rejected Descriptor Counter Register (i = 0, 1)	FWCTRDCNi
0x6548 E: 0x7048	Port 2 Direct Descriptor Rejected Descriptor Counter Register	FWDDRDCN2
0x650C + 0x20 × i E: 0x700C + 0x20 × i	Port i Layer 3 Rejected Descriptor Counter Register (i = 0 to 2)	FWLTHRDCNi
0x6514 + 0x20 × i E: 0x7014 + 0x20 × i	Port i Layer 2 Rejected Descriptor Counter Register (i = 0 to 2)	FWLTWRDCNi
0x6518 + 0x20 × i E: 0x7018 + 0x20 × i	Port i Port Based Rejected Descriptor Counter Register (i = 0 to 2)	FWPBRDCNi
0x6700 + 0x04 × i E: 0x7200 + 0x04 × i	PSFP MSDU i Filtered Descriptor Counter Register (i = 0 to 15)	FWPMFDCNi

Table 30.3 MFWD registers (6 of 7)

Offset address	Register name	Register symbol
0x6800 + 0x10 × i E: 0x7300 + 0x10 × i	PSFP Meter i Green Descriptor Counter Register (i = 0 to 31)	FWPMGDCNi
0x6804 + 0x10 × i E: 0x7304 + 0x10 × i	PSFP Meter i Yellow Descriptor Counter Register (i = 0 to 7)	FWPMYDCNi
0x6808 + 0x10 × i E: 0x7308 + 0x10 × i	PSFP Meter i Red Descriptor Counter Register (i = 0 to 31)	FWPMRDCNi
0x6A00 + 0x08 × i E: 0x7500 + 0x08 × i	FRER i Passed Packet Counter Register (i = 0 to 127)	FWFRPPCNi
0x6A04 + 0x08 × i E: 0x7504 + 0x08 × i	FRER i Discarded Packet Counter Register (i = 0 to 127)	FWFRDPCNi
0x7900 + 0x10 × i	Port i Error Interrupt Status Register 0 (i = 0 to 2)	FWEIS0i
0x7904 + 0x10 × i	Port i Error Interrupt Enable Register 0 (i = 0 to 2)	FWEIE0i
0x7908 + 0x10 × i	Port i Error Interrupt Disable Register 0 (i = 0 to 2)	FWEID0i
0x7A00	Error Interrupt Status Register 1	FWEIS1
0x7A04	Error Interrupt Enable Register 1	FWEIE1
0x7A08	Error Interrupt Disable Register 1	FWEID1
0x7A10	Error Interrupt Status Register 2	FWEIS2
0x7A14	Error Interrupt Enable Register 2	FWEIE2
0x7A18	Error Interrupt Disable Register 2	FWEID2
0x7A40	Error Interrupt Status Register 5	FWEIS5
0x7A44	Error Interrupt Enable Register 5	FWEIE5
0x7A48	Error Interrupt Disable Register 5	FWEID5
0x7A50	Error Interrupt Status Register 60	FWEIS60
0x7A54	Error Interrupt Enable Register 60	FWEIE60
0x7A58	Error Interrupt Disable Register 60	FWEID60
0x7A60	Error Interrupt Status Register 61	FWEIS61
0x7A64	Error Interrupt Enable Register 61	FWEIE61
0x7A68	Error Interrupt Disable Register 61	FWEID61
0x7A70	Error Interrupt Status Register 62	FWEIS62
0x7A74	Error Interrupt Enable Register 62	FWEIE62
0x7A78	Error Interrupt Disable Register 62	FWEID62
0x7A80	Error Interrupt Status Register 63	FWEIS63
0x7A84	Error Interrupt Enable Register 63	FWEIE63
0x7A88	Error Interrupt Disable Register 63	FWEID63
0x7A90	Error Interrupt Status Register 70	FWEIS70
0x7A94	Error Interrupt Enable Register 70	FWEIE70
0x7A98	Error Interrupt Disable Register 70	FWEID70
0x7AA0	Error Interrupt Status Register 71	FWEIS71
0x7AA4	Error Interrupt Enable Register 71	FWEIE71
0x7AA8	Error Interrupt Disable Register 71	FWEID71
0x7AB0	Error Interrupt Status Register 72	FWEIS72
0x7AB4	Error Interrupt Enable Register 72	FWEIE72
0x7AB8	Error Interrupt Disable Register 72	FWEID72
0x7AC0	Error Interrupt Status Register 73	FWEIS73

Table 30.3 MFWD registers (7 of 7)

Offset address	Register name	Register symbol
0x7AC4	Error Interrupt Enable Register 73	FWEIE73
0x7AC8	Error Interrupt Disable Register 73	FWEID73
0x7AD0	Error Interrupt Status Register 80	FWEIS80
0x7AD4	Error Interrupt Enable Register 80	FWEIE80
0x7AD8	Error Interrupt Disable Register 80	FWEID80
0x7AE0	Error Interrupt Status Register 81	FWEIS81
0x7AE4	Error Interrupt Enable Register 81	FWEIE81
0x7AE8	Error Interrupt Disable Register 81	FWEID81
0x7AF0	Error Interrupt Status Register 82	FWEIS82
0x7AF4	Error Interrupt Enable Register 82	FWEIE82
0x7AF8	Error Interrupt Disable Register 82	FWEID82
0x7B00	Error Interrupt Status Register 83	FWEIS83
0x7B04	Error Interrupt Enable Register 83	FWEIE83
0x7B08	Error Interrupt Disable Register 83	FWEID83
0x7C00	Monitoring Interrupt Status Register 0	FWMIS0
0x7C04	Monitoring Interrupt Enable Register 0	FWMIE0
0x7C08	Monitoring Interrupt Disable Register 0	FWMID0

Note: Base address: MFWD = 0x403C_0000, MFWD_NS = 0x503C_0000
Note: The address preceded by “E:” correspond to an emulation address, which allows to read a register without modifying its content.

Note: If nothing specified, a register need an authorization from secured APB interface to be accessed by unsecured APB interface. This applies for Read and Write accesses. Any exception to this rule will be specified under “Security restrictions” or “Security no restrictions” labels.

30.3 MFWD Register Descriptions

30.3.1 MFWD Function Registers

This section describes MFWD function registers.

For MFWD function procedure details, see section 30.4.1. Software Flows.

30.3.1.1 General Function Registers

30.3.1.1.1 FWGC : General Configuration Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x0000

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

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

Bit	Symbol	Function	R/W
1:0	SVM[1:0]	Switch VLAN Mode 0 0: No VLAN mode (VLAN not used for forwarding) 0 1: C-TAG mode (C-TAG used for forwarding) 1 0: SC-TAG mode (S-TAG used for forwarding) 1 1: Reserved	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

SVM[1:0] bits (Switch VLAN Mode)

HW: If FWGC.SVM is set to 00b, incoming packet PCP and DEI fields will be considered as 0 by MFWD.

For more details, see section 30.5.1.2. ESWM Modes.

30.3.1.1.2 FWTTC0 : TAG TPID Configuration Register 0

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x0010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	STT[15:0]															
Value after reset:	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0

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

Bit	Symbol	Function	R/W
15:0	CTT[15:0]	C-TAG TPID [801.2Q] Configure the C-TAG TPID.	R/W
31:16	STT[15:0]	S-TAG TPID [801.2Q] Configure the S-TAG TPID.	R/W

Note:
- SW: FWTTC0.CTT, FWTTC0.STT and FWTTC1.RTT registers should always be set to different values.
- SW: This register cannot be set to 0x0000 to 0x05FF, 0x0800 nor 0x86DD.

30.3.1.1.3 FWTTC1 : TAG TPID Configuration Register 1

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x0014

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

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

Bit	Symbol	Function	R/W
15:0	RTT[15:0]	R-TAG TPID [801.2CB] Configure the R-TAG TPID.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note:
- SW: FWTTC0.CTT, FWTTC0.STT and FWTTC1.RTT registers should always be set to different values.
- SW: This register cannot be set to 0x0000 to 0x05FF, 0x0800 nor 0x86DD.

30.3.1.1.4 FWCEPTC : CPU Exceptional Path Target Configuration Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x0020

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

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

Bit	Symbol	Function	R/W
5:0	EPCSD[5:0]	Exceptional Path CPU Sub Destination Configures the CPU sub destination to which exceptional frames will be sent.	R/W
11:6	—	These bits are read as 0. The write value should be 0.	R/W
14:12	EPIPV[2:0]	Exceptional Path Internal Priority Value Configures the priority to which exceptional frames will be sent.	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
16	EPCS	Exceptional Path CPU Select*1 Configures CPU to which exceptional frames will be sent. See Fabric specification for port mapping [MFAB].	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W
24	EPSL	Exceptional Path Security Level Configures descriptor security level for exceptional path.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: These bits should not be set to a value greater than 1.

30.3.1.1.5 FWCEPRC0 : CPU Exceptional Path Reason Configuration Register 0

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x0024

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	ETFEF	EOEEF	EUEEF	EBOEEF	ERPOOEF	ERFFEF	ECFFCEEF	ECFSEEF	EFFMEEF	EFCSEEF	ENIBEEF	EPCRCEEF	EPHYEEF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EPHYEEF	Ethernet PHY Error Exceptional Forwarding 0: Frames received with PHY Error are discarded. 1: Frames received with PHY Error are forwarded to exceptional path.	R/W

Bit	Symbol	Function	R/W
1	EPCRCEEF	Ethernet PCH CRC Error Exceptional Forwarding 0: Frames received with PCH CRC Error are discarded. 1: Frames received with PCH CRC Error are forwarded to exceptional path.	R/W
2	ENIBEEF	Ethernet Nibble Error Exceptional Forwarding 0: Frames received with Nibble Error are discarded. 1: Frames received with Nibble Error are forwarded to exceptional path.	R/W
3	EFCSEEF	Ethernet FCS Error Exceptional Forwarding 0: Frames received with FCS Error are discarded. 1: Frames received with FCS Error are forwarded to exceptional path.	R/W
4	EFFMEEF	Ethernet Final Fragment Missing Error Exceptional Forwarding 0: Frames received with Final Fragment Missing Error are discarded. 1: Frames received with Final Fragment Missing Error are forwarded to exceptional path.	R/W
5	ECFSEEF	Ethernet C-Fragment SMD Error Exceptional Forwarding 0: Frames received with C-Fragment SMD Error are discarded. 1: Frames received with C-Fragment SMD Error are forwarded to exceptional path.	R/W
6	ECFFCEEF	Ethernet C-Fragment FRAG_COUNT Error Exceptional Forwarding 0: Frames received with C-Fragment FRAG_COUNT Error are discarded. 1: Frames received with C-Fragment FRAG_COUNT Error are forwarded to exceptional path.	R/W
7	ERFFEF	Ethernet RMAC Frame Filtered Exceptional Forwarding 0: Frames received with RMAC Frame Filtered flag are discarded. 1: Frames received with RMAC Frame Filtered flag are forwarded to exceptional path.	R/W
8	ERPOOEF	Ethernet Reception Partially Out of Operation Exceptional Forwarding 0: Frames received with Reception Partially Out of Operation flag are discarded. 1: Frames received with Reception Partially Out of Operation flag are forwarded to exceptional path.	R/W
9	EBOEEF	Ethernet Buffer Overflow Error Exceptional Forwarding 0: Frames received with Buffer Overflow Error are discarded. 1: Frames received with Buffer Overflow Error are forwarded to exceptional path.	R/W
10	EUEEF	Ethernet Undersize Error Exceptional Forwarding 0: Frames received with Undersize Error are discarded. 1: Frames received with Undersize Error are forwarded to exceptional path.	R/W
11	EOEEF	Ethernet Oversize Error Exceptional Forwarding 0: Frames received with Oversize Error are discarded. 1: Frames received with Oversize Error are forwarded to exceptional path.	R/W
12	ETFEF	Ethernet TAG Filtering Exceptional Forwarding 0: Frames received with TAG Filtering flag are discarded. 1: Frames received with TAG Filtering flag are forwarded to exceptional path.	R/W
16:13	—	These bits are read as 0. The write value should be 0.	R/W
17	GAXEEF	GWCA AXI Error Exceptional Forwarding 0: Frames received with AXI Error are discarded. 1: Frames received with AXI Error are forwarded to exceptional path.	R/W
18	GSEQEEF	GWCA Sequence Error Exceptional Forwarding 0: Frames received with Sequence Error are discarded. 1: Frames received with Sequence Error are forwarded to exceptional path.	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
20	GTFEF	GWCA TAG Filtering Exceptional Forwarding 0: Frames received with TAG Filtering flag are discarded. 1: Frames received with TAG Filtering flag are forwarded to exceptional path.	R/W
21	GDNEEF	GWCA Descriptor Number Error Exceptional Forwarding 0: Frames received with Descriptor Number Error are discarded. 1: Frames received with Descriptor Number Error are forwarded to exceptional path.	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
24	DDEEF	Direct Descriptor Error Exceptional Forwarding 0: Frames with Direct Descriptor Error are discarded. 1: Frames with Direct Descriptor Error are forwarded to exceptional path.	R/W
25	—	This bit is read as 0. The write value should be 0.	R/W
26	DDFSFEF	Direct Descriptor Format Security Filtering Exceptional Forwarding 0: Frames with Direct Descriptor Format Security Error are discarded. 1: Frames with Direct Descriptor Format Security Error are forwarded to exceptional path.	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note:
- For ethernet error/flag explanations, refer to Ethernet agent specification document [ETHA].
- For GWCA error/flag explanations, refer to GWCA specification document [GWCA].

30.3.1.1.6 FWCEPRC1 : CPU Exceptional Path Reason Configuration Register 1

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x0028

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

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

Bit	Symbol	Function	R/W
0	FMSDUFEF	MSDU Filtering Exceptional Forwarding 0: Frames filtered by PSFP MSDU filter are discarded. 1: Frames filtered by PSFP MSDU filter are forwarded to exceptional path.	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	FMTRFEF	Meter Filtering Exceptional Forwarding 0: Frames filtered by PSFP Meter filter are discarded. 1: Frames filtered by PSFP Meter filter are forwarded to exceptional path.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	FIFFEF	Individual FRER Filtering Exceptional Forwarding 0: Frames filtered by FRER individual recovery are discarded. 1: Frames filtered by FRER individual recovery are forwarded to exceptional path.	R/W
9	FSFFEF	Sequence FRER Filtering Exceptional Forwarding 0: Frames filtered by FRER Sequence recovery are discarded. 1: Frames filtered by FRER Sequence recovery are forwarded to exceptional path.	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.1.7 FWCEPRC2 : CPU Exceptional Path Reason Configuration Register 2

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x002C

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	FPBNTFEF	FLTWNTFEF	—	FLTHNTFEF	FDDNTFEF	—	—	FVLANUFEF	FSMACUFEF	FDMACUFEF	—	—	FLTHUFEF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FLTHUFEF	Layer 3 Unknown Filtering Exceptional Forwarding 0: Frames filtered because their stream ID is unknown are discarded. 1: Frames filtered because their stream ID is unknown are forwarded to exceptional path.	R/W
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	FDMACUFEF	Destination MAC Unknown Filtering Exceptional Forwarding 0: Frames filtered because their Destination MAC is unknown are discarded. 1: Frames filtered because their Destination MAC is unknown are forwarded to exceptional path.	R/W
4	FSMACUFEF	Source MAC Unknown Filtering Exceptional Forwarding 0: Frames filtered because their Source MAC is unknown are discarded. 1: Frames filtered because their Source MAC is unknown are forwarded to exceptional path.	R/W
5	FVLANUFEF	VLAN Unknown Filtering Exceptional Forwarding 0: Frames filtered because their VLAN is unknown are discarded. 1: Frames filtered because their VLAN is unknown are forwarded to exceptional path.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	FDDNTFEF	Direct Descriptor No Target Filtering Exceptional Forwarding 0: Frames filtered because no target during direct forwarding is detected are discarded. 1: Frames filtered because no target during direct forwarding is detected are forwarded to exceptional path.	R/W
9	FLTHNTFEF	Layer 3 No Target Filtering Exceptional Forwarding 0: Frames filtered because no target during Layer 3 forwarding is detected are discarded. 1: Frames filtered because no target during Layer 3 forwarding is detected are forwarded to exceptional path.	R/W
10	—	This bit is read as 0. The write value should be 0.	R/W
11	FLTWNTFEF	Layer 2 No Target Filtering Exceptional Forwarding 0: Frames filtered because the combination on VLAN destination vector and destination MAC destination vector is null are discarded. 1: Frames filtered because the combination on VLAN destination vector and destination MAC destination vector is null are forwarded to exceptional path.	R/W
12	FPBNTFEF	Port Based No Target Filtering Exceptional Forwarding 0: Frames filtered because no target during Port Based forwarding is detected are discarded. 1: Frames filtered because no target during Port Based forwarding is detected are forwarded to exceptional path.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
16	FLTHSLFEF	Layer 3 Source Lock Filtering Exceptional Forwarding 0: Frames filtered because of Layer 3 source lock are discarded. 1: Frames filtered because of Layer 3 source lock are forwarded to exceptional path.	R/W
18:17	—	These bits are read as 0. The write value should be 0.	R/W
19	FDMACSLFEF	Destination MAC Source Lock Filtering Exceptional Forwarding 0: Frames filtered because of Destination MAC source lock are discarded. 1: Frames filtered because of Destination MAC source lock are forwarded to exceptional path.	R/W
20	FSMACSLFEF	Source MAC Source Lock Filtering Exceptional Forwarding 0: Frames filtered because of Source MAC source lock are discarded. 1: Frames filtered because of Source MAC source lock are forwarded to exceptional path.	R/W
21	FVLANSLFEF	VLAN Source Lock Filtering Exceptional Forwarding 0: Frames filtered because of VLAN source lock are discarded. 1: Frames filtered because of VLAN source lock are forwarded to exceptional path.	R/W
25:22	—	These bits are read as 0. The write value should be 0.	R/W
26	FWMFEF	Watermark Filtering Exceptional Forwarding 0: Frames filtered because of watermark are discarded. 1: Frames filtered because of watermark are forwarded to exceptional path.	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.1.8 FWCLPTC : CPU Learning Path Target Configuration Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x0030

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

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

Bit	Symbol	Function	R/W
5:0	LPCSD[5:0]	Learning Path CPU Sub Destination Configures the CPU sub destination to which Learning frames will be sent.	R/W
11:6	—	These bits are read as 0. The write value should be 0.	R/W
14:12	LPIPV[2:0]	Learning Path Internal Priority Value Configures the priority to which Learning frames will be sent.	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
16	LPCS	Learning Path CPU Select*1 Configures CPU to which Learning frames will be sent. See Fabric specification for port mapping [MFAB].	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W
24	LPSL	Learning Path Security Level Configures descriptor security level for Learning path.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: These bits should not be set to a value greater than 1.

30.3.1.1.9 FWCLPRC : CPU Learning Path Reason Configuration Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x0034

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

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

Bit	Symbol	Function	R/W
0	USIDLF	Unknown Stream ID Learning Forwarding 0: Frames received with an unknown stream ID are not forwarded to Learning path. 1: Frames received with an unknown stream ID are forwarded to Learning path.	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	UDMACLF	Unknown Destination MAC Learning Forwarding 0: Frames received with an unknown destination MAC address are not forwarded to Learning path. 1: Frames received with an unknown destination MAC address are forwarded to Learning path.	R/W
5	USMACLF	Unknown Source MAC Learning Forwarding 0: Frames received with an unknown Source MAC are not forwarded to Learning path. 1: Frames received with an unknown Source MAC are forwarded to Learning path.	R/W
6	UPSMACLF	Unknown Port for Source MAC Learning Forwarding 0: Frames received with an unknown port for Source MAC are not forwarded to Learning path. 1: Frames received with an unknown port for Source MAC are forwarded to Learning path.	R/W
7	UVLANLF	Unknown VLAN Learning Forwarding 0: Frames received with an unknown VLAN are not forwarded to Learning path. 1: Frames received with an unknown VLAN are forwarded to Learning path.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.1.10 FWCMPTC : CPU Mirroring Path Target Configuration Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x0040

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

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

Bit	Symbol	Function	R/W
5:0	CMPCSD[5:0]	CPU Mirroring Path CPU Sub Destination Configures the CPU sub destination to which CPU mirroring frames will be sent.	R/W
11:6	—	These bits are read as 0. The write value should be 0.	R/W
14:12	CMPIPV[2:0]	CPU Mirroring Path Internal Priority Value	R/W
15	CMPIPU	CPU Mirroring Path Internal Priority Update 0: Frames mirrored to CPU path are sent with the same priority as for forwarding. 1: Frames mirrored to CPU path are sent with FWCMPTC.CMPIPV priority.	R/W
16	CMPCS	CPU Mirroring Path CPU Select*1 Configures CPU to which CPU mirroring frames will be sent. See Fabric specification for port mapping [MFAB].	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W
24	CMPSL	CPU Mirroring Path Security Level Configures descriptor security level for CPU mirroring path.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: These bits should not be set to a value greater than 1.

30.3.1.1.11 FWEMPTC : Ethernet Mirroring Path Target Configuration Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x0044

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

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

Bit	Symbol	Function	R/W
11:0	—	These bits are read as 0. The write value should be 0.	R/W
14:12	EMPIPV[2:0]	Ethernet Mirroring Path Internal Priority Value	R/W
15	EMPIPU	Ethernet Mirroring Path Internal Priority Update 0: Frames mirrored to Ethernet path are sent with the same priority as for forwarding. 1: Frames mirrored to Ethernet path are sent with FWEMPTC.EMPIPV priority.	R/W
16	EMPPS	Ethernet Mirroring Path Port Select*1 Configures port to which Ethernet mirroring frames will be sent. See Fabric specification for port mapping [MFAB].	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W
24	EMPSL	Ethernet Mirroring Path Security Level Configures descriptor security level for Ethernet mirroring path.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: These bits should not be set to a value greater than 1.

30.3.1.1.12 FWSDMPTC : Source-Destination Mirroring Path Target Configuration Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x0050

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

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

Bit	Symbol	Function	R/W
5:0	SDMPCSD[5:0]	Source-Destination Mirroring Path CPU Sub Destination Configures the CPU sub destination to which Source-Destination mirroring frames will be sent.	R/W
11:6	—	These bits are read as 0. The write value should be 0.	R/W
14:12	SDMPIPV[2:0]	Source-Destination Mirroring Path Internal Priority Value	R/W
15	SDMPIPU	Source-Destination Mirroring Path Internal Priority Update 0: Frames mirrored to Source-Destination path are sent with the same priority as for forwarding. 1: Frames mirrored to Source-Destination path are sent with FWSDMPTC.SDMPIPV priority.	R/W
17:16	SDMPPS[1:0]	Source-Destination Mirroring Path Port Select*1 Configures Port to which Source-Destination mirroring frames will be sent. See Fabric specification for port mapping [MFAB].	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
24	SDMPSL	Source-Destination Mirroring Path Security Level Configures descriptor security level for Source-Destination mirroring path.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: These bits should not be set to a value greater than 3.

30.3.1.1.13 FWSDMPVC : Source-Destination Mirroring Path Vector Configuration Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x0054

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

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

Bit	Symbol	Function	R/W
2:0	SDMDV[2:0]	Source-Destination Mirroring Destination Vector	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W
18:16	SDMSV[2:0]	Source-Destination Mirroring Source Vector	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

SDMDV[2:0] bits (Source-Destination Mirroring Destination Vector)

Frames coming from a port set in FWSDMPVC.SDMSV register and sent to a port set in FWSDMPVC.SDMDV register is also duplicated to Source-Destination mirroring path.

Setting this register to 0 disable Source-Destination mirroring path.

SDMSV[2:0] bits (Source-Destination Mirroring Source Vector)

Frames coming from a port set in FWSDMPVC.SDMSV register and sent to a port set in FWSDMPVC.SDMDV register is also duplicated to Source-Destination mirroring path.

Setting this register to 0 disable Source-Destination mirroring path.

30.3.1.1.14 FWLBWMCi : Port i Level Based Watermark Configuration Register (i = 0 to 2)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x0080 + 0x4 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	WMFLPR15 to WMFLPR0															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WMCLPR15 to WMCLPR0															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	WMCLPR0 to WMCLPR15	Watermark Critical Level Priority n Reject (n = 0 to 15)*1 0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to j are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to j are discarded when watermark critical level is set for port i (WM.CRITICAL[i] is set [COMA]).	R/W
31:16	WMFLPR0 to WMFLPR15	Watermark Flush Level Priority n Reject (n = 0 to 15)*1 0: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to j are forwarded. 1: Frames coming for port i and that should be forwarded with {DEI,IPV} equal to j are discarded when watermark flush level is set for port i (WM.FLUSH[i] is set [COMA]).	R/W

Note 1. {DEI,IPV} refers to the bit association {DEI,IPV[2:0]}, so in decimal it can be written 8*DEI+IPV.

30.3.1.1.15 FWPCi0 : Port i Configuration Register 0 (i = 0 to 2)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x0100 + 0x10 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	VLAN RUS	VLAN RU	VLAN SA	MACH MA	MACH LA	MACR USSA	MACR USA	MACS SA	MACR UDSA	MACR UDA	MACD SA	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	L2SE	IP6OE	IP6TE	IP6UE	IP4OE	IP4TE	IP4UE	LTHR USS	LTHR US	LTHTA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LTHTA	L3 Table Active 0: L3 table is disabled for source port i. 1: L3 table is enabled for source port i.	R/W
1	LTHRUS	L3 Reject Unknown Streams 0: Unknown streams coming from source port i are not rejected. 1: Unknown streams coming from source port i are rejected.	R/W
2	LTHRUSS	L3 Reject Unknown Secure Streams 0: Unknown secure streams coming from source port i are not rejected. 1: Unknown secure streams coming from source port i are rejected.	R/W
3	IP4UE	IPv4 UDP Enable 0: IPv4/UDP detection disabled for source port i. 1: IPv4/UDP detection enabled for source port i.	R/W
4	IP4TE	IPv4 TCP Enable 0: IPv4/TCP detection disabled for source port i. 1: IPv4/TCP detection enabled for source port i.	R/W
5	IP4OE	IPv4 Other Enable 0: Non-TCP non-UDP IPv4 detection disabled for source port i. 1: Non-TCP non-UDP IPv4 detection enabled for source port i.	R/W
6	IP6UE	IPv6 UDP Enable 0: IPv6/UDP detection disabled for source port i. 1: IPv6/UDP detection enabled for source port i.	R/W
7	IP6TE	IPv6 TCP Enable 0: IPv6/TCP detection disabled for source port i. 1: IPv6/TCP detection enabled for source port i.	R/W
8	IP6OE	IPv6 Other Enable 0: Non-TCP non-UDP IPv6 detection disabled for source port i. 1: Non-TCP non-UDP IPv6 detection enabled for source port i.	R/W
9	L2SE	L2 Stream Enable 0: L2 stream disabled for source port i. 1: L2 stream enabled for source port i.	R/W
19:10	—	These bits are read as 0. The write value should be 0.	R/W
20	MACDSA	MAC Destination Search Active 0: MAC Destination search is disabled for source port i. 1: MAC Destination search is enabled for source port i.	R/W
21	MACRUDA	MAC Reject Unknown Destination Addresses 0: Unknown MAC Destination addresses coming from source port i are not rejected. 1: Unknown MAC Destination addresses coming from source port i are rejected.	R/W

Bit	Symbol	Function	R/W
22	MACRUDSA	MAC Reject Unknown Destination Secure Addresses 0: Unknown secure MAC Destination addresses coming from source port i are not rejected. 1: Unknown secure MAC Destination addresses coming from source port i are rejected.	R/W
23	MACSSA	MAC Source Search Active 0: MAC Source search is disabled for source port i. 1: MAC Source search is enabled for source port i.	R/W
24	MACRUSA	MAC Reject Unknown Source Addresses 0: Unknown MAC Source addresses coming from source port i are not rejected. 1: Unknown MAC Source addresses coming from source port i are rejected.	R/W
25	MACRUSSA	MAC Reject Unknown Source Secure Addresses 0: Unknown secure MAC Source addresses coming from source port i are not rejected. 1: Unknown secure MAC Source addresses coming from source port i are rejected.	R/W
26	MACHLA	MAC Hardware Learning Active 0: Unknown unicast MAC Source addresses coming from source port i are not added to the MAC table. 1: Unknown unicast MAC Source addresses coming from source port i are added to the MAC table.	R/W
27	MACHMA	MAC Hardware Migration Active 0: Known unicast MAC Source addresses coming from source port i with a wrong source port in MAC.DV are not overwritten. 1: Known unicast MAC Source addresses coming from source port i with a wrong source port in MAC.DV are overwritten.	R/W
28	VLANSA	VLAN Search Active*1 HW: If VLAN Destination search is disabled for port i, all frames coming from port i are considered as unknown for VLAN search. 0: VLAN search is disabled for source port i. 1: VLAN search is enabled for source port i.	R/W
29	VLANRU	VLAN Reject Unknown*1 0: Unknown VLAN coming from source port i are not rejected. 1: Unknown VLAN coming from source port i are rejected.	R/W
30	VLANRUS	VLAN Reject Unknown Secure*1 0: Unknown secure VLAN coming from source port i are not rejected. 1: Unknown secure VLAN coming from source port i are rejected.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. SW: This bit should be set to 1 if FWGC.SVM is set to 00b.

30.3.1.1.16 FWPCi1 : Port i Configuration Register 1 (i = 0 to 2)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x0104 + 0x10 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	LTHFM2 to LTHFM0		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	DDE	Direct Descriptor Enable*1 0: Direct Descriptor disabled for port i 1: Direct Descriptor enabled for port i	R/W
1	DDSL	Direct Descriptor Security Level*1 0: Secured Direct Descriptor discarded for port i. 1: Secured Direct Descriptor forwarded for port i.	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
18:16	LTHFM2 to LTHFM0	Layer 3 Forwarding Mask n (n = 0 to 2) 0: Frames forwarded by Layer 3 forwarding can be forwarded to port j. 1: Frames forwarded by Layer 3 forwarding cannot be forwarded to port j.	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. HW: This bit is only valid for i = 2, 3.

30.3.1.1.17 FWPCi2 : Port i Configuration Register 2 (i = 0 to 2)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x0108 + 0x10 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	LTWFM2 to LTWFM0		
Value after reset:	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

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

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
18:16	LTWFM2 to LTWFM0	Layer 2 Forwarding Mask n (n = 0 to 2) 0: Frames forwarded by Layer 2 forwarding can be forwarded to port j. 1: Frames forwarded by Layer 2 forwarding cannot be forwarded to port j.	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.2 Cut-through Forwarding Function Registers

30.3.1.2.1 FWCTGCi0 : Cut-Through General Configuration Register i0 (i = 0 to 7)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x0400 + 0x40 × i

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	CTRTGI	CTVCTRL[1:0]		CTFI	—	—	CTETE	CTSDE	CTSPE	CTSVE	CTCDE	CTCPE	CTCVE	CTMSE	CTMDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CTMDE	Cut-Through MAC Destination Enable 0: MAC destination not included in Cut-Through separation for separation rule i. 1: MAC destination included in Cut-Through separation for separation rule i.	R/W
1	CTMSE	Cut-Through MAC Source Enable 0: MAC Source not included in Cut-Through separation for separation rule i. 1: MAC Source included in Cut-Through separation for separation rule i.	R/W
2	CTCVE	Cut-Through C-TAG VLAN Enable*1 0: C-TAG VLAN not included in Cut-Through separation for separation rule i. 1: C-TAG VLAN included in Cut-Through separation for separation rule i.	R/W
3	CTCPE	Cut-Through C-TAG PCP Enable*1 0: C-TAG PCP not included in Cut-Through separation for separation rule i. 1: C-TAG PCP included in Cut-Through separation for separation rule i.	R/W
4	CTCDE	Cut-Through C-TAG DEI Enable*1 0: C-TAG DEI not included in Cut-Through separation for separation rule i. 1: C-TAG DEI included in Cut-Through separation for separation rule i.	R/W
5	CTSVE	Cut-Through S-TAG VLAN Enable*2 0: S-TAG VLAN not included in Cut-Through separation for separation rule i. 1: S-TAG VLAN included in Cut-Through separation for separation rule i.	R/W
6	CTSPE	Cut-Through S-TAG PCP Enable*2 0: S-TAG PCP not included in Cut-Through separation for separation rule i. 1: S-TAG PCP included in Cut-Through separation for separation rule i.	R/W
7	CTSDE	Cut-Through S-TAG DEI Enable*2 0: S-TAG DEI not included in Cut-Through separation for separation rule i. 1: S-TAG DEI included in Cut-Through separation for separation rule i.	R/W
8	CTETE	Cut-Through Ethernet Type Enable 0: Ethernet Type not included in Cut-Through separation for separation rule i. 1: Ethernet Type included in Cut-Through separation for separation rule i.	R/W
10:9	—	These bits are read as 0. The write value should be 0.	R/W
11	CTFI	Cut-Through FCS In*3 This bit is used as FDESCR.FI for cut-through. 0: The frame matching separation rule i as no FCS included. 1: The frame matching separation rule i as an FCS included.	R/W
13:12	CTVCTRL[1:0]	Cut-Through VLAN Control [GWCA] [ETHA] These bits are used as FDESCR.VCTRL[1:0] for cut-through. 0 0: The ingress matching separation rule i frame is a No TAG frame. 0 1: The ingress matching separation rule i frame is a C-TAG frame. 1 0: The ingress matching separation rule i frame is an SC-TAG frame. 1 1: The ingress matching separation rule i frame is a CoS TAG frame.	R/W
14	CTRTGI	Cut-Through R-TAG In [GWCA] [ETHA] This bit is used as FDESCR.RTGI for cut-through. 0: R-TAG is not included in cut-through frame. 1: R-TAG is included in cut-through frame.	R/W
31:15	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: This bit can be set to 1 only if FWGC.SVM is set to 10b or 01b.
Note 2. SW: This bit can be set to 1 only if FWGC.SVM is set to 10b.
Note 3. SW: This bit should be set to the frame source port MRGC.RCPT (same) value [RMAC].

- Case 1: CTFI = 1, RCPT = 1 and correct FCS frame incoming
  → correct FCS frame forwarded
  (The FCS attached to the input frame is output as it is. If the tag is converted, the FCS value will be invalid.)
- Case 2: CTFI = 1, RCPT = 1 and incorrect FCS frame incoming
  → incorrect FCS frame forwarded
- Case 3: CTFI = 0, RCPT = 0 and correct FCS frame incoming
  → correct FCS frame forwarded
- Case 4: CTFI = 0, RCPT = 0 and incorrect FCS frame incoming
  → correct FCS frame forwarded (The correct FCS will be added after the incorrect FCS.)

30.3.1.2.2 FWCTGCi1 : Cut-Through General Configuration Register i1 (i = 0 to 7)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x0404 + 0x40 × i

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CTMT[25:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
25:0	CTMT[25:0]	Cut-Through Maximum time	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

CTMT[25:0] bits (Cut-Through Maximum time)

Describes maximum time on the PHY in ns of frames matching separation rule I.

This time should take in account the whole frame (including preamble, SFD, TAGs and FCS) and IFG.

This size will be used for TAS calculations [ETHA].

Note:
- If this register does not have the correct value, some frames could cross a closed gate in TAS [ETHA].
- if TAS is enabled, sending frames smaller than this register value will decrease the PHY throughput because the frame time will be reserved at this register value [ETHA]. Therefore, TAS gate open time must be bigger than CTMT value.

30.3.1.2.3 FWCTTCi0 : Cut-Through Target Configuration Register i0 (i = 0 to 7)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x0408 + 0x40 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CTDFM1 to CTDFM0	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	CTDV[2:0]	Cut-through Destination Vector*1 Select the ports to which the frames matching separation rule i will be sent. Enables cut-through for separation rule i.	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W
17:16	CTDFM1 to CTDFM0	Cut-Through Destination n Forwarding Mode (n = 0 to 1)*2 0: Frame is forwarded in Cut-through mode. 1: Frame is forwarded in Store and forward mode.	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. When cut-through is enabled for rule i, at least one bit set to 1 in FWCTTCi0.CTDV[2:0] and should be set FWCTTCi0.CTDFM to 0 for a cut-through able destination agent (when cut-through is enabled, at least one Ethernet destination port should use with cut-through mode).

Note 2. HW: In Cut-through mode, a frame with an error cannot be rejected and is forwarded to its destination ports.

30.3.1.2.4 FWCTTCi1 : Cut-Through Target Configuration Register i1 (i = 0 to 7)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x040C + 0x40 × i

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

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

Bit	Symbol	Function	R/W
11:0	—	These bits are read as 0. The write value should be 0.	R/W
14:12	CTIPV[2:0]	Cut-through Internal Priority Value	R/W
15	CTIPU	Cut-through Internal Priority Update 0: Use the output descriptor priority to forwarding frames matching separation rule i. 1: Use FWCTTCi1.CTIPV priority to forwarding frames matching separation rule i.	R/W
16	CTCME	Cut-through CPU Mirroring Enable 0: CPU mirroring is disabled for frames matching separation rule i. 1: CPU mirroring is enabled for frames matching separation rule i.	R/W
17	CTEME	Cut-through Ethernet Mirroring Enable 0: Ethernet mirroring is disabled for frames matching separation rule i. 1: Ethernet mirroring is enabled for frames matching separation rule i.	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.2.5 FWCTTCi20 : Cut-Through Target Configuration Register i20 (i = 0 to 7)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x0410 + 0x40 × i

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

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

Bit	Symbol	Function	R/W
5:0	CTCSD[5:0]	Cut-Through CPU Sub Destination Configures the CPU sub destination to which frames matching separation rule i will go for slow port number 0.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.2.6 FWCTSCi0 : Cut-Through Separation Configuration Register i0 (i = 0 to 7)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x0420 + 0x40 × i

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

Bit	Symbol	Function	R/W
31:0	CTDMAU[31:0]	Cut-Through Destination MAC Address Upper Part	R/W

30.3.1.2.7 FWCTSCi1 : Cut-Through Separation Configuration Register i1 (i = 0 to 7)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x0424 + 0x40 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CTDMAL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
15:0	CTSMAU[15:0]	Cut-Through Source MAC Address Upper Part	R/W
31:16	CTDMAL[15:0]	Cut-Through Destination MAC Address Lower Part	R/W

30.3.1.2.8 FWCTSCi2 : Cut-Through Separation Configuration Register i2 (i = 0 to 7)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x0428 + 0x40 × i

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

Bit	Symbol	Function	R/W
31:0	CTSMAL[31:0]	Cut-Through Source MAC Address Lower Part	R/W

30.3.1.2.9 FWCTSCi3 : Cut-Through Separation Configuration Register i3 (i = 0 to 7)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x042C + 0x40 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CTSD	CTSP[2:0]			CTSV[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
11:0	CTCV[11:0]	Cut-Through C-TAG VLAN	R/W
14:12	CTCP[2:0]	Cut-Through C-TAG PCP	R/W
15	CTCD	Cut-Through C-TAG DEI	R/W
27:16	CTSV[11:0]	Cut-Through S-TAG VLAN	R/W
30:28	CTSP[2:0]	Cut-Through S-TAG PCP	R/W
31	CTSD	Cut-Through S-TAG DEI	R/W

30.3.1.2.10 FWCTSCi4 : Cut-Through Separation Configuration Register i4 (i = 0 to 7)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x0430 + 0x40 × i

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

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

Bit	Symbol	Function	R/W
15:0	CTET[15:0]	Cut-Through Ethernet Type	R/W
16	CTSPN	Cut-Through Source Port Number*1	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. HW: In Cut-through mode (not Store and Forward modes), this bit should be set to a value smaller than 2.

30.3.1.3 Layer 3 Forwarding/Routing Perfect Filter Function Registers

30.3.1.3.1 FWTWBFCi : Two-Byte Filter Configuration Register i (i = 0 to 15)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x1000 + 0x10 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	TWBFOV[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
1:0	TWBFUM[1:0]	Two-Byte Filter Unit Mode*1 0 0: Mask mode 0 1: Expand mode 1 0: Precise mode 1 1: Reserved	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	TWBFM	Two-Byte Filtering Mode*2 0: Offset filtering 1: VLAN TAG filtering	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
23:16	TWBFOV[7:0]	Two-Byte Filter Offset Value*3	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: These bits should be set to 01b if FWTWBFCi.TWBFM is set to 1.
Note 2. SW: This bit should be set to 01b if FWGC.SVM is set to 00b.
Note 3. ● SW: If FWTWBFCi.TWBFM is set to 1, these bits should be set to 0 to filter S-TAG and set to 2 to filter C-TAG.
● SW: If FWTWBFCi.TWBFM is set to 1 and FWGC.SVM is set to 01b, these bits should not be set to 0.

TWBFOV[7:0] bits (Two-Byte Filter Offset Value)

- If FWTWBFCi.TWBFM is 0: Offset in byte where the filter i will start filtering the frame data.
- If FWTWBFCi.TWBFM is 1: Offset in byte where the filter i will start filtering the frame TAGs.

30.3.1.3.2 FWTWBFVCi : Two-Byte Filter Value Configuration Register i (i = 0 to 15)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x1004 + 0x10 × i

Bit position:	31–16	15–0
Bit field:	TWBFV1[15:0]	TWBFV0[15:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	TWBFV0[15:0]	Two-Byte Filter Value 0	R/W
31:16	TWBFV1[15:0]	Two-Byte Filter Value 1	R/W

TWBFV0[15:0] bits (Two-Byte Filter Value 0)

- If FWTWBFCi.TWBFUM is 00b: Used as a 2-byte filter value masked by FWTWBFVCi.TWBFV1 with a filter ID of 2*i.
- If FWTWBFCi.TWBFUM is 01b: Used as the first 2-byte of a 4-byte filter extended by FWTWBFVCi.TWBFV1 with a filter ID of 2*i (Compare value is {FWTWBFVCi.TWBFV0, FWTWBFVCi.TWBFV1}).
- If FWTWBFCi.TWBFUM is 10b: Used as 2-byte filter value with a filter ID of 2*i.

TWBFV1[15:0] bits (Two-Byte Filter Value 1)

- If FWTWBFCi.TWBFUM is 00b: Used as a mask for FWTWBFVCi.TWBFV0.
- If FWTWBFCi.TWBFUM is 01b: Used as an extension for FWTWBFVCi.TWBFV0 (Compare value is { FWTWBFVCi.TWBFV0, FWTWBFVCi.TWBFV1}).
- If FWTWBFCi.TWBFUM is 10b: Used as 2-byte filter value with a filter ID of 2*i +1.

30.3.1.3.3 FWTHBFCi : Three-Byte Filter Configuration Register i (i = 0 to 15)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x1400 + 0x10 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	THBFOV[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
1:0	THBFUM[1:0]	Three-Byte Filter Unit Mode 0 0: Mask mode 0 1: Expand mode 1 0: Precise mode 1 1: Reserved	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
23:16	THBFOV[7:0]	Three-Byte Filter Offset Value Offset in byte where the filter i will start filtering the frame. (The offset ignores all TAGs)	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.3.4 FWTHBFV0Ci : Three-Byte Filter Value 0 Configuration Register i (i = 0 to 15)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x1404 + 0x10 × i

Bit position:	31	30	29	28	27	26	25	24	23																							0
Bit field:	—	—	—	—	—	—	—	—	THBFV0[23:0]																							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
23:0	THBFV0[23:0]	Three-Byte Filter Value 0	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

THBFV0[23:0] bits (Three-Byte Filter Value 0)

- If FWTHBFCi.THBFUM[1:0] is 00b: Used as a 3-byte filter value masked by FWTHBFV1Ci.THBFV1[23:0] with a filter ID of 2 × (i + 16).
- If FWTHBFCi.THBFUM[1:0] is 01b: Used as the first 3-byte of a 6-byte filter extended by FWTHBFV1Ci.THBFV1[23:0] with a filter ID of 2 × (i + 16) (Compare value is {FWTHBFV0Ci.THBFV0[23:0], FWTHBFV1Ci.THBFV1[23:0]}).
- If FWTHBFCi.THBFUM[1:0] is 10b: Used as 3-byte filter value with a filter ID of 2 × (i + 16).

30.3.1.3.5 FWTHBFV1Ci : Three-Byte Filter Value 1 Configuration Register i (i = 0 to 15)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x1408 + 0x10 × i

Bit position:	31										23 ... 0
Bit field:	—	—	—	—	—	—	—	—	—	—	THBFV1[23:0]

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
23:0	THBFV1[23:0]	Three-Byte Filter Value 1	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

THBFV1[23:0] bits (Three-Byte Filter Value 1)

- If FWTHBFCi.THBFUM[1:0] is 00b: Used as a mask for FWTHBFV0Ci.THBFV0[23:0].
- If FWTHBFCi.THBFUM[1:0] is 01b: Used as an extension for FWTHBFV0Ci.THBFV0[23:0] (Compare value is { FWTHBFV0Ci.THBFV0[23:0], FWTHBFV1Ci.THBFV1[23:0]}).
- If FWTHBFCi.THBFUM[1:0] is 10b: Used as 3-byte filter value with a filter ID of 2 × (i + 16) + 1.

30.3.1.3.6 FWFOBFCi : Four-Byte Filter Configuration Register i (i = 0 to 15)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x1800 + 0x10 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	FOBFOV[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
1:0	FOBFUM[1:0]	Four-Byte Filter Unit Mode 0 0: Mask mode 0 1: Expand mode 1 0: Precise mode 1 1: Reserved	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
23:16	FOBFOV[7:0]	Four-Byte Filter Offset Value Offset in byte where the filter i will start filtering the frame. (The offset ignores all TAGs)	R/W

Bit	Symbol	Function	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.3.7 FWFOBFV0Ci : Four-Byte Filter Value 0 Configuration Register i (i = 0 to 15)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x1804 + 0x10 × i

Bit position:	31																															0
Bit field:	FOBFV0[31:0]																															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	FOBFV0[31:0]	Four-Byte Filter Value 0	R/W

FOBFV0[31:0] bits (Four-Byte Filter Value 0)

- If FWFOBFCi. FOBFUM[1:0] is 00b: Used as a 4-byte filter value masked by FWFOBFV1Ci. FOBFV1[31:0] with a filter ID of 2 × (i + 36).
- If FWFOBFCi. FOBFUM[1:0] is 01b: Used as the first 4-byte of an 8-byte filter extended by FWFOBFV1Ci. FOBFV1[31:0] with a filter ID of 2 × (i + 36) (Compare value is {FWFOBFV0Ci.FOBFV0[31:0], FWFOBFV1Ci. FOBFV1[31:0]}).
- If FWFOBFCi. FOBFUM[1:0] is 10b: Used as 4-byte filter value with a filter ID of 2 × (i + 36).

30.3.1.3.8 FWFOBFV1Ci : Four-Byte Filter Value 1 Configuration Register i (i = 0 to 15)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x1808 + 0x10 × i

Bit position:	31																															0
Bit field:	FOBFV1[31:0]																															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	FOBFV1[31:0]	Four-Byte Filter Value 1	R/W

FOBFV1[31:0] bits (Four-Byte Filter Value 1)

- If FWFOBFCi. FOBFUM[1:0] is 00b: Used as a mask for FWFOBFV0Ci. FOBFV0[31:0].
- If FWFOBFCi. FOBFUM[1:0] is 01b: Used as an extension for FWFOBFV0Ci. FOBFV0[31:0] (Compare value is { FWFOBFV0Ci. FOBFV0[31:0], FWFOBFV1Ci. FOBFV1[31:0]}).
- If FWFOBFCi. FOBFUM[1:0] is 10b: Used as 4-byte filter value with a filter ID of 2 × (i + 36) + 1.

30.3.1.3.9 FWRFCi : Range Filter Configuration Register i (i = 0 to 15)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x1C00 + 0x10 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	RFOV[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	RFM	Range Filtering Mode*1 0: Offset filtering 1: VLAN TAG filtering	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
23:16	RFOV[7:0]	Range Filter Offset Value*2	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: This bit should not be set to 1 if FWGC.SVM is set to 00b.
Note 2.
- SW: If FWRFCi.RFM is set to 1, these bits should be set to 0 or 1 to filter S-TAG and set to 2 or 3 to filter C-TAG.
- SW: If FWRFCi.RFM is set to 1 and FWGC.SVM is set to 01b, these bits should not be set to 0 or 1.

RFOV[7:0] bits (Range Filter Offset Value)

- If FWRFCi.RFM is 0: Offset in byte where the filter i will start filtering the frame data.
- If FWRFCi.RFM is 1: Offset in byte where the filter i will start filtering the frame TAGs.

30.3.1.3.10 FWRFVCi : Range Filter Value Configuration Register i (i = 0 to 15)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x1C04 + 0x10 × i

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RFSV1[7:0]								RFSV0[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RFSV0[7:0]	Range Filter Start Value 0	R/W
15:8	RFSV1[7:0]	Range Filter Start Value 1	R/W
19:16	RFRV[3:0]	Range Filter Range Value*1 Used as range value for filters values FWRFVCi.RFSV1 and FWRFVCi.RFSV0.	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: FWRFVCi.RFSV0+FWRFVCi.RFRV and FWRFVCi.RFSV1+FWRFVCi.RFRV should always be smaller or equal to 255.

RFSV0[7:0] bits (Range Filter Start Value 0)

Used has range filter start value with a filter ID of 2 × (i + 49) (matching values are contained in [FWRFVCi.RFSV0: FWRFVCi.RFSV0+FWRFVCi.RFRV]).

RFSV1[7:0] bits (Range Filter Start Value 1)

Used has range filter start value with a filter ID of 2 × (i + 48) + 1 (matching values are contained in [FWRFVCi.RFSV1: FWRFVCi.RFSV1+FWRFVCi.RFRV]).

30.3.1.3.11 FWCFCi : Cascade Filter Configuration Register i (i = 0 to 15)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x2000 + 0x40 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFPFFV1 to CFPFFV0	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	CFEFFV2 to CFEFFV0		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CFEFFV2 to CFEFFV0	Cascade Filter E-Frame Filter Valid n (n = 0 to 2) Bit j is 0: Cascade filter i is disabled for port j e-frames (All slow ports frames are e-frames) [GWCA]. Bit j is 1: Cascade filter i is enabled for port j e-frames (All slow ports frames are e-frames) [GWCA].	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W
17:16	CFPFFV1 to CFPFFV0	Cascade Filter E-Frame Filter Valid n (n = 0 to 1) Bit j is 0: Cascade filter i is disabled for port j p-frames. Bit j is 1: Cascade filter i is enabled for port j p-frames.	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.3.12 FWCFMCij : Cascade Filter Mapping Configuration Register ij (i = 0 to 15) (j = 0 to 6)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x2004 + 0x40 × i + 0x4 ×j

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

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

Bit	Symbol	Function	R/W
6:0	CFFN[6:0]	Cascade Filter Filter Number Map filter number FWCFMCij.CFFN to cascade filter number i.	R/W

Bit	Symbol	Function	R/W
14:7	—	These bits are read as 0. The write value should be 0.	R/W
15	CFFV	Cascade Filter Valid Enables FWCFMCij.CFFN for cascade filter i. If no one of the FWCFMCij.CFFV is equal to 1, cascade filter i is disabled.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.4 Layer 3 Forwarding/Routing L2 Stream/Hash Filter Function Registers

30.3.1.4.1 FWIP4SC : IPv4 Stream Configuration Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	IP4IDPTS	IP4IIDS	IP4IISS	IP4ICDS	IP4ICPS	IP4ICVS	IP4ISDS	IP4ISPS	IP4ISVS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	IP4IDPTH	IP4ISPTH	IP4IPH	IP4IIDH	IP4IISH	IP4ICDH	IP4ICPH	IP4ICVH	IP4ISDH	IP4ISPH	IP4ISVH	IP4IMSH	IP4IMDH
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IP4IMDH	IPv4 Include MAC Destination in Hash 0: MAC Destination address is not included in hash calculation 1: MAC Destination address is included in hash calculation	R/W
1	IP4IMSH	IPv4 Include MAC Source in Hash 0: MAC Source address is not included in hash calculation 1: MAC Source address is included in hash calculation	R/W
2	IP4ISVH	IPv4 Include S-TAG VLAN ID in Hash*1 0: S-TAG VLAN ID is not included in hash calculation 1: S-TAG VLAN ID is included in hash calculation	R/W
3	IP4ISPH	IPv4 Include S-TAG PCP in Hash*1 0: S-TAG PCP is not included in hash calculation 1: S-TAG PCP is included in hash calculation	R/W
4	IP4ISDH	IPv4 Include S-TAG DEI in Hash*1 0: S-TAG DEI is not included in hash calculation 1: S-TAG DEI is included in hash calculation	R/W
5	IP4ICVH	IPv4 Include C-TAG VLAN ID in Hash*2 0: C-TAG VLAN ID is not included in hash calculation 1: C-TAG VLAN ID is included in hash calculation	R/W
6	IP4ICPH	IPv4 Include C-TAG PCP in Hash*2 0: C-TAG PCP is not included in hash calculation 1: C-TAG PCP is included in hash calculation	R/W
7	IP4ICDH	IPv4 Include C-TAG DEI in Hash*2 0: C-TAG DEI is not included in hash calculation 1: C-TAG DEI is included in hash calculation	R/W
8	IP4IISH	IPv4 Include IP Source in Hash 0: IPv4 Source IP address is not included in hash calculation 1: IPv4 Source IP address is included in hash calculation	R/W
9	IP4IIDH	IPv4 Include IP Destination in Hash 0: IPv4 Destination IP address is not included in hash calculation 1: IP v4 Destination IP address is included in hash calculation	R/W

Bit	Symbol	Function	R/W
10	IP4IPH	IPv4 Include Protocol in Hash 0: IPv4 Protocol is not included in hash calculation 1: IPv4 Protocol is included in hash calculation	R/W
11	IP4ISPTH	IPv4 Include Source Port in Hash 0: TCP/UDP Source port is not included in hash calculation 1: TCP/UDP Source port is included in hash calculation	R/W
12	IP4IDPTH	IPv4 Include Destination Port in Hash 0: TCP/UDP Destination port is not included in hash calculation 1: TCP/UDP Destination port is included in hash calculation	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
16	IP4ISVS	IPv4 Include S-TAG VLAN ID in Stream*1 0: S-TAG VLAN ID is not included in L3 IPv4 stream ID 1: S-TAG VLAN ID is included in L3 IPv4 stream ID	R/W
17	IP4ISPS	IPv4 Include S-TAG PCP in Stream*1 0: S-TAG PCP is not included in L3 IPv4 stream ID 1: S-TAG PCP is included in L3 IPv4 stream ID	R/W
18	IP4ISDS	IPv4 Include S-TAG DEI in Stream*1 0: S-TAG DEI is not included in L3 IPv4 stream ID 1: S-TAG DEI is included in L3 IPv4 stream ID	R/W
19	IP4ICVS	IPv4 Include C-TAG VLAN ID in Stream*2 0: C-TAG VLAN ID is not included in L3 IPv4 stream ID 1: C-TAG VLAN ID is included in L3 IPv4 stream ID	R/W
20	IP4ICPS	IPv4 Include C-TAG PCP in Stream*2 0: C-TAG PCP is not included in L3 IPv4 stream ID 1: C-TAG PCP is included in L3 IPv4 stream ID	R/W
21	IP4ICDS	IPv4 Include C-TAG DEI in Stream*2 0: C-TAG DEI is not included in L3 IPv4 stream ID 1: C-TAG DEI is included in L3 IPv4 stream ID	R/W
22	IP4IISS	IPv4 Include IP Source in Stream 0: IPv4 Source IP address is not included in L3 IPv4 stream ID 1: IPv4 Source IP address is included in L3 IPv4 stream ID	R/W
23	IP4IIDS	IPv4 Include IP Destination in Stream 0: IPv4 Destination IP address is not included in L3 IPv4 stream ID 1: IPv4 Destination IP address is included in L3 IPv4 stream ID	R/W
24	IP4IDPTS	IPv4 Include Destination Port in Stream 0: TCP/UDP Destination port is not included in L3 IPv4 stream ID 1: TCP/UDP Destination port is included in L3 IPv4 stream ID	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: If FWGC.SVM is set to 00b or 01b, this bit should be set to 0.
Note 2. SW: If FWGC.SVM is set to 00b, this bit should be set to 0.

30.3.1.4.2 FWIP6SC : IPv6 Stream Configuration Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4018

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	IP6IDPTS	IP6II1S	IP6II0S	IP6ICDS	IP6ICPS	IP6ICVS	IP6ISDS	IP6ISPS	IP6ISVS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	IP6IDPTH	IP6ISPTH	IP6IPH	IP6IIDH	IP6IISH	IP6ICDH	IP6ICPH	IP6ICVH	IP6ISDH	IP6ISPH	IP6ISVH	IP6IMSH	IP6IMDH
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IP6IMDH	IPv6 Include MAC Destination in Hash 0: MAC Destination address is not included in hash calculation 1: MAC Destination address is included in hash calculation	R/W
1	IP6IMSH	IPv6 Include MAC Source in Hash 0: MAC Source address is not included in hash calculation 1: MAC Source address is included in hash calculation	R/W
2	IP6ISVH	IPv6 Include S-TAG VLAN ID in Hash*1 0: S-TAG VLAN ID is not included in hash calculation 1: S-TAG VLAN ID is included in hash calculation	R/W
3	IP6ISPH	IPv6 Include S-TAG PCP in Hash*1 0: S-TAG PCP is not included in hash calculation 1: S-TAG PCP is included in hash calculation	R/W
4	IP6ISDH	IPv6 Include S-TAG DEI in Hash*1 0: S-TAG DEI is not included in hash calculation 1: S-TAG DEI is included in hash calculation	R/W
5	IP6ICVH	IPv6 Include C-TAG VLAN ID in Hash*2 0: C-TAG VLAN ID is not included in hash calculation 1: C-TAG VLAN ID is included in hash calculation	R/W
6	IP6ICPH	IPv6 Include C-TAG PCP in Hash*2 0: C-TAG PCP is not included in hash calculation 1: C-TAG PCP is included in hash calculation	R/W
7	IP6ICDH	IPv6 Include C-TAG DEI in Hash*2 0: C-TAG DEI is not included in hash calculation 1: C-TAG DEI is included in hash calculation	R/W
8	IP6IISH	IPv6 Include IP Source in Hash 0: IPv6 Source IP address is not included in hash calculation 1: IPv6 Source IP address is included in hash calculation	R/W
9	IP6IIDH	IPv6 Include IP Destination in Hash 0: IPv6 Destination IP address is not included in hash calculation 1: IPv6 Destination IP address is included in hash calculation	R/W
10	IP6IPH	IPv6 Include Protocol in Hash (Next Header) 0: IPv6 Protocol is not included in hash calculation 1: IPv6 Protocol is included in hash calculation	R/W
11	IP6ISPTH	IPv6 Include Source Port in Hash 0: TCP/UDP Source port is not included in hash calculation 1: TCP/UDP Source port is included in hash calculation	R/W
12	IP6IDPTH	IPv6 Include Destination Port in Hash 0: TCP/UDP Destination port is not included in hash calculation 1: TCP/UDP Destination port is included in hash calculation	R/W

Bit	Symbol	Function	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
16	IP6ISVS	IPv6 Include S-TAG VLAN ID in Stream*1 0: S-TAG VLAN ID is not included in L3 IPv6 stream ID 1: S-TAG VLAN ID is included in L3 IPv6 stream ID	R/W
17	IP6ISPS	IPv6 Include S-TAG PCP in Stream*1 0: S-TAG PCP is not included in L3 IPv6 stream ID 1: S-TAG PCP is included in L3 IPv6 stream ID	R/W
18	IP6ISDS	IPv6 Include S-TAG DEI in Stream*1 0: S-TAG DEI is not included in L3 IPv6 stream ID 1: S-TAG DEI is included in L3 IPv6 stream ID	R/W
19	IP6ICVS	IPv6 Include C-TAG VLAN ID in Stream*2 0: C-TAG VLAN ID is not included in L3 IPv6 stream ID 1: C-TAG VLAN ID is included in L3 IPv6 stream ID	R/W
20	IP6ICPS	IPv6 Include C-TAG PCP in Stream*2 0: C-TAG PCP is not included in L3 IPv6 stream ID 1: C-TAG PCP is included in L3 IPv6 stream ID	R/W
21	IP6ICDS	IPv6 Include C-TAG DEI in Stream*2 0: C-TAG DEI is not included in L3 IPv6 stream ID 1: C-TAG DEI is included in L3 IPv6 stream ID	R/W
22	IP6II0S	IPv6 Include IP 0 in Stream IPv6 IP address part 0 is a 4-byte byte length field extracted from the IP source or destination address depending on FWIP6OC configuration. 0: IPv6 IP address part 0 is not included in L3 IPv6 stream ID 1: IPv6 IP address part 0 is included in L3 IPv6 stream ID	R/W
23	IP6II1S	IPv6 Include IP 1 in Stream IPv6 IP address part 1 is a 4-byte byte length field extracted from the IP source or destination address depending on FWIP6OC configuration. 0: IPv6 IP address part 1 is not included in L3 IPv6 stream ID 1: IPv6 IP address part 1 is included in L3 IPv6 stream ID	R/W
24	IP6IDPTS	IPv6 Include Destination Port in Stream 0: TCP/UDP Destination port is not included in L3 IPv6 stream ID 1: TCP/UDP Destination port is included in L3 IPv6 stream ID	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: If FWGC.SVM is set to 00b or 01b, this bit should be set to 0.
Note 2. SW: If FWGC.SVM is set to 00b, this bit should be set to 0.

30.3.1.4.3 FWIP6OC : IPv6 Offset Configuration Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x401C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	IP6IPO1[3:0]				—	—	—	IP6IPOM1
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	IP6IPO0[3:0]				—	—	—	IP6IPOM0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IP6IPOM0	IPv6 IP Offset mode 0 0: IPv6 IP address part 0 is extracted from IP source address 1: IPv6 IP address part 0 is extracted from IP destination address	R/W

Bit	Symbol	Function	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
7:4	IP6IPO0[3:0]	IPv6 IP Offset 0*1 i: Byte [i:i+3] of selected IP address are used for IPv6 IP address part 0.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
16	IP6IPOM1	IPv6 IP Offset mode 1 0: IPv6 IP address part 1 is extracted from IP source address 1: IPv6 IP address part 1 is extracted from IP destination address	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
23:20	IP6IPO1[3:0]	IPv6 IP Offset 1*1 i: Byte [i:i+3] of selected IP address are used for IPv6 IP address part 1.	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: These bits should not be set to a value greater than 12.

30.3.1.4.4 FWL2SC : Layer 2 Stream Configuration Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4020

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	L2ICDS	L2ICPS	L2ICVS	L2ISDS	L2ISPS	L2ISVS	L2IMSS	L2IMDS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	L2IMDS	Layer 2 Include MAC Destination in Stream 0: MAC Destination address is not included in L2 stream ID 1: MAC Destination address is included in L2 stream ID	R/W
1	L2IMSS	Layer 2 Include MAC Source in Stream 0: MAC Source address is not included in L2 stream ID 1: MAC Source address is included in L2 stream ID	R/W
2	L2ISVS	Layer 2 Include S-TAG VLAN ID in Stream*1 0: MAC S-TAG VLAN ID is not included in L2 stream ID 1: MAC S-TAG VLAN ID is included in L2 stream ID	R/W
3	L2ISPS	Layer 2 Include S-TAG PCP ID in Stream*1 0: MAC S-TAG PCP is not included in L2 stream ID 1: MAC S-TAG PCP is included in L2 stream ID	R/W
4	L2ISDS	Layer 2 Include S-TAG DEI in Stream*1 0: MAC S-TAG DEI is not included in L2 stream ID 1: MAC S-TAG DEI is included in L2 stream ID	R/W
5	L2ICVS	Layer 2 Include C-TAG VLAN ID in Stream*2 0: MAC C-TAG VLAN ID is not included in L2 stream ID 1: MAC C-TAG VLAN ID is included in L2 stream ID	R/W
6	L2ICPS	Layer 2 Include C-TAG PCP ID in Stream*2 0: MAC C-TAG PCP is not included in L2 stream ID 1: MAC C-TAG PCP is included in L2 stream ID	R/W
7	L2ICDS	Layer 2 Include C-TAG DEI in Stream*2 0: MAC C-TAG DEI is not included in L2 stream ID 1: MAC C-TAG DEI is included in L2 stream ID	R/W

Bit	Symbol	Function	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: If FWGC.SVM is set to 00b or 01b, this bit should be set to 0.
Note 2. SW: If FWGC.SVM is set to 00b, this bit should be set to 0.

30.3.1.4.5 FWSFHEC : Stream Filter Hash Equation Configuration Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4030

Bit position:	31–16	15–0
Bit field:	IP6HE15 to IP6HE0	IP4HE15 to IP4HE0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	IP4HE15 to IP4HE0	IPv4 Stream Filter Hash Equation n (n = 0 to 15) If several Hash equations are activated, the result of all activated equation are XORed to give the final Hash ID. Bit 0 is set: Hash equation $1 + x^{17}$ is activated in Hash function for IPv4 Hash calculation. Bit i ($i \neq 0$) is set: Hash equation $1 + x^{i+1} + x^{17}$ is activated in Hash function for IPv4 Hash calculation.	R/W
31:16	IP6HE15 to IP6HE0	IPv6 Stream Filter Hash Equation n (n = 0 to 15) If several Hash equations are activated, the result of all activated equation are XORed to give the final Hash ID. Bit 0 is set: Hash equation $1 + x^{17}$ is activated in Hash function for IPv6 Hash calculation. Bit i ($i \neq 0$) is set: Hash equation $1 + x^{i+1} + x^{17}$ is activated in Hash function for IPv6 Hash calculation.	R/W

30.3.1.4.6 FWSHCR0 : Software Hash Calculation Request Register 0

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4040

Bit position:	31–0
Bit field:	SHCMDP0[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	SHCMDP0[31:0]	Software Hash Calculation MAC Destination Part 0 SW: This register is used to accelerate hash calculation.	R/W

30.3.1.4.7 FWSHCR1 : Software Hash Calculation Request Register 1

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4044

Bit position:	31–16	15–0
Bit field:	SHCMDP1[15:0]	SHCMSP0[15:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	SHCMSP0[15:0]	Software Hash Calculation MAC Source Part 0 SW: These bits are used to accelerate hash calculation.	R/W
31:16	SHCMDP1[15:0]	Software Hash Calculation MAC Destination Part 1 SW: These bits are used to accelerate hash calculation.	R/W

30.3.1.4.8 FWSHCR2 : Software Hash Calculation Request Register 2

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4048

Bit position:	31 … 0
Bit field:	SHCMSP1[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	SHCMSP1[31:0]	Software Hash Calculation MAC Source Part 1 SW: This register is used to accelerate hash calculation.	R/W

30.3.1.4.9 FWSHCR3 : Software Hash Calculation Request Register 3

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x404C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SHCSP[2:0]			SHCSD	SHCSV[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
11:0	SHCCV[11:0]	Software Hash Calculation C-TAG VLAN SW: These bits are used to accelerate hash calculation.	R/W
12	SHCCD	Software Hash Calculation C-TAG DEI SW: This bit is used to accelerate hash calculation.	R/W
15:13	SHCCP[2:0]	Software Hash Calculation C-TAG PCP SW: These bits are used to accelerate hash calculation.	R/W
27:16	SHCSV[11:0]	Software Hash Calculation S-TAG VLANs SW: These bits are used to accelerate hash calculation.	R/W
28	SHCSD	Software Hash Calculation S-TAG DEI SW: This bit is used to accelerate hash calculation.	R/W
31:29	SHCSP[2:0]	Software Hash Calculation S-TAG PCP SW: These bits are used to accelerate hash calculation.	R/W

30.3.1.4.10 FWSHCR4 : Software Hash Calculation Request Register 4

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4050

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

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

Bit	Symbol	Function	R/W
7:0	SHCP[7:0]	Software Hash Calculation Protocol (NextHeader for IPv6) SW: These bits are used to accelerate hash calculation.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
16	SHCFF	Software Hash Calculation Frame Format SW: This bit is used to accelerate hash calculation. 0: IPv4 Hash Calculation 1: IPv6 Hash Calculation	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.4.11 FWSHCR5 : Software Hash Calculation Request Register 5

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4054

Bit position: 31 ... 0

Bit field: SHCISP0[31:0]

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	SHCISP0[31:0]	Software Hash Calculation IP Source Part 0[*1] SW: This register is used to accelerate hash calculation.	R/W

Note 1. SW: Should be set to 0 for IPv4 Hash calculation.

30.3.1.4.12 FWSHCR6 : Software Hash Calculation Request Register 6

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4058

Bit position: 31 ... 0

Bit field: SHCISP1[31:0]

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	SHCISP1[31:0]	Software Hash Calculation IP Source Part 1[*1] SW: This register is used to accelerate hash calculation.	R/W

Note 1. SW: Should be set to 0 for IPv4 Hash calculation.

30.3.1.4.13 FWSHCR7 : Software Hash Calculation Request Register 7

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x405C

Bit position:	31 … 0
Bit field:	SHCISP2[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	SHCISP2[31:0]	Software Hash Calculation IP Source Part 2*1 SW: This register is used to accelerate hash calculation.	R/W

Note 1. SW: Should be set to 0 for IPv4 Hash calculation.

30.3.1.4.14 FWSHCR8 : Software Hash Calculation Request Register 8

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4060

Bit position:	31 … 0
Bit field:	SHCISP3[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	SHCISP3[31:0]	Software Hash Calculation IP Source Part 3*1 SW: This register is used to accelerate hash calculation.	R/W

Note 1. SW: Should be set to IP source address for IPv4 Hash calculation.

30.3.1.4.15 FWSHCR9 : Software Hash Calculation Request Register 9

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4064

Bit position:	31 … 0
Bit field:	SHCIDP0[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	SHCIDP0[31:0]	Software Hash Calculation IP Destination Part 0*1 SW: This register is used to accelerate hash calculation.	R/W

Note 1. SW: Should be set to 0 for IPv4 Hash calculation.

30.3.1.4.16 FWSHCR10 : Software Hash Calculation Request Register 10

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4068

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SHCIDP1[31:0]																															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SHCIDP1[31:0]	Software Hash Calculation IP Source Destination Part 1*1 SW: This register is used to accelerate hash calculation.	R/W

Note 1. SW: Should be set to 0 for IPv4 Hash calculation.

30.3.1.4.17 FWSHCR11 : Software Hash Calculation Request Register 11

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x406C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SHCIDP2[31:0]																															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SHCIDP2[31:0]	Software Hash Calculation IP Destination Part 2*1 SW: This register is used to accelerate hash calculation.	R/W

Note 1. - SW: Should be set to 0 for IPv4 Hash calculation.

30.3.1.4.18 FWSHCR12 : Software Hash Calculation Request Register 12

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4070

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SHCIDP3[31:0]																															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SHCIDP3[31:0]	Software Hash Calculation IP Destination Part 3*1 SW: This register is used to accelerate hash calculation.	R/W

Note 1. SW: Should be set to IP destination address for IPv4 Hash calculation.

30.3.1.4.19 FWSHCR13 : Software Hash Calculation Request Register 13

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4074

Bit position:	31–16	15–0
Bit field:	SHCSP[15:0]	SHCDP[15:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	SHCDP[15:0]	Software Hash Calculation Destination Port SW: These bits are used to accelerate hash calculation.	R/W
31:16	SHCSP[15:0]	Software Hash Calculation Source Port SW: These bits are used to accelerate hash calculation.	R/W

30.3.1.4.20 FWSHCRR : Software Hash Calculation Request Result Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4078

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

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

Bit	Symbol	Function	R/W
15:0	SHCR[15:0]	Software Hash Calculation Result SW: These bits are used to accelerate hash calculation.	R
30:16	—	These bits are read as 0.	R
31	SHC	Software Hash Calculation	R

SHC bit (Software Hash Calculation)

SW: This bit is used to accelerate hash calculation.

[Setting condition]

HW: Writing FWSHCR13 register will set this bit.

[Clearing condition]

HW: This bit is deasserted when hash calculation is completed.

30.3.1.5 Layer 3/4 Forwarding/Routing Hash Function Registers

30.3.1.5.1 FWLTHHEC : L3 Hash Entry Configuration Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4090

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	LTHHMUE[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
7:0	LTHHMC[7:0]	L3 Hash Maximum Collision Maximum collision number from which hash search by HW will be given up.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
24:16	LTHHMUE[8:0]	L3 Hash Maximum Unsecure Entry SW: Maximum unsecure entry number that can be set in table.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.5.2 FWLTHHC : L3 Hash Configuration Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4094

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	LTHHE7 to LTHHE0							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	LTHHE7 to LTHHE0	L3 Hash Equation n (n = 0 to 7) ● Bit 0 is set: Hash equation $1 + x^9$ is activated in Hash function for L3 Hash calculation. ● Bit i ($i \neq 0$) is set: Hash equation $1 + x^{i+1} + x^9$ is activated in Hash function for L3 Hash calculation.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

LTHHEn bits (L3 Hash Equation n (n = 0 to 7))

HW: If several Hash equations are activated, the result of all activated equation are XORed to give the final Hash ID.

30.3.1.5.3 FWLTHTL0 : L3 Table Learn Register 0

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x40A0

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	LTHSLL	—	—	—	—	—	LTHSLP0[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	LTHSLP0[2:0]	L3 Stream Learn Part 0	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	LTHSLL	L3 Security Level Learn Used for learning/overwriting an entry in L3 table.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	LTHED	L3 Entry Delete 0: Learn/overwrite the set stream ID in L3 table. 1: Delete the set stream ID in L3 table.	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

LTHSLP0[2:0] bits (L3 Stream Learn Part 0)

Used for learning/overwriting an entry in L3 table.

Stream ID correspond to {FWLTHTL0. LTHSLP0, FWLTHTL1. LTHSLP1, FWLTHTL2. LTHSLP2, FWLTHTL3. LTHSLP3, FWLTHTL4. LTHSLP4}.

30.3.1.5.4 FWLTHTL1 : L3 Table Learn Register 1

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x40A4

Bit position:	31 … 0
Bit field:	LTHSLP1[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	LTHSLP1[31:0]	L3 Stream Learn Part 1	R/W

LTHSLP1[31:0] bits (L3 Stream Learn Part 1)

Used for learning/overwriting an entry in L3 table.

Stream ID correspond to {FWLTHTL0. LTHSLP0, FWLTHTL1. LTHSLP1, FWLTHTL2. LTHSLP2, FWLTHTL3. LTHSLP3, FWLTHTL4. LTHSLP4}.

30.3.1.5.5 FWLTHTL2 : L3 Table Learn Register 2

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x40A8

Bit position: 31 … 0

Bit field: LTHSLP2[31:0]

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	LTHSLP2[31:0]	L3 Stream Learn Part 2	R/W

LTHSLP2[31:0] bits (L3 Stream Learn Part 2)

Used for learning/overwriting an entry in L3 table.

Stream ID correspond to {FWLTHTL0. LTHSLP0, FWLTHTL1. LTHSLP1, FWLTHTL2. LTHSLP2, FWLTHTL3. LTHSLP3, FWLTHTL4. LTHSLP4}.

30.3.1.5.6 FWLTHTL3 : L3 Table Learn Register 3

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x40AC

Bit position: 31 … 0

Bit field: LTHSLP3[31:0]

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	LTHSLP3[31:0]	L3 Stream Learn Part 3	R/W

LTHSLP3[31:0] bits (L3 Stream Learn Part 3)

Used for learning/overwriting an entry in L3 table.

Stream ID correspond to {FWLTHTL0. LTHSLP0, FWLTHTL1. LTHSLP1, FWLTHTL2. LTHSLP2, FWLTHTL3. LTHSLP3, FWLTHTL4. LTHSLP4}.

30.3.1.5.7 FWLTHTL4 : L3 Table Learn Register 4

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x40B0

Bit position: 31 … 0

Bit field: LTHSLP4[31:0]

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	LTHSLP4[31:0]	L3 Stream Learn Part 4	R/W

LTHSLP4[31:0] bits (L3 Stream Learn Part 4)

Used for learning/overwriting an entry in L3 table.

Stream ID correspond to {FWLTHTL0. LTHSLP0, FWLTHTL1. LTHSLP1, FWLTHTL2. LTHSLP2, FWLTHTL3. LTHSLP3, FWLTHTL4. LTHSLP4}.

30.3.1.5.8 FWLTHTL5 : L3 Table Learn Register 5

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x40B4

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

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

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
19:16	LTHMSDUNL[3:0]	L3 MSDU Number Learn SW: Used for learning/overwriting an entry in L3 table.	R/W
30:20	—	These bits are read as 0. The write value should be 0.	R/W
31	LTHMSDUVL	L3 MSDU Valid Learn Used for learning/overwriting an entry in L3 table.	R/W

30.3.1.5.9 FWLTHTL6 : L3 Table Learn Register 6

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x40B8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	LTHMTRVL	—	—	—	—	—	—	—	—	—	—	LTHMTRNL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
6:0	LTHFRERNL[6:0]	L3 FRER Number Learn Used for learning/overwriting an entry in L3 table.	R/W
14:7	—	These bits are read as 0. The write value should be 0.	R/W
15	LTHFRERVL	L3 FRER Valid Learn Used for learning/overwriting an entry in L3 table.	R/W
20:16	LTHMTRNL[4:0]	L3 Meter Number Learn Used for learning/overwriting an entry in L3 table.	R/W
30:21	—	These bits are read as 0. The write value should be 0.	R/W
31	LTHMTRVL	L3 Meter Valid Learn Used for learning/overwriting an entry in L3 table.	R/W

30.3.1.5.10 FWLTHTL7 : L3 Table Learn Register 7

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x40BC

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

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

Bit	Symbol	Function	R/W
7:0	LTHRNL[7:0]	L3 Routing Number Learn Used for learning/overwriting an entry in L3 table.	R/W
14:8	—	These bits are read as 0. The write value should be 0.	R/W
15	LTHRVL	L3 Routing Valid Learn Used for learning/overwriting an entry in L3 table.	R/W
18:16	LTHSLVL[2:0]	L3 Source Lock Vector Learn Used for learning/overwriting an entry in L3 table.	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.5.11 FWLTHTL80 : L3 Table Learn Register 80

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x40C0 + 0x4 × i

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

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

Bit	Symbol	Function	R/W
5:0	LTHCSDL[5:0]	L3 CPU Sub-Destination Learn Used for learning/overwriting an entry in L3 table.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.5.12 FWLTHTL9 : L3 Table Learn Register 9

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x40D0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	LTHCMEL	LTHEMEL	LTHIPUL	LTHIPVL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	LTHDVL[2:0]	L3 Destination Vector Learn Used for learning/overwriting an entry in L3 table.	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W
18:16	LTHIPVL[2:0]	L3 Internal Priority Value Learn Used for learning/overwriting an entry in L3 table.	R/W
19	LTHIPUL	L3 Internal Priority Update Learn Used for learning/overwriting an entry in L3 table.	R/W
20	LTHEMEL	L3 Ethernet Mirroring Enable Learn SW: Used for learning/overwriting an entry in L3 table.	R/W
21	LTHCMEL	L3 CPU Mirroring Enable Learn Used for learning/overwriting an entry in L3 table.	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.5.13 FWLTHTLR : L3 Table Learn Result Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x40D4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	LTHTL	—	—	—	—	—	—	—	LTHLCN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	LTHLF	L3 Learn Fail Used for learning/overwriting an entry in L3 table. 0: Entry learning did not fail. (See (2) SW Learning.) 1: Entry learning failed. (See (2) SW Learning.)	R
1	LTHLSF	L3 Learn Security Fail Used for learning/overwriting an entry in L3 table. 0: Entry learning did not fail. (See (2) SW Learning.) 1: Entry learning failed. (See (2) SW Learning.)	R
2	—	This bit is read as 0.	R

Bit	Symbol	Function	R/W
3	LTHLO	L3 Learn Overwrite Used for learning/overwriting an entry in L3 table. 0: The entry learning did not overwrite an existing entry. 1: The entry learning overwrote an existing entry.	R
15:4	—	These bits are read as 0.	R
23:16	LTHLCN[7:0]	L3 Learn Collision Number Used for learning/overwriting an entry in L3 table.	R
30:24	—	These bits are read as 0.	R
31	LTHTL	L3 Table Learn	R

LTHTL bit (L3 Table Learn)

Used for learning/overwriting an entry in L3 table.

[Setting condition]

HW: Writing FWLTHTL9 register will set this bit.

[Clearing condition]

HW: This bit will be de-asserted when learning is completed.

30.3.1.5.14 FWLTHTIM : L3 Table Initialization Monitoring Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x40E0

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

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

Bit	Symbol	Function	R/W
0	LTHTIOG	L3 Table Initialization Ongoing	R/W
1	LTHTR	L3 Table Ready	R
31:2	—	These bits are read as 0. The write value should be 0.	R/W

LTHTIOG bit (L3 Table Initialization Ongoing)

[Setting condition]

SW: By writing 1 to this bit. It starts L3 Table initialization.

[Clearing condition]

HW: This bit is cleared when L3 Table initialization is finished.

LTHTR bit (L3 Table Ready)

[Setting condition]

- HW: When FWLTHTIM.LTHTIOG is getting cleared.
- HW: This bit is set at clk_period [ns] × 256 time from L3 table initialization starting.

[Clearing condition]

SW: By writing 1 to FWLTHTIM.LTHTIOG.

30.3.1.5.15 FWLTHTEM : L3 Table Entry Monitoring Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x40E4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	LTHTUEN[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
8:0	LTHTEN[8:0]	L3 Table Entry Number	R
15:9	—	These bits are read as 0.	R
24:16	LTHTUEN[8:0]	L3 Table Unsecure Entry Number	R
31:25	—	These bits are read as 0.	R

LTHTEN[8:0] bits (L3 Table Entry Number)

[Increment condition]

HW: An entry is learned successfully.

[Decrement condition]

HW: An entry is deleted successfully.

LTHTUEN[8:0] bits (L3 Table Unsecure Entry Number)

[Increment condition]

HW: An unsecure entry is learned successfully.

[Decrement condition]

HW: An unsecure entry is deleted successfully.

30.3.1.5.16 FWLTHTS0 : L3 Table Search Register 0

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4100

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	LTHSSP0[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	LTHSSP0[2:0]	L3 Stream Search Part 0 Used for searching an entry in L3 table.	R/W

Bit	Symbol	Function	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.5.17 FWLTHTS1 : L3 Table Search Register 1

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4104

Bit position:	31 … 0
Bit field:	LTHSSP1[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	LTHSSP1[31:0]	L3 Stream Search Part 1 Used for searching an entry in L3 table.	R/W

30.3.1.5.18 FWLTHTS2 : L3 Table Search Register 2

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4108

Bit position:	31 … 0
Bit field:	LTHSSP2[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	LTHSSP2[31:0]	L3 Stream Search Part 2 Used for searching an entry in L3 table.	R/W

30.3.1.5.19 FWLTHTS3 : L3 Table Search Register 3

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x410C

Bit position:	31 … 0
Bit field:	LTHSSP3[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	LTHSSP3[31:0]	L3 Stream Search Part 3 Used for searching an entry in L3 table.	R/W

30.3.1.5.20 FWLTHTS4 : L3 Table Search Register 4

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4110

Bit position:	31		0
Bit field:		LTHSSP4[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		

Bit	Symbol	Function	R/W
31:0	LTHSSP4[31:0]	L3 Stream Search Part 4 Used for searching an entry in L3 table.	R/W

30.3.1.5.21 FWLTHTSR0 : L3 Table Search Result Register 0

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4120

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	LTHTS	—	—	—	—	—	—	—	LTHSCN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	LTHSNF	L3 Search Not found Used for searching an entry in L3 table. 0: Stream ID found in L3 Table. 1: Stream ID not found in L3 Table.	R
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	LTHSLS	L3 Security Level Search Used for searching an entry in L3 table.	R
15:9	—	These bits are read as 0. The write value should be 0.	R/W
23:16	LTHSCN[7:0]	L3 Search Collision Number Used for searching an entry in L3 table.	R
30:24	—	These bits are read as 0. The write value should be 0.	R/W
31	LTHTS	L3 Table Search	R

LTHTS bit (L3 Table Search)

Used for searching an entry in L3 table.

[Setting condition]

HW: Writing FWLTHTS4 register will set this bit.

[Clearing condition]

HW: This bit will be de-asserted when searching is completed.

30.3.1.5.22 FWLTHTSR1 : L3 Table Search Result Register 1

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4124

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

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

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0.	R
19:16	LTHMSDUNS[3:0]	L3 MSDU Number Search Used for searching an entry in L3 table.	R
30:20	—	These bits are read as 0.	R
31	LTHMSDUVS	L3 MSDU Valid Search Used for searching an entry in L3 table.	R

30.3.1.5.23 FWLTHTSR2 : L3 Table Search Result Register 2

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4128

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	LTHMTRVS	—	—	—	—	—	—	—	—	—	—	LTHMTRNS[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
6:0	LTHFRERNS[6:0]	L3 FRER Number Search Used for searching an entry in L3 table.	R
14:7	—	These bits are read as 0.	R
15	LTHFRERVS	L3 FRER Valid Search Used for searching an entry in L3 table.	R
20:16	LTHMTRNS[4:0]	L3 Meter Number Search Used for searching an entry in L3 table.	R
30:21	—	These bits are read as 0.	R
31	LTHMTRVS	L3 Meter Valid Search Used for searching an entry in L3 table.	R

30.3.1.5.24 FWLTHTSR3 : L3 Table Search Result Register 3

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x412C

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

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

Bit	Symbol	Function	R/W
2:0	LTHRNS[2:0]	L3 Routing Number Search Used for searching an entry in L3 table.	R
14:3	—	These bits are read as 0.	R
15	LTHRVS	L3 Routing Valid Search Used for searching an entry in L3 table.	R
18:16	LTHSLVS[2:0]	L3 Source Lock Vector Search Used for searching an entry in L3 table.	R
31:19	—	These bits are read as 0.	R

30.3.1.5.25 FWLTHTSR40 : L3 Table Search Result Register 40

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4130 + 0x4 × i

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

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

Bit	Symbol	Function	R/W
5:0	LTHCSDS[5:0]	L3 CPU Sub-Destination Search Used for searching an entry in L3 table.	R
31:6	—	These bits are read as 0.	R

30.3.1.5.26 FWLTHTSR5 : L3 Table Search Result Register 5

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4140

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	LTHCMES	LTHEMES	LTHIPUS	LTHIPVS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	LTHDVS[2:0]	L3 Destination Vector Search Used for searching an entry in L3 table.	R
15:3	—	These bits are read as 0.	R
18:16	LTHIPVS[2:0]	L3 Internal Priority Value Search Used for searching an entry in L3 table.	R
19	LTHIPUS	L3 Internal Priority Update Search Used for searching an entry in L3 table.	R
20	LTHEMES	L3 Ethernet Mirroring Enable Search Used for searching an entry in L3 table.	R
21	LTHCMES	L3 CPU Mirroring Enable Search Used for searching an entry in L3 table.	R
31:22	—	These bits are read as 0.	R

30.3.1.5.27 FWLTHTR : L3 Table Read Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4150

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

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

Bit	Symbol	Function	R/W
7:0	LTHAR[7:0]	L3 Address Read Used for reading an entry in L3 table.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.5.28 FWLTHTRR0 : L3 Table Read Result Register 0

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4154

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

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

Bit	Symbol	Function	R/W
0	—	This bit is read as 0.	R
1	LTHEVR	L3 Entry Valid Read Used for reading an entry in L3 table.	R
30:2	—	These bits are read as 0.	R
31	LTHTR	L3 Table Read	R

LTHTR bit (L3 Table Read)

Used for reading an entry in L3 table.

[Setting condition]

HW: Writing FWLTHTR register will set this bit.

[Clearing condition]

HW: This bit will be de-asserted when reading is completed.

30.3.1.5.29 FWLTHTRR1 : L3 Table Read Result Register 1

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4158

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	LTHSLR	—	—	—	—	—	LTHSRP0[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	LTHSRP0[2:0]	L3 Stream Read Part 0 Used for reading an entry in L3 table.	R
7:3	—	These bits are read as 0.	R
8	LTHSLR	L3 Security Level Read Used for reading an entry in L3 table.	R
31:9	—	These bits are read as 0.	R

30.3.1.5.30 FWLTHTRR2 : L3 Table Read Result Register 2

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x415C

Bit position:	31 … 0
Bit field:	LTHSRP1[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	LTHSRP1[31:0]	L3 Stream Read Part 1 Used for reading an entry in L3 table.	R

30.3.1.5.31 FWLTHTRR3 : L3 Table Read Result Register 3

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4160

Bit position:	31 … 0
Bit field:	LTHSRP2[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	LTHSRP2[31:0]	L3 Stream Read Part 2 Used for reading an entry in L3 table.	R

30.3.1.5.32 FWLTHTRR4 : L3 Table Read Result Register 4

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4164

Bit position:	31 … 0
Bit field:	LTHSRP3[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	LTHSRP3[31:0]	L3 Stream Read Part 3 Used for reading an entry in L3 table.	R

30.3.1.5.33 FWLTHTRR5 : L3 Table Read Result Register 5

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4168

Bit position:	31 … 0
Bit field:	LTHSRP4[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	LTHSRP4[31:0]	L3 Stream Read Part 4 Used for reading an entry in L3 table.	R

30.3.1.5.34 FWLTHTRR6 : L3 Table Read Result Register 6

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x416C

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

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

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0.	R
19:16	LTHMSDUNR[3:0]	L3 MSDU Number Read Used for reading an entry in L3 table.	R
30:20	—	These bits are read as 0.	R
31	LTHMSDUVR	L3 MSDU Valid Read Used for reading an entry in L3 table.	R

30.3.1.5.35 FWLTHTRR7 : L3 Table Read Result Register 7

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4170

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	LTHMTRVR	—	—	—	—	—	—	—	—	—	—	LTHMTRNR[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
6:0	LTHFRERNR[6:0]	L3 FRER Number Read Used for reading an entry in L3 table.	R
14:7	—	These bits are read as 0.	R
15	LTHFRERVR	L3 FRER Valid Read Used for reading an entry in L3 table.	R
20:16	LTHMTRNR[4:0]	L3 Meter Number Read Used for reading an entry in L3 table.	R
30:21	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
31	LTHMTRVR	L3 Meter Valid Read Used for reading an entry in L3 table.	R

30.3.1.5.36 FWLTHTRR8 : L3 Table Read Result Register 8

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4174

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

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

Bit	Symbol	Function	R/W
7:0	LTHRNR[7:0]	L3 Routing Number Read Used for reading an entry in L3 table.	R
14:8	—	These bits are read as 0.	R
15	LTHRVR	L3 Routing Valid Read Used for reading an entry in L3 table.	R
18:16	LTHSLVR[2:0]	L3 Source Lock Vector Read Used for reading an entry in L3 table.	R
31:19	—	These bits are read as 0.	R

30.3.1.5.37 FWLTHTRR90 : L3 Table Read Result Register 90

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4180

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

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

Bit	Symbol	Function	R/W
5:0	LTHCSDR[5:0]	L3 CPU Sub-Destination Read Used for reading an entry in L3 table.	R
31:6	—	These bits are read as 0.	R

30.3.1.5.38 FWLTHTRR10 : L3 Table Read Result Register 10

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4190

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	LTHCMER	LTHEMER	LTHIPUR	LTHIPVR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	LTHDVR[2:0]	L3 Destination Vector Read Used for reading an entry in L3 table.	R
15:3	—	These bits are read as 0.	R
18:16	LTHIPVR[2:0]	L3 Internal Priority Value Read Used for reading an entry in L3 table.	R
19	LTHIPUR	L3 Internal Priority Update Read Used for reading an entry in L3 table.	R
20	LTHEMER	L3 Ethernet Mirroring Enable Read Used for reading an entry in L3 table.	R
21	LTHCMER	L3 CPU Mirroring Enable Read Used for reading an entry in L3 table.	R
31:22	—	These bits are read as 0.	R

30.3.1.6 Layer 2 Forwarding Function Registers

30.3.1.6.1 FWMACHEC : MAC Hash Entry Configuration Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4620

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	MACHMUE[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
10:0	MACHMC[10:0]	MAC Hash Maximum Collision Maximum collision number from which hash search by HW will be given up.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
27:16	MACHMUE[11:0]	MAC Hash Maximum Unsecure Entry SW: Maximum unsecure entry number that can be set in table.	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.6.2 FWMACHC : MAC Hash Configuration Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4624

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	MACHE10 to MACHE0										
Value after reset:	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
10:0	MACHE10 to MACHE0	MAC Hash Equation n (n = 0 to 10) ● Bit 0 is set: Hash equation $1 + x^{12}$ is activated in Hash function for MAC Hash calculation. ● Bit i (i ≠ 0) is set: Hash equation $1 + x^{i+1} + x^{12}$ is activated in Hash function for MAC Hash calculation.	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

MACHEn bits (MAC Hash Equation n (n = 0 to 10))

HW: If several Hash equations are activated, the result of all activated equation are XORed to give the final Hash ID.

30.3.1.6.3 FWMACTL0 : MAC Table Learn Register 0

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4630

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

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

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	MACSLL	MAC Security Level Learn Used for learning/overwriting an entry in MAC table.	R/W
9	MACDEL	MAC Dynamic Entry Learn Used for learning/overwriting an entry in MAC table.	R/W
10	MACHLDL	MAC Hardware Learning Disable Learn Used for learning/overwriting an entry in MAC table.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
16	MACED	MAC Entry Delete 0: Learn/overwrite the set MAC address in MAC table. 1: Delete the set MAC address in MAC table.	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.6.4 FWMACTL1 : MAC Table Learn Register 1

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4634

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MACMALP0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	MACMALP0[15:0]	MAC MAC address Learn Part 0 Used for learning/overwriting an entry in MAC table.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.6.5 FWMACTL2 : MAC Table Learn Register 2

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4638

Bit position:	31 ... 0
Bit field:	MACMALP1[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	MACMALP1[31:0]	MAC MAC address Learn Part 1 Used for learning/overwriting an entry in MAC table.	R/W

30.3.1.6.6 FWMACTL3 : MAC Table Learn Register 3

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x463C

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

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

Bit	Symbol	Function	R/W
2:0	MACSSLVL[2:0]	MAC Source Source Lock Vector Learn Used for learning/overwriting an entry in MAC table.	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
18:16	MACDSLVL[2:0]	MAC Destination Source Lock Vector Learn Used for learning/overwriting an entry in MAC table.	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.6.7 FWMACTL40 : MAC Table Learn Register 40

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4640 + 0x4 × i

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

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

Bit	Symbol	Function	R/W
5:0	MACCSDL[5:0]	MAC CPU Sub-Destination Learn Used for learning/overwriting an entry in MAC table.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.6.8 FWMACTL5 : MAC Table Learn Register 5

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4650

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	MACCMEL	MACEMEL	MACIPUL	MACIPVL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	MACDVL[2:0]	MAC Destination Vector Learn Used for learning/overwriting an entry in MAC table.	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W
18:16	MACIPVL[2:0]	MAC Internal Priority Value Learn Used for learning/overwriting an entry in MAC table.	R/W
19	MACIPUL	MAC Internal Priority Update Learn Used for learning/overwriting an entry in MAC table.	R/W
20	MACEMEL	MAC Ethernet Mirroring Enable Learn Used for learning/overwriting an entry in MAC table.	R/W
21	MACCMEL	MAC CPU Mirroring Enable Learn Used for learning/overwriting an entry in MAC table.	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.6.9 FWMACTLR : MAC Table Learn Result Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4654

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MACTL	—	—	—	—	MACLCN[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	MACLF	MAC Learn Fail Used for learning/overwriting an entry in MAC table. 0: Entry learning did not fail. (See (2) SW Learning). 1: Entry learning failed. (See (2) SW Learning).	R
1	MACLSF	MAC Learn Security Fail 0: Entry learning did not fail because of a security error. (See (2) SW Learning). 1: Entry learning failed because of a security error. (See (2) SW Learning).	R
2	—	This bit is read as 0.	R
3	MACLO	MAC Learn Overwrite Used for learning/overwriting an entry in MAC table. 0: The entry learning did not overwrite an existing entry. 1: The entry learning overwrote an existing entry.	R
15:4	—	These bits are read as 0.	R
26:16	MACLCN[10:0]	MAC Learn Collision Number Used for learning/overwriting an entry in MAC table.	R
30:27	—	These bits are read as 0.	R
31	MACTL	MAC Table Learn	R

MACTL bit (MAC Table Learn)

Used for learning/overwriting an entry in MAC table.

[Setting condition]

HW: Writing FWMACTL5 register will set this bit.

[Clearing condition]

HW: This bit will be de-asserted when learning is completed.

30.3.1.6.10 FWMACTIM : MAC Table Initialization Monitoring Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4660

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

Bit	Symbol	Function	R/W
0	MACTIOG	MAC Table Initialization Ongoing	R/W
1	MACTR	MAC Table Ready	R
31:2	—	These bits are read as 0. The write value should be 0.	R/W

MACTIOG bit (MAC Table Initialization Ongoing)

[Setting condition]

SW: By writing 1 to this bit. It starts MAC Table initialization.

[Clearing condition]

HW: This bit is cleared when MAC Table initialization is finished.

MACTR bit (MAC Table Ready)

[Setting condition]

- HW: When FWMACTIM.MACTIOG is getting cleared.
- HW: This bit is set at clk_period [ns] × 2048 time from MAC table initialization starting.

[Clearing condition]

SW: By writing 1 to FWMACTIM.MACTIOG.

30.3.1.6.11 FWMACTEM : MAC Table Entry Monitoring Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4664

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

Bit	Symbol	Function	R/W
11:0	MACTEN[11:0]	MAC Table Entry Number	R

Bit	Symbol	Function	R/W
15:12	—	These bits are read as 0.	R
27:16	MACTUEN[11:0]	MAC Table Unsecure Entry Number	R
31:28	—	These bits are read as 0.	R

MACTEN[11:0] bits (MAC Table Entry Number)

[Increment condition]

HW: An entry is learned successfully.

[Decrement condition]

HW: An entry is deleted successfully.

MACTUEN[11:0] bits (MAC Table Unsecure Entry Number)

[Increment condition]

HW: An unsecure entry is learned successfully.

[Decrement condition]

HW: An unsecure entry is deleted successfully.

30.3.1.6.12 FWMACTS0 : MAC Table Search Register 0

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4670

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MACMASP0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	MACMASP0[15:0]	MAC MAC Address Search Part 0 Used for searching an entry in MAC table.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.6.13 FWMACTS1 : MAC Table Search Register 1

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4674

Bit position:	31 ... 0
Bit field:	MACMASP1[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	MACMASP1[31:0]	MAC MAC Address Search Part 1 Used for searching an entry in MAC table.	R/W

30.3.1.6.14 FWMACTSR0 : MAC Table Search Result Register 0

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4678

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MACTS	—	—	—	—	MACSCN[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	MACSNF	MAC Search Not found Used for searching an entry in MAC table. 0: MAC address found in MAC Table. 1: MAC address not found in MAC Table.	R
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	MACSLS	MAC Security Level Search Used for searching an entry in MAC table.	R
9	MACDES	MAC Dynamic Entry Search Used for searching an entry in MAC table.	R
10	MACHLDS	MAC Hardware Learning Disable Search Used for searching an entry in MAC table.	R
15:11	—	These bits are read as 0. The write value should be 0.	R/W
26:16	MACSCN[10:0]	MAC Search Collision Number Used for searching an entry in MAC table.	R
30:27	—	These bits are read as 0. The write value should be 0.	R/W
31	MACTS	MAC Table Search	R

MACTS bit (MAC Table Search)

Used for searching an entry in MAC table.

[Setting condition]

HW: Writing FWMACTS1 register will set this bit.

[Clearing condition]

HW: This bit will be de-asserted when searching is completed.

30.3.1.6.15 FWMACTSR1 : MAC Table Search Result Register 1

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x467C

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

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

Bit	Symbol	Function	R/W
2:0	MACSSLVS[2:0]	MAC Source Source Lock Vector Search Used for searching an entry in MAC table.	R
15:3	—	These bits are read as 0.	R
18:16	MACDSLVS[2:0]	MAC Destination Source Lock Vector Search Used for searching an entry in MAC table.	R
31:19	—	These bits are read as 0.	R

30.3.1.6.16 FWMACTSR20 : MAC Table Search Result Register 20

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4680 + 0x4 × i

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

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

Bit	Symbol	Function	R/W
5:0	MACCSDS[5:0]	MAC CPU Sub-Destination Search Used for searching an entry in MAC table.	R
31:6	—	These bits are read as 0.	R

30.3.1.6.17 FWMACTSR3 : MAC Table Search Result Register 3

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4690

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	MACCMES	MACEMES	MACIPUS	MACIPVS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	MACDVS[2:0]	MAC Destination Vector Search Used for searching an entry in MAC table.	R
15:3	—	These bits are read as 0.	R
18:16	MACIPVS[2:0]	MAC Internal Priority Value Search Used for searching an entry in MAC table.	R
19	MACIPUS	MAC Internal Priority Update Search Used for searching an entry in MAC table.	R
20	MACEMES	MAC Ethernet Mirroring Enable Search Used for searching an entry in MAC table.	R
21	MACCMES	MAC CPU Mirroring Enable Search Used for searching an entry in MAC table.	R
31:22	—	These bits are read as 0.	R

30.3.1.6.18 FWMACTR : MAC Table Read Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x46A0

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

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

Bit	Symbol	Function	R/W
10:0	MACAR[10:0]	MAC Address Read Used for reading an entry in MAC table	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.6.19 FWMACTRR0 : MAC Table Read Result Register 0

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x46A4

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

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

Bit	Symbol	Function	R/W
0	MACEVR	MAC Entry Valid Read - Used for reading an entry in MAC table.	R
30:1	—	These bits are read as 0.	R
31	MACTR	MAC Table Read Used for reading an entry in MAC table.	R

MACTR bit (MAC Table Read)

[Setting condition]

HW: Writing FWMACTR register will set this bit.

[Clearing condition]

HW: This bit will be de-asserted when reading is completed.

30.3.1.6.20 FWMACTRR1 : MAC Table Read Result Register 1

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x46A8

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

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

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0.	R
8	MACSLR	MAC Security Level Read Used for reading an entry in MAC table.	R
9	MACDER	MAC Dynamic Entry Read Used for reading an entry in MAC table.	R
10	MACHLDR	MAC Hardware Learn Disable Read Used for reading an entry in MAC table.	R
11	MACABR	MAC Aging Bit Read Used for reading an entry in MAC table.	R

Bit	Symbol	Function	R/W
31:12	—	These bits are read as 0.	R

30.3.1.6.21 FWMACTRR2 : MAC Table Read Result Register 2

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x46AC

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MACMARP0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	MACMARP0[15:0]	MAC MAC address Read Part 0 Used for reading an entry in MAC table.	R
31:16	—	These bits are read as 0.	R

30.3.1.6.22 FWMACTRR3 : MAC Table Read Result Register 3

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x46B0

Bit position:	31 ... 0
Bit field:	MACMARP1[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	MACMARP1[31:0]	MAC MAC Address Read Part 1 Used for reading an entry in MAC table.	R

30.3.1.6.23 FWMACTRR4 : MAC Table Read Result Register 4

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x46B4

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

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

Bit	Symbol	Function	R/W
2:0	MACSSLVR[2:0]	MAC Source Source Lock Vector Read Used for reading an entry in MAC table.	R
15:3	—	These bits are read as 0.	R
18:16	MACDSLVR[2:0]	MAC Destination Source Lock Vector Read Used for reading an entry in MAC table.	R
31:19	—	These bits are read as 0.	R

30.3.1.6.24 FWMACTRR50 : MAC Table Read Result Register 50

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x46C0 + 0x4 × i

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

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

Bit	Symbol	Function	R/W
5:0	MACCSDR[5:0]	MAC CPU Sub-Destination Read Used for reading an entry in MAC table.	R
31:6	—	These bits are read as 0.	R

30.3.1.6.25 FWMACTRR6 : MAC Table Read Result Register 6

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x46D0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	MACCMER	MACEMER	MACIPUR	MACIPVR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	MACDVR[2:0]	MAC Destination Vector Read Used for reading an entry in MAC table.	R
15:3	—	These bits are read as 0.	R
18:16	MACIPVR[2:0]	MAC Internal Priority Value Read Used for reading an entry in MAC table.	R
19	MACIPUR	MAC Internal Priority Update Read Used for reading an entry in MAC table.	R
20	MACEMER	MAC Ethernet Mirroring Enable Read Used for reading an entry in MAC table.	R

Bit	Symbol	Function	R/W
21	MACCMER	MAC CPU Mirroring Enable Read Used for reading an entry in MAC table.	R
31:22	—	These bits are read as 0.	R

30.3.1.6.26 FWMACAGUSPC : MAC Aging US Prescaler Configuration Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4880

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

Bit	Symbol	Function	R/W
9:0	MACAGUSP[9:0]*1	MAC Aging US prescaler Used to creates an internal clock for aging at 1 MHz to derive the aging 1 Hz clock.	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: These bits should be set to the number of clocks clk contained in 1 µs.

30.3.1.6.27 FWMACAGC : MAC Aging Configuration Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4884

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

Bit	Symbol	Function	R/W
15:0	MACAGT[15:0]	MAC Aging Time Defines the time in seconds the MFWD will wait between two MAC address aging.	R/W
16	MACAGE	MAC Aging Enable 0: MAC aging is disabled 1: MAC aging is enabled and will happen every FWMACAGC.MACAGT seconds.	R/W
17	MACAGSL	MAC Aging Security Level 0: Only unsecure entries will be aged (MAC.SL == 0) 1: All entries will be aged.	R/W
18	MACAGPM	MAC Aging Polling Mode When set, allows software to check all MAC addresses suppresses by aging. 0: MAC aging polling mode disabled 1: MAC aging polling mode enabled	R/W

Bit	Symbol	Function	R/W
23:19	—	These bits are read as 0. The write value should be 0.	R/W
24	MACDES*1	MAC Dynamic Entry Suppression Writing 1 to this bit will start dynamic entry suppression.	R
27:25	—	These bits are read as 0. The write value should be 0.	R/W
28	MACAGOG	MAC Aging Ongoing 0: MAC aging is not ongoing. 1: MAC aging is ongoing.	R
29	MACDESOG	MAC Dynamic Entry Suppression Ongoing 0: MAC Dynamic Entry Suppression is not ongoing. 1: MAC Dynamic Entry Suppression is ongoing.	R
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Dynamic Entry Suppression can only happen when aging is disabled (FWMACAGC.MACAGE is not set)

30.3.1.6.28 FWMACAGM0 : MAC Aging Monitoring Register 0

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4888

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AGMACAP0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	AGMACAP0[15:0]	Aged MAC Address Part 0 Shows the MAC address part 0 of the latest aged entry.	R
31:16	—	These bits are read as 0.	R

30.3.1.6.29 FWMACAGM1 : MAC Aging Monitoring Register 1

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x488C

Bit position:	31 ... 0
Bit field:	AGMACAP1[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	AGMACAP1[31:0]	Aged MAC Address Part 1 Shows the MAC address part 1 of the latest aged entry.	R

30.3.1.6.30 FWVLANTEC : VLAN Table Entry Configuration Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4900

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	VLANTMUE[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
28:16	VLANTMUE[12:0]	VLAN Table Maximum Unsecure Entry SW: Maximum unsecure entry number that can be set in table	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.6.31 FWVLANTL0 : VLAN Table Learn Register 0

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4910

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

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

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	VLANSLL	VLAN Security Level Learn Used for learning/overwriting an entry in VLAN table.	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
10	VLANHLDL	VLAN Hardware Learning Disable Learn Used for learning/overwriting an entry in VLAN table.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
16	VLANED	VLAN Entry Delete Used for learning/overwriting an entry in VLAN table. 0: Learn/overwrite the set VLAN address in VLAN table. 1: Delete the set VLAN address in VLAN table.	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.6.32 FWVLANTL1 : VLAN Table Learn Register 1

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4914

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

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

Bit	Symbol	Function	R/W
11:0	VLANVIDL[11:0]	VLAN VID Learn Used for learning/overwriting an entry in VLAN table.	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.6.33 FWVLANTL2 : VLAN Table Learn Register 2

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4918

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

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

Bit	Symbol	Function	R/W
2:0	VLANSLVL[2:0]	VLAN Source Lock Vector Learn Used for learning/overwriting an entry in VLAN table.	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.6.34 FWVLANTL30 : VLAN Table Learn Register 30

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4920 + 0x4 × i

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

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

Bit	Symbol	Function	R/W
5:0	VLANCSDL[5:0]	VLAN CPU Sub-Destination Learn Used for learning/overwriting an entry in VLAN table.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.6.35 FWVLANTL4 : VLAN Table Learn Register 4

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4930

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	VLANCMEL	VLANEMEL	VLANIPUL	VLANIPVL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	VLANDVL[2:0]	VLAN Destination Vector Learn Used for learning/overwriting an entry in VLAN table.	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W
18:16	VLANIPVL[2:0]	VLAN Internal Priority Value Learn Used for learning/overwriting an entry in VLAN table.	R/W
19	VLANIPUL	VLAN Internal Priority Update Learn Used for learning/overwriting an entry in VLAN table.	R/W
20	VLANEMEL	VLAN Ethernet Mirroring Enable Learn Used for learning/overwriting an entry in VLAN table.	R/W
21	VLANCMEL	VLAN CPU Mirroring Enable Learn Used for learning/overwriting an entry in VLAN table.	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.6.36 FWVLANTLR : VLAN Table Learn Result Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4934

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

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

Bit	Symbol	Function	R/W
0	VLANLF	VLAN Learn Fail Used for learning/overwriting an entry in VLAN table. 0: Entry learning did not fail. (See (2) SW Learning.) 1: Entry learning failed. (See (2) SW Learning.)	R
1	VLANLSF	VLAN Learn Security Fail Used for learning/overwriting an entry in VLAN table. 0: Entry learning did not fail because of a security error. (See (2) SW Learning.) 1: Entry learning failed because of a security error. (See (2) SW Learning.)	R
2	—	This bit is read as 0.	R
3	VLANLO	VLAN Learn Overwrite Used for learning/overwriting an entry in VLAN table. 0: The entry learning did not overwrite an existing entry. 1: The entry learning overwrote an existing entry.	R
30:4	—	These bits are read as 0.	R
31	VLANTL	VLAN Table Learn	R

VLANTL bit (VLAN Table Learn)

Used for learning/overwriting an entry in VLAN table.

[Setting condition]

HW: Writing FWVLANTL4 register will set this bit.

[Clearing condition]

HW: This bit will be de-asserted when learning is completed.

30.3.1.6.37 FWVLANTIM : VLAN Table Initialization Monitoring Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4940

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

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

Bit	Symbol	Function	R/W
0	VLANTIOG	VLAN Table Initialization Ongoing	R/W*1
1	VLANTR	VLAN Table Ready	R
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

VLANTIOG bit (VLAN Table Initialization Ongoing)

[Setting condition]

SW: By writing 1 to this bit. It starts VLAN Table initialization.

[Clearing condition]

HW: This bit is cleared when VLAN Table initialization is finished.

VLANTR bit (VLAN Table Ready)

[Setting condition]

HW: When FWVLANTIM.VLANTIOG is getting cleared.

HW: This bit is set at “clk_period[ns]*4096” time from VLAN table initialization starting.

[Clearing condition]

SW: By writing 1 to FWVLANTIM.VLANTIOG.

30.3.1.6.38 FWVLANTEM : VLAN Table Entry Monitoring Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4944

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	VLANTUEN[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
12:0	VLANTEN[12:0]	VLAN Table Entry Number	R
15:13	—	These bits are read as 0.	R
28:16	VLANTUEN[12:0]	VLAN Table Unsecure Entry Number	R
31:29	—	These bits are read as 0.	R

VLANTEN[12:0] bits (VLAN Table Entry Number)

[Increment condition]

HW: An unsecure entry is learned successfully.

[Decrement condition]

HW: An unsecure entry is deleted successfully.

VLANTUEN[12:0] bits (VLAN Table Unsecure Entry Number)

[Increment condition]

HW: An entry is learned successfully.

[Decrement condition]

HW: An entry is deleted successfully.

30.3.1.6.39 FWVLANTS : VLAN Table Search Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4950

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

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

Bit	Symbol	Function	R/W
11:0	VLANVIDS[11:0]	VLAN VID Search Used for searching an entry in VLAN table.	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.6.40 FWVLANTSR0 : VLAN Table Search Result Register 0

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4954

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

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

Bit	Symbol	Function	R/W
0	—	This bit is read as 0.	R
1	VLANSNF	VLAN Search Not found Used for searching an entry in VLAN table. 0: VLAN found in VLAN Table. 1: VLAN not found in VLAN Table.	R
7:2	—	These bits are read as 0.	R
8	VLANSLS	VLAN Security Level Search Used for searching an entry in VLAN table.	R
9	—	These bits are read as 0.	R
10	VLANHLDS	VLAN Hardware Learning Disable Search Used for searching an entry in VLAN table.	R
30:11	—	These bits are read as 0.	R
31	VLANTS	VLAN Table Search	R

VLANTS bit (VLAN Table Search)

Used for searching an entry in VLAN table.

[Setting condition]

HW: Writing FWVLANTS register will set this bit.

[Clearing condition]

HW: This bit will be de-asserted when searching is completed.

30.3.1.6.41 FWVLANTSR1 : VLAN Table Search Result Register 1

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4958

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

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

Bit	Symbol	Function	R/W
2:0	VLANSLVS[2:0]	VLAN Source Lock Vector Search Used for searching an entry in VLAN table.	R
31:3	—	These bits are read as 0.	R

30.3.1.6.42 FWVLANTSR20 : VLAN Table Search Result Register 20

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4960 + 0x4 × i

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

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

Bit	Symbol	Function	R/W
5:0	VLANCSDS[5:0]	VLAN CPU Sub-Destination Search Used for searching an entry in VLAN table.	R
31:6	—	These bits are read as 0.	R

30.3.1.6.43 FWVLANTSR3 : VLAN Table Search Result Register 3

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4970

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	VLANCMES	VLANEMES	VLANIPUS	VLANIPVS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	VLANDVS[2:0]	VLAN Destination Vector Search Used for searching an entry in VLAN table.	R
15:3	—	These bits are read as 0.	R
18:16	VLANIPVS[2:0]	VLAN Internal Priority Value Search Used for searching an entry in VLAN table.	R
19	VLANIPUS	VLAN Internal Priority Update Search Used for searching an entry in VLAN table.	R
20	VLANEMES	VLAN Ethernet Mirroring Enable Search Used for searching an entry in VLAN table.	R
21	VLANCMES	VLAN CPU Mirroring Enable Search Used for searching an entry in VLAN table.	R
31:22	—	These bits are read as 0.	R

30.3.1.7 Port Based Forwarding Function Registers

30.3.1.7.1 FWPBFCi : Port i Port Based Forwarding Configuration Register (i = 0 to 2)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4A00 + 0x10 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	FAIFP	IP6PDE	IP4PDM	IP4PDE	PBSL	PBCME	PBEME	PBIPU	PBIPV[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	PBDV[2:0]	Port Based Destination Vector Destination vector for port i port-based forwarding.	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W
18:16	PBIPV[2:0]	Port Based Internal Priority Value Internal priority value for port i port-based forwarding.	R/W

Bit	Symbol	Function	R/W
19	PBIPU	Port Based Internal Priority Update Enables frame priority update to internal priority value (FWPBFCi.PBIPV) for port i.	R/W
20	PBEME	Port Based Ethernet Mirroring Enabled Enables ethernet mirroring for port i.	R/W
21	PBCME	Port Based CPU Mirroring Enabled Enables CPU mirroring for port i.	R/W
22	PBSL	Port Based Security Level Set descriptor security level for port i.	R/W
23	IP4PDE	IPv4 Priority Decode Enable 0: For port i, Input frame priority is taken from VLAN TAGs for port i. 1: For port i, If an IPv4 frame is detected, input frame priority is taken from precedence ToS. Else Input frame priority is taken from VLAN TAGs.	R/W
24	IP4PDM	IPv4 Priority Decode Mode 0: For port i, Input frame priority is taken from precedent field in ToS. 1: For port i, Input frame priority is decoded from DSCP field in ToS.	R/W
25	IP6PDE	IPv6 Priority Decode Enable 0: For port i, Input frame priority is taken from VLAN TAGs. 1: For port i, If an IPv6 frame is detected, input frame priority is decoded from DSCP field in ToS. Else Input frame priority is taken from VLAN TAGs.	R/W
26	FAIFP	Force All Input Frame Priority Enable 0: For port i, Input frame priority is given by frame decoding. 1: For port i, Input frame priority is forced to FWPBFCi.PBIPV.	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.7.2 FWPBFCSDCi0 : Port i Port Based Forwarding CSD Configuration Register 0 (i = 0 to 2)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4A04 + 0x10 × i

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

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

Bit	Symbol	Function	R/W
5:0	PBCSD[5:0]	Port Based CPU Sub Destination CPU Sub Destination from GWCAj [GWCA] for port i port-based forwarding.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.8 Layer 2/Layer 3 Update Function Registers

30.3.1.8.1 FWL23URL0 : Layer 2/Layer 3 Update Rule Learn Register 0

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4E00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	L23URPVL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	L23URNL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	L23URNL[7:0]	Layer 2/Layer 3 Update Routing Number Learn Used to learn a Layer 2/Layer 3 update rule	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
18:16	L23URPVL[2:0]	Layer 2/Layer 3 Update Routing Port Valid Learn Used to learn a Layer 2/Layer 3 update rule	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.8.2 FWL23URL1 : Layer 2/Layer 3 Update Rule Learn Register 1

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4E04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	L23URTUL[1:0]		L23USDEIUL	L23USPCPUL	L23USVIDUL	L23UCDEIUL	L23UCPCPUL	L23UCVIDUL	L23UMSAUL	L23UMDAUL	L23UTTLUL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	L23UMDALP0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	L23UMDALP0[15:0]	Layer 2/Layer 3 Update MAC Destination Address Learn Part 0 Used to learn a Layer 2/Layer 3 update rule	R/W
16	L23UTTLUL	Layer 2/Layer 3 Update Time To Live Update Learn Used to learn a Layer 2/Layer 3 update rule	R/W
17	L23UMDAUL	Layer 2/Layer 3 Update MAC Destination Address Update Learn Used to learn a Layer 2/Layer 3 update rule	R/W
18	L23UMSAUL	Layer 2/Layer 3 Update MAC Source Address Update Learn Used to learn a Layer 2/Layer 3 update rule	R/W
19	L23UCVIDUL	Layer 2/Layer 3 Update C-TAG VID Update Learn Used to learn a Layer 2/Layer 3 update rule	R/W
20	L23UCPCPUL	Layer 2/Layer 3 Update C-TAG PCP Update Learn Used to learn a Layer 2/Layer 3 update rule	R/W

Bit	Symbol	Function	R/W
21	L23UCDEIUL	Layer 2/Layer 3 Update C-TAG DEI Update Learn Used to learn a Layer 2/Layer 3 update rule	R/W
22	L23USVIDUL	Layer 2/Layer 3 Update S-TAG VID Update Learn Used to learn a Layer 2/Layer 3 update rule	R/W
23	L23USPCPUL	Layer 2/Layer 3 Update S-TAG PCP Update Learn Used to learn a Layer 2/Layer 3 update rule	R/W
24	L23USDEIUL	Layer 2/Layer 3 Update S-TAG DEI Update Learn Used to learn a Layer 2/Layer 3 update rule	R/W
26:25	L23URTUL[1:0]	Layer 2/Layer 3 Update R-TAG Update Learn Used to learn a Layer 2/Layer 3 update rule	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.8.3 FWL23URL2 : Layer 2/Layer 3 Update Rule Learn Register 2

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4E08

Bit position:	31 … 0
Bit field:	L23UMDALP1[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	L23UMDALP1[31:0]	Layer 2/Layer 3 Update MAC Destination Address Learn Part 1 Used to learn a Layer 2/Layer 3 update rule	R/W

30.3.1.8.4 FWL23URL3 : Layer 2/Layer 3 Update Rule Learn Register 3

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4E0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	L23USDEIL	L23USPCPL[2:0]			L23USVIDL[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	L23UCDEIL	L23UCPCPL[2:0]			L23UCVIDL[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	L23UCVIDL[11:0]	Layer 2/Layer 3 Update C-TAG VID Learn Used to learn a Layer 2/Layer 3 update rule	R/W
14:12	L23UCPCPL[2:0]	Layer 2/Layer 3 Update C-TAG PCP Learn Used to learn a Layer 2/Layer 3 update rule	R/W
15	L23UCDEIL	Layer 2/Layer 3 Update C-TAG DEI Learn Used to learn a Layer 2/Layer 3 update rule	R/W
27:16	L23USVIDL[11:0]	Layer 2/Layer 3 Update S-TAG VID Learn Used to learn a Layer 2/Layer 3 update rule	R/W
30:28	L23USPCPL[2:0]	Layer 2/Layer 3 Update S-TAG PCP Learn Used to learn a Layer 2/Layer 3 update rule	R/W

Bit	Symbol	Function	R/W
31	L23USDEIL	Layer 2/Layer 3 Update S-TAG DEI Learn Used to learn a Layer 2/Layer 3 update rule	R/W

30.3.1.8.5 FWL23URLR : Layer 2/Layer 3 Update Rule Learn Result Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4E10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	L23URL	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	L23ULF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	L23ULF	Layer 2/Layer 3 Update Learn Fail 0: Entry learning did not fail because the Layer 2/Layer 3 table is ready. 1: Entry learning failed because the Layer 2/Layer 3 table is not ready.	R
30:1	—	These bits are read as 0.	R
31	L23URL	Layer 2/Layer 3 Update Rule Learn	R

L23URL bit (Layer 2/Layer 3 Update Rule Learn)

[Setting condition]

HW: Writing FWL23URL3 register will set this bit.

[Clearing condition]

HW: This bit will be de-asserted when learning is completed.

30.3.1.8.6 FWL23UTIM : Layer 2/Layer 3 Update Table Initialization Monitoring Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4E20

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	L23UTR	L23UTIOG
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	L23UTIOG	Layer 2/Layer 3 Update Table Initialization Ongoing	R/W*1
1	L23UTR	Layer 2/Layer 3 Update Table Ready	R
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

L23UTIOG bit (Layer 2/Layer 3 Update Table Initialization Ongoing)

[Setting condition]

SW: By writing 1 to this bit. It starts Layer 2/Layer 3 Update Table initialization.

[Clearing condition]

HW: This bit is cleared when Layer 2/Layer 3 Update Table initialization is finished.

L23UTR bit (Layer 2/Layer 3 Update Table Ready)

[Setting condition]

- HW: When FWL23UTIM.L23UTIOG is getting cleared.
- HW: This bit is set at clk_period [ns] × 256 time from L23 update table initialization starting.

[Clearing condition]

SW: By writing 1 to FWL23UTIM.L23UTIOG.

30.3.1.8.7 FWL23URR : Layer 2/Layer 3 Update Rule Read Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4E30

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	L23RNR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	L23RNR[7:0]	Layer 2/Layer 3 Routing Number Read Used to read a Layer 2/Layer 3 update rule	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.8.8 FWL23URRR0 : Layer 2/Layer 3 Update Rule Read Result Register 0

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4E34

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	L23URR	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	L23URPVR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	L23URPVR[2:0]	Layer 2/Layer 3 Update Routing Port Valid Read Used to read a Layer 2/Layer 3 update rule	R

Bit	Symbol	Function	R/W
30:3	—	These bits are read as 0.	R
31	L23URR	Layer 2/Layer 3 Update Rule Read	R

L23URR bit (Layer 2/Layer 3 Update Rule Read)

[Setting condition]

HW: Writing FWL23URR register will set this bit.

[Clearing condition]

HW: This bit will be de-asserted when reading is completed.

30.3.1.8.9 FWL23URRR1 : Layer 2/Layer 3 Update Rule Read Result Register 1

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4E38

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	L23URTUR[1:0]		L23USDEIUR	L23USPCPUR	L23USVIDUR	L23UCDEIUR	L23UCPCPUR	L23UCVIDUR	L23UMSAUR	L23UMDAUR	L23UTTLUR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	L23UMDARP0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	L23UMDARP0[15:0]	Layer 2/Layer 3 MAC Destination Address Read Part 0 Used to read a Layer 2/Layer 3 update rule	R
16	L23UTTLUR	Layer 2/Layer 3 Time To Live Update Read Used to read a Layer 2/Layer 3 update rule	R
17	L23UMDAUR	Layer 2/Layer 3 MAC Destination Address Update Read Used to read a Layer 2/Layer 3 update rule	R
18	L23UMSAUR	Layer 2/Layer 3 MAC Source Address Update Read Used to read a Layer 2/Layer 3 update rule	R
19	L23UCVIDUR	Layer 2/Layer 3 C-TAG VID Update Read Used to read a Layer 2/Layer 3 update rule	R
20	L23UCPCPUR	Layer 2/Layer 3 C-TAG PCP Update Read Used to read a Layer 2/Layer 3 update rule	R
21	L23UCDEIUR	Layer 2/Layer 3 C-TAG DEI Update Read Used to read a Layer 2/Layer 3 update rule	R
22	L23USVIDUR	Layer 2/Layer 3 S-TAG VID Update Read Used to read a Layer 2/Layer 3 update rule	R
23	L23USPCPUR	Layer 2/Layer 3 S-TAG PCP Update Read Used to read a Layer 2/Layer 3 update rule	R
24	L23USDEIUR	Layer 2/Layer 3 S-TAG DEI Update Read Used to read a Layer 2/Layer 3 update rule	R
26:25	L23URTUR[1:0]	Layer 2/Layer 3 R-TAG Update Read Used to read a Layer 2/Layer 3 update rule	R
31:27	—	These bits are read as 0.	R

30.3.1.8.10 FWL23URRR2 : Layer 2/Layer 3 Update Rule Read Result Register 2

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4E3C

Bit position:	31–0
Bit field:	L23UMDARP1[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	L23UMDARP1[31:0]	Layer 2/Layer 3 Update MAC Destination Address Read Part 1 Used to read a Layer 2/Layer 3 update rule	R

30.3.1.8.11 FWL23URRR3 : Layer 2/Layer 3 Update Rule Read Result Register 3

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4E40

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	L23USDEIR	L23USPCPR[2:0]			L23USVIDR[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	L23UCDEIR	L23UCPCPR[2:0]			L23UCVIDR[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	L23UCVIDR[11:0]	Layer 2/Layer 3 Update MAC C-TAG VID Read Used to read a Layer 2/Layer 3 update rule	R
14:12	L23UCPCPR[2:0]	Layer 2/Layer 3 Update MAC C-TAG PCP Read Used to read a Layer 2/Layer 3 update rule	R
15	L23UCDEIR	Layer 2/Layer 3 Update MAC C-TAG DEI Read Used to read a Layer 2/Layer 3 update rule	R
27:16	L23USVIDR[11:0]	Layer 2/Layer 3 Update MAC S-TAG VID Read Used to read a Layer 2/Layer 3 update rule	R
30:28	L23USPCPR[2:0]	Layer 2/Layer 3 Update MAC S-TAG PCP Read Used to read a Layer 2/Layer 3 update rule	R
31	L23USDEIR	Layer 2/Layer 3 Update MAC S-TAG DEI Read Used to read a Layer 2/Layer 3 update rule	R

30.3.1.8.12 FWL23URMCi : Layer 2/Layer 3 Update Remapping Configuration Register i (i = 0 to 31)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x4F00 + 0x4 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	RME	—	—	—	—	RMNRN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
7:0	RMRN[7:0]	Remapping Rule Number Set the rule number that should be remapped for remapping rule i.	R/W
11:8	—	These bits are read as 0. The write value should be 0.	R/W
13:12	RMDPN[1:0]	Remapping Destination Port Number Set the port number for which rule number should be remapped for remapping rule i.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
23:16	RMNRN[7:0]	Remapping New Rule Number Set the rule number to which routing should be remapped for remapping rule i.	R/W
27:24	—	These bits are read as 0. The write value should be 0.	R/W
28	RME	Remapping Enable 0: Remapping rule i disabled 1: Remapping rule i enabled	R/W*1
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

RME bit (Remapping Enable)

[Setting condition]

SW: Writing 1 to this bit will set it.

30.3.1.9 PSFP (Per Stream Filtering and Policing) [802.1Qci]

30.3.1.9.1 FWPMFGCi : PSFP MSDU Filter Global Configuration Register i (i = 0 to 15)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x5000 + 0x4 × i

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

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

Bit	Symbol	Function	R/W
15:0	MSDUV[15:0]	MSDU Value HW: Maximum frame size for stream linked to MSDU filter i thanks to L3 table.	R/W
30:16	—	These bits are read as 0. The write value should be 0.	R/W
31	MFM	MSDU Filter Mode 0: Normal mode: Any frame linked to MSDU filter i thanks to L3 table received with LDESCR.TPL [GWCA] bigger than FWPMPCi.MSDUV will be filtered. 1: Throttle mode: Any frame linked to MSDU filter i thanks to L3 table received with LDESCR.TPL [GWCA] bigger than FWPMPCi.MSDUV will be filtered. Any frame linked to MSDU filter i thanks to L3 table received when FWEIS2.PMFSi is set will be filtered.	R/W

30.3.1.9.2 FWPMTRFCi : PSFP Meter i Filter Configuration Register (i = 0 to 31)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x5600 + 0x20 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MTRCM15 to MTRCM0															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	MTRFE	Meter Filter Enable 0: Meter i disabled (Pass through all frames.) 1: Meter i enabled (Reject target stream frames by meter filter.)	R/W
2:1	MTRFM[1:0]	Meter Filter Mode 0 0: Normal mode: Any frame linked to Meter filter i thanks to L3 table received when not enough token is available will be red. 0 1: Throttle mode: Any frame linked to Meter filter i thanks to L3 table received when not enough token is available or when FWEIS5.PMRFSi is set will be red. 1 0: ATS mode: Any frame linked to Meter filter i thanks to L3 table received when not enough token is available will be stored in ATS RAM until enough tokens are available. When enough tokens are available, frame will be forwarded as yellow or green. 1 1: ATS throttle mode: Any frame linked to Meter filter i thanks to L3 table received when not enough token is available will be stored in ATS RAM until enough tokens are available. When enough tokens are available, frame will be forwarded as yellow or green if FWEIS5.PMRFSi is not set and will be red if FWEIS5.PMRFSi is set.	R/W
3	MTRFRFD	Meter Filter Red Frame Drop 0: Red frames are not dropped by meter i. 1: Red frames are dropped by meter i.	R/W
4	MTRCF	Meter Coupling Flag*1 0: When green bucket is full, CIR tokens are lost. 1: When green bucket is full, CIR tokens are added to the yellow bucket.	R/W
15:5	—	These bits are read as 0. The write value should be 0.	R/W
31:16	MTRCM15 to MTRCM0	Meter Color Mode n (n = 0 to 15)*2	R/W

Note 1. HW: This bit is only valid if i = 0 to 7.
Note 2. HW: These bits are only valid if i = 0 to 7.
SW: in ATS mode (FWPMTRFCi.MTRFM set to 10b) or in ATS throttle mode (FWPMTRFCi.MTRFM set to 11b) Color mode is not available, so these bits should be set to All0.

MTRCMn bits (Meter Color Mode n (n = 0 to 15))

If bit j is set to 0: Incoming frame with {PCP,DEI} == j is coming as green (Green or yellow bucket can be used by the frame).

If bit j is set to 1: Incoming frame with {PCP,DEI} == j is coming as Yellow (Only yellow bucket can be used by the frame).

30.3.1.9.3 FWPMTRCBSCi : PSFP Meter i CBS Configuration Register (i = 0 to 31)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x5604 + 0x20 × i

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

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

Bit	Symbol	Function	R/W
17:0	CBS[17:0]	CBS Maximum number of tokens (in bytes) contained in meter i green bucket.	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.9.4 FWPMTRCIRCi : PSFP Meter i CIR Configuration Register (i = 0 to 31)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x5608 + 0x20 × i

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

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

Bit	Symbol	Function	R/W
19:0	CIR[19:0]	CIR	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

CIR[19:0] bits (CIR)

Used to increment green token tokens when meter i green bucket is not full.

CIR[19:16] is the credit increment in byte per clock and CIR[15:0] is the credit increment in sub-byte per clock.

Note: SW: In ATS mode, setting these bits to 0 or a value close to 0 would set corresponding queue to a very low throughput or could stick the corresponding queue and lead to a queue overflow.

30.3.1.9.5 FWPMTREBSCi : PSFP Meter i EBS Configuration Register (i = 0 to 7)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x560C + 0x20 × i

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

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

Bit	Symbol	Function	R/W
17:0	EBS[17:0]	EBS Maximum number of tokens (in bytes) contained in meter i yellow bucket.	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.9.6 FWPMTREIRCi : PSFP Meter i EIR Configuration Register (i = 0 to 7)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x5610 + 0x20 × i

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

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

Bit	Symbol	Function	R/W
19:0	EIR[19:0]	EIR	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

EIR[19:0] bits (EIR)

Used to increment green token tokens when meter i yellow bucket is not full.

EIR[19:16] is the credit increment in byte per clock and EIR[15:0] is the credit increment in sub-byte per clock.

Note: SW: In ATS mode, setting these bits to 0 or a value close to 0 would set the corresponding queue to a very low throughput or could stick the corresponding queue and lead to a queue overflow.

30.3.1.9.7 FWPMTRFMi : PSFP Meter i Filter Monitoring Register (i = 0 to 31)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x5614 + 0x20 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	MTRARDNMN[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
4:0	MTRARDN[4:0]	Meter ATS RAM Descriptor Number	R
15:5	—	These bits are read as 0.	R
20:16	MTRARDNMN[4:0]	Meter ATS RAM Descriptor Number Maximum Number	R
31:21	—	These bits are read as 0.	R

MTRARDN[4:0] bits (Meter ATS RAM Descriptor Number)

Shown the number of descriptors in the ATS RAM for meter number i.

[Increment condition]

A descriptor is received for meter i but there is already a descriptor in the ATS RAM for meter i or there is not even token available for the frame.

30.3.1.10 FRER (Frame Replication and Elimination for Reliability) [802.1CB]

30.3.1.10.1 FWFTL0 : FRER Table Learn Register 0

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	FSRPL[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
6:0	FEAL[6:0]	FRER Entry Address Learn Used to set an FRER entry in FRER RAM.	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W
22:16	FSRPL[6:0]	FRER Sequence Recovery Pointer Learn Used to set an FRER entry in FRER RAM.	R/W
31:23	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.10.2 FWFTL1 : FRER Table Learn Register 1

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	FSRRTL[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
3:0	FSHLL[3:0]	FRER Sequence History Length Learn*1 Used to set an FRER entry in FRER RAM.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	FTNSL	FRER Take No Sequence Learn Used to set an FRER entry in FRER RAM.	R/W
9	FSRPVL	FRER Sequence Recovery Pointer Valid Learn Used to set an FRER entry in FRER RAM.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	FSRRTL[9:0]	FRER Sequence Recovery Remaining Ticks Learn Used to set an FRER entry in FRER RAM.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: These bits should not be set to a value greater than 14.

30.3.1.10.3 FWFTLR : FRER Table Learn Result Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6008

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

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

Bit	Symbol	Function	R/W
0	FLF	FRER Learn Fail 0: Entry learning did not fail because the FRER RAM is ready. 1: Entry learning failed because the FRER RAM is not ready.	R
30:1	—	These bits are read as 0.	R
31	FTL	FRER Table Learn	R

FTL bit (FRER Table Learn)

[Setting condition]

HW: Writing FWFTL1 register will set this bit.

[Clearing condition]

HW: This bit will be deasserted when learning is completed.

30.3.1.10.4 FWFTOC : FRER Timeout Configuration Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6010

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

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

Bit	Symbol	Function	R/W
15:0	TOT[15:0]	Timeout Time (ms) When Timeout check is enabled (FWFTOC.TOCE set to 1), timeout check will happen every FWFTOC.TOT milliseconds.	R/W
16	TOCE	Timeout Check Enable 0: Timeout check is disabled. 1: Timeout check is enabled.	R/W
17	TOOG	Timeout Ongoing 0: Timeout check is not ongoing. 1: Timeout check is ongoing.	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.10.5 FWFTOPC : FRER Timeout Prescaler Configuration Register 0

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6014

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

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

Bit	Symbol	Function	R/W
9:0	USP[9:0]	Microsecond Prescaler*1 Used to creates an internal clock for aging at 1 MHz to derive the Timeout 1 kHz clock.	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: These bits should be set to the number of clk clocks contained in 1 μs.

30.3.1.10.6 FWFTIM : FRER Table Initialization Monitoring Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6020

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

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

Bit	Symbol	Function	R/W
0	FTIOG	FRER Table Initialization Ongoing	R/W*1
1	FTR	FRER Table Ready	R
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

FTIOG bit (FRER Table Initialization Ongoing)

[Setting condition]

SW: By writing 1 to this bit. It starts FRER Table initialization.

[Clearing condition]

HW: This bit is cleared when FRER Table initialization is finished.

FTR bit (FRER Table Ready)

[Setting condition]

- HW: When FWFTIM.FTIOG is getting cleared.
- HW: This bit is set at clk_period [ns] × 128 time from buffer pool initialization starting.

[Clearing condition]

SW: By writing 1 to FWFTIM.FTIOG.

30.3.1.10.7 FWFTR : FRER Table Read Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6030

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

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

Bit	Symbol	Function	R/W
6:0	FEAR[6:0]	FRER Entry Address Read Used to read an FRER entry in FRER RAM.	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.10.8 FWFTRR0 : FRER Table Read Result Register 0

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6034

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	FTR	—	—	—	—	—	FSRRTR[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
3:0	FSHLR[3:0]	FRER Sequence History Length Read	R
7:4	—	These bits are read as 0.	R
8	FTNSR	FRER Take No Sequence Read	R
9	FSRPVR	FRER Sequence Recovery Pointer Valid Read	R
15:10	—	These bits are read as 0.	R
25:16	FSRRTR[9:0]	FRER Set Recovery Remaining Ticks Read	R
30:26	—	These bits are read as 0.	R
31	FTR	FRER Table Read	R

FTR bit (FRER Table Read)

[Setting condition]

HW: Writing FWFTR register will set this bit.

[Clearing condition]

HW: This bit will be de-asserted when reading is completed.

30.3.1.10.9 FWFTRR1 : FRER Table Read Result Register 1

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6038

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	FSRPR[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
14:0	FSHR[14:0]	FRER Sequence History Read	R
15	—	This bit is read as 0.	R
22:16	FSRPR[6:0]	FRER Sequence Recovery Pointer Read	R
31:23	—	These bits are read as 0.	R

30.3.1.10.10 FWFTRR2 : FRER Table Read Result Register 2

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x603C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	FRRTR[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
15:0	FRSNR[15:0]	FRER Recovery Sequence Number Read	R
25:16	FRRTR[9:0]	FRER Recovery Remaining Ticks Read	R
31:26	—	These bits are read as 0.	R

30.3.1.10.11 FWSEQNGCi : Sequence Number Generation i Configuration Register (i = 0 to 31)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6100 + 0x8 × i

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

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

Bit	Symbol	Function	R/W
7:0	SEQNGRN[7:0]	SEQuence Number Generation Routing Number Specifies the routing number for which sequence number will be generated.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
16	SEQNGE	SEQuence Number Generation Enable 0: Sequence number generation disabled for routing number FWSEQNGCi.SEQNGRN 1: Sequence number generation enabled for routing number FWSEQNGCi.SEQNGRN	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

30.3.1.10.12 FWSEQNGMi : Sequence Number Generation i Monitoring Register (i = 0 to 31)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6104 + 0x8 × i

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

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

Bit	Symbol	Function	R/W
15:0	SEQN[15:0]	SEQuence Number	R
31:16	—	These bits are read as 0.	R

SEQN[15:0] bits (SEQuence Number)

Displays the next sequence number to be appended to a descriptor for sequence number generation rule i.

[Increment condition]

HW: When FWSEQNGCi.SEQNGE is set and a descriptor is forwarded with FDESCR.RV set and FDESCR.RN set to FWSEQNGCi.SEQNGRN.

[Clearing condition]

SW: Writing 1 to FWSEQNRC.SEQNR will clear these bits.

30.3.1.10.13 FWSEQNRC : Sequence Number Reset Configuration Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6200

Bit position:	31 … 0
Bit field:	SEQNR31 to SEQNR0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	SEQNR31 to SEQNR0	Sequence Number Generation Reset n (n = 0 to 31) 0: No effect 1: Bit i of this register resets the sequence generator number i counter to 0	W

30.3.2 MFWD Counter Registers

This section describes MFWD counter registers.

30.3.2.1 FWCTFDCNi : Port i Cut-Through Forwarded Descriptor Counter Register (i = 0, 1)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6300 + 0x20 × i

Bit position:	31 … 0
Bit field:	CTFDN[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	CTFDN[31:0]	Cut-Through Forwarded Descriptor Number	R

CTFDN[31:0] bits (Cut-Through Forwarded Descriptor Number)

These bits count the number of descriptors forwarded by Cut-Through forwarding for port i.

[Clearing condition]

SW: Reading these bits clears it.

[Increment condition]

HW: Incremented by 1 when a frame coming from port i is forwarded by Cut-Through Forwarding in store and forward mode and if these bits have a value other than 0xFFFFFFFF.

Note: HW: These bits do not take in account the cut-through path of cut-through forwarding but only the store and forward path so, any frame forwarded in store and forward with cut-through (for at least one value of j, if FWCTTCi0.CTDV[j] is set, FWCTTCi0.CTDFMj is also set) with be counted here.

30.3.2.2 FWDDFDCN2 : Port 2 Direct Descriptor Forwarded Descriptor Counter Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6340

Bit position:	31 … 0
Bit field:	DDFDN[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	DDFDN[31:0]	Direct Descriptor Forwarded Descriptor Number	R

DDFDN[31:0] bits (Direct Descriptor Forwarded Descriptor Number)

These bits count the number of descriptors forwarded by Direct Descriptor forwarding for port 2.

[Clearing condition]

SW: Reading these bits clears it.

[Increment condition]

HW: Incremented by 1 when a frame coming from port 2 is forwarded by Direct Descriptor Forwarding and if these bits have a value other than 0xFFFFFFFF.

30.3.2.3 FWLTHFDCNi : Port i Layer 3 Forwarded Descriptor Counter Register (i = 0 to 2)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6304 + 0x20 × i

Bit position:	31–0
Bit field:	LTHFDN[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	LTHFDN[31:0]	Layer 3 Forwarded Descriptor Number	R

LTHFDN[31:0] bits (Layer 3 Forwarded Descriptor Number)

These bits count the number of descriptors forwarded by Layer 3 forwarding for port i.

[Clearing condition]

SW: Reading these bits clears it.

[Increment condition]

HW: Incremented by 1 when a frame coming from port i is forwarded by Layer 3 Forwarding and if these bits have a value different from 0xFFFFFFFF.

Note: HW: These bits count the frame selected by Layer 3 forwarding mechanism before PSFP/ATS and FRER filtering functions. Frame rejected by these filtering functions are still taken in account here.

30.3.2.4 FWLTWFDCNi : Port i Layer 2 Forwarded Descriptor Counter Register (i = 0 to 2)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x630C + 0x20 × i

Bit position:	31–0
Bit field:	LTWFDN[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	LTWFDN[31:0]	Layer 2 Forwarded Descriptor Number	R

LTWFDN[31:0] bits (Layer 2 Forwarded Descriptor Number)

These bits count the number of descriptors forwarded by Layer 2 forwarding for port i.

[Clearing condition]

SW: Reading these bits clears it.

[Increment condition]

HW: Incremented by 1 when a frame coming from port i is forwarded by Layer 2 Forwarding and if these bits have a value different from 0xFFFFFFFF.

30.3.2.5 FWPBFDCNi : Port i Port Based Forwarded Descriptor Counter Register (i = 0 to 2)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6310 + 0x20 × i

Bit position:	31 … 0
Bit field:	PBFDN[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	PBFDN[31:0]	Port Based Forwarded Descriptor Number	R

PBFDN[31:0] bits (Port Based Forwarded Descriptor Number)

These bits count the number of descriptors forwarded by Port Based forwarding for port i.

[Clearing condition]

SW: Reading these bits clears it.

[Increment condition]

HW: Incremented by 1 when a frame coming from port i is forwarded by Port Based Forwarding and if these bits have a value different from 0xFFFFFFFF.

30.3.2.6 FWMHLCNi : Port i MAC Hardware Learn Counter Register (i = 0 to 2)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6314 + 0x20 × i

Bit position:	31 … 0
Bit field:	MHLN[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	MHLN[31:0]	MAC Hardware Learn Number	R

MHLN[31:0] bits (MAC Hardware Learn Number)

These bits count the number of MAC source addresses that has been learnt or migrated for port i.

[Clearing condition]

SW: Reading these bits clears it.

[Increment condition]

HW: Incremented by 1 when a frame is selected for source MAC address HW learning or HW migrated (section 30.5.1.5.2. CPU Learning Path) and learning rate is not too high and if these bits have a value different from {32 {1}}.

This counter is incremented regardless of the MAC learn result Table 30.45. (It counts even if it fails)

This counter is incremented only learning or migrated HW request. So, the counter increments twice even for consecutive requests with the same content. Since the processing is the same, it looks like one time.

30.3.2.7 FWWMRDCNi : Port i Watermark Rejected Descriptor Counter Register (i = 0 to 2)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6504 + 0x20 × i

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

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

Bit	Symbol	Function	R/W
15:0	WMRDN[15:0]	Watermark rejected Descriptor Number	R
31:16	—	These bits are read as 0.	R

WMRDN[15:0] bits (Watermark rejected Descriptor Number)

These bits count the number of descriptors rejected by Watermark for port i.

[Clearing condition]

SW: Reading these bits clears it.

[Increment condition]

HW: Incremented by 1 when a frame coming from port i is rejected by Watermark and if these bits have a value different from 0xFFFFFFFF.

30.3.2.8 FWCTRDCNi : Port i Cut-Through Rejected Descriptor Counter Register (i = 0, 1)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6508 + 0x20 × i

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

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

Bit	Symbol	Function	R/W
15:0	CTRDN[15:0]	Cut-through rejected Descriptor Number	R
31:16	—	These bits are read as 0.	R

CTRDN[15:0] bits (Cut-through rejected Descriptor Number)

These bits count the number of descriptors rejected by Cut-through forwarding for port i.

[Clearing condition]

SW: Reading these bits clears it.

[Increment condition]

HW: Incremented by 1 when a frame coming from port i is forwarded by Cut-Through Forwarding but not in store and forward mode and if these bits have a value different from 0xFFFFFFFF.

Note: HW: These bits do not take in account the cut-through path of cut-through forwarding but only the store and forward path so, any frame not forwarded in store and forward with cut-through (for any j, if FWCTTCi0.CTDV[j] is set, FWCTTCi0.CTDFMj is not set) with be counted here.

30.3.2.9 FWDDRDCN2 : Port 2 Direct Descriptor Rejected Descriptor Counter Register

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6548

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

Bit	Symbol	Function	R/W
15:0	DDRDN[15:0]	Direct Descriptor rejected Descriptor Number	R
31:16	—	These bits are read as 0.	R

DDRDN[15:0] bits (Direct Descriptor rejected Descriptor Number)

These bits count the number of descriptors rejected by Direct Descriptor forwarding for port 2.

[Clearing condition]

SW: Reading these bits clears it.

[Increment condition]

HW: Incremented by 1 when a frame coming from port 2 is rejected by Direct Descriptor forwarding and if these bits have a value different from 0xFFFFFFFF.

30.3.2.10 FWLTHRDCNi : Port i Layer 3 Rejected Descriptor Counter Register (i = 0 to 2)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x650C + 0x20 × i

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

Bit	Symbol	Function	R/W
15:0	LTHRDN[15:0]	Layer 3 rejected Descriptor Number	R
31:16	—	These bits are read as 0.	R

LTHRDN[15:0] bits (Layer 3 rejected Descriptor Number)

These bits count the number of descriptors rejected by Layer 3 forwarding for port i.

[Clearing condition]

SW: Reading these bits clears it.

[Increment condition]

HW: Incremented by 1 when a frame coming from port i is rejected by Layer 3 forwarding and if these bits have a value other than 0xFFFF.

30.3.2.11 FWLTWRDCNi : Port i Layer 2 Rejected Descriptor Counter Register (i = 0 to 2)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6514 + 0x20 × i

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

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

Bit	Symbol	Function	R/W
15:0	LTWRDN[15:0]	Layer 2 rejected Descriptor Number	R
31:16	—	These bits are read as 0.	R

LTWRDN[15:0] bits (Layer 2 rejected Descriptor Number)

These bits count the number of descriptors rejected by Layer 2 forwarding for port i.

[Clearing condition]

SW: Reading these bits clears it.

[Increment condition]

HW: Incremented by 1 when a frame coming from port i is rejected by Layer 2 forwarding and if these bits have a value other than 0xFFFF.

30.3.2.12 FWPBRDCNi : Port i Port Based Rejected Descriptor Counter Register (i = 0 to 2)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6518 + 0x20 × i

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

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

Bit	Symbol	Function	R/W
15:0	PBRDN[15:0]	Port Based rejected Descriptor Number	R
31:16	—	These bits are read as 0.	R

PBRDN[15:0] bits (Port Based rejected Descriptor Number)

These bits count the number of descriptors rejected by Port Based forwarding for port i.

[Clearing condition]

SW: Reading these bits clears it.

[Increment condition]

HW: Incremented by 1 when a frame coming from port i is rejected by Port Based forwarding and if these bits have a value other than 0xFFFF.

30.3.2.13 FWPMFDCNi : PSFP MSDU i Filtered Descriptor Counter Register (i = 0 to 15)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6700 + 0x4 × i

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

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

Bit	Symbol	Function	R/W
15:0	PMFDN[15:0]	PSFP MSDU Filtered Descriptor Number	R
31:16	—	These bits are read as 0.	R

PMFDN[15:0] bits (PSFP MSDU Filtered Descriptor Number)

These bits count the number of descriptors rejected by PSFP MSDU Filter i.

[Clearing condition]

SW: Reading these bits clears it.

[Increment condition]

HW: Incremented by 1 when a frame is rejected by PSFP MSDU Filter i and if these bits have a value different from 0xFFFF.

30.3.2.14 FWPMGDCNi : PSFP Meter i Green Descriptor Counter Register (i = 0 to 31)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6800 + 0x10 × i

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

Bit	Symbol	Function	R/W
15:0	PMGDN[15:0]	PSFP Meter Green Descriptor Number	R
31:16	—	These bits are read as 0.	R

PMGDN[15:0] bits (PSFP Meter Green Descriptor Number)

These bits count the number of descriptors with the color green given by PSFP meter Filter i.

[Clearing condition]

SW: Reading these bits clears it.

[Increment condition]

HW: Incremented by 1 when a frame has green color associated by PSFP meter Filter i and if these bits have a value different from 0xFFFF.

30.3.2.15 FWPMYDCNi : PSFP Meter i Yellow Descriptor Counter Register (i = 0 to 7)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6804 + 0x10 × i

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

Bit	Symbol	Function	R/W
15:0	PMYDN[15:0]	PSFP Meter Yellow Descriptor Number	R
31:16	—	These bits are read as 0.	R

PMYDN[15:0] bits (PSFP Meter Yellow Descriptor Number)

These bits count the number of descriptors with the color Yellow given by PSFP meter Filter i.

[Clearing condition]

SW: Reading these bits clears it.

[Increment condition]

HW: Incremented by 1 when a frame has yellow color associated by PSFP meter Filter i and if these bits have a value different from 0xFFFF.

30.3.2.16 FWPMRDCNi : PSFP Meter i Red Descriptor Counter Register (i = 0 to 31)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6808 + 0x10 × i

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

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

Bit	Symbol	Function	R/W
15:0	PMRDN[15:0]	PSFP Meter Red Descriptor Number	R
31:16	—	These bits are read as 0.	R

PMRDN[15:0] bits (PSFP Meter Red Descriptor Number)

These bits count the number of descriptors with the color Red given by PSFP meter Filter i. (All and only frames rejected by meter filter are accounted)

[Clearing condition]

SW: Reading these bits clears it.

[Increment condition]

- HW: Incremented by 1 when a frame has Red color associated and is rejected by PSFP meter Filter i and if these bits have a value different from 0xFFFF.
- HW: Incremented by 1 when a frame is rejected by PSFP meter Filter i because ATS RAM corresponding queue is full and if these bits have a value different from 0xFFFF.
- HW: Incremented by 1 when a frame is rejected by PSFP meter Filter i because corresponding meter is disabled and if these bits have a value different from 0xFFFF.

30.3.2.17 FWFRPPCNi : FRER i Passed Packet Counter Register (i = 0 to 127)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6A00 + 0x8 × i

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

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

Bit	Symbol	Function	R/W
15:0	PPC[15:0]	Passed Packet Count	R

Bit	Symbol	Function	R/W
31:16	—	These bits are read as 0.	R

PPC[15:0] bits (Passed Packet Count)

These bits count the number of packets that passed FRER recovery rule number i.

[Clearing condition]

SW: Reading these bits clears it.

[Increment condition]

HW: Incremented by 1 when a frame passed FRER recovery rule number i and if these bits have a value different from 0xFFFF.

30.3.2.18 FWFRDPCNi : FRER i Discarded Packet Counter Register (i = 0 to 127)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x6A04 + 0x8 × i

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

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

Bit	Symbol	Function	R/W
15:0	DPC[15:0]	Discarded Packet Count	R
31:16	—	These bits are read as 0.	R

DPC[15:0] bits (Discarded Packet Count)

These bits count the number of packets that has been discarded by FRER recovery rule number i.

[Clearing condition]

SW: Reading these bits clears it.

[Increment condition]

HW: Incremented by 1 when a frame has been discarded by FRER recovery rule number i and if these bits have a value different from 0xFFFF.

30.3.3 MFWD Interrupt Registers

This section describes MFWD interrupt registers.

30.3.3.1 FWEIS0i : Port i Error Interrupt Status Register 0 (i = 0 to 2)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7900 + 0x10 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	DDNTFS	DDSES	—	DDES	WMIUFS	WMISFS	WMFFS	WMCFS	—	—	SMHMFS	SMHLFS	PBNTFS	LTWVUFS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	LTWDUFS	LTWSUFS	LTWNTFS	LTWVSPFS	LTWSSPFS	LTWDSPFS	—	—	—	—	—	—	LTHUFS	LTHNTFS	—	LTHSPFS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LTHSPFS	Layer 3 Source Port Filtering Status Flag	R/W*1
1	—	These bits are read as 0. The write value should be 0.	R/W
2	LTHNTFS	Layer 3 No Target Filtering Status Flag	R/W*1
3	LTHUFS	Layer 3 Unknown Filtering Status Flag	R/W*1
9:4	—	These bits are read as 0. The write value should be 0.	R/W
10	LTWDSPFS	Layer 2 Destination Source Port Filtering Status Flag	R/W*1
11	LTWSSPFS	Layer 2 Source Source Port Filtering Status Flag	R/W*1
12	LTWVSPFS	Layer 2 VLAN Source Port Filtering Status Flag	R/W*1
13	LTWNTFS	Layer 2 No Target Filtering Status Flag	R/W*1
14	LTWSUFS	Layer 2 Source Unknown Filtering Status Flag	R/W*1
15	LTWDUFS	Layer 2 Destination Unknown Filtering Status Flag	R/W*1
16	LTWVUFS	Layer 2 VLAN Unknown Filtering Status Flag	R/W*1
17	PBNTFS	Port Based No Target Filtering Status Flag	R/W*1
18	SMHLFS	Source MAC Hardware Learning Fail Status Flag	R/W*1
19	SMHMFS	Source MAC Hardware Migration Fail Status Flag	R/W*1
21:20	—	These bits are read as 0. The write value should be 0.	R/W
22	WMCFS	Watermark Critical Filtering Status Flag	R/W*1
23	WMFFS	Watermark Flush Filtering Status Flag	R/W*1
24	WMISFS	Watermark IPV Secure Filtering Status Flag	R/W*1
25	WMIUFS	Watermark IPV Unsecure Filtering Status Flag	R/W*1
26	DDES	Direct Descriptor Error Status Flag*2	R/W*1
27	—	This bit is read as 0. The write value should be 0.	R/W
28	DDSES	Direct Descriptor Security Error Status Flag*2	R/W*1
29	DDNTFS	Direct Descriptor No Target Filtering Status Flag*2	R/W*1
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.
Note 2. HW: This bit is only valid for i = 2, 3.

LTHSPFS bit (Layer 3 Source Port Filtering Status Flag)

[Setting condition]

- HW: When forwarding flow for a frame coming from port i goes until “Layer 3 routing secure error” block in Figure 30.85 and L3.SLV[i] is not set.
- HW: When forwarding flow for a frame coming from port i goes until “Layer 3 routing error” block in Figure 30.86 and L3.SLV[i] is not set.

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

LTHNTFS bit (Layer 3 No Target Filtering Status Flag)

[Setting condition]

- HW: When forwarding flow for a frame coming from port i goes until “Layer 3 routing secure error” block in Figure 30.85 and L3.DV & !FWPCi1.LTHFM is null.
- HW: When forwarding flow for a frame coming from port i goes until “Layer 3 routing error” block in Figure 30.86 and L3.DV & !FWPCi1.LTHFM is null.

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

LTHUFS bit (Layer 3 Unknown Filtering Status Flag)

[Setting condition]

- HW: When forwarding flow for a frame coming from port i goes until “Layer 3 routing unknown secure stream error” block in Figure 30.85 and FWPCi0.LTHRUSS is set.
- HW: When forwarding flow for a frame coming from port i goes until “Layer 3 routing error” block in Figure 30.86 and FWPCi0.LTHRUS is set.

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

LTWDSPFS bit (Layer 2 Destination Source Port Filtering Status Flag)

[Setting condition]

- HW: When forwarding flow for a frame coming from port i goes until “L2 forwarding secure error” block in Figure 30.95 and, FWPCi0.MACDSA is set, Destination MAC is set in the MAC table, MAC.SL is set and MAC.DSLV is not set for destination MAC.
- HW: When forwarding flow for a frame coming from port i goes until “L2 forwarding error” block in Figure 30.96 and, FWPCi0.MACDSA is set, Destination MAC is set in the MAC table and MAC.DSLV is not set for destination MAC.

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

LTWSSPFS bit (Layer 2 Source Source Port Filtering Status Flag)

[Setting condition]

- HW: When forwarding flow for a frame coming from port i goes until "L2 forwarding secure error" block in Figure 30.95 and, FWPCi0.MACSSA is set, Source MAC is set in the MAC table, MAC.SL is set and MAC.SSLV is not set for source MAC.
- HW: When forwarding flow for a frame coming from port i goes until "L2 forwarding error" block in Figure 30.96 and, FWPCi0.MACSSA is set, Source MAC is set in the MAC table and MAC.SSLV is not set for source MAC.

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

LTWVSPFS bit (Layer 2 VLAN Source Port Filtering Status Flag)

[Setting condition]

- HW: When forwarding flow for a frame coming from port i goes until "L2 forwarding secure error" block in Figure 30.95 and, FWPCi0.VLANSA is set, VLAN ID is valid, VLAN ID is set in the VLAN table, VLAN.SL is set and VLAN.SLV is not set for VLAN ID.
- HW: When forwarding flow for a frame coming from port i goes until "L2 forwarding error" block in Figure 30.96 and, FWPCi0.VLANSA is set, VLAN ID is valid, VLAN ID is set in the VLAN table and VLAN.SLV is not set for VLAN ID.

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

LTWNTFS bit (Layer 2 No Target Filtering Status Flag)

[Setting condition]

- HW: When forwarding flow for a frame coming from port i goes until "L2 forwarding secure error" block inFigure 30.95 and, FWPCi0.MACDSA is set, Destination MAC is set in the MAC table, MAC.SL is set, FWPCi0.VLANSA is set, VLAN ID is valid, VLAN ID is set in the VLAN table, VLAN.SL is set and MAC.DV & VLAN.DV & ! FWPCi2.LTWFM is null.
- HW: When forwarding flow for a frame coming from port i goes until "L2 forwarding secure error" block in Figure 30.95 and, FWPCi0.MACDSA is set, Destination MAC is set in the MAC table, MAC.SL is set, { FWPCi0.VLANSA is not set or VLAN ID is not valid or VLAN ID is not set in the VLAN table or VLAN.SL is not set} and MAC.DV & !FWPCi2.LTWFM is null.
- HW: When forwarding flow for a frame coming from port i goes until "L2 forwarding secure error" block in Figure 30.95 and, {FWPCi0.MACDSA is not set or Destination MAC is not set in the MAC table or MAC.SL is not set}, VLAN ID is valid, VLAN ID is set in the VLAN table, VLAN.SL is set and VLAN.DV & !FWPCi2.LTWFM is null.
- HW: When forwarding flow for a frame coming from port i goes until "L2 forwarding error" block in Figure 30.96 and, FWPCi0.MACDSA is set, Destination MAC is set in the MAC table, FWPCi0.VLANSA is set, VLAN ID is valid, VLAN ID is set in the VLAN table and MAC.DV & VLAN.DV & !FWPCi2.LTWFM is null.
- HW: When forwarding flow for a frame coming from port i goes until "L2 forwarding error" block in Figure 30.96 and, FWPCi0.MACDSA is set, Destination MAC is set in the MAC table, { FWPCi0.VLANSA is not set or VLAN ID is not valid or VLAN ID is not set in the VLAN table } and MAC.DV & !FWPCi2.LTWFM is null.
- HW: When forwarding flow for a frame coming from port i goes until "L2 forwarding error" block in Figure 30.96 and, {FWPCi0.MACDSA is not set or Destination MAC is not set in the MAC table}, VLAN ID is valid, VLAN ID is set in the VLAN table and VLAN.DV & !FWPCi2.LTWFM is null.

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

LTWSUFS bit (Layer 2 Source Unknown Filtering Status Flag)

[Setting condition]

- HW: When forwarding flow for a frame coming from port i goes until “L2 forwarding secure error” block in Figure 30.95 and, { FWPCi0.MACSSA is not set or Source MAC is not set in the MAC table or MAC.SL is not set for Source MAC} and FWPCi0.MACRUSSA is set.
- HW: When forwarding flow for a frame coming from port i goes until “L2 forwarding error” block in Figure 30.96 and, {FWPCi0.MACSSA is not set or Source MAC is not set in the MAC table} and FWPCi0.MACRUSA is set.

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

LTWDUFS bit (Layer 2 Destination Unknown Filtering Status Flag)

[Setting condition]

- HW: When forwarding flow for a frame coming from port i goes until “L2 forwarding secure error” block in Figure 30.95 and, {FWPCi0.MACDSA is not set or Destination MAC is not set in the MAC table or MAC.SL is not set for destination MAC} and FWPCi0.MACRUDSA is set.
- HW: When forwarding flow for a frame coming from port i goes until “L2 forwarding error” block in Figure 30.96 and, {FWPCi0.MACDSA is not set or Destination MAC is not set in the MAC table} and FWPCi0.MACRUDA is set.

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

LTWVUFS bit (Layer 2 VLAN Unknown Filtering Status Flag)

[Setting condition]

- HW: When forwarding flow for a frame coming from port i goes until “L2 forwarding secure error” block in Figure 30.95 and, {FWPCi0.VLANSA is not set or VLAN is not valid or VLAN ID is not set in the VLAN table or VLAN.SL is not set for VLAN ID} and FWPCi0.VLANRUS is set.
- HW: When forwarding flow for a frame coming from port i goes until “L2 forwarding error” block in Figure 30.96 and, {FWPCi0.VLANSA is not set or VLAN is not valid or VLAN ID is not set in the VLAN table} and FWPCi0.VLANRU is set.

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

PBNTFS bit (Port Based No Target Filtering Status Flag)

[Setting condition]

- HW: When forwarding flow for a frame coming from port i goes until “Port based forwarding secure error” block in Figure 30.99 and FWPBFCi.PBDV is null.
- HW: When forwarding flow for a frame coming from port i goes until “Port based forwarding unsecure error” block in Figure 30.100 and FWPBFCi.PBDV is null.

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

SMHLFS bit (Source MAC Hardware Learning Fail Status Flag)

[Setting condition]

HW: When a frame is selected for source MAC address HW learning (section 30.5.1.5.2. CPU Learning Path) and it could not be learnt because the learning rate is too high.

This error has nothing to do with fail of Table 30.45.

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

SMHMFS bit (Source MAC Hardware Migration Fail Status Flag)

[Setting condition]

- HW: When a frame is selected for source MAC address HW migration (section 30.5.1.5.2. CPU Learning Path) and it could not be learnt because the learning rate is too high.
- HW: When a frame is selected for source MAC address HW migration and its source MAC address contained in the table in static (MAC.DE is not set).
- HW: When a frame is selected for source MAC address HW migration and its source MAC address contained in the table is secure (MAC.SL is set).
- HW: When a frame is selected for source MAC address HW migration and its source MAC address contained in the table has one of its destination port which does not allow migration (for at least one i, MAC.DV[i] is set and FWPCi0.MACHMA is not set).

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

WMCFS bit (Watermark Critical Filtering Status Flag)

[Setting condition]

HW: A frame coming from port i is forwarded with a Layer 3/IP/Layer 2/Port based normal descriptor and FWLBWMCi.WMCLPR bit {DEI,IPV} is set and coa_watermark_critical[i] pin is set (where DEI is set to input frame C-TAG DEI if FWGC.SVM is set to 01b, input frame S-TAG DEI if FWGC.SVM is set to 10b and set to 0 if FWGC.SVM is set to 00b and IPV is the Frame corresponding normal descriptor, selected by the forwarding flow, FDESCR.IPV value).

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

WMFFS bit (Watermark Flush Filtering Status Flag)

[Setting condition]

HW: A frame coming from port i is forwarded with a Layer 3/IP/Layer 2/Port based normal descriptor and FWLBWMCi.WMFLPR bit {DEI,IPV} is set and coa_watermark_flush[i] pin is set (where DEI is set to input frame C-TAG DEI if FWGC.SVM is set to 01b, input frame S-TAG DEI if FWGC.SVM is set to 10b and set to 0 if FWGC.SVM is set to 00b and IPV is the Frame corresponding normal descriptor, selected by the forwarding flow, FDESCR.IPV value).

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

WMISFS bit (Watermark IPV Secure Filtering Status Flag)

[Setting condition]

HW: A frame coming from port i is forwarded with a Layer 3/IP/Layer 2/Port based normal descriptor and corresponding IPV is rejected by IPV based unsecure watermark [COMA] (where IPV is Frame corresponding normal descriptor, selected by the forwarding flow, FDESCR.IPV value before filtering).

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

WMIUFS bit (Watermark IPV Unsecure Filtering Status Flag)

[Setting condition]

HW: A frame coming from port i is forwarded with a Layer 3/IP/Layer 2/Port based normal descriptor and SEC is not set and corresponding IPV is rejected by IPV based secure watermark [COMA] (where SEC and IPV are respectively Frame corresponding normal descriptor, selected by the forwarding flow, FDESCR.SEC and FDESCR.IPV values before filtering).

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

DDES bit (Direct Descriptor Error Status Flag)

[Setting condition]

HW: When a direct descriptor (LDESCR.FMT [GWCA] set to 1) is received from port 2 + i and FWPCi1.DDE is not set.

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

DDSES bit (Direct Descriptor Security Error Status Flag)

[Setting condition]

HW: When a secure direct descriptor (LDESCR.FMT [GWCA] and LDESCR.SEC [GWCA] are both set to 1) is received from port 2 + i , FWPCi1.DDE is set and FWPCi1.DDSL is not set.

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

DDNTFS bit (Direct Descriptor No Target Filtering Status Flag)

[Setting condition]

HW: When a directed descriptor (LDESCR.FMT [GWCA] set to 1) is received from port 2 + i and LDESCR.DV [GWCA] is set to All0.

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

30.3.3.2 FWEIE0i : Port i Error Interrupt Enable Register 0 (i = 0 to 2)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7904 + 0x10 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	DDNTFE	DDSEE	DDFEE	DDEE	WMIUFE	WMISFE	WMFFE	WMCFE	—	—	SMHMFE	SMHLFE	PBNTFE	LTWVUFE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	LTWDUFE	LTWSUFE	LTWNTFE	LTWVSPFE	LTWSSPFE	LTWDSPFE	—	—	—	—	—	—	LTHUFE	LTHNTFE	—	LTHSPFE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LTHSPFE	Layer 3 Source Port Filtering Enable 0: Interrupt i disabled. 1: Interrupt i Enabled.	R/W*1
1	—	This bit is read as 0. The write value should be 0.	R/W
2	LTHNTFE	Layer 3 No Target Filtering Enable 0: Interrupt i disabled. 1: Interrupt i Enabled.	R/W*1
3	LTHUFE	Layer 3 Unknown Filtering Enable 0: Interrupt i disabled. 1: Interrupt i Enabled.	R/W*1
9:4	—	These bits are read as 0. The write value should be 0.	R/W
10	LTWDSPFE	Layer 2 Destination Source Port Filtering Enable 0: Interrupt i disabled. 1: Interrupt i Enabled.	R/W*1
11	LTWSSPFE	Layer 2 Source Source Port Filtering Enable 0: Interrupt i disabled. 1: Interrupt i Enabled.	R/W*1
12	LTWVSPFE	Layer 2 VLAN Source Port Filtering Enable 0: Interrupt i disabled. 1: Interrupt i Enabled.	R/W*1
13	LTWNTFE	Layer 2 No Target Filtering Enable 0: Interrupt i disabled. 1: Interrupt i Enabled.	R/W*1
14	LTWSUFE	Layer 2 Source Unknown Filtering Enable 0: Interrupt i disabled. 1: Interrupt i Enabled.	R/W*1
15	LTWDUFE	Layer 2 Destination Unknown Filtering Enable 0: Interrupt i disabled. 1: Interrupt i Enabled.	R/W*1
16	LTWVUFE	Layer 2 VLAN Unknown Filtering Enable 0: Interrupt i disabled. 1: Interrupt i Enabled.	R/W*1
17	PBNTFE	Port Based No Target Filtering Enable 0: Interrupt i disabled. 1: Interrupt i Enabled.	R/W*1
18	SMHLFE	Source MAC Hardware Learning Fail Enable 0: Interrupt i disabled. 1: Interrupt i Enabled.	R/W*1

Bit	Symbol	Function	R/W
19	SMHMFE	Source MAC Hardware Migration Fail Enable 0: Interrupt i disabled. 1: Interrupt i Enabled.	R/W*1
21:20	—	These bits are read as 0. The write value should be 0.	R/W
22	WMCFE	Watermark Critical Filtering Enable 0: Interrupt i disabled. 1: Interrupt i Enabled.	R/W*1
23	WMFFE	Watermark Flush Filtering Enable 0: Interrupt i disabled. 1: Interrupt i Enabled.	R/W*1
24	WMISFE	Watermark IPV Secure Filtering Enable 0: Interrupt i disabled. 1: Interrupt i Enabled.	R/W*1
25	WMIUFE	Watermark IPV Unsecure Filtering Enable 0: Interrupt i disabled. 1: Interrupt i Enabled.	R/W*1
26	DDEE	Direct Descriptor Error Enable 0: Interrupt i disabled. 1: Interrupt i Enabled.	R/W*1
27	DDFEE	Direct Descriptor Format Error Enable 0: Interrupt i disabled. 1: Interrupt i Enabled.	R/W*1
28	DDSEE	Direct Descriptor Security Error Enable 0: Interrupt i disabled. 1: Interrupt i Enabled.	R/W*1
29	DDNTFE	Direct Descriptor No Target Filtering Enable 0: Interrupt i disabled. 1: Interrupt i Enabled.	R/W*1
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

LTHSPFE bit (Layer 3 Source Port Filtering Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWEID0i.LTHSPFD register will clear this bit.

LTHNTFE bit (Layer 3 No Target Filtering Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWEID0i.LTHNTFD register will clear this bit.

LTHUFE bit (Layer 3 Unknown Filtering Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWEID0i.LTHUFD register will clear this bit.

LTWDSPFE bit (Layer 2 Destination Source Port Filtering Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWEID0i.LTWDSPFD register will clear this bit.

LTWSSPFE bit (Layer 2 Source Source Port Filtering Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWEID0i.LTWSSPFD register will clear this bit.

LTWVSPFE bit (Layer 2 VLAN Source Port Filtering Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWEID0i.LTWVSPFD register will clear this bit.

LTWNTFE bit (Layer 2 No Target Filtering Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWEID0i.LTWNTFD register will clear this bit.

LTWSUFE bit (Layer 2 Source Unknown Filtering Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWEID0i.LTWSUFD register will clear this bit.

LTWDUFE bit (Layer 2 Destination Unknown Filtering Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWEID0i.LTWDUFD register will clear this bit.

LTWVUFE bit (Layer 2 VLAN Unknown Filtering Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWEID0i.LTWVUFD register will clear this bit.

PBNTFE bit (Port Based No Target Filtering Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWEID0i.PBNTFD register will clear this bit.

SMHLFE bit (Source MAC Hardware Learning Fail Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWEID0i.SMHLFD register will clear this bit.

SMHMFE bit (Source MAC Hardware Migration Fail Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWEID0i.SMHMFD register will clear this bit.

WMCFE bit (Watermark Critical Filtering Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWEID0i.WMFFD register will clear this bit.

WMFFE bit (Watermark Flush Filtering Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWEID0i.WMFFE register will clear this bit.

WMISFE bit (Watermark IPV Secure Filtering Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWEID0i.WMISFD register will clear this bit.

WMIUFE bit (Watermark IPV Unsecure Filtering Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWEID0i.WMIUFD register will clear this bit.

DDEE bit (Direct Descriptor Error Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWEID0i.DDED register will clear this bit.

DDFEE bit (Direct Descriptor Format Error Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWEID0i.DDFED register will clear this bit.

DDSEE bit (Direct Descriptor Security Error Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWEID0i.DDSED register will clear this bit.

DDNTFE bit (Direct Descriptor No Target Filtering Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWEID0i.DDNTFD register will clear this bit.

30.3.3.3 FWEID0i : Port i Error Interrupt Disable Register 0 (i = 0 to 2)

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7908 + 0x10 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	DDNTFD	DDSED	DDFED	DDED	WMIUFD	WMISFD	WMFFD	WMCFD	—	—	SMHMFD	SMHLFD	PBNTFD	LTWVUFD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	LTWDUFD	LTWSUFD	LTWNTFD	LTWVSPFD	LTWSSPFD	LTWDSPFD	—	—	—	—	—	—	LTHUFD	LTHNTFD	—	LTHSPFD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LTHSPFD	Layer 3 Source Port Filtering Disable Writing 1 to this bit clears FWEIE0.LTHSPFE register.	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	LTHNTFD	Layer 3 No Target Filtering Disable Writing 1 to this bit clears FWEIE0.LTHNTFE register.	R/W
3	LTHUFD	Layer 3 Unknown Filtering Disable Writing 1 to this bit clears FWEIE0.LTHNTFE register.	R/W
9:4	—	These bits are read as 0. The write value should be 0.	R/W
10	LTWDSPFD	Layer 2 Destination Source Port Filtering Disable Writing 1 to this bit clears FWEIE0.LTWDSPFE register.	R/W
11	LTWSSPFD	Layer 2 Source Source Port Filtering Disable Writing 1 to this bit clears FWEIE0.LTWSSPFE register.	R/W
12	LTWVSPFD	Layer 2 VLAN Source Port Filtering Disable Writing 1 to this bit clears FWEIE0.LTWVSPFE register.	R/W
13	LTWNTFD	Layer 2 No Target Filtering Disable Writing 1 to this bit clears FWEIE0.LTWNTFE register.	R/W
14	LTWSUFD	Layer 2 Source Unknown Filtering Disable Writing 1 to this bit clears FWEIE0.LTWSUFE register.	R/W
15	LTWDUFD	Layer 2 Destination Unknown Filtering Disable Writing 1 to this bit clears FWEIE0.LTWDUFE register.	R/W
16	LTWVUFD	Layer 2 VLAN Unknown Filtering Disable Writing 1 to this bit clears FWEIE0.LTWVUFE register.	R/W
17	PBNTFD	Port Based No Target Filtering Disable Writing 1 to this bit clears FWEIE0.PBNTFE register.	R/W
18	SMHLFD	Source MAC Hardware Learning Fail Disable Writing 1 to this bit clears FWEIE0.SMHLFE register.	R/W

Bit	Symbol	Function	R/W
19	SMHMFD	Source MAC Hardware Migration Fail Disable Writing 1 to this bit clears FWEIE0.SMHMFE register.	R/W
21:20	—	These bits are read as 0. The write value should be 0.	R/W
22	WMCFD	Watermark Critical Filtering Disable Writing 1 to this bit clears FWEIE0.WMCFE register.	R/W
23	WMFFD	Watermark Flush Filtering Disable Writing 1 to this bit clears FWEIE0.WMFFE register.	R/W
24	WMISFD	Watermark IPV Secure Filtering Disable Writing 1 to this bit clears FWEIE0.WMISFE register.	R/W
25	WMIUFD	Watermark IPV Unsecure Filtering Disable Writing 1 to this bit clears FWEIE0.WMIUFE register.	R/W
26	DDED	Direct Descriptor Error Disable Writing 1 to this bit clears FWEIE0.DDEE register.	R/W
27	DDFED	Direct Descriptor Format Error Disable Writing 1 to this bit clears FWEIE0.DDFEE register.	R/W
28	DDSED	Direct Descriptor Security Error Disable Writing 1 to this bit clears FWEIE0.DDSEE register.	R/W
29	DDNTFD	Direct Descriptor No Target Filtering Disable Writing 1 to this bit clears FWEIE0.DDNTFE register.	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

30.3.3.4 FWEIS1 : Error Interrupt Status Register 1

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7A00

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

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

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	LTHTSES	L3 Table Security Error Status Flag	R/W*1
4:2	—	These bits are read as 0. The write value should be 0.	R/W
5	MACTSES	MAC Table Security Error Status Flag	R/W*1
6	—	This bit is read as 0. The write value should be 0.	R/W
7	VLANTSES	VLAN Table Security Error Status Flag	R/W*1
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

LTHTSES bit (L3 Table Security Error Status Flag)

[Setting condition]

HW: A CPU tried to learn an unsecure entry, but the unsecure entry maximum number is already reached.

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

- HW: Unsecure entry won't be learned
- SW: System dependent, cannot be defined here.

MACTSES bit (MAC Table Security Error Status Flag)

[Setting condition]

HW: A CPU tried to learn an unsecure entry, but the unsecure entry maximum number is already reached.

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

- HW: Unsecure entry won't be learned
- SW: System dependent, cannot be defined here.

VLANTSES bit (VLAN Table Security Error Status Flag)

[Setting condition]

HW: A CPU tried to learn an unsecure entry, but the unsecure entry maximum number is already reached.

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

- HW: Unsecure entry won't be learned
- SW: System dependent, cannot be defined here.

30.3.3.5 FWEIE1 : Error Interrupt Enable Register 1

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7A04

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

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

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	LTHTSEE	L3 Table Security Error Enable 0: Interrupt disabled. 1: Interrupt Enabled.	R/W*1
4:2	—	These bits are read as 0. The write value should be 0.	R/W
5	MACTSEE	MAC Table Security Error Enable 0: Interrupt disabled. 1: Interrupt Enabled.	R/W*1
6	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7	VLANTSEE	VLAN Table Security Error Enable 0: Interrupt disabled. 1: Interrupt Enabled.	R/W*1
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

LTHTSEE bit (L3 Table Security Error Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWEID1.LTHTSED register will clear this bit.

MACTSEE bit (MAC Table Security Error Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWEID1.MACTSED register will clear this bit.

VLANTSEE bit (VLAN Table Security Error Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWEID1.VLANTSED register will clear this bit.

30.3.3.6 FWEID1 : Error Interrupt Disable Register 1

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7A08

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	L23UEED	VLANTSED	—	MACTSED	—	—	—	LTHTSED	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	LTHTSED	L3 Table Security Error Disable Writing 1 to this bit will clear FWEIE1.LTHTSEE.	R/W
4:2	—	These bits are read as 0. The write value should be 0.	R/W
5	MACTSED	MAC Table Security Error Disable Writing 1 to this bit will clear FWEIE1.MACTSEE.	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	VLANTSED	VLAN Table Security Error Disable Writing 1 to this bit will clear FWEIE1.VLANTSEE.	R/W

Bit	Symbol	Function	R/W
8	L23UEED	Layer 2/Layer 3 Update Error Disable Writing 1 to this bit will clear FWEIE1.L23UEEE.	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

30.3.3.7 FWEIS2 : Error Interrupt Status Register 2

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7A10

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PMFS15 to PMFS0															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	PMFS15 to PMFS0	PSFP MSDU n Filtering Status Flag (n = 0 to 15)	R/W*1
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

PMFSn bits (PSFP MSDU n Filtering Status Flag (n = 0 to 15))

[Setting condition]

HW: Bit i is set when a descriptor is rejected by MSDU filter number i. (See section 30.5.3.1.1. MSDU Filter).

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

30.3.3.8 FWEIE2 : Error Interrupt Enable Register 2

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7A14

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PMFE15 to PMFE0															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	PMFE15 to PMFE0	PSFP MSDU n Filtering Enable (n = 0 to 15) 0: Interrupt disabled. 1: Interrupt Enabled.	R/W*1

Bit	Symbol	Function	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

PMFEn bits (PSFP MSDU n Filtering Enable (n = 0 to 15))

[Setting condition]

Writing 1 to one of these bits will set it.

[Clearing condition]

Writing 1 to FWEID2.PMFD register bit i will clear this register bit i.

30.3.3.9 FWEID2 : Error Interrupt Disable Register 2

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7A18

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PMFD15 to PMFD0															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	PMFD15 to PMFD0	PSFP MSDU n Filtering Disable (n = 0 to 15) Writing 1 to this register bit i will clear FWEIE2.PMFE register bit i.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

30.3.3.10 FWEIS5 : Error Interrupt Status Register 5

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7A40

Bit position: 31 to 0

Bit field: PMRFS31 to PMRFS0

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	PMRFS31 to PMRFS0	PSFP Meter n Filtering Status Flag (n = 0 to 31)	R/W*1

Note 1. Read value differs from written value.

PMRFSn bits (PSFP Meter n Filtering Status Flag (n = 0 to 31))

[Setting condition]

HW: Bit i is set when a descriptor is filtered by PSFP Meter filter number i. (See section 30.5.3.1.2. Meter Filter/ATS).

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

30.3.3.11 FWEIE5 : Error Interrupt Enable Register 5

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7A44

Bit position:	31 … 0
Bit field:	PMRFE31 to PMRFE0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	PMRFE31 to PMRFE0	PSFP Meter n Filtering Enable (n = 0 to 31)	R/W*1

Note 1. Read value differs from written value.

PMRFEn bits (PSFP Meter n Filtering Enable (n = 0 to 31))

- Bit i set to 0: Interrupt i disabled.
- Bit i set to 1: Interrupt i Enabled.

[Setting condition]

Writing 1 to one of these bits will set it.

[Clearing condition]

Writing 1 to FWEID5.PMRFD register bit i will clear this register bit i.

30.3.3.12 FWEID5 : Error Interrupt Disable Register 5

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7A48

Bit position:	31 … 0
Bit field:	PMRFD31 to PMRFD0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	PMRFD31 to PMRFD0	PSFP Meter n Filtering Disable (n = 0 to 31) Writing 1 to this register bit i will clear FWEIE5.PMRFD register bit i.	R/W

30.3.3.13 FWEIS60 : Error Interrupt Status Register 60

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7A50

Bit position:	31 … 0
Bit field:	FFS31 to FFS0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	FFS31 to FFS0	FRER n Filtering Status Flag (n = 96 to 127)	R/W*1

Note 1. Read value differs from written value.

FFSn bits (FRER n Filtering Status Flag (n = 96 to 127))

[Setting condition]

- HW: If FRER rule i+96 is used for individual recovery, bit i is set when a descriptor is filtered by FRER rule number i+96 because of no sequence error or struck error. (See (e) Individual recovery algorithms.)
- HW: If FRER rule i+96 is used for sequence recovery, bit i is set when a descriptor is filtered by FRER rule number i+96 because of no sequence error. (See (f) Sequence recovery algorithms.)

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

30.3.3.14 FWEIE60 : Error Interrupt Enable Register 60

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7A54

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FFE31 to FFE0																															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	FFE31 to FFE0	FRER n Filtering Enable (n = 96 to 127)	R/W*1

Note 1. Read value differs from written value.

FFEn bits (FRER n Filtering Enable (n = 96 to 127))

- Bit i set to 0: Interrupt i disabled.
- Bit i set to 1: Interrupt i Enabled.

[Setting condition]

Writing 1 to one of these bits will set it.

[Clearing condition]

Writing 1 to FWEID60.FFDi register bit i will clear this register bit i.

30.3.3.15 FWEID60 : Error Interrupt Disable Register 60

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7A58

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FFD31 to FFD0																															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	FFD31 to FFD0	FRER n Filtering Disable (n = 96 to 127) Writing 1 to this register bit i will clear FWEIE60.FFEi register bit i.	R/W

30.3.3.16 FWEIS61 : Error Interrupt Status Register 61

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7A60

Bit position:	31																														0
Bit field:	FFS31 to FFS0																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	FFS31 to FFS0	FRER n Filtering Status Flag (n = 64 to 95)	R/W*1

Note 1. Read value differs from written value.

FFSn bits (FRER n Filtering Status Flag (n = 64 to 95))

[Setting condition]

- HW: If FRER rule i+64 is used for individual recovery, bit i is set when a descriptor is filtered by FRER rule number i+64 because of no sequence error or struck error. (See (e) Individual recovery algorithms.)
- HW: If FRER rule i+64 is used for sequence recovery, bit i is set when a descriptor is filtered by FRER rule number i+64 because of no sequence error. (See (f) Sequence recovery algorithms.)

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

30.3.3.17 FWEIE61 : Error Interrupt Enable Register 61

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7A64

Bit position:	31																														0
Bit field:	FFE31 to FFE0																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	FFE31 to FFE0	FRER n Filtering Enable (n = 64 to 95)	R/W*1

Note 1. Read value differs from written value.

FFEn bits (FRER n Filtering Enable (n = 64 to 95))

- Bit i set to 0: Interrupt i disabled.
- Bit i set to 1: Interrupt i Enabled.

[Setting condition]

Writing 1 to one of these bits will set it.

[Clearing condition]

Writing 1 to FWEID61.FFDi register bit i will clear this register bit i.

30.3.3.18 FWEID61 : Error Interrupt Disable Register 61

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7A68

Bit position: 31 ... 0

Bit field: FFD31 to FFD0

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	FFD31 to FFD0	FRER n Filtering Disable (n = 64 to 95) Writing 1 to this register bit i will clear FWEIE61.FFEi register bit i.	R/W

30.3.3.19 FWEIS62 : Error Interrupt Status Register 62

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7A70

Bit position: 31 ... 0

Bit field: FFS31 to FFS0

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	FFS31 to FFS0	FRER n Filtering Status Flag (n = 32 to 63)	R/W*1

Note 1. Read value differs from written value.

FFSn bits (FRER n Filtering Status Flag (n = 32 to 63))

[Setting condition]

- HW: If FRER rule i+32 is used for individual recovery, bit i is set when a descriptor is filtered by FRER rule number i+32 because of no sequence error or struck error. (See (e) Individual recovery algorithms.)
- HW: If FRER rule i+32 is used for sequence recovery, bit i is set when a descriptor is filtered by FRER rule number i+32 because of no sequence error. (See (f) Sequence recovery algorithms.)

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

30.3.3.20 FWEIE62 : Error Interrupt Enable Register 62

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7A74

Bit position: 31 ... 0

Bit field: FFE31 to FFE0

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	FFE31 to FFE0	FRER n Filtering Enable (n = 32 to 63)	R/W*1

Note 1. Read value differs from written value.

FFEn bits (FRER n Filtering Enable (n = 32 to 63))

- Bit i set to 0: Interrupt i disabled.
- Bit i set to 1: Interrupt i Enabled.

[Setting condition]

Writing 1 to one of these bits will set it.

[Clearing condition]

Writing 1 to FWEID62.FFDi register bit i will clear this register bit i.

30.3.3.21 FWEID62 : Error Interrupt Disable Register 62

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7A78

Bit position:	31 ... 0
Bit field:	FFD31 to FFD0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	FFD31 to FFD0	FRER n Filtering Disable (n = 32 to 63) Writing 1 to this register bit i will clear FWEIE62.FFEi register bit i.	R/W

30.3.3.22 FWEIS63 : Error Interrupt Status Register 63

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7A80

Bit position:	31 ... 0
Bit field:	FFS31 to FFS0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	FFS31 to FFS0	FRER n Filtering Status Flag (n = 0 to 31)	R/W*1

Note 1. Read value differs from written value.

FFSn bits (FRER n Filtering Status Flag (n = 0 to 31))

[Setting condition]

- HW: If FRER rule i is used for individual recovery, bit i is set when a descriptor is filtered by FRER rule number i because of no sequence error or struck error. (See (e) Individual recovery algorithms.)
- HW: If FRER rule i is used for sequence recovery, bit i is set when a descriptor is filtered by FRER rule number i because of no sequence error. (See (f) Sequence recovery algorithms.)

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

Error recovery:

HW: Descriptor corresponding frame is rejected.

30.3.3.23 FWEIE63 : Error Interrupt Enable Register 63

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7A84

Bit position:	31 to 0
Bit field:	FFE31 to FFE0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	FFE31 to FFE0	FRER n Filtering Enable (n = 0 to 31)	R/W*1

Note 1. Read value differs from written value.

FFEn bits (FRER n Filtering Enable (n = 0 to 31))

- Bit i set to 0: Interrupt i disabled.
- Bit i set to 1: Interrupt i Enabled.

[Setting condition]

Writing 1 to one of these bits will set it.

[Clearing condition]

Writing 1 to FWEID63.FFDi register bit i will clear this register bit i.

30.3.3.24 FWEID63 : Error Interrupt Disable Register 63

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7A88

Bit position:	31 to 0
Bit field:	FFD31 to FFD0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	FFD31 to FFD0	FRER n Filtering Disable (n = 0 to 31) Writing 1 to this register bit i will clear FWEIE63.FFEi register bit i.	R/W

30.3.3.25 FWEIS70 : Error Interrupt Status Register 70

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7A90

Bit position:	31 to 0
Bit field:	FOORS31 to FOORS0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	FOORS31 to FOORS0	FRER n Out Of Range Status Flag (n = 96 to 127)	R/W*1

Note 1. Read value differs from written value.

FOORSn bits (FRER n Out Of Range Status Flag (n = 96 to 127))

[Setting condition]

- HW: If FRER rule i+96 is used with match algorithm, bit i is set when a descriptor an out of order error occurs. (See (e) Individual recovery algorithms and (f) Sequence recovery algorithms).
- HW: : If FRER rule i+96 is used with vector algorithm, bit i is set when a descriptor is filtered by FRER rule number i+96 because of an out of range error. (See (e) Individual recovery algorithms and (f) Sequence recovery algorithms).

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

30.3.3.26 FWEIE70 : Error Interrupt Enable Register 70

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7A94

Bit position:	31	0
Bit field:	FOORE31 to FOORE0	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	FOORE31 to FOORE0	FRER n Out Of Range Enable (n = 96 to 127)	R/W*1

Note 1. Read value differs from written value.

FOOREn bits (FRER n Out Of Range Enable (n = 96 to 127))

- Bit i set to 0: Interrupt i disabled.
- Bit i set to 1: Interrupt i Enabled.

[Setting condition]

Writing 1 to one of these bits will set it.

[Clearing condition]

Writing 1 to FWEID70.FOORDi register bit i will clear this register bit i.

30.3.3.27 FWEID70 : Error Interrupt Disable Register 70

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7A98

Bit position:	31	0
Bit field:	FOORD31 to FOORD0	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	FOORD31 to FOORD0	FRER n Out Of Range Disable (n = 96 to 127) Writing 1 to this register bit i will clear FWEIE70.FOOREi register bit i.	R/W

30.3.3.28 FWEIS71 : Error Interrupt Status Register 71

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7AA0

Bit position:	31 … 0
Bit field:	FOORS31 to FOORS0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	FOORS31 to FOORS0	FRER n Out Of Range Status Flag (n = 64 to 95)	R/W*1

Note 1. Read value differs from written value.

FOORSn bits (FRER n Out Of Range Status Flag (n = 64 to 95))

[Setting condition]

- HW: If FRER rule i+64 is used with match algorithm, bit i is set when a descriptor an out of order error occurs. (See (e) Individual recovery algorithms and (f) Sequence recovery algorithms).
- HW: : If FRER rule i+64 is used with vector algorithm, bit i is set when a descriptor is filtered by FRER rule number i+64 because of an out of range error. (See (e) Individual recovery algorithms and (f) Sequence recovery algorithms).

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

30.3.3.29 FWEIE71 : Error Interrupt Enable Register 71

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7AA4

Bit position:	31 … 0
Bit field:	FOORE31 to FOORE0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	FOORE31 to FOORE0	FRER n Out Of Range Enable (n = 64 to 95)	R/W*1

Note 1. Read value differs from written value.

FOOREn bits (FRER n Out Of Range Enable (n = 64 to 95))

- Bit i set to 0: Interrupt i disabled.
- Bit i set to 1: Interrupt i Enabled.

[Setting condition]

Writing 1 to one of these bits will set it.

[Clearing condition]

Writing 1 to FWEID71.FOORDi register bit i will clear this register bit i.

30.3.3.30 FWEID71 : Error Interrupt Disable Register 71

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7AA8

Bit position:	31 … 0
Bit field:	FOORD31 to FOORD0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	FOORD31 to FOORD0	FRER n Out Of Range Disable (n = 64 to 95) Writing 1 to this register bit i will clear FWEIE71.FOOREi register bit i.	R/W

30.3.3.31 FWEIS72 : Error Interrupt Status Register 72

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7AB0

Bit position:	31 … 0
Bit field:	FOORS31 to FOORS0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	FOORS31 to FOORS0	FRER n Out Of Range Status Flag (n = 32 to 63)	R/W*1

Note 1. Read value differs from written value.

FOORSn bits (FRER n Out Of Range Status Flag (n = 32 to 63))

[Setting condition]

- HW: If FRER rule i+32 is used with match algorithm, bit i is set when a descriptor an out of order error occurs. (See (e) Individual recovery algorithms and (f) Sequence recovery algorithms).
- HW: If FRER rule i+32 is used with vector algorithm, bit i is set when a descriptor is filtered by FRER rule number i+32 because of an out of range error. (See (e) Individual recovery algorithms and (f) Sequence recovery algorithms).

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

30.3.3.32 FWEIE72 : Error Interrupt Enable Register 72

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7AB4

Bit position:	31 … 0
Bit field:	FOORE31 to FOORE0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	FOORE31 to FOORE0	FRER n Out Of Range Enable (n = 32 to 63)	R/W*1

Note 1. Read value differs from written value.

FOOREn bits (FRER n Out Of Range Enable (n = 32 to 63))

- Bit i set to 0: Interrupt i disabled.
- Bit i set to 1: Interrupt i Enabled.

[Setting condition]

Writing 1 to one of these bits will set it.

[Clearing condition]

Writing 1 to FWEID72.FOORDi register bit i will clear this register bit i.

30.3.3.33 FWEID72 : Error Interrupt Disable Register 72

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7AB8

Bit position:	31 … 0
Bit field:	FOORD31 to FOORD0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	FOORD31 to FOORD0	FRER n Out Of Range Disable (n = 32 to 63) Writing 1 to this register bit i will clear FWEIE72.FOOREi register bit i.	R/W

30.3.3.34 FWEIS73 : Error Interrupt Status Register 73

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7AC0

Bit position:	31 … 0
Bit field:	FOORS31 to FOORS0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	FOORS31 to FOORS0	FRER n Out Of Range Status Flag (n = 0 to 31)	R/W*1

Note 1. Read value differs from written value.

FOORSn bits (FRER n Out Of Range Status Flag (n = 0 to 31))

[Setting condition]

- HW: If FRER rule i is used with match algorithm, bit i is set when a descriptor an out of order error occurs. (See (e) Individual recovery algorithms and (f) Sequence recovery algorithms).
- HW: If FRER rule i is used with vector algorithm, bit i is set when a descriptor is filtered by FRER rule number i because of an out of range error. (See (e) Individual recovery algorithms and (f) Sequence recovery algorithms).

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

30.3.3.35 FWEIE73 : Error Interrupt Enable Register 73

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7AC4

Bit position:	31 … 0
Bit field:	FOORE31 to FOORE0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	FOORE31 to FOORE0	FRER n Out Of Range Enable (n = 0 to 31)	R/W*1

Note 1. Read value differs from written value.

FOOREn bits (FRER n Out Of Range Enable (n = 0 to 31))

- Bit i set to 0: Interrupt i disabled.
- Bit i set to 1: Interrupt i Enabled.

[Setting condition]

Writing 1 to one of these bits will set it.

[Clearing condition]

Writing 1 to FWEID73.FOORDi register bit i will clear this register bit i.

30.3.3.36 FWEID73 : Forwarding Engine Error Interrupt Disable 73

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7AC8

Bit position:	31 … 0
Bit field:	FOORD31 to FOORD0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	FOORD31 to FOORD0	FRER n Out Of Range Disable (n = 0 to 31) Writing 1 to this register bit i will clear FWEIE73.FOOREi register bit i.	R/W

30.3.3.37 FWEIS80 : Error Interrupt Status Register 80

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7AD0

Bit position:	31 … 0
Bit field:	TOS31 to TOS0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TOS31 to TOS0	Timeout Status Flag n (n = 96 to 127)	R/W*1

Note 1. Read value differs from written value.

TOSn bits (Timeout Status Flag n (n = 96 to 127))

[Setting condition]

HW: When timeout happens for FRER rule i+96. (See (3) Timeout.)

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

30.3.3.38 FWEIE80 : Error Interrupt Enable Register 80

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7AD4

Bit position:	31																														0
Bit field:	TOE31 to TOE0																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TOE31 to TOE0	Timeout Enable n (n = 96 to 127)	R/W*1

Note 1. Read value differs from written value.

TOEn bits (Timeout Enable n (n = 96 to 127))

- Bit i set to 0: Interrupt i disabled.
- Bit i set to 1: Interrupt i Enabled.

[Setting condition]

Writing 1 to one of these bits will set it.

[Clearing condition]

Writing 1 to FWEID80.TODi register bit i will clear this register bit i.

30.3.3.39 FWEID80 : Error Interrupt Disable Register 80

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7AD8

Bit position:	31																														0
Bit field:	TOD31 to TOD0																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TOD31 to TOD0	Timeout Disable n (n = 96 to 127) Writing 1 to this register bit i will clear FWEIE80.TOEi register bit i.	R/W

30.3.3.40 FWEIS81 : Error Interrupt Status Register 81

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7AE0

Bit position:	31	0
Bit field:	TOS31 to TOS0	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	TOS31 to TOS0	Timeout Status Flag n (n = 64 to 95)	R/W*1

Note 1. Read value differs from written value.

TOSn bits (Timeout Status Flag n (n = 64 to 95))

[Setting condition]

HW: When timeout happens for FRER rule i+64. (See (3) Timeout.)

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

30.3.3.41 FWEIE81 : Error Interrupt Enable Register 81

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7AE4

Bit position:	31	0
Bit field:	TOE31 to TOE0	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	TOE31 to TOE0	Timeout Enable n (n = 64 to 95)	R/W*1

Note 1. Read value differs from written value.

TOEn bits (Timeout Enable n (n = 64 to 95))

- Bit i set to 0: Interrupt i disabled.
- Bit i set to 1: Interrupt i Enabled.

[Setting condition]

Writing 1 to one of these bits will set it.

[Clearing condition]

Writing 1 to FWEID81.TODi register bit i will clear this register bit i.

30.3.3.42 FWEID81 : Error Interrupt Disable Register 81

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7AE8

Bit position:	31 … 0
Bit field:	TOD31 to TOD0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TOD31 to TOD0	Timeout Disable n (n = 64 to 95) Writing 1 to this register bit i will clear FWEIE81.TOEi register bit i.	R/W

30.3.3.43 FWEIS82 : Error Interrupt Status Register 82

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7AF0

Bit position:	31 … 0
Bit field:	TOS31 to TOS0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TOS31 to TOS0	Timeout Status Flag n (n = 32 to 63)	R/W*1

Note 1. Read value differs from written value.

TOSn bits (Timeout Status Flag n (n = 32 to 63))

[Setting condition]

HW: When timeout happens for FRER rule i+32. (See (3) Timeout.)

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

30.3.3.44 FWEIE82 : Error Interrupt Enable Register 82

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7AF4

Bit position:	31 … 0
Bit field:	TOE31 to TOE0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TOE31 to TOE0	Timeout Enable n (n = 32 to 63)	R/W*1

Note 1. Read value differs from written value.

TOEn bits (Timeout Enable n (n = 32 to 63))

- Bit i set to 0: Interrupt i disabled.
- Bit i set to 1: Interrupt i Enabled.

[Setting condition]

Writing 1 to one of these bits will set it.

[Clearing condition]

Writing 1 to FWEID82.TODi register bit i will clear this register bit i.

30.3.3.45 FWEID82 : Error Interrupt Disable Register 82

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7AF8

Bit position:	31 to 0
Bit field:	TOD31 to TOD0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TOD31 to TOD0	Timeout Disable n (n = 32 to 63) Writing 1 to this register bit i will clear FWEIE82.TOEi register bit i.	R/W

30.3.3.46 FWEIS83 : Error Interrupt Status Register 83

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7B00

Bit position:	31 to 0
Bit field:	TOS31 to TOS0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TOS31 to TOS0	Timeout Status Flag n (n = 0 to 31)	R/W*1

Note 1. Read value differs from written value.

TOSn bits (Timeout Status Flag n (n = 0 to 31))

[Setting condition]

HW: When timeout happens for FRER rule i. (See (3) Timeout.)

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Descriptor corresponding frame is rejected.

30.3.3.47 FWEIE83 : Error Interrupt Enable Register 83

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7B04

Bit position:	31–0
Bit field:	TOE31 to TOE0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TOE31 to TOE0	Timeout Enable n (n = 0 to 31)	R/W*1

Note 1. Read value differs from written value.

TOEn bits (Timeout Enable n (n = 0 to 31))

- Bit i set to 0: Interrupt i disabled.
- Bit i set to 1: Interrupt i Enabled.

[Setting condition]

Writing 1 to one of these bits will set it.

[Clearing condition]

Writing 1 to FWEID83.TODi register bit i will clear this register bit i.

30.3.3.48 FWEID83 : Error Interrupt Disable Register 83

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7B08

Bit position:	31–0
Bit field:	TOD31 to TOD0
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TOD31 to TOD0	Timeout Disable n (n = 0 to 31) Writing 1 to this register bit i will clear FWEIE83.TOEi register bit i.	R/W

30.3.3.49 FWMIS0 : Monitoring Interrupt Status Register 0

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7C00

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

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

Bit	Symbol	Function	R/W
0	LTHTFS	L3 Table Full Status Flag	R/W*1
1	—	This bit is read as 0. The write value should be 0.	R/W
2	MACTFS	MAC Table Full Status Flag	R/W*1
3	VLANTFS	VLAN Table Full Status Flag	R/W*1
16:4	—	These bits are read as 0. The write value should be 0.	R/W
17	MACADAS	MAC Address Deleted Aging Status Flag	R/W*1
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

LTHTFS bit (L3 Table Full Status Flag)

[Setting condition]

HW: L3 table becomes full

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

MACTFS bit (MAC Table Full Status Flag)

[Setting condition]

HW: MAC table becomes full

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

VLANTFS bit (VLAN Table Full Status Flag)

[Setting condition]

HW: VLAN table becomes full

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

MACADAS bit (MAC Address Deleted Aging Status Flag)

[Setting condition]

HW: A MAC address has been suppressed by aging or dynamic entry delete

[Clearing condition]

SW: Writing 1 to one of these bits will clear it.

30.3.3.50 FWMIE0 : Monitoring Interrupt Enable Register 0

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7C04

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

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

Bit	Symbol	Function	R/W
0	LTHTFE	L3 Table Full Enable 0: Interrupt disabled. 1: Interrupt Enabled.	R/W*1
1	—	This bit is read as 0. The write value should be 0.	R/W
2	MACTFE	MAC Table Full Enable 0: Interrupt disabled. 1: Interrupt Enabled.	R/W*1
3	VLANTFE	VLAN Table Full Enable 0: Interrupt disabled. 1: Interrupt Enabled.	R/W*1
16:4	—	These bits are read as 0. The write value should be 0.	R/W
17	MACADAE	MAC Address Deleted Aging Enable 0: Interrupt disabled. 1: Interrupt Enabled.	R/W*1
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

LTHTFE bit (L3 Table Full Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWMID0.LTHTFD register will clear this bit.

MACTFE bit (MAC Table Full Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWMID0.MACTFD register will clear this bit.

VLANTFE bit (VLAN Table Full Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWMID0.VLANTFD register will clear this bit.

MACADAE bit (MAC Address Deleted Aging Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to FWMID0.MACADAD register will clear this bit.

30.3.3.51 FWMID0 : Monitoring Interrupt Disable Register 0

Base address: MFWD = 0x403C_0000
MFWD_NS = 0x503C_0000

Offset address: 0x7C08

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

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

Bit	Symbol	Function	R/W
0	LTHTFD	L3 Table Full Disable Writing 1 to this bit will clear FWMIE0.LTHTFE.	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	MACTFD	MAC Table Full Disable Writing 1 to this bit will clear FWMIE0.MACTFE.	R/W
3	VLANTFD	VLAN Table Full Disable Writing 1 to this bit will clear FWMIE0.VLANTFE.	R/W
16:4	—	These bits are read as 0. The write value should be 0.	R/W
17	MACADAD	MAC Address Deleted Aging Disable Writing 1 to this bit will clear FWMIE0.MACADAD.	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

30.4 Register Utilization

30.4.1 Software Flows

Note:
- SW: Forwarding engine registers can only be written using the flows specified in this section
- SW: All the software flows described in this section can only be used sequentially.

30.4.1.1 Software Flow Legend

Figure 30.2 shows the software flow legend.

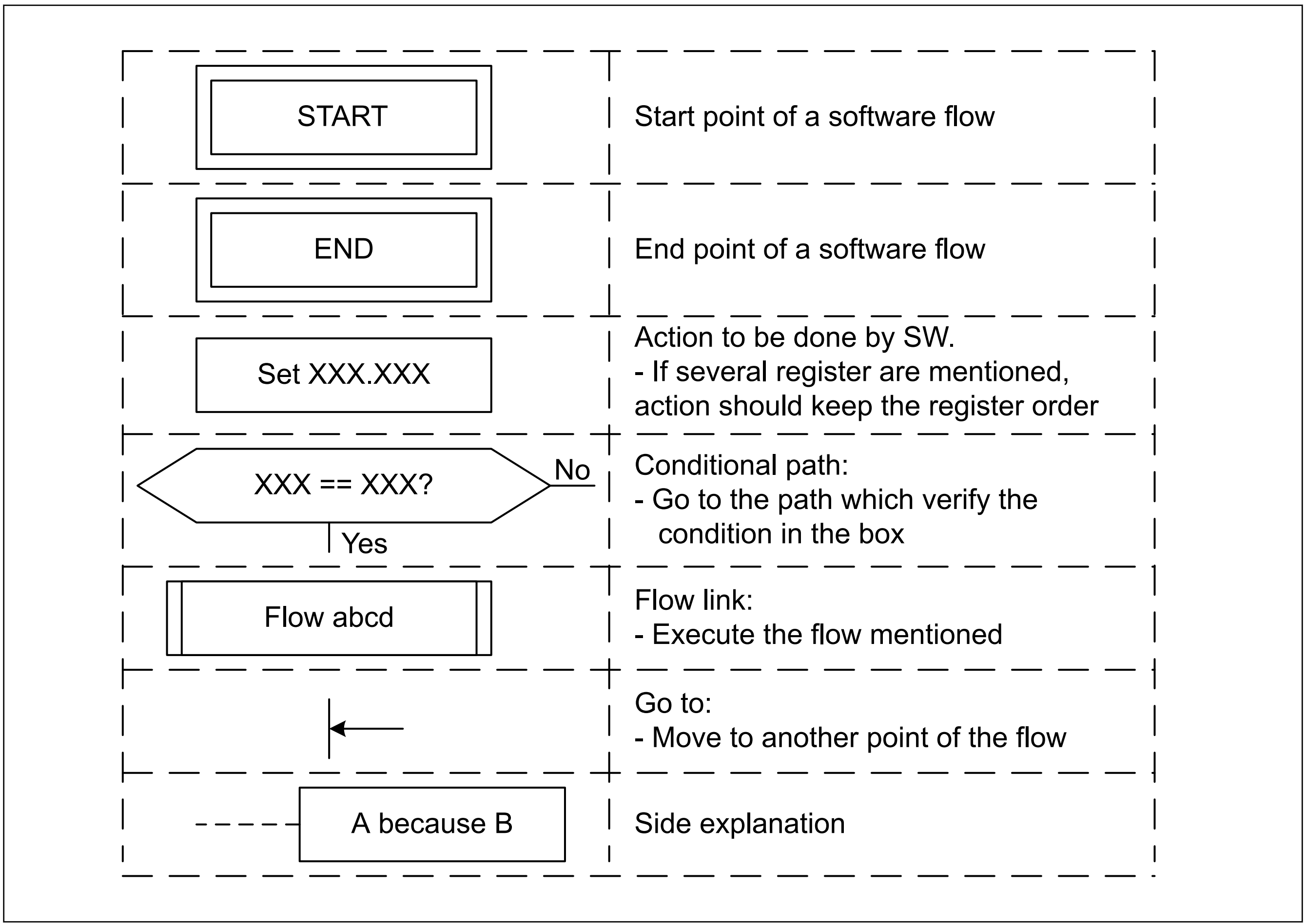


Figure 30.2 Software flow legend

30.4.1.2 Interrupt Handling Flow

Interrupt handling flow can be used in any mode except RESET mode.

Figure 30.3 describes the interrupt handling flow.

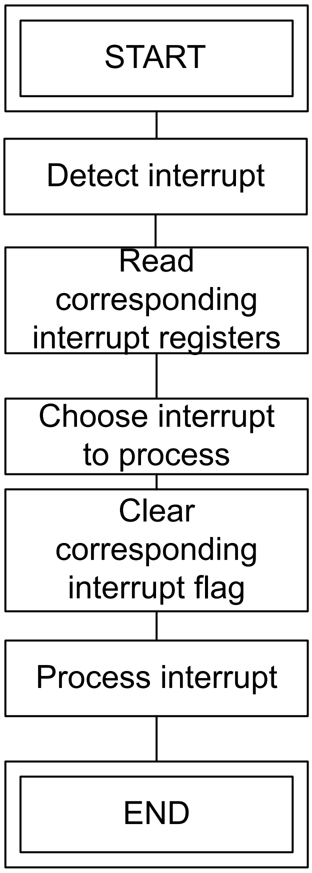


Figure 30.3 Interrupt handling flow

30.4.1.3 Port i Setting Flow

Port i setting flow is described in Figure 30.4.

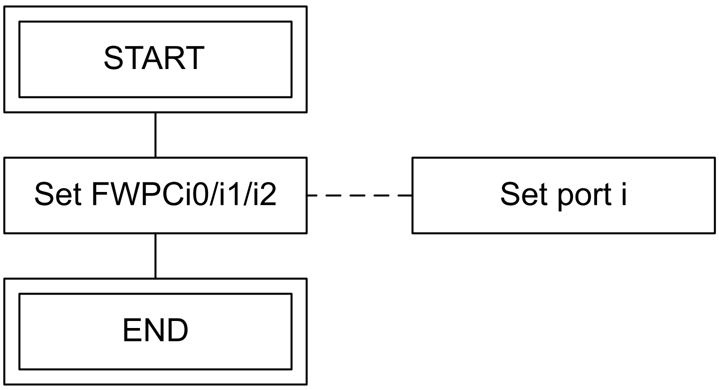


Figure 30.4 Port i setting flow

30.4.1.4 Cut-through Forwarding Flows

30.4.1.4.1 Cut-through Rule i Setting Flow

Cut-through rule i setting flow is described in Figure 30.5.

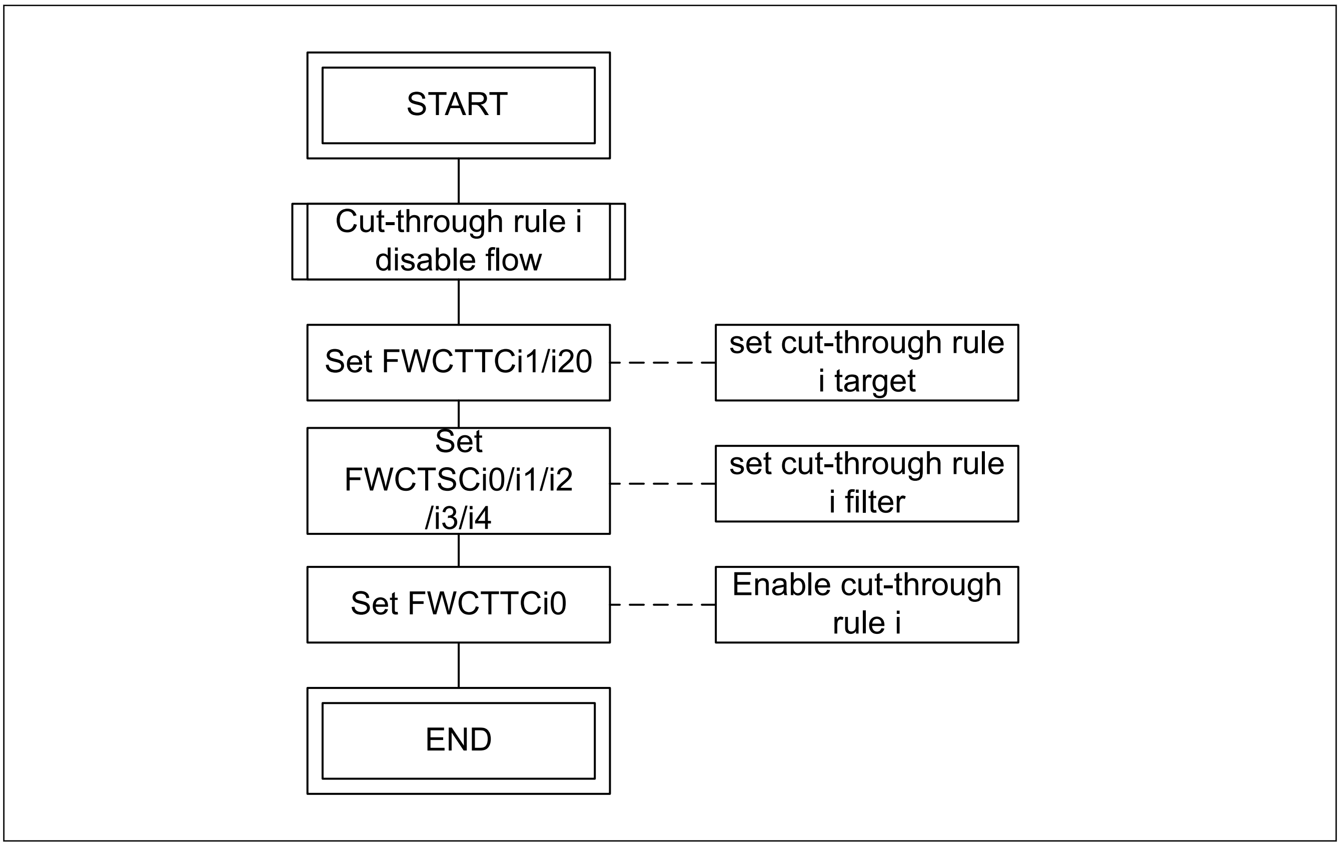


Figure 30.5 Cut-through rule i setting flow

30.4.1.4.2 Cut-through Rule i Disable Flow

Cut-through rule i disable flow is described in Figure 30.6.

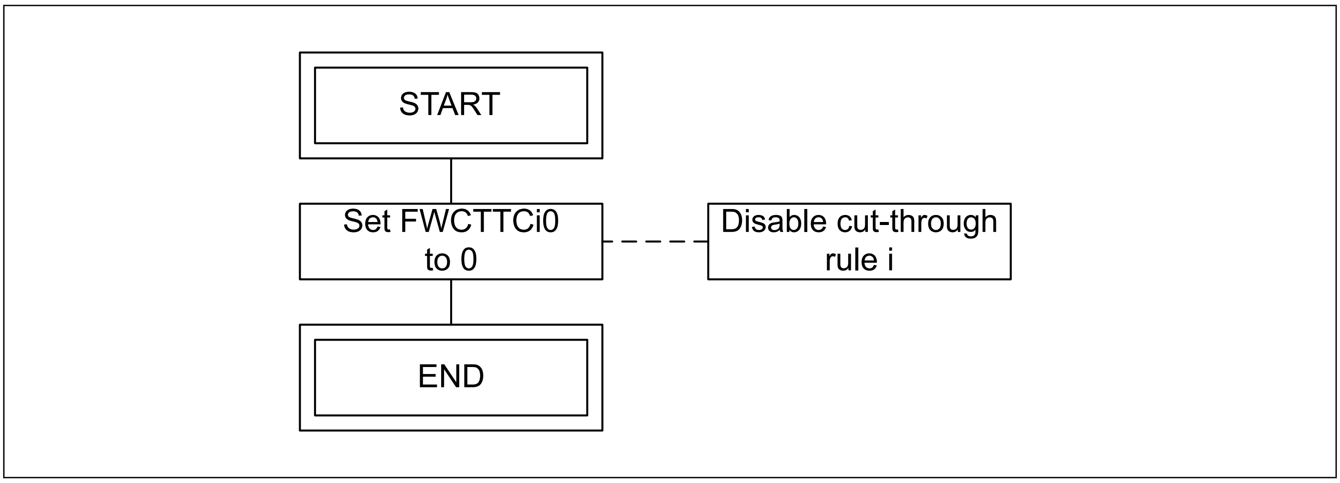


Figure 30.6 Cut-through rule i disable flow

30.4.1.5 Layer 3 Forwarding/Routing/Filtering Flows

30.4.1.5.1 Perfect Filter i Setting Flow

Figure 30.7 describes the setting flow of perfect filter i.

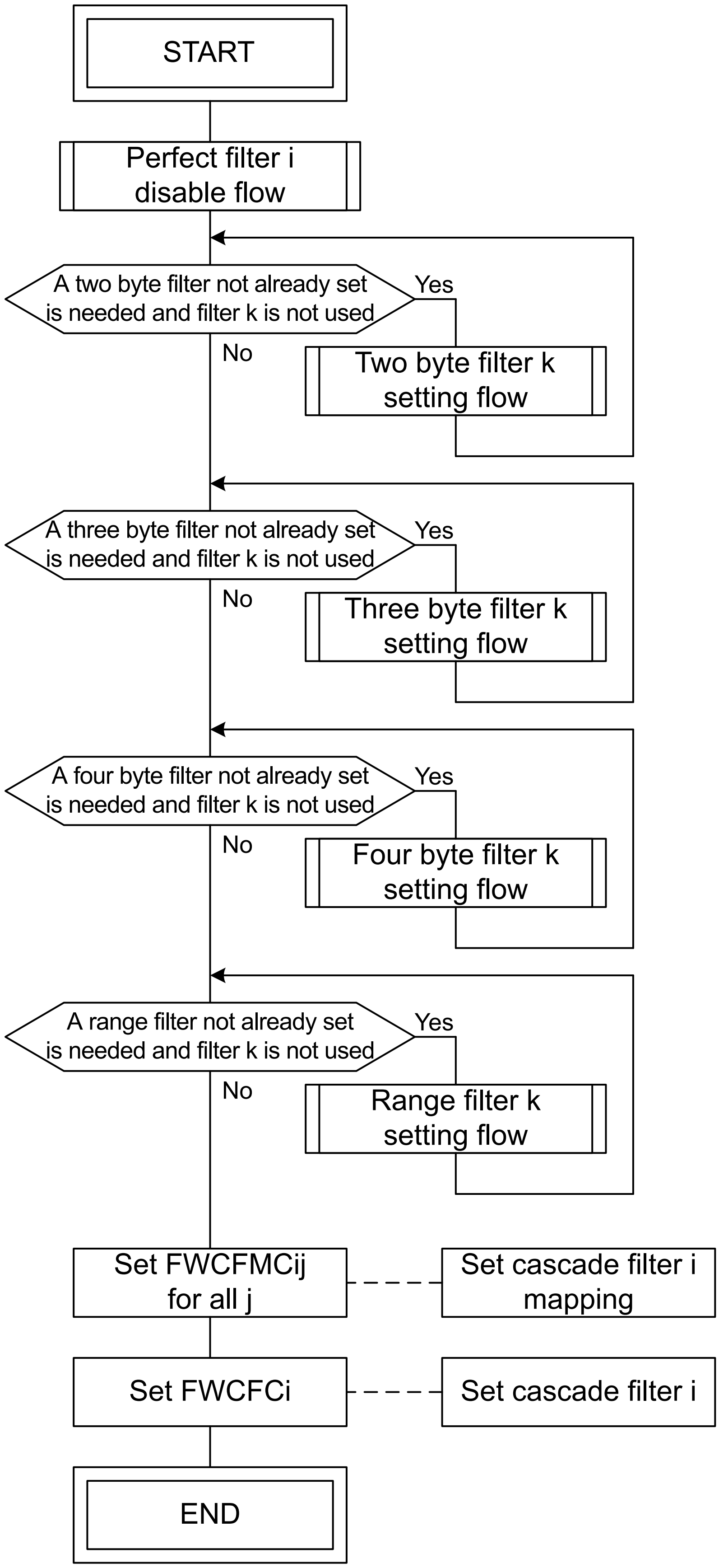


Figure 30.7 Setting flow of perfect filter i

30.4.1.5.2 Perfect Filter i Disable Flow

Figure 30.8 describes the disable flow of perfect filter i.

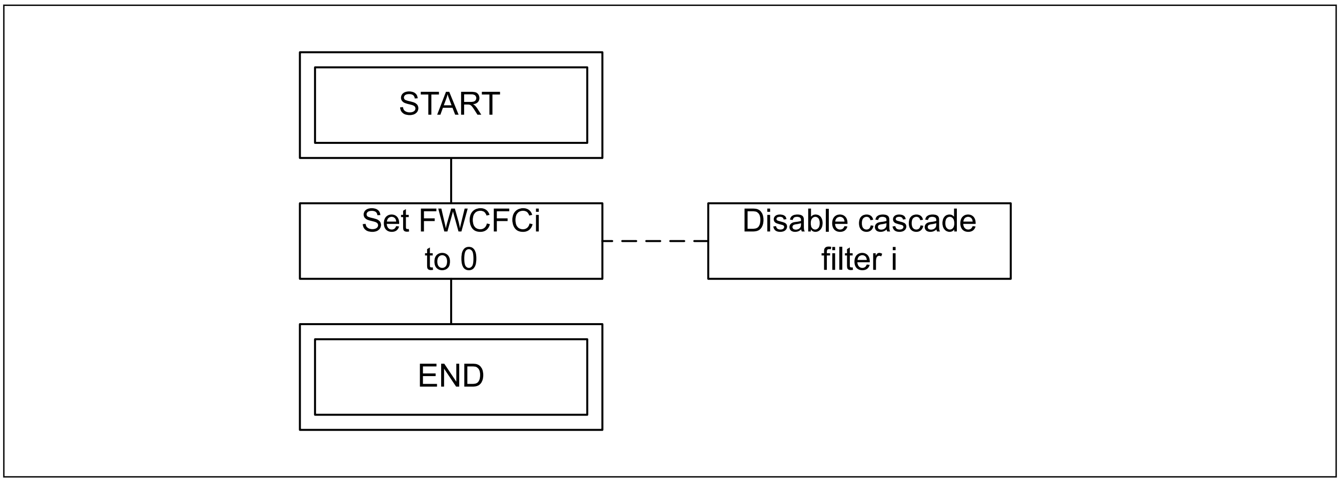


Figure 30.8 Disable flow of perfect filter i

30.4.1.5.3 Perfect Filter Priority Change Flow

Figure 30.9 describes the change flow of perfect filter priority.

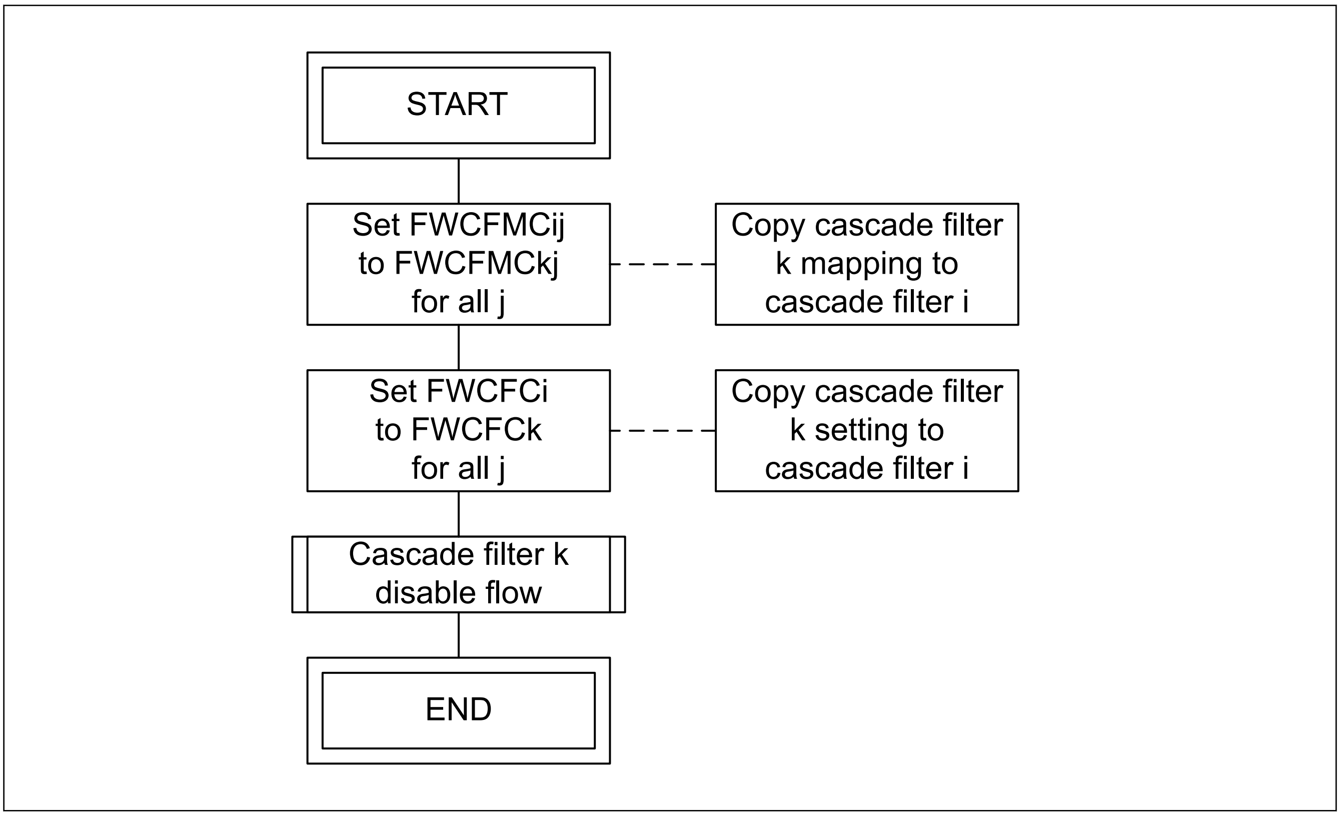


Figure 30.9 Change flow of perfect filter priority (from k to i)

30.4.1.5.4 Hash Hardware Calculation Flow (for IPv4/IPv6 Stream Creation)

Hash hardware calculation flow is described in Figure 30.10.

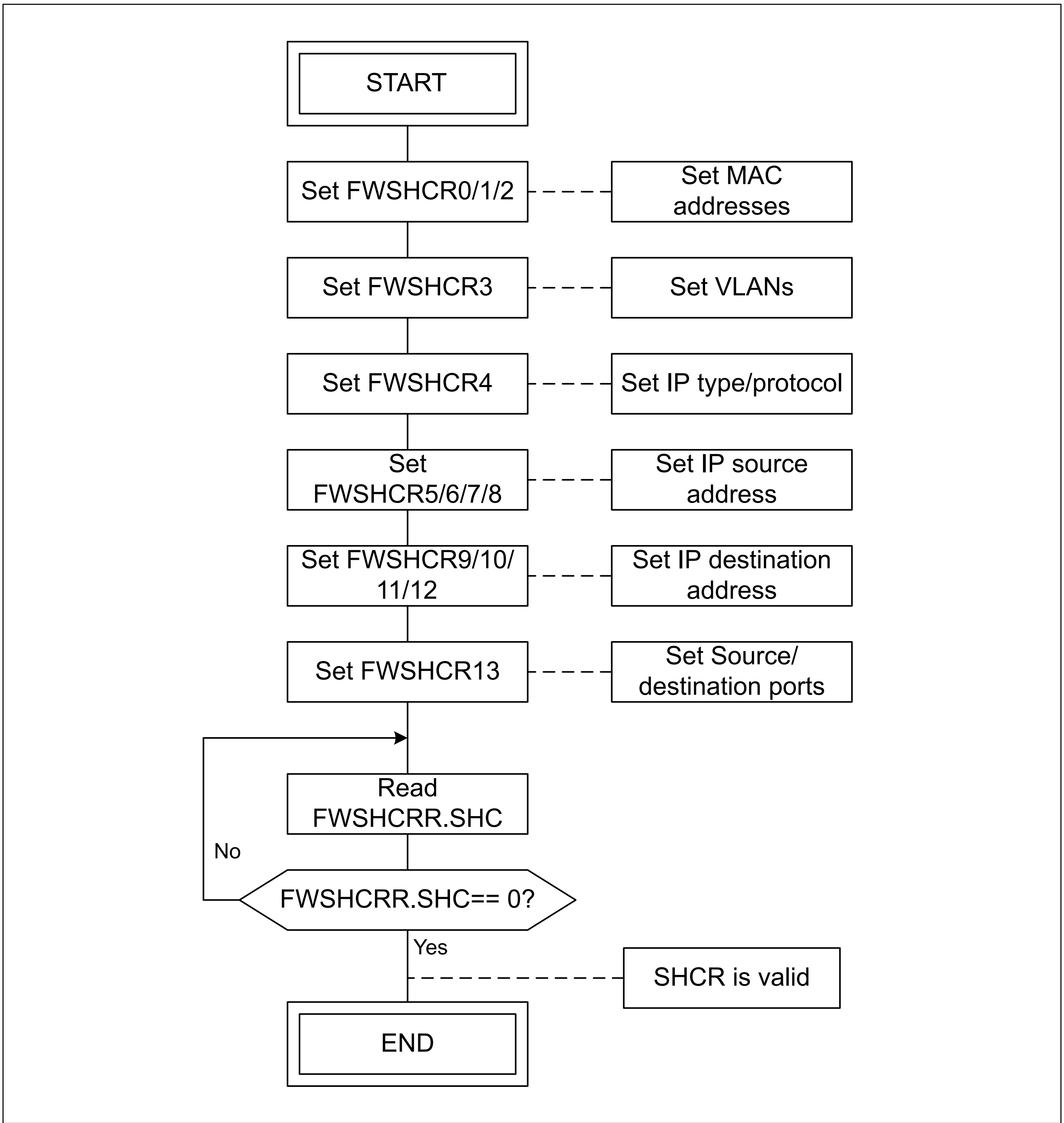


Figure 30.10 Hash hardware calculation flow

30.4.1.5.5 Layer 3 Table Reset Flow

Figure 30.11 describes the reset flow of Layer 3 table.

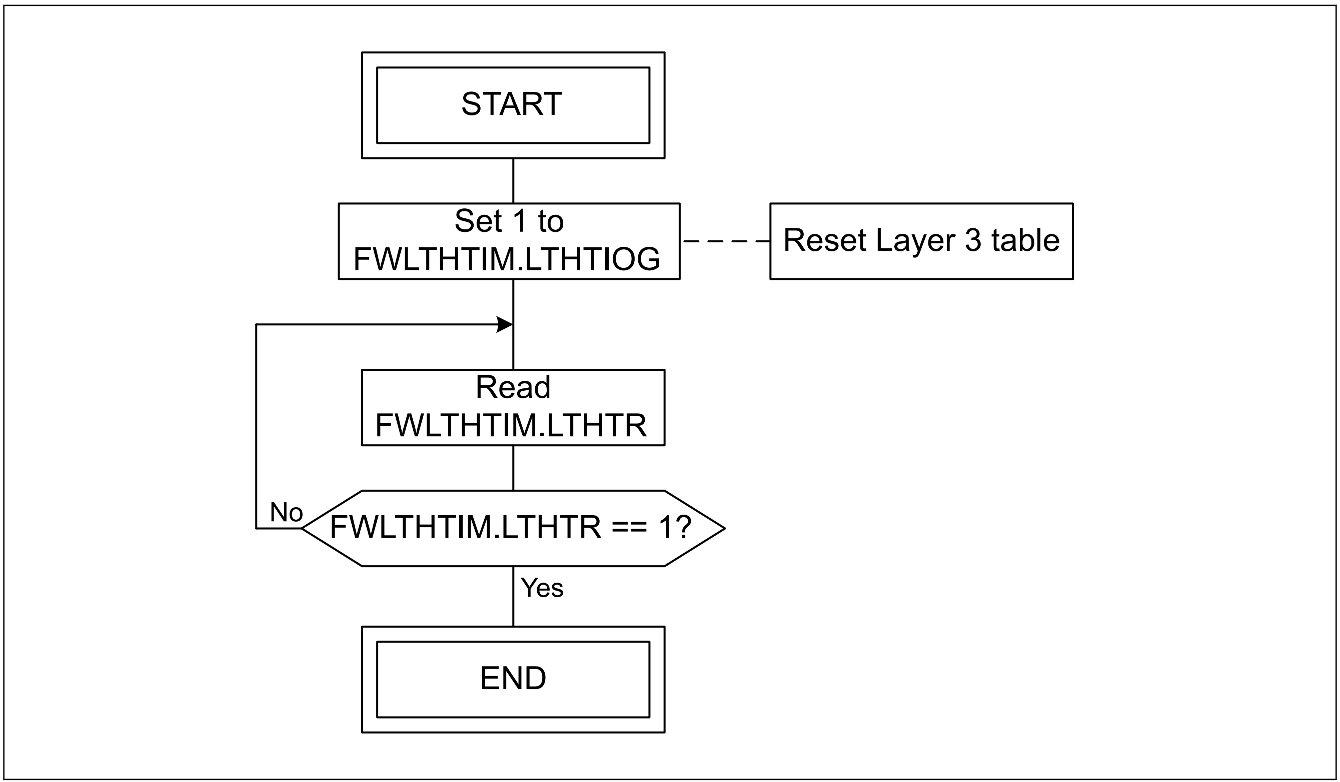


Figure 30.11 Reset flow of Layer 3 table

30.4.1.5.6 Layer 3 Entry Learn Flow

Layer 3 entry learn flow is described in Figure 30.12.

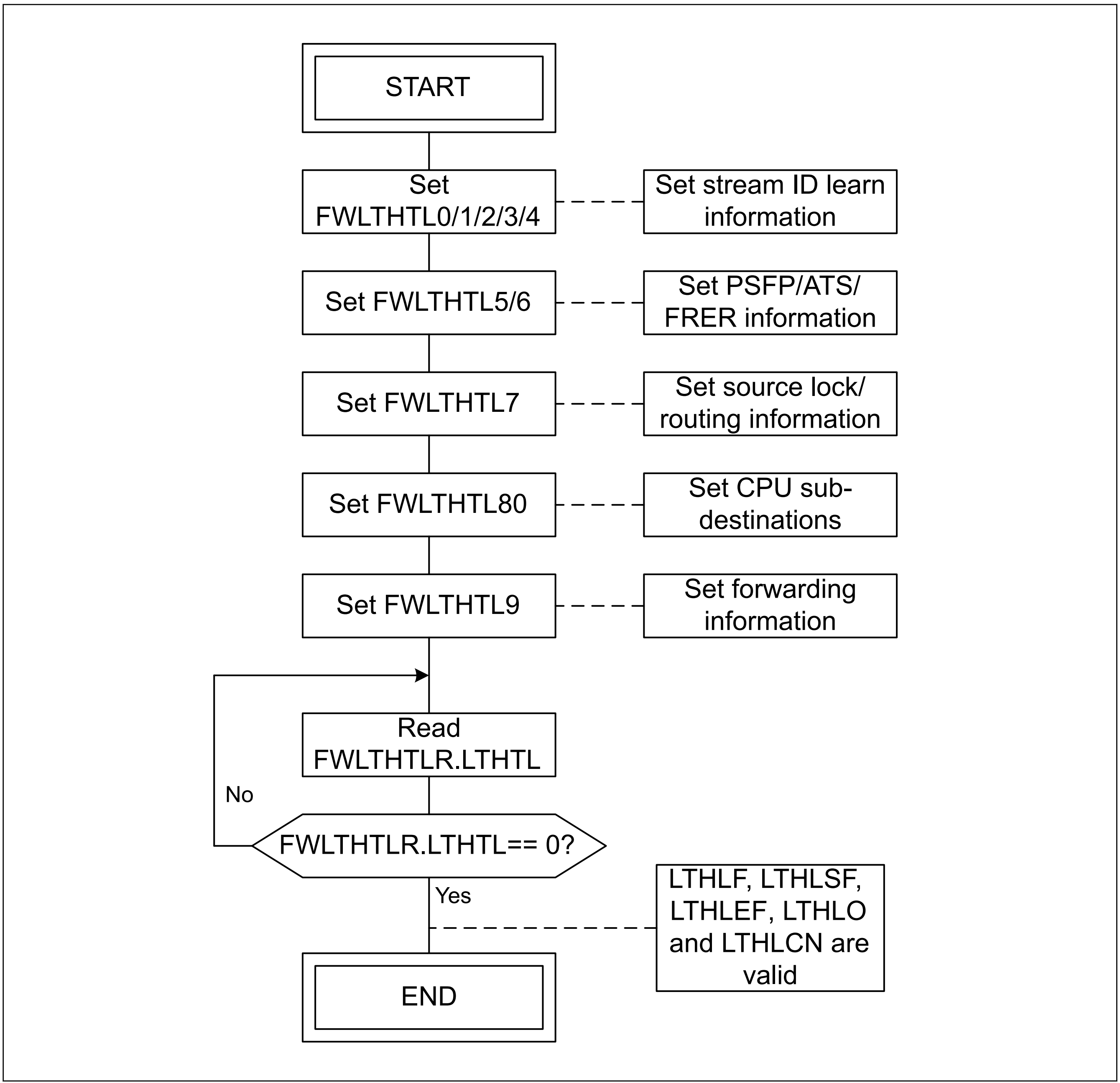


Figure 30.12 Layer 3 entry learn flow

30.4.1.5.7 Layer 3 Entry Search Flow

Layer 3 entry search flow is described in Figure 30.13.

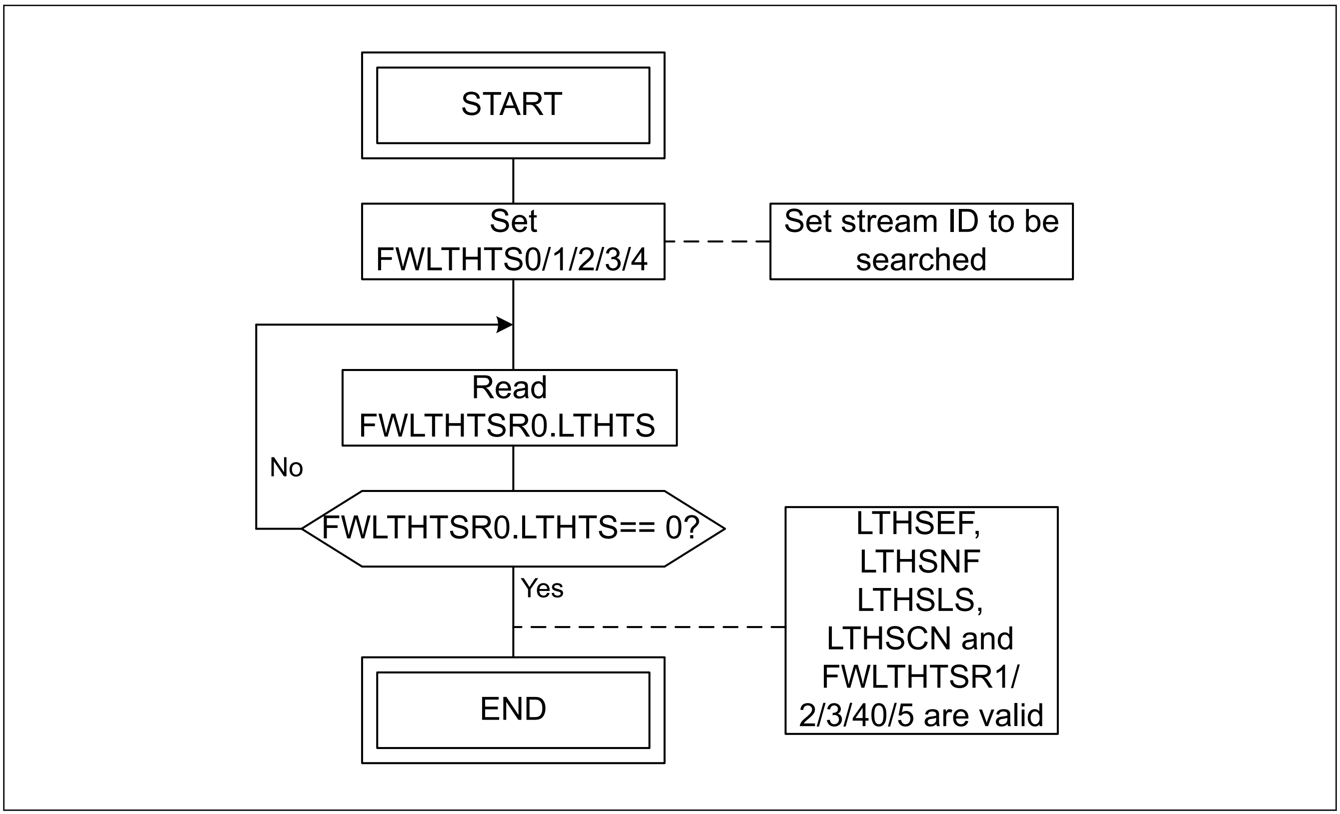


Figure 30.13 Layer 3 entry search flow

30.4.1.5.8 Layer 3 Entry Read Flow

Layer 3 entry read flow is described in Figure 30.14.

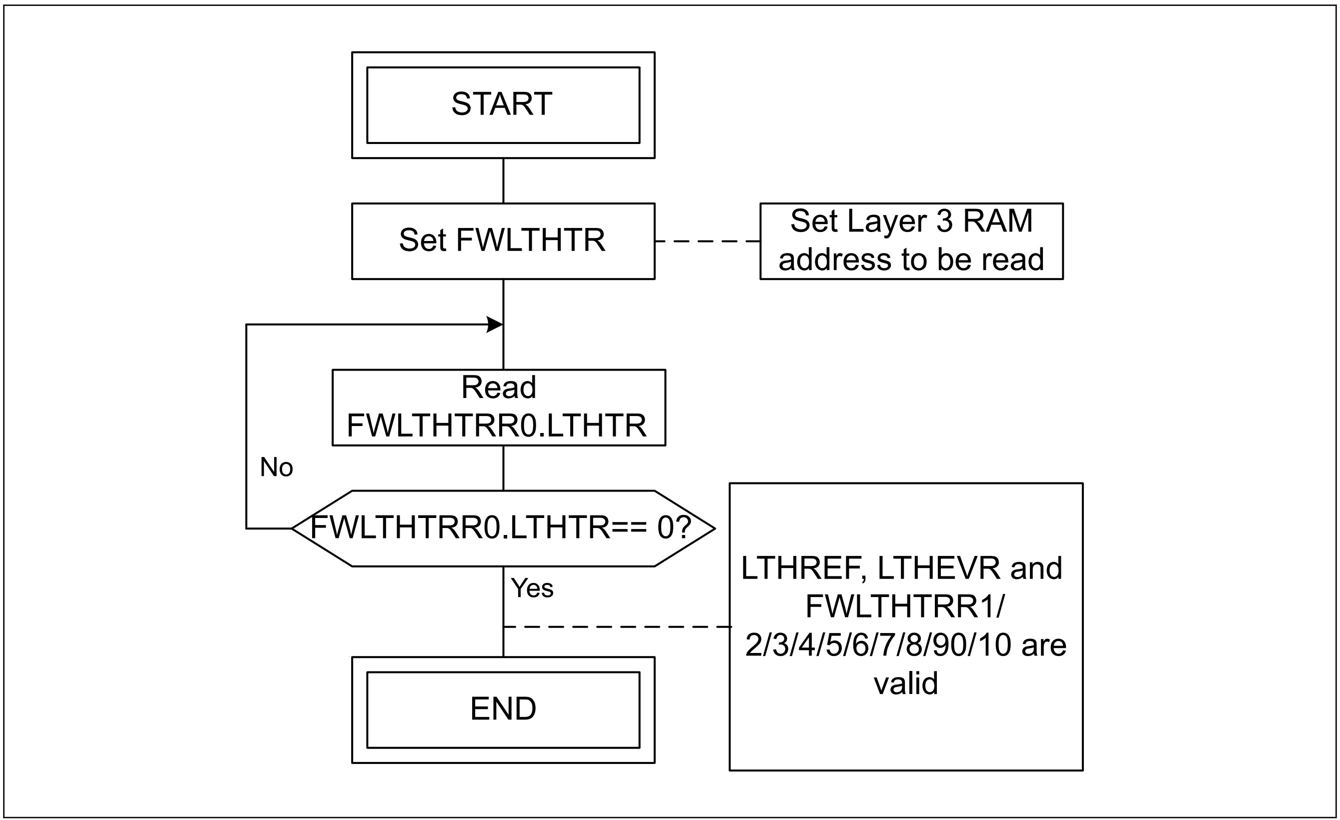


Figure 30.14 Layer 3 entry read flow

30.4.1.6 Layer 2 Forwarding Flows

30.4.1.6.1 MAC Table Reset Flow

Figure 30.15 describes the reset flow of MAC table.

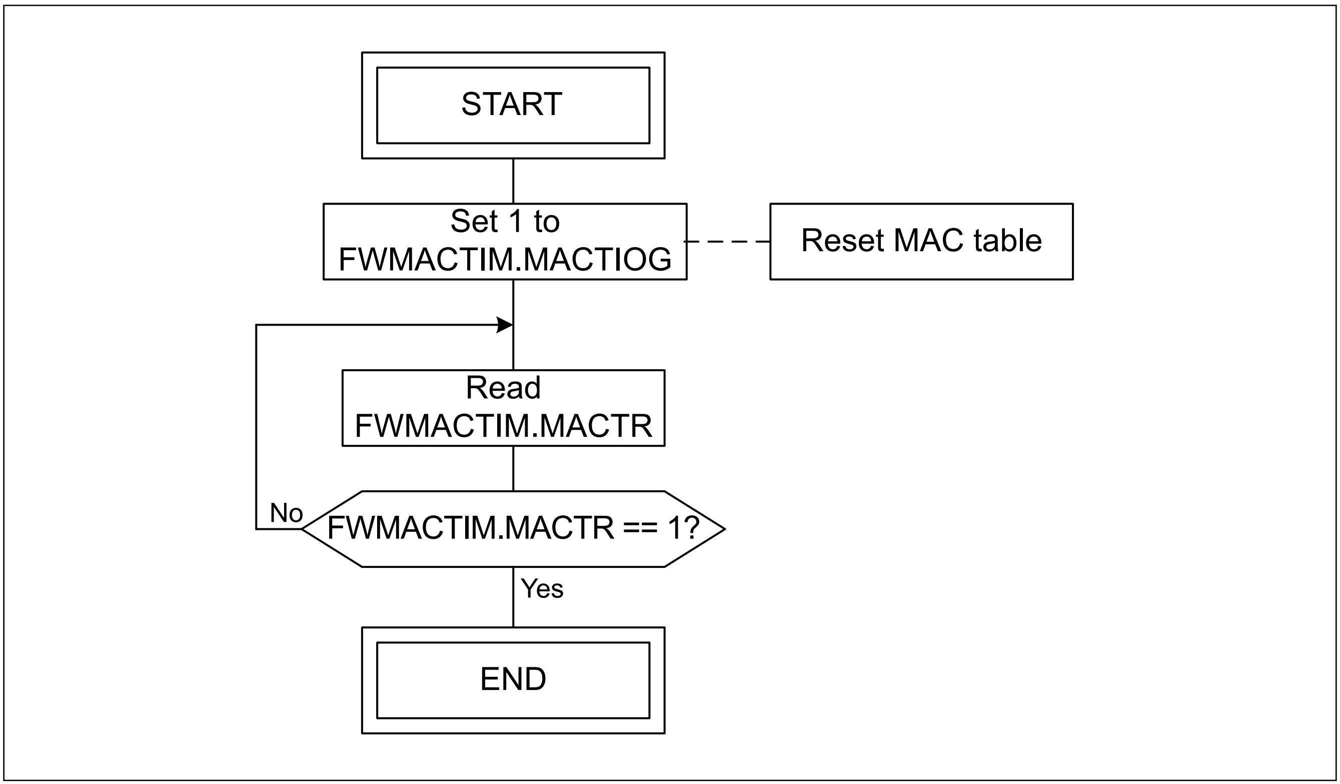


Figure 30.15 Reset flow of MAC table

30.4.1.6.2 VLAN Table Reset Flow

Figure 30.16 describes the reset flow of VLAN table.

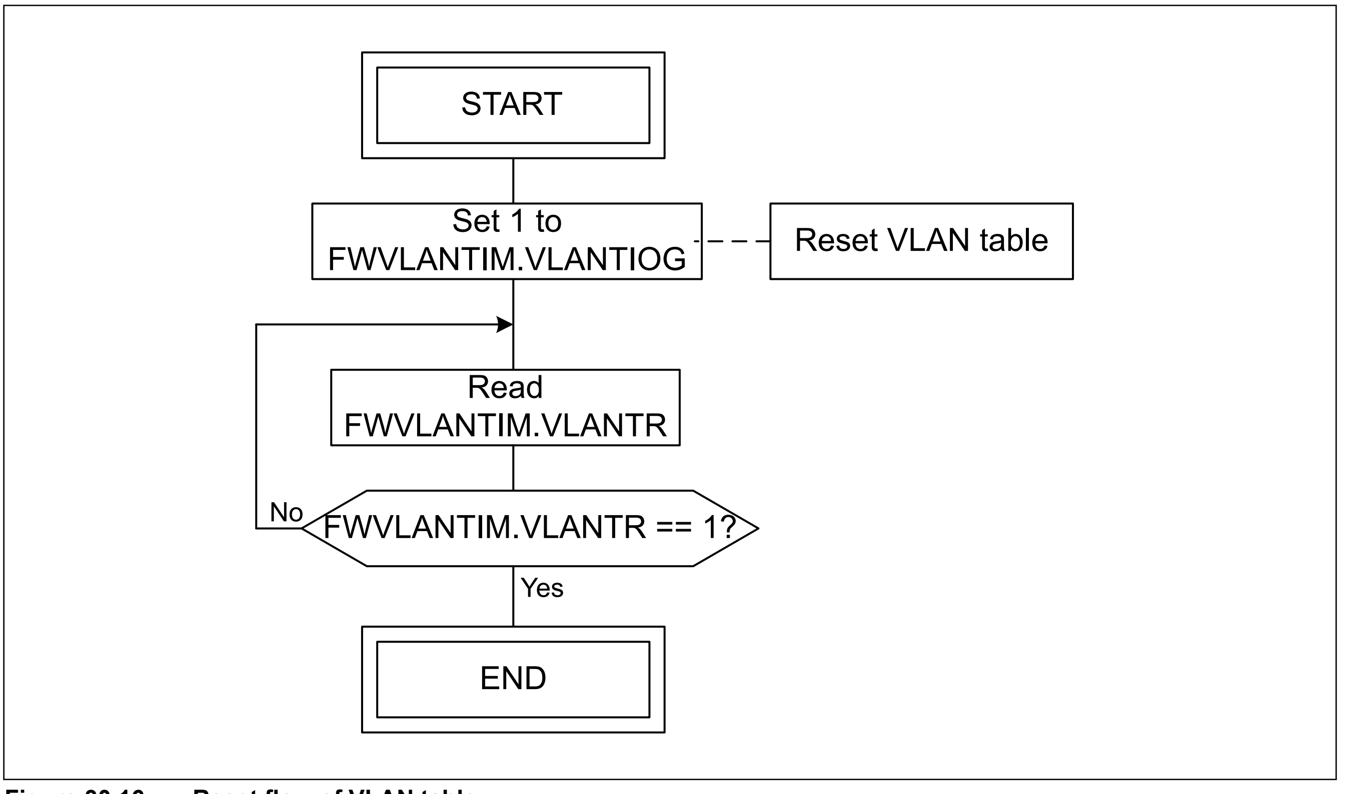


Figure 30.16 Reset flow of VLAN table

30.4.1.6.3 MAC Entry Learn Flow

MAC entry learn flow is described in Figure 30.17.

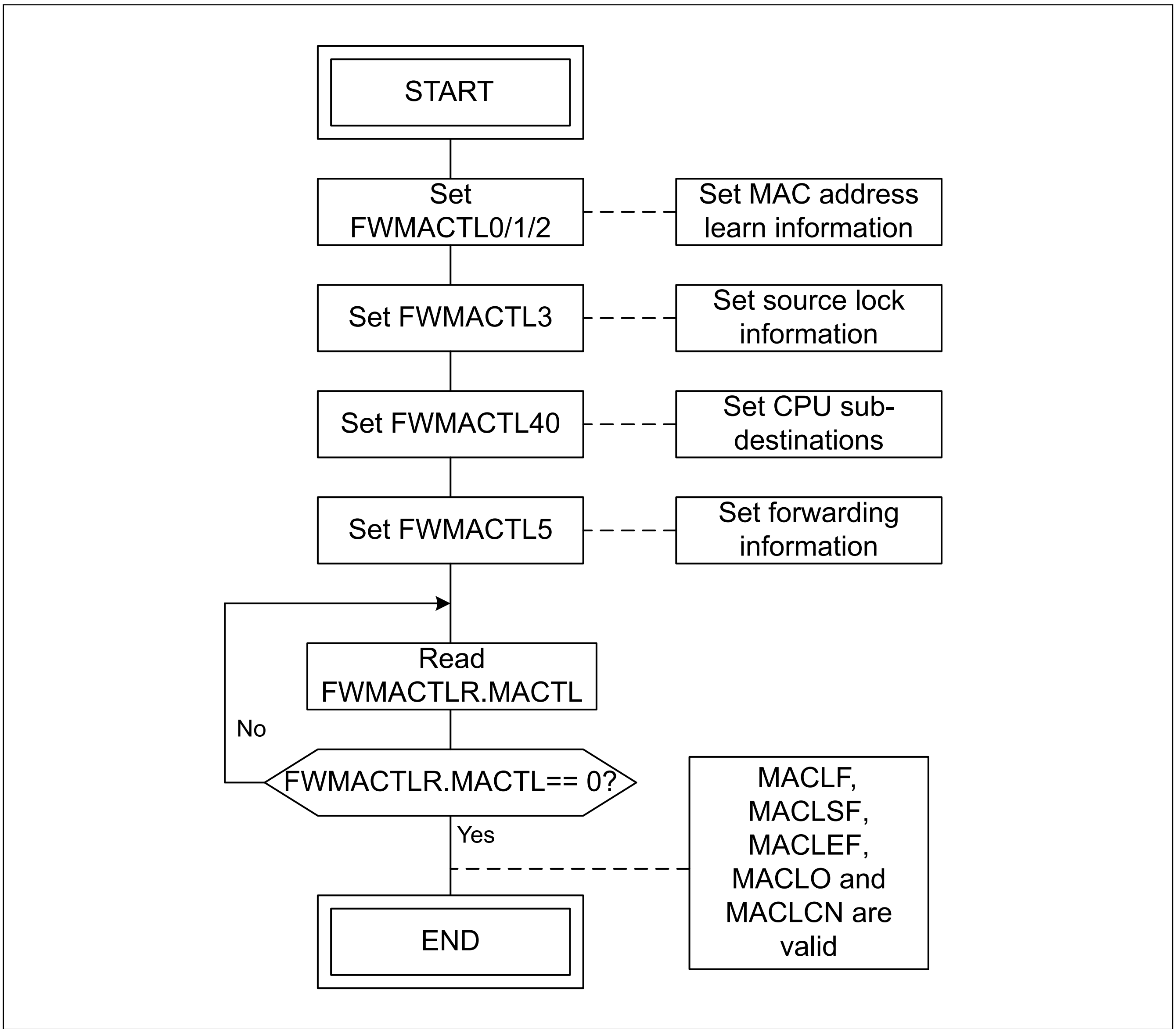


Figure 30.17 MAC entry learn flow

30.4.1.6.4 VLAN Entry Learn Flow

VLAN entry learn flow is described in Figure 30.18.

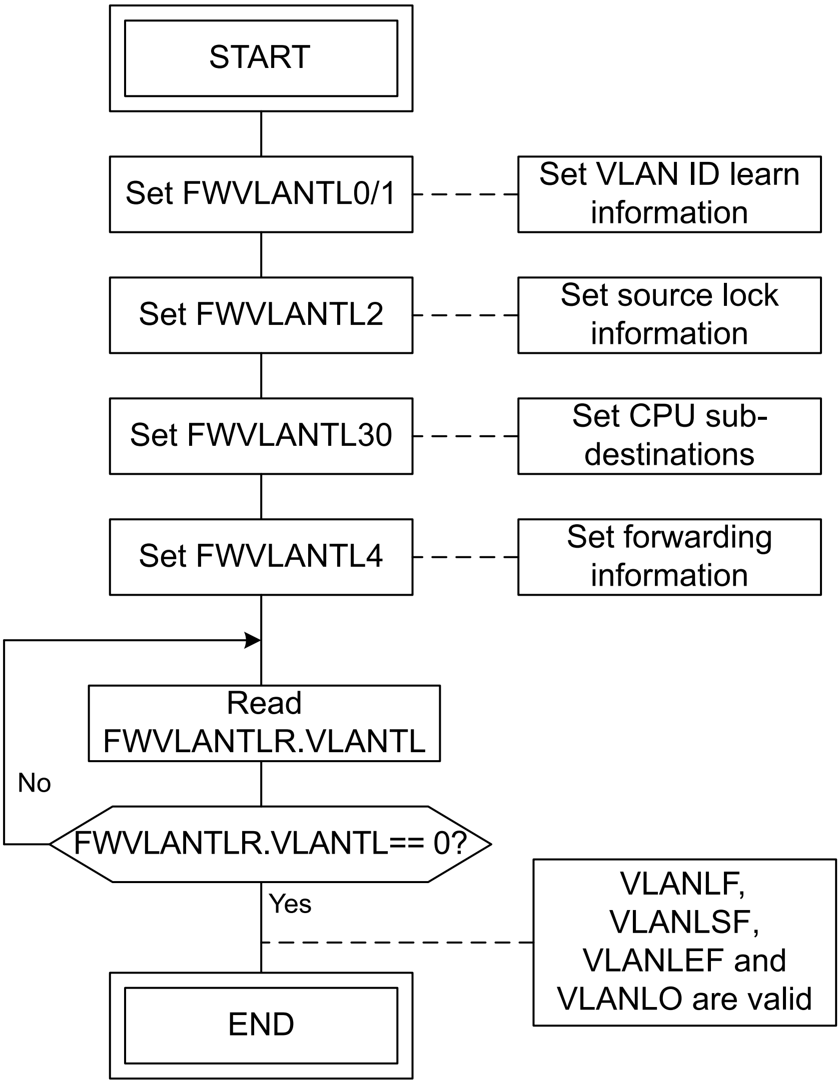


Figure 30.18 VLAN entry learn flow

30.4.1.6.5 MAC Entry Search Flow

MAC entry search flow is described in Figure 30.19.

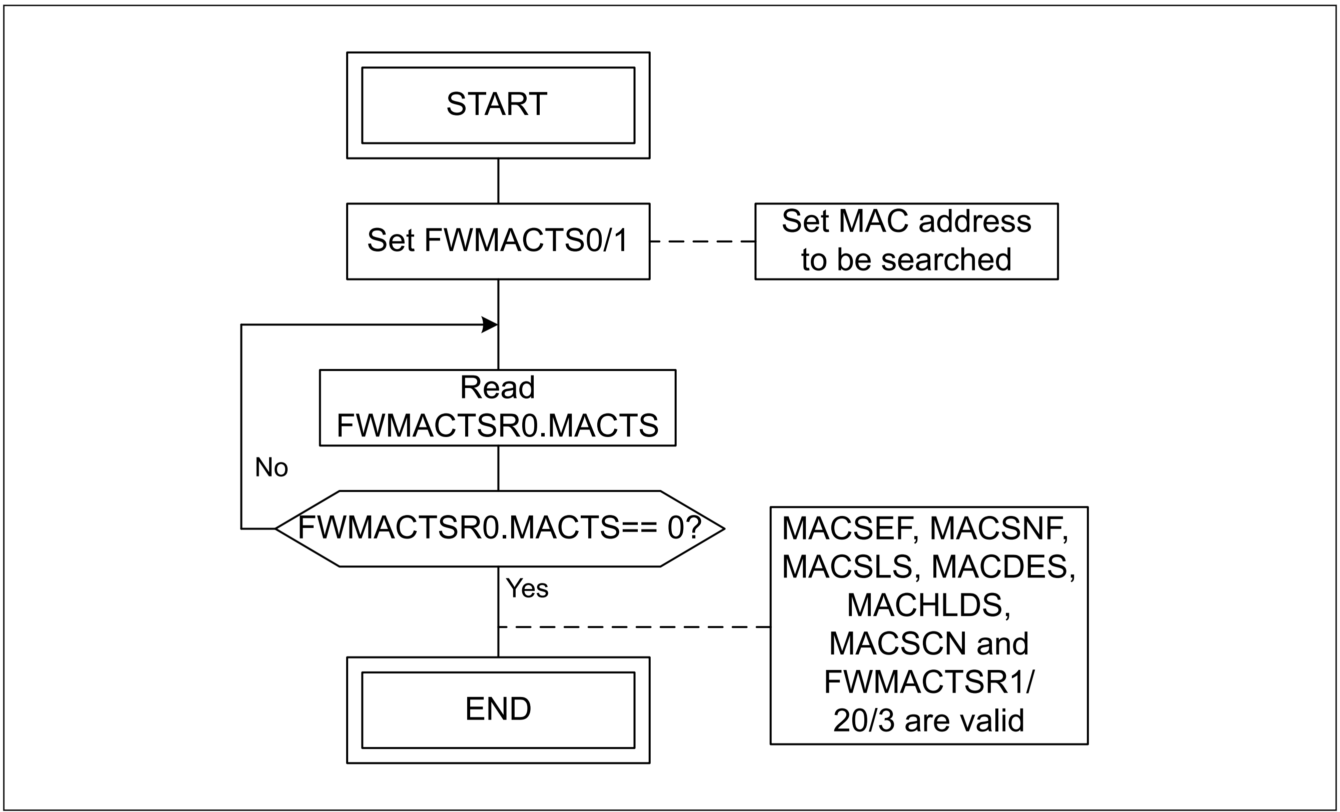


Figure 30.19 MAC entry search flow

30.4.1.6.6 VLAN Entry Search Flow

VLAN entry search flow is described in Figure 30.20.

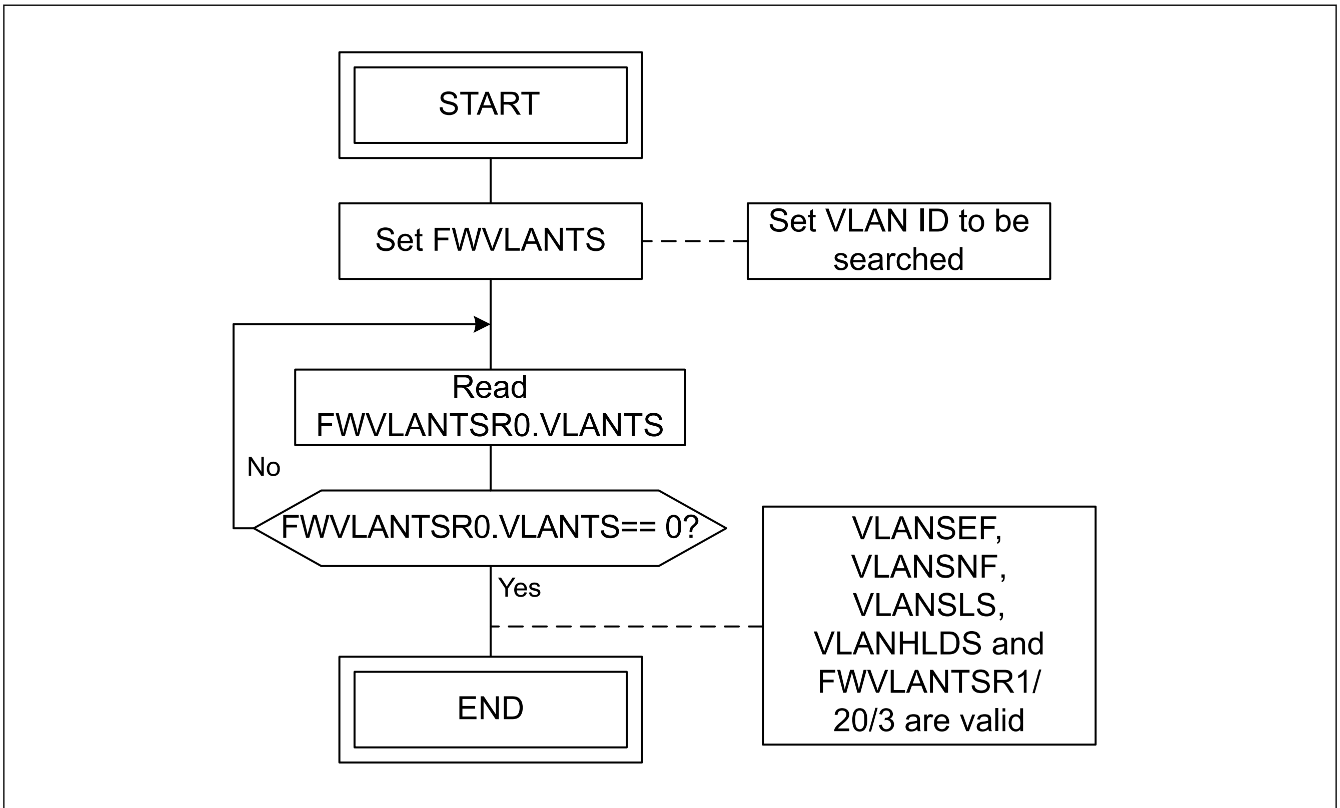


Figure 30.20 VLAN entry search flow

30.4.1.6.7 MAC Entry Read Flow

MAC entry read flow is described in Figure 30.21.

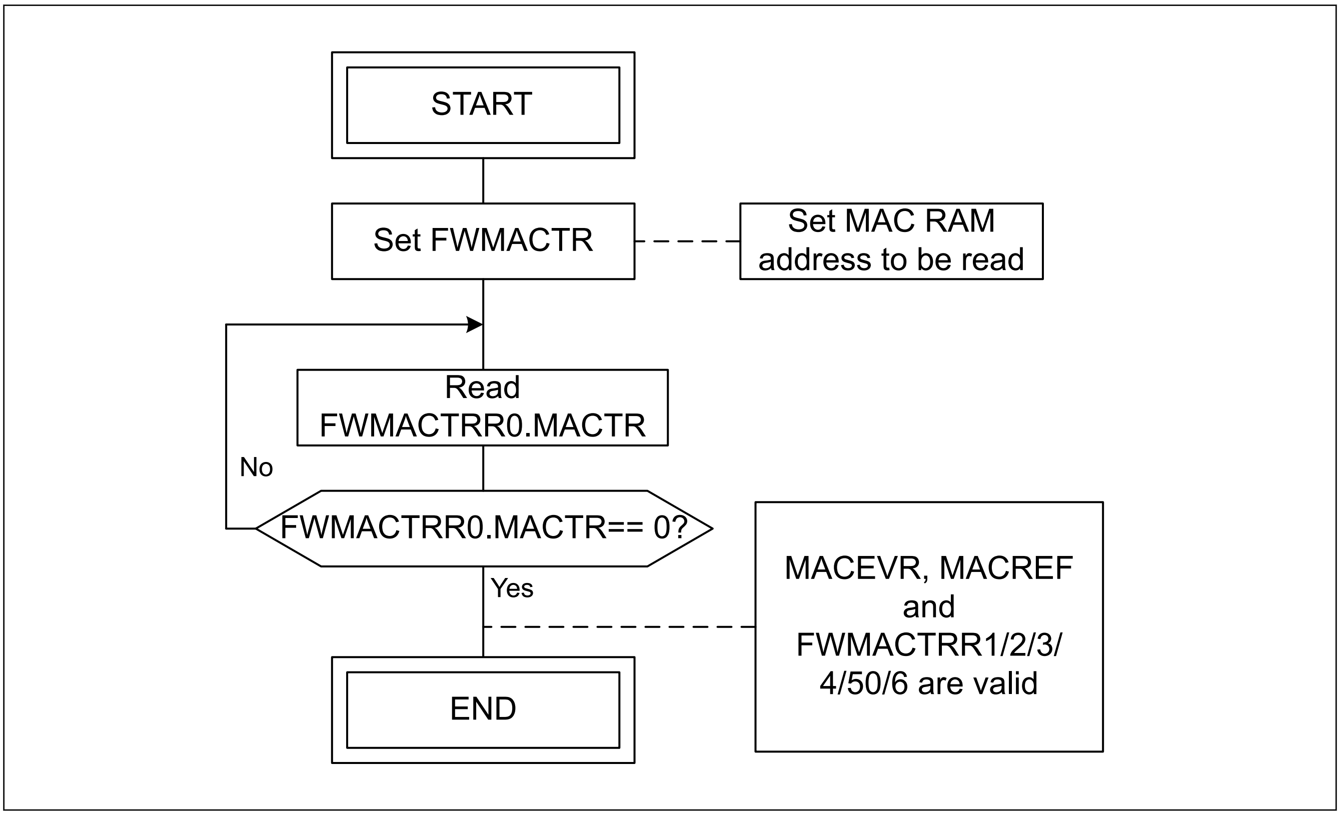


Figure 30.21 MAC entry read flow

30.4.1.7 Port based forwarding flows

30.4.1.7.1 Port i Port-based Setting Flow

Port i port-based setting flow is described in Figure 30.22.

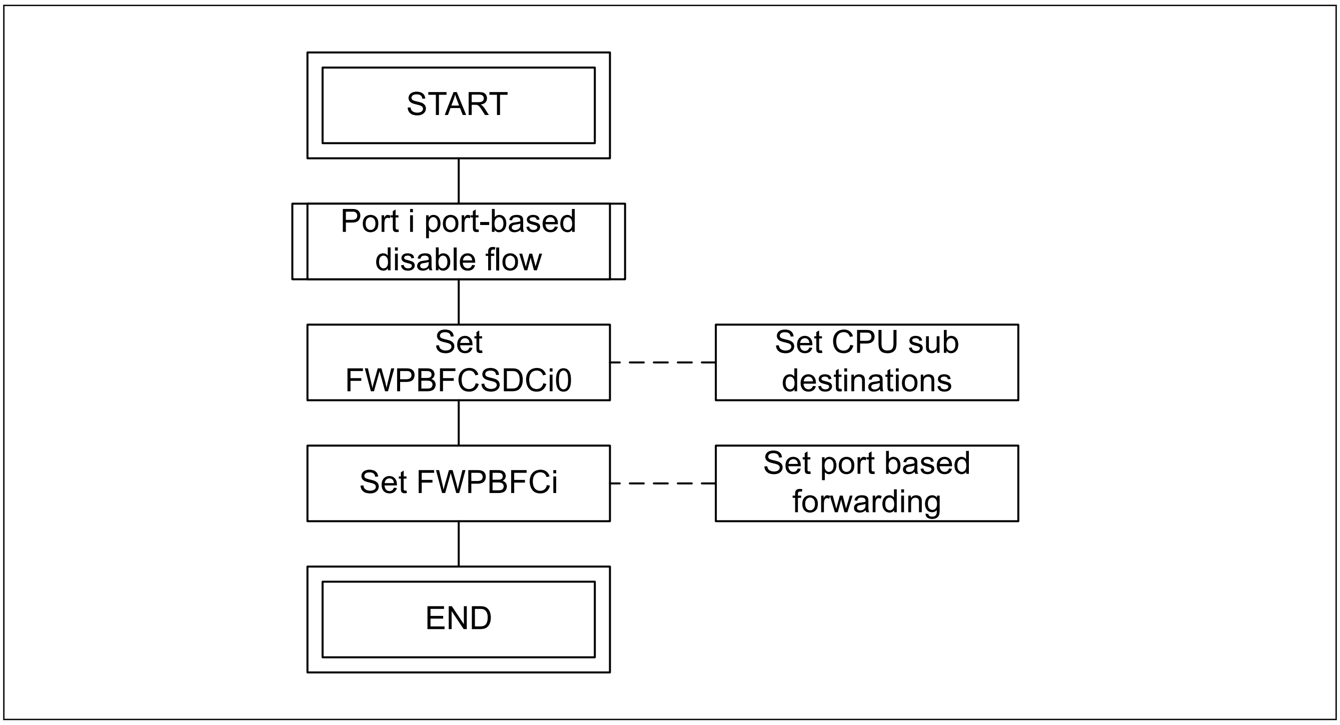


Figure 30.22 Port i port-based setting flow

30.4.1.7.2 Port i Port-based Disable Flow

Port i port-based disable flow is described in Figure 30.23.

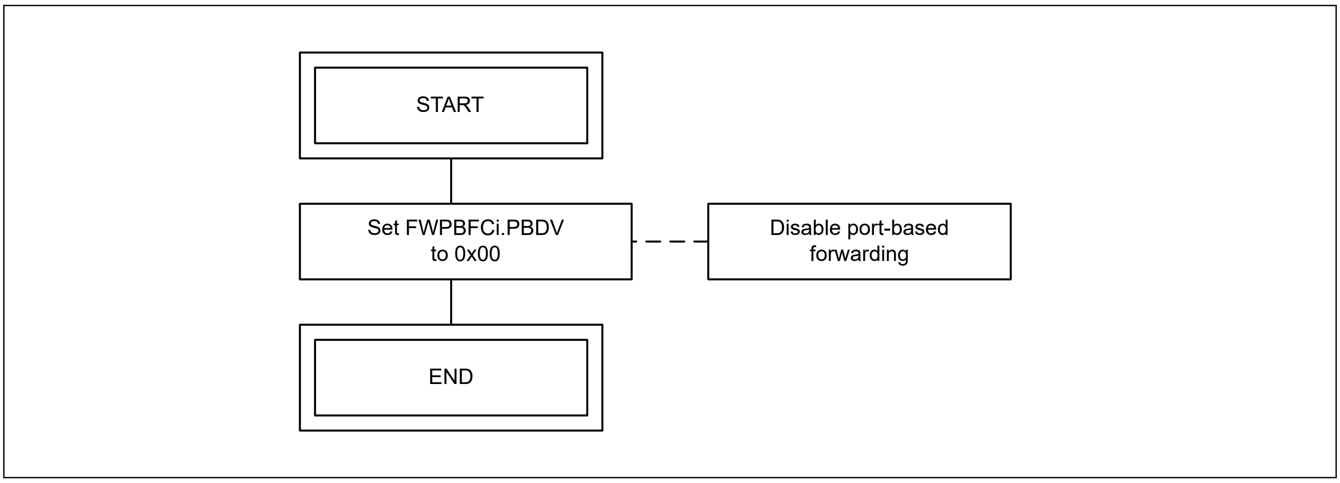


Figure 30.23 Port i port-based disable flow

30.4.1.8 Layer 2/Layer 3 Update Flows

30.4.1.8.1 Layer 2/Layer 3 Update Table Reset Flow

Figure 30.24 describes the reset flow of Layer 2/Layer 3 Update table.

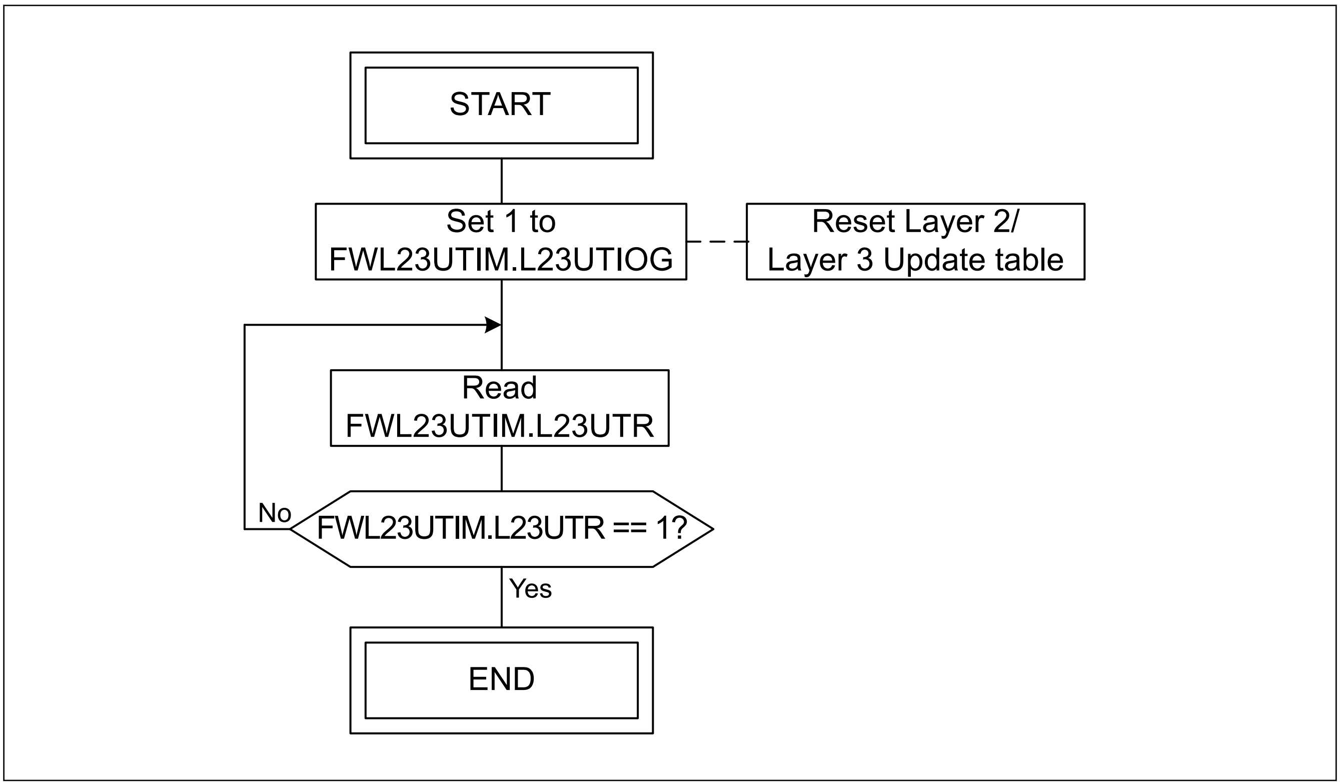


Figure 30.24 Reset flow of Layer 2/Layer 3 Update table

30.4.1.8.2 Layer 2/Layer 3 Update Rule Learn Flow

Figure 30.25 describes the learn flow of Layer 2/Layer 3 update rule.

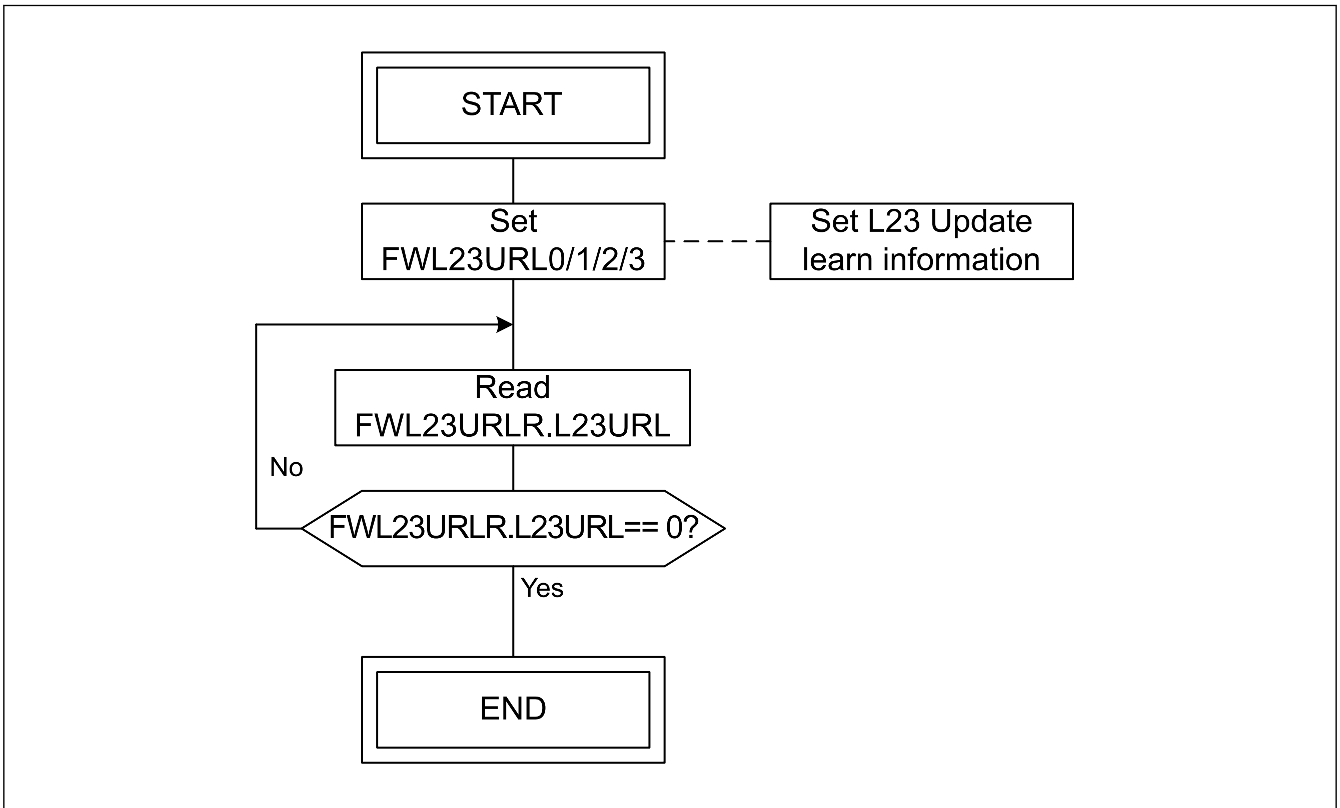


Figure 30.25 Learn flow of Layer 2/Layer 3 update rule

30.4.1.8.3 Layer 2/Layer 3 Update Rule Read Flow

Figure 30.26 describes the read flow of Layer 2/Layer 3 Update rule.

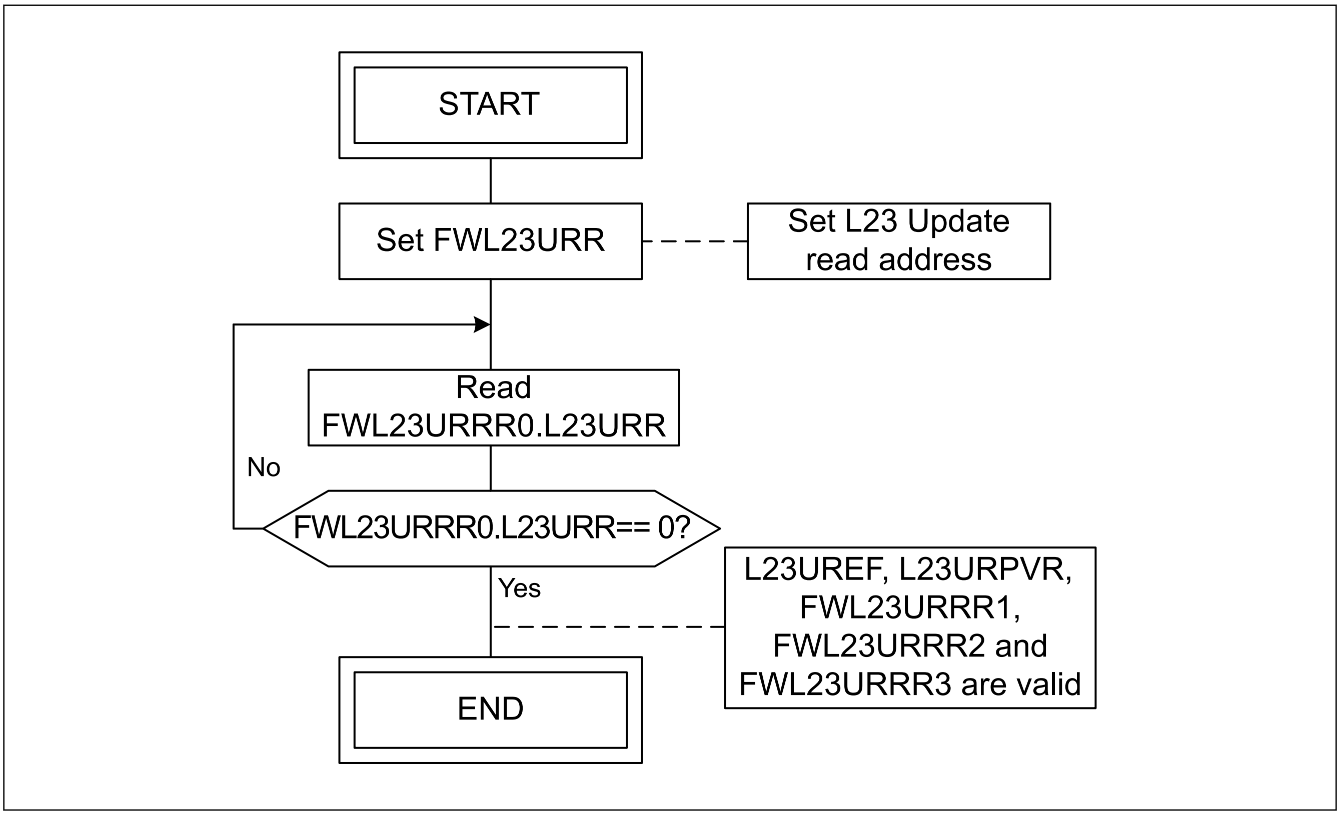


Figure 30.26 Read flow of Layer 2/Layer 3 Update rule

30.4.1.9 PSFP Flows

30.4.1.9.1 MSDU Filter i Setting Flow

Figure 30.27 describes the setting flow of MSDU filter i.

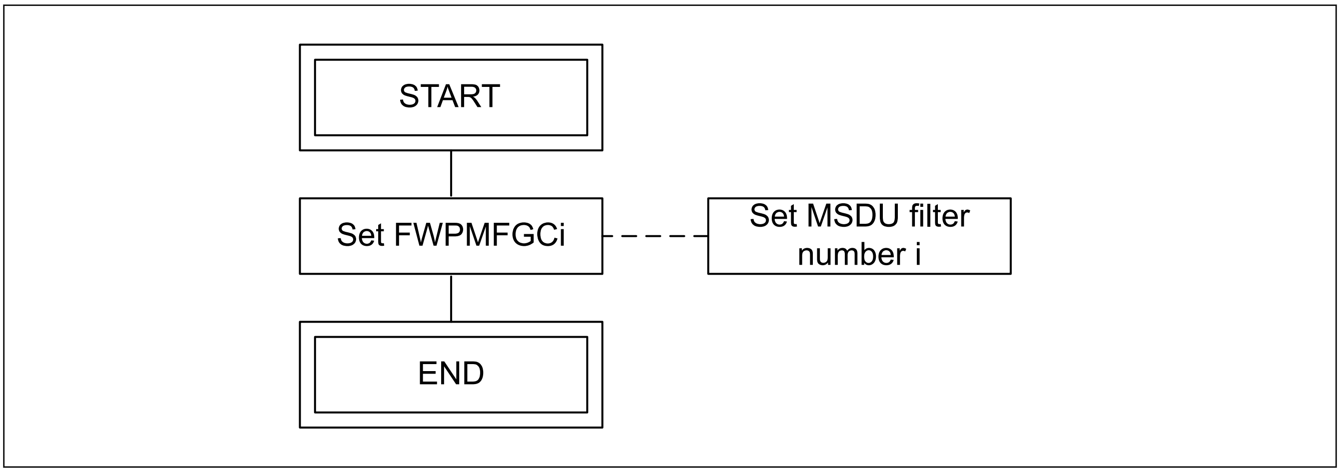


Figure 30.27 Setting flow of MSDU filter i

30.4.1.9.2 Meter i Setting Flow

Meter i setting flow is described in Figure 30.28.

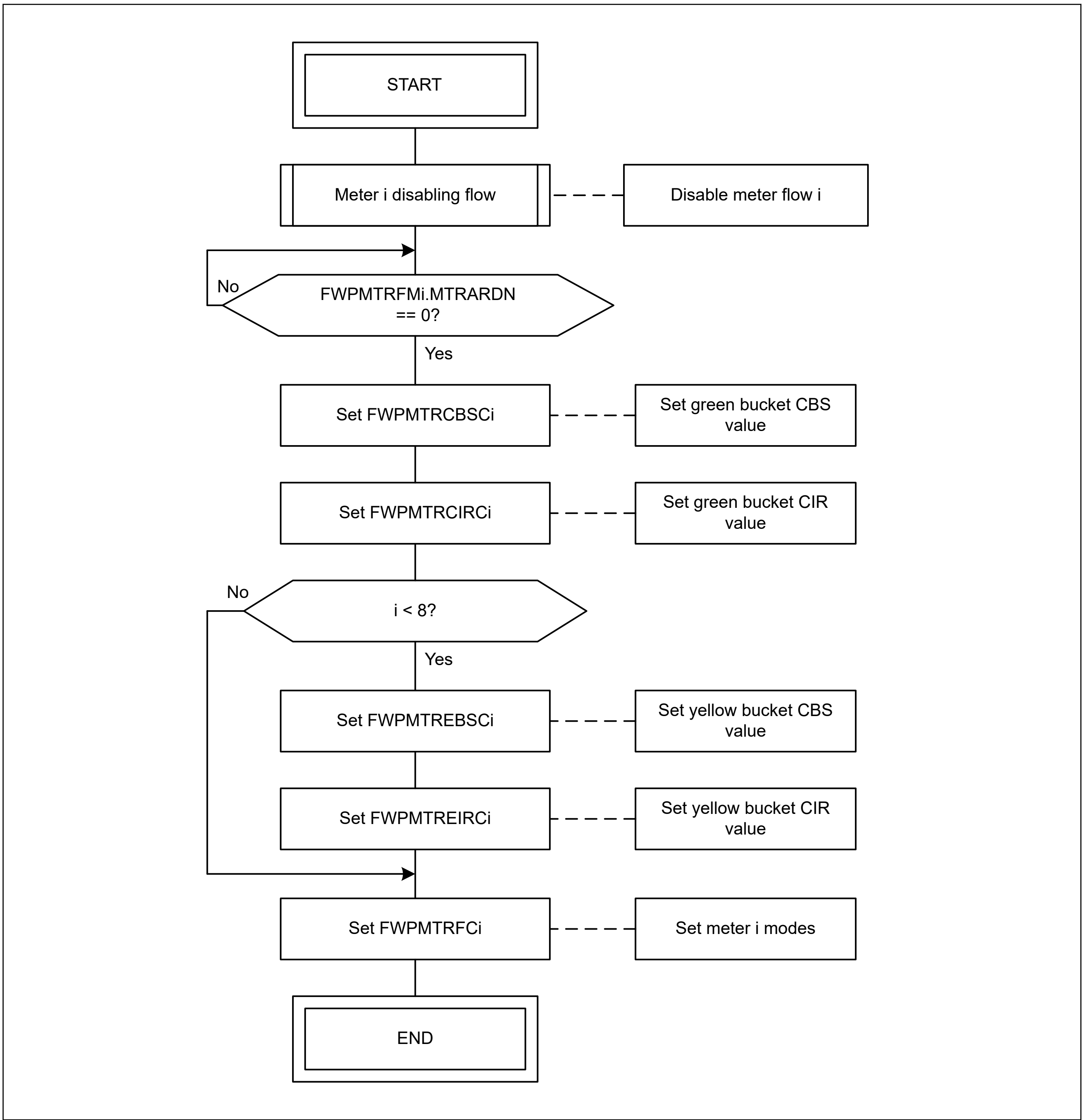


Figure 30.28 Meter i setting flow

30.4.1.9.3 Meter i Disabling Flow

Figure 30.29 describes the disabling flow of meter i.

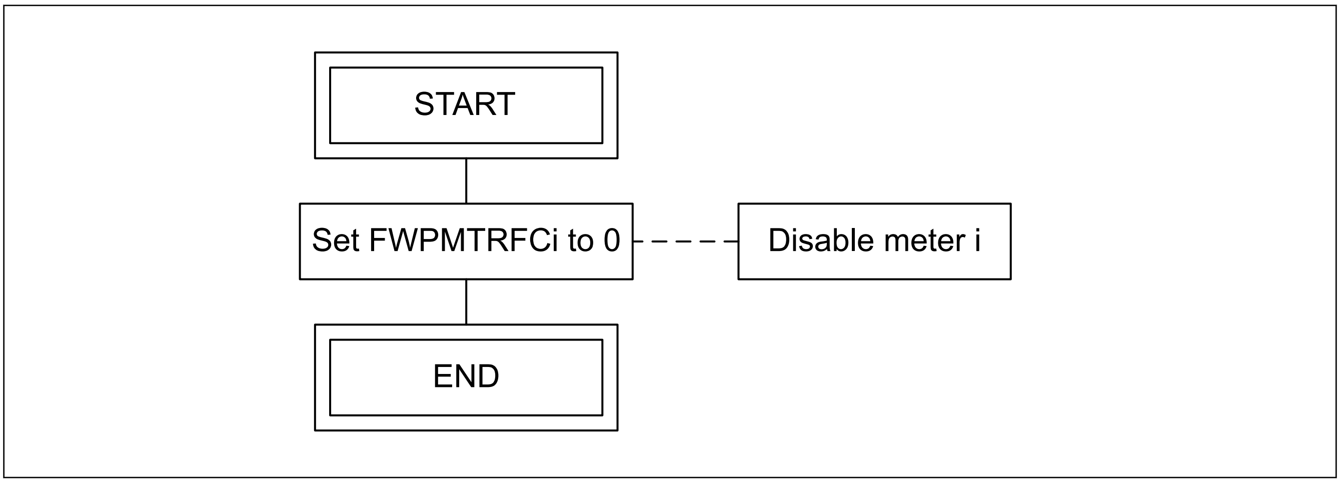


Figure 30.29 Disabling flow of meter i

30.4.1.10 FRER Flows

30.4.1.10.1 FRER Table Reset Flow

Figure 30.30 describes the reset flow of FRER table.

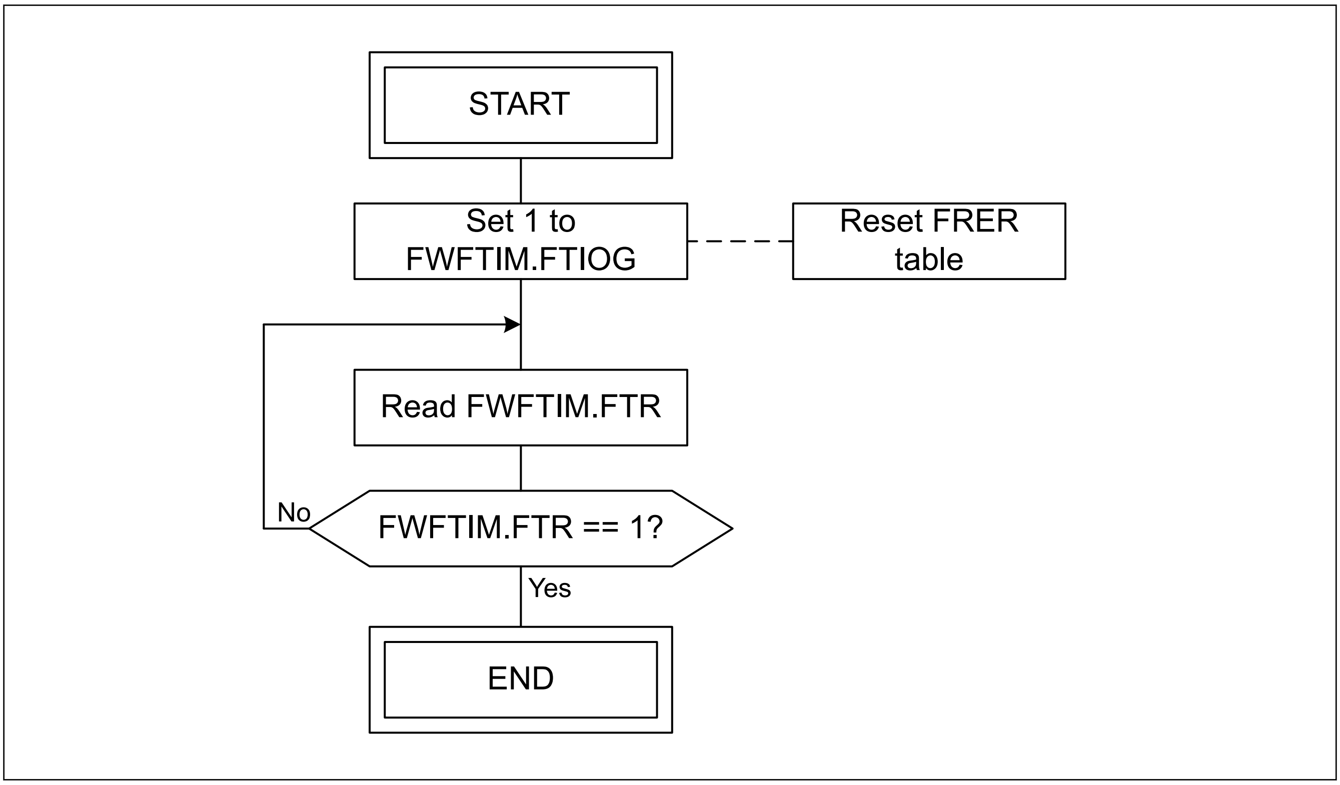


Figure 30.30 Reset flow of FRER table

30.4.1.10.2 FRER Entry i Learn Flow

Figure 30.31 describes the learn flow of FRER entry i.

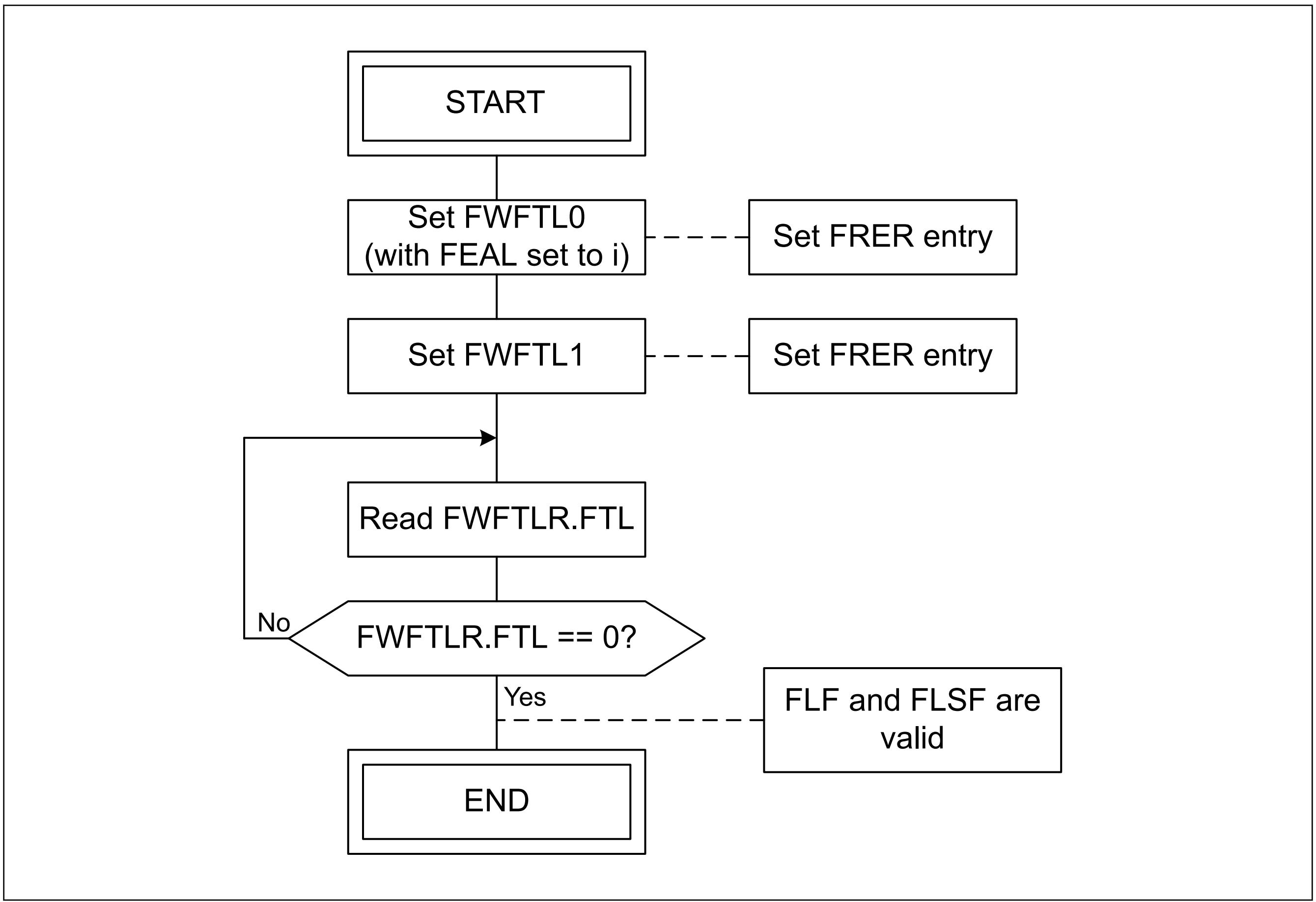


Figure 30.31 Learn flow of FRER entry i

30.4.1.10.3 FRER Entry i Read Flow

Figure 30.32 describes the read flow of FRER entry i.

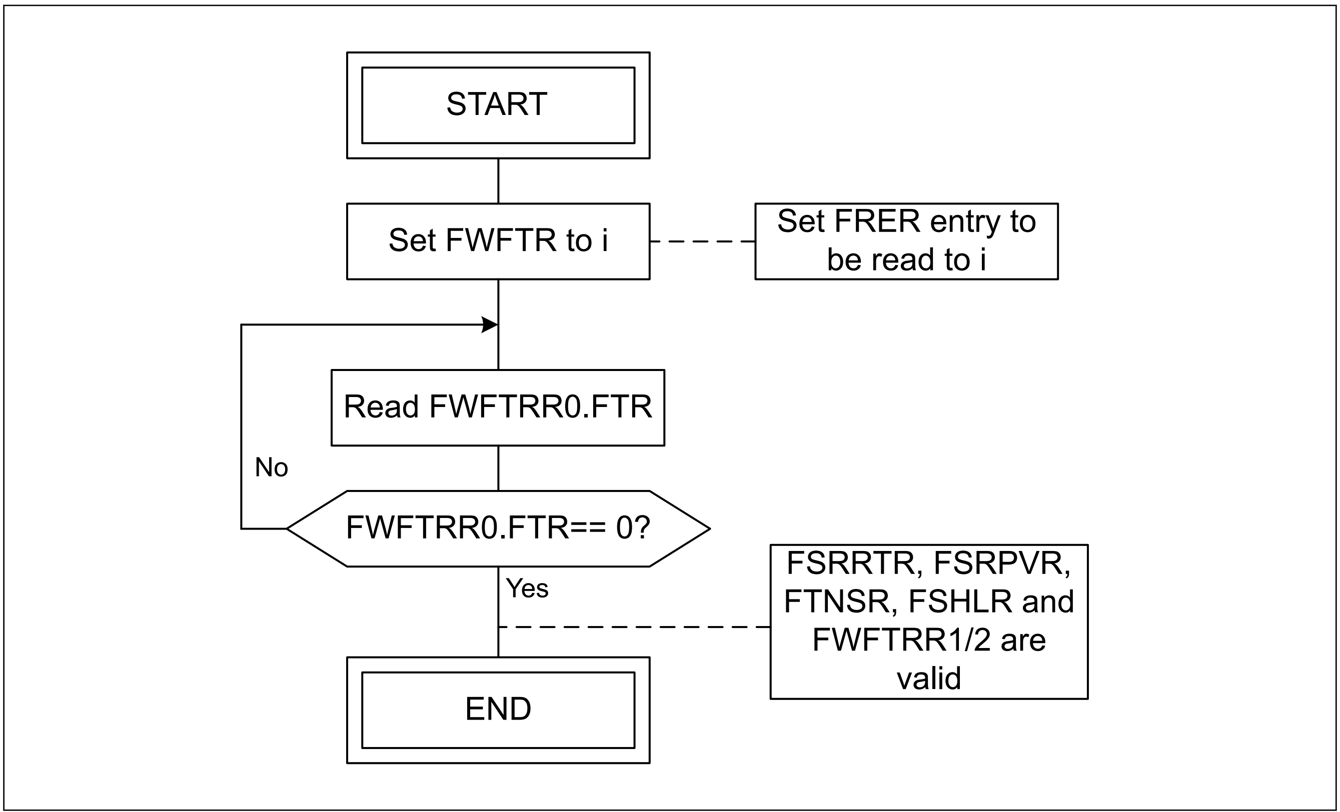


Figure 30.32 Read flow of FRER entry i

30.4.1.11 Called Software Flows

The flows described in this section can only be called from other flows (as shown in the "flow link" box in Figure 30.2) and cannot be used alone.

30.4.1.11.1 Full Setting Flow

Full setting flow is described in Figure 30.33.

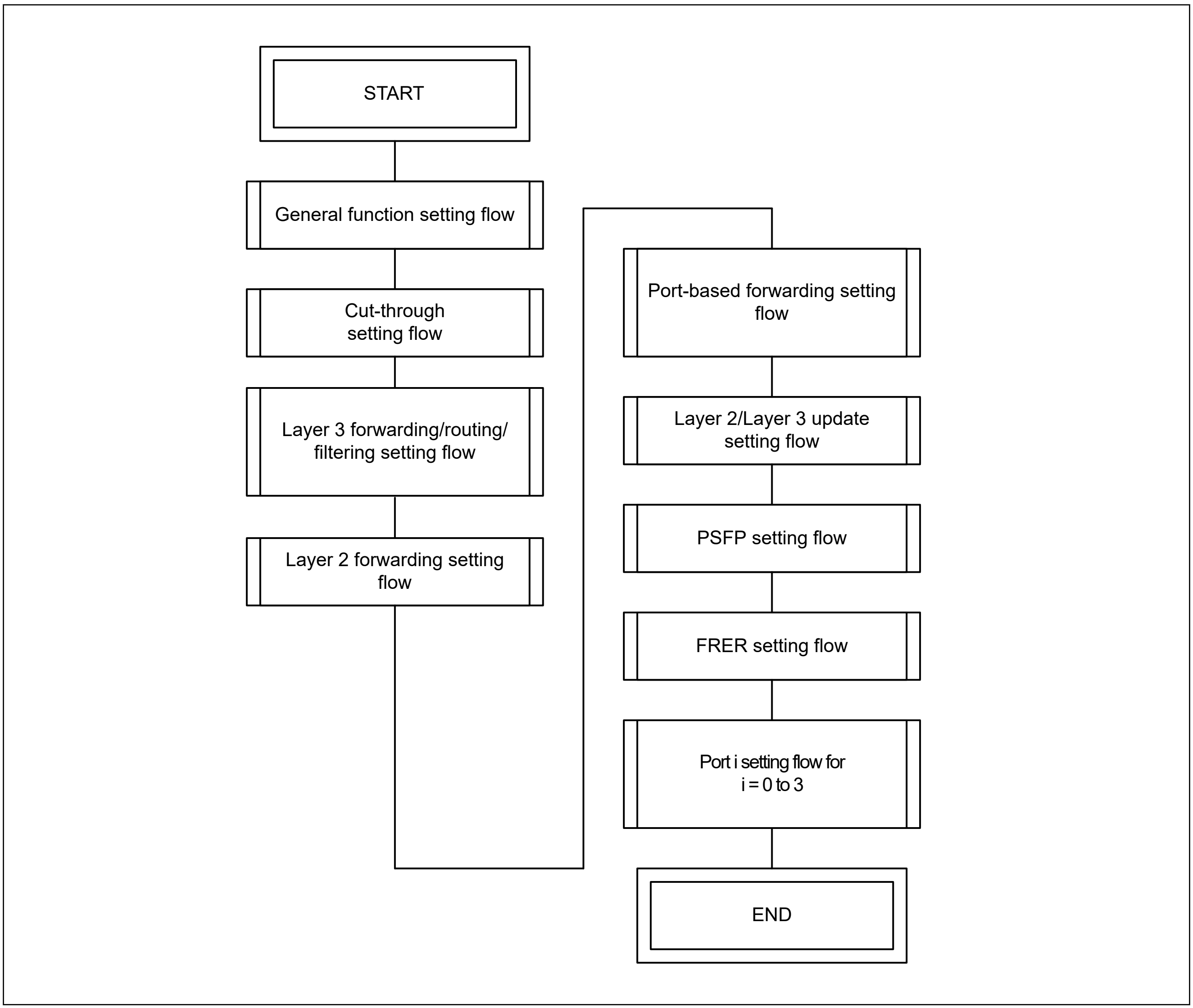


Figure 30.33 Full setting flow

30.4.1.11.2 General Function Setting Flow

Figure 30.34 decribes the setting flow of general function.

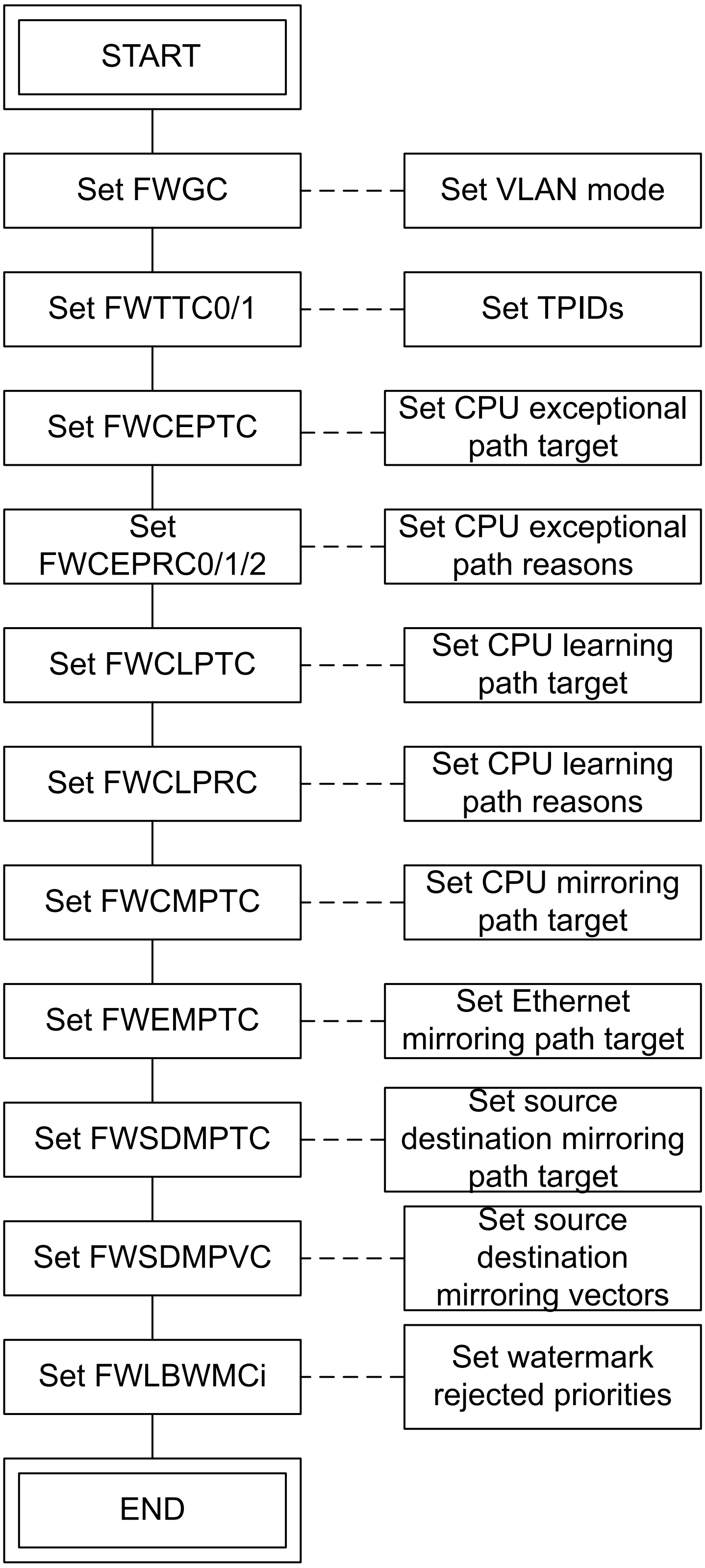


Figure 30.34 Setting flow of general function

30.4.1.11.3 Cut-through Setting Flow

Figure 30.35 describes the setting flow of cut-through.

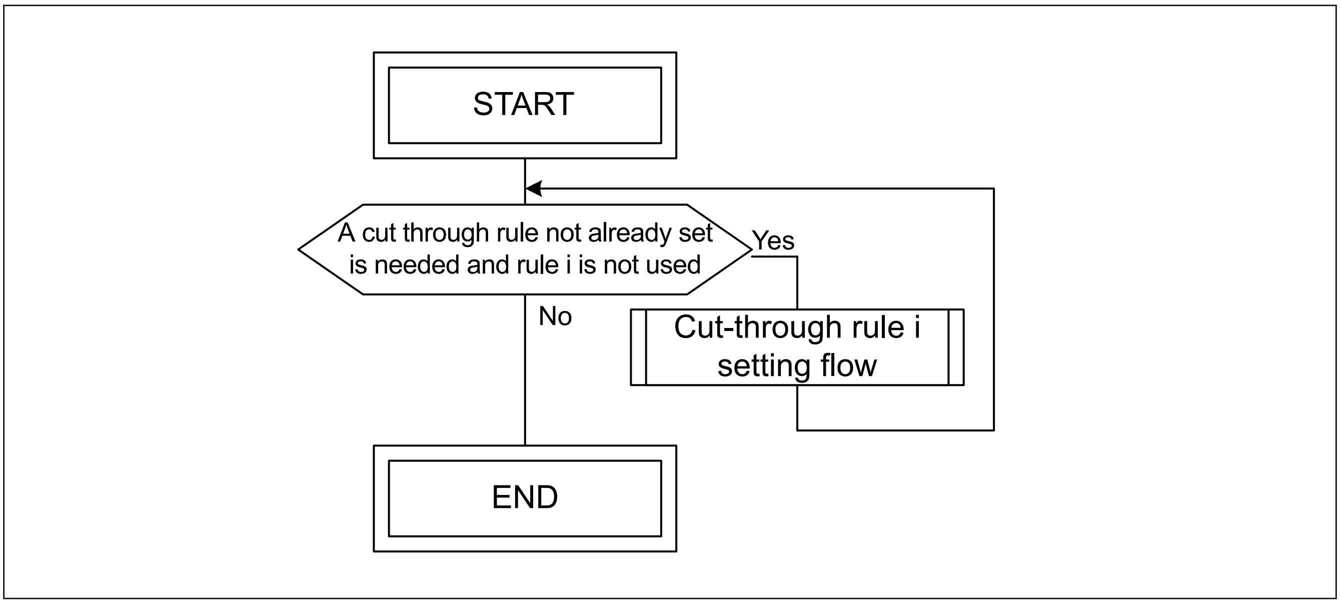


Figure 30.35 Setting flow of cut-through

30.4.1.11.4 Two-byte Filter i Setting Flow

Figure 30.36 describes the setting flow of two-byte filter i.

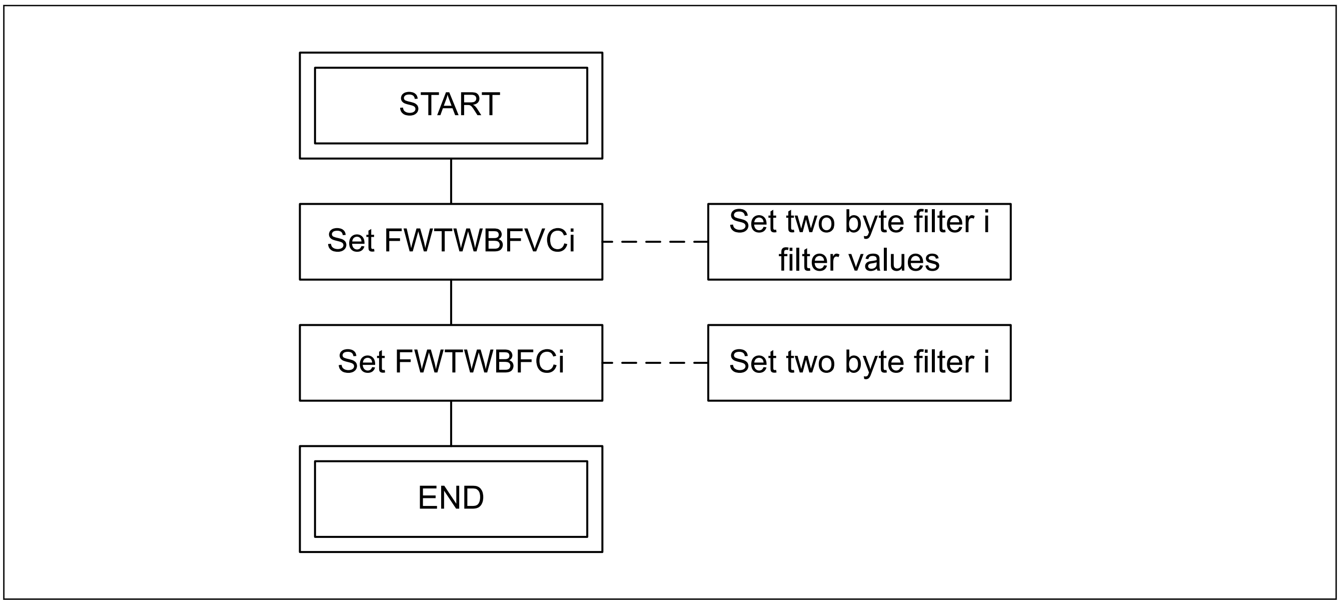


Figure 30.36 Setting flow of two-byte filter i

30.4.1.11.5 Three-byte Filter i Setting Flow

Figure 30.37 describes the setting flow of three-byte filter i.

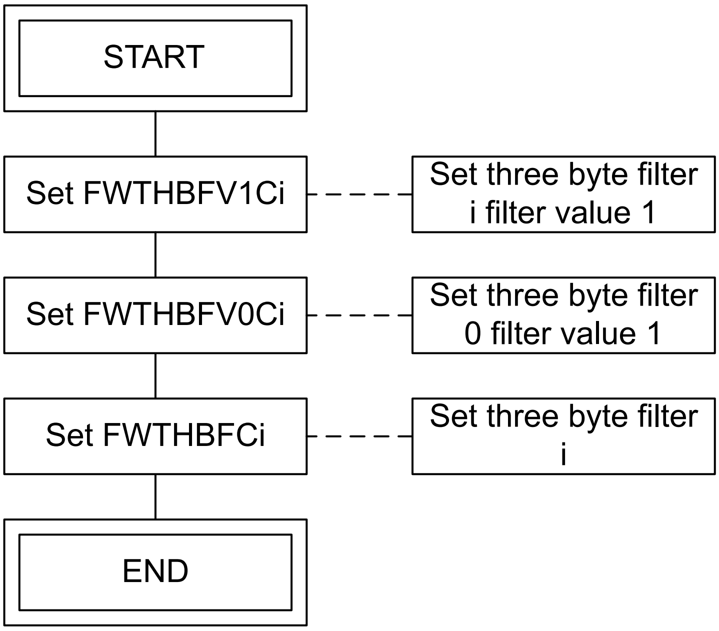


Figure 30.37 Setting flow of three-byte filter i

30.4.1.11.6 Four-byte Filter i Setting Flow

Figure 30.38 describes the setting flow of four-byte filter i.

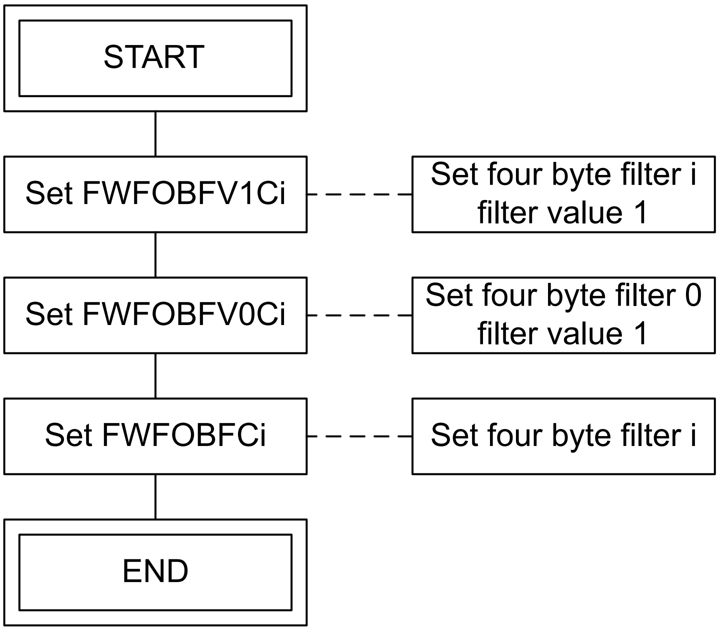


Figure 30.38 Setting flow of four-byte filter i

30.4.1.11.7 Range Filter i Setting Flow

Figure 30.39 describes the setting flow of range filter i.

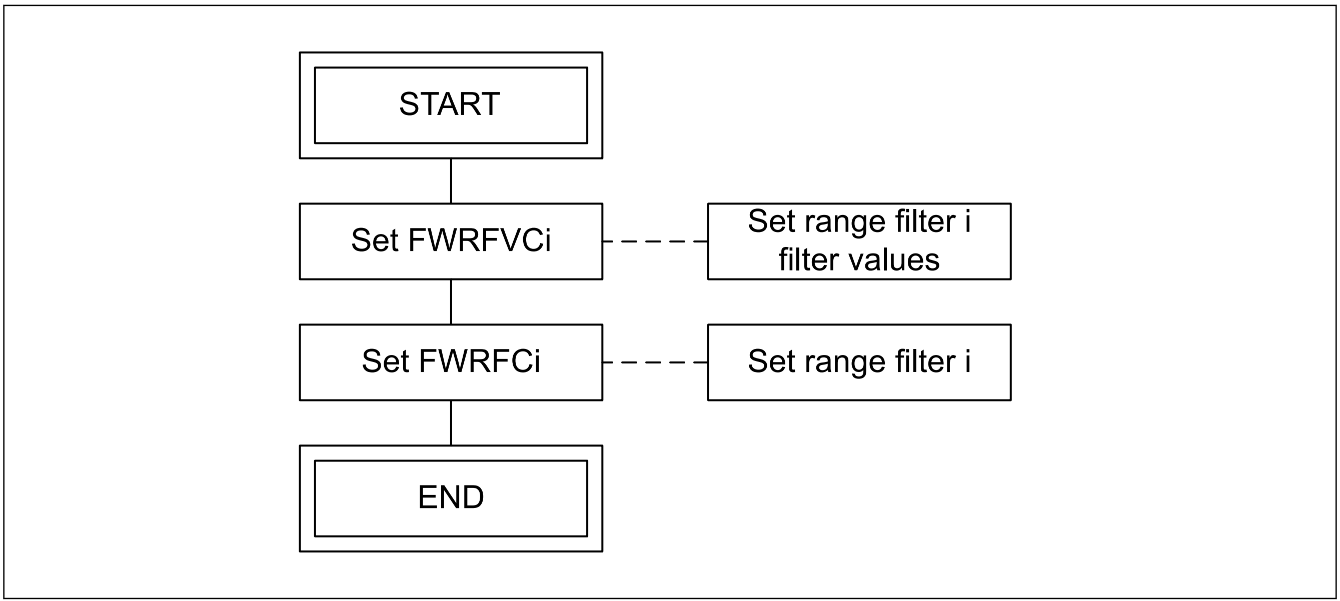


Figure 30.39 Setting flow of range filter i

30.4.1.11.8 IPv4/IPv6/Layer 2 Stream Filter Setting Flow

Figure 30.40 describes the setting flow of IPv4/IPv6/Layer 2 stream filter.

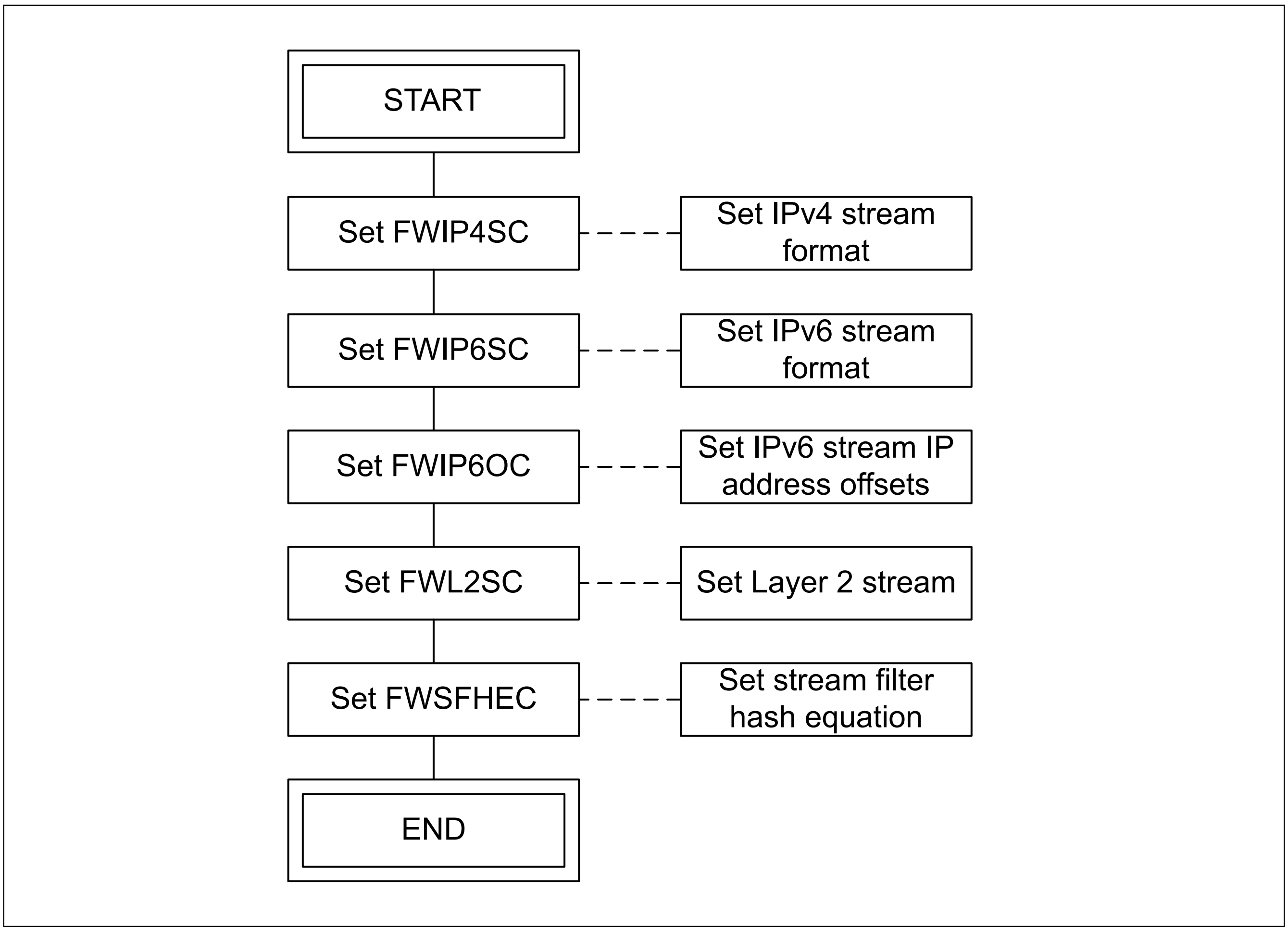


Figure 30.40 Setting flow of IPv4/IPv6/Layer 2 stream filter

30.4.1.11.9 Layer 3 Forwarding/Routing/Filtering Setting Flow

Figure 30.41 describes the setting flow of Layer 3 forwarding/routing/filtering.

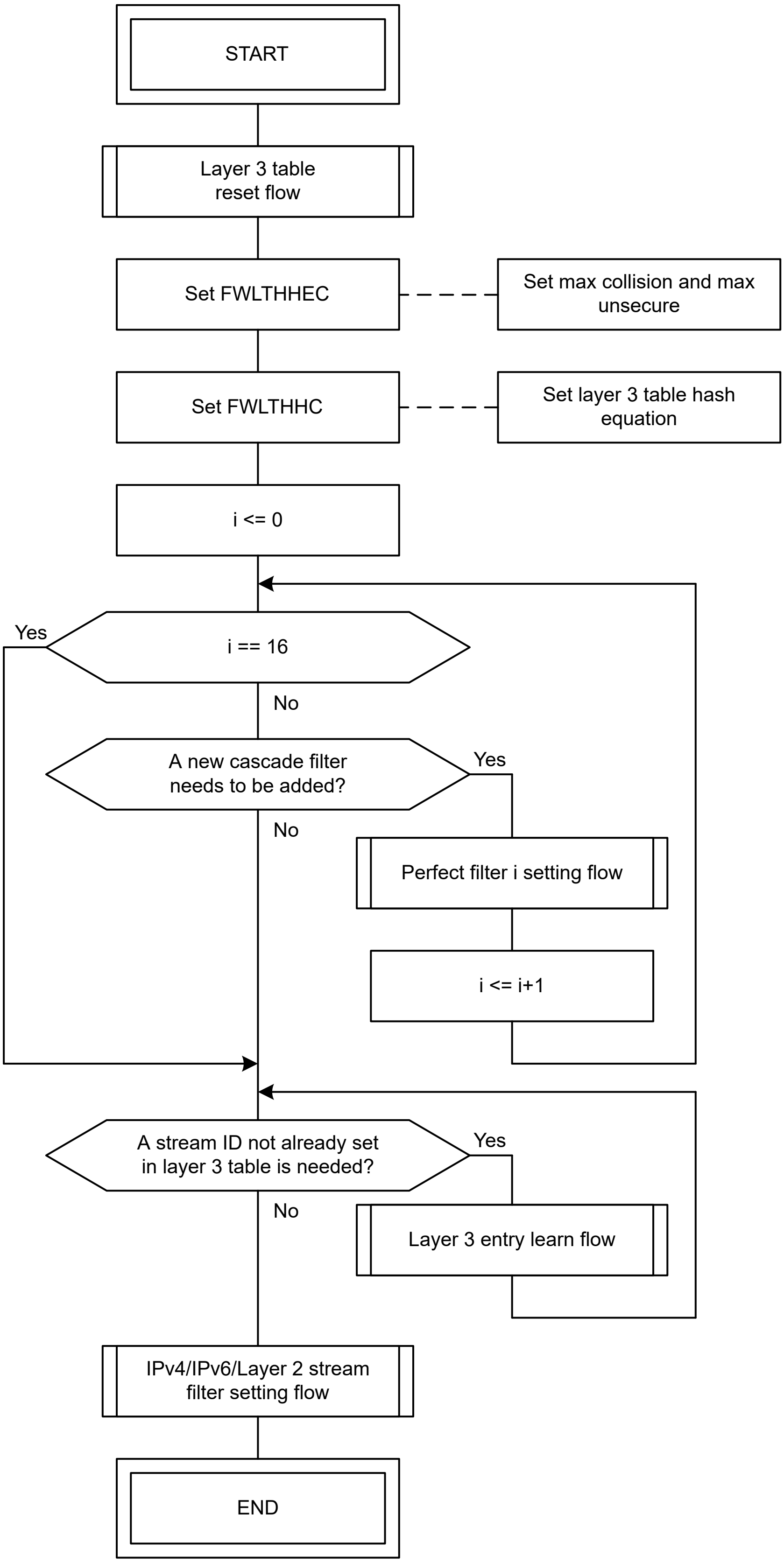


Figure 30.41 Setting flow of Layer 3 forwarding/routing/filtering

30.4.1.11.10 Layer 2 Forwarding Setting Flow

Figure 30.42 describes the setting flow of Layer 2 forwarding.

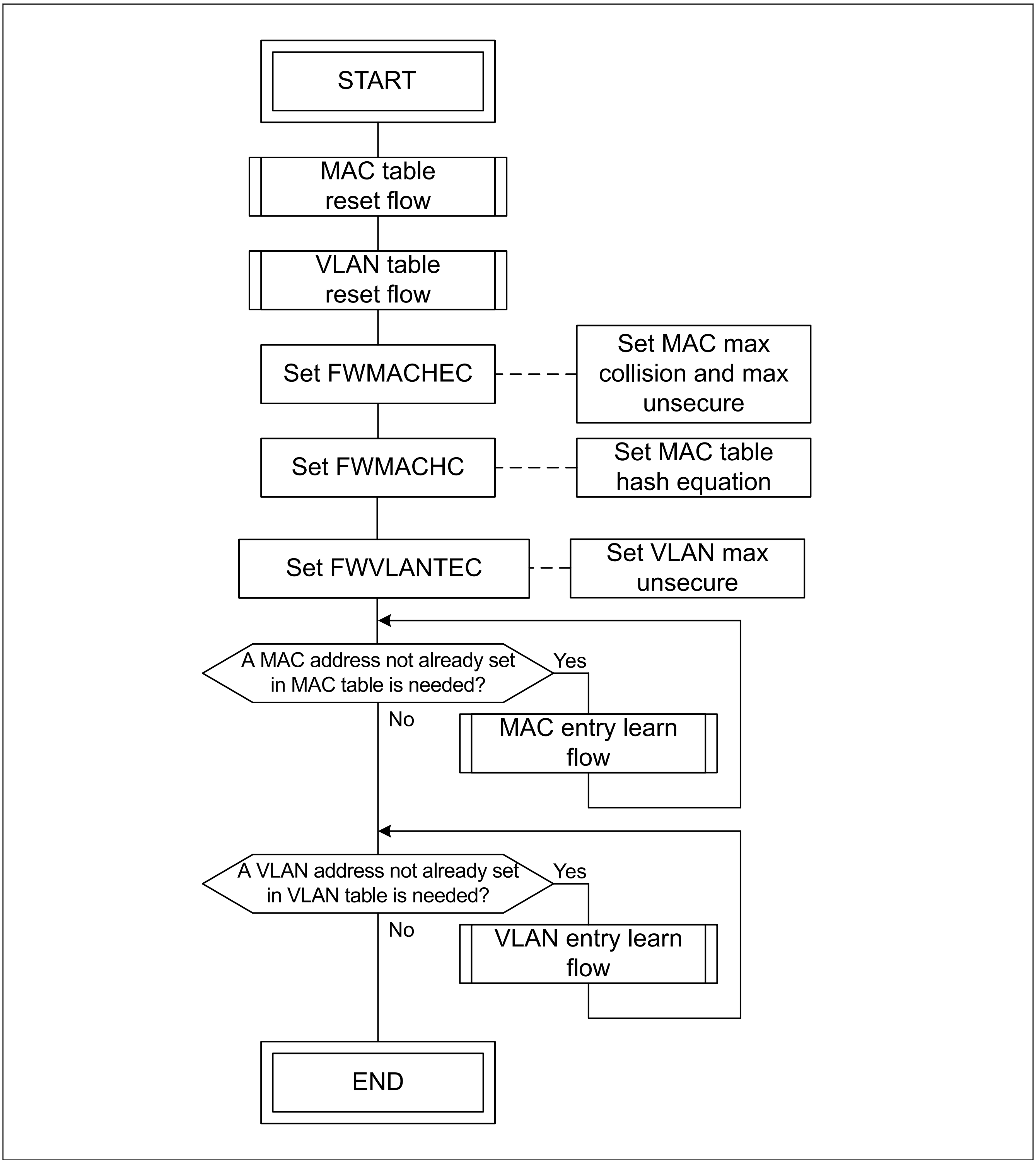


Figure 30.42 Setting flow of Layer 2 forwarding

30.4.1.11.11 Port-based Forwarding Setting Flow

Figure 30.43 describes the setting flow of port-based forwarding.

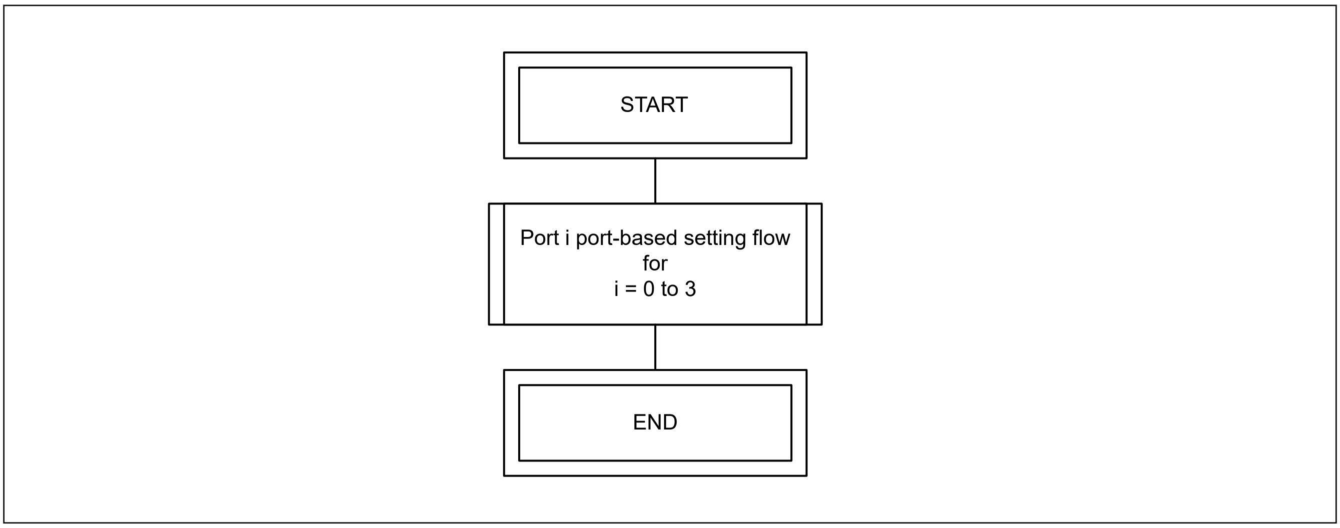


Figure 30.43 Setting flow of port-based forwarding

30.4.1.11.12 Layer 2/Layer 3 Update Setting Flow

Figure 30.44 describes the setting flow of Layer 2/Layer 3 update.

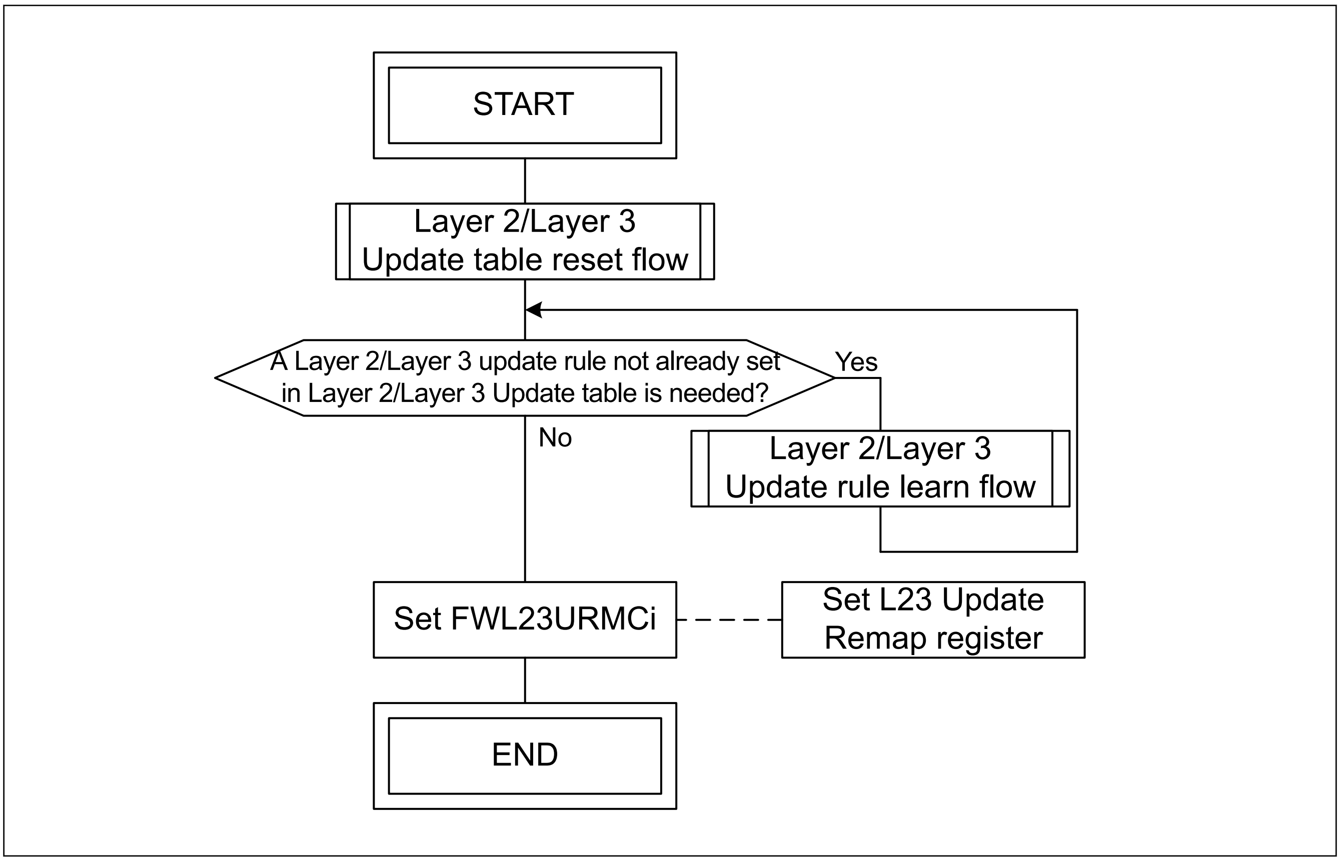


Figure 30.44 Setting flow of Layer 2/Layer 3 update

30.4.1.11.13 PSFP Setting Flow

Figure 30.45 describes the PSFP setting flow.

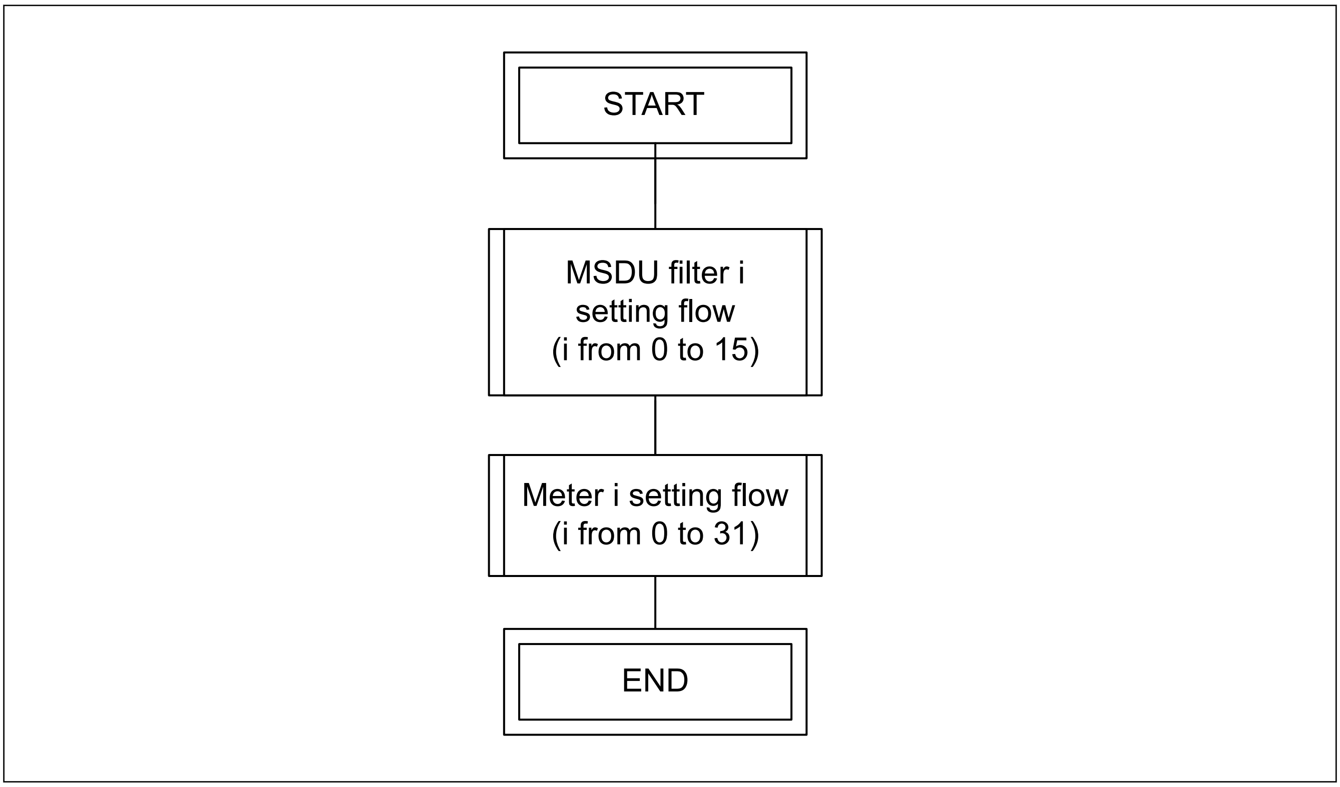


Figure 30.45 PSFP setting flow

30.4.1.11.14 FRER Setting Flow

FRER setting flow is described in Figure 30.46.

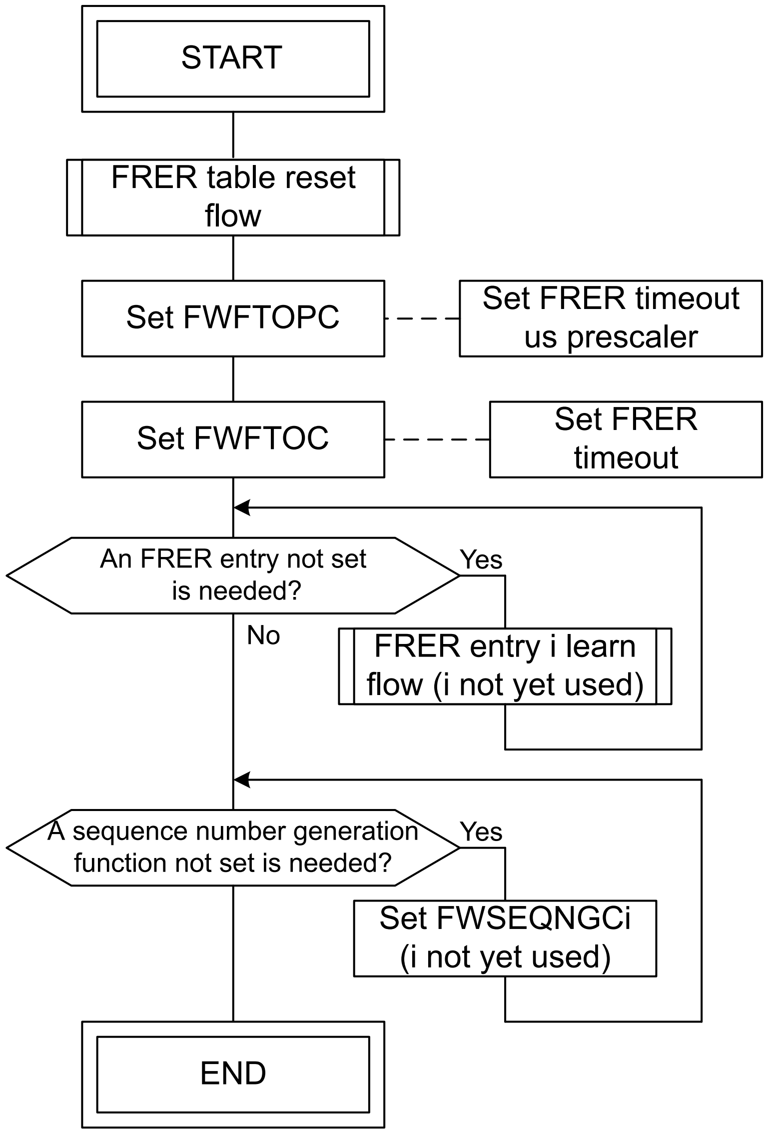


Figure 30.46 FRER setting flow

30.4.1.12 Register Writable Without Software Flow after Initialization

Table 30.4 describes the registers writable without software flow and which mode they can be written without software flow.

Table 30.4 Writable without software flow registers (1 of 2)

(✓ = writable, × = not writable)

Register name	Writable anytime
FWCEPTC	✓
FWCEPRC0/1/2	✓
FWCLPTC	✓
FWCLPRC	✓
FWCMPTC	✓
FWEMPTC	✓
FWSDMPTC	✓

Table 30.4 Writable without software flow registers (2 of 2)

(✓ = writable, × = not writable)

Register name	Writable anytime
FWSDMPVC	✓
FWLBWMCi	✓
FWPC0i/1i/2i	✓
FWL23URMCi	✓

30.5 Functional Details

30.5.1 General Functionalities

General functionalities are separated in several types:

- Descriptors
- ESWM modes
- ESWM TPIDs
- ESWM debug functions
- ESWM watermark

30.5.1.1 Descriptors

To forward a frame to one or several port, the forwarding emits descriptors through the descriptor bus. All descriptor emitted by the forwarding engine shall follow the descriptor format described in Table 30.5. In following sections when a function emits a descriptor it will be explained using the field name. Only fields that are not determined by the input local descriptor or the input frame will be explained.

Table 30.5 Descriptor field (1 of 2)

Field name FDESCR	Bit width	Field description	Values
DV	3	Values: ● Bit i set to 0: Descriptor not valid for port i ● Bit i set to 1: Descriptor valid for port i	Depends on descriptor. Will be explained per descriptor in following sections.
SEC	1	Values: ● 0: Descriptor is unsecure ● 1: Descriptor is secure	Depends on descriptor. Will be explained per descriptor in following sections.
CSD	6	Bits [5:0] (written as FDESCR.CSDi) define the CPU sub-destination for GWCA number i (CPU sub-destination is equivalent to AXI descriptor queue number)	Depends on descriptor. Will be explained per descriptor in following sections.
IPV	3	Internal priority value before agent internal priority remapping	Depends on descriptor. Will be explained per descriptor in following sections.
SPN	2	Port number from which the frame came from	Set to the source agent port number given by mfw_spn_fwd pin
TPL	16	Input frame total payload length.	Set to the TPL value given by agent through local descriptors
MINFO	24	Meta-information associated to a frame for CPU forwarding	Depends on descriptor. Will be explained per descriptor in following sections.
RV	1	Routing valid. Informs agents that routing rules should be fetched to forwarding engine through L23 update bus to update frame information	Depends on descriptor. Will be explained per descriptor in following sections.
RN	8	Routing number. Informs agents that routing number should be used when FDESCR.RV is set.	Depends on descriptor. Will be explained per descriptor in following sections.
VCTRL	3	VLAN control information	Set to the VCTRL value given by agent through local descriptors

Table 30.5 Descriptor field (2 of 2)

Field name FDESCR	Bit width	Field description	Values
RTGI	1	R-TAG in	Set to the RTGI value given by agent through local descriptors
SEQN	16	Frame sequence number	Depends on descriptor. Will be explained per descriptor in following sections.
FI	1	Values: ● 0: Input frame does not contain its FCS ● 1: Input frame contains its FCS	Set to the FI value given by agent through local descriptors
FW	1	Values: ● 0: If input frame contains its FCS, it was correct when given to the switch ● 1: If input frame contains its FCS, it was incorrect when given to the switch	Set to the FW value given by agent through local descriptors

30.5.1.2 ESWM Modes

ESWM modes define the general behavior of the switch using FWGC registers. The functions of ESWM modes are as follows:

- FWGC register sets the VLAN format that is used by the forwarding engine. Depending on this setting, all the agents should convert ethernet frames in the required format (for format conversion, see GWCA specification document [GWCA]).

30.5.1.3 ESWM TPIDs

ESWM TPIDs for C-TAG, S-TAG and R-TAG [801.2Q] [801.2CB] can be set by the user using FWTTC0 and FWTTC1 registers.

Functions:

- FWTTC0 register sets the S-TAG and C-TAG TPIDs [801.2Q].
- FWTTC1 register sets the R-TAG TPID [801.2CB].
- FWTTC0 and FWTTC1 register reset values are as per defined in IEEE standards [801.2Q] [801.2CB].

30.5.1.4 ESWM Frame Input Priority

Any incoming frame has an input priority given by forwarding engine depending on the switch mode FWGC.SVM, and the incoming priority settings in FWPBFCi register including the frame format. Figure 30.47 describes the frame input priority assignment for a frame coming from port i (the ToS DSCP field decoding is fixed in hardware and described in Table 30.6).

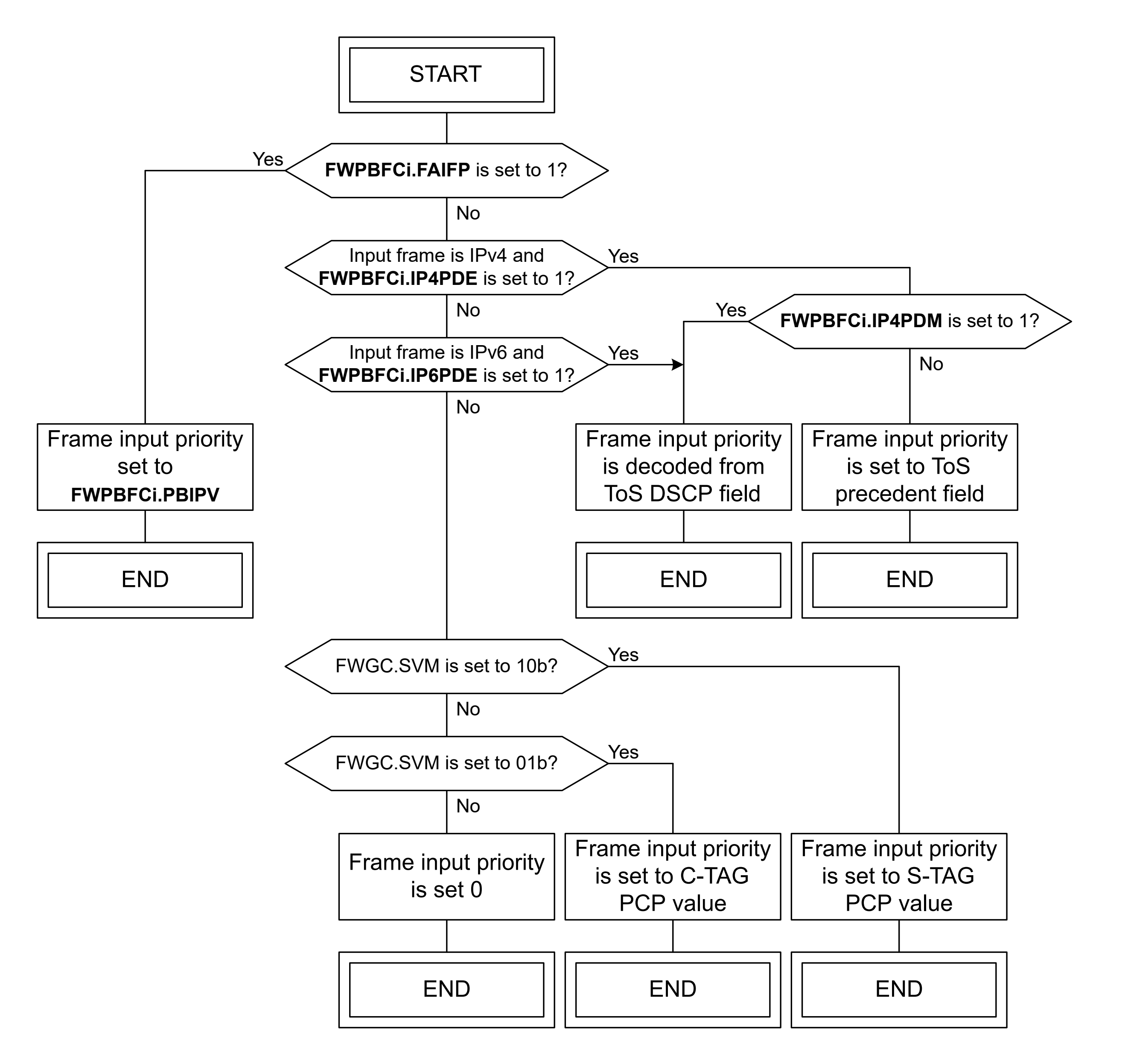


Figure 30.47 Priority assignment of frame input for port i

Note: HW: For cut-through forwarding, ToS and DSCP are not considered as frame input priority. Frame input priority is always extracted from S-TAG, S-TAG or set to 0.

Table 30.6 DSCP to priority mapping

DSCP values in decimal	Frame input priority
56	7
48	6
40, 46	5
32, 34, 36, 38	4
24, 26, 28, 30	3
16, 18, 20, 22	2
8, 10, 12, 14	1
Others	0

30.5.1.5 ESWM Debug Functions

30.5.1.5.1 CPU Exceptional Path

Exceptional path is used to forward faulty frames to CPU for diagnosis using FWCEPTC and FWCEPRC0/1/2 registers. When a frame is selected by exceptional path, an exceptional descriptor is emitted. The functions of exceptional path are as follows:

- FWCEPTC register sets the target to which type of faulty frames is forwarded
- FWCEPRC0/1/2 registers set which error frame is forwarded to the exceptional path.

Note: For exceptional path descriptor formats and associated error details, see section 30.5.2.3.3. Direct Exceptional Descriptor, (4) Layer 3 exceptional descriptor, (3) Layer 2 exceptional descriptor, section 30.5.2.6.3. Port Based Exceptional Descriptor and (5) Filter exceptional descriptor.

30.5.1.5.2 CPU Learning Path

Learning path is used to forward frames containing an unknown field to CPU for learning by using the FWCLPTC and FWCLPRC registers for the CPU to learn it. Learning can also be done by hardware using the FWPCi0 register and can be monitored using the FWEIS0i interrupt register. Table 30.7 sums-up the field that can be learned in software (corresponding frames will be forwarded to learning path) and Table 30.8. Any frame sent to learning path will be sent using learning descriptor described in (1) Learning descriptor.

Note: HW: Any frame rejected by Watermark is not analyzed for learning. As a result, learning path is deactivated and hardware learning does not happen.

Functions:

- FWCLPTC register sets the target to which unknown frames will be forwarded.
- FWCLPRC registers set which field should be unknown for a frame to be forwarded to the learning path.
- FWPCi0 registers sets for which fields hardware learning should happen.
- FWEIS0i interrupt register notifies that a field could not be learnt in hardware.

Table 30.7 Software learning summary table

Field to learn	Learning type	Learning detection conditions for a frame coming from port i	Learning enable bit in FWCLPRC register	Hardware learning possible
Stream ID	Learning	FWPCi0.LTHTA is set. At least a stream ID is valid. Stream ID is not set in the layer 3 table.	USIDLF	No
Destination MAC	Learning	FWPCi0.MACDSA is set. Destination MAC is not set in the MAC table.	UDMACLF	No
Source MAC	Learning	FWPCi0.MACSSA is set. Source MAC is a unicast address. Source MAC is not set in the MAC table.	USMACLF	Yes See Table 30.8.
Source MAC port	Migration	FWPCi0.MACSSA is set. Source MAC is a unicast address. Source MAC is set in the MAC table. MAC.DV[i] is not set.	UPSMACLF	Yes See Table 30.8.
VLAN	Learning	FWPCi0.VLANSA is set. VLAN is valid. VLAN ID is not set in the VLAN table.	UVLANLF	No

Table 30.8 Hardware learning summary table

Field to learn	Learning type	Hardware learning enable bit in FWPCi0 register for port i	Hardware learning fail interrupt in FWEIS0i register for port i	Hardware learning condition	Hardware learning disable condition (for certain frames HW learning can be disabled)
Source MAC	Learning	MACHLA	SMHLFS	Same as SW learning. See Table 30.7.	FWPCi0.MACDSA is set. Destination MAC is set in the MAC table. MAC.HLD is set. Or FWPCi0.VLANSA is set. VLAN is valid. VLAN ID is set in the VLAN table. VLAN.HLD is set.
Source MAC port	Migration	MACHMA	SMHMFS	Same as SW learning. See Table 30.7.	FWPCi0.MACDSA is set. Destination MAC is set in the MAC table. MAC.HLD is set. Or FWPCi0.VLANSA is set. VLAN is valid. VLAN ID is set in the VLAN table. VLAN.HLD is set.

Note: A multicast MAC address is an address with bit 40 set to 1.

(1) Learning descriptor

Learning descriptor fields are described in Table 30.9.

Table 30.9 Learning descriptor fields

Field name FDESCR.	Values
DV	Learning descriptors are forwarded to GWCA number FWCLPTC.LPCS (port number 2 + FWCLPTC.LPCS).
SEC	FWCLPTC.LPSL
CSDj	FWCLPTC.LPCSD
IPV	FWCLPTC.LPIPV
MINFO	See Figure 30.48 and Table 30.10.
RV	0
RN	ALL0
SEQN	Set to corresponding normal descriptor sequence number.

<table>
<tr><th>Byte</th><th>Bit7</th><th>Bit6</th><th>Bit5</th><th>Bit4</th><th>Bit3</th><th>Bit2</th><th>Bit1</th><th>Bit0</th></tr>
<tr><td>0</td><td colspan="4">—</td><td colspan="4">FWDC[3:0]</td></tr>
<tr><td>1</td><td>VL</td><td>SMPL</td><td>SML</td><td>DML</td><td colspan="3">—</td><td>SIDL</td></tr>
<tr><td>2</td><td colspan="4">—</td><td>SMHMF</td><td>SMHLF</td><td>SMHM</td><td>SMHL</td></tr>
</table>

Figure 30.48 Learning descriptor MINFO format

Table 30.10 Learning descriptor MINFO field explanation (1 of 2)

Field name FDESCR.	Bit width	Field explanation	Values
FWDC	4	ForWarDing Code Present in all descriptors, allowing the type of forwarding to be identified	Fixed to 13 for Learning descriptors

Table 30.10 Learning descriptor MINFO field explanation (2 of 2)

Field name FDESCR.	Bit width	Field explanation	Values
SIDL	1	Stream ID Learn	See section 30.5.1.5.2. CPU Learning Path.
DML	1	Destination MAC Learn	See section 30.5.1.5.2. CPU Learning Path.
SML	1	Source MAC Learn	See section 30.5.1.5.2. CPU Learning Path.
SMPL	1	Source MAC Port Learn (migration)	See section 30.5.1.5.2. CPU Learning Path.
VL	1	VLAN Learn	See section 30.5.1.5.2. CPU Learning Path.
SMHL	1	Source MAC Hardware Learn*1	See section 30.5.1.5.2. CPU Learning Path.
SMHM	1	Source MAC Hardware Migration*1	See section 30.5.1.5.2. CPU Learning Path.
SMHLF	1	Source MAC Hardware Learn Fail*1	See section 30.5.1.5.2. CPU Learning Path.
SMHMF	1	Source MAC Hardware Migration Fail*1	See section 30.5.1.5.2. CPU Learning Path.

Note 1. "Hardware Learn/Migration" will set regardless of "Hardware Learn/Migration Fail". (If "Hardware Learn Fail" is set, "Hardware Learn" is always set.)

30.5.1.5.3 Mirroring Paths

Forwarding engine contains 3 mirroring paths:

- CPU mirroring path
- Ethernet mirroring path
- Source-destination mirroring path

(1) CPU mirroring path

CPU Mirroring path is used to copy frames to a CPU for monitoring purpose using FWCMPTC register. Table 30.11 describes CPU mirroring selection conditions depending on the forwarding mechanism selected by forwarding flow. When a frame is selected for CPU mirroring, a mirroring descriptor is emitted ((4) Mirroring descriptor).

Functions:

- FWCMPTC register sets the target to which mirrored frames will be forwarded.

Table 30.11 CPU mirroring selection conditions

Descriptor selected by forwarding flow	Mirroring selection conditions
Cut-through normal descriptor (Figure 30.53)	Cut-through rule corresponding FWCTTCi1.CTCME is set.
Layer 3 normal descriptor (Figure 30.85, Figure 30.86)	L3.CME is set for stream ID
Layer 2 normal descriptor (Figure 30.95, Figure 30.96)	If {MAC.CME is set for destination MAC or VLAN.CME is set} when MAC destination address and VLAN ID are both found and MAC.SL = VLAN.SL. If MAC.CME is set for destination MAC when MAC destination only is found or if MAC destination address and VLAN ID are both found and MAC.SL & !VLAN.SL. If VLAN.CME is set when VLAN ID only is found or if MAC destination address and VLAN ID are both found and !MAC.SL & VLAN.SL.
Port based normal descriptor (Figure 30.99, Figure 30.100)	If FWPBFCi.PBCME is set for incoming frame source port.

(2) Ethernet mirroring path

Ethernet Mirroring path is used to copy frames to an ethernet agent for monitoring purpose using FWEMPTC register. Table 30.12 describes Ethernet mirroring selection conditions depending on the forwarding mechanism selected by forwarding flow. When a frame is selected for Ethernet mirroring, a mirroring descriptor is emitted ((4) Mirroring descriptor).

Functions:

- FWEMPTC register sets the target to which mirrored frames will be forwarded.

Table 30.12 Ethernet mirroring selection conditions

Descriptor selected by forwarding flow	Mirroring selection conditions
Cut-through normal descriptor (Figure 30.53)	Cut-through rule corresponding FWCTTCi1.CTEME is set.
Layer 3 normal descriptor (Figure 30.85, Figure 30.86)	L3.EME is set for stream ID
Layer 2 normal descriptor (Figure 30.95, Figure 30.96)	If {MAC.EME is set for destination MAC or VLAN.EME is set} when MAC destination address and VLAN ID are both found and MAC.SL = VLAN.SL. If MAC.EME is set for destination MAC when MAC destination only is found or if MAC destination address and VLAN ID are both found and MAC.SL & !VLAN.SL. If VLAN.EME is set when VLAN ID only is found or if MAC destination address and VLAN ID are both found and !MAC.SL & VLAN.SL.
Port based normal descriptor (Figure 30.99, Figure 30.100)	If FWPBFCi.PBEME is set for incoming frame source port.

(3) Source-destination Mirroring path

Mirroring path is used to copy frames to an Ethernet port or a CPU for monitoring purpose using FWSDMPTC and FWSDMPVC register. When a frame is selected for Source-destination mirroring, a mirroring descriptor is emitted ((4) Mirroring descriptor).

Functions:

- FWSDMPTC register sets the target to which mirrored frames will be forwarded.
- FWSDMPVC register sets the source from which should come and the destination where should go a frame to be mirrored.
- A frame is selected for source-destination mirroring if the descriptor selected by forwarding flow is {Layer 3 normal descriptor (Figure 30.85, Figure 30.86) or IP normal descriptor (Figure 30.95, Figure 30.96) or Layer 2 normal descriptor (Figure 30.95, Figure 30.96) or Port based normal descriptor (Figure 30.99, Figure 30.100)}, if the frame source port is set in FWSDMPVC.SDMSV register and if FDESCR.DV & FWSDMPVC.SDMDV is not null.

(4) Mirroring descriptor

Mirroring descriptor fields are described in Table 30.13.

Table 30.13 Mirroring descriptor fields (1 of 2)

Field name FDESCR.	Values for CPU mirroring	Values for Ethernet mirroring	Values for Source-destination mirroring
DV	Mirroring descriptors are forwarded to GWCA number FWCMPTC.CMPCS (port number 2 + FWCMPTC.CMPCS)	Mirroring descriptors are forwarded to ETHA number FWEMPTC.EMPPS (port number FWEMPTC.EMPPS)	Mirroring descriptors are forwarded to port number FWSDMPTC.SDMPPS
SEC	FWCMPTC.CMPSL	FWEMPTC.EMPSL	FWSDMPTC.SDMPSL
CSDj	FWCMPTC.CMPCSD	All0	All0
IPV	If FWCMPTC.CMPIPU is set, FWCMPTC.CMPIPV If FWCMPTC.CMPIPU is not set, same IPV as normal descriptor	If FWEMPTC.EMPIPU is set, FWEMPTC.EMPIPV If FWEMPTC.EMPIPU is not set, same IPV as normal descriptor	If FWSDMPTC.SDMPIPU is set, FWSDMPTC.SDMPIPV If FWSDMPTC.SDMPIPU is not set, same IPV as normal descriptor
MINFO	See Figure 30.49 and Table 30.14.	See Figure 30.49 and Table 30.14.	See Figure 30.49 and Table 30.14.
RV	0	0	0

Table 30.13 Mirroring descriptor fields (2 of 2)

Field name FDESCR.	Values for CPU mirroring	Values for Ethernet mirroring	Values for Source-destination mirroring
RN	ALL0	ALL0	ALL0
SEQN	Set to corresponding normal descriptor sequence number	Set to corresponding normal descriptor sequence number	Set to corresponding normal descriptor sequence number

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	—				FWDC[3:0]			
1	NDFB[7:0]							
2	—							

Figure 30.49 Mirroring descriptor MINFO format

Table 30.14 Mirroring descriptor MINFO field explanation

Field name FDESCR.	Bit width	Field explanation	Values
FWDC	4	ForWarDing Code Present in all descriptors, it allows identifying the type of forwarding.	Fixed to 14 for Mirroring descriptors
NDFB	8	Normal Descriptor First Byte	Set to mirrored frame corresponding normal descriptor FDESCR.MINFO[7:0] value

30.5.1.6 Watermark

ESWM watermark defines the switch behavior in case of overflow using FWLBWMCi and FWCEPRC2 registers and can be monitored using FWEIS0i interrupt register.

Functions:

- FWLBWMCi is used to set which frame should be discarded by level-based watermark [COMA].
- FWCEPRC2 register enables exceptional path for error descriptors.
- FWEIS0i interrupt register notifies that a frame coming from port i has been rejected because of watermark.
- Watermark uses has frame priority the IPV value given by forwarding mechanism before filtering. This correspond to the FDESCR.IPV in the normal descriptor emitted by the selected forwarding.
- Watermark uses as DEI value the value extracted from the input frame which is the S-TAG DEI value when FWGC.SVM is set to 10b, the C-TAG DEI value when FWGC.SVM is set to 01b, and 0 when FWGC.SVM is set to 00b.

Note:
- SW: The watermark is an indispensable functionality for switch QoS behavior definition, it is not advised to disable it.
- Error frames are not going through watermark.

(1) Watermark errors

Table 30.15 describes the Watermark errors. Instead of being discarded by forwarding engine, an error frame can be forwarded to exceptional path using Watermark exceptional descriptor described in (2) Watermark exceptional descriptor by setting corresponding error exceptional bit. For error set conditions, refer to corresponding error interrupt register explanations.

Table 30.15 Watermark errors (1 of 2)

Error	Error name	Error interrupt	Error exceptional bit
WMCF	Watermark Critical Filtering	FWEIS0i.WMCFS	FWCEPRC2.FWMFEF

Table 30.15 Watermark errors (2 of 2)

Error	Error name	Error interrupt	Error exceptional bit
WMFF	Watermark Flush Filtering	FWEIS0i.WMFFS	FWCEPRC2.FWMFEF
WMISF	Watermark IPV Secure Filtering	FWEIS0i.WMISFS	FWCEPRC2.FWMFEF
WMIUF	Watermark IPV Unsecure Filtering	FWEIS0i.WMIUFS	FWCEPRC2.FWMFEF

(2) Watermark exceptional descriptor

Watermark exceptional descriptor fields are described in Table 30.16.

Table 30.16 Watermark exceptional descriptor fields

Field name FDESCR.	Values
DV	Exceptional descriptors are forwarded to GWCA number FWCEPTC.EPCS (port number 2 + FWCEPTC.EPCS)
SEC	FWCEPTC.EPSL
CSDj	FWCEPTC.EPCSD
IPV	FWCEPTC.EPIPV
MINFO	See Figure 30.50 and Table 30.17.
RV	0
RN	ALL0
SEQN	Set to frame input sequence number

<table>
<tr><th>Byte</th><th>Bit7</th><th>Bit6</th><th>Bit5</th><th>Bit4</th><th>Bit3</th><th>Bit2</th><th>Bit1</th><th>Bit0</th></tr>
<tr><td>0</td><td colspan="4">—</td><td colspan="4">FWDC[3:0]</td></tr>
<tr><td>1</td><td colspan="4">—</td><td>WMIUF</td><td>WMISF</td><td>WMFF</td><td>WMCF</td></tr>
<tr><td>2</td><td colspan="8">—</td></tr>
</table>

Figure 30.50 Watermark exceptional descriptor MINFO format

Table 30.17 Watermark exceptional descriptor MINFO field explanation

Field name FDESCR.	Bit width	Field explanation	Values
FWDC	4	ForWarDing Code Present in all descriptors, it allows identifying the type of forwarding.	Fixed to 4 for Watermark exceptional descriptors
WMCF	1	Watermark Critical Filtering	See (1) Watermark errors.
WMFF	1	Watermark Flush Filtering	See (1) Watermark errors.
WMISF	1	Watermark IPV Secure Filtering	See (1) Watermark errors.
WMIUF	1	Watermark IPV Unsecure Filtering	See (1) Watermark errors.

30.5.2 Forwarding/Routing

Forwarding engine contains different types of forwarding which are:

- Cut-through forwarding
- Direct descriptor forwarding
- L3 forwarding/routing
- L2 forwarding
- Port based forwarding

30.5.2.1 Cut-through Forwarding

Cut-through forwarding is used to forward high priority frames without storing them in the local memory to suppress the frame reception time impact on the latency. Under the condition of ESWM insure best latency (no busy traffic on MFWD and no TAG frame) of

0.7 µs* + "PHY clock 4 cycle"(ns), 4.6 µs* + "PHY clock 4 cycle"(ns) and 46 µs* + "PHY clock 4 cycle"(ns) at respectively 10/5/2.5/1 Gbps, 100 Mbps and 10 Mbps [ETHA].

Depending on the setting conditions, the value may be smaller than these.

Generally, "Cut-through" only checks 6 bytes (Destination MAC Address) of a frame head and starts frame forwarding after this check immediately. Since internal data bus width in this switch is wider than 6 bytes, it needs more data bytes to start frame forwarding. Because of this logic, "Cut-through" forwarding time of this switch needs above times.

The Cut-through forwarding block diagram is represented in Figure 30.51.

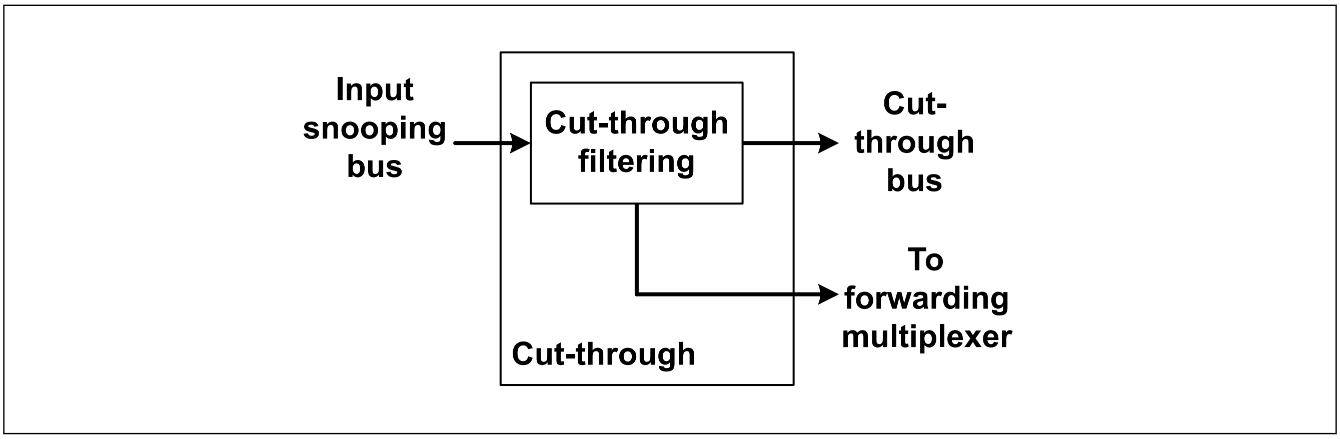


Figure 30.51 Cut-through forwarding block diagram

The Cut-through forwarding function has one functional block:

- Cut-through filtering: This function filters incoming frame first beat of data to select set frames for cut-through forwarding, creates the cut-through bus protocol destinated to time agents [ETHA] and send required information to forwarding multiplexer to apply store and forward to cut-through frames.

Note:
- Forwarding multiplexer contains the forwarding flow. The forwarding flow is explained per forwarding type. For Cut-though, related Forwarding flow is described in Figure 30.53
- HW: The usage "From-one-to-one between same PHY speed" is recommended. So in case of forwarding engine back pressure, it can happen that cut-through frames are cut in the middle and lost.
- HW: P-frames cannot be forwarded by cut-through.
- HW: Cut-through forwarding should not be set for ingress ports in Port-based VLAN mode because cut-through can only forward frame as they were received. ([ETHA] EAVCC.VIM set to 0)
- HW: Watermark is not applied on Cut-through frames.
- HW: Cut-through frames can be ingress tagging by FWCTGCi0.CTVCRL but not egress tagging.
- HW: L23 update is not applied on Cut-through frames.
- HW: Data padding applied on Cut-through frames.
- HW: Cut-through frames can be shaped using TAS but not CBS.
- HW: Cut-through frames will be forward in cut-through mode as there are received and will not be updated.
- HW: For cut-through, only mirroring set in FWCTTCi1/i2 register is available. All other debugging and mirroring are disabled.
- HW: Cut-through frames will not be filtered by [RMAC] MAC Address filtering on Reception. Because [ETHA] and [GWCA] reception filter information are ignored.
  For example, a frame which is having a nibble alignment error ([ETHA] reception filter information) is ignored by cut through forwarding. And a frame is transmitted out by padding bits of data to complete a byte.

30.5.2.1.1 Cut-through Filtering

Cut-through filtering is used to identify cut-through frames using FWCTGCi0/i1 and FWCTSCi0/i1/i2/i3/i4 registers. Cut-through filtering and register utilization are summarized in Figure 30.52.

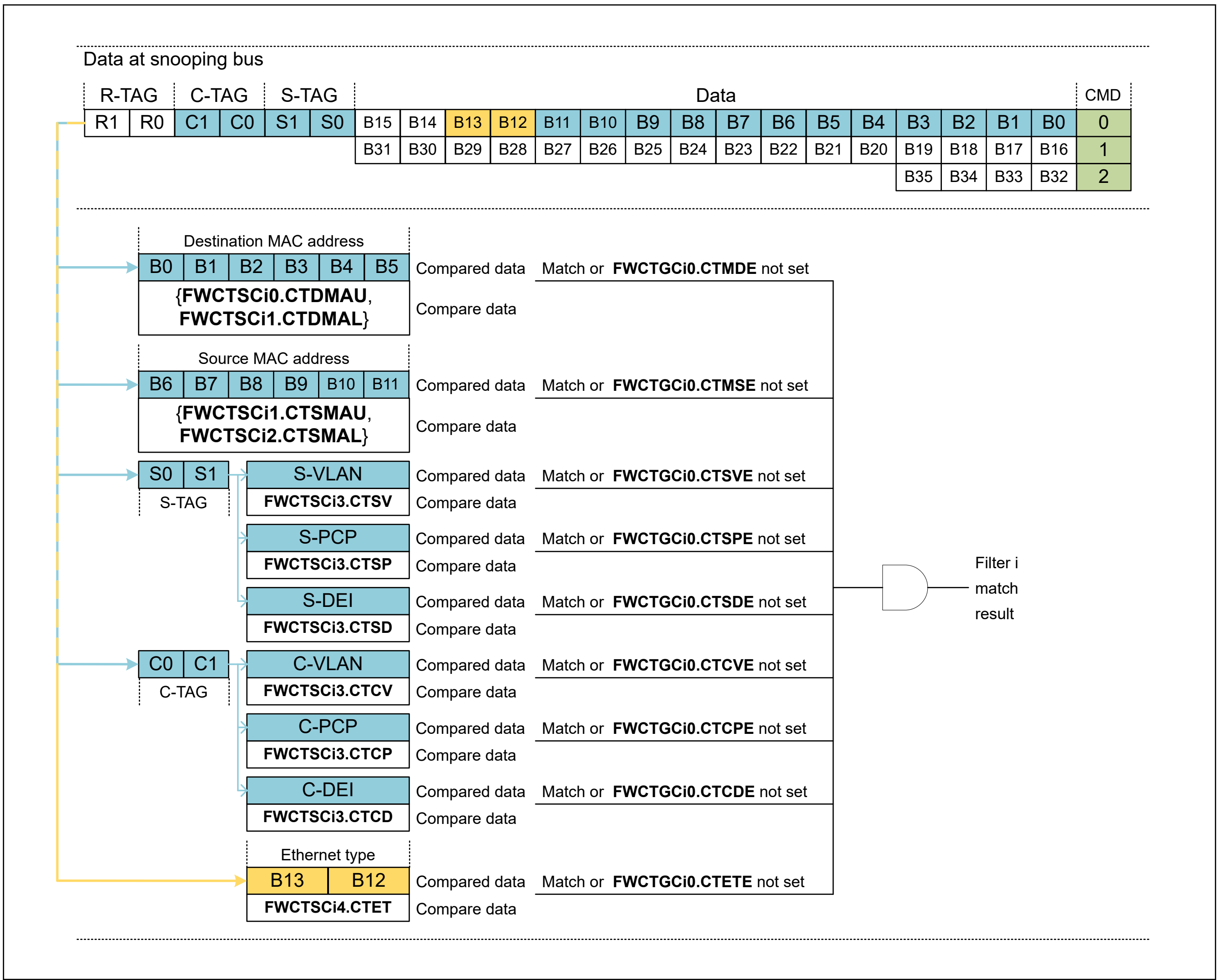


Figure 30.52 Cut-through filtering

Functions:

- A cut-through rule can only match frames from corresponding source port FWCTSCi4.CTSPN.
- If several Cut-through rules match the same frame, the Cut-through rules with the highest number will be selected.
- A cut-through rule is only valid when corresponding FWCTTCi0.CTDV is different from All0.
- A cut-through matching frames is forwarded by cut-through and store and forward independently of any error that could have happened in the ingress agent.
- For register FWCTGCi0 bit 0-8 set to 0 (all fields compare match is disabled), Cut through becomes “port based cut through” by FWCTTCi0.CTDV.

30.5.2.1.2 Cut-through Forwarding

Cut-through forwarding function creates the cut-through bus protocol destinated to time agents [ETHA] using FWCTTCi0, FWCTGCi0/i1 registers. It is used to send frames while receiving them to reduce latency though cut-through bus.

Functions:

- FWCTTCi0 register is to determine the target port for cut-through. All ports set in FWCTTCi0.CTDV[PORT_TIME_N-1:0] and not set in FWCTTCi0.CTDFM are targeted for rule i cut-through forwarding. Any frame received from an agent with an error will still be forwarded/mirrored by cut-through forwarding.
- FWCTGCi1.CTMT, FWCTGCi0.CTVCTRL, and FWCTGCi0.CTFI registers are respectively used as Maximum Time on the PHY, VLAN control and FCS in for the matching cut-through frame.

Note:
- HW: Ethernet timestamping functionalities are not available during cut-through forwarding [ETHA]. E.g. while forwarding frames to RMAC [RMAC], the ethernet agent will fix “Timestamp capture”, “Insert egress timestamp” and “Calculate residence time” TX interface bit to 0.
- HW: For cut-through forwarding, all (From & To) PHY speeds have to be the same[ETHA] [RMAC].
- SW: FWCTGCi0.CTRTGI and FWCTGCi0.CTVCTRL should always correspond to the ingress frame format. If then do no correspond, errors can happen.

30.5.2.1.3 Store and Forward Forwarding

Store and forward forwarding function apply store and forward to cut-through frames using FWCTTCi0/i1/i2. Cut-through forwarding flow for Store and forward forwarding is described in Figure 30.53.

Functions:

- FWCTTCi0 register is to determine the target port for store and forward. All ports set in FWCTTCi0.CTDV[2:0] and set in FWCTTCi0.CTDFM are targeted for rule i store and forward forwarding, so are ports set in FWCTTCi0.CTDV[3:2]. Any frame received from an agent with an error will still be forwarded/mirrored in store and forward mode.
- FWCTTCi1/i2 registers sets the forwarding information.

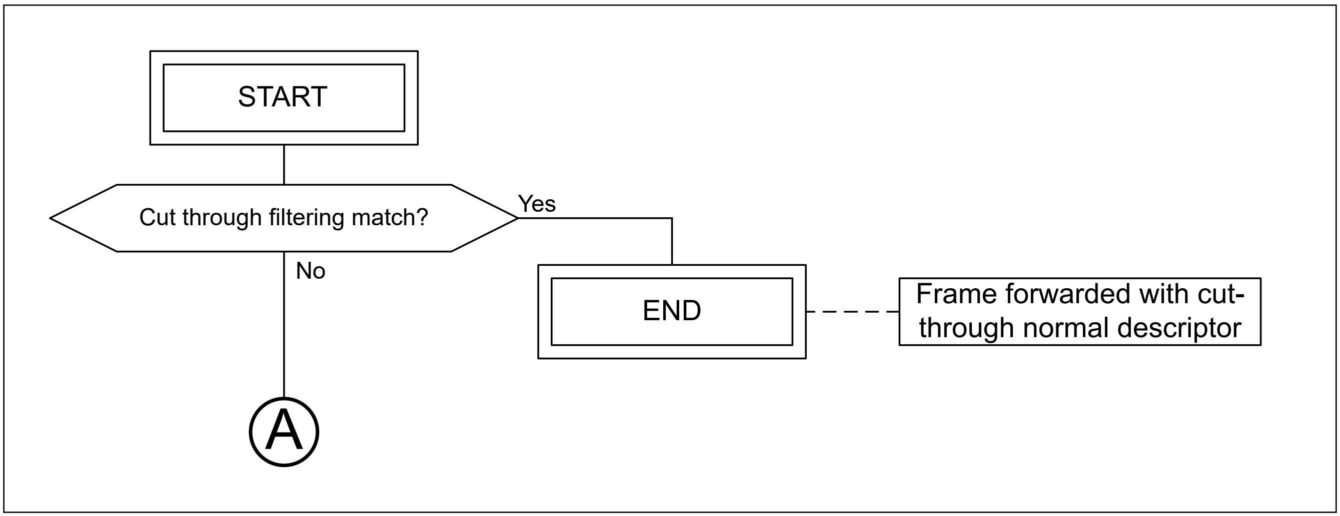


Figure 30.53 Cut-through forwarding flow

Note: Link A links to Figure 2.

(1) Cut-through normal descriptor

Cut-through normal descriptor fields for a frame matching cut-through rule number i are described in Table 30.18.

Table 30.18 Cut-through normal descriptor fields

Field name FDESCR.	Values
DV	{FWCTTCi0.CTDV[3:2], {FWCTTCi0.CTDV[1] & FWCTTCi0.CTDFM}}
SEC	1
CSDj	FWCTTC2ji.CTCSD
IPV	If FWCTTCi1.CTIPU is set, set to FWCTTCi1.CTIPV If FWCTTCi1.CTIPU not set and FWGC.SVM is set to 10b, set to frame input S-TAG PCP If FWCTTCi1.CTIPU not set and FWGC.SVM is set to 01b, set to frame input C-TAG PCP If FWCTTCi1.CTIPU not set and FWGC.SVM is set to 00b, set to frame input 0
MINFO	See Figure 30.54 and Table 30.19
RV	0
RN	ALL0
SEQN	Set to frame input sequence number

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	—				FWDC[3:0]			
1	—					CTRN[2:0]		
2	—							

Figure 30.54 Cut-through normal descriptor MINFO format

Table 30.19 Cut-through normal descriptor MINFO field explanation

Field name FDESCR.	Bit width	Field explanation	Values
FWDC	4	ForWarDing Code Present in all descriptors, allowing the type of forwarding to be identified	Fixed to 0 for Cut-through normal descriptors
CTRN	3	Cut-through rule number	Set to the cut-through rule number which forwarded the frame
—	- -	Reserved field	Set to 0

30.5.2.2 Integrity Check

Integrity check is used to check it a frame can be forwarded or if it contains an error. The Integrity check block diagram is represented in Figure 30.55. Integrity check flow is described in Figure 30.56.

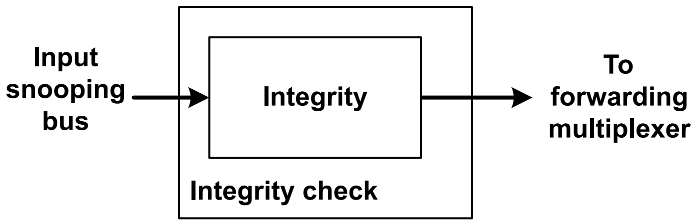
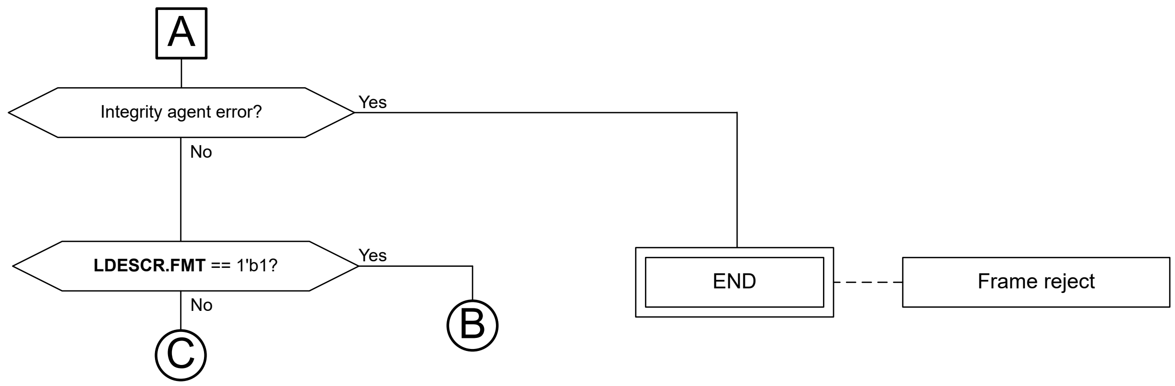


Figure 30.55 Integrity check block diagram

The Integrity check function has one functional block:

- Integrity: The ether type 0x8100, 0x88A8 and 0xF1C1 are ignored and the next ether type field are extracted. However, invalid frames (exp CC-TAG) are judged to have no header, and the first ether type field is the extract.
- Integrity: This functions check if an error has been received from the input agent or if incoming frames don't respect the required format.
- Integrity check for both IPv4 filtering and IPv6 filtering will happen based on ether type field. i.e. Ether type is only used for parsing frame type in MFWD integrity check and is irrespective of header length or TPL size of the required frame type.

Note: • Forwarding multiplexer contains the forwarding flow. The forwarding flow is explained per forwarding type. For integrity check, related Forwarding flow is described in Figure 30.56.



Note:
- Integrity check errors are described in section 30.5.2.2.1. Integrity Check Errors.
- Link A is linked from Figure 30.53.
- Link B links to Figure 30.58.
- Link C links to Figure 30.85.

Figure 30.56 Integrity check flow

30.5.2.2.1 Integrity Check Errors

Table 30.20 describes the Integrity check errors. Instead of being discarded by forwarding engine, an error frame can be forwarded to exceptional path using Integrity exceptional descriptor described in section 30.5.2.2.2. Integrity Exceptional Descriptor by setting its corresponding error exceptional bit.

Table 30.20 Integrity check errors (1 of 2)

Error	Error name	Error interrupt/Explanation	Error exceptional bit
Integrity agent errors			
GTFE	GWCA TAG filtering error	Happens when a frame is received from a GWCA and LDESCR.SAEF[0] is set [GWCA]	FWCEPRC0.GTFEF
GAXE	GWCA AXI Error	Happens when a frame is received from a GWCA and LDESCR.SAEF[3] is set [GWCA]	FWCEPRC0.GAXEEF
GSEGE	GWCA Sequence Error	Happens when a frame is received from a GWCA and LDESCR.SAEF[4] is set [GWCA]	FWCEPRC0.GSEQEEF
GDNE	GWCA Descriptor Number Error	Happens when a frame is received from a GWCA and LDESCR.SAEF[6] is set [GWCA]	FWCEPRC0.GDNEEF
ETFE	Ethernet TAG filtering Error	Happens when a frame is received from an Ethernet agent and LDESCR.SAEF[0] is set [ETHA]	FWCEPRC0.ETFEF
EPHYE	Ethernet TAG PHY Error	Happens when a frame is received from an Ethernet agent and LDESCR.SAEF.EC0 is set to 4’d1 [ETHA]	FWCEPRC0.EPHYEEF
EPCRCE	Ethernet PCH CRC Error	Happens when a frame is received from an Ethernet agent and LDESCR.SAEF.EC0 is set to 4’d2 [ETHA]	FWCEPRC0.EPCRCEEF
ENIBE	Ethernet Nibble Error	Happens when a frame is received from an Ethernet agent and LDESCR.SAEF.EC0 is set to 4’d3 [ETHA]	FWCEPRC0.ENIBEEF
EFCSE	Ethernet FCS Error	Happens when a frame is received from an Ethernet agent and LDESCR.SAEF.EC0 is set to 4’d4 [ETHA]	FWCEPRC0.EFCSEEF
EFFME	Ethernet Final Fragment Missing Error	Happens when a frame is received from an Ethernet agent and LDESCR.SAEF.EC0 is set to 4’d5 [ETHA]	FWCEPRC0.EFFMEEF
ECFSE	Ethernet C-Fragment SMD Error	Happens when a frame is received from an Ethernet agent and LDESCR.SAEF.EC0 is set to 4’d6 [ETHA]	FWCEPRC0.ECFSEEF
ECFFCE	Ethernet C-Fragment FRAG_COUNT Error	Happens when a frame is received from an Ethernet agent and LDESCR.SAEF.EC0 is set to 4’d7 [ETHA]	FWCEPRC0.ECFFCEEF

Table 30.20 Integrity check errors (2 of 2)

Error	Error name	Error interrupt/Explanation	Error exceptional bit
ERFF	Ethernet RMAC Frame Filtered	Happens when a frame is received from an Ethernet agent and LDESCR.SAEF.EC0 is set to 4'd8 [ETHA]	FWCEPRC0.ERFFEF
ERPOO	Ethernet Reception Partially Out of Operation	Happens when a frame is received from an Ethernet agent and LDESCR.SAEF.EC0 is set to 4'd9 [ETHA]	FWCEPRC0.ERPOOEF
EBOE	Ethernet Buffer Overflow Error	Happens when a frame is received from an Ethernet agent and LDESCR.SAEF.EC1 is set to 2'd1 [ETHA]	FWCEPRC0.EBOEEF
EUE	Ethernet Undersize Error	Happens when a frame is received from an Ethernet agent and LDESCR.SAEF.EC1 is set to 2'd2 [ETHA]	FWCEPRC0.EUEEF
EOE	Ethernet Oversize Error	Happens when a frame is received from an Ethernet agent and LDESCR.SAEF.EC1 is set to 2'd3 [ETHA]	FWCEPRC0.EOEEF

30.5.2.2.2 Integrity Exceptional Descriptor

Integrity exceptional descriptor fields are described in Table 30.21.

Table 30.21 Integrity exceptional descriptor fields

Field name FDESCR.	Values
DV	Exceptional descriptors are forwarded to GWCA number FWCEPTC.EPCS (port number PORT_TIME_N+ FWCEPTC.EPCS)
SEC	FWCEPTC.EPSL
CSDj	FWCEPTC.EPCSD
IPV	FWCEPTC.EPIPV
MINFO	Refer to Figure 30.57 and Table 30.22
RV	1'b0
RN	ALL0
SEQN	Set to frame input sequence number

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	–				FWDC[3:0]			
1	–				L4F	IP6F	IP4F	TYPEF
2	–							

Figure 30.57 Integrity exceptional descriptor MINFO format

Table 30.22 Integrity exceptional descriptor MINFO field explanation (1 of 2)

Field name FDESCR.	Bit width	Field explanation	Values
FWDC	4	ForWarDing Code Present in all descriptors, it allows identifying the type of forwarding.	Fixed to 1 for Integrity exceptional descriptors
TYPEF	1	Type filtering	Refer to section 30.5.2.2.1. Integrity Check Errors
IP4F	1	IPv4 filtering	Refer to section 30.5.2.2.1. Integrity Check Errors
IP6F	1	IPv6 filtering	Refer to section 30.5.2.2.1. Integrity Check Errors
L4F	1	Layer 4 filtering	Refer to section 30.5.2.2.1. Integrity Check Errors

Table 30.22 Integrity exceptional descriptor MINFO field explanation (2 of 2)

Field name FDESCR.	Bit width	Field explanation	Values
—	—	Reserved field	Set to 0

30.5.2.3 Direct Descriptor Forwarding

Direct descriptor forwarding is used to forward local direct descriptors (LDESCR.FMT [GWCA] set to 1) received from a slow interface [GWCA] using FWPCi1 and FWCEPRC0 register and can be monitored using FWEIS0i interrupt register. Direct descriptor forwarding flow is described in Figure 30.58.

Functions:

- FWPCi1.DDE register is used to enable direct descriptor reception from port i.
- FWPCi1.DDSL register is used to enable secure direct descriptor reception from port i when direct descriptor reception from port I is enabled.
- FWCEPRC0 register enables exceptional path for error descriptors.
- FWEIS0i interrupt register notifies that a direct descriptor has been rejected.

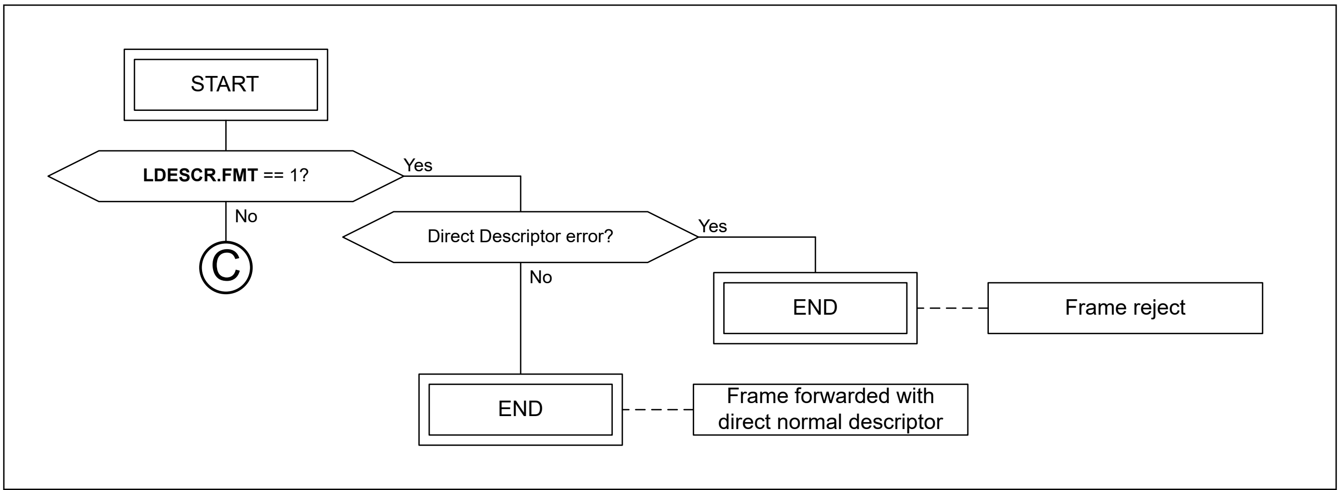


Figure 30.58 Direct descriptor forwarding flow

Note:
- Direct descriptor errors are described in section section 30.5.2.3.1. Direct Descriptor Forwarding Errors.
- Direct normal descriptor format is described in section section 30.5.2.3.2. Direct Normal Descriptor.
- Link C links to Figure 30.85.

30.5.2.3.1 Direct Descriptor Forwarding Errors

Table 30.23 describes the Direct descriptor forwarding errors. Instead of being discarded by forwarding engine, an error frame can be forwarded to exceptional path using Direct exceptional descriptor described in section section 30.5.2.3.3. Direct Exceptional Descriptor by setting corresponding error exceptional bit. For error set conditions, refer to corresponding error interrupt register explanations.

Table 30.23 Direct descriptor errors

Error	Error name	Error interrupt	Error exceptional bit
DDE	Direct Descriptor Error	FWEIS0i.DDES	FWCEPRC0.DDEEF
DDSF	Direct Descriptor Security Filtering	FWEIS0i.DDSES	FWCEPRC0.DDFSFEF
NTF	No Target Filtering	FWEIS0i.DDNTFS	FWCEPRC2.DDNTFEF

30.5.2.3.2 Direct Normal Descriptor

Table 30.24 describes fields of direct normal descriptor. For LDESCR explanation, see GWCA specification [GWCA].

Table 30.24 Direct normal descriptor fields

Field name FDESCR.	Values
DV	LDESCR.DV
SEC	LDESCR.SEC
CSD	LDESCR.CSD
IPV	LDESCR.IPV
MINFO	Refer to Figure 30.59 and Table 30.25
RV	LDESCR.RV
RN	If LDESCR.RV is set, set to LDESCR.RN If LDESCR.RV is not set, set to 0
SEQN	Set to frame input sequence number

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	—				FWDC[3:0]			
1	—							
2								

Figure 30.59 MINFO format of direct normal descriptor

Table 30.25 MINFO field explanation of direct normal descriptor

Field name FDESCR.	Bit width	Field explanation	Values
FWDC	4	Forwarding Code Present in all descriptors, it allows identifying the type of forwarding.	Fixed to 2 for direct normal descriptors
—	- -	Reserved field	Set to 0

30.5.2.3.3 Direct Exceptional Descriptor

Direct exceptional descriptor fields are described in Table 30.26.

Table 30.26 Direct exceptional descriptor fields

Field name FDESCR.	Values
DV	Exceptional descriptors are forwarded to GWCA number FWCEPTC.EPCS (port number 2 + FWCEPTC.EPCS)
SEC	FWCEPTC.EPSL
CSDj	FWCEPTC.EPCSD
IPV	FWCEPTC.EPIPV
MINFO	Refer to Figure 30.60 and Table 30.27
RV	0
RN	ALL0
SEQN	Set to frame input sequence number

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	—				FWDC[3:0]			
1	—				NTF	DDFF	DDSF	DDF
2	—							

Figure 30.60 Direct exceptional descriptor MINFO format

Table 30.27 Direct exceptional descriptor MINFO field explanation

Field name FDESCR.	Bit width	Field explanation	Values
FWDC	4	ForWarDing Code Present in all descriptors, it allows identifying the type of forwarding.	Fixed to 3 for Direct exceptional descriptors
DDF	1	Direct Descriptor Filtered	See section 30.5.2.3.1. Direct Descriptor Forwarding Errors.
DDSF	1	Direct Descriptor Security Filtered	See section 30.5.2.3.1. Direct Descriptor Forwarding Errors.
DDFF	1	Direct Descriptor Format Filtered	See section 30.5.2.3.1. Direct Descriptor Forwarding Errors.
NTF	1	No Target Filtered	See section 30.5.2.3.1. Direct Descriptor Forwarding Errors.

30.5.2.4 Layer 3 Forwarding/Routing/Filtering

Figure 30.61 shows the block diagram of Layer 3 forwarding/routing/filtering. This block is denoted by L3, but it contains all the stream-based functions of the forwarding engine, even the Layer 2 streams.

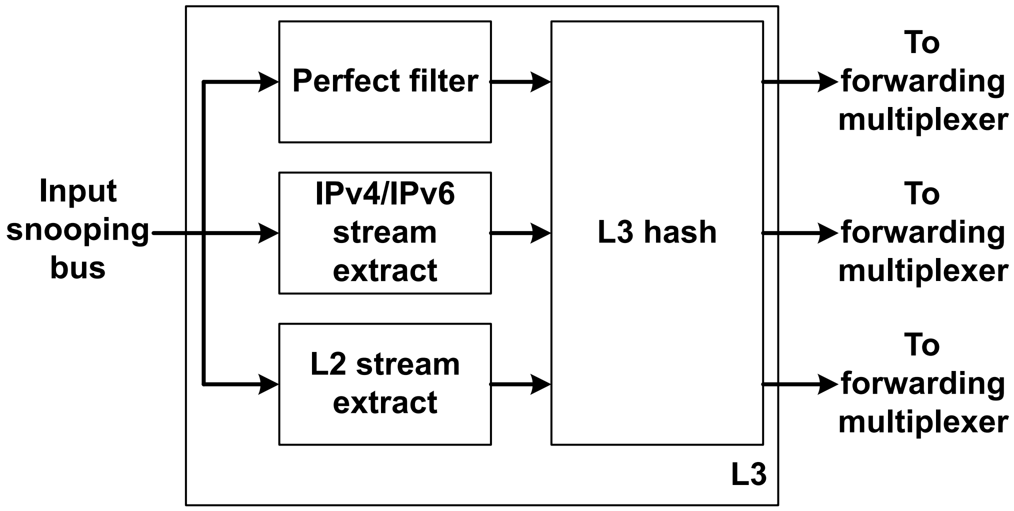


Figure 30.61 Block diagram of Layer 3 forwarding/routing/filtering

The Layer 3 forwarding/routing/filtering function is separated in four functional blocks:

- Perfect filter: This block snoops all the required information from the Forwarding engine snooping bus and create a user-defined stream.
- IPv4/IPv6 stream extract: This block snoops all the required information from the Forwarding engine snooping bus and create an IPv4/IPv6 stream.

- L2 stream extract: This block snoops all the required information from the Forwarding engine snooping bus and create a L2 stream.
- L3 hash: This block handles the stream meta information such as forwarding, routing and filtering information using a hash function.

Note: Forwarding multiplexer contains the forwarding flow. The forwarding flow is explained by forwarding type. For Layer 3 forwarding/routing/filtering, related forwarding flows are described in Figure 30.85, Figure 30.86 and Figure 30.87.

30.5.2.4.1 Perfect Filter

The perfect filter is used to identify streams. This filter has the advantage to be flexible and can identify a stream without any statistic error but, because of its high cost, is available in a limited number (16 streams available).

The Perfect filter block diagram is represented in Figure 30.62.

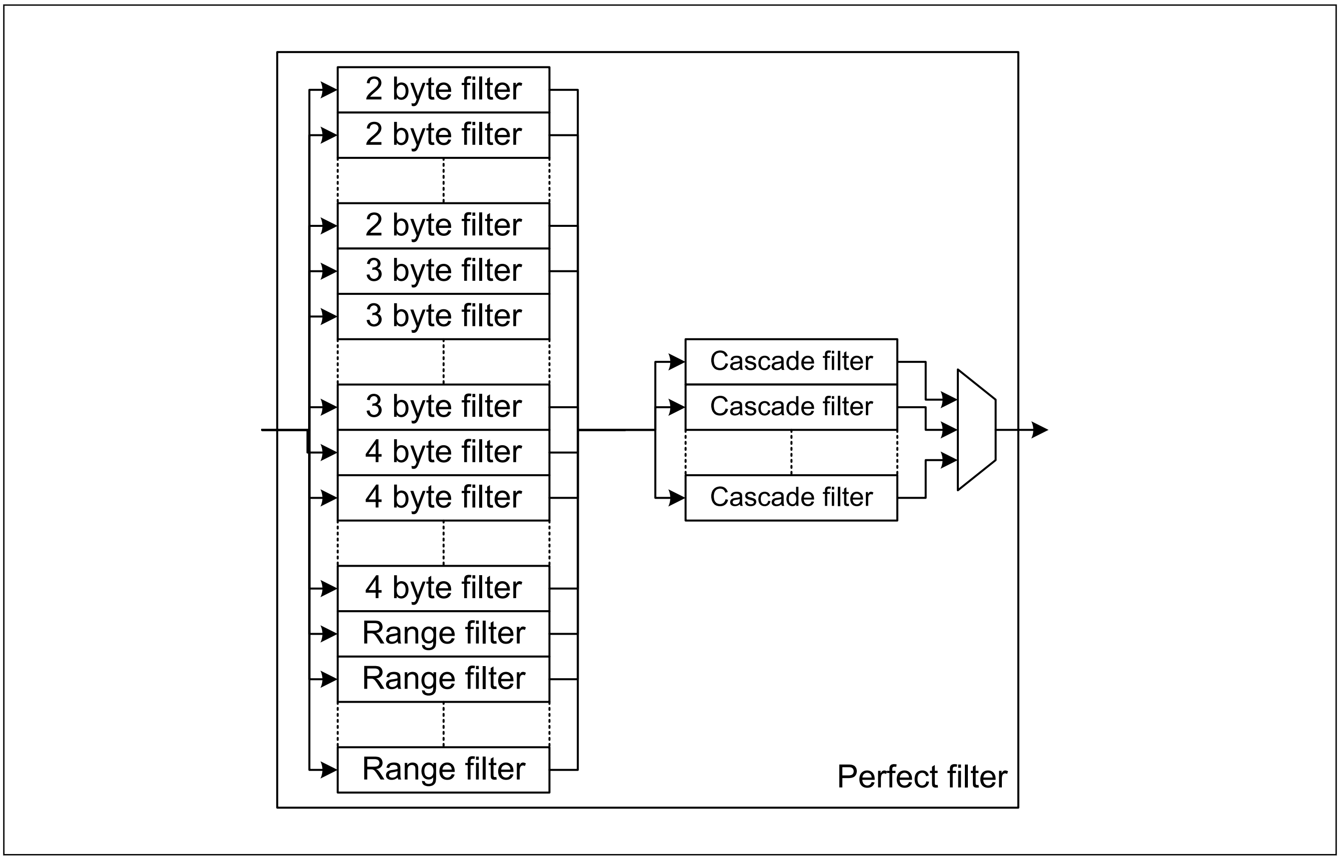


Figure 30.62 Perfect filter block diagram

The perfect filtering happens in two steps:

- The first step is done by 2-byte, 3-byte, 4-byte and range filters. Each one of these filters is independent of others but only able to filter a precise part of a frame.
- The second step is done by Cascade filters. Each one of the cascade filters can associate 7 filtering results from the first step to create a stream ID.

Note: HW: All frames smaller than 33 bytes (received with no command 1 from snooping bus, the 33 bytes are not including the TAG saved in the TAG RAM) will be considered has unknown frames by the perfect filter.

(1) 2-byte filters

A 2-byte filter aims at filtering a specific a portion of a frame and happens in two steps, the data extraction and the filtering.

(a) Data extraction for filtering

A 2-byte filter has two modes to extract the data from a frame. Figure 30.63 describes these modes using examples.

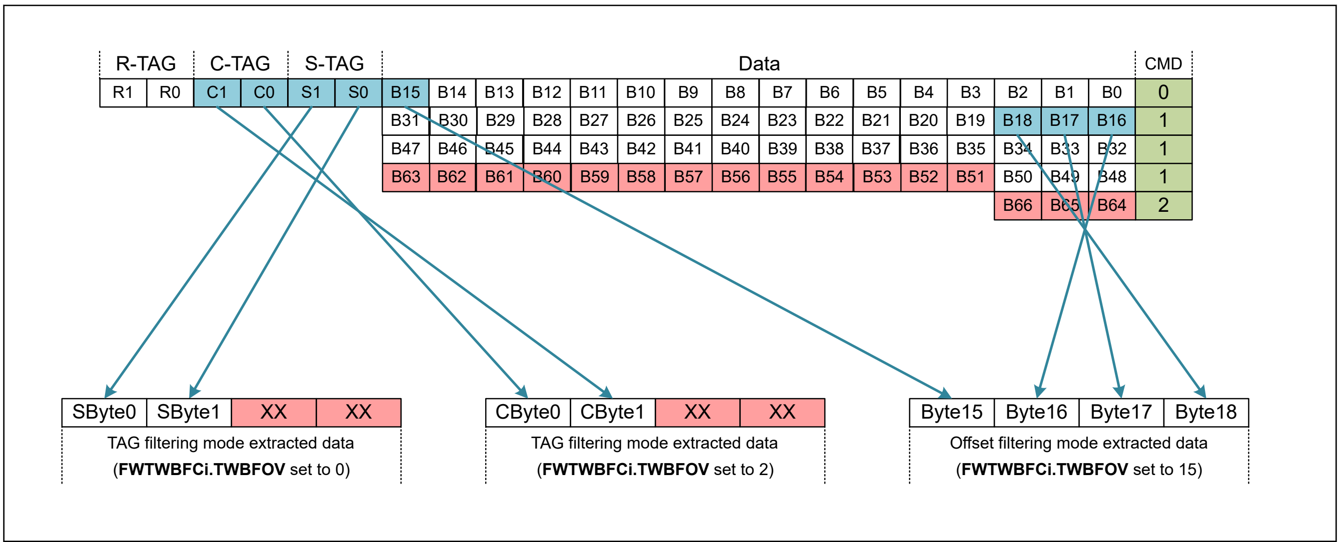


Figure 30.63 2-byte filter data extraction for filtering

In Offset filtering mode (FWTWBFCi.TWBFM == 0):

- Data bytes from byte FWTWBFCi.TWBFOV to byte FWTWBFCi.TWBFOV+3 are extracted for filtering.

In TAG filtering mode (FWTWBFCi.TWBFM == 1):

- If FWTWBFCi.TWBFOV is set to 0, VLAN S-TAG is extracted for filtering.
- If FWTWBFCi.TWBFOV is set to 2, VLAN C-TAG is extracted for filtering.

Note:
- HW: The data for filtering in 2-byte filters can only be extracted until the first three bytes in the penultimate beat of data (first three bytes of the last time CMD is equal to 1. In Figure 30.63, the bytes colored in red are the none-extractible bytes for filtering). I.e., the offset filtering will only be effective if the offset value is set until the last byte of the antepenultimate beat of data (the last before last time CMD is equal to 1. Byte 47 in Figure 30.63).
- HW: A filter with a setting that should use non-extractible data for filtering will always unmatched.
- HW: In TAG filtering mode, if S-TAG or C-TAG is not present in the frame for filtering, the value 0 will be extracted by the filter.

(b) Extracted data filtering

A 2-byte filter has three modes to filter extracted data. Figure 30.64 describes these modes.

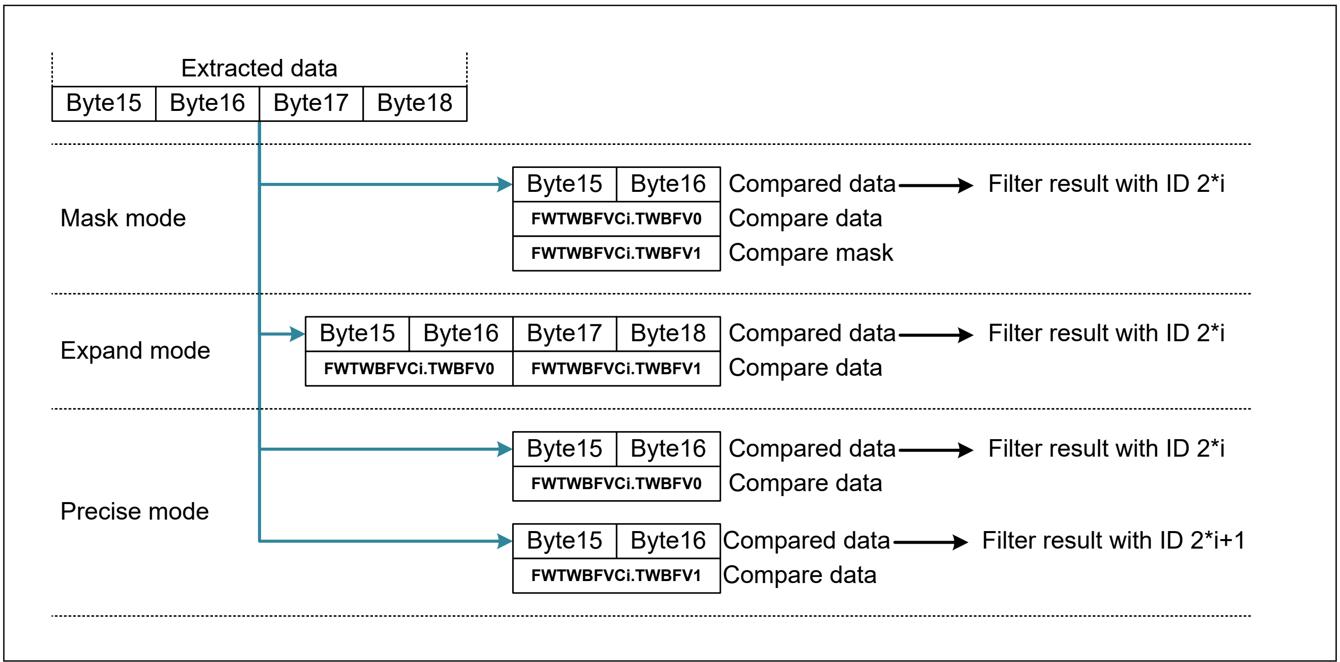


Figure 30.64 2-byte filter data extraction for filtering

In mask mode (FWTWBFCi.TWBFUM == 00b):

- The first two bytes of the extracted data are compared to FWTWBFVCi.TWBFV0 using FWTWBFVCi.TWBFV1 as a mask (when a bit is set to 1, the corresponding bit is not used for comparison). The result is associated to 2*i filter ID.

In Expand mode (FWTWBFCi.TWBFUM == 01b):

- The four-byte extracted data is compared to {FWTWBFVCi.TWBFV0,FWTWBFVCi.TWBFV1}. The result is associated to 2*i filter ID.

In Precise mode (FWTWBFCi.TWBFUM == 10b):

- The first two bytes of the extracted data are compared to FWTWBFVCi.TWBFV0. The result is associated to 2*i filter ID.
- The first two bytes of the extracted data are compared to FWTWBFVCi.TWBFV1. The result is associated to 2*i+1 filter ID.

(2) 3-byte filters

A 3-byte filter aims at filtering a specific a portion of a frame and happens in two steps, the data extraction and the filtering.

(a) Data extraction for filtering

A 3-byte filter has one mode to extract the data from a frame. Figure 30.65 describes this mode using an example.

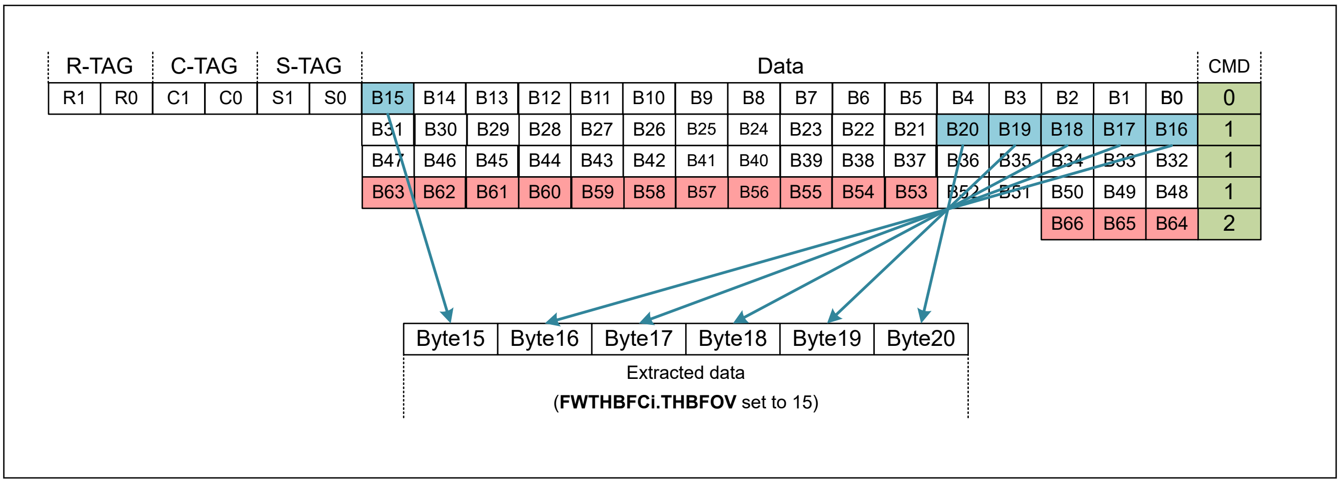


Figure 30.65 3-byte filter data extraction for filtering

For 3-byte filters:

- Data bytes from byte FWTHBFCi.THBFOV to byte FWTHBFCi.THBFOV+5 are extracted for filtering.

Note:
- HW: The data for filtering in 3-byte filters can only be extracted until the first five bytes in the penultimate beat of data (first five bytes of the last time CMD is equal to 1. In Figure 30.65, the bytes colored in red are the none-extractible bytes for filtering). I.e. the offset filtering will only be effective if the offset value is set until the last byte of the antepenultimate beat of data (the last before last time CMD is equal to 1. Byte 47 in Figure 30.65).
- HW: A filter with a setting that should use non-extractible data for filtering will always unmatched.

(b) Extracted data filtering

A 3-byte filter has three modes to filter extracted data. Figure 30.66 describes these modes.

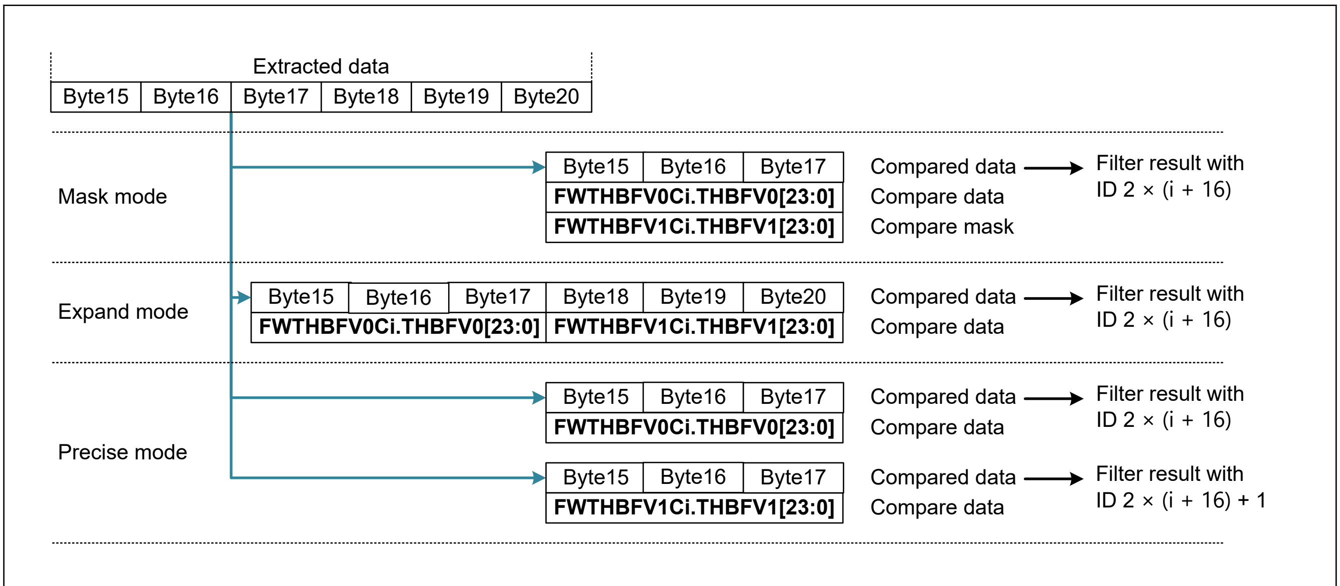


Figure 30.66 3-byte filter data extraction for filtering

In mask mode (FWTHBFCi.THBFUM[1:0] == 00b):

- The first three bytes of the extracted data are compared to FWTHBFV0Ci.THBFV0[23:0] using FWTHBFV1Ci.THBFV1[23:0] as a mask (when a bit is set to 1, the corresponding bit is not used for comparison). The result is associated to 2 × (i + 16) filter ID.

In Expand mode (FWTHBFCi.THBFUM[1:0] == 01b):

- The six-byte extracted data is compared to {FWTHBFV0Ci.THBFV0[23:0],FWTHBFV1Ci.THBFV1[23:0]}. The result is associated to 2 × (i + 16) filter ID.

In Precise mode (FWTHBFCi.THBFUM[1:0] == 10b):

- The first three bytes of the extracted data are compared to FWTHBFV0Ci.THBFV0[23:0]. The result is associated to 2 × (i + 16) filter ID.
- The first three bytes of the extracted data are compared to FWTHBFV1Ci.THBFV1[23:0]. The result is associated to 2 × (i + 16) + 1 filter ID.

(3) 4-byte filters

A 4-byte filter aims at filtering a specific a portion of a frame and happens in two steps, the data extraction and the filtering.

(a) Data extraction for filtering

A 4-byte filter has one mode to extract the data from a frame. Figure 30.67 describes this mode using an example.

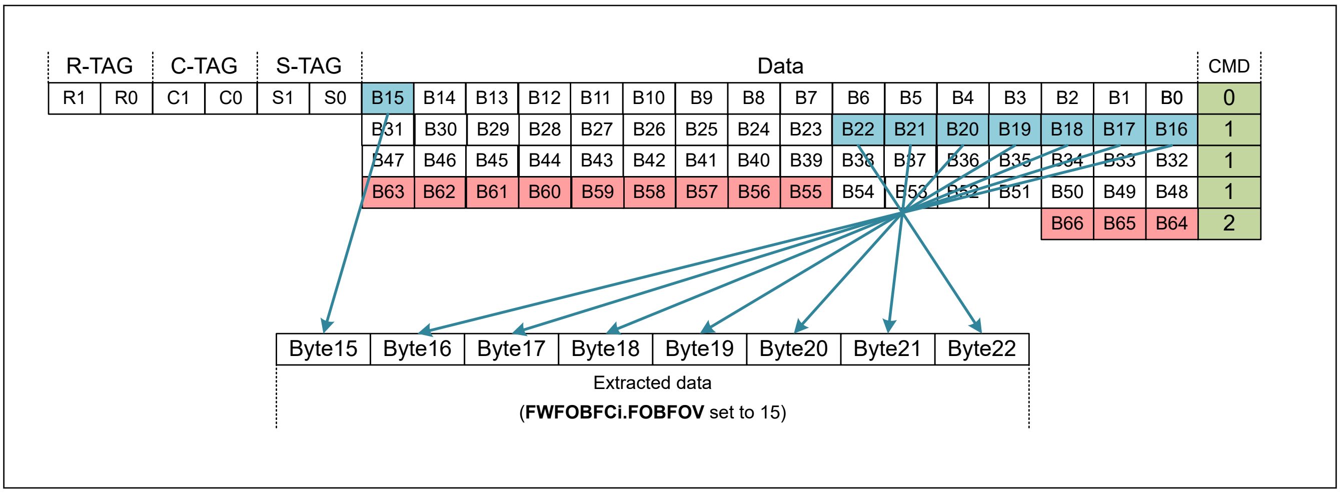


Figure 30.67 4-byte filter data extraction for filtering

For 4-byte filters:

- Data bytes from byte FWFOBFCi.FOBFOV to byte FWFOBFCi.FOBFOV+7 are extracted for filtering.

Note:
- HW: The data for filtering in 4-byte filters can only be extracted until the first seven bytes in the penultimate beat of data (first seven bytes of the last time CMD is equal to 1. In Figure 30.67, the bytes colored in red are the none-extractible bytes for filtering). I.e. the offset filtering will only be effective if the offset value is set until the last byte of the antepenultimate beat of data (the last before last time CMD is equal to 1. Byte 47 in Figure 30.67).
- HW: A filter with a setting that should use non-extractible data for filtering will always unmatched.

(b) Extracted data filtering

A 4-byte filter has three modes to filter extracted data. Figure 30.68 describes these modes.

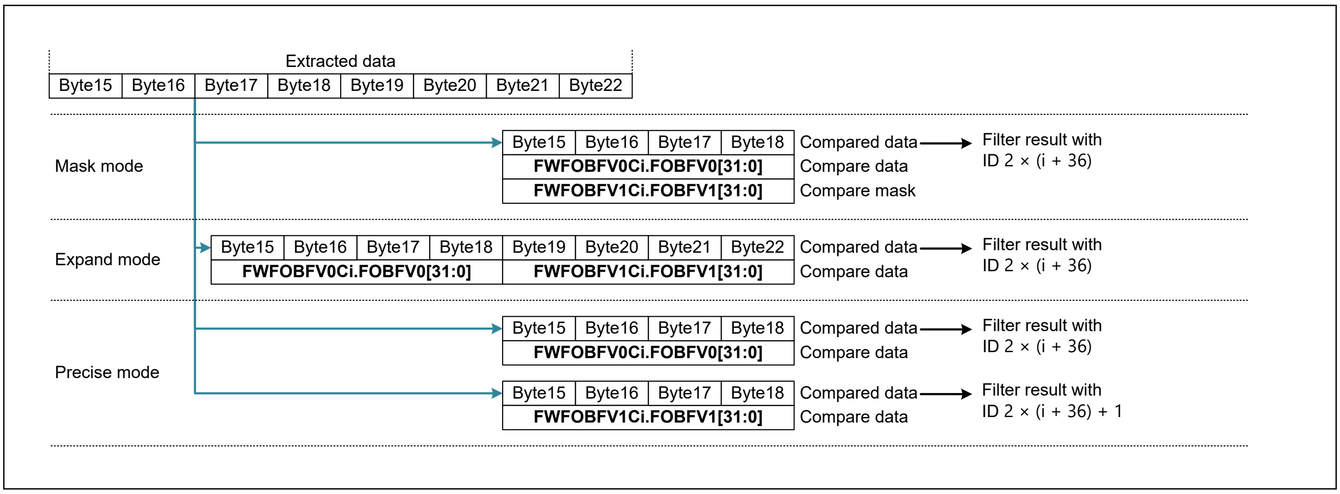


Figure 30.68 4-byte filter data extraction for filtering

In mask mode (FWFOBFCi.FOBFUM[1:0] == 00b):

- The first four bytes of the extracted data are compared to FWFOBFV0Ci.FOBFV0[31:0] using FWFOBFV1Ci.FOBFV1[31:0] as a mask (when a bit is set to 1, the corresponding bit is not used for comparison). The result is associated to 2 × (i + 36) filter ID.

In Expand mode (FWFOBFCi.FOBFUM[1:0] == 01b):

- The eight-byte extracted data is compared to {FWFOBFV0Ci.FOBFV0[31:0],FWFOBFV1Ci.FOBFV1[31:0]}. The result is associated to 2 × (i + 36) filter ID.

In Precise mode (FWFOBFCi.FOBFUM[1:0] == 10b):

- The first three bytes of the extracted data are compared to FWFOBFV0Ci.FOBFV0[31:0]. The result is associated to 2 × (i + 36) filter ID.
- The first three bytes of the extracted data are compared to FWFOBFV1Ci.FOBFV1[31:0]. The result is associated to 2 × (i + 36) + 1 filter ID.

(4) Range filters

A Range filter aims at filtering a specific a portion of a frame and happens in two steps, the data extraction and the filtering.

(a) Data extraction for filtering

A Range filter has two modes to extract the data from a frame. Figure 30.69 describes these modes using examples.

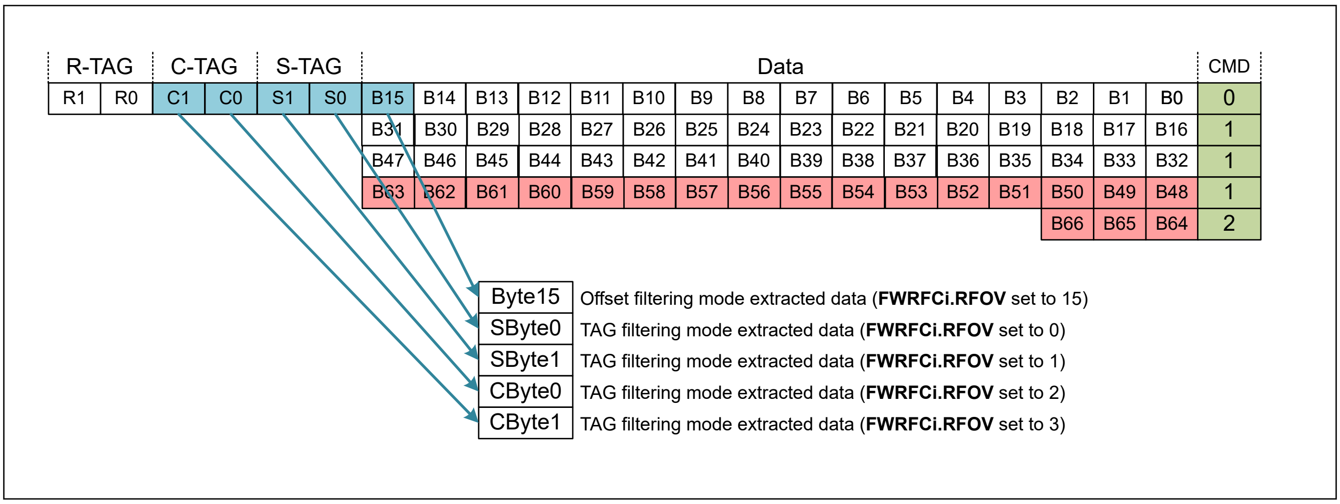


Figure 30.69 Range filter data extraction for filtering

In Offset filtering mode (FWRFCi.RFM == 0):

- Data byte FWRFCi.RFOV is extracted for filtering.

In TAG filtering mode (FWRFCi.RFM == 1):

- If FWRFCi.RFOV is set to 0, VLAN S-TAG VID upper 4 bits are extracted for filtering (PCP and DEI are fixed to 0).
- If FWRFCi.RFOV is set to 1, VLAN S-TAG VID lower 8 bits are extracted for filtering.
- If FWRFCi.RFOV is set to 2, VLAN C-TAG VID upper 4 bits are extracted for filtering (PCP and DEI are fixed to 0).
- If FWRFCi.RFOV is set to 3, VLAN C-TAG VID lower 8 bits are extracted for filtering.

Restrictions:

Note:
- HW: PCP and DEI values in C-TAGs and S-TAGs are ignored by the range filter (Set to 0).
- HW: The data for filtering in range filters can only be extracted until the antepenultimate beat of data (the last before last time CMD is equal to 1. In Figure 30.69, the bytes colored in red are the none-extractible bytes for filtering). I.e. the offset filtering will only be effective if the offset value is set until the last byte of the antepenultimate beat of data (the last before last time CMD is equal to 1. Byte 47 in Figure 30.69).
- HW: A filter with a setting that should use non-extractible data for filtering will always unmatched.

(b) Extracted data filtering

A Range filter has one mode to filter extracted data. Figure 30.70 describes this mode.

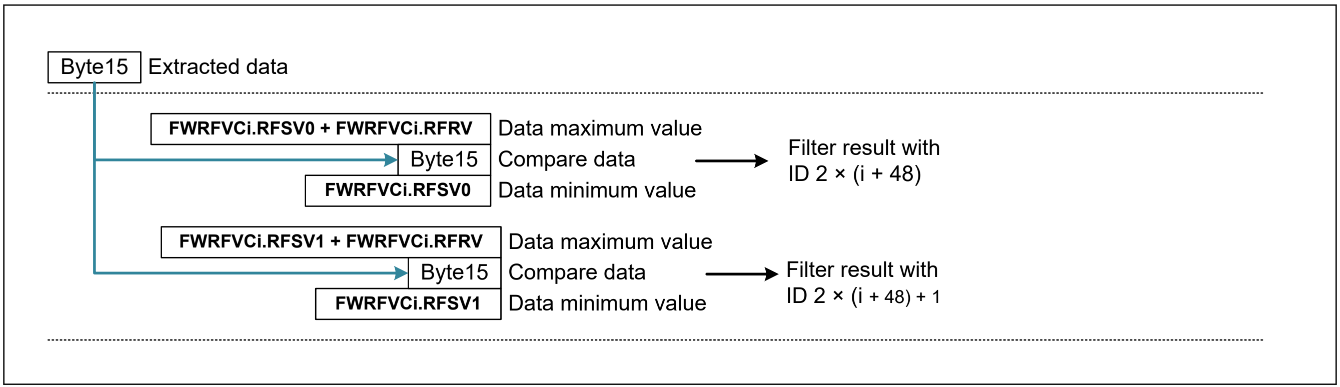


Figure 30.70 Range filter data extraction for filtering

For Range filters:

- The one-byte extracted data is checked to be included in [FWRFVCi.RFSV0: FWRFVCi.RFSV0 + FWRFVCi.RFRV]. The result is associated to 2 × (i + 48) filter ID.
- The one-byte extracted data is checked to be included in [FWRFVCi.RFSV1: FWRFVCi.RFSV1 + FWRFVCi.RFRV]. The result is associated to 2 × (i + 48) + 1 filter ID.

(5) Cascade filters

A Cascade filter aims at gathering several filter results (from 2-byte, 3-byte, 4byte and Range filters) to regroup them to create a stream ID. The number of cascade filters defines the number of possible stream IDs in the perfect filter 16.

Functions:

- FWCFMCij register is used to map until 7 filters results to cascade filter i.
- FWCFCi register is used to select from which port the stream is expected and with which type of frame (p or e)
- If several cascade filters match the same frame, the cascade filter with the highest number will be selected.
- If a cascade filter matches a frame, a frame format code of 3 bits equal to 0 is associated to the frame.

Cascade filter stream ID:

- When cascade filter i matches a frame, the stream ID associated to this frame will be a 131-bit stream ID equal to {frame format code,128’di}.

(6) Perfect filter utilization example

Figure 30.71 contains an example of perfect filter utilization. In this example, the user wants to filter a frame which:

- Has a stream ID based on MAC destination address, and VLAN C-TAG.
- Has a MAC destination address equal to ‘hAABBCCDDEEFF.
- Has a VLAN C-TAG equal to ABCD.
- Is known to be and IPv4 frame (IPv4 type is added in filtering for security).

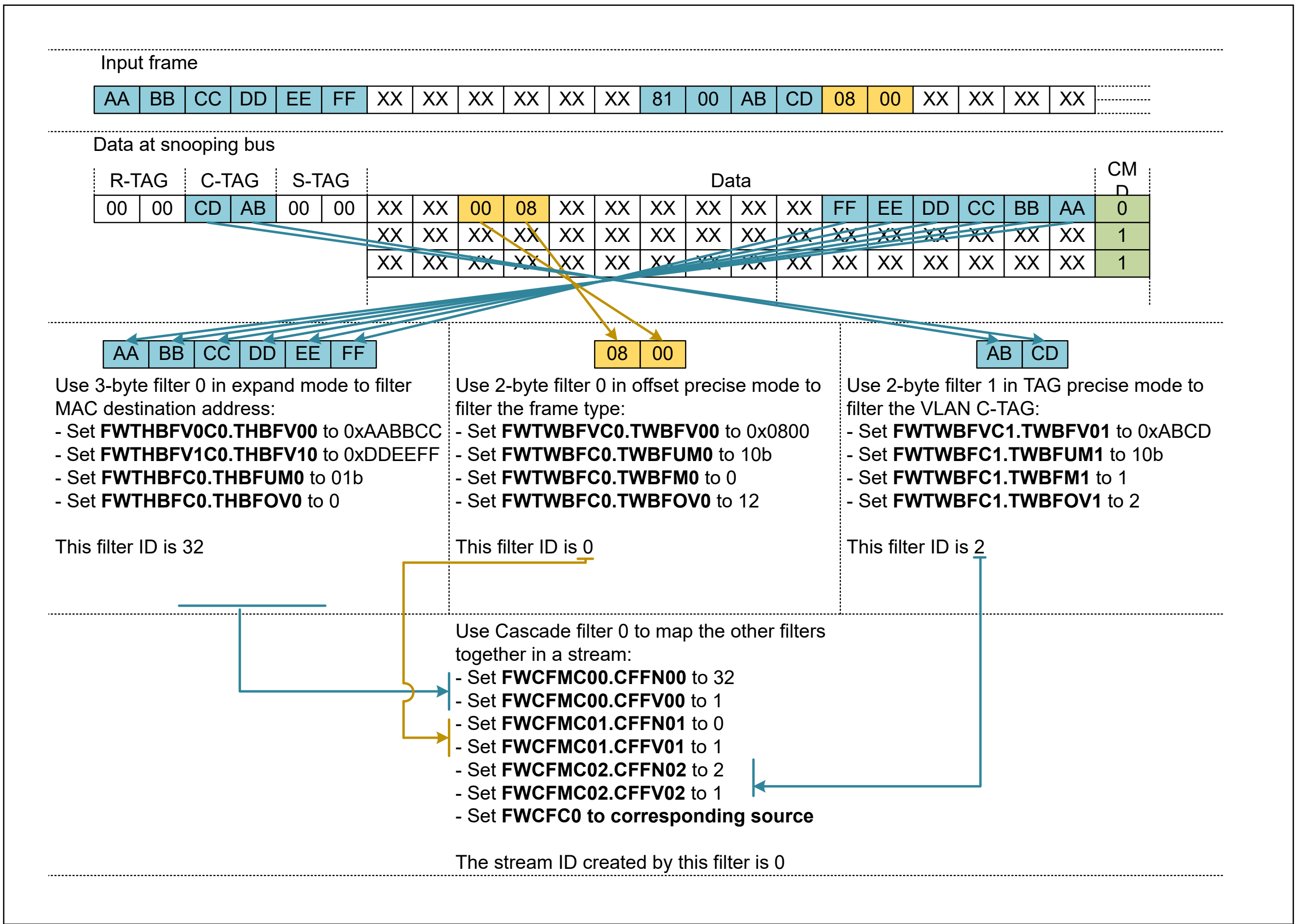


Figure 30.71 Perfect filter utilization example

30.5.2.4.2 IPv4/IPv6 Hash Stream Filter

The Hash filter is used to identify streams. This filter has the advantage to be able to generate a big number of streams but, because of the use of hash, can have some stream ID collision problem (that should be resolved using the perfect filter).

The Hash filter block diagram is represented in Figure 30.72.

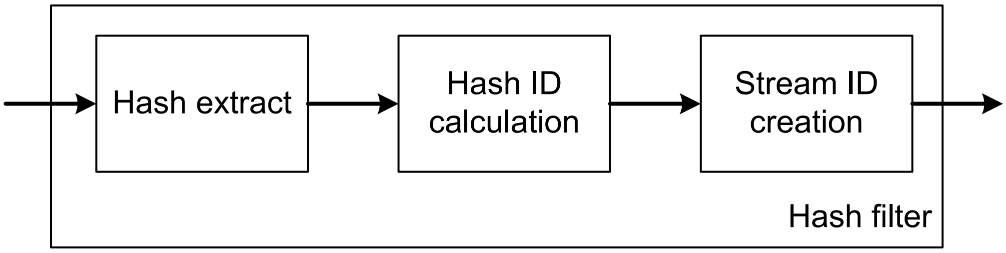


Figure 30.72 Hash filter block diagram

The Hash filtering happens in Three steps:

- The first step is done by Hash extract. It consists in extracting all the possible field for the IPv4/IPv6 stream ID creation.
- The second step is done by Hash ID calculation. It consists in calculating the hash ID associated to each frame depending on the extracted data and the fields chosen by user for Hash.
- The third step is done by Stream ID creation. It consists in the stream ID assembling depending on the fields chosen by user.

(1) Hash extract

The Hash extract is used to extract the Stream IDs from IPv4 and IPv6 frames to create their associated stream ID. The extraction happens on six formats which are IPv4, IPv4/TCP, IPv4/UDP, IPv6, IPv6/TCP and IPv6/UDP.

Note: • HW: Hash extract will ignore any data received from Fabric [MFAB] with command 2.

(a) IPv4 detection and data extraction

Figure 30.73 describes how IPv4 detection and data extraction happens. This data extraction happens only if the frame did not match a cascade filter, if the frame format decode condition are satisfied and if FWPCi0.IP4OE is set for the corresponding source port. In this case, a frame format code of 3 bits equal to 1 is associated to the frame.

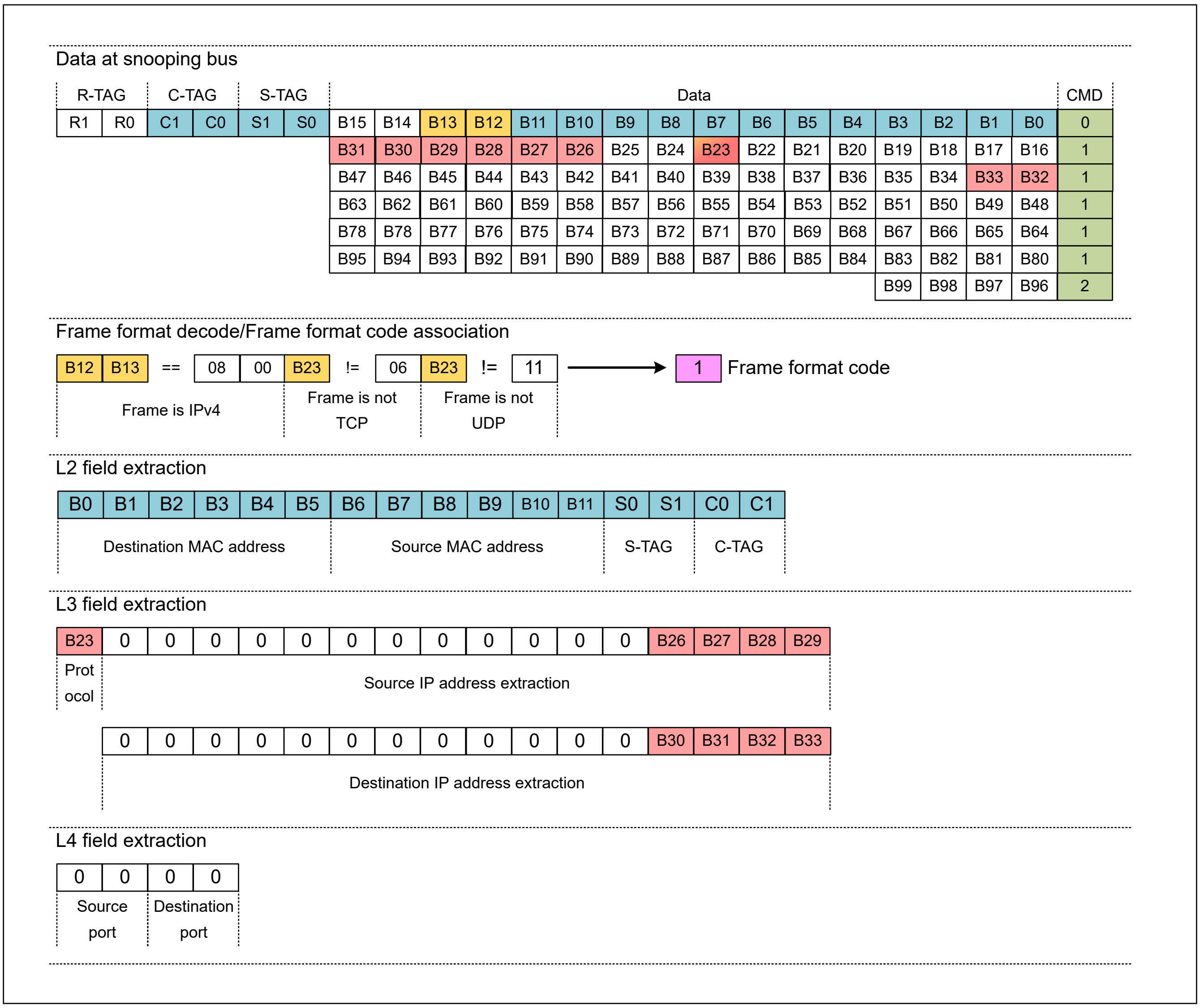


Figure 30.73 IPv4 detection and data extraction

(b) IPv4/UDP detection and data extraction

Figure 30.74 describes how IPv4/UDP detection and data extraction happens. This data extraction happens only if the frame did not match a cascade filter, if the frame format decode condition are satisfied and if FWPCi0.IP4UE is set for the corresponding source port. In this case, a frame format code of 3 bits equal to 2 is associated to the frame.

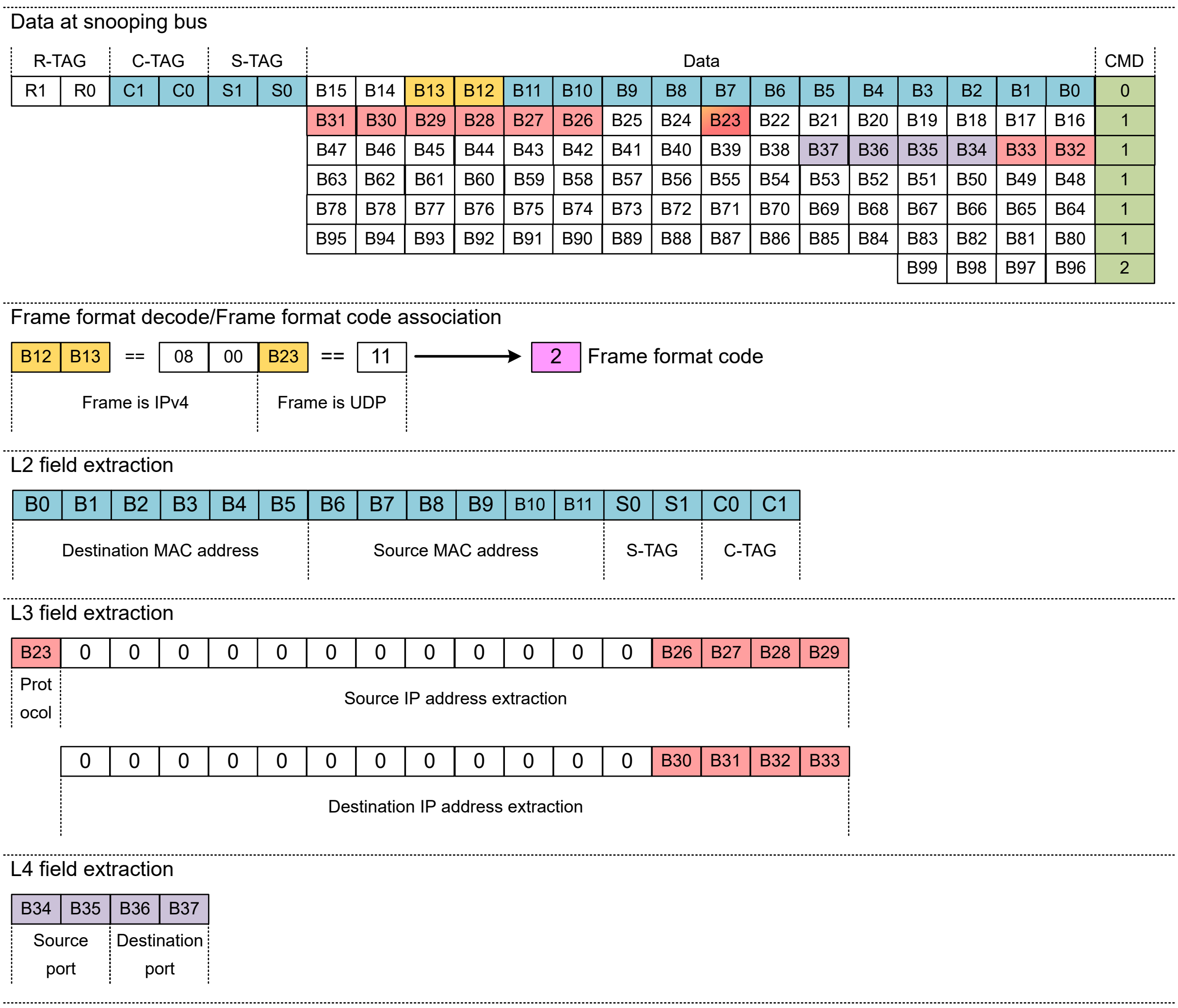


Figure 30.74 IPv4/UDP detection and data extraction

Function:

- The L4 extraction function takes in account the IPv4 HeaderLength field. Figure 30.74 only describes the case where HeaderLength is equal to 5 (default value).

(c) IPv4/TCP detection and data extraction

Figure 30.75 describes how IPv4/TCP detection and data extraction happens. This data extraction happens only if the frame did not match a cascade filter, if the frame format decode condition are satisfied and if FWPCi0.IP4TE is set for the corresponding source port. In this case, a frame format code of 3 bits equal to 3 is associated to the frame.

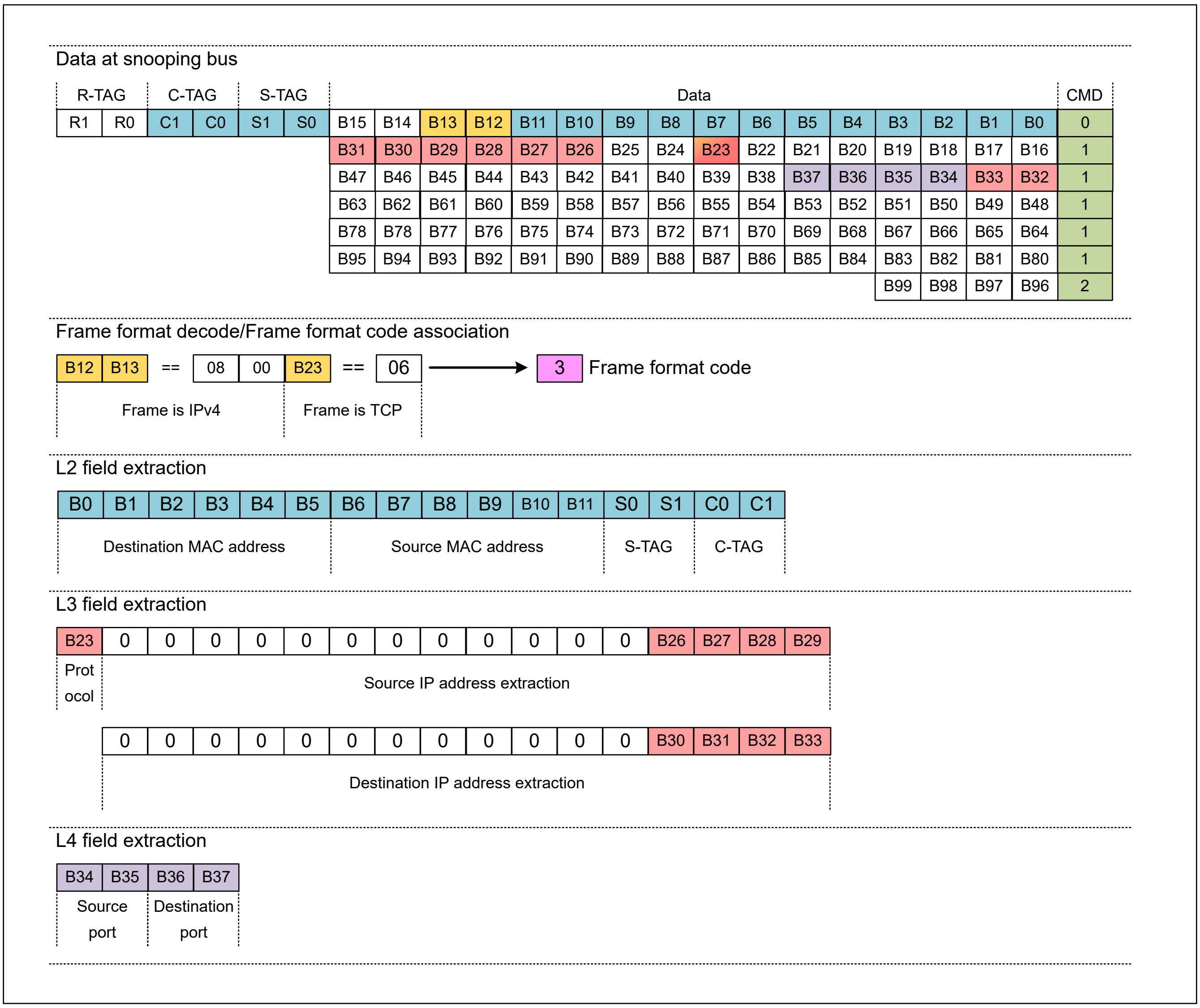


Figure 30.75 IPv4/TCP detection and data extraction

Function:

- The L4 extraction function takes in account the IPv4 HeaderLength field. Figure 30.75 only describes the case where HeaderLength is equal to 5 (default value).

(d) IPv6 detection and data extraction

Figure 30.76 describes how IPv6 detection and data extraction happens. This data extraction happens only if the frame did not match a cascade filter, if the frame format decode condition are satisfied and if FWPCi0.IP6OE is set for the corresponding source port. In this case, a frame format code of 3 bits equal to 4 is associated to the frame.

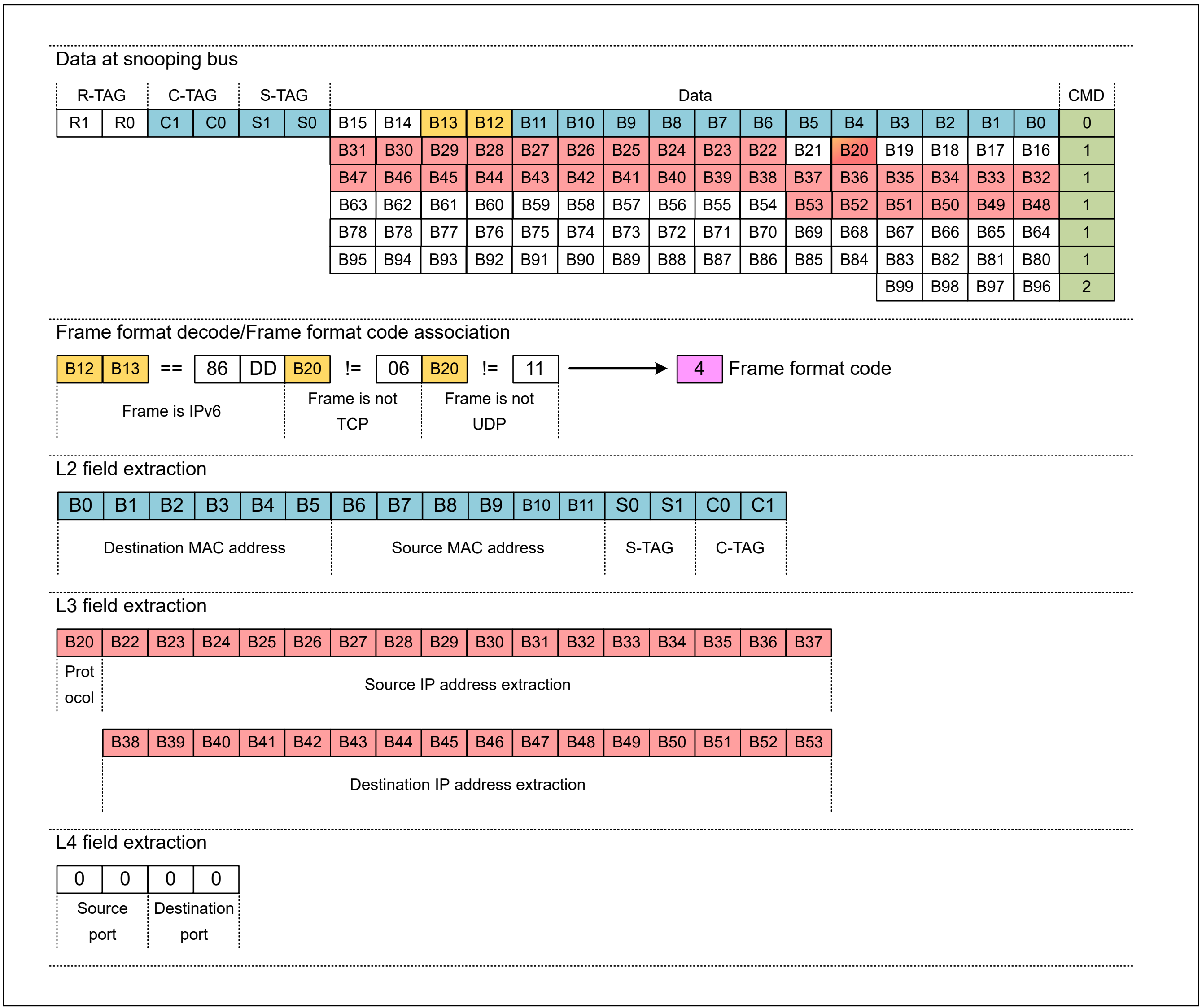


Figure 30.76 IPv6 detection and data extraction

(e) IPv6/UDP detection and data extraction

Figure 30.77 describes how IPv6/UDP detection and data extraction happens. This data extraction happens only if the frame did not match a cascade filter, if the frame format decode condition are satisfied and if FWPCi0.IP6UE is set for the corresponding source port. In this case, a frame format code of 3 bits equal to 5 is associated to the frame.

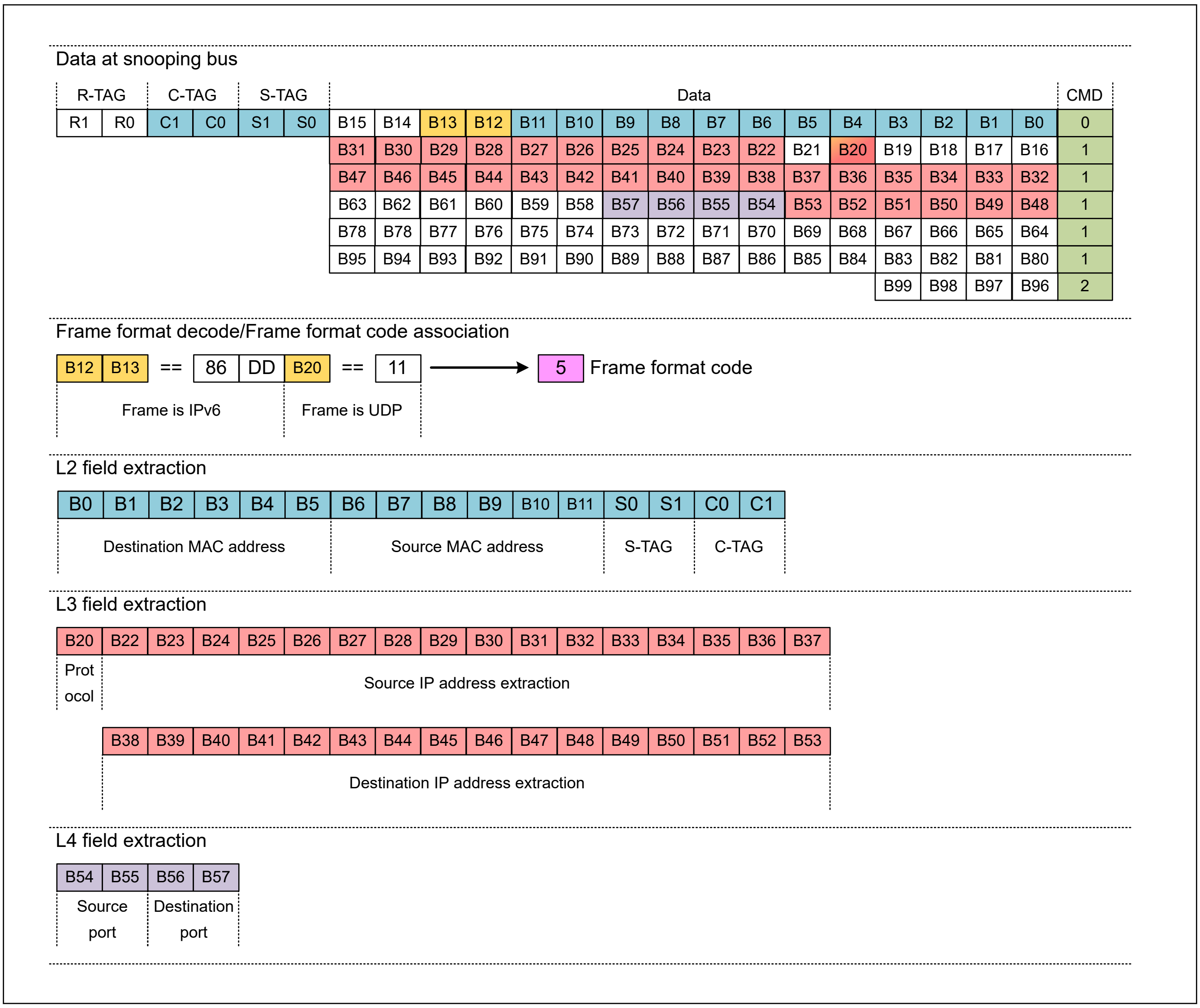


Figure 30.77 IPv6/UDP detection and data extraction

(f) IPv6/TCP detection and data extraction

Figure 30.78 describes how IPv6/TCP detection and data extraction happens. This data extraction happens only if the frame did not match a cascade filter, if the frame format decode condition are satisfied and if FWPCi0.IP6TE is set for the corresponding source port. In this case, a frame format code of 3 bits equal to 6 is associated to the frame.

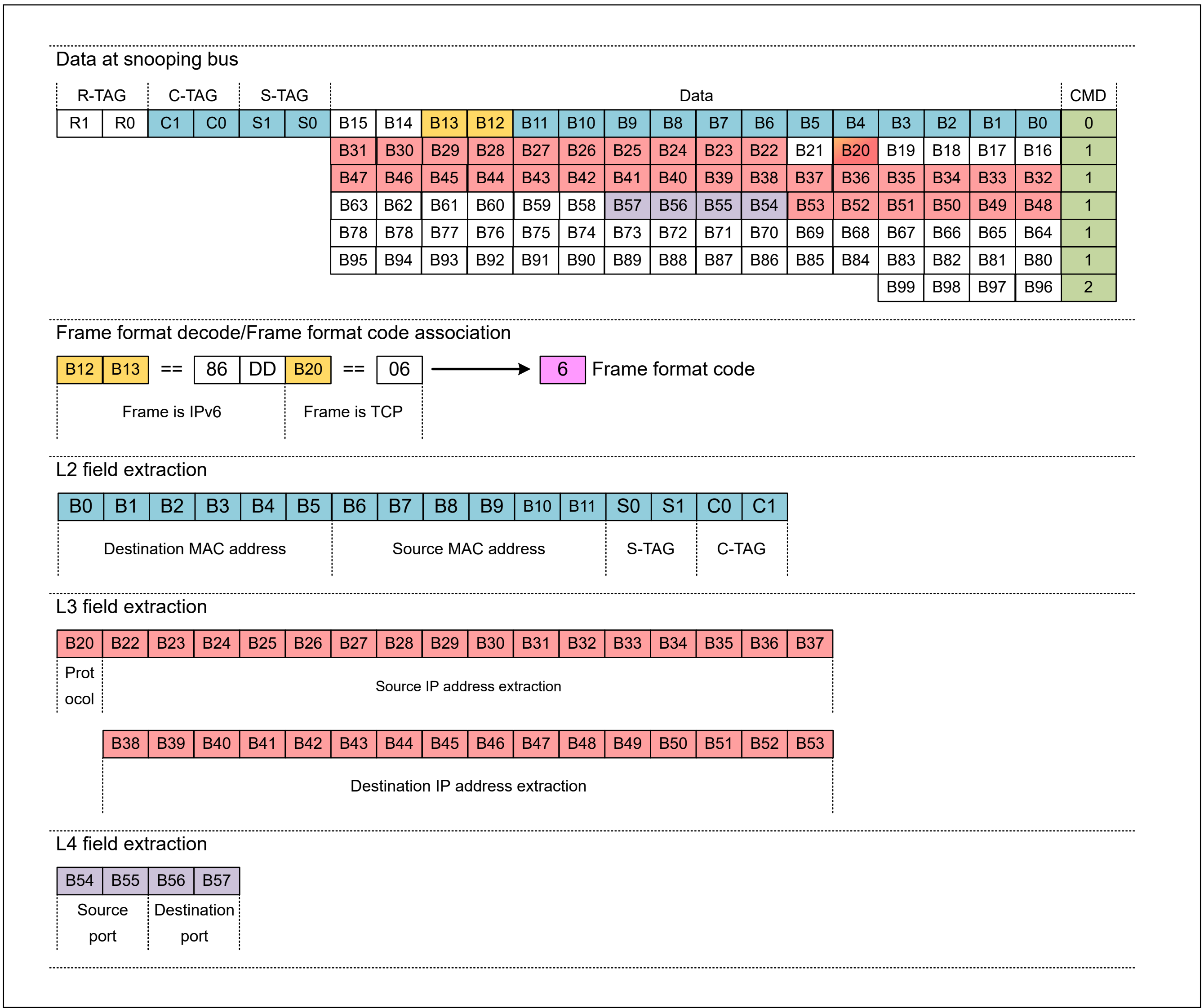


Figure 30.78 IPv6/TCP detection and data extraction

(2) Hash ID calculation

The Hash ID calculation is used to select the interesting fields for the Hash ID (security purpose) and to calculate the hash ID.

(a) Hash equation

Refer to FWSFHEC register explanation.

(b) IPv4 Hash ID calculation

The IPv4 Hash ID calculation is done depending on extracted fields in Hash extract. The calculation is described in Figure 30.79.

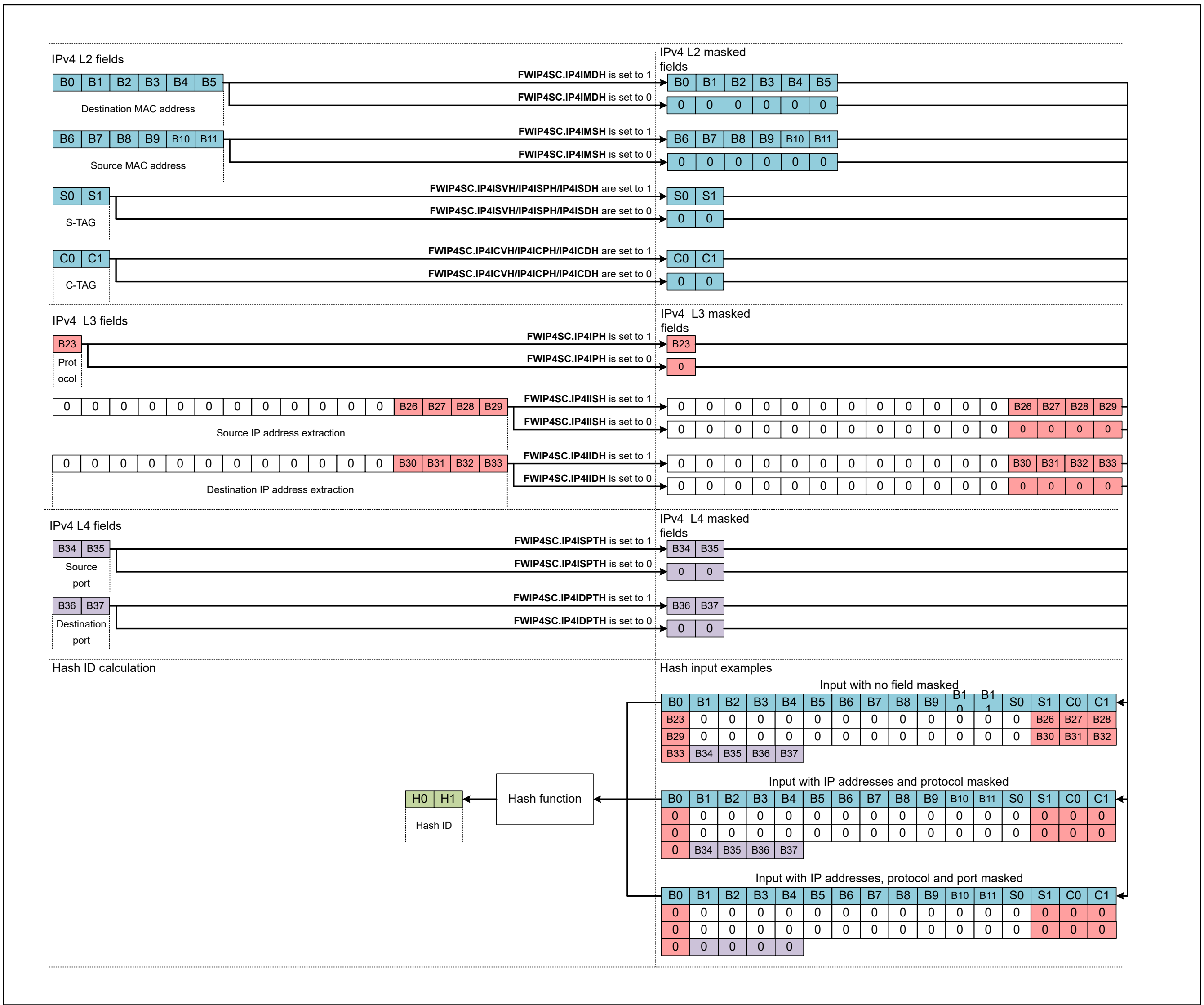


Figure 30.79 IPv4 Hash ID calculation

Functions:

- In Figure 30.79, The C-TAG and S-TAG masking is represented done at once, but masking is possible independently on PCP, DEI and VID.
- Setting all the field inclusion bits for Hash to 0 will fix the Hash ID to 0 in the stream (FWIP4SC.IP4IDPTH/IP4ISPTH/IP4IPH/IP4IIDH/IP4IISH/IP4ICDH/IP4ICPH/IP4ICVH/IP4ISDH/IP4ISPH/IP4ISVH/IP4IMSH/IP4IMDH).
- Setting the hash equation to 0 will fix the Hash ID to 0 in the stream (FWSFHEC.IP4HE).

(c) IPv6 Hash ID calculation

The IPv6 Hash ID calculation is done depending on extracted fields in Hash extract. The calculation is described in Figure 30.80.

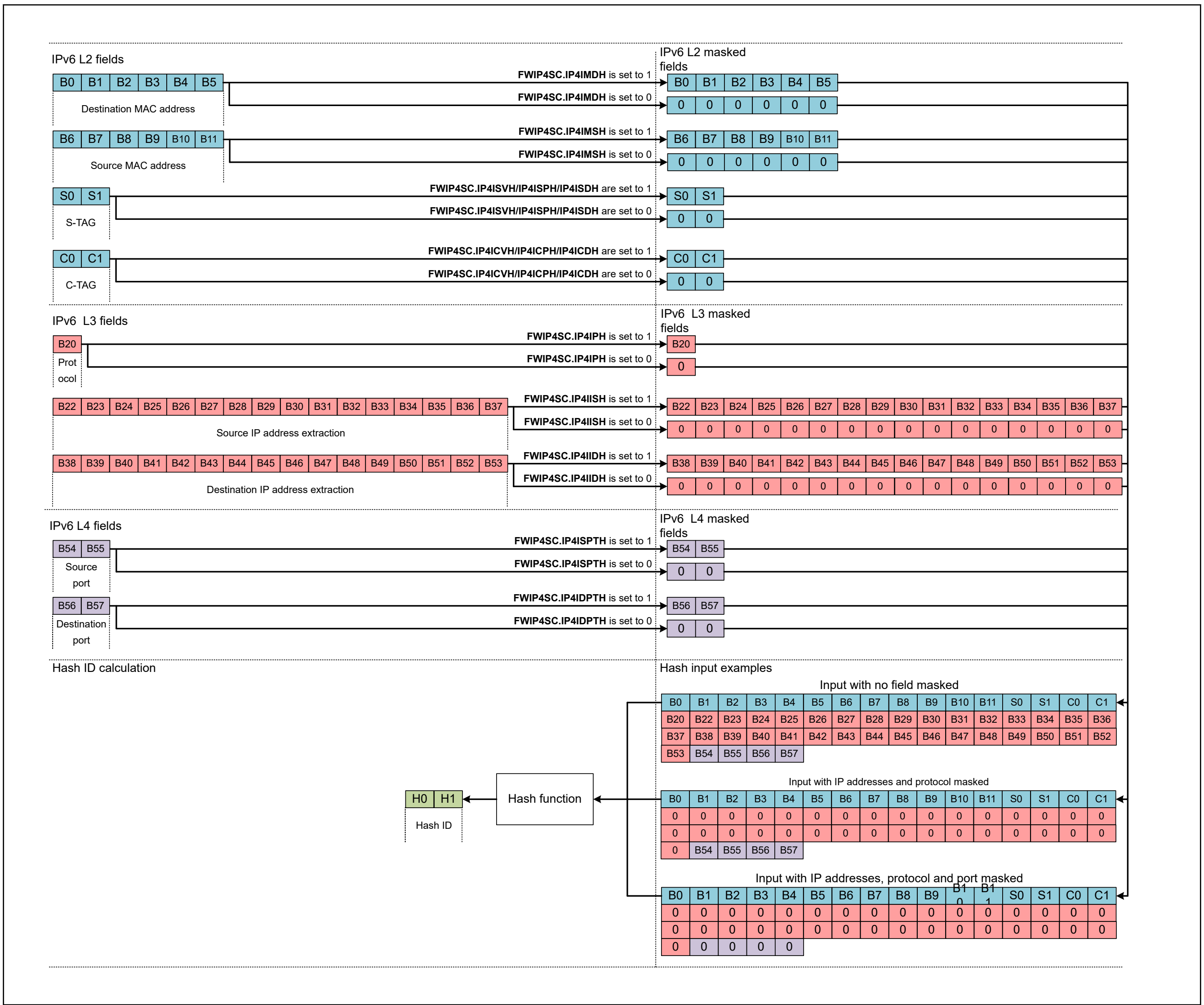


Figure 30.80 IPv6 Hash ID calculation

Functions:

- In Figure 30.80, The C-TAG and S-TAG masking is represented done at once, but masking is possible independently on PCP, DEI and VID.
- Setting all the field inclusion bits for Hash to 0 will fix the Hash ID to 0 in the stream (FWIP6SC.IP6IDPTH/IP6ISPTH/IP6IPH/IP6IIDH/IP6IISH/IP6ICDH/IP6ICPH/IP6ICVH/IP6ISDH/IP6ISPH/IP6ISVH/IP6IMSH/IP6IMDH)
- Setting the hash equation to 0 will fix the Hash ID to 0 in the stream (FWSFHEC.IP6HE)

(3) Stream ID creation

(a) IPv4 Stream ID creation

The IPv4 Stream ID is assembled from some of the extracted fields in Hash extract, the Hash ID calculated in Hash ID calculation and the frame format code. The stream ID creation is described in Figure 30.81.

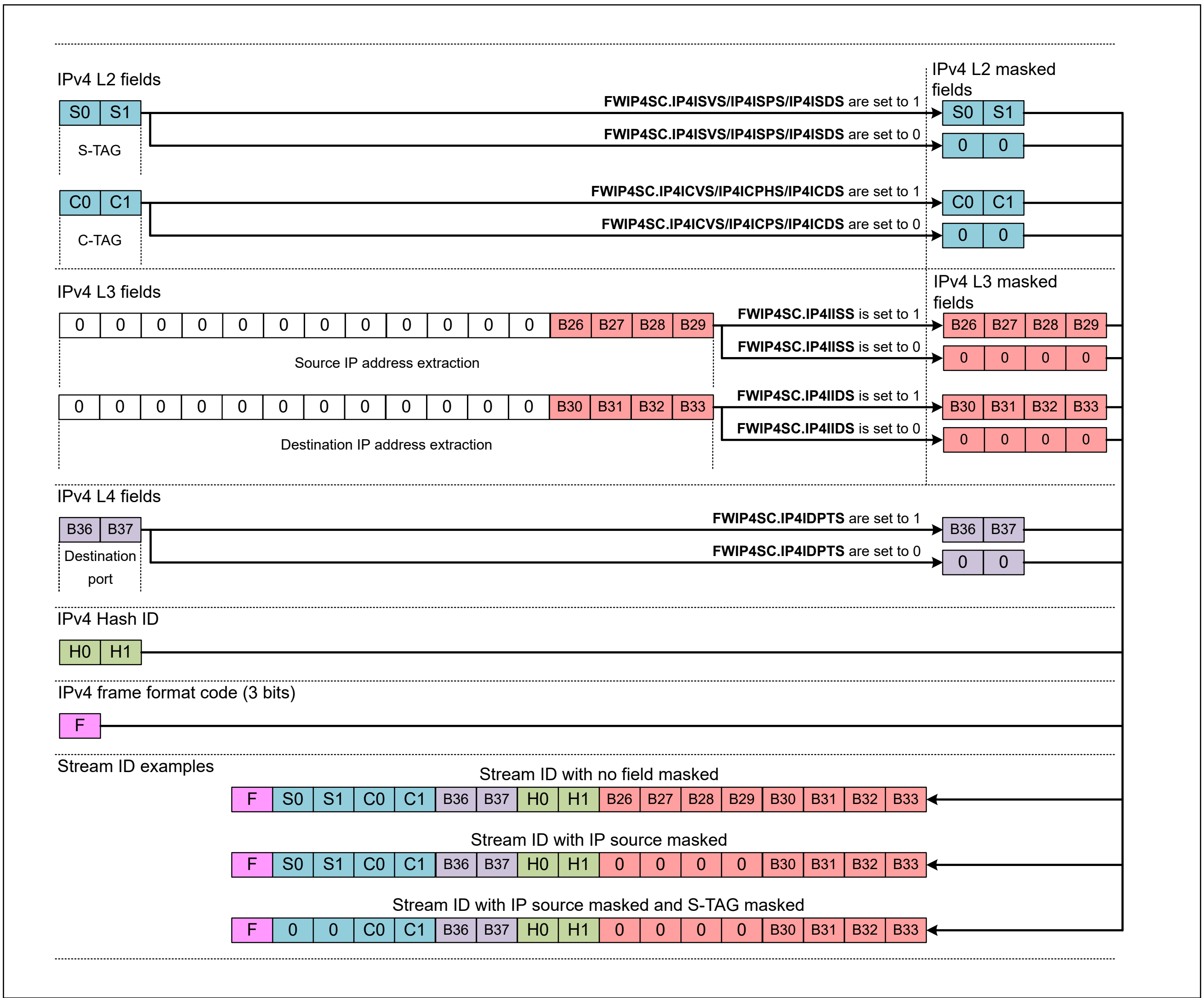


Figure 30.81 IPv4 Stream ID creation

Functions:

- In Figure 30.81, The C-TAG and S-TAG masking is represented done at once, but masking is possible independently on PCP, DEI and VID.
- The Stream ID field (FWLTHTL1-4) should be configured as 0 related to disabled part (bits) with “Stream in (FWIP4SC.IP4IDPTS/IP4IIDS/IP4IISS/IP4ICDS/IP4ICPS/IP4ICVTS/IP4ISDS/IP4ISPS/IP4ISVS) = 0x0”

(b) IPv6 Stream ID creation

The IPv6 Stream ID is assembled from some of the extracted fields in Hash extract, the Hash ID calculated in Hash ID calculation and the frame format code. The stream ID creation is described in Figure 30.82.

Figure 30.82 IPv6 Stream ID creation

Functions:

- In Figure 30.82, The C-TAG and S-TAG masking is represented done at once, but masking is possible independently on PCP, DEI and VID.
- In Figure 30.82, IP 0 is extracted from IPv6 source IP address and IP 1 is extracted from destination IP address, but it both can extract their 4 bytes from any of the IP addresses. This setting is done through FWIP6OC register.
- The Stream ID field (FWLTHTL1-4) should be configured as 0 related to disabled part (bits) with “Stream in (FWIP6SC.IP6IDPTS/IP6II1S/IP6II0S/IP6ICDS/IP6ICPS/IP6ICVS/IP6ISDS/IP6ISPS/IP6ISVS) = 0x0”

(4) Software Hash ID calculation

Because hash ID calculation is complex to implement in software, a hardware accelerator is available for hash ID calculation. Figure 30.83 gives software hash ID calculation examples.

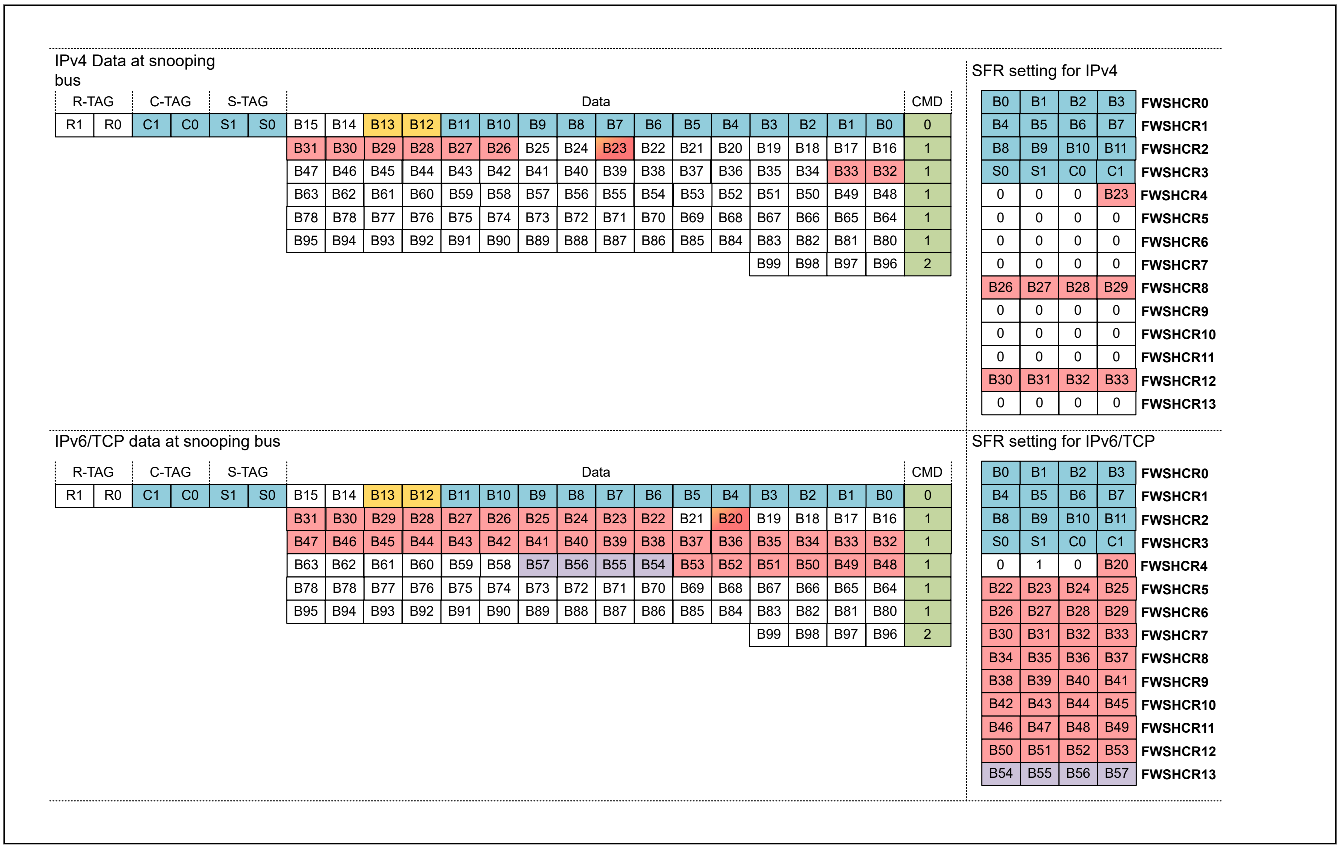


Figure 30.83 Software hash ID calculation example

Note: SW: For software hash ID calculation, the software should set to 0 the fields that are masked by the Hash ID calculation function.

30.5.2.4.3 L2 Stream Filter

The L2 stream filter is used to create a stream based on L2 fields. This stream creating happens only if the frame did not match a cascade filter or an IPv4/IPv6 hash stream filter and if FWPCi0.L2SE is set for the corresponding source port. In this case, a frame format code of 3 bits equal to 7 is associated to the frame. Figure 30.84 describes how the L2 stream ID is created.

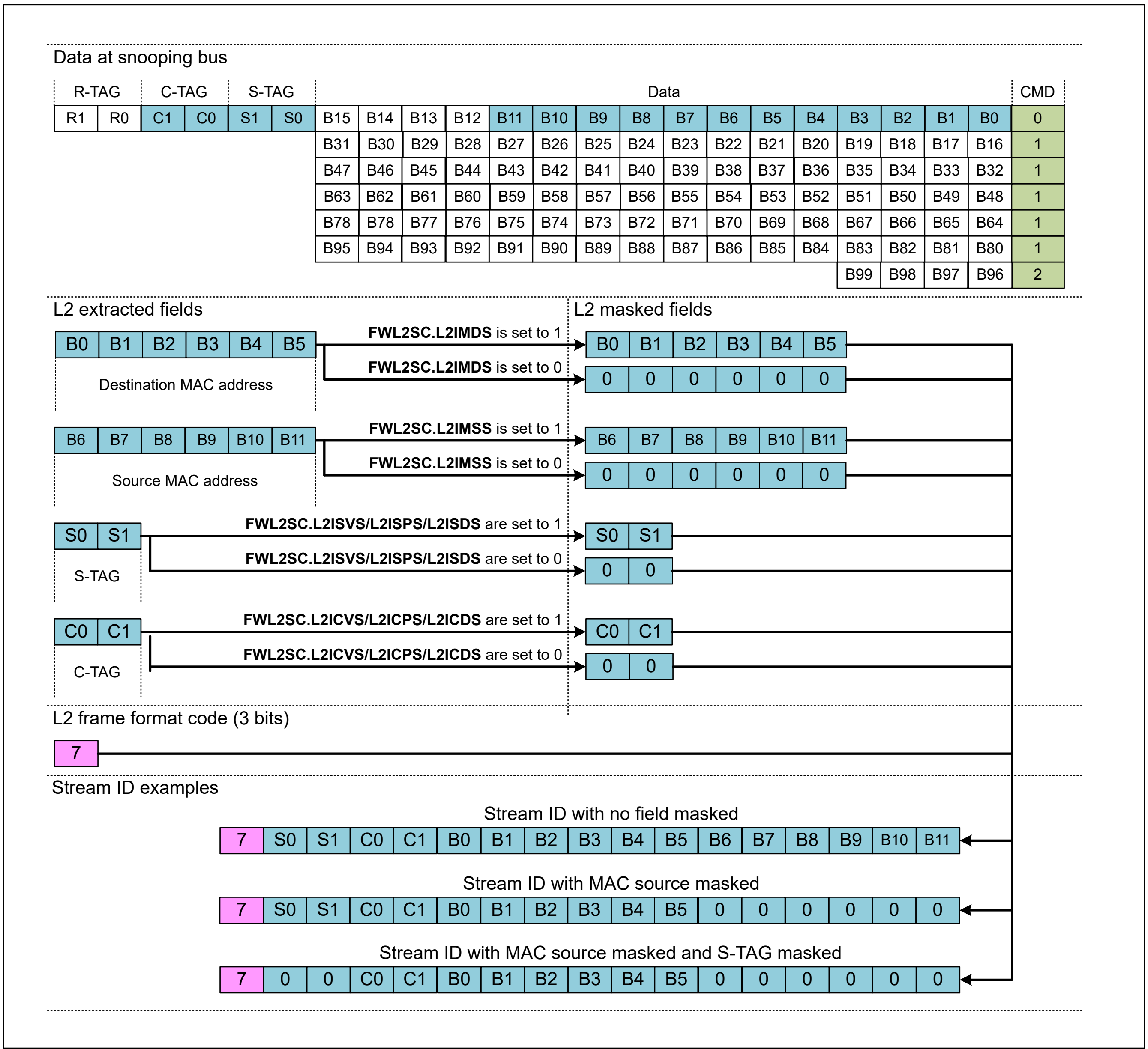


Figure 30.84 L2 stream ID creation

Functions:

- In Figure 30.84, The C-TAG and S-TAG masking is represented done at once, but masking is possible independently on PCP, DEI and VID.
- A frame which did not match a cascade filter, an IPv4/IPv6 hash stream filter or the L2 stream filter is considered an unknown frame.
- The Stream ID field (FWLTHTL1-4) should be configured as 0 related to disabled part (bits) with “Stream in (FWL2SC.L2ICDS/L2ICPS/L2ICVS/L2ISDS/L2ISPS/L2ISVS/L2IMSS/L2IMDS) = 0x0”.

30.5.2.4.4 L3 Hash

L3 hash aims at searching/learning/reading L3 rules thanks to a programmable hash. The hash algorithm is able to solve all the collisions in hardware, so it does not need any software intervention to operate.

L3 hash is divided in the following items:

- Rule format: Determines what information will be store along with the stream ID in the table.
- SW Learning: Function to add/suppress an entry from the L3 table by software.

- Hash maximum collision number: Avoids hash search to slow down the forwarding pipeline and apply back pressure on the ingress ports.
- Hash equation: Function to set hash calculation to minimize collisions.
- SW Searching: Function to find the forwarding/routing/filtering information related to a stream by software.
- SW Reading: Function to read the entries in the L3 table by software.
- HW searching: Function to find the forwarding/routing/filtering information related to a stream by forwarding mechanism.

(1) Rule format

The L3 table is used to store L3 rules. A L3 rule contains the information the frame forwarding, routing and filtering. All the fields in this table, if quoted, will be written L3.{Field name}. Table 30.28 describes fields contained in a L3 rule.

Table 30.28 L3 rule format (1 of 2)

Field name	Field size (bit)	Field Explanation
EV	1	Entry Valid Values: ● 0: Entry is not valid ● 1: Entry is valid
SID	131	Stream ID of the entry. The stream ID is created in perfect filter, IPv4/IPv6 hash stream filter or L2 stream filter (respectively refer to section (5) Cascade filters, section (3) Stream ID creation and section section 30.5.2.4.3. L2 Stream Filter)
SL	1	Security Level Values: ● 0: Entry is unsecure ● 1: Entry is secure
MSDUV	1	MSDU Valid Values: ● 0: Frames matching this entry will be ignored by PSFP MSDU filter. ● 1: Frames matching this entry will be processed by PSFP MSDU filter.
MSDUN	4	MSDU Number Functions: ● HW: If corresponding entry L3.MSDUV value is set to 1, frames matching this entry will be processed by PSFP MSDU filter number L3.MSDUN.
MTRV	1	Meter Valid Values: ● 0: Frames matching this entry will be ignored by PSFP Meter filter. ● 1: Frames matching this entry will be processed by PSFP Meter filter.
MTRN	5	Meter Number Functions: ● HW: If corresponding entry L3.MTRV value is set to 1, frames matching this entry will be processed by PSFP MSDU filter number L3.MTRN.
FRERV	1	FRER Valid Values: ● 0: Frames matching this entry will be ignored by FRER individual recovery. LINK ● 1: Frames matching this entry will be processed by FRER individual recovery. LINK
FRERN	7	FRER Number Functions: ● HW: If corresponding entry L3.FRERV value is set to 1, frames matching this entry will be processed by FRER entry number L3.FRERN for individual recovery.
RV	1	Routing Valid Values: ● 0: Frames matching this entry will be ignored by routing. ● 1: Frames matching this entry will be processed by routing.
RN	8	Routing Number Functions: ● HW: If corresponding entry L3.RV value is set to 1, frames matching this entry will be processed by routing rule number L3.RN.

Table 30.28 L3 rule format (2 of 2)

Field name	Field size (bit)	Field Explanation
SLV	4	Source Lock Vector Values: ● Bit i set to 0: Frames matching this entry coming from port i are rejected by source port lock error. ● Bit i set to 1: Frames matching this entry coming from port i are forwarded/routed.
DV	4	Destination vector Functions: ● HW: Ports to where the frame should be sent.
CSD	6	CPU Sub-Destinations Functions: ● HW: CPU Sub-Destinations the frame should be sent.
CME	1	CPU Mirroring Enable Values: ● 0: CPU mirroring is disabled. ● 1: CPU mirroring is enabled.
EME	1	Ethernet Mirroring Enable Values: ● 0: Ethernet mirroring is disabled. ● 1: Ethernet mirroring is enabled.
IPU	1	Internal Priority Update Values: ● 0: Frames matching this entry will not have their internal priority updated. ● 1: Frames matching this entry will have their internal priority updated.
IPV	3	Internal Priority Value Functions: ● HW: If corresponding entry L3.IPU value is set to 1, frames matching this entry will have their priority updated to L3.IPV.

(2) SW Learning

Learning is used to add/remove an entry in the L3 table. Table 30.29 describes register used to learn a stream ID in the Layer 3 table. Table 30.30 describes the learn results.

Table 30.29 L3 learn registers (1 of 2)

Register name	Field name/corresponding field in l3 table	Field explanation
FWLTHTL0.LTHED	Entry delete	Not present in L3 table, used to select between learning an entry and deleting an entry
FWLTHTL0.LTHSLL	L3.SL	See (1) Rule format.
FWLTHTL0.LTHSLP0/ FWLTHTL1.LTHSLP1/ FWLTHTL2.LTHSLP2/ FWLTHTL3.LTHSLP3/ FWLTHTL4.LTHSLP4	L3.SID	See (1) Rule format.
FWLTHTL5.LTHMSDUVL	L3.MSDUV	See (1) Rule format.
FWLTHTL5.LTHMSDUNL	L3.MSDUN	See (1) Rule format.
FWLTHTL6.LTHMTRVL	L3.MTRV	See (1) Rule format.
FWLTHTL6.LTHMTRNL	L3.MTRN	See (1) Rule format.
FWLTHTL6.LTHFRERVL	L3.FRERV	See (1) Rule format.
FWLTHTL6.LTHFRERNL	L3.FRERN	See (1) Rule format.
FWLTHTL7.LTHRVL	L3.RV	See (1) Rule format.
FWLTHTL7.LTHRNL	L3.RN	See (1) Rule format.
FWLTHTL7.LTHSLVL	L3.SLV	See (1) Rule format.
FWLTHTL80.LTHCSDL	L3.CSD[5:0]	See (1) Rule format.
FWLTHTL9.LTHDVL	L3.DV	See (1) Rule format.

Table 30.29 L3 learn registers (2 of 2)

Register name	Field name/corresponding field in l3 table	Field explanation
FWLTHTL9.LTHCMEL	L3.CME	See (1) Rule format.
FWLTHTL9.LTHEMEL	L3.EME	See (1) Rule format.
FWLTHTL9.LTHIPUL	L3.IPU	See (1) Rule format.
FWLTHTL9.LTHIPVL	L3.IPV	See (1) Rule format.

Table 30.30 L3 learn result

Register name	Field name/ corresponding field in l3 table	Field explanation
FWLTHTLR.LTHLF	Learning fail	Learning fail with happen in one of the following conditions: ● The learn entry is not yet in the table and the table is already full (FWLTHTEM.LTHTEN == 256). ● The Layer 3 table is not ready (FWLTHTIM.LTHTR is not set) ● The learning function is used to delete an entry (FWLTHTL0.LTHED set for learning) and the entry is not found in the Layer 3 table
FWLTHTLR.LTHLSF	Learning security fail	Security fail with happen in one of the following conditions: ● The learn entry is an unsecure entry (FWLTHTL0.LTHSLL not set for learning) and the unsecure entry number FWLTHTEM.LTHTUEN has reached the maximum unsecure entry number FWLTHHEC.LTHHMUE. ● The learn action is done by the APB which tries to learn an unsecure entry when the entry is already in the table, is secure (L3.SL is set) and when FWLTHTEM.LTHTUEN has reached the maximum unsecure entry number FWLTHHEC.LTHHMUE.
FWLTHTLR.LTHLO	Learning overwrite	Overwrite with happen in one of the following conditions: ● The learn stream ID is already present in the table, and no error happened during learning.
FWLTHTLR.LTHLCN	Learning collision number	Give the number of collisions that needs to be resolved to access the learnt entry if the lean succeeded. If FWLTHTLR.LTHLCN value comes out bigger than FWLTHHEC.LTHHMC register value, the newly learnt entry will be ignored by forwarding engine during forwarding. A switch reconfiguration with a different L3 Hash equation should be considered.

Note: HW: The fuller the table is, the slower Learning gets.

(3) Hash maximum collision number

Hash maximum number avoids the ESWM to have back pressure due to a slow forwarding mechanism.

Equation (1) shows how to calculate this number. This number should be set in FWLTHHEC.LTHHMC register.

(1) FWLTHHEC.LTHHMC = (*clk_freq*[Hz] * *Average_frame_size*[bit] / *Incoming_throughput*[bps] - 4) / 3

Where:

- *clk_freq* is the clock frequency of ESWCLK.
- *Average_frame_size* is the average frame size expected on switch inputs
- *Incoming_throughput* is the total throughput expected on switch inputs.

For example, if *clk_freq* is 150 MHz, *Average_frame_size* is 128 bytes and *Incoming_throughput* is 3 Gbps (1 Gbps of data on 2 GMII and 1 Gbps of data on 1 GWCA), the value 15 should be set to FWLTHHEC.LTHHMC register.

(4) Hash equation

The hash equation is controlled by register FWLTHHC.LTHHE.

The initial value of the equation is $1 + x^8$.

If there are too many collisions in the table (FWLTHTLR.LTHLCN > FWLTHHEC.LTHHMC during learning), this hash equation can be changed to minimize the number of collisions by:

- SW: Modelling the hash equations and find an equation which have a reasonable number of collisions.

- SW: Trying randomly some equations until finding an equation which have a reasonable number of collisions.

(5) SW Searching

Searching is used to find the information related to a stream ID. Table 30.31 describes register used to search a stream ID in the Layer 3 table. Table 30.32 describes the search results.

Table 30.31 L3 search registers

Register name	Field name/corresponding field in l3 table	Field explanation
FWLTHTS0.LTHSSP0/ FWLTHTS1.LTHSSP1/ FWLTHTS2.LTHSSP2/ FWLTHTS3.LTHSSP3/ FWLTHTS4.LTHSSP4	L3.SID	See (1) Rule format.

Table 30.32 L3 search result

Register name	Field name/corresponding field in l3 table	Field explanation
FWLTHTSR0.LTHSNF	Search not found.	Search failed because the searched stream ID is not in the table
FWLTHTSR0.LTHSCN	Search collision number	Give the number of collisions occurred to find the entry.
FWLTHTSR0.LTHSLS	L3.SL	See (1) Rule format.
FWLTHTSR1.LTHMSDUVS	L3.MSDUV	See (1) Rule format.
FWLTHTSR1.LTHMSDUNS	L3.MSDUN	See (1) Rule format.
FWLTHTSR2.LTHMTRVS	L3.MTRV	See (1) Rule format.
FWLTHTSR2.LTHMTRNS	L3.MTRN	See (1) Rule format.
FWLTHTSR3.LTHFRERVS	L3.FRERV	See (1) Rule format.
FWLTHTSR3.LTHFRERNS	L3.FRERN	See (1) Rule format.
FWLTHTSR3.LTHSLVS	L3.SLV	See (1) Rule format.
FWLTHTSR3.LTHRVS	L3.RV	See (1) Rule format.
FWLTHTSR3.LTHRNS	L3.RN	See (1) Rule format.
FWLTHTSR40.LTHCSDS	L3.CSD[5:0]	See (1) Rule format.
FWLTHTSR5.LTHDVS	L3.DV	See (1) Rule format.
FWLTHTSR5.LTHCMES	L3.CME	See (1) Rule format.
FWLTHTSR5.LTHEMES	L3.EME	See (1) Rule format.
FWLTHTSR5.LTHIPUS	L3.IPU	See (1) Rule format.
FWLTHTSR5.LTHIPVS	L3.IPV	See (1) Rule format.

(6) SW Reading

Reading is used to dump the Layer 3 table content. This function cannot be used to read a specific entry because it is not possible for the software to know at which address an entry can be found in the Layer 3 RAM. Table 30.33 describes register used to read an entry in the Layer 3 table. Table 30.34 describes the read results.

Table 30.33 L3 read registers

Register name	Field name/corresponding field in l3 table	Field explanation
FWLTHTR.LTHAR	Read address	Address that will be used to read an entry in the Layer 3 table

Table 30.34 L3 read result

Register name	Field name/corresponding field in l3 table	Field explanation
FWLTHTRR0.LTHEVR	L3.EV	The read address holds a valid entry Only if this bit is set when reading an address, the corresponding entry and the following registers in this table are valid. See (1) Rule format
FWLTHTRR1.LTHSRP0/ FWLTHTRR2.LTHSRP1/ FWLTHTRR3.LTHSRP2/ FWLTHTRR4.LTHSRP3/ FWLTHTRR5.LTHSRP4	L3.SID	See (1) Rule format.
FWLTHTRR1.LTHSLR	L3.SL	See (1) Rule format.
FWLTHTRR6.LTHMSDUVR	L3.MSDUV	See (1) Rule format.
FWLTHTRR6.LTHMSDUNR	L3.MSDUN	See (1) Rule format.
FWLTHTRR7.LTHMTRVR	L3.MTRV	See (1) Rule format.
FWLTHTRR7.LTHMTRNR	L3.MTRN	See (1) Rule format.
FWLTHTRR7.LTHFRERVR	L3.FRERV	See (1) Rule format.
FWLTHTRR7.LTHFRERNR	L3.FRERN	See (1) Rule format.
FWLTHTRR8.LTHSLVR	L3.SLV	See (1) Rule format.
FWLTHTRR8.LTHRVR	L3.RV	See (1) Rule format.
FWLTHTRR8.LTHRNR	L3.RN	See (1) Rule format.
FWLTHTRR90.LTHCSDR	L3.CSD[5:0]	See (1) Rule format.
FWLTHTRR10.LTHDVR	L3.DV	See (1) Rule format.
FWLTHTRR10.LTHCMER	L3.CME	See (1) Rule format.
FWLTHTRR10.LTHEMER	L3.EME	See (1) Rule format.
FWLTHTRR10.LTHIPUR	L3.IPU	See (1) Rule format.
FWLTHTRR10.LTHIPVR	L3.IPV	See (1) Rule format.

30.5.2.4.5 Layer 3 Forwarding/Routing/Filtering

Layer 3 forwarding/routing/filtering is used to forward local ethernet descriptors (LDESCR.FMT [GWCA] set to 1’b0) received from agents using FWPCi0/FWPCi1 and FWCEPRC2 register and can be monitored using FWEIS0i interrupt register. Layer 3 forwarding/routing/filtering searches received valid stream IDs (section (5) Cascade filters, section (a) IPv4 Stream ID creation and section (b) IPv6 Stream ID creation) in the layer 3 table to extract corresponding entry fields (section (1) Rule format). Depending on the entry fields, Layer 3 forwarding/routing/filtering will forward/route/filter frames following Layer 3 forwarding/routing secure flow in Figure 30.85, Layer 3 forwarding/routing unsecure flow in Figure 30.86 and Layer 3 filtering flow in Figure 30.87.

Functions:

- FWPCi0.LTHTA register is used to activate layer 3 table for frames received from port i.
- FWPCi0.LTHRUS register is used to reject unknown streams received from port i.
- FWPCi0.LTHRUSS register is used to reject secure unknown streams received from port i.
- FWPCi1.LTHFM register is used to set to which ports frames received from port i can be forwarded.
- FWCEPRC2 register enables exceptional path for error descriptors.
- FWEIS0i interrupt register notifies that an ethernet descriptor has been rejected because of an error.

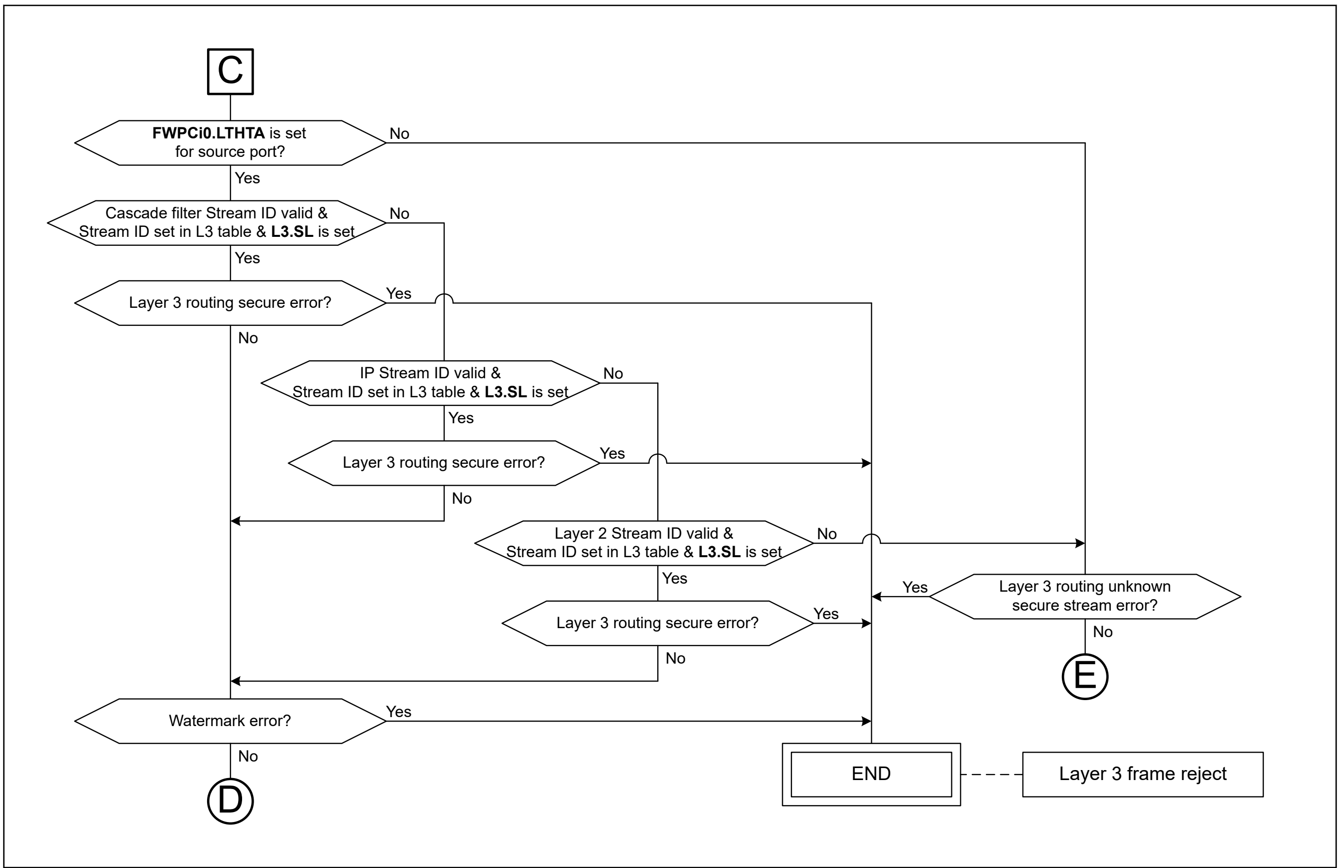


Figure 30.85 Layer 3 forwarding/routing secure flow

Note:
- Layer 3 routing secure errors and Layer 3 routing unknown secure stream errors are described in section (1) Layer 3 routing errors.
- Watermark errors are described in section (1) Watermark errors.
- Link C is linked from Figure 2.
- Link D links to Figure 30.87.
- Link E links to Figure 30.95

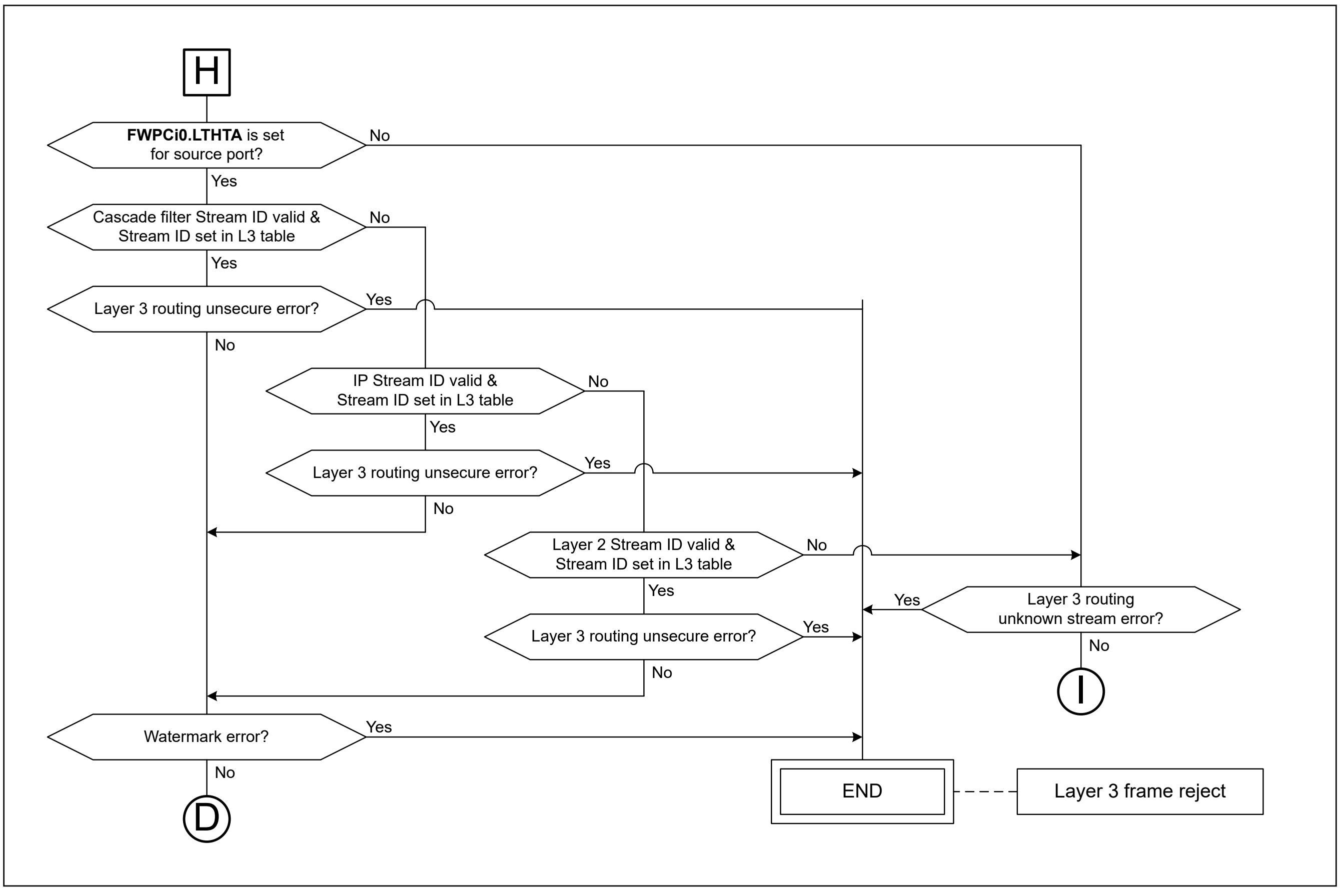


Figure 30.86 Layer 3 forwarding/routing unsecure flow

Note:
- Layer 3 routing errors and Layer 3 routing unknown stream errors are described in section (1) Layer 3 routing errors.
- Watermark errors are described in section (1) Watermark errors.
- Link H is linked from Figure 30.99.
- Link D links to Figure 30.87.
- Link I links to Figure 30.96.

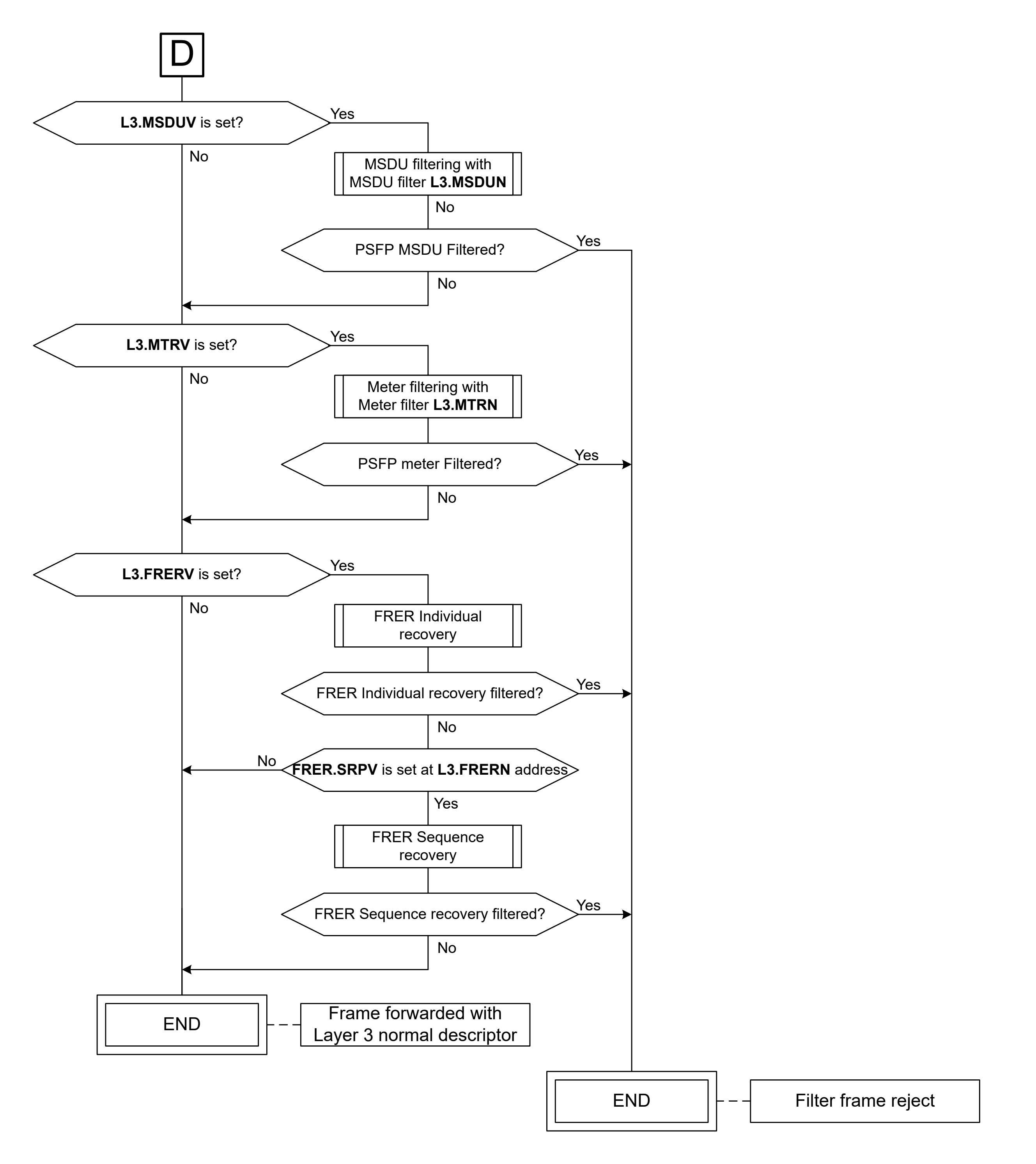


Figure 30.87 Layer 3 filtering flow

Note:
- Link D is linked from Figure 30.85 and Figure 30.86.
- Filter frame reject is described in section (2) Filtering errors.
- MSDU filtering is described in Figure 30.103.
- Meter filtering is described in Figure 30.104.
- Individual recovery is described in section (e) Individual recovery algorithms.
- Sequence recovery is described in section (f) Sequence recovery algorithms.

(1) Layer 3 routing errors

Table 30.35 describes the Layer 3 routing errors. Instead of being discarded by forwarding engine, an error frame can be forwarded to exceptional path using Layer 3 exceptional descriptor described in section (4) Layer 3 exceptional descriptor by setting corresponding error exceptional bit. For error set conditions, refer to corresponding error interrupt register explanations.

Table 30.35 Layer 3 routing errors

Error	Error name	Error interrupt	Error exceptional bit
Layer 3 routing secure errors			
SNTF	Secure No Target Filtering	FWEIS0i.LTHNTFS	FWCEPRC2.FLTHNTFEF
SSPF	Secure Source Port Filtering	FWEIS0i.LTHSPFS	FWCEPRC2.FLTHSLFEF
Layer 3 routing unknown secure stream errors			
SUF	Secure Unknown filtering	FWEIS0i.LTHUFS	FWCEPRC2.FLTHUFEF
Layer 3 routing errors			
NTF	No Target Filtering	FWEIS0i.LTHNTFS	FWCEPRC2.FLTHNTFEF
SPF	Source Port Filtering	FWEIS0i.LTHSPFS	FWCEPRC2.FLTHSLFEF
Layer 3 routing unknown stream errors			
UF	Unknown filtering	FWEIS0i.LTHUFS	FWCEPRC2.FLTHUFEF

(2) Filtering errors

Table 30.36 describes the Filter errors. Instead of being discarded by forwarding engine, an error frame can be forwarded to exceptional path using Filter exceptional descriptor described in section (5) Filter exceptional descriptor by setting corresponding error exceptional bit. For error set conditions, refer to corresponding error interrupt register explanations.

Table 30.36 Filtering errors

Error	Error name	Error interrupt	Error exceptional bit
MSDUF	MSDU filtered	FWEIS2.PMFS	FWCEPRC1.FMSDUFEF
MTRF	Meter filtered	FWEIS5.PMRFS	FWCEPRC1.FMTRFEF
IFRERF	Individual FRER Filtered	FWEIS60/61/62/63.FFS for no sequence error of stuck error FWEIS70/71/72/73.FOORS for out of range error	FWCEPRC1.FIFFEF
SFRERF	Sequence FRER Filtered	FWEIS60/61/62/63.FFS for no sequence error FWEIS70/71/72/73.FOORS for out of range error No interrupt if a frame is discarded because it was a duplicate	FWCEPRC1.FSFFEF

(3) Layer 3 normal descriptor

Layer 3 normal descriptor fields are described in Table 30.37.

Table 30.37 Layer 3 normal descriptor fields (1 of 2)

Field name FDESCR.	Values
DV	L3.DV & !FWPCi1.LTHFM
SEC	L3.SL
CSD	L3.CSD
IPV	If L3.IPU is set, set to L3.IPV. If L3.IPU is not set, set to the frame input priority. (See section 30.5.1.4. ESWM Frame Input Priority.)
MINFO	See Figure 30.88 and Table 30.38.

Table 30.37 Layer 3 normal descriptor fields (2 of 2)

Field name FDESCR.	Values
RV	L3.RV
RN	L3.RN
SEQN	Set to frame input sequence number if L3.RN is not mapped to a sequence number generation function (refer to section section 30.5.3.2.2. Sequence Number Generation)

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	FCLR[2:0]		FFC[2:0]		FWDC[3:0]			
1	FID[15:0]							
2								

Figure 30.88 Layer 3 normal descriptor MINFO format

Table 30.38 Layer 3 normal descriptor MINFO field explanation

Field name FDESCR.	Bit width	Field explanation	Values
FWDC	4	ForWarDing Code Present in all descriptors, allowing the type of forwarding to be identified	Fixed to 5 for Layer 3 normal descriptors
FFC	2	Frame Format Code	Set to 0 for perfect filter forwarding Set to 1 for IPv4 forwarding Set to 2 for IPv6 forwarding Set to 3 for Layer 2 stream forwarding
FCLR	2	Frame CoLoR	Set to 0 when frame has no color Set to 1 when frame is green Set to 2 when frame is yellow Set to 3 when frame is red Refer to section section 30.5.3.1.2. Meter Filter/ATS for color attribution
FID	16	Frame IDentifier	For Cascade filter stream ID, set to the cascade filter number that matched. For IPV stream ID, set to the hash ID inserted in the stream ID. For L2 stream, set to All0.

(4) Layer 3 exceptional descriptor

Layer 3 exceptional descriptor fields are described in Table 30.39.

Table 30.39 Layer 3 exceptional descriptor fields

Field name FDESCR.	Values
DV	Exceptional descriptors are forwarded to GWCA number FWCEPTC.EPCS (port number 2)
SEC	FWCEPTC.EPSL
CSDj	FWCEPTC.EPCSD
IPV	FWCEPTC.EPIPV
MINFO	See Figure 30.89 and Table 30.40.
RV	0
RN	ALL0
SEQN	Set to frame input sequence number

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	FCLR[1:0]		FFC[1:0]		FWDC[3:0]			
1	—					MTRF	GATEF	MSDUF
2	—							

Figure 30.89 Layer 3 exceptional descriptor MINFO format

Table 30.40 Layer 3 exceptional descriptor MINFO field explanation

Field name FDESCR.	Bit width	Field explanation	Values
FWDC	4	ForWarDing Code Present in all descriptors, allowing the type of forwarding to be identified	Fixed to 6 for Layer 3 exceptional descriptors
FFC	2	Frame Format Code	Set to 0 for perfect filter forwarding Set to 1 for IPv4 forwarding Set to 2 for IPv6 forwarding Set to 3 for Layer 2 stream forwarding Restrictions: • HW: This field is not valid when Secure Unknown filtering or Unknown filtering happened.
SNTF	1	Secure No Target Filtering	See (1) Layer 3 routing errors.
SSPF	1	Secure Source Port Filtering	See (1) Layer 3 routing errors.
SUF	1	Secure Unknown filtering	See (1) Layer 3 routing errors.
NTF	1	No Target Filtering	See (1) Layer 3 routing errors.
SPF	1	Source Port Filtering	See (1) Layer 3 routing errors.
UF	1	Unknown filtering	See (1) Layer 3 routing errors.

(5) Filter exceptional descriptor

Filter exceptional descriptor fields are described in Table 30.41.

Table 30.41 Filter exceptional descriptor fields

Field name FDESCR.	Values
DV	Exceptional descriptors are forwarded to GWCA number FWCEPTC.EPCS (port number 2)
SEC	FWCEPTC.EPSL
CSDj	FWCEPTC.EPCSD
IPV	FWCEPTC.EPIPV
MINFO	Refer to Figure 30.90 and Table 30.42
RV	0
RN	ALL0
SEQN	Set to frame input sequence number

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	FCLR[1:0]		FFC[1:0]		FWDC[3:0]			
1	—					MTRF	GATEF	MSDUF
2	—							

Figure 30.90 Filter exceptional descriptor MINFO format

Table 30.42 Filter exceptional descriptor MINFO field explanation

Field name FDESCR.	Bit width	Field explanation	Values
FWDC	4	ForWarDing Code Present in all descriptors, allowing the type of forwarding to be identified	Fixed to 15 for Layer 3 exceptional descriptors
FFC	2	Frame Format Code	Set to 0 for perfect filter forwarding Set to 1 for IPv4 forwarding Set to 2 for IPv6 forwarding Set to 3 for Layer 2 stream forwarding
FCLR	2	Frame CoLoR	Set to 0 when frame has no color Set to 1 when frame is green Set to 2 when frame is yellow Set to 3 when frame is red Refer to section section 30.5.3.1.2. Meter Filter/ATS for color attribution
MSDUF	1	MSDU Filtered	See (2) Filtering errors.
MTRF	1	Meter Filtered	See (2) Filtering errors.
IFRERF	1	Individual FRER Filtered	See (2) Filtering errors.
IFNOSEQ	1	Individual FRER NO SEQuence	See (2) Filtering errors.
IFOOR	1	Individual FRER Out Of Range (For match algorithm, this flag is set for Out of order error)	See (2) Filtering errors.
FSTUCK	1	FRER Stuck	See (2) Filtering errors.
SFRERF	1	Sequence FRER Filtered	See (2) Filtering errors.
SFNOSEQ	1	Sequence FRER NO SEQuence	See (2) Filtering errors.
SFOOR	1	Sequence FRER Out Of Range (For match algorithm, this flag is set for Out of order error)	See (2) Filtering errors.

30.5.2.5 Layer 2 Forwarding

Figure 30.91 describes the block diagram of Layer 2 forwarding.

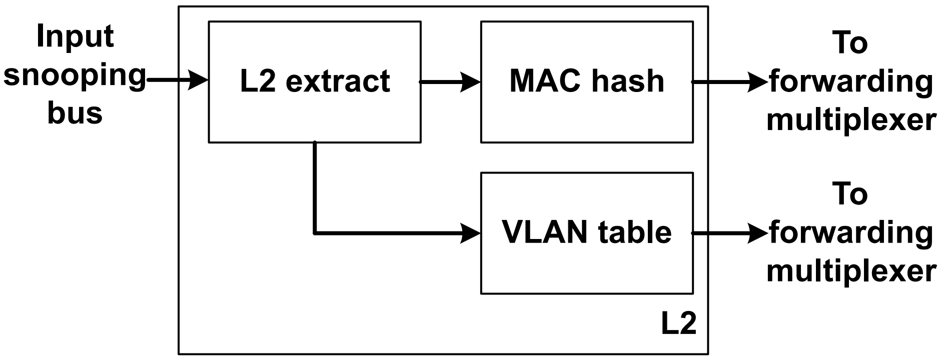


Figure 30.91 Block diagram of Layer 2 forwarding

The Layer 2 forwarding function is separated in three functional blocks:

- L2 extract: This block snoops all the required information from the forwarding engine snooping bus and extracts the MAC destination, MAC source addresses and the VLAN ID from frames.
- MAC hash: This block handles the MAC addresses meta information such as forwarding and routing information using a hash function.

- VLAN table: This block handles the VLAN ID meta information such as forwarding and routing information.

Note: Forwarding multiplexer contains the forwarding flow. The forwarding flow is explained by forwarding type. For Layer 3 forwarding/routing/filtering, related forwarding flows are described in Figure 30.95 and Figure 30.96.

30.5.2.5.1 L2 Extract

The L2 extract is used to extract the source and destination MAC addresses and the VLAN ID all frames.

Figure 30.92 describes MAC addresses and VLAN ID extraction. MAC addresses extraction always occurs. VLAN extraction only occurs when FWGC.SVM is different from 00b. If VLAN ID extraction does not occur, all VLANs are considered as unknown.

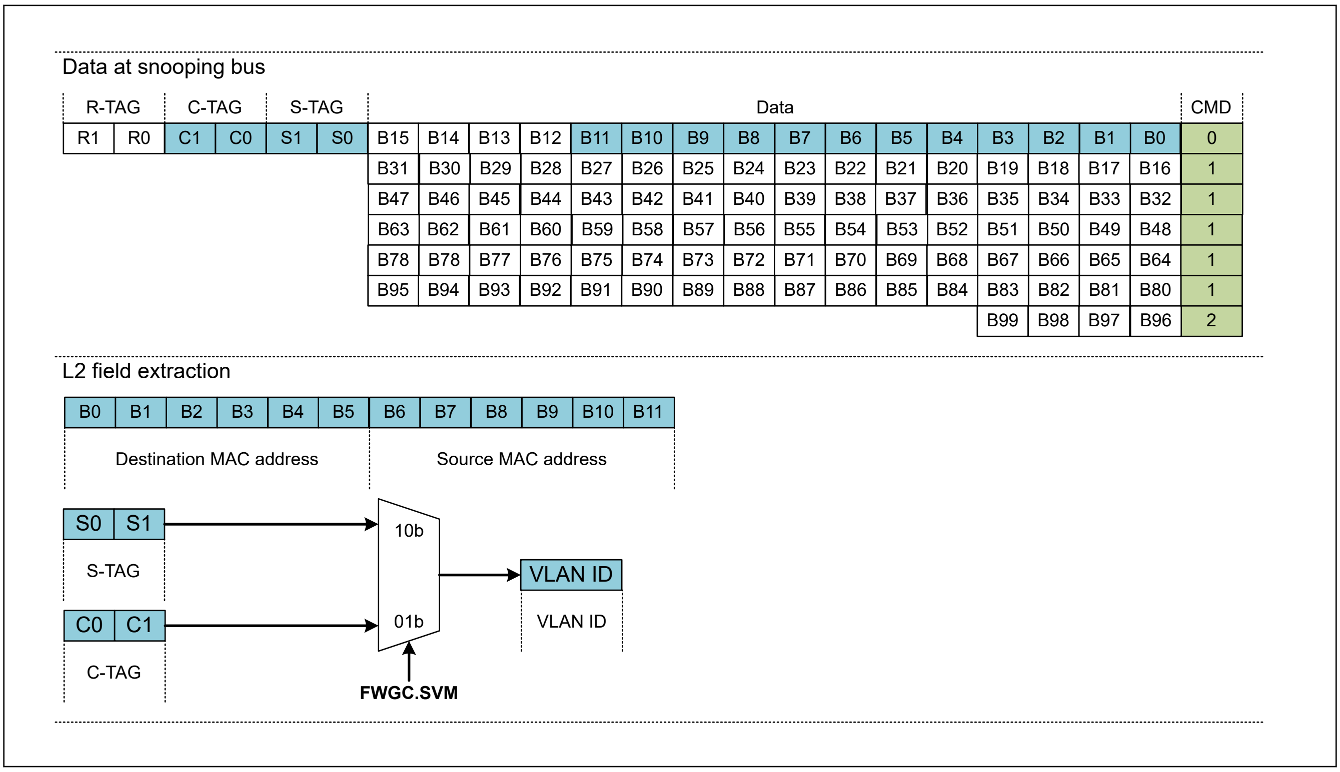


Figure 30.92 MAC address and VLAN ID extraction

30.5.2.5.2 MAC Hash

MAC hash aims at searching/learning/reading MAC rules thanks to a programmable hash. The hash algorithm is able to solve all the collisions in hardware, so it does not need any software intervention to operate.

MAC hash is divided in the following items:

- Rule format: Determines what information will be store along with the MAC address in the table.
- SW Learning: Function to add/suppress an entry from the MAC table by software.
- Hash maximum collision number: Avoids hash search to slow down the forwarding pipeline and apply back pressure on the ingress ports.
- Hash equation: Function to set hash calculation to minimize collisions.
- SW Searching: Function to find the forwarding information related to a MAC address by software.
- SW Reading: Function to read the entries in the MAC table by software.
- HW Searching: Function to find the forwarding information related to a MAC address by forwarding mechanism.
- HW Learning: Function to learn unicast MAC source addresses by forwarding mechanism.
- HW aging: Function used to suppress unused dynamic MAC addresses.

(1) Rule format

The MAC table is used to store MAC rules. A MAC rule contains the information for the frame forwarding. All the fields in this table, if quoted, will be written MAC.{Field name}. Table 30.43 describes fields contained in a MAC rule.

Table 30.43 MAC rule format (1 of 2)

Field name	Field size (bit)	Field Explanation
EV	1	Entry Valid Values: ● 0: Entry is not valid ● 1: Entry is valid
SL	1	Security Level Values: ● 0: Entry is unsecure ● 1: Entry is secure
MA	48	MAC Address of the entry. The MAC address is extracted by L2 extract. (See section 30.5.2.5.1. L2 Extract) This address is in big endian.
HLD	1	Hardware Learn Disable Values: ● 0: Source MAC hardware learning is enabled for frames containing MAC.MAC as destination MAC address. ● 1: Source MAC hardware learning is disabled for frames containing MAC.MAC as destination MAC address.
DSLV	4	Destination Source Lock Vector Values: ● Bit i set to 0: Frames with a destination MAC matching this entry coming from port i are rejected by source port lock error. ● Bit i set to 1: Frames with a destination MAC matching this entry coming from port i are forwarded/routed.
SSLV	4	Source Source Lock Vector Values: ● Bit i set to 0: Frames with a source MAC matching this entry coming from port i are rejected by source port lock error. ● Bit i set to 1: Frames with a source MAC matching this entry coming from port i are forwarded/routed.
DV	4	Destination vector Functions: ● HW: Ports to where the frame should be sent.
CSD	6	CPU Sub-Destinations Functions: ● HW: CPU Sub-Destinations the frame should be sent.
CME	1	CPU Mirroring Enable Values: ● 0: CPU mirroring is disabled. ● 1: CPU mirroring is enabled.
EME	1	Ethernet Mirroring Enable Values: ● 0: Ethernet mirroring is disabled. ● 1: Ethernet mirroring is enabled.
IPU	1	Internal Priority Update Values: ● 0: Frames matching this entry will not have their internal priority updated. ● 1: Frames matching this entry will have their internal priority updated.
IPV	3	Internal Priority Value Functions: ● HW: If corresponding entry MAC.IPU value is set to 1, frames matching this entry will have their priority updated to MAC.IPV.
DE	1	Dynamic Entry Values: ● 0: Entry is static. Migration and aging cannot be applied to the entry. ● 1: Entry is dynamic. Migration and aging can be applied to the entry.

Table 30.43 MAC rule format (2 of 2)

Field name	Field size (bit)	Field Explanation
AB	1	Aging Bit: Values: ● 0: Entry has been searched as a MAC source address by forwarding mechanism since previous aging. The entry should not be deleted during next aging. ● 1: Entry has not been searched as a MAC source address by forwarding mechanism since previous aging. The entry should be deleted during next aging if it is not searched as a MAC source address by forwarding mechanism before.

(2) SW Learning

Learning is used to add/remove an entry in the MAC table.Table 30.44 describes register used to learn an entry in the MAC table. Table 30.45 describes the learn results.

Table 30.44 MAC learn registers

Register name	Field name/corresponding field in MAC table	Field explanation
FWMACTL0.MACED	Entry delete	Not present in L3 table, used to select between learning an entry and deleting an entry
FWMACTL0.MACSLL	MAC.SL	See (1) Rule format.
{FWMACTL1.MACMALP0, FWMACTL2.MACMALP1}	MAC.MA	See (1) Rule format.
FWMACTL0.MACDEL	MAC.DE	See (1) Rule format.
FWMACTL0.MACHLDL	MAC.HLD	See (1) Rule format.
FWMACTL3.MACDSLVL	MAC.DSLV	See (1) Rule format.
FWMACTL3.MACSSLVL	MAC.SSLV	See (1) Rule format.
FWMACTL40.MACCSD	MAC.CSD[5:0]	See (1) Rule format.
FWMACTL5.MACDVL	MAC.DV	See (1) Rule format.
FWMACTL5.MACCMEL	MAC.CME	See (1) Rule format.
FWMACTL5.MACEMEL	MAC.EME	See (1) Rule format.
FWMACTL5.MACIPUL	MAC.IPU	See (1) Rule format.
FWMACTL5.MACIPVL	MAC.IPV	See (1) Rule format.

Table 30.45 MAC learn result (1 of 2)

Register name	Field name/ corresponding field in MAC table	Field explanation
FWMACTLR.MACLF	Learning fail	Learning fail with happen in one of the following conditions: ● The learn entry is not yet in the table and the table is already full (FWMACTEM.MACTEN == 2048). ● The MAC table is not ready (FWMACTIM.MACTR is not set) ● The learning function is used to delete an entry (FWMACTL0.MACED set for learning) and the entry is not found in the MAC table
FWMACTLR.MACLSF	Learning security fail	Security fail with happen in one of the following conditions: ● The learn entry is an unsecure entry (FWMACTL0.MACSLL not set for learning) and the unsecure entry number FMACTEM.MACTUEN has reached the maximum unsecure entry number FWMACHEC.MACHMUE. ● The learn action is done by the APB which tries to learn an unsecure entry when the entry is already in the table, is secure (MAC.SL is set) and when FWMACTEM.MACTUEN has reached the maximum unsecure entry number FWMACHEC.MACHMUE.
FWMACTLR.MACLO	Learning overwrite	Overwrite with happen in one of the following conditions: ● The learn MAC is already present in the table, and no error happened during learning.

Table 30.45 MAC learn result (2 of 2)

Register name	Field name/ corresponding field in MAC table	Field explanation
FWMACTLR.MACLCN	Learning collision number	Give the number of collisions that needs to be resolved to access the learnt entry if the lean succeeded. If FWMACTLR.MACLCN value comes out bigger than FWMACHEC.MACHMC register value, the newly learnt entry will be ignored by forwarding engine during forwarding. A switch reconfiguration with a different MAC Hash equation should be considered.

Note: HW: The fuller the table is, the slower Learning gets.

(3) Hash maximum collision number

Hash maximum number avoids the ESWM to have back pressure due to a slow forwarding mechanism.

Equation (2) shows how to calculate this number. This number should be set in FWMACHEC.MACHMC register.

(2) FWMACHEC.MACHMC = (*clk_freq*[Hz] * *Average_frame_size*[bit] / *Incoming_throughput*[bps] - 4) / 3

Where:

- *clk_freq* is the clock frequency of ESWCLK.
- *Average_frame_size* is the average frame size expected on switch inputs
- *Incoming_throughput* is the total throughput expected on switch inputs.

For example, if *clk_freq* is 150 MHz, *Average_frame_size* is 128 bytes and *Incoming_throughput* is 3 Gbps (1 Gbps of data on 2 GMII and 1 Gbps of data on 1 GWCA), the value 15 should be set to FWLTHHEC.LTHHMC register.

(4) Hash equation

The hash equation is controlled by register FWMACHC.MACHE.

The initial value of the equation is $1 + x^{11}$.

If there are too many collisions in the table (FWMACTLR.MACLCN > FWMACHEC.MACHMC during learning), this hash equation can be changed to minimize the number of collisions by:

- SW: Modelling the hash equations and find an equation which have a reasonable number of collisions.
- SW: Trying randomly some equations until finding an equation which have a reasonable number of collisions.

(5) SW Searching

Searching is used to find the information related to a MAC address. Table 30.46 describes register used to search an MAC address in the MAC table. Table 30.47 describes the search results.

Table 30.46 MAC search registers

Register name	Field name/corresponding field in MAC table	Field explanation
{FWMACTS0.MACMASP0, FWMACTS1.MACMASP1}	MAC.MA	See (1) Rule format.

Table 30.47 MAC search result (1 of 2)

Register name	Field name/corresponding field in MAC table	Field explanation
FWMACTSR0.MACSNF	Search not found.	Search failed because the searched MAC address is not in the table
FWMACTSR0.MACSCN	Search collision number	Give the number of collisions occurred to find the entry.
FWMACTSR0.MACSLS	MAC.SL	See (1) Rule format.
FWMACTSR0.MACDES	MAC.DE	See (1) Rule format.

Table 30.47 MAC search result (2 of 2)

Register name	Field name/corresponding field in MAC table	Field explanation
FWMACTSR0.MACHLDS	MAC.HLD	See (1) Rule format.
FWMACTSR1.MACDSLVS	MAC.DSLV	See (1) Rule format.
FWMACTSR1.MACSSLVS	MAC.SSLV	See (1) Rule format.
FWMACTSR20.MACCSDS	MAC.CSD[5:0]	See (1) Rule format.
FWMACTSR3.MACDVS	MAC.DV	See (1) Rule format.
FWMACTSR3.MACCMES	MAC.CME	See (1) Rule format.
FWMACTSR3.MACEMES	MAC.EME	See (1) Rule format.
FWMACTSR3.MACIPUS	MAC.IPU	See (1) Rule format.
FWMACTSR3.MACIPVS	MAC.IPV	See (1) Rule format.

(6) SW Reading

Reading is used to dump the MAC table content. This function cannot be used to read a specific entry because it is not possible for the software to know at which address an entry can be found in the MAC RAM.

Table 30.48 MAC read registers

Register name	Field name/corresponding field in MAC table	Field explanation
FWMACTR.MACAR	Read address	Address that will be used to read an entry in the Layer 3 table

Table 30.49 MAC read result

Register name	Field name/corresponding field in MAC table	Field explanation
FWMACTRR0.MACEVR	MAC.EV	The read address holds a valid entry Only if this bit is set when reading an address, the corresponding entry and the following registers in this table are valid. See (1) Rule format.
{FWMACTRR2.MACMARP0, FWMACTRR3.MACMARP1}	MAC.MA	See (1) Rule format.
FWMACTRR1.MACSLR	MAC.SL	See (1) Rule format.
FWMACTRR1.MACDER	MAC.DE	See (1) Rule format.
FWMACTRR1.MACHLDR	MAC.HLD	See (1) Rule format.
FWMACTRR1.MACAB	MAC.AB	See (1) Rule format.
FWMACTRR4.MACDSLVR	MAC.DSLV	See (1) Rule format.
FWMACTRR4.MACSSLVR	MAC.SSLV	See (1) Rule format.
FWMACTRR50.MACCSDR	MAC.CSD[5:0]	See (1) Rule format.
FWMACTRR6.MACDVR	MAC.DV	See (1) Rule format.
FWMACTRR6.MACCMER	MAC.CME	See (1) Rule format.
FWMACTRR6.MACEMER	MAC.EME	See (1) Rule format.
FWMACTRR6.MACIPUR	MAC.IPU	See (1) Rule format.
FWMACTRR6.MACIPVR	MAC.IPV	See (1) Rule format.

(7) HW learning/migration

MAC HW learning/migration conditions and settings are described in section section 30.5.1.5.2. CPU Learning Path. Table 30.50 describes to what value the MAC entry fields will be set while learning in hardware a frame coming from port i.

For Source MAC multicast address, HW learning/migration will invalid (ignored).

Table 30.50 MAC hardware learning values

Field name	Field Explanation
MAC.EV	1
MAC.SL	0
MAC.MA	Incoming frame MAC source
MAC.HLD	0
MAC.DSLV	0xF
MAC.SSLV	0xF
MAC.DV	001b << i
MAC.CSD	0x00
MAC.CME	0
MAC.EME	0
MAC.IPU	0
MAC.IPV	000b
MAC.DE	1
MAC.AB	0

(8) HW Aging

MAC HW aging function is used to remove the unused MAC addresses conditions using FWMACAGUSPC and FWMACAGC registers and can be monitored using FWMACAGM0/1 registers and FWMIS0.MACADAS interrupt register. MAC HW aging function thanks to two algorithms, the aging algorithm described in Figure 30.93 and the MAC source address search algorithm described in Figure 30.94.

Functions:

- FWMACAGUSPC register sets a prescaler to create a 1 Hz for aging.
- FWMACAGC register sets/enables the aging function
- FWMIS0.MACADAS and FWMACAGM0/1 can be used to keep track in SW of the aged MAC addresses. These registers should be used only in polling mode (FWMACAGC.MACAGPM set).

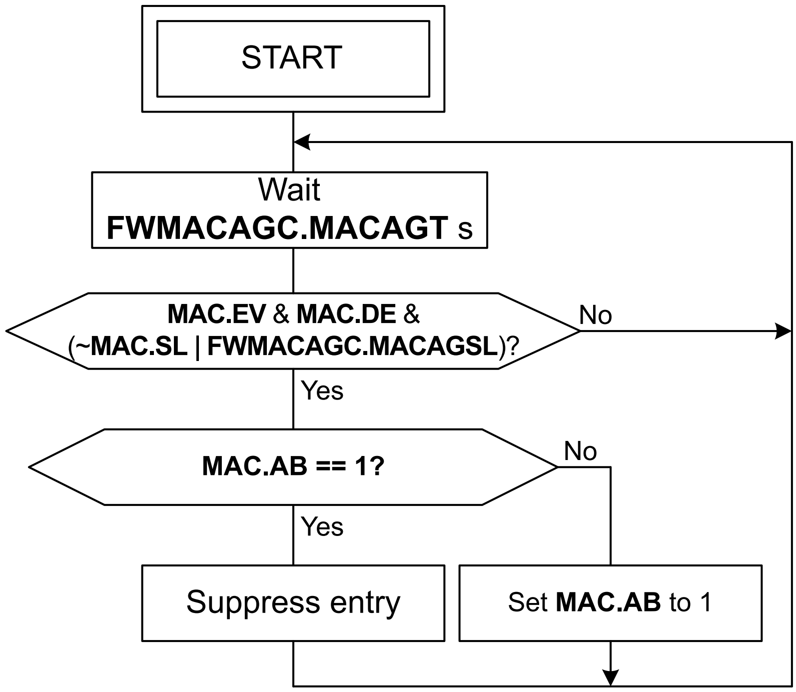


Figure 30.93 MAC aging algorithm

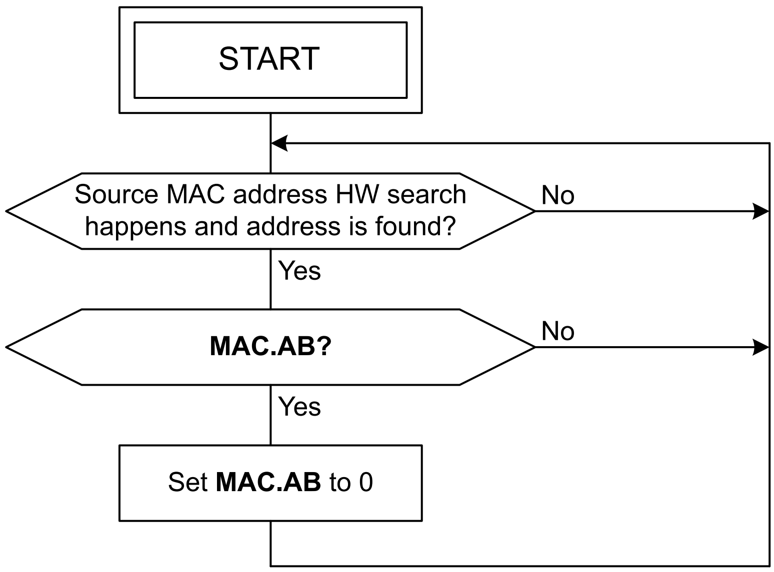


Figure 30.94 MAC source address search algorithm

30.5.2.5.3 VLAN Table

VLAN hash aims at searching/learning VLAN rules. The VLAN ID to VLAN RAM address mapping is a 1:1 mapping, so no hash is needed for VLAN handling.

VLAN table is divided in the following items:

- Rule format: Determines what information will be store in a VLAN rule. The VLAN ID is not stored a VLAN rule.
- SW Learning: Function to add/suppress an entry from the VLAN table by software.

- SW Searching: Function to find the forwarding information related information to a VLAN ID by software.
- HW Searching: Function to find the forwarding information related to a VLAN ID by forwarding mechanism.

(1) Rule format

The VLAN table is used to store VLAN rules. A VLAN rule contains the information for the frame forwarding. All the fields in this table, if quoted, will be written VLAN.{Field name}. Table 30.51 describes fields contained in a VLAN rule.

Table 30.51 VLAN rule format

Field name	Field size (bit)	Field description
EV	1	Entry Valid Values: ● 0: Entry is not valid ● 1: Entry is valid
SL	1	Security Level Values: ● 0: Entry is unsecure ● 1: Entry is secure
HLD	1	Hardware Learn Disable Values: ● 0: Source MAC hardware learning is enabled for frames containing corresponding VLAN ID. ● 1: Source MAC hardware learning is disabled for frames containing corresponding VLAN ID.
SLV	4	Source Lock Vector Values: ● Bit i set to 0: Frames with a VLAN ID matching this entry coming from port i are rejected by source port lock error. ● Bit i set to 1: Frames with a VLAN ID matching this entry coming from port i are forwarded/routed.
DV	4	Destination vector Functions: ● HW: Ports to where the frame should be sent.
CSD	6	CPU Sub-Destinations Functions: ● HW: CPU Sub-Destinations the frame should be sent
CME	1	CPU Mirroring Enable Values: ● 0: CPU mirroring is disabled. ● 1: CPU mirroring is enabled.
EME	1	Ethernet Mirroring Enable Values: ● 0: Ethernet mirroring is disabled. ● 1: Ethernet mirroring is enabled.
IPU	1	Internal Priority Update Values: ● 0: Frames matching this entry will not have their internal priority updated. ● 1: Frames matching this entry will have their internal priority updated.
IPV	3	Internal Priority Value Functions: ● HW: If corresponding entry VLAN.IPU value is set to 1, frames matching this entry will have their priority updated to VLAN.IPV.

(2) SW Learning

Learning is used to add/remove an entry in the VLAN table. Table 30.52 describes register used to learn an entry in the VLAN table. Table 30.53 describes the learn results.

Table 30.52 VLAN learn registers

Register name	Field name/corresponding field in VLAN table	Field explanation
FWVLANTL0.VLANED	Entry delete	Not present in L3 table, used to select between learning an entry and deleting an entry
FWVLANTL0.VLANSLL	VLAN.SL	See (1) Rule format.
FWVLANTL1.VLANVIDL	VLAN ID	Correspond to the address of the learnt entry.
FWVLANTL0.VLANHLDL	VLAN.HLD	See (1) Rule format.
FWVLANTL2.VLANSLVL	VLAN.SLV	See (1) Rule format.
FWVLANTL30.VLANCSDL	VLAN.CSD[5:0]	See (1) Rule format.
FWVLANTL4.VLANDVL	VLAN.DV	See (1) Rule format.
FWVLANTL4.VLANCMEL	VLAN.CME	See (1) Rule format.
FWVLANTL4.VLANEMEL	VLAN.EME	See (1) Rule format.
FWVLANTL4.VLANIPUL	VLAN.IPU	See (1) Rule format.
FWVLANTL4.VLANIPVL	VLAN.IPV	See (1) Rule format.

Table 30.53 VLAN learn result

Register name	Field name/ corresponding field in VLAN table	Field explanation
FWVLANTLR.VLANLF	Learning fail	Learning fail with happen in one of the following conditions: ● The VLAN table is not ready (FWVLANTIM.VLANTR is not set). ● The learning function is used to delete an entry (FWVLANTL0.VLANED set for learning) and the entry is not VLAN table.
FWVLANTLR.VLANLSF	Learning security fail	Security fail with happen in one of the following conditions: ● The learning entry is an unsecure entry (FWVLANTL0.VLANSLL not set for learning) and the unsecure entry number FWVLANTEM.VLANTUEN has reached the maximum unsecure entry number FWVLANTEC.VLANTMUE. ● The learning action is done by the APB which tries to learn an unsecure entry when the entry is already in the table, is secure (VLAN.SL is set) and when FWVLANTEM.VLANTUEN has reached the maximum unsecure entry number FWVLANTEC.VLANTMUE.
FWVLANTLR.VLANLO	Learning overwrite	Overwrite with happen in one of the following conditions: ● The learning VLAN is already present in the table, and no error happened during learning.

(3) SW Searching

Searching is used to find the information related to a VLAN address. Table 30.54 describes register used to search a VLAN address in the VLAN table. Table 30.55 describes the search results. This function is not actually searching the address but only reading the address corresponding to the VLAN ID in the VLAN table.

Table 30.54 VLAN search registers

Register name	Field name/corresponding field in VLAN table	Field explanation
FWVLANTS.VLANVIDS	VLAN ID	Address that should be read in VLAN table. Correspond to the VLAN ID

Table 30.55 VLAN search result (1 of 2)

Register name	Field name/corresponding field in VLAN table	Field explanation
FWVLANTSR0.VLANSNF	Search not found.	Search failed because the searched VLAN ID is not in the table
FWVLANTSR0.VLANSLS	VLAN.SL	See (1) Rule format.
FWVLANTSR0.VLANHLDS	VLAN.HLD	See (1) Rule format.

Table 30.55 VLAN search result (2 of 2)

Register name	Field name/corresponding field in VLAN table	Field explanation
FWVLANTSR1.VLANSLVS	VLAN.SLV	See (1) Rule format.
FWVLANTSR20.VLANCSDS	VLAN.CSD[5:0]	See (1) Rule format.
FWVLANTSR3.VLANDVS	VLAN.DV	See (1) Rule format.
FWVLANTSR3.VLANCMES	VLAN.CME	See (1) Rule format.
FWVLANTSR3.VLANEMES	VLAN.EME	See (1) Rule format.
FWVLANTSR3.VLANIPUS	VLAN.IPU	See (1) Rule format.
FWVLANTSR3.VLANIPVS	VLAN.IPV	See (1) Rule format.

30.5.2.5.4 Layer 2 Forwarding

Layer 2 forwarding is used to forward local ethernet descriptors (LDESCR.FMT [GWCA] set to 0) received from agents using FWPCi0 and FWCEPRC2 register and can be monitored using FWEIS0i interrupt register. Layer 2 forwarding searches received MAC addresses (section 30.5.2.5.1. L2 Extract) in the MAC table to extract corresponding entry fields ((1) Rule format) and searches valid VLAN IDs (section 30.5.2.5.1. L2 Extract) in the VLAN table to extract corresponding entry fields ((1) Rule format). Depending on the entry fields, Layer 2 forwarding will forward frames following L2 forwarding secure flow in Figure 30.95 and L2 forwarding unsecure flow in Figure 30.96.

Functions:

- FWPCi0.MACDSA register is used to activate MAC table for frames received from port i for destination MAC.
- FWPCi0.MACRUDA register is used to reject unknown destination MAC received from port i.
- FWPCi0.MACRUDSA register is used to reject secure unknown destination MAC received from port i.
- FWPCi0.MACSSA register is used to activate MAC table for frames received from port i for source MAC.
- FWPCi0.MACRUSA register is used to reject unknown source MAC received from port i.
- FWPCi0.MACRUSSA register is used to reject secure unknown source MAC received from port i.
- FWPCi0.VLANSA register is used to activate VLAN table for frames received from port i for source MAC.
- FWPCi0.VLANRU register is used to reject unknown VLAN IDs received from port i.
- FWPCi0.VLANRUS register is used to reject secure unknown VLAN IDs received from port i.
- FWPCi2.LTWFM register is used to set to which ports frames received from port i can be forwarded.
- FWCEPRC2 register enables exceptional path for error descriptors.
- FWEIS0i interrupt register notifies that an ethernet descriptor has been rejected because of an error.

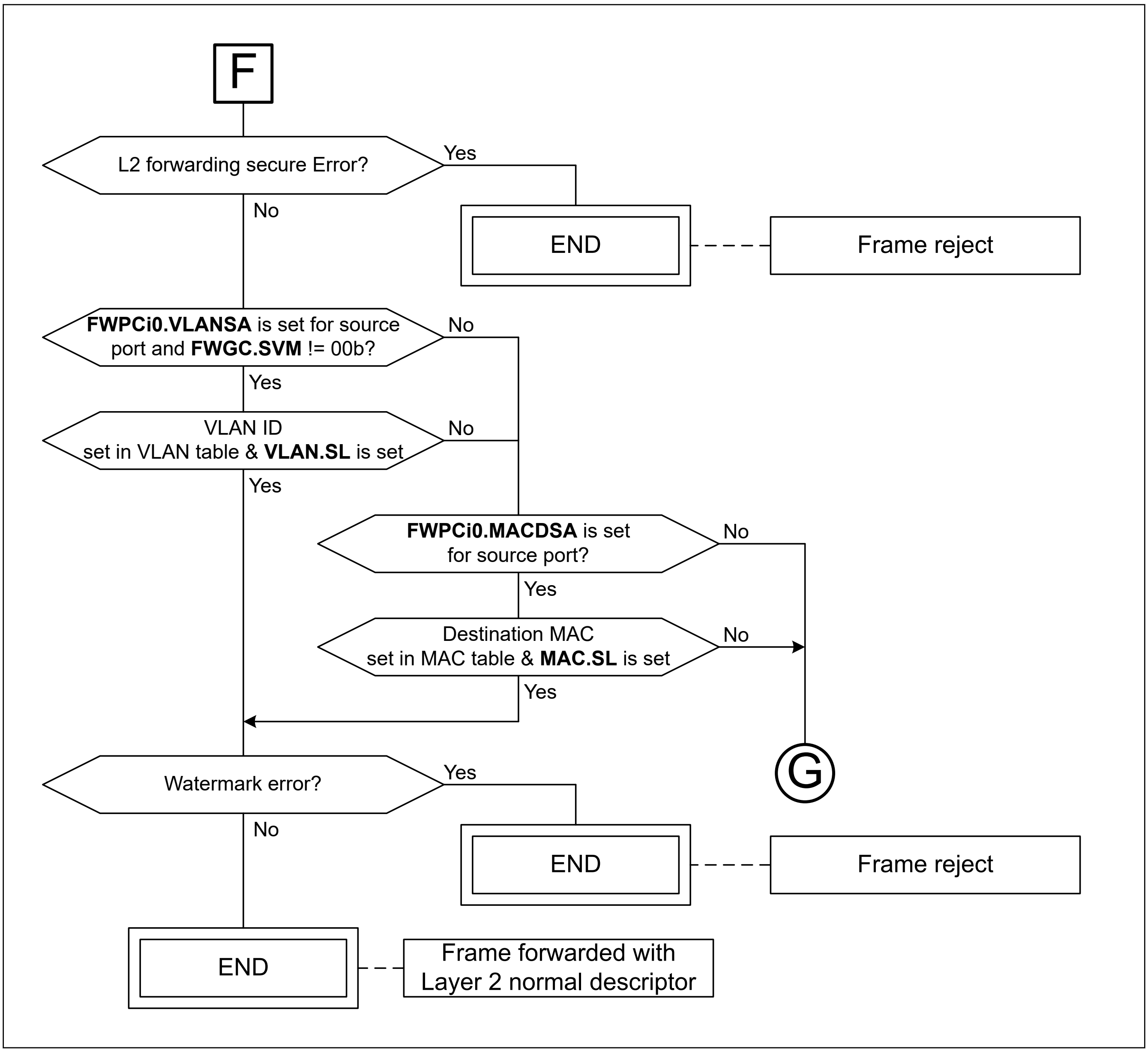


Figure 30.95 L2 forwarding secure flow

Note:
- L2 forwarding secure errors are described in section (1) L2 forwarding errors.
- Watermark errors are described in section (1) Watermark errors.
- Link F is linked from Figure 30.85 point E.
- Link G links to Figure 30.99.

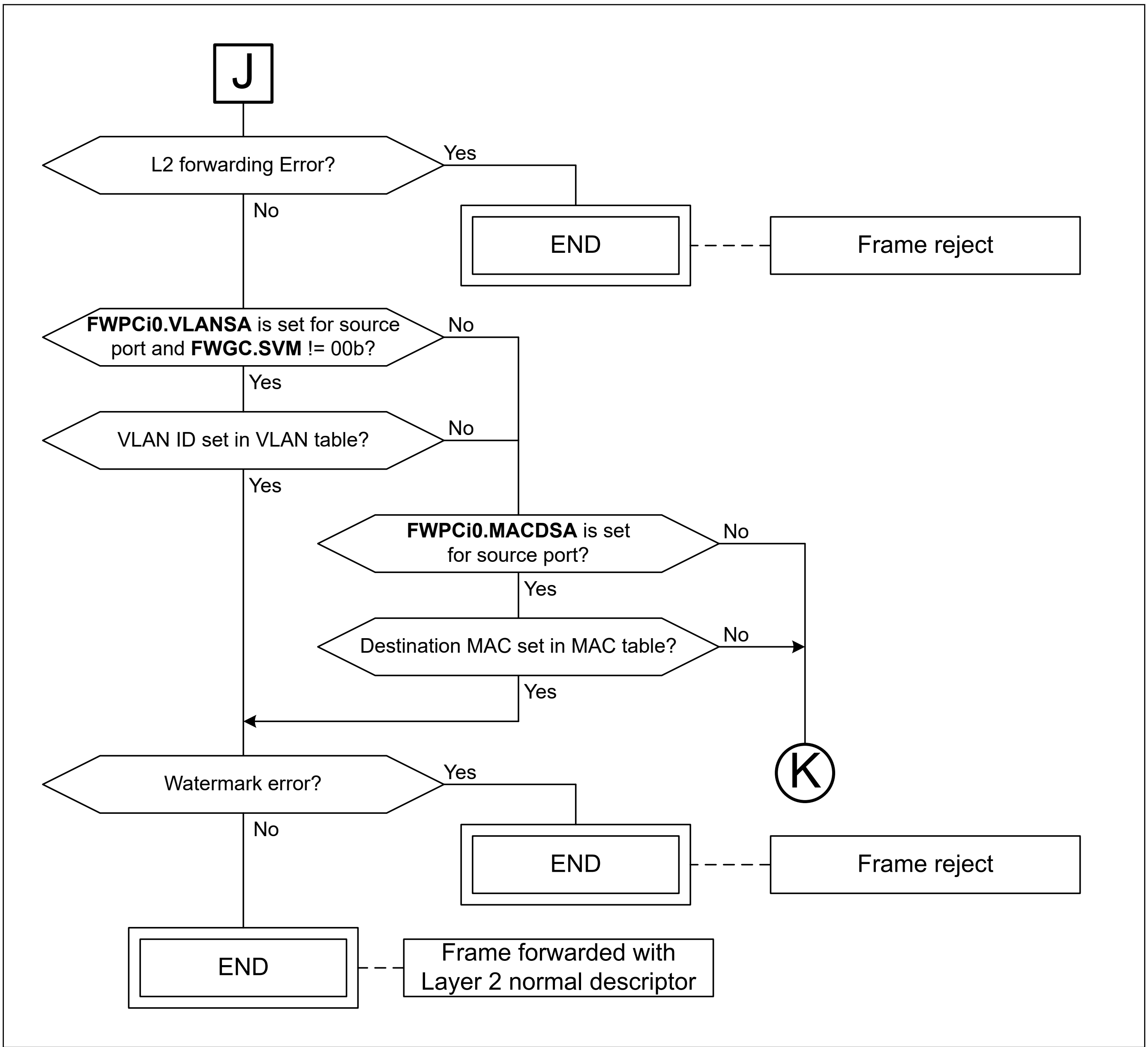


Figure 30.96 L2 forwarding unsecure flow

Note:
- L2 forwarding errors are described in section (1) L2 forwarding errors.
- Watermark errors are described in section (1) Watermark errors.
- Link J is linked from Figure 30.86 point I.
- Link K links to Figure 30.100.

(1) L2 forwarding errors

Table 30.56 describes the L2 forwarding errors. Instead of being discarded by forwarding engine, an error frame can be forwarded to exceptional path using Layer 2 exceptional descriptor described in section (3) Layer 2 exceptional descriptor by setting corresponding error exceptional bit. For error set conditions, refer to corresponding error interrupt register explanations.

Table 30.56 L2 forwarding Errors (1 of 2)

Error	Error name	Error interrupt	Error exceptional bit
L2 forwarding secure errors			
SNTF	Secure No Target Filtering	FWEIS0i.LTWNTFS	FWCEPRC2.FLTWNTFEF
SDSPF	Secure Destination Source Port Filtering	FWEIS0i.LTWDSPFS	FWCEPRC2.FDMACSLFEF

Table 30.56 L2 forwarding Errors (2 of 2)

Error	Error name	Error interrupt	Error exceptional bit
SSSPF	Secure Source Source Port Filtering	FWEIS0i.LTWSSPFS	FWCEPRC2.FSMACSLFEF
SVSPF	Secure VLAN Source Port Filtering	FWEIS0i.LTWVSPFS	FWCEPRC2.FVLANSLFEF
SDUF	Secure Destination Unknown filtering	FWEIS0i.LTWDUFS	FWCEPRC2.FDMACUFEF
SSUF	Secure Source Unknown filtering	FWEIS0i.LTWSUFS	FWCEPRC2.FSMACUFEF
SVUF	Secure VLAN Unknown filtering	FWEIS0i.LTWVUFS	FWCEPRC2.FVLANUFEF
L2 forwarding errors			
NTF	No Target Filtering	FWEIS0i.LTWNTFS	FWCEPRC2.FLTWNTFEF
DSPF	Destination Source Port Filtering	FWEIS0i.LTWDSPFS	FWCEPRC2.FDMACSLFEF
SSPF	Source Source Port Filtering	FWEIS0i.LTWSSPFS	FWCEPRC2.FSMACSLFEF
VSPF	VLAN Source Port Filtering	FWEIS0i.LTWVSPFS	FWCEPRC2.FVLANSLFEF
DUF	Destination Unknown filtering	FWEIS0i.LTWDUFS	FWCEPRC2.FDMACUFEF
SUF	Source Unknown filtering	FWEIS0i.LTWSUFS	FWCEPRC2.FSMACUFEF
VUF	VLAN Unknown filtering	FWEIS0i.LTWVUFS	FWCEPRC2.FVLANUFEF

(2) Layer 2 normal descriptor

For Layer 2 normal descriptor creation, only MAC table fields read from MAC table thanks to the destination MAC and VLAN table fields read from VLAN table are used. Layer 2 normal descriptor fields are described in Table 30.57.

Table 30.57 Layer 2 normal descriptor fields

Field name FDESCR.	Values if MAC destination address and VLAN ID are both found and MAC.SL = VLAN.SL	Values if MAC destination only is found or if MAC destination address and VLAN ID are both found and MAC.SL & !VLAN.SL	Values if VLAN ID only is found or if MAC destination address and VLAN ID are both found and !MAC.SL & VLAN.SL
DV	MAC.DV & VLAN.DV & ! FWPCi2.LTWFM	MAC.DV & !FWPCi2.LTWFM	VLAN.DV & !FWPCi2.LTWFM
SEC	MAC.SL	MAC.SL	VLAN.SL
CSD	MAC.CSD	MAC.CSD	VLAN.CSD
IPV	If MAC.IPU is set, set to MAC.IPV If MAC.IPU is not set and VLAN.IPU is set, set to VLAN.IPV If MAC.IPU and VLAN.IPU both are not set, set to the frame input priority (section section 30.5.1.4. ESWM Frame Input Priority)	If MAC.IPU is set, set to MAC.IPV If MAC.IPU is not set, set to the frame input priority (section section 30.5.1.4. ESWM Frame Input Priority)	If VLAN.IPU is set, set to VLAN.IPV If VLAN.IPU is not set, set to the frame input priority (section section 30.5.1.4. ESWM Frame Input Priority)
MINFO	See Figure 30.97 and Table 30.58.	See Figure 30.97 and Table 30.58.	See Figure 30.97 and Table 30.58.
RV	0	0	0
RN	ALL0	ALL0	ALL0
SEQN	Set to frame input sequence number	Set to frame input sequence number	Set to frame input sequence number

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	—		FL2C[1:0]		FWDC[3:0]			
1	—							
2								

Figure 30.97 Layer 2 normal descriptor MINFO format

Table 30.58 Layer 2 normal descriptor MINFO field explanation

Field name FDESCR.	Bit width	Field explanation	Values
FWDC	4	ForWarDing Code Present in all descriptors, it allows identifying the type of forwarding.	Fixed to 9 for Layer 2 normal descriptors
FL2C	2	Forwarding layer 2 code	If MAC destination address and VLAN ID are both found and MAC.SL = VLAN.SL, set to 0. If MAC destination only is found or if MAC destination address and VLAN ID are both found and MAC.SL & !VLAN.SL, set to 1. If VLAN ID only is found or if MAC destination address and VLAN ID are both found and ! MAC.SL & VLAN.SL, set to 2.
—	- -	Reserved field	Set to 0

(3) Layer 2 exceptional descriptor

Layer 2 exceptional descriptor fields are described.

Table 30.59 Layer 2 exceptional descriptor fields

Field name FDESCR.	Values
DV	Exceptional descriptors are forwarded to GWCA number FWCEPTC.EPCS (port number 2 + FWCEPTC.EPCS)
SEC	FWCEPTC.EPSL
CSDj	FWCEPTC.EPCSD
IPV	FWCEPTC.EPIPV
MINFO	See Figure 30.98 and Table 30.60.
RV	0
RN	ALL0
SEQN	Set to frame input sequence number

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	—				FWDC[3:0]			
1	NTF	SVUF	SSUF	SDUF	SVSPF	SSSPF	SDSPF	SNTF
2	—		VUF	SUF	DUF	VSPF	SSPF	DSPF

Figure 30.98 Layer 2 exceptional descriptor MINFO format

Table 30.60 Layer 2 exceptional descriptor MINFO field explanation (1 of 2)

Field name FDESCR.	Bit width	Field explanation	Values
FWDC	4	ForWarDing Code Present in all descriptors, it allows identifying the type of forwarding.	Fixed to 10 for Layer 2 exceptional descriptors
SNTF	1	Secure No Target Filtering	See (1) L2 forwarding errors.
SDSPF	1	Secure Destination Source Port Filtering	See (1) L2 forwarding errors.
SSSPF	1	Secure Source Source Port Filtering	See (1) L2 forwarding errors.
SVSPF	1	Secure VLAN Source Port Filtering	See (1) L2 forwarding errors.
SDUF	1	Secure Destination Unknown filtering	See (1) L2 forwarding errors.
SSUF	1	Secure Source Unknown filtering	See (1) L2 forwarding errors.
SVUF	1	Secure VLAN Unknown filtering	See (1) L2 forwarding errors.

Table 30.60 Layer 2 exceptional descriptor MINFO field explanation (2 of 2)

Field name FDESCR.	Bit width	Field explanation	Values
NTF	1	No Target Filtering	See (1) L2 forwarding errors.
DSPF	1	Destination Source Port Filtering	See (1) L2 forwarding errors.
SSPF	1	Source Source Port Filtering	See (1) L2 forwarding errors.
VSPF	1	VLAN Source Port Filtering	See (1) L2 forwarding errors.
DUF	1	Destination Unknown filtering	See (1) L2 forwarding errors.
SUF	1	Source Unknown filtering	See (1) L2 forwarding errors.
VUF	1	VLAN Unknown filtering	See (1) L2 forwarding errors.

30.5.2.6 Port Based Forwarding

Port based forwarding aims at forwarding all frames from a port to other ones. It used FWPBFCi and FWPBFCSDCi0 registers. Port based forwarding will forward frames following Port based forwarding secure flow in and Port based forwarding unsecure flow in .

Functions:

- FWPBFCi and FWPBFCSDCi0 registers are used to set forwarding information for frames coming from port i.

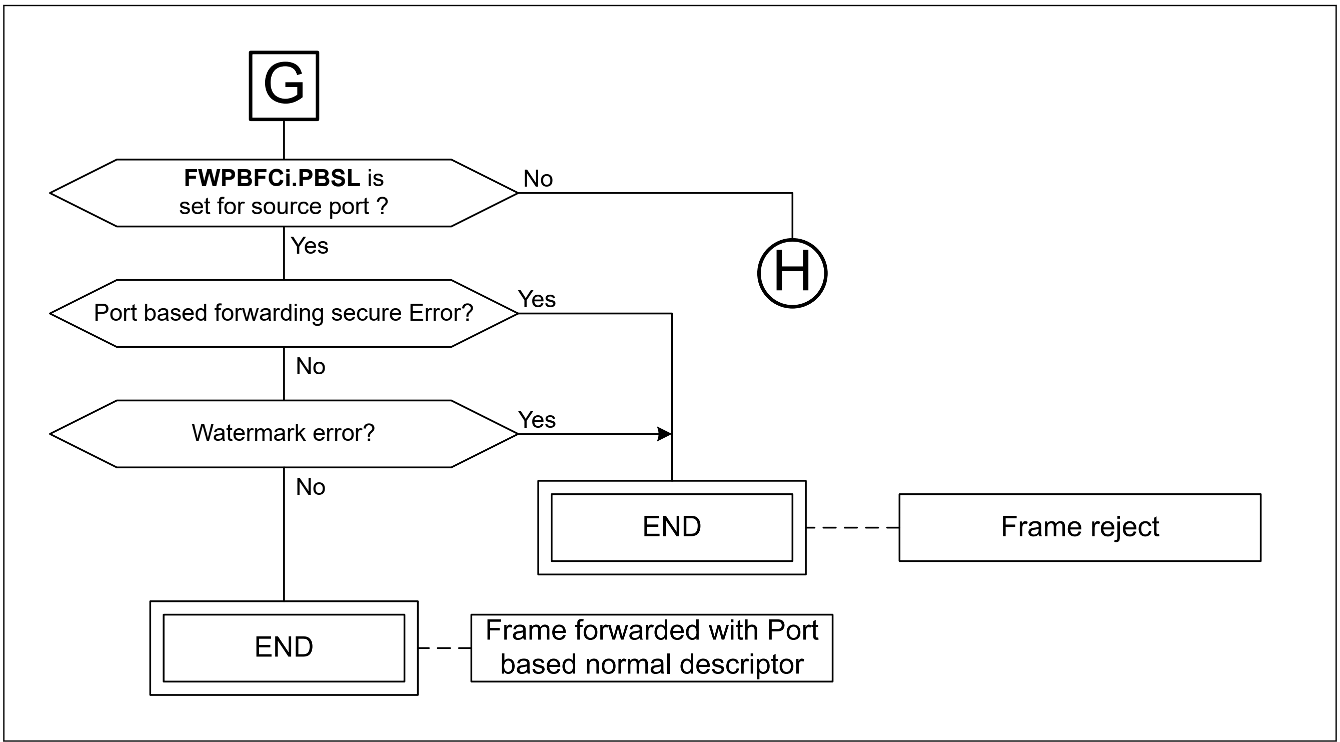


Figure 30.99 Port based forwarding secure flow

Note:
- Port based forwarding secure errors are described in section.
- Watermark errors are described in section (1) Watermark errors.
- Link G is linked from Figure 30.95.
- Link H links to Figure 30.86.

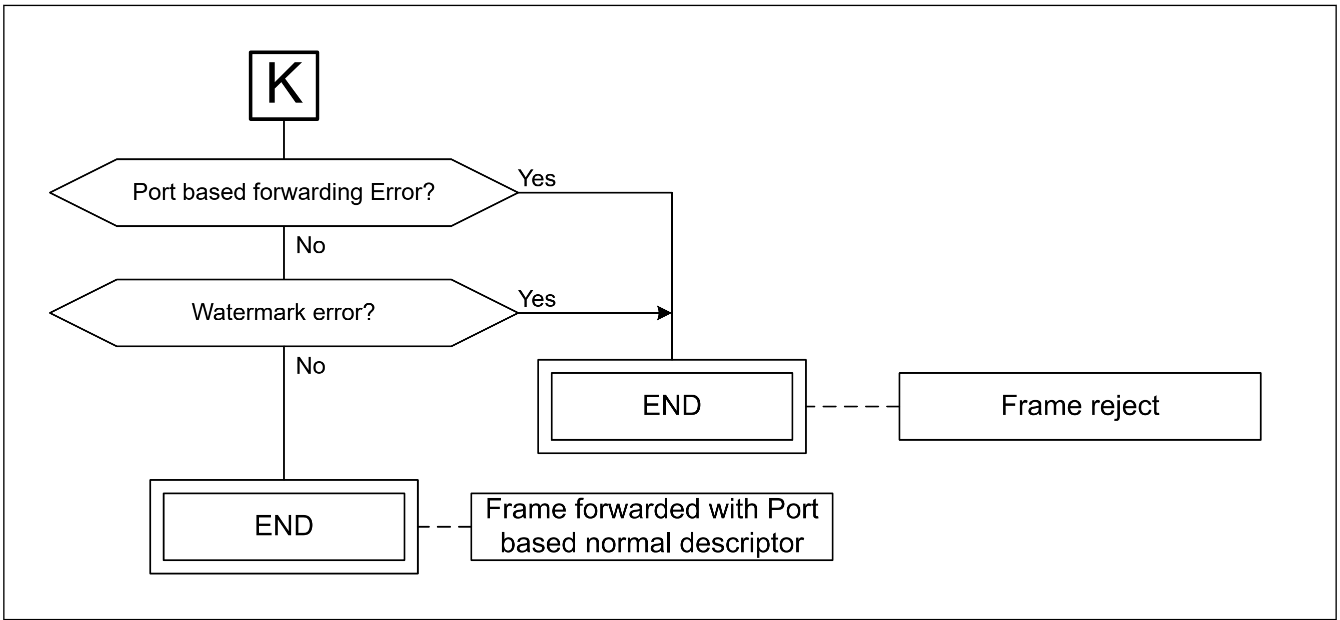


Figure 30.100 Port based forwarding unsecure flow

Note:
- Port based forwarding errors are described in (1) L2 forwarding errors.
- Watermark errors are described in (1) Watermark errors.
- Link K is linked from Figure 30.96.

30.5.2.6.1 Port based Forwarding Errors

Table 30.61describes the Port based forwarding errors. Instead of being discarded by forwarding engine, an error frame can be forwarded to exceptional path using Port based exceptional descriptor described in section 30.5.2.6.3. Port Based Exceptional Descriptor by setting corresponding error exceptional bit. For error set conditions, refer to corresponding error interrupt register explanations.

Table 30.61 Port based forwarding Errors

Error	Error name	Error interrupt	Error exceptional bit
Port based forwarding secure errors			
SNTF	Secure No Target Filtering	FWEIS0i.PBNTFS	FWCEPRC2.FPBNTFEF
Port based forwarding errors			
NTF	No Target Filtering	FWEIS0i.PBNTFS	FWCEPRC2.FPBNTFEF

30.5.2.6.2 Port based Normal Descriptor

Port based normal descriptor fields are described in Table 30.62.

Table 30.62 Port based normal descriptor fields (1 of 2)

Field name FDESCR.	Values
DV	FWPBFCi.PBDV
SEC	FWPBFCi.PBSL
CSDj	FWPBFCSDCi0.PBCSD
IPV	If FWPBFCi.PBIPU is set, set to FWPBFCi.PBIPV If FWPBFCi.PBIPU is not set, set to the frame input priority (section section 30.5.1.4. ESWM Frame Input Priority)
MINFO	Refer to Figure 30.101 and Table 30.63
RV	0
RN	ALL0

Table 30.62 Port based normal descriptor fields (2 of 2)

Field name FDESCR.	Values
SEQN	Set to frame input sequence number

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	—				FWDC[3:0]			
1	—							
2								

Figure 30.101 Port based normal descriptor MINFO format

Table 30.63 Port based normal descriptor MINFO field explanation

Field name FDESCR.	Bit width	Field explanation	Values
FWDC	4	ForWarDing Code Present in all descriptors, it allows identifying the type of forwarding.	Fixed to 11 for Port based normal descriptors
—	—	Reserved field	Set to 0

30.5.2.6.3 Port Based Exceptional Descriptor

Port based exceptional descriptor fields are described in Table 30.64.

Table 30.64 Port based exceptional descriptor fields

Field name FDESCR.	Values
DV	Exceptional descriptors are forwarded to GWCA number FWCEPTC.EPCS (port number 2)
SEC	FWCEPTC.EPSL
CSDj	FWCEPTC.EPCSD
IPV	FWCEPTC.EPIPV
MINFO	See Figure 30.102 and Table 30.65.
RV	0
RN	ALL0
SEQN	Set to frame input sequence number

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	—				FWDC[3:0]			
1	—						NTF	SNTF
2	—							

Figure 30.102 Port based exceptional descriptor MINFO format

Table 30.65 Port based exceptional descriptor MINFO field explanation (1 of 2)

Field name FDESCR.	Bit width	Field explanation	Values
FWDC	4	ForWarDing Code Present in all descriptors, it allows identifying the type of forwarding.	Fixed to 12 for Port based exceptional descriptors

Table 30.65 Port based exceptional descriptor MINFO field explanation (2 of 2)

Field name FDESCR.	Bit width	Field explanation	Values
SNTF	1	Secure No Target Filtering	See section 30.5.2.6.1. Port based Forwarding Errors.
NTF	1	No Target Filtering	See section 30.5.2.6.1. Port based Forwarding Errors.

30.5.2.7 Layer 2/Layer 3 Update

Layer 2/Layer 3 Update aims at updating frames before sending them. Agents will use the L23 update interface to fetch the update information from the forwarding engine and will automatically update the frames following the Layer 2/Layer 3 Update rule. Frames Layer 2/Layer 3 Update information fetching will happen for Layer 3 forwarding/routing/filtering frames with respectively L3.RV set.

Layer 2/Layer 3 Update is divided in the following items:

- Rule format: Determines what information will be store along with the stream ID in the table.
- Learning: Function to add/suppress a Layer 2/Layer 3 Update rule by software.
- Reading: Function to read a Layer 2/Layer 3 Update rule by software.
- Rule fetching: Function used by agents to read a Layer 2/Layer 3 Update rule.

30.5.2.7.1 Rule Format

The Layer 2/Layer 3 Update RAM is used to store Layer 2/Layer 3 Update rules. A Layer 2/Layer 3 Update rule contains the information for the frame routing. All the fields in this table, if quoted, will be written L23U.{Field name}. Table 30.66 describes fields contained in a L3 rule.

Table 30.66 Layer 2/Layer 3 Update rule format (1 of 2)

Field name	Field size (bit)	Field Explanation
RPV	4	Routing Port Valid. Values: • Bit i set to 0: Routing rule won't apply for port i. • Bit i set to 1: Routing rule will apply for port i. Functions: • If set to all0, this field will disable the corresponding routing entry. In case RPV is 0 for an egress port, the frame is not updated and will be transmitted unmodified.
TTLU	1	TimeToLive update Functions: • If this bit is set and the corresponding frame is detected as an IPv4 frame, the TTL field will be decremented by 1 and the header checksum will be updated accordingly. • If this bit is set and the corresponding frame is detected as an IPv6 frame, the HopLimit field will be decremented by 1.
MDAU	1	MAC Destination Address Update Functions: • If this bit is set, the rule corresponding frame MAC address will be updated to L23U.MDA.
MSAU	1	MAC Source Address Update Functions: • If this bit is set, the rule corresponding frame MAC address will be updated to corresponding agent MAC address.
CVIDU	1	C-tag VLAN ID Update Functions: • If this bit is set, the rule corresponding frame C-tag VLAN ID will be updated to L23U.CVID.
CPCPU	1	C-tag PCP Update Functions: • If this bit is set, the rule corresponding frame C-tag PCP will be updated to L23U.CPCP.

Table 30.66 Layer 2/Layer 3 Update rule format (2 of 2)

Field name	Field size (bit)	Field Explanation
CDEIU	1	C-tag DEI Update Functions: ● If this bit is set, the rule corresponding frame C-tag DEI will be updated to L23U.CDEI.
SVIDU	1	S-tag VLAN ID Update Functions: ● If this bit is set, the rule corresponding frame S-tag VLAN ID will be updated to L23U.SVID.
SPCPU	1	S-tag PCP Update Functions: ● If this bit is set, the rule corresponding frame S-tag PCP will be updated to L23U.SPCP.
SDEIU	1	S-tag DEI Update Functions: ● If this bit is set, the rule corresponding frame S-tag DEI will be updated to L23U.SDEI.
RTU	2	R-TAG update Values: ● 00b: Frame received without R-TAG will be forwarded without R-TAG, Frame received with R-TAG will be forwarded with an R-TAG containing FDESCR.SEQN sequence number. ● 01b: All frames will be forwarded with an R-TAG containing FDESCR.SEQN sequence number (used for sequence number generation, refer to section section 30.5.3.2.2. Sequence Number Generation). ● 10b: All frames will be forward without R-TAG (used to strip R-TAG after elimination, refer to section (1) Stream identification, FRER recovery, initial stream recovery).
MDA	48	MAC Destination Address Functions: ● Used to update MAC destination address.
CVID	12	C-tag VLAN ID Functions: ● Used to update C-tag VLAN ID.
CPCP	3	C-tag PCP Functions: ● Used to update C-tag PCP.
CDEI	1	C-tag DEI Functions: ● Used to update C-tag DEI.
SVID	12	S-tag VLAN ID Functions: ● Used to update S-tag VLAN ID.
SPCP	3	S-tag PCP Functions: ● Used to update S-tag PCP.
SDEI	1	S-tag DEI Functions: ● Used to update S-tag DEI.

30.5.2.7.2 SW Learning

Learning is used to add/remove an entry in the Layer 2/Layer 3 Update rule table.Table 30.67 describes register used to learn an entry in the Layer 2/Layer 3 Update rule table. Table 30.68 describes the learn results.

Table 30.67 Layer 2/Layer 3 Update rule learn registers (1 of 2)

Register name	Field name/corresponding field in l23u table	Field explanation
FWL23URL0.L23URNL	Routing number Learn	Address that will be used to learn an entry in the Layer 2/Layer 3 Update rule table
FWL23URL0.L23URPVL	L23U.RPV	See section 30.5.2.7.1. Rule Format .
FWL23URL1.L23UTTLUL	L23U.TTLU	See section 30.5.2.7.1. Rule Format .

Table 30.67 Layer 2/Layer 3 Update rule learn registers (2 of 2)

Register name	Field name/corresponding field in l23u table	Field explanation
FWL23URL1.L23UMDAUL	L23U.MDAU	See section 30.5.2.7.1. Rule Format .
FWL23URL1.L23UMSAUL	L23U.MSAU	See section 30.5.2.7.1. Rule Format .
FWL23URL1.L23UCVIDUL	L23U.CVIDU	See section 30.5.2.7.1. Rule Format .
FWL23URL1.L23UCPCPUL	L23U.CPCPU	See section 30.5.2.7.1. Rule Format .
FWL23URL1.L23UCDEIUL	L23U.CDEIU	See section 30.5.2.7.1. Rule Format .
FWL23URL1.L23USVIDUL	L23U.SVIDU	See section 30.5.2.7.1. Rule Format .
FWL23URL1.L23USPCPUL	L23U.SPCPU	See section 30.5.2.7.1. Rule Format .
FWL23URL1.L23USDEIUL	L23U.SDEIU	See section 30.5.2.7.1. Rule Format.
FWL23URL1.L23URTUL	L23U.RTU	See section 30.5.2.7.1. Rule Format.
FWL23URL1.L23UMDALP0/ FWL23URL2.L23UMDALP1	L23U.MDA	See section 30.5.2.7.1. Rule Format.
FWL23URL3.L23UCVIDL	L23U.CVID	See section 30.5.2.7.1. Rule Format.
FWL23URL3.L23UCPCPL	L23U.CPCP	See section 30.5.2.7.1. Rule Format.
FWL23URL3.L23UCDEIL	L23U.CDEI	See section 30.5.2.7.1. Rule Format.
FWL23URL3.L23USVIDL	L23U.SVID	See section 30.5.2.7.1. Rule Format.
FWL23URL3.L23USPCPL	L23U.SPCP	See section 30.5.2.7.1. Rule Format.
FWL23URL3.L23USDEIL	L23U.SDEI	See section 30.5.2.7.1. Rule Format.

Table 30.68 Layer 2/Layer 3 Update rule learn result

Register name	Field name/ corresponding field in l23u table	Field explanation
FWL23URLR.L23ULF	Learning fail	Learning fail with happen in one of the following conditions: ● The Layer 2/Layer 3 table is not ready (FWL23UTIM.L23UTR is not set)

30.5.2.7.3 SW Reading

Reading is used to read an entry in the Layer 2/Layer 3 Update rule table.Table 30.69 describes register used to read an entry in the Layer 2/Layer 3 Update rule table. Table 30.70 describes the read results.

Table 30.69 Layer 2/Layer 3 Update rule read registers

Register name	Field name/corresponding field in l23u table	Field explanation
FWL23URR.L23RNR	Routing number Read	Address that will be used to read an entry in the Layer 2/Layer 3 Update rule table

Table 30.70 Layer 2/Layer 3 Update rule read result (1 of 2)

Register name	Field name/corresponding field in l23u table	Field explanation
FWL23URRR0.L23URPVL	L23U.RPV	See section 30.5.2.7.1. Rule Format.
FWL23URRR1.L23UTTLUR	L23U.TTLU	See section 30.5.2.7.1. Rule Format.
FWL23URRR1.L23UMDAUR	L23U.MDAU	See section 30.5.2.7.1. Rule Format.
FWL23URRR1.L23UMSAUR	L23U.MSAU	See section 30.5.2.7.1. Rule Format.
FWL23URRR1.L23UCVIDUR	L23U.CVIDU	See section 30.5.2.7.1. Rule Format.
FWL23URRR1.L23UCPCPUR	L23U.CPCPU	See section 30.5.2.7.1. Rule Format.
FWL23URRR1.L23UCDEIUR	L23U.CDEIU	See section 30.5.2.7.1. Rule Format.

Table 30.70 Layer 2/Layer 3 Update rule read result (2 of 2)

Register name	Field name/corresponding field in l23u table	Field explanation
FWL23URRR1.L23USVIDUR	L23U.SVIDU	See section 30.5.2.7.1. Rule Format.
FWL23URRR1.L23USPCPUR	L23U.SPCPU	See section 30.5.2.7.1. Rule Format.
FWL23URRR1.L23USDEIUR	L23U.SDEIU	See section 30.5.2.7.1. Rule Format.
FWL23URRR1.L23URTUR	L23U.RTU	See section 30.5.2.7.1. Rule Format.
FWL23URRR1.L23UMDARP0/ FWL23URRR2.L23UMDARP1/	L23U.MDA	See section 30.5.2.7.1. Rule Format.
FWL23URRR3.L23UCVIDR	L23U.CVID	See section 30.5.2.7.1. Rule Format.
FWL23URRR3.L23UCPCPR	L23U.CPCP	See section 30.5.2.7.1. Rule Format.
FWL23URRR3.L23UCDEIR	L23U.CDEI	See section 30.5.2.7.1. Rule Format.
FWL23URRR3.L23USVIDR	L23U.SVID	See section 30.5.2.7.1. Rule Format.
FWL23URRR3.L23USPCPR	L23U.SPCP	See section 30.5.2.7.1. Rule Format.
FWL23URRR3.L23USDEIR	L23U.SDEI	See section 30.5.2.7.1. Rule Format.

30.5.2.7.4 HW Reading

Hardware reading is used by agents [GWCA][ETHA] to fetch the Layer 2/Layer 3 Update rule corresponding to a frame and update the frame. The read result is given to agents to perform Layer 2/Layer 3 update along with L23U.ERR which is used to notify to agents that the corresponding data should be rejected.

30.5.2.7.5 Routing Number Remapping Function

The routing number remapping function is used to change the routing number for a specific port using FWL23URMCi register. That way, different ports can use different Routing Rules for the same frame. It can be used for FRER during replication (section section 30.5.3.2.1. Replication) to send a duplicated frame through several ports with a different stream ID.

Functions:

- When FWL23URMCi.RME is set, a frame sent to port FWL23URMCi.RMDPN with FDESCR.RV set and FDESCR.RN equal to FWL23URMCi.RMRN will be routed with rule number FWL23URMCi.RMNRN.

30.5.3 Filtering

Forwarding engine allows frame filtering and shaping using the following functions:

- PSFP (Per Stream Filtering and Policing) [802.1Qci]
- ATS (Asynchronous Traffic Shaping) [802.1Qcr]
- FRER (Frame Replication and Elimination for Reliability) [802.1CB]

30.5.3.1 PSFP/ATS

PSFP functions is divided in three functions that happen sequential following forwarding flow in Figure 30.87:

- MSDU filter
- Meter filter/ATS

30.5.3.1.1 MSDU Filter

The MSDU filter aims at checking frame size using FWPMFGCi register and can be monitored using FWEIS2 interrupt register. MSDU filtering flow for MSDU filter i is described in Figure 30.103.

Functions:

- FWPMFGCi is used to set MSDU filter number i maximum frame size and mode.
- FWEIS2 interrupt register notifies that an ethernet descriptor has been rejected because of an error.

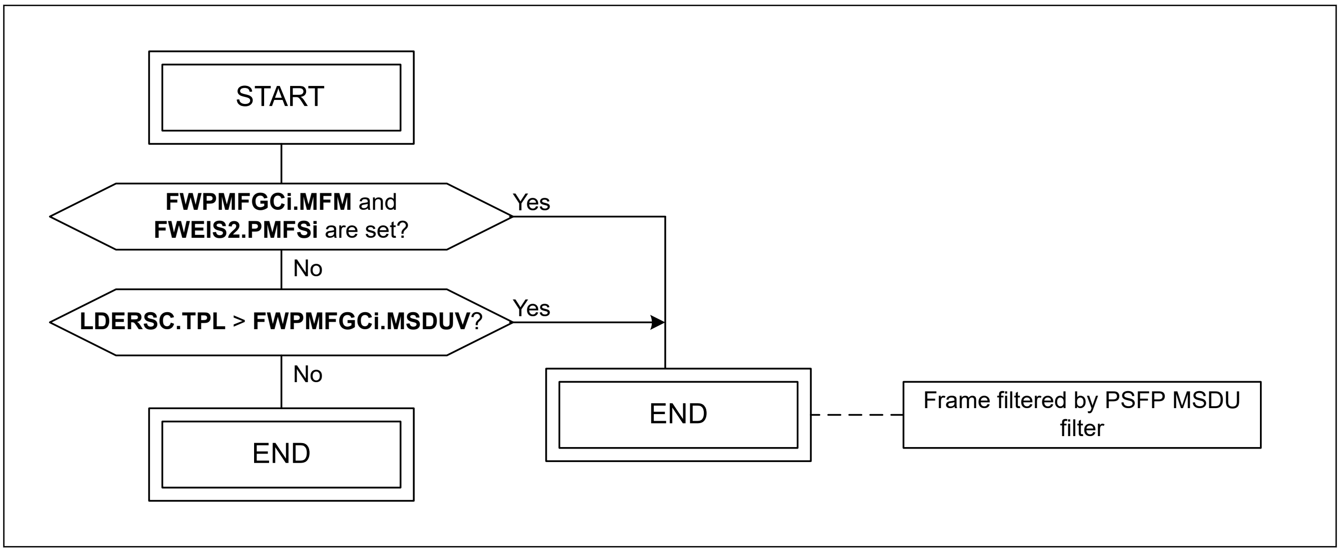


Figure 30.103 MSDU filtering flow for MSDU filter i

30.5.3.1.2 Meter Filter/ATS

The meter filter aims at checking a stream throughput or, in ATS, at controlling a stream throughput using FWPMTRFCi, FWPMTRCBSCi, FWPMTRCIRCi, FWPMTREBSCi, FWPMTREIRCi registers and can be monitored using FWPMTRFMi register and FWEIS5 interrupt registers. Meter filtering flow for meter i is described in Figure 30.104.

Functions:

- FWPMTRFCi register is used to set Meter filter i modes.
- FWPMTRCBSCi and FWPMTRCIRCi registers are used to set Meter filter i green bucket.
- FWPMTREBSCi and FWPMTREIRCi registers are used to set Meter filter i yellow bucket.
- FWPMTRFMi is used to monitor ATS shaper i.
- FWEIS5 interrupt register notifies that an ethernet descriptor has been rejected because of an error.

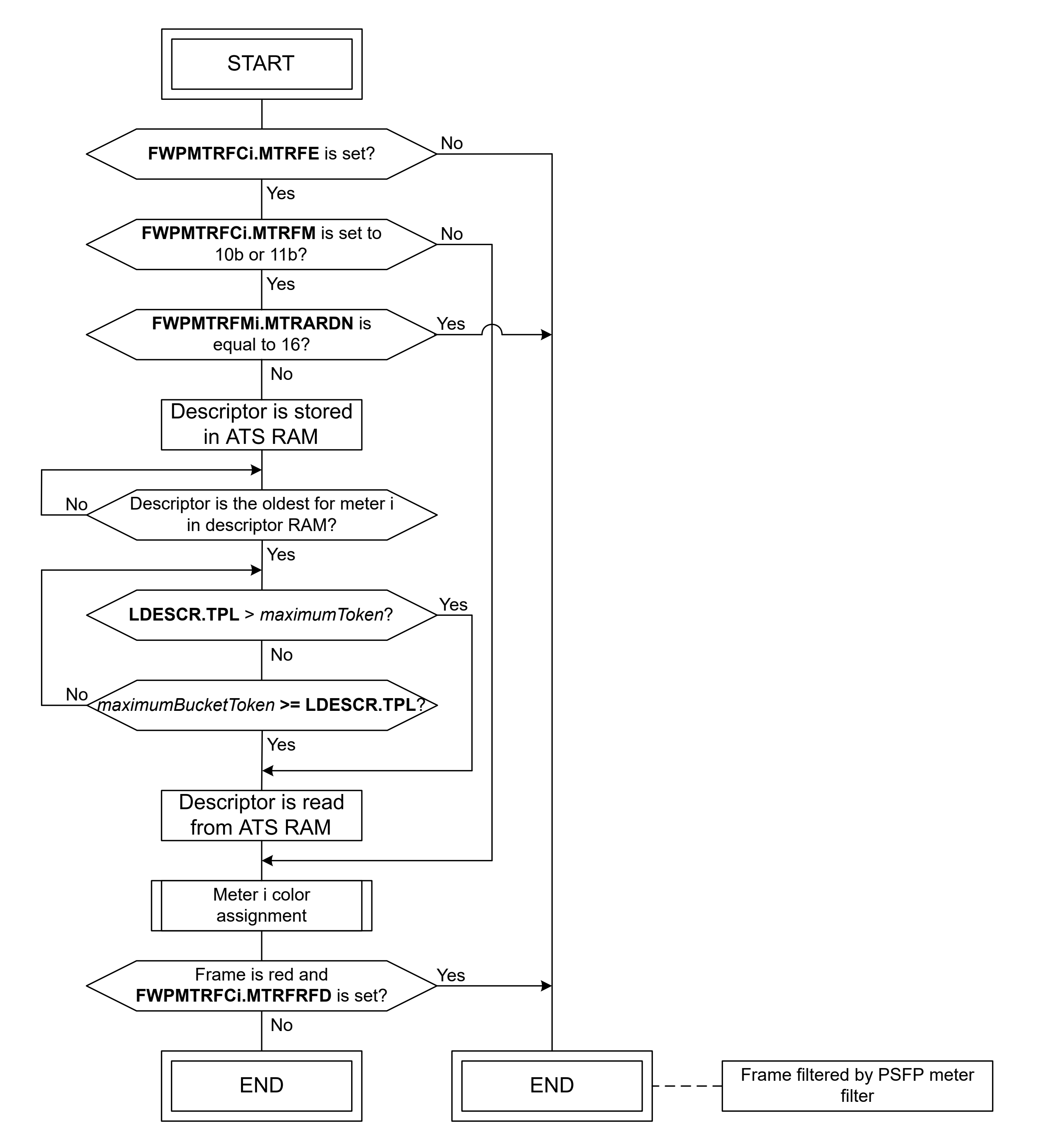


Figure 30.104 Meter filtering flow for meter i

Note:
- For Meter i color assignment refer to Figure 30.105 and Figure 30.106
- In Figure 30.104, *maximumToken* is the maximum number of tokens that can be contained in one meter biggest bucket. For Double bucket meters, it corresponds to the maximum value between FWPMTRCBSCi and FWPMTREBSCi and for Single bucket meters, it corresponds to FWPMTRCBSCi value. Any frame rejected by "LDESCR.TPL [GWCA] > *maximumToken*" condition will be marked as red.
- In Figure 30.104, *maximumBucketToken* is, for a double bucket the maximum number between the number of tokens actually contained in the yellow bucket and the number of tokens actually contained in the green bucket, for a single bucket the number of tokens actually contained in the green bucket.
- In Figure 30.104, any frame rejected by "FWPMTRFMi.MTRARDN is equal to 16" conditions (meter corresponding ATS descriptor queue full) will have no color.

Recommendations:

- For every L3 rule mapped to a meter, it is recommended to map also an MSDU filter with a value smaller or equal to the meter corresponding *maximumToken* value. It will be allowed to know that whenever FWEIS5.PMRFSi is set, it is due to an ATS RAM overflow in ATS mode and due to a maximum throughput reached error in other modes and, not due to a frame format error (frame being too big for the corresponding meter).

(1) Double bucket meters

Double bucket meters aim at giving a color to a frame. Meters from meter number 0 to meter number 7 are Double bucket meters.

(a) Double bucket color assignment

Double bucket meter i color assignment flow is described in Figure 30.105.

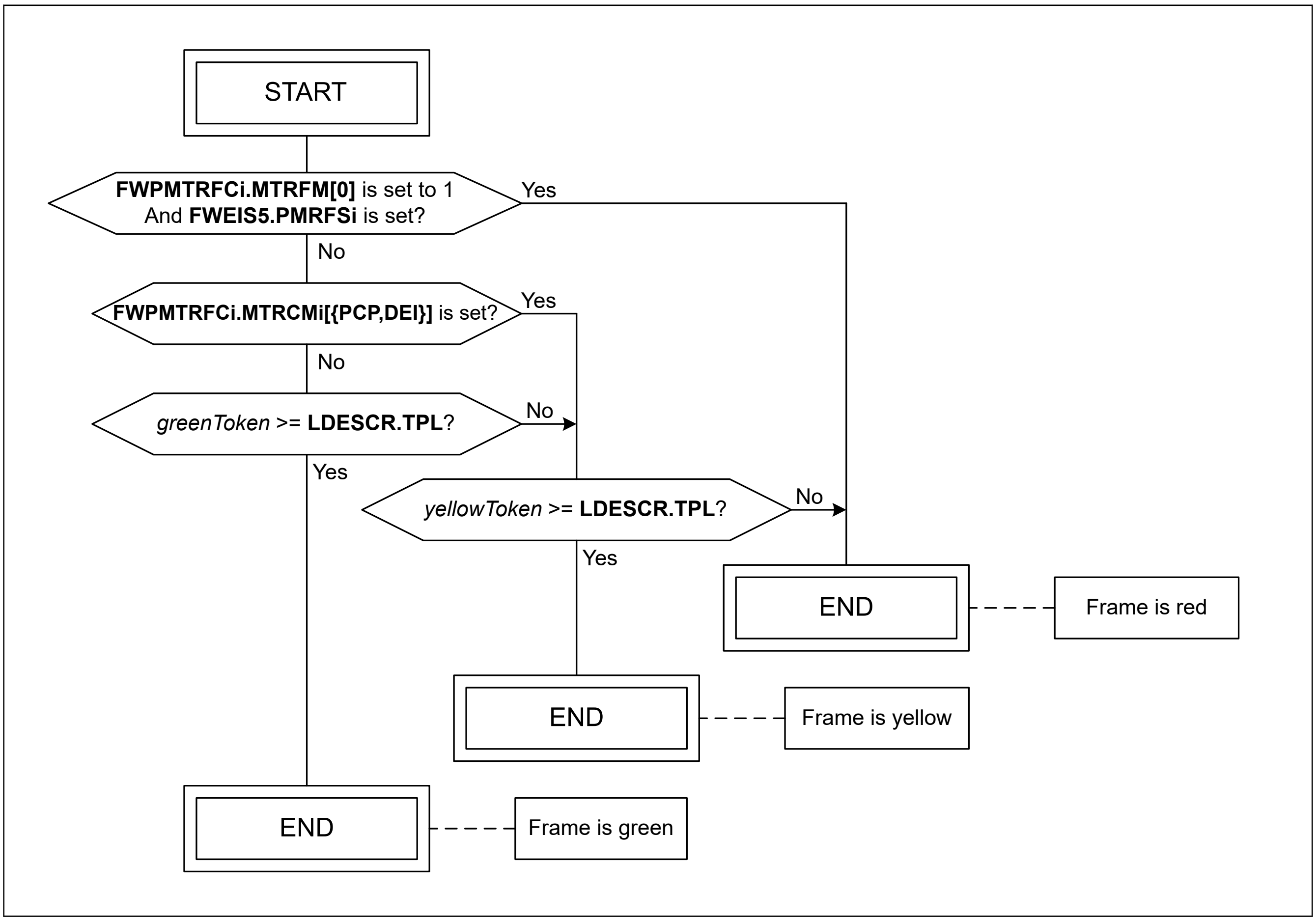


Figure 30.105 Double bucket meter i color assignment flow

In Figure 30.105:

- PCP and DEI are respectively the frame incoming S-TAG PCP and DEI values if **FWGC.SVM** is set to 10b, the frame incoming C-TAG PCP and DEI values if **FWGC.SVM** is set to 01b and, 000b and 0 if **FWGC.SVM** is set to 00b.
- *greenToken* is the number of tokens contained in the green bucket.
- *yellowToken* is the number of tokens contained in the yellow bucket.

(b) Double bucket setting

For double buckets, the green bucket (‘Committed' or ‘C' bucket) and the yellow bucket (‘Excess' or ‘E' bucket) needs to be set. For detail explanations concerning coupling flag, color mode and bucket utilization, refer to MEF standard [MEF10.3]. Equation (6) shows how to set information rates in FWPMTRCIRCi.CIR and FWPMTREIRCi.EIR. FWPMTRCBSCi.CBS and FWPMTREBSCi.EBS should be respectively set to the maximum excepted burst size for green and yellow frame in byte.

(6) *InformationRate[byte/cycle]* = *MaximumThroughput[bps]*/(8**clk_freq[Hz]*)

Where:

- *InformationRate* is the value to set to FWPMTRCIRCi.CIR or FWPMTREIRCi.EIR register.
- *MaximumThroughput[bps]* is the expected maximum throughput.
- *clk_freq* is the clock frequency of ESWCLK.

[Example]

If the meter is to be programmed for information rate of 200Mbps and system clock of 100 MHz :

InformationRate = $((200 * 1000000) / (8 * 100 * 1000000)) * 2^{16}$

CIR[15:0] or EIR[15:0] = $(\underline{0}.25) * 2^{16} = 16384$ (0x4000)

CIR[19:16] or EIR[19:16] = 0 (Underlined integer)

$(\text{CIR} \times 2^{-16})$ is added to the committed token bucket every system clock.

If the meter is to be programmed for information rate of 200 Mbps and system clock of 320 MHz :

InformationRate = $((1000 * 1000000) / (8 * 320 * 1000000)) \times 2^{16}$

CIR[15:0] or EIR[15:0] = $(\underline{0}.078125) \times 2^{16} = 5120$ (0x1400)

CIR[19:16] or EIR[19:16] = 0 (Underlined integer)

$(\text{CIR} \times 2^{-16})$ is added to the committed token bucket every system clock.

Note: • HW: DropOnYellow is not available in PSFP. To achieve drop on yellow function, the yellow bucket *InformationRate* FWPMTREIRCi.EIR and FWPMTREBSCi.EBS should be set to 0. As a result, all yellow frames will be marked as red and dropped.

(2) Single bucket meters

Single bucket meters aim at giving a color to a frame. Meters from meter number 8 to meter number 31 are Single bucket meters.

(a) Single bucket color assignment

Single bucket meter i color assignment flow is described in Figure 30.106.

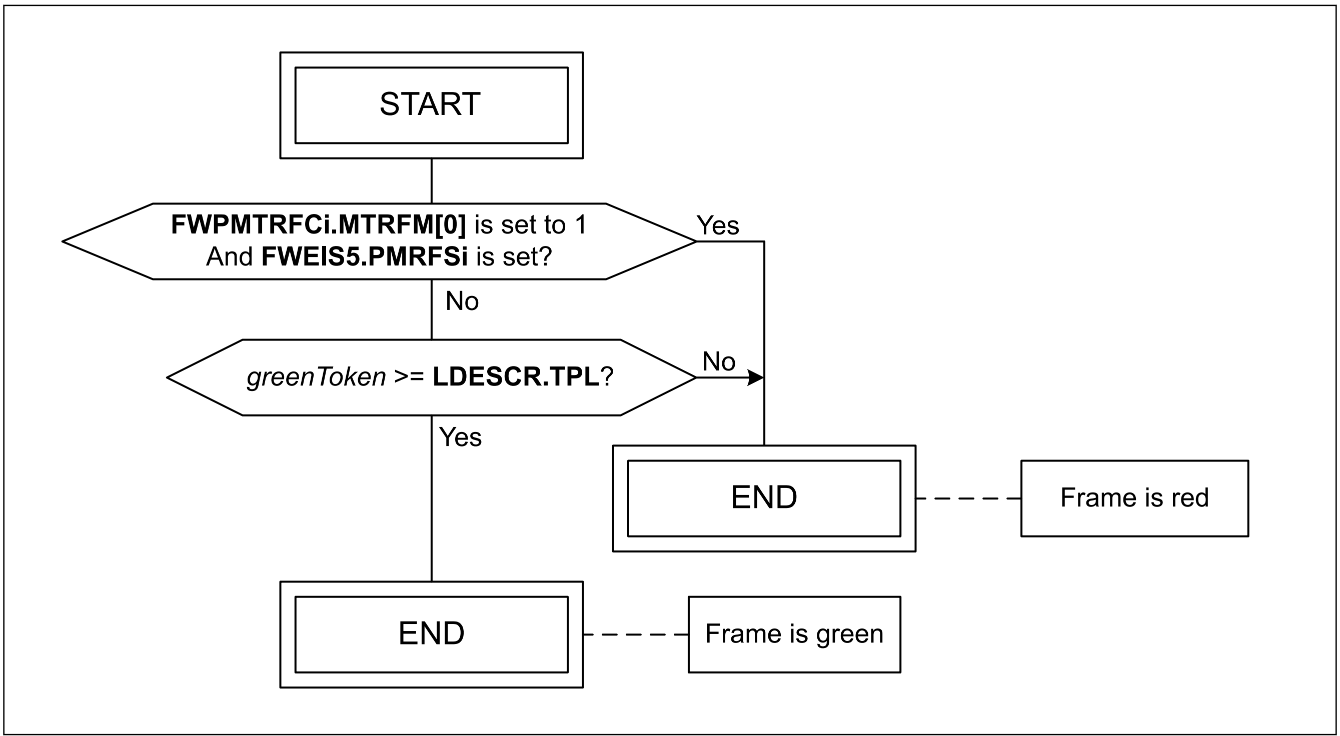


Figure 30.106 Single bucket meter i color assignment flow

In Figure 30.106:

- *greenToken* is the number of tokens contained in the green bucket.

(b) Single bucket setting

The single bucket should be set as the double bucket green bucket, refer to section (b) Double bucket setting.

30.5.3.2 FRER

FRER functions is divided in three functions that happen independently (they can be used simultaneously):

- Replication
- Sequence number generation
- Elimination

Note: • SW: Latent error detection is not implemented in HW and should be implemented in SW using FWFRPPCNi and FWFRDPCNi counters.

30.5.3.2.1 Replication

FRER replication function is used to send a same frame to different ports with different streams. To do so, Layer 3 forwarding/routing/filtering (section section 30.5.2.4. Layer 3 Forwarding/Routing/Filtering) should be used along with Layer 2/Layer 3 Update function (section section 30.5.2.7. Layer 2/Layer 3 Update).

Layer 3 forwarding/routing/filtering is used to multicast the incoming FRER stream to several ports and to map the FRER stream to a routing number. Table 30.71 shows an example where Layer 3 table is used to multicast a frame coming from port 2 to port 0 and port 1 and to map the corresponding stream to routing number 5.

Table 30.71 Layer 3 table setting for replication (1 of 2)

Field name L3.	Values	Explanation
SLV	100b	Not mandatory, ensure that the corresponding stream only comes from port 2.
DV	011b	Send the frame to port 0 and port 1
RV	1	Enable routing

Table 30.71 Layer 3 table setting for replication (2 of 2)

Field name L3.	Values	Explanation
RN	5	Map the stream to Layer 2/Layer 3 Update rule number 5

Layer 2/Layer 3 rule remapping is used to remap Routing number L3.RN for one of the ports to send the frame with different stream IDs and allow to learn the rules and the new fields for stream update. Table 30.72 shows an example where port 1 Layer 2/Layer 3 Update rule number is remapped to rule number 8 with 8.2.8.5 Routing number remapping rule number 4. Table 30.73 shown an example for Layer 2/Layer 3 Update rule setting for FRER.

Table 30.72 Layer 2/Layer 3 Update rule number remapping

Register name	Values	Explanation
FWL23URMC4.RME	1	Enable rule number remapping
FWL23URMC4.RMRN	5	Remap rule number 5
FWL23URMC4.RMDPN	1	Remap when routing access comes from port 1
FWL23URMC4.RMNRN	8	Remap rule to rule number 8

Table 30.73 Layer 2/Layer 3 Update rule settings

Field name L23U.	Values for rule 5	Values for rule 8	Explanation
RPV	001b	001b	Each rule is used for its corresponding port
TTLU	0	0	Do not update TimeToLive (user define)
MDAU	1	1	Update destination MAC address (user define)
MSAU	0	0	Do not update source MAC address (user define)
CVIDU	1	1	Update C-TAG (user define)
CPCPU	1	1	Update C-TAG (user define)
CDEIU	1	1	Update C-TAG (user define)
SVIDU	0	0	Do not update S-TAG (user define)
SPCPU	0	0	Do not update S-TAG (user define)
SDEIU	0	0	Do not update S-TAG (user define)
RTU	00b or 01b	00b or 01b	Set to 00b if the replicated stream contains an R-TAG at ingress Set to 01b if the replicated stream does not contain R-TAG at ingress (section section 30.5.3.2.2. Sequence Number Generation)
MDA	MAC0	MAC1	Set the new MAC address per each rule (user define)
CVID	VLAN0	VLAN1	Set the new VLAN ID per each rule (user define)
CPCP	PCP0	PCP1	Set the new PCP per each rule (user define)
CDEI	DEI0	DEI1	Set the new DEI per each rule (user define)
SVID	0x0	0x0	Do not update S-TAG (user define)
SPCP	000b	000b	Do not update S-TAG (user define)
SDEI	0	0	Do not update S-TAG (user define)

30.5.3.2.2 Sequence Number Generation

FRER sequence number generation is used during replication to generate a sequential sequence number to insert in stream corresponding frames (refer to section section 30.5.3.2.1. Replication for replication) using FWSEQNGCi and FWSEQNRC and can be monitored using FWSEQNGMi.

Functions:

- FWSEQNGCi register is used to enable sequence generation. While enabled, any frame forwarded with FWSEQNGCi.SEQNGRN as routing number will see its sequence number (FDESCR.SEQN) set to FWSEQNGMi.SEQN.
- FWSEQNGMi register is used to monitor the next sequence number to be used by sequence number generation rule i.

- FWSEQNRC.SEQNR is used to reset to 16’b0 the next sequence number to be used by sequence number generation rule i.

Note:
- SW: Sequence number generation should not be used along with elimination.
- A frame rejected by PSFP will not see its sequence number updated by sequence number generation.

30.5.3.2.3 Elimination

FRER elimination aims at identifying the FRER streams (Layer 3 forwarding/routing/filtering, section section 30.5.2.4. Layer 3 Forwarding/Routing/Filtering), verifying the integrity of a stream (FRER individual recovery, section (e) Individual recovery algorithms), at eliminating duplicated frames (FRER sequence recovery, (f) Sequence recovery algorithms), at recovering the stream ID (Layer 2/Layer 3 Update function (section section 30.5.2.7. Layer 2/Layer 3 Update)) and at checking that either the individual recovery or sequence recovery algorithm are stuck (Timeout function, section (3) Timeout). FRER elimination uses FWFTL0/1/R, FWFTOC and FWFTOPC registers and can be monitored using FWFTIM and FWFTR/RR0/RR1/RR2 registers and FWEIS60/61/62/63, FWEIS70/71/72/73 and FWEIS80/81/82/83 interrupt registers.

Functions:

- FWFTL0/1/R registers are used to set/reset an entry in FRER table.
- FWFTOC and FWFTOPC registers are used to set timeout function.
- FWFTIM register is used to reset the FRER table before utilization.
- FWFTR/RR0/RR1/RR2 registers are used to monitor FRER entries in FRER table.
- FWEIS60/61/62/63, FWEIS70/71/72/73 and FWEIS80/81/82/83 interrupt registers notify that an error happened during FRER elimination.

Note:
- SW: Sequence number generation should not be used along with elimination.

(1) Stream identification, FRER recovery, initial stream recovery

In this section, example will be given to explain the different function dependencies and settings. In this example two streams, 0 and 1, are respectively entering the switch from port 0 and 1 that separately should pass through individual recovery algorithm for integrity check and through the same sequence recovery algorithm for duplicate elimination. The eliminated stream should see its stream ID updated to new one and should be forwarded to port 2.

Stream identification is done by Layer 3 forwarding/routing/filtering function, refer to section section 30.5.2.4. Layer 3 Forwarding/Routing/Filtering. L3 table is used to map each stream to its corresponding FRER individual recovery individual recovery algorithm and also used to map stream to a Layer 2/Layer 3 update rule for stream ID recovery. In Table 30.74 example, stream 0 and stream 1 are respectively mapped to FRER entry number 15 and 23 for individual recovery and both streams are mapped to routing rule number 5 for stream ID recovery.

Table 30.74 Layer 3 table setting for elimination

Field name L3.	Values for stream 0	Values for stream 1	Explanation
SLV	001b	010b	Not mandatory, ensure that each stream comes from its corresponding source port.
DV	100b	100b	All streams for the same sequence recovery algorithm are targeting the port
FRERV	1	1	FRER individual recovery
FRERN	15	23	All streams are targeting a different individual recovery algorithm
RV	1	1	Enable routing
RN	5	5	All streams for the same sequence recovery algorithm are targeting the same Layer 2/Layer 3 update rule for stream ID recovery

FRER table is then used to map stream 0 and 1 individual recovery algorithms to a sequence recovery algorithm and used to set all the recovery algorithms. Table 30.75 shows an example were individual recovery algorithm are mapped to FRER entry number 67 for sequence recovery.

Table 30.75 FRER table setting for sequence recovery

Field name FRER.	Values for FRER entry number 15	Values for FRER entry number 23	Values for FRER entry number 67	Explanation
SRPV	1	1	0	Enable sequence recovery for an individual recovery algorithm
SRP	67	67	All0	Sequence recovery entry number

Layer 2/Layer 3 Update functions is used to recover the output stream ID in the frame. Table 30.76 shown an example for Layer 2/Layer 3 Update rule setting for FRER.

Table 30.76 Layer 2/Layer 3 Update rule settings

Field name L23U.	Values for rule 5	Explanation
RPV	100b	Set to the target port
TTLU	0	Do not update TimeToLive (user define)
MDAU	1	Update destination MAC address (user define)
MSAU	0	Do not update source MAC address (user define)
CVIDU	1	Update C-TAG (user define)
CPCPU	1	Update C-TAG (user define)
CDEIU	1	Update C-TAG (user define)
SVIDU	0	Do not update S-TAG (user define)
SPCPU	0	Do not update S-TAG (user define)
SDEIU	0	Do not update S-TAG (user define)
RTU	00b or 01b	Set to 00b if the output stream should contain the ingress R-TAG Set to 10b if the output stream should be sent without R-TAG
MDA	MAC0	Set the new MAC address per each rule (user define)
CVID	VLAN0	Set the new VLAN ID per each rule (user define)
CPCP	PCP0	Set the new PCP per each rule (user define)
CDEI	DEI0	Set the new DEI per each rule (user define)
SVID	0x0	Do not update S-TAG (user define)
SPCP	000b	Do not update S-TAG (user define)
SDEI	0	Do not update S-TAG (user define)

In this section only an example has been developed, by using the same kind of settings it is possible to:

- Have only one ingress stream and one output stream with the same stream ID and only apply individual recovery.
- Have more than 2 input streams for elimination.
- Have more than 1 output stream after elimination (by applying FRER replication, section section 30.5.3.2.1. Replication)

(2) FRER table

FRER table is used to store setting and information used for recovery algorithms.

FRER table is divided in the following items:

- Rule format: Determines what information will be store in the FRER table
- SW Learning: Function to set an FRER entry in the FRER table.
- SW Reading: Function to read an FRER entry in the FRER table.

(a) Rule format

The FRER table is used to store FRER entries. A FRER entry contains information recovery algorithms. All the fields in this table, if quoted, will be written FRER.{Field name}. Table 30.77 describes fields contained in a FRER entry.

Table 30.77 FRER entry format

Field name	Field size (bit)	Field explanation
SHL	4	Sequence History Length Values: ● 0: Recovery match algorithm ● i different from 0: Recovery vector algorithm with history length of i+1
SH	15	Sequence History: Functions: ● Used to save sequence history for recovery algorithms
RSN	16	Recovery Sequence Number Functions: ● Used to save last highest sequence number (modulo 2^{16}) for recovery algorithms
TNS	1	Take No Sequence Values: ● 0: Frames received without R-TAG (LDESCR.RTGI [GWCA] not set) are rejected ● 1: Frames received without R-TAG (LDESCR.RTGI [GWCA]not set) are forwarded
SRPV	1	Sequence Recovery Pointer Valid Values: ● 0: After using this entry for individual recovery, sequence recovery will not happen ● 1: After using this entry for individual recovery, sequence recovery will happen using FRER entry number FRER.SRP
SRP	7	Sequence Recovery Pointer Functions: ● Set the number of the FRER entry that should be used for sequence recovery Note: Any entry used for sequence recovery cannot be used for individual recovery. As a result, when a Recovery number is set in FRER.SRT when FRER.SRPV is set, the same number cannot be set in L3.FRERN when L3.FRERV is set.
SRRT	10	Set Recovery Remaining Tick Functions: ● Time in tick (one tick is FWFTOC.TOCE) before timeout. ● Setting this register to All0 disable timeout function for the corresponding entry
RRT	10	Recovery Remaining Tick Functions: ● Used by timeout function to detect timeout error.

(b) SW Learning

Learning is used to set/reset an entry in the FRER table. Table 30.78 describes register used to learn an entry in the FRER table. Table 30.79 describes the learn results. To reset an entry, software should re-learn the same entry again.

Table 30.78 FRER entry learn registers

Register name	Field name/corresponding field in l23u table	Field explanation
FWFTL0.FEAL	Entry Learn address	Corresponding the entry number that should be learnt in FRER table
FWFTL1.FSHLL	FRER.SHL	See (a) Rule format.
No register Set to All0 While learning (TakeAny)	FRER.SH	See (a) Rule format.
No register Set to All1 While learning	FRER.RSN	See (a) Rule format.
FWFTL1.FTNSL	FRER.TNS	See (a) Rule format.
FWFTL1.FSRPVL	FRER.SRPV	See (a) Rule format.
FWFTL0.FSRPL	FRER.SRP	See (a) Rule format.
FWFTL1.FSRRTL	FRER.SRRT	See (a) Rule format.
No register Set to All0 While learning	FRER.RRT	See (a) Rule format.

Table 30.79 FRER entry learn result

Register name	Field name/corresponding field in FRER table	Field explanation
FWFTLR.FLF	Learning fail	Learning fail with happen in one of the following conditions: ● The FRER table is not ready (FWFTIM.FTR is not set)

(c) SW Reading

Learning is used to read an entry in the FRER table. Table 30.80 describes register used to read an entry in the FRER table. Table 30.81 describes the read results.

Table 30.80 FRER entry read registers

Register name	Field name/corresponding field in l23u table	Field explanation
FWFTR.FEAR	Entry Read address	Corresponding the entry number that should be read from FRER table

Table 30.81 FRER entry learn result

Register name	Field name/corresponding field in l23u table	Field explanation
FWFTRR0.FSHLR	FRER.SHL	See (a) Rule format.
FWFTRR1.FSHR	FRER.SH	See (a) Rule format.
FWFTRR2.FRSNR	FRER.RSN	See (a) Rule format.
FWFTRR0.FTNSR	FRER.TNS	See (a) Rule format.
FWFTRR0.FSRPVR	FRER.SRPV	See (a) Rule format.
FWFTRR0.FSRPR	FRER.SRP	See (a) Rule format.
FWFTRR0.FSRRTR	FRER.SRRT	See (a) Rule format.
FWFTRR2.FRRTR	FRER.RRT	See (a) Rule format.

(d) Recovery algorithms

Individual recovery or sequence recovery algorithm can both be performed using either a match recovery algorithm or a vector recovery algorithm. For an entry, match recovery algorithm can be selected by setting All0 to its corresponding FRER.SHL field. If any other value is set vector recovery algorithm will be selected.

Match recovery algorithm is only used to detect if a stream is stuck (two frames are received consecutively with the same sequence number).

Vector recovery algorithm is used to check a stream sequence by accepting all packets within a given range (±FRER.SHL) from the last accepted highest sequence recovery number. This algorithm also ensures than within this range no sequence number is received more than on time.

(e) Individual recovery algorithms

Individual recovery match algorithm and vector algorithm are respectively described in Figure 30.107 and Figure 30.108. The variable *Delta* present in these figures is the distance from the ingress frame sequence number to FRER.RSN (This number should be considering overflow and should be contained in [-32767:32768] range).

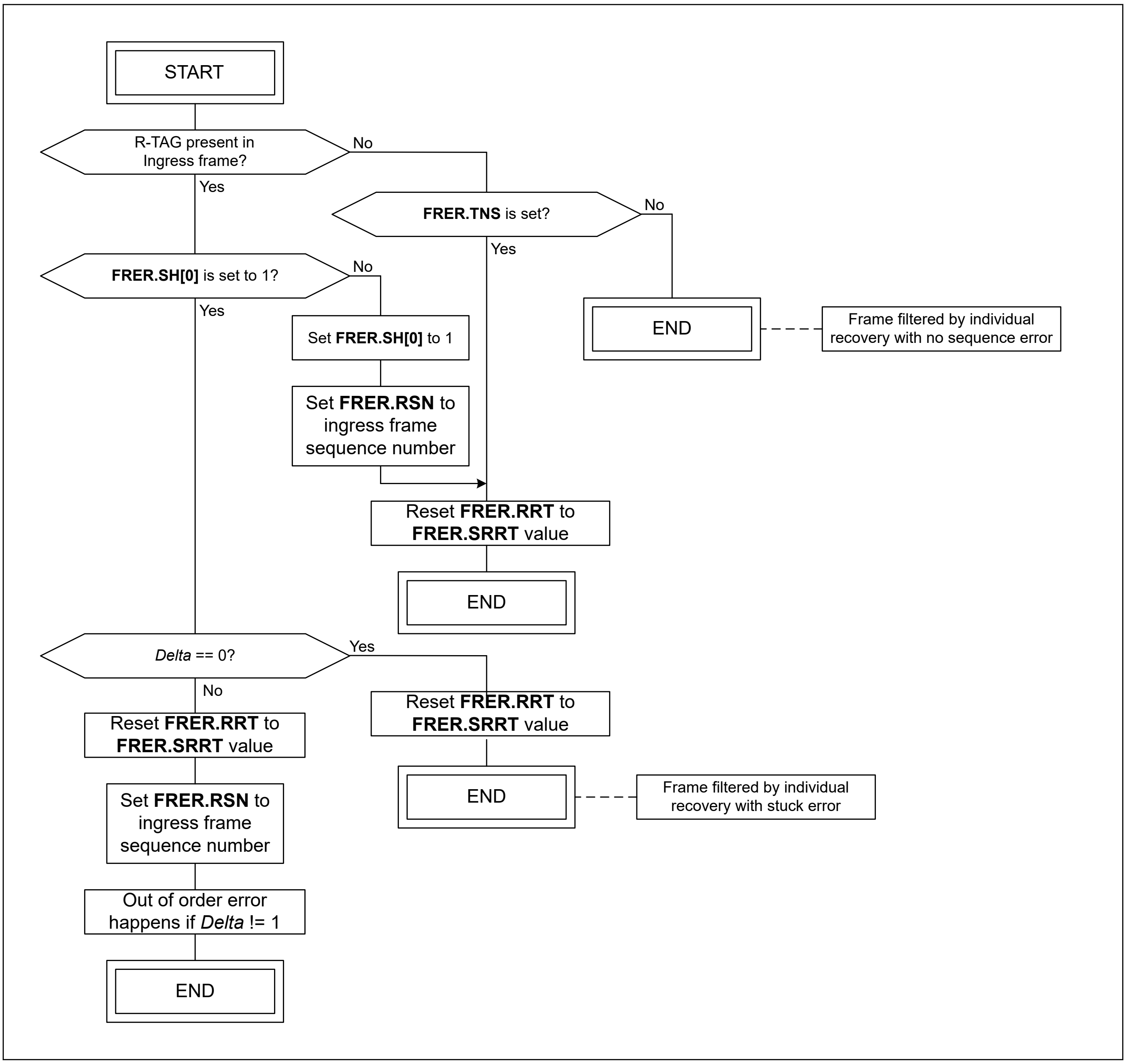


Figure 30.107 Individual recovery match algorithm

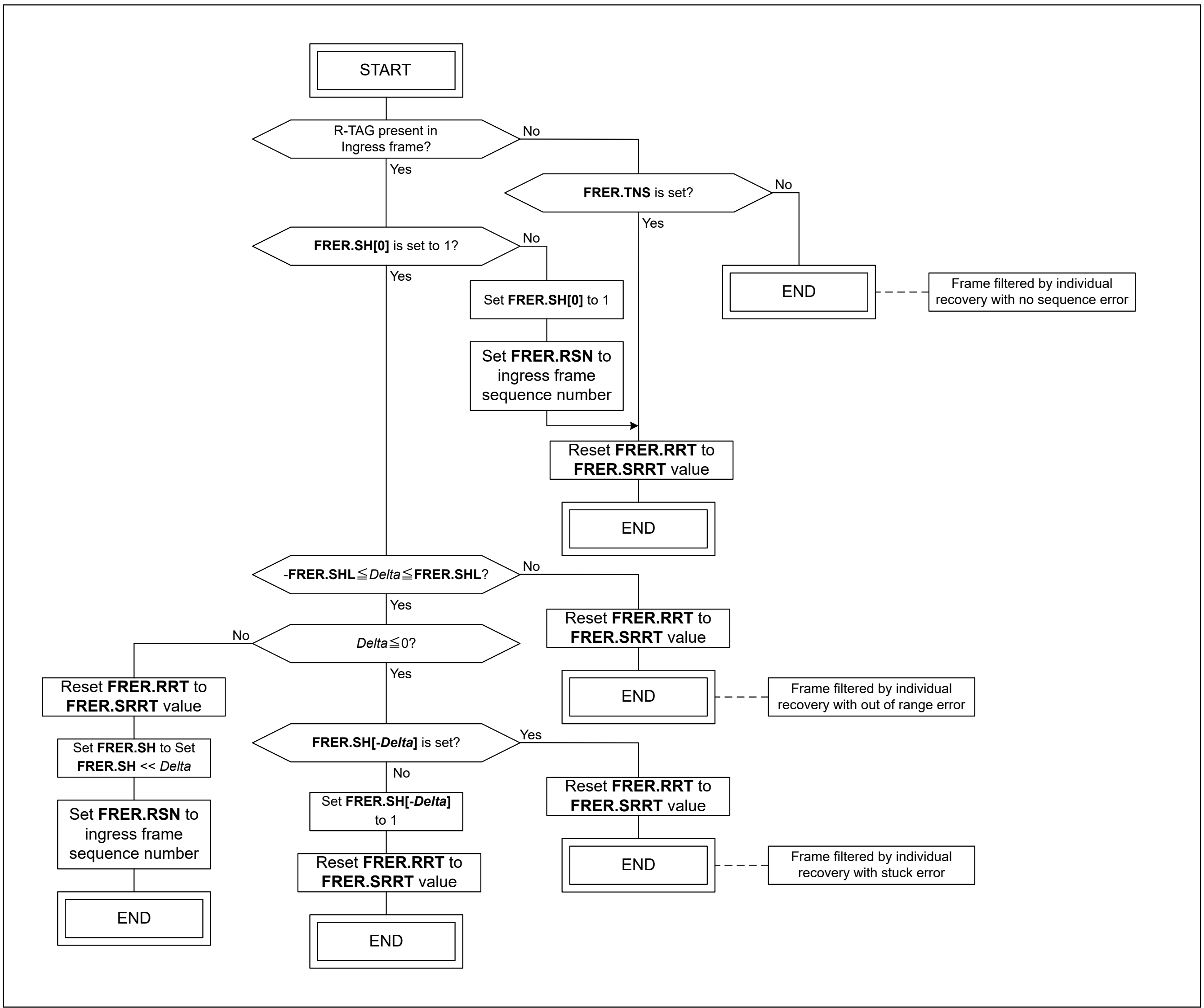


Figure 30.108 Individual recovery vector algorithm

Note: Even if the recovery length is smaller than 14, the sequence history saved in the FRER table will contain the history until 14. Vector recovery algorithm will ignore the useless bits.

(f) Sequence recovery algorithms

Sequence recovery match algorithm and vector algorithm are respectively described in Figure 30.109 and Figure 30.110. The variable *Delta* present in these figures is the distance from the ingress frame sequence number to FRER.RSN (This number should be considering overflow and should be contained in [-32767:32768] range).

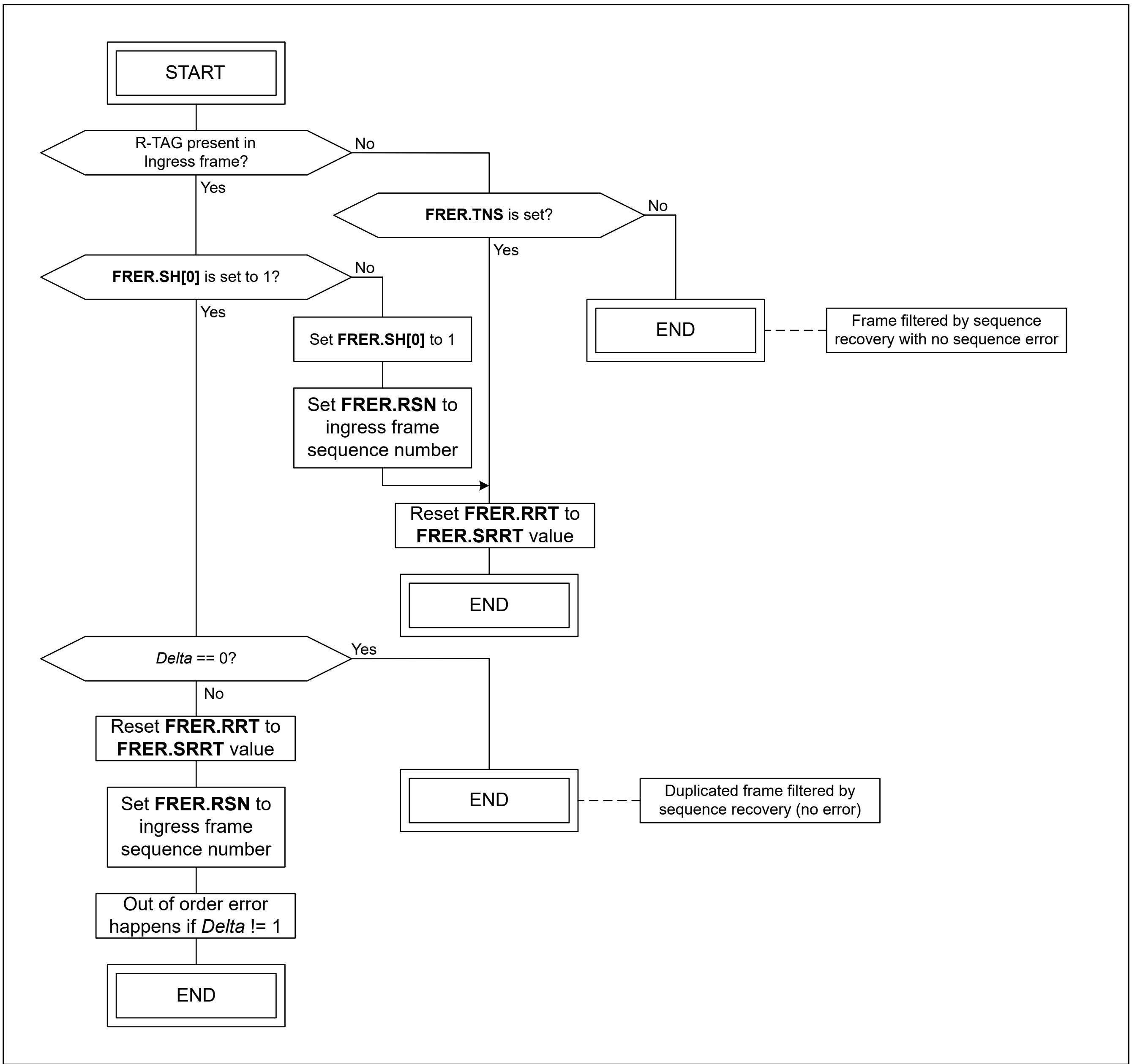


Figure 30.109 Sequence recovery match algorithm

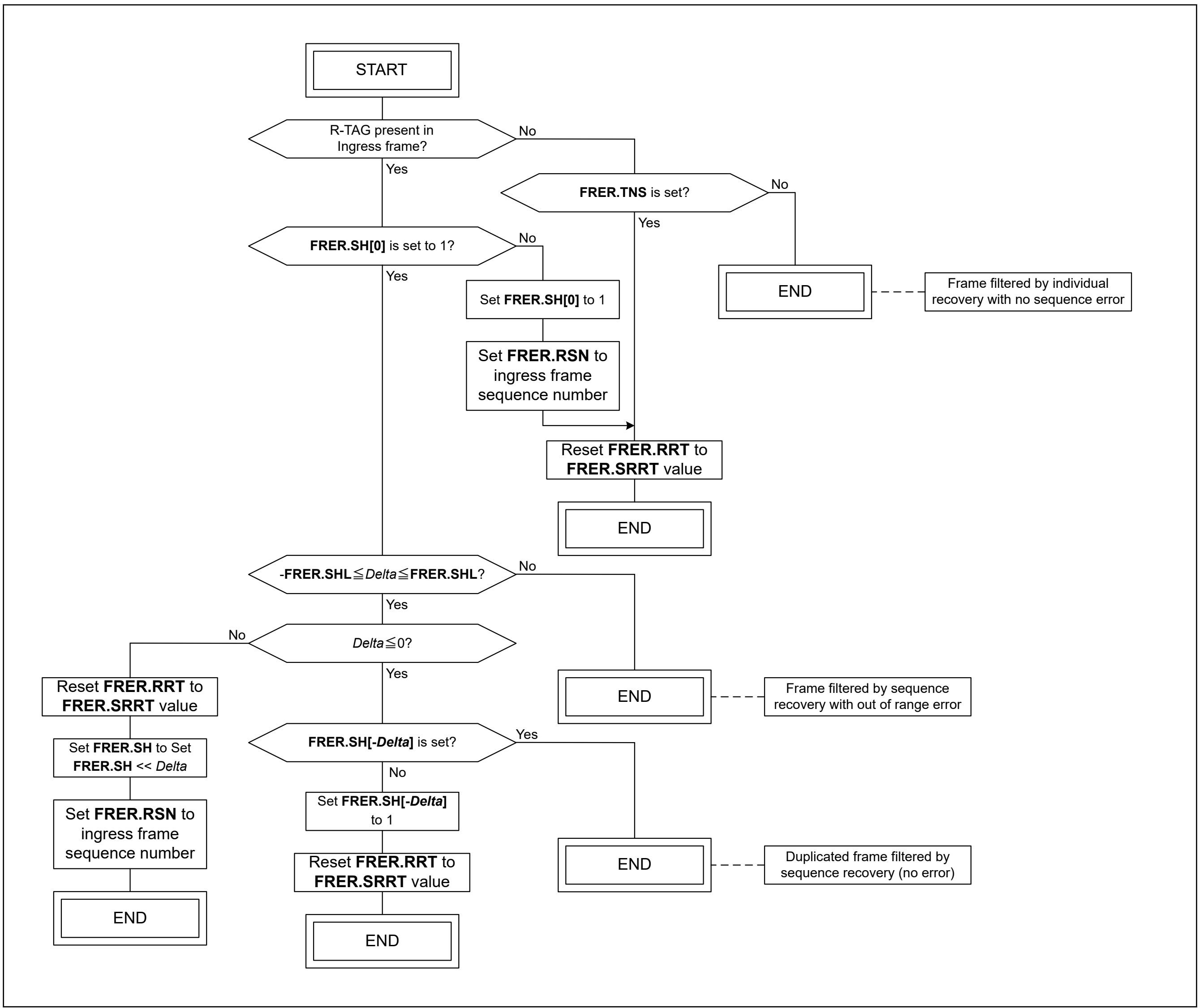


Figure 30.110 Sequence recovery vector algorithm

Note: Even if the recovery length is smaller than 14, the sequence history saved in the FRER table will contain the history until 14. Vector recovery algorithm will ignore the useless bits.

(3) Timeout

The timeout function aims at detecting when a recovery algorithm had no passing frame since a certain time and should be reset because it could be stuck. The timeout functional flow is described in for when timeout functions is enabled (FWFTOC.TOCE set to 1). This flow happens for all the FRER entries.

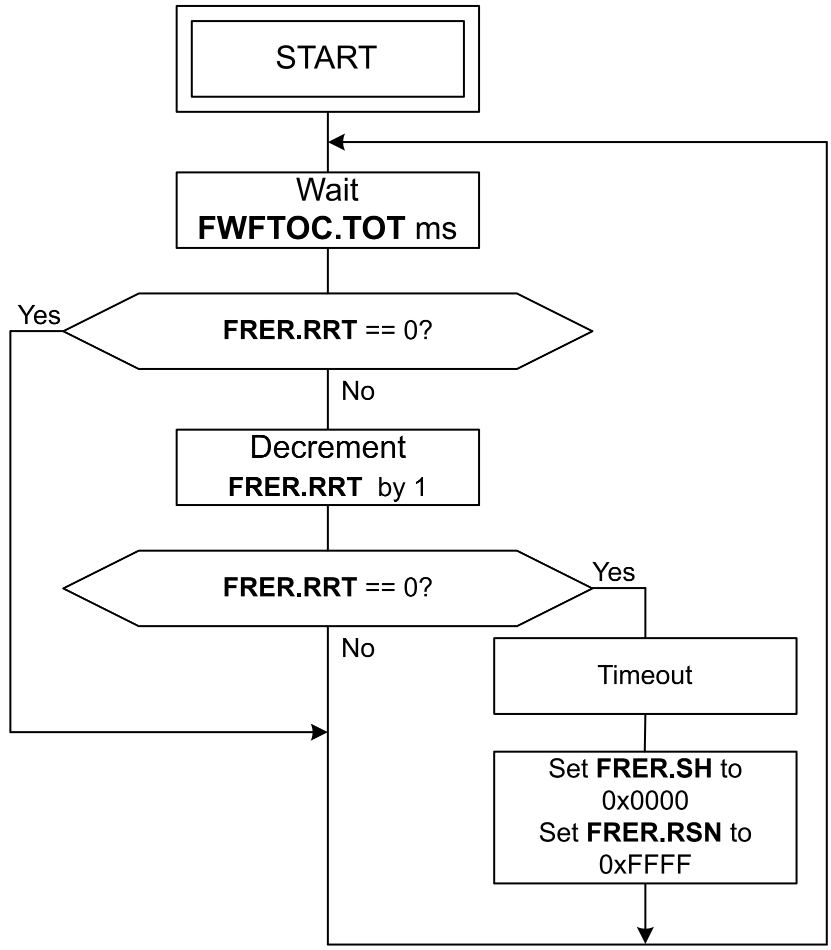


Figure 30.111 Timeout functional flow

30.5.4 Fabric (MFAB)

Fabric Bus (MFAB) is an internal bus used to exchange data between Ethernet Agent [ETHA], Ethernet Common Agent [COMA], Ethernet CPU Agent [GWCA], local RAM, TAG RAM, pointer RAM, and Ethernet Message Forwarding Engine [MFWD].

Table 30.82 lists Fabric features.

Table 30.82 Fabric feature list (1 of 2)

Function		Details
Overall function	Data transfer	• Function to transfer data and access to RAMs
Data provision		• Data for Fabric • Initial frame data + Intermediate data + Last frame data + Frame Descriptor • Data width: 128 + 48 + 9 bits

Table 30.82 Fabric feature list (2 of 2)

Function		Details
Control function	Time arbiter	● Arbitration between Time-critical agents [ETHA] for data bus. Arbitration scheme depends on each Time critical agents [ETHA] PHY link/Internal speed.
	Arbitration	● LRU for Time critical-agents [ETHA], Slow agents [GWCA] and Ethernet Common Agent [COMA] error read requests arbitration. ● LRU between Slow agents [GWCA] for read/write bus arbitration. ● Strict priority like arbitration between time arbiter, Error LRU, and slow LRU. ● Priority: time arbiter > slow LRU > Error LRU
	RAM access	● Function that writes and reads to RAM
	Transfer write request to Forwarding Engine	● Function to transfer write request sent from ETHA Agents [ETHA] or GWCAs [GWCA] to Forwarding Engine.

Figure 30.112 shows Fabric interface protocol.

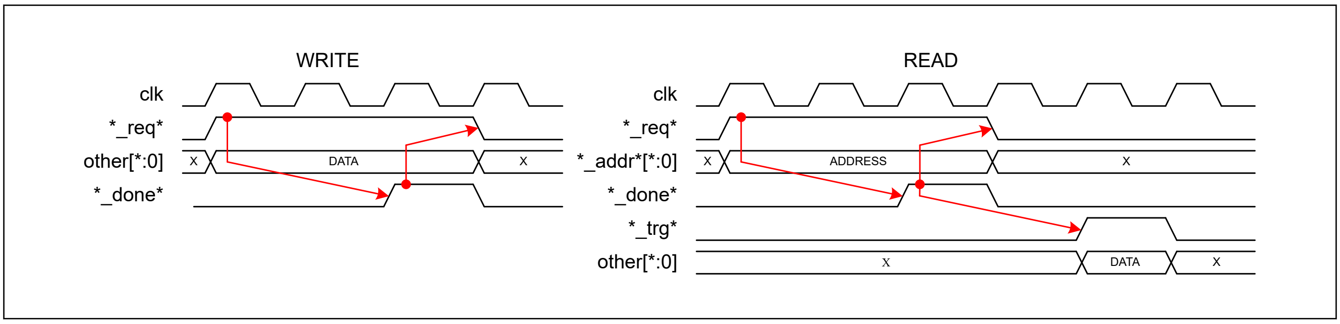


Figure 30.112 Fabric interface protocol

30.5.4.1 Fabric Time Arbiter

Time arbiter controls done signal of Ethernet Agent [ETHA] Write and Read buses depending on link speed.

The faster a link is the more the corresponding agent will be able access to the local RAM.

Agents Done Signal Distribution

Table 30.83 lists done signal distribution depending on PHY link speed possible combinations. Any disabled agent has a doneSignalDistributionValue equal to 0.

Table 30.83 Done signal output ratio by combination of PHY link speed

PHY link speeds of time-critical agents in OPERATION mode [ETHA]	doneSignalDistributionValue [cycle]
Only one Agent enabled	1
PHY link speed of all Agents is the same	1
100 Mbps : 1 Gbps	1 : 10

Done Signal Patterns

Figure 30.113 shows a done signal pattern example where time-critical port 0 has a 100 Mbps link and time-critical port 1 has a 1 Gbps link. When a request signals is not asserted at the timing when the corresponding done signal should have been asserted the done signals is not asserted and the handshake does not happen. In Figure 30.113 the request signals are always asserted to make the done pattern visible but, in a real system, request signals can be asserted anytime.

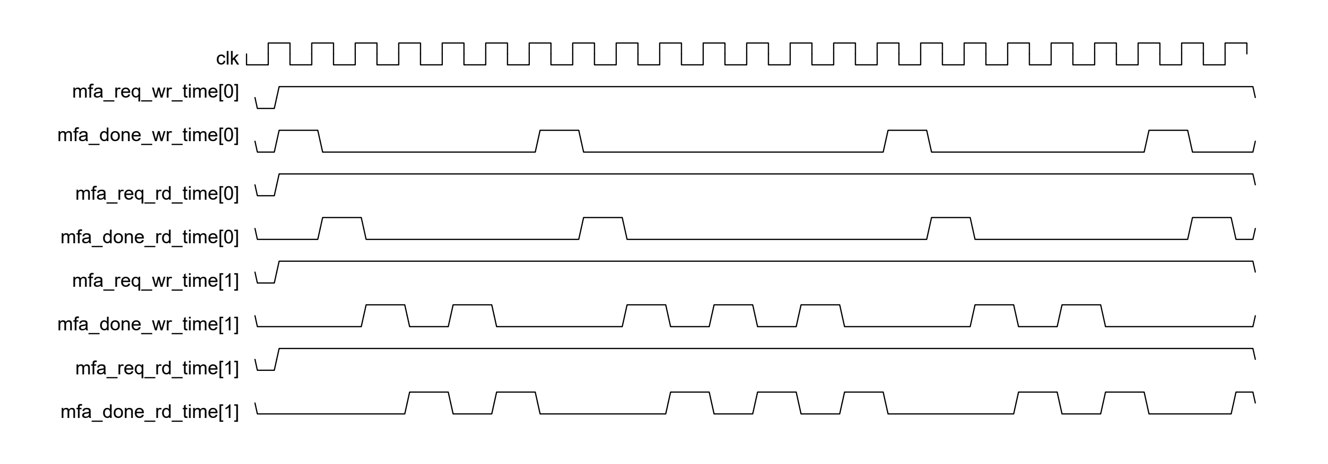


Figure 30.113 Done signal pattern example

The done signal pattern has inherent properties that are always respected:

- For a static configuration the done pattern is cyclic with a constant cycle time *doneCycleClk* described in equation (4).
- During a period of doneCycleClk clocks starting from anywhere, a time-critical agent receives for both read and write buses a number of done signals equal to its associated done signal distribution value (Table 30.83).

(4) doneCycleClk [cycle] = (Σ_2 doneSignalDistributionValue [cycle]) × 2

Where:

- Σ_2 doneSignalDistributionValue is the sum of all time-critical agents in OPERATION mode doneSignalDistributionValue (Table 30.83).

Restriction:

- HW: Done signals are only given to ethernet agents in OPERATION mode [ETHA].
- HW: If an unknown link speed is set for time-critical port i [ETHA] (mfa_xmii_speed[3 × (i + 1) - 1:3 × i] > 101b [RMAC]), Fabric will consider port i as disabled and port i will not be able to access the time arbiter.

Fabric Jitter

When an Ethernet Agent reads data using fabric time arbiter, some jitter is induced in the reading process by Fabric and it should be taken in account while using TAS [802.1Qbv] or PSFP [802.1Qci]. The fabric jitter can be calculated for an agent using equation (6).

(6) fabricJitter [ns] = (⌈ doneCycleClk[cycle] / doneSignalDistributionValue[cycle] ⌉ + (PortTimeOperation[number] - 1) × 2) × clk_period [ns]

Where:

- doneCycleClk is detailed in Done Signal Patterns.
- doneSignalDistributionValue is corresponding agent value detailed in Agents Done Signal Distribution.
- PortTimeOperation is the number of Ethernet Agents in OPERATION mode.
- clk_period is the clock clk period.

Restrictions:

- SW: The fabricJitter equation takes in account the number of Ethernet Agents [ETHA] in OPERATION mode and also doneCycleClk. In fact, enabling/disabling an agent or changing an agent PHY speed will indeed affect the fabric jitter value. However, reconfiguration TAS [802.1Qbv] or PSFP [802.1Qci] jitter in OPERATION is not a recommended option, so the maximum number of Ethernet Agents in OPERATION mode and their corresponding maximum PHY speed should be determined to calculate the worst fabricJitter per agent and it should be kept to the same value all the time.

Fabric Minimum Latency

When an Ethernet Agent reads data using fabric time arbiter, some latency is induced in the reading process by Fabric and it should be taken in account while using TAS [802.1Qbv] or PSFP [802.1Qci]. The fabric minimum latency can be calculated for an agent using equation (7).

(7) fabricLatency [ns] = (⌊ doneCycleClk [cycle] / doneSignalDistributionValue [cycle] ⌋ - 1) × clk_period [ns]

Where:

- doneCycleClk is detailed in Done Signal Patterns.
- doneSignalDistributionValue is corresponding agent value detailed in Agents Done Signal Distribution.
- clk_period is the clock clk period.

Restrictions:

- SW: The fabricLatency equation takes in account doneCycleClk. In fact, enabling/disabling an agent or changing an agent PHY speed will indeed affect the fabric latency value. However, reconfiguration TAS [802.1Qbv] or PSFP [802.1Qci] latency in OPERATION is not a recommended option, so if ethernet agent are disabled or PHY speed is changed during OPERATION, it is recommended to set the fabricLatency to 2 × clk_period [ns] and add the fabricLatency calculated value in the worst case to fabricJitter.
- SW: When preemption is enabled for a port, fabricLatency should be set to 0ns and fabricLatency calculated value when all agents are enabled should be added to fabricJitter.

30.5.4.2 Source Port Generation

Fabric gives automatically a source port number to each Agent depending on the bus they are accessing to. Table 30.84 shows the relation between the Bus accessed by an agent and its source port number.

Table 30.84 Source port number assignment

Bus from which agent accesses the fabric	Source port number
ETHA0	0
ETHA1	1
GWCA0	2

31. Ethernet Common Agent (COMA)

31.1 Overview

Ethernet Common Agent (COMA) is an agent of ESWM which aims at gathering common functionalities for other agents. COMA handles the APB to SFR bus conversion, the buffer pointer release for rejected frames and the pointer handling for local RAM. Software is required for the configuration of Ethernet Common Agent.

Table 31.1 lists the COMA specifications and Figure 31.1 shows the block diagram.

Table 31.1 COMA specifications

Item	Function	Description
Overall function	Pointer management	• Manage the local RAM pointers and their distribution
	APB to SFR	• Convert APB bus to SFR bus • Distribute SFR access among other switch IPs
	Descriptor reject	• Reject the unused descriptors by agents for transmission
Data provision	Fabric error bus	• Pointer received from fabric through error bus
	Descriptor bus	• Descriptor received from MFWD for rejection
Control function	RAM access	• Function that writes and reads to RAM

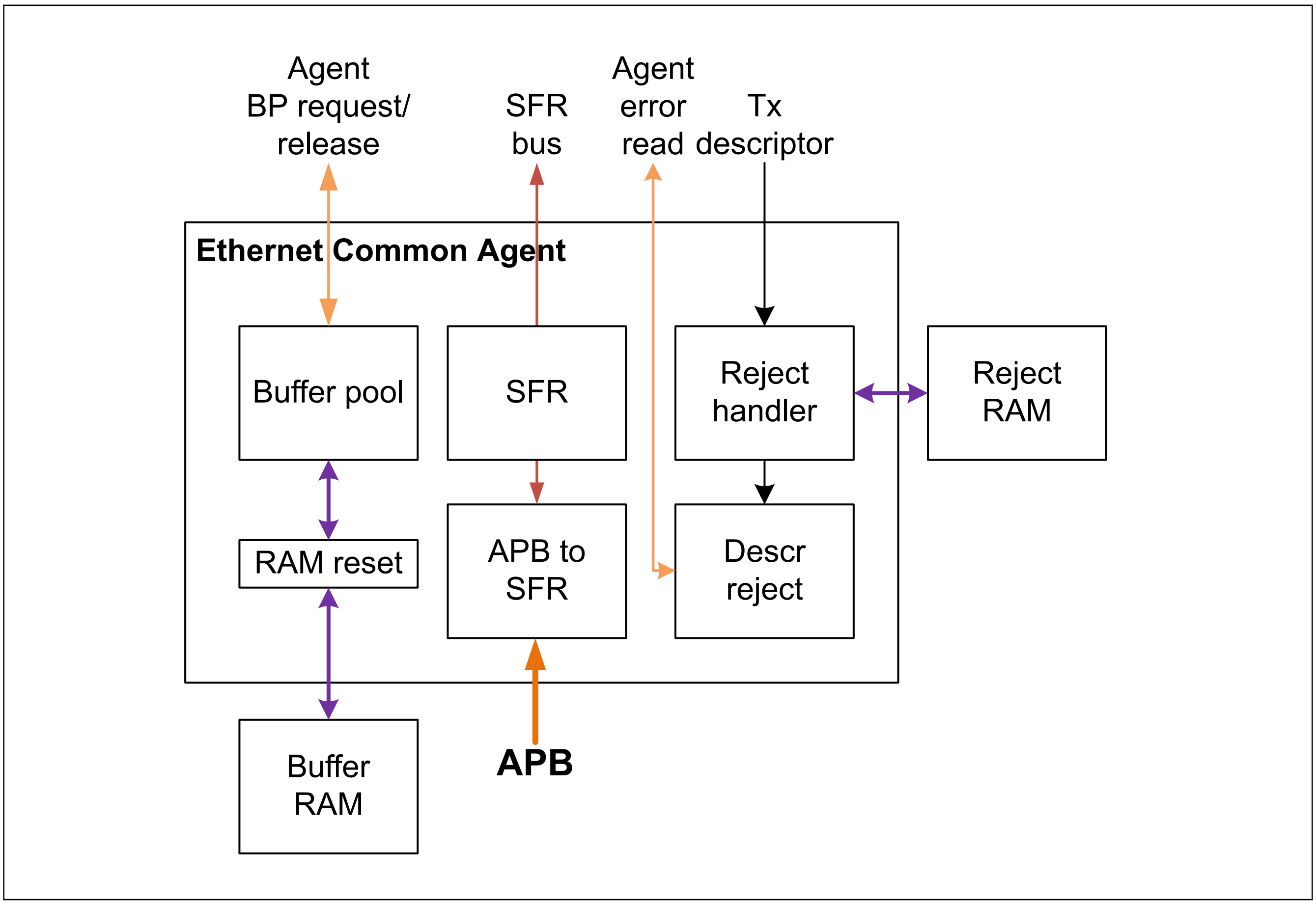


Figure 31.1 COMA block diagram

Table 31.2 COMA functional blocks (1 of 2)

Block name	Function
APB to SFR	Convert APB bus to SFR bus for every IP in ESWM
SFR	Ethernet Common Agent SFR

Table 31.2 COMA functional blocks (2 of 2)

Block name	Function
Reject handler	Receive descriptors to be rejected from forwarding engine and save them in reject RAM
Descriptor reject	Reject the descriptors
Free buffer pool	Handles the free buffer pointers
RAM reset	Handle RAM reset

31.2 COMA Register List

Table 31.3 COMA registers

Offset address	Register symbol	Register name
0x0000	RIPV	IP Version Register
0x0004	RRC	Reset Configuration Register
0x0008	RCEC	Clock Enable Configuration Register
0x000C	RCDC	Clock Disable Configuration Register
0x0020 + 0x04 × i	CABPIBWMCi	Buffer Pool IPV Based Watermark Configuration Register i (i = 0 to 7)
0x0040	CABPWMLC	Buffer Pool Watermark Level Configuration Register
0x0050 + 0x04 × i	CABPPFLCi	Buffer Pointer Pause Frame Level i Configuration Register (i = 0, 1)
0x0060 + 0x04 × i	CABPPWMLCi	Port i Buffer Pool Watermark Level Configuration Register i (i = 0 to 2)
0x00A0 + 0x04 × (i × 2 + j)	CABPPPFLCij	Port i Buffer Pointer Pause Frame Level j Configuration Register (i = 0 to 2) (j = 0, 1)
0x0100 + 0x04 × i	CABPULCi	Port i Buffer Pointer Utilization Level Configuration Register i (i = 0 to 2)
0x0140	CABPIRM	Buffer Pool Initialization Register Monitoring Register
0x0144	CABPPCM	Buffer Pool Pointer Count Monitoring Register
0x0148 E: 0x014C	CABPLCM	Buffer Pool Pointer Least Count Monitoring Register
0x0180 + 0x04 × i	CABPCPMi	Port i Buffer Pointer Count Monitoring Register (i = 0 to 2)
0x0200 + 0x04 × i E: 0x0280 + 0x04 × i	CABPMCPMi	Port i Buffer Pointer Maximum Count Monitoring Register (i = 0 to 2)
0x0300	CARDNM	Rejected Descriptor Number Monitoring Register
0x0304 E: 0x0308	CARDMNM	Rejected Descriptor Maximum Number Monitoring Register
0x0310 E: 0x0314	CARDCN	Rejected Descriptor Counter Register
0x0400	CAEIS0	Error Interrupt Status Register 0
0x0404	CAEIE0	Error Interrupt Enable Register 0
0x0408	CAEID0	Error Interrupt Disable Register 0
0x0410	CAEIS1	Error Interrupt Status Register 1
0x0414	CAEIE1	Error Interrupt Enable Register 1
0x0418	CAEID1	Error Interrupt Disable Register 1
0x0440	CAMIS0	Monitoring Interrupt Status Register 0
0x0444	CAMIE0	Monitoring Interrupt Enable Register 0
0x0448	CAMID0	Monitoring Interrupt Disable Register 0
0x0450	CAMIS1	Monitoring Interrupt Status Register 1
0x0454	CAMIE1	Monitoring Interrupt Enable Register 1
0x0458	CAMID1	Monitoring Interrupt Disable Register 1

Note: Base address: COMA = 0x403C_90000, COMA_NS = 0x503C_90000

Note: The address preceded by E: corresponds to an emulation address which allows a register to be read without modifying its content.

31.3 COMA Register Descriptions

31.3.1 ESWM Registers

31.3.1.1 RIPV : IP Version Register

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	CAIPV[3:0]				FBIPV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1

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

Bit	Symbol	Function	R/W
3:0	TIPV[3:0]	Top Module IP Version Number	R
7:4	GWIPV[3:0]	Gateway CPU Agent IP Version Number	R
11:8	FWIPV[3:0]	Forwarding Engine IP Version Number	R
15:12	EAIPV[3:0]	Ethernet Agent IP Version Number	R
19:16	FBIPV[3:0]	Fabric Bus IP Version Number	R
23:20	CAIPV[3:0]	Ethernet Common Agent IP Version Number	R
31:24	—	These bits are read as 0.	R

31.3.1.2 RRC : Reset Configuration Register

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0004

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

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

Bit	Symbol	Function	R/W
0	RR	Software Reset 0: ESWM not reset 1: ESWM reset	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

RR bit (Software Reset)

- When set, this bit resets all the ESWM except the APB interface

- The synchronous reset RCC.RR is power-consuming, so it is only internally asserted for 7 clock cycles
- The synchronous reset is also internally asserted for 7 clock cycles after a hardware reset release.

[Caution]

- Because the synchronous reset RCC.RR is power-consuming, it should always be deasserted right after assertion to avoid power consumption.

31.3.1.3 RCEC : Clock Enable Configuration Register

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0008

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	ACE2 to ACE0		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	ACE2 to ACE0	Agent i Clock Enable (i = 0 to 2) Bit i enables agent i clock. 0: Agent i clock disabled 1: Agent i clock enabled	R/W*1
15:3	—	These bits are read as 0. The write value should be 0.	R/W
16	RCE	Clock Enable 0: ESWM clock disabled 1: ESWM clock enabled	R/W*1
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: ESWM has three system clocks:

- Peripheral clock: Can be stopped only by the upper layer (MCU)
- MFWD clock: Controlled by RCE
- Agent clock: Controlled by ACEi.

Note 1. Read value differs from written value.

The COMA registers are controlled by peripheral clocks, which can only be stopped by the upper layer (MCU). Therefore, COMA registers always have read and write access.

When peripheral clock is running and agent clock is stopped (ACEi is 0):

- GWCA and ETHA have APB read access but do not have APB write access.

When peripheral clock is running and MFWD clock is stopped (RCE is 0):

- MFWD has APB read access but does not have APB write access
- Agent clocks are stopped.

ACEi bits (Agent i Clock Enable (i = 0 to 2))

- Bit i enables agent i clock. For agent mapping details, see the Fabric specifications [MFAB].
- When RRC.RR register is set, all agent i clock is internally enabled to allow software reset even if RCC.ACEi is not set.
- When RCC.RCE register is not set, agent i clock is internally disabled even if RCC.ACEi is set.

[Setting condition]

- SW: Writing 1 to this bit.

[Clearing condition]

- SW: Writing 1 to RCDC.ACDi register.

RCE bit (Clock Enable)

The RCE bit enables the switch clocks. When RRC.RR register is set, ESWM clock is internally enabled to allow software reset even if RCC.RCE is not set.

[Setting condition]

- SW: Writing 1 to this bit.

[Clearing condition]

- SW: Writing 1 to RCDC.RCD register.

31.3.1.4 RCDC : Clock Disable Configuration Register

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x000C

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	ACD2 to ACD0		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	ACD2 to ACD0	Agent i Clock Disable (i = 0 to 2) 0: No effect 1: Clears bit i in RCEC.ACEi register	W*2
15:3	—	These bits are read as 0.	R
16	RCD	Clock Disable*1 Writing 1 to this bit clears RCEC.RCE register.	W*2
31:17	—	These bits are read as 0.	R

Note 1. SW: Only write to this bit when all agents are disabled and when CABPPCM.RPC is equal to 512.
Note 2. Read value differs from written value.

31.3.2 Ethernet Common Agent Function Registers

31.3.2.1 CABPIBWMCi : Buffer Pool IPV Based Watermark Configuration Register i (i = 0 to 7)

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0020 + 0x04 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	IBSWMPN[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
9:0	IBUWMPN[9:0]	IPV Based Unsecure Watermark Pointer Number*1	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	IBSWMPN[9:0]	IPV Based Secure Watermark Pointer Number*2	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Do not set this register to a value greater than the following:

- SW: 512
- SW: CABPIBWMC{i + 1}.IBUWMPN{i + 1}.

Note 2. Do not set this register to a value greater than the following:

- SW: 512
- SW: CABPIBWMC{i + 1}.IBSWMPN{i + 1}.

IBUWMPN[9:0] bits (IPV Based Unsecure Watermark Pointer Number)

- For i equal to 7, setting this register to 512 disables the IPV based watermark function for IPV 7
- For i other than 7, setting this register to CABPIBWMC{i + 1}.IBUWMPN{i + 1} disables the IPV-based watermark function for IPV
- See section 31.5.1.1. IPV-based Watermark for details.

IBSWMPN[9:0] bits (IPV Based Secure Watermark Pointer Number)

- For i equal to 7, setting this register to 512 disables the IPV based watermark function for IPV 7
- For i other than 7, setting this register to CABPIBWMC{i + 1}.IBSWMPN{i + 1} disables the IPV-based watermark function for IPV i
- See section 31.5.1.1. IPV-based Watermark for details.

31.3.2.2 CABPWMLC : Buffer Pool Watermark Level Configuration Register

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0040

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	WMCL[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
9:0	WMFL[9:0]	Watermark Flush Level*1 • Setting this register to 512 disables the watermark flush level function • See section 31.5.1.2. Global Level-based Watermark for details	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	WMCL[9:0]	Watermark Critical Level*2 • Setting this register to 512 disables the watermark critical level function • See section 31.5.1.2. Global Level-based Watermark for details	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: Do not set this register to a value greater than 512.
Note 2. SW: Always set this register to a value less than or equal to CABPWMLC.WMFL but not greater than 512.

31.3.2.3 CABPPFLCi : Buffer Pointer Pause Frame Level i Configuration Register

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0050 + 0x0004 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	PAL[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
9:0	PDL[9:0]	Pause De-Assertion Level*1 • See section 31.5.2.1. Global Pause Function for details.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	PAL[9:0]	Pause Assertion Level*2 • Setting this register to 512 disables pause frames • See section 31.5.2.1. Global Pause Function for details.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. • SW: Always set this register to a value less than or equal to CABPPFLCi.PAL.
• SW: Do not set this register to a value greater than 512 or less than 24.
Note 2. SW: Do not set this register to a value greater than 512 or less than 24.

PDL[9:0] bits (Pause De-Assertion Level)

[Precautions]

If CABPPFLCi.PAL value is too close to CABPPFLCi.PDL value, the corresponding PAUSE signals change state rapidly. Additionally, if no automatic pause frames are activated [RMAC], this can result in a succession of pause frames on RMAC transmission which can decrease the link performance.

31.3.2.4 CABPPWMLCi : Port i Buffer Pool Watermark Level Configuration Register (i = 0 to 2)

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0060 + 0x0004 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	PWMCL[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
9:0	PWMFL[9:0]	Watermark Flush Level*1 ● Setting this register to 512 disables the watermark flush level function ● See section 31.5.1.3. Per-port Level-based Watermark i (i = 0 to 3) for details	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	PWMCL[9:0]	Watermark Critical Level*2 ● Setting this register to 512 disables the watermark critical level function ● See section 31.5.1.3. Per-port Level-based Watermark i (i = 0 to 3) for details	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: Do not set this register to a value greater than 512.
Note 2. SW: Always set this register to a value less than or equal to CABPPWMLCi.PWMFL but not greater than 512.

31.3.2.5 CABPPPFLCij : Port i Buffer Pointer Pause Frame Level j Configuration Register (i = 0 to 2) (j = 0, 1)

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x00A0 + 0x04 × (i × 2 + j)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	PPAL[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
9:0	PPDL[9:0]	Pause De-Assertion Level*1 See section 31.5.2.2. Per-port Pause Function i (i = 0 to 3) for details. If CABPPPFLCij.PPAL value is too close to CABPPPFLCij.PPDLij value, the corresponding PAUSE signals change state rapidly. Additionally, if no automatic pause frames are activated [RMAC], this can result in a succession of pause frames on RMAC transmission which can decrease the link performance.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	PPAL[9:0]	Pause Assertion Level*2 Setting this register to 512 disables pause frames. See section 31.5.2.2. Per-port Pause Function i (i = 0 to 3) for details.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Restrictions:

- SW: Always set this register to a value less than or equal to CABPPPFLCij.PPAL.
- SW: Do not set this register to a value greater than 512 or less than 6.

Note 2. SW: Do not set this register to a value greater than 512 or less than 6.

31.3.2.6 CABPULCi : Port i Buffer Pointer Utilization Level Configuration Register (i = 0 to 2)

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0100 + 0x0004 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	MNNPN[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
9:0	MXNPN[9:0]	Maximum Number of Pointer*1 Maximum number of pointers that can be used by port i.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	MNNPN[9:0]	Minimum Number of Pointer*2 Minimum number of pointers that can be used by port i.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note: If CABPULC[i].MNNPN[i] of port[i] is changed (reconfigured) from its default value, then it restricts the reception maximum frame size of ports other than port[i]. The restricted reception maximum frame size of ports other than port[i] in such case is given in the following expression.
Restricted reception maximum frame size = ((512 – CABPULC[i].MNNPN[i])/(3)) × 128 bytes. This is because ports other than port[i] must share the remaining pointers (512 – CABPULC[i].MNNPN[i]).
If this restriction is not followed, then the consumed buffer pointers are not released by other ports and the frame is stuck inside the switch. Because the total buffer pointers are insufficient for the port in the middle of reception, to resolve this, reconfigure CABPULC[i].MNNPN[i] value to a smaller value.

Note 1. SW: Set the maximum number of pointers to a value large enough to save two frames with a size equal to RMAC reception maximum frame size (set in RMAC).

Note 2.
- SW: The sum of all CABPULCi.MNNPN should be smaller or equal to 512.
- SW: This register should always be smaller or equal to CABPULCi.MXNPN.

MXNPN[9:0] bits (Maximum Number of Pointer)

- If register CABPCPMi.RPCP is equal to this register value, port i no longer receives pointer
- See section 31.5.3. Per-port Memory Allocation Function for details.

MNNPN[9:0] bits (Minimum Number of Pointer)

- If SUM (CABPULCi.MNNPN) >= CABPPCM.RPC + SUM (CABPCPMi.RPCP) only ports for which CABPCPMi.RPCP < CABPULCi.MNNPN can receive pointers
- See section 31.5.3. Per-port Memory Allocation Function for details.

31.3.2.7 CABPIRM : Buffer Pool Initialization Register Monitoring Register

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0140

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

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

Bit	Symbol	Function	R/W
0	BPIOG	Buffer Pool Initialization Ongoing*1	R/W*2
1	BPR	Buffer Pool Ready	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. HW: CPU cannot write 1 to this bit when CABPIRM.BPR is set.
Note 2. Read value differs from written value.

BPIOG bit (Buffer Pool Initialization Ongoing)

[Setting condition]

- SW: Writing 1 to this register.

[Clearing condition]

- HW: When buffer pool initialization is complete.

BPR bit (Buffer Pool Ready)

[Setting conditions]

- When CABPIRM.BPIOG is cleared
- HW: This bit is set at clk_period [ns] × 512 time from the start of buffer pool initialization.

31.3.2.8 CABPPCM : Buffer Pool Pointer Count Monitoring Register

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0144

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	TPC[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
9:0	RPC[9:0]	Remaining Pointer Count	R
15:10	—	These bits are read as 0.	R
25:16	TPC[9:0]	Total Pointer Count	R
31:26	—	These bits are read as 0.	R

RPC[9:0] bits (Remaining Pointer Count)

[Increment condition]

- HW: Incremented by 1 when a pointer is released.

[Decrement condition]

- HW: Decremented by 1 when a pointer is given to an agent.

TPC[9:0] bits (Total Pointer Count)

- Read 512, fixed to 512
- Reading this register gives the number of pointers contained in the buffer pool. The switch local data RAM size is equal to 128 × CABPPCM.TPC bytes.

31.3.2.9 CABPLCM : Buffer Pool Pointer Least Count Monitoring Register

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0148

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

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

Bit	Symbol	Function	R/W
12:0	LRC[12:0]	Least Remaining Pointer Count This register remembers the smaller value that is taken by the remaining counter pointer. [Update condition] ● HW: This register is updated to CABPPCM.RPC when CABPPCM.RPC < CABPLCM.LRC. [Clear condition] ● Reading this register clears it to CABPPCM.RPC value.	R
31:13	—	These bits are read as 0.	R

31.3.2.10 CABPCPMi : Port i Buffer Pointer Count Monitoring Register (i = 0 to 2)

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0180 + 0x0004 × i

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

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

Bit	Symbol	Function	R/W
12:0	RPCP[12:0]	Received Pointer Count	R
31:13	—	These bits are read as 0.	R

RPCP[12:0] bits (Received Pointer Count)

This register shows the number of pointers that are used by agent i to store data.

[Increment condition]

- HW: Incremented by 1 when a pointer is given to port I.

[Decrement condition]

- HW: Decremented by 1 when a pointer given to port i is being released.

31.3.2.11 CABPMCPMi : Port i Buffer Pointer Maximum Count Monitoring Register (i = 0 to 2)

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0200 + 0x0004 × i

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

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

Bit	Symbol	Function	R/W
12:0	RPMCP[12:0]	Received Pointer Maximum Count*1	R
31:13	—	These bits are read as 0.	R

Note 1. May exceed several upper and lower limit settings.

RPMCP[12:0] bits (Received Pointer Maximum Count)

This register remembers the biggest value that is taken by CABPMCPMi.RPCP since the last read access.

[Update condition]

- HW: This register is updated to CABPMCPMi.RPCP when CABPMCPMi.RPCP > CABPMCPMi.RPMCP.

[Clear condition]

- Reading this register clears it to CABPMCPMi.RPCP value.

31.3.2.12 CARDNM : Rejected Descriptor Number Monitoring Register

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0300

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

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

Bit	Symbol	Function	R/W
12:0	RDNRR[12:0]	Rejected Descriptor Number in Reject RAM	R
31:13	—	These bits are read as 0.	R

RDNRR[12:0] bits (Rejected Descriptor Number in Reject RAM)

This register shows the number of descriptors currently contained in the reject RAM.

[Increment condition]

- HW: Incremented by 1 when a descriptor is received from the forwarding engine.

[Decrement condition]

- HW: Decremented by 1 when a descriptor is read from the reject RAM.

31.3.2.13 CARDMNM : Rejected Descriptor Maximum Number Monitoring Register

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0304

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

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

Bit	Symbol	Function	R/W
12:0	RDMNRR[12:0]	Rejected Descriptor Maximum Number in Reject RAM	R
31:13	—	These bits are read as 0.	R

RDMNRR[12:0] bits (Rejected Descriptor Maximum Number in Reject RAM)

This register remembers the biggest value that is taken by CARDNM.RDNRR since the last read access.

[Update condition]

- Set to CARDNM.RDNRR when CARDNM.RDNRR > CARDMNM.RDMNRR.

[Clearing condition]

- Reading this register clears it to CARDNM.RDNRR value.

31.3.3 Ethernet Common Agent Counter Registers

This section describes Ethernet Common Agent Counter registers.

31.3.3.1 CARDCN : Rejected Descriptor Counter Register

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0310

Bit position:	31–0
Bit field:	RDN[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	RDN[31:0]	Rejected Descriptor Number	R

RDN[31:0] bits (Rejected Descriptor Number)

This register counts the number of descriptors that is rejected since the last read.

[Clearing condition]

- SW: Reading this register clears it.

[Increment condition]

- HW: Incremented by 1 when a rejected descriptor is received from MFWD and if this register has a value other than 0xFFFF.

31.3.4 Ethernet Common Agent Interrupt Registers

This section describes Ethernet Common Agent Interrupt registers.

31.3.4.1 CAEIS0 : Error Interrupt Status Register 0

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0400

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

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

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	BPOPS	Buffer Pool Out of Pointer Status Flag	R/W*1
9	WMCLOS	Watermark Critical Level Overtook Status Flag	R/W*1
10	WMFLOS	Watermark Flush Level Overtook Status Flag	R/W*1
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

BPOPS bit (Buffer Pool Out of Pointer Status Flag)

[Setting condition]

- HW: When CABPPCM.RPC becomes null.

[Clearing condition]

- SW: Writing 1 to this bit.

[Error recovery]

- SW: When there is no error recovery, the switch can overflow [RMAC]. When using Cut through, use the watermark function so that this status does not occur. If this error occurs (it may get hung-up in the transmit state [ETHA and RMAC]), initialize the state with the emergency reset flow [TOP].

WMCLOS bit (Watermark Critical Level Overtook Status Flag)

[Setting condition]

- HW: When watermark critical level WM.CRITICAL is asserted to all 1s because of global level-based watermark (see Figure 31.12).

[Clearing condition]

- SW: Writing 1 to this bit.

[Error recovery]

- SW: When there is no recovery, data can be lost [FWD].

WMFLOS bit (Watermark Flush Level Overtook Status Flag)

[Setting condition]

- HW: When watermark flush level WM.FLUSH is asserted to all 1s because of global level-based watermark (see Figure 31.12).

[Clearing condition]

- SW: Writing 1 to this bit.

[Error recovery]

- SW: When there is no recovery, data can be lost [FWD].

31.3.4.2 CAEIE0 : Error Interrupt Enable Register 0

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0404

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

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

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	BPOPE	Buffer Pool Out of Pointer Enable	R/W*1
9	WMCLOE	Watermark Critical Level Overtook Enable	R/W*1
10	WMFLOE	Watermark Flush Level Overtook Enable	R/W*1
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

BPOPE bit (Buffer Pool Out of Pointer Enable)

[Setting condition]

- Writing 1 to this bit.

[Clearing condition]

- Writing 1 to CAEID0.BPOPD register.

WMCLOE bit (Watermark Critical Level Overtook Enable)

[Setting condition]

- Writing 1 to this bit.

[Clearing condition]

- Writing 1 to CAEID0.WMCLOD register.

WMFLOE bit (Watermark Flush Level Overtook Enable)

[Setting condition]

- Writing 1 to this bit.

[Clearing condition]

- Writing 1 to CAEID0.WMFLOD register.

31.3.4.3 CAEID0 : Error Interrupt Disable Register 0

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0408

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

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

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0.	R
8	BPOPD	Buffer Pool Out of Pointer Disable Writing 1 to this bit clears CAEIE0.BPOPE register.	R
9	WMCLOD	Watermark Critical Level Overtook Disable Writing 1 to this bit clears CAEIE0.WMCLOE register.	R
10	WMFLOD	Watermark Flush Level Overtook Disable Writing 1 to this bit clears CAEIE0.WMFLOE register.	R
31:11	—	These bits are read as 0.	R

31.3.4.4 CAEIS1 : Error Interrupt Status Register 1

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0410

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	PWMFLOS2 to PWMFLOS0		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	PWMCLOS2 to PWMCLOS0		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	PWMCLOS2 to PWMCLOS0	Port n Watermark Critical Level Overtook Status Flag (n = 0 to 2)	R/W*1
15:3	—	These bits are read as 0.	R/W
18:16	PWMFLOS2 to PWMFLOS0	Port n Watermark Flush Level Overtook Status Flag (n = 0 to 2)	R/W*1
31:19	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

PWMCLOSn bits (Port n Watermark Critical Level Overtook Status Flag (n = 0 to 2))

[Setting condition]

- HW: For bit i, when watermark critical level bit i WM.CRITICAL[i] is asserted to 1 because of per-port level-based watermark (see Figure 31.13).

[Clearing condition]

- SW: Writing 1 to one of these bits.

[Error recovery]

- When there is no error recovery, data can be lost [FWD].

PWMFLOSn bits (Port n Watermark Flush Level Overtook Status Flag (n = 0 to 2))

[Setting condition]

- HW: For bit i, when watermark flush level bit i WM.FLUSH[i] is asserted to 1 because of per-port level-based watermark (see Figure 31.13).

[Clearing condition]

- SW: Writing 1 to one of these bits.

[Error recovery]

- When there is no error recovery, data can be lost [FWD].

31.3.4.5 CAEIE1 : Error Interrupt Enable Register 1

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0414

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	PWMFLOE2 to PWMFLOE0		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	PWMCLOE2 to PWMCLOE0		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	PWMCLOE2 to PWMCLOE0	Port n Watermark Critical Level Overtook Enable (n = 0 to 2)	R/W*1
15:3	—	These bits are read as 0. The write value should be 0.	R/W
18:16	PWMFLOE2 to PWMFLOE0	Port n Watermark Flush Level Overtook Enable (n = 0 to 2)	R/W*1
31:19	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

PWMCLOEn bits (Port n Watermark Critical Level Overtook Enable (n = 0 to 2))

[Setting condition]

- Writing 1 to one of these bits.

[Clearing condition]

- SW: Writing 1 to one of CAEID1.PWMCLOD bits.

PWMFLOEn bits (Port n Watermark Flush Level Overtook Enable (n = 0 to 2))

[Setting condition]

- Writing 1 to one of these bits.

[Clearing condition]

- SW: Writing 1 to one of CAEID1.PWMFLOD bits.

31.3.4.6 CAEID1 : Error Interrupt Disable Register 1

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0418

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	PWMFLOD2 to PWMFLOD0		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	PWMCLOD2 to PWMCLOD0		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	PWMCLOD2 to PWMCLOD0	Port n Watermark Critical Level Overtook Disable (n = 0 to 2)	R
15:3	—	These bits are read as 0. The write value should be 0.	R
18:16	PWMFLOD2 to PWMFLOD0	Port n Watermark Flush Level Overtook Disable (n = 0 to 2)	R
31:19	—	These bits are read as 0. The write value should be 0.	R

PWMCLODn bits (Port n Watermark Critical Level Overtook Disable (n = 0 to 2))

[Clearing condition]

- SW: Writing 1 to one of these bits clears the corresponding CAEIE1.PWMCLOE bit.

PWMFLODn bits (Port n Watermark Flush Level Overtook Disable (n = 0 to 2))

[Clearing condition]

- SW: Writing 1 to one of these bits clears the corresponding CAEIE1.PWMFLOE bit.

31.3.4.7 CAMIS0 : Monitoring Interrupt Status Register 0

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0440

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PFS1 to PFS0	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PFS1 to PFS0	Pause Frame Status Flag n (n= 0 to 1)	R/W*1
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

PFSn bits (Pause Frame Status Flag n (n= 0 to 1))

[Setting condition]

- HW: For bit i, when pause frame i begins (pause assertion timing), see section 31.5.2. Pause Function.

[Clearing condition]

- SW: Writing 1 to one of these bits.

31.3.4.8 CAMIE0 : Monitoring Interrupt Enable Register 0

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0444

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PFE1 to PFE0	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PFE1 to PFE0	Pause Frame Enable n (n = 0 to 1)	R/W*1
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

PFEn bits (Pause Frame Enable n (n = 0 to 1))

[Setting condition]

- Writing 1 to one of these bits.

[Clearing condition]

- SW: Writing 1 to one of CAMID0.PFD bits.

31.3.4.9 CAMID0 : Monitoring Interrupt Disable Register 0

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0448

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PFD1 to PFD0	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PFD1 to PFD0	Pause Frame Disable n (n = 0 to 1)	R
31:2	—	These bits are read as 0. The write value should be 0.	R

PFDn bits (Pause Frame Disable n (n = 0 to 1))

[Clearing condition]

- SW: Writing 1 to one of these bits clears the corresponding CAMIE0.PFE bit.

31.3.4.10 CAMIS1 : Monitoring Interrupt Status Register 1

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0450

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	PPFS21	PPFS20	PPFS11	PPFS10	PPFS01	PPFS00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PPFS00	Port 0 Pause Frame Status Flag 0	R/W*1
1	PPFS01	Port 0 Pause Frame Status Flag 1	R/W*1
2	PPFS10	Port 1 Pause Frame Status Flag 0	R/W*1
3	PPFS11	Port 1 Pause Frame Status Flag 1	R/W*1
4	PPFS20	Port 2 Pause Frame Status Flag 0	R/W*1
5	PPFS21	Port 2 Pause Frame Status Flag 1	R/W*1
31:6	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

PPFSmn bits (Port m Pause Frame Status n)

[Setting condition]

- HW: For bit i, when pause frame i begins (pause assertion timing), see section 31.5.2. Pause Function.

[Clearing condition]

- SW: Writing 1 to one of these bits.

31.3.4.11 CAMIE1 : Monitoring Interrupt Enable Register 1

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0454

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	PPFE21	PPFE20	PPFE11	PPFE10	PPFE01	PPFE00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PPFE00	Port 0 Pause Frame Enable 0	R/W*1
1	PPFE01	Port 0 Pause Frame Enable 1	R/W*1
2	PPFE10	Port 1 Pause Frame Enable 0	R/W*1
3	PPFE11	Port 1 Pause Frame Enable 1	R/W*1
4	PPFE20	Port 2 Pause Frame Enable 0	R/W*1
5	PPFE21	Port 2 Pause Frame Enable 1	R/W*1
31:6	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

PPFEmn bits (Port m Pause Frame Enable n)

[Setting condition]

- Writing 1 to one of these bits.

[Clearing condition]

- SW: Writing 1 to one of CAMID1.PPFD bits.

31.3.4.12 CAMID1 : Monitoring Interrupt Disable Register 1

Base address: COMA = 0x403C_9000
COMA_NS = 0x503C_9000

Offset address: 0x0458

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	PPFD21	PPFD20	PPFD11	PPFD10	PPFD01	PPFD00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PPFD00	Port 0 Pause Frame Disable 0	W*1
1	PPFD01	Port 0 Pause Frame Disable 1	W*1
2	PPFD10	Port 1 Pause Frame Disable 0	W*1
3	PPFD11	Port 1 Pause Frame Disable 1	W*1
4	PPFD20	Port 2 Pause Frame Disable 0	W*1
5	PPFD21	Port 2 Pause Frame Disable 1	W*1
31:6	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

PPFDmn bits (Port m Pause Frame Disable n)

[Clearing condition]

- SW: Writing 1 to one of these bits clears the corresponding CAMIE1.PPFE bit.

31.4 Register Utilization

31.4.1 Software Flows

SW: Follow the flows described in this section.

31.4.1.1 Software Flow Legend

Figure 31.2 shows the software flow legend.

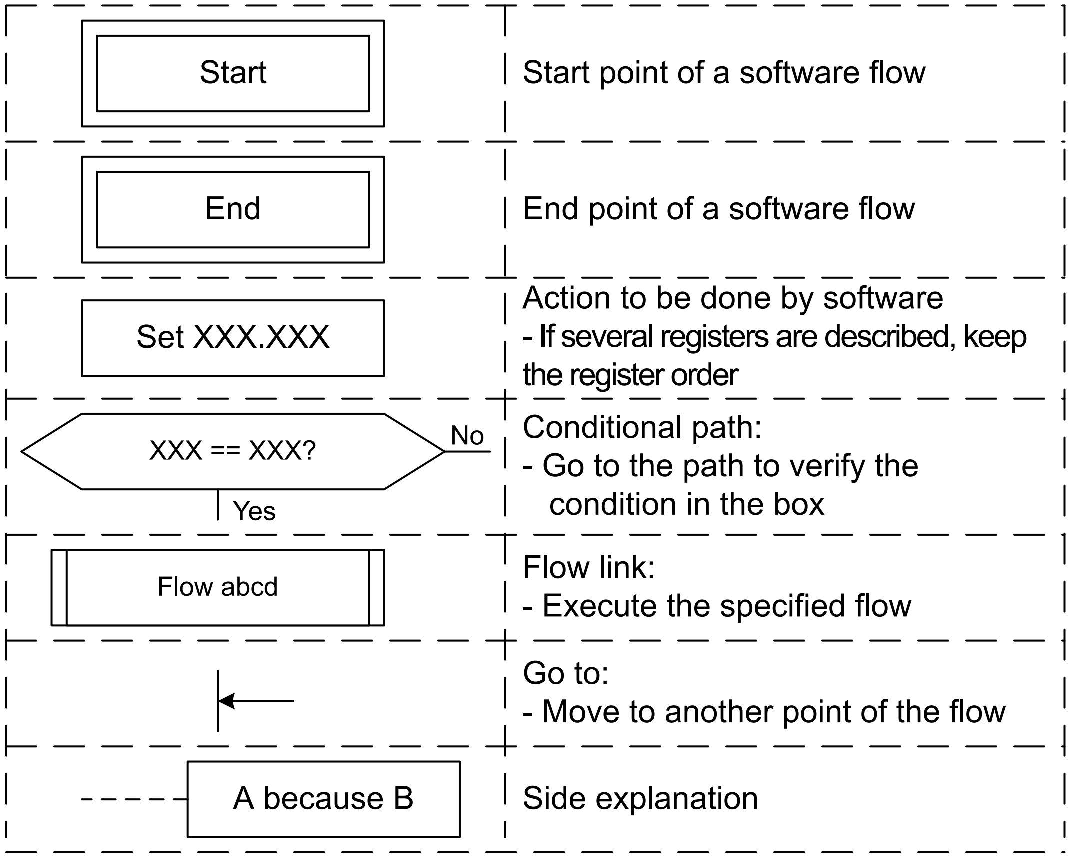


Figure 31.2 Software flow legend

31.4.1.2 Interrupt Handling Flow

Interrupt handling flow can be used in any mode except Reset mode.

Figure 31.3 shows the interrupt handling flow.

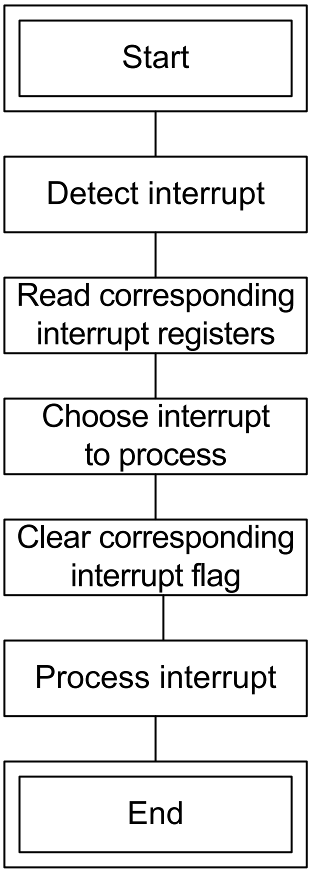


Figure 31.3 Interrupt handling flow

31.4.1.3 Called Software Flows

The flows described in this section can only be called from other flows as indicated in the flow link step in Figure 31.2 and cannot be used alone.

31.4.1.3.1 Buffer Pool Initialization Flow

Figure 31.4 shows the buffer pool initialization flow.

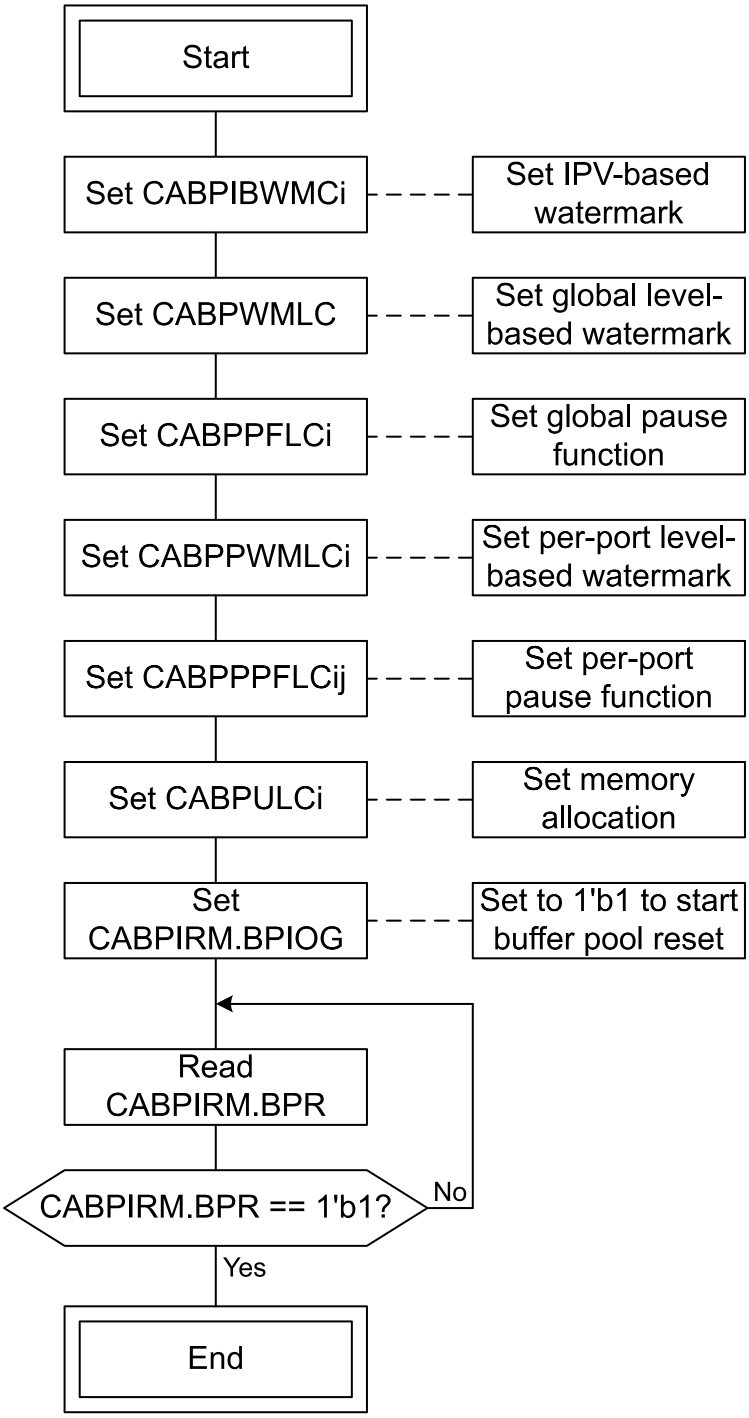


Figure 31.4 Buffer pool initialization flow

31.4.1.3.2 ESWM Reset Flow

Figure 31.5 shows the ESWM reset flow.

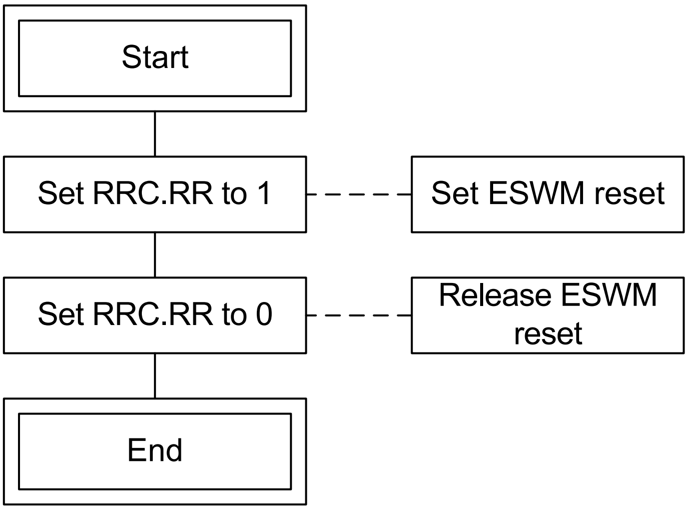


Figure 31.5 ESWM reset flow

31.4.1.3.3 Agent i Clock Control Flows (i = 0 to 3)

Figure 31.6, Figure 31.7 and Figure 31.8 show the agent i clock control flows.

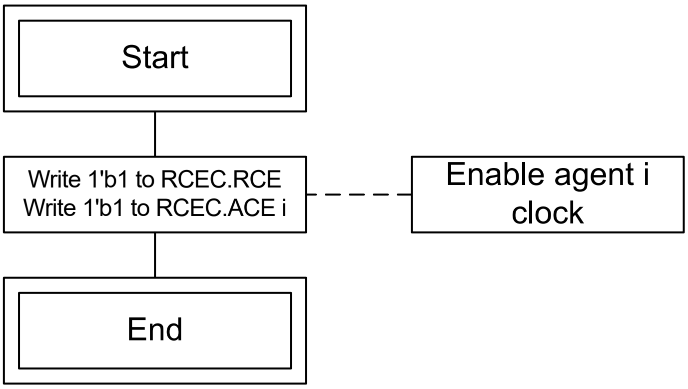


Figure 31.6 Agent i clock enable flow

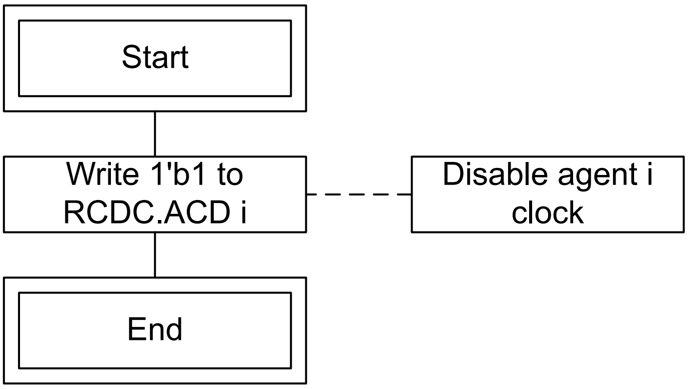


Figure 31.7 Agent i clock disable flow

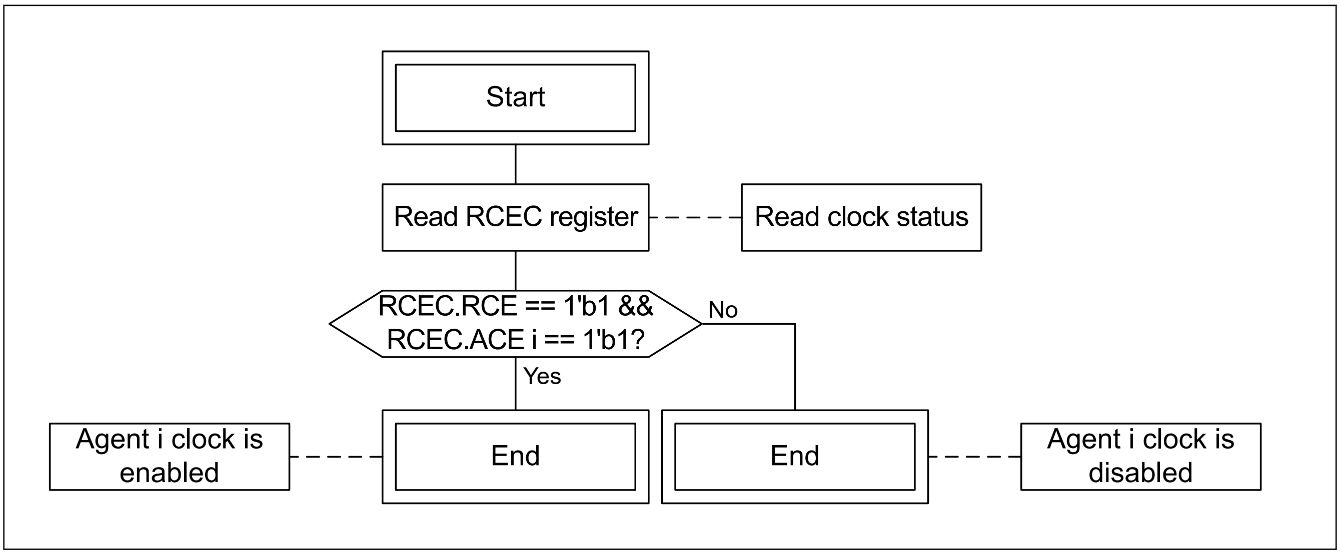


Figure 31.8 Agent i clock status check flow

31.4.1.3.4 Switch Clock Control Flows

Figure 31.9 and Figure 31.10 show the switch clock control flows.

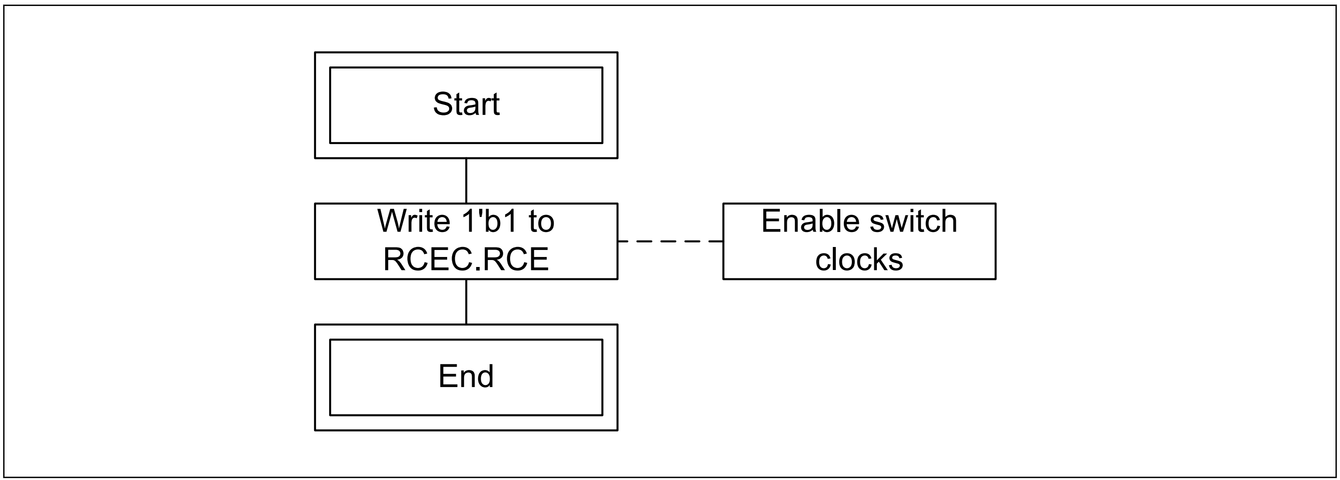


Figure 31.9 Switch clock enable flow

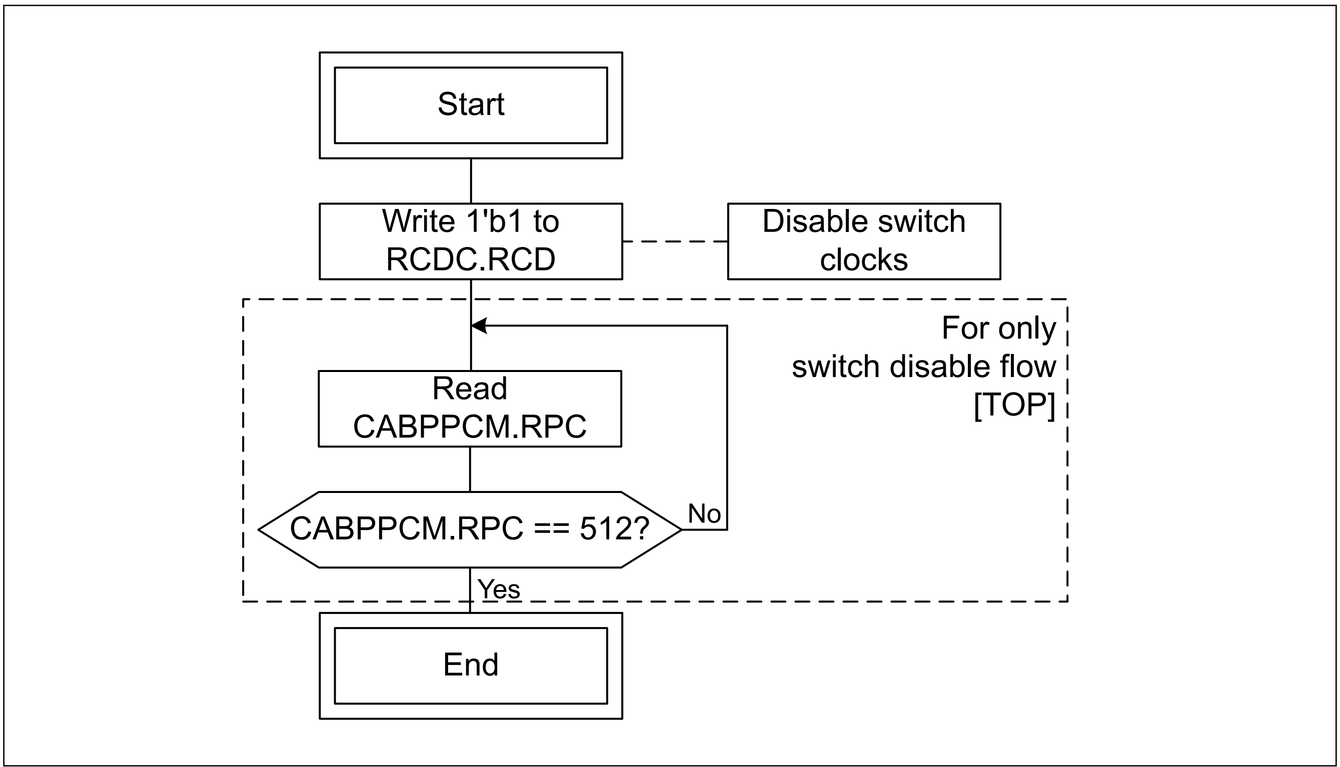


Figure 31.10 Switch clock disable flow

31.4.1.4 Register Writable Without Software Flow

This section describes registers that have not been described. These registers can be changed dynamically. However, it is necessary that the initial settings such as clock enabling have been completed.

31.5 Functional Description

31.5.1 Watermark Function

If a switch overflow occurs, frames in the switch are lost at the ingress part of agents [GWCA] [ETHA]. This result in a random loss of data is independent of the actual priority of frames. To avoid this, the watermark function allows to selectively reject frames in MFWD before a switch overflow occurs.

Ethernet Common Agent includes three watermark functions:

- IPV-based watermark
- Global level-based watermark
- Per-port level-based watermark

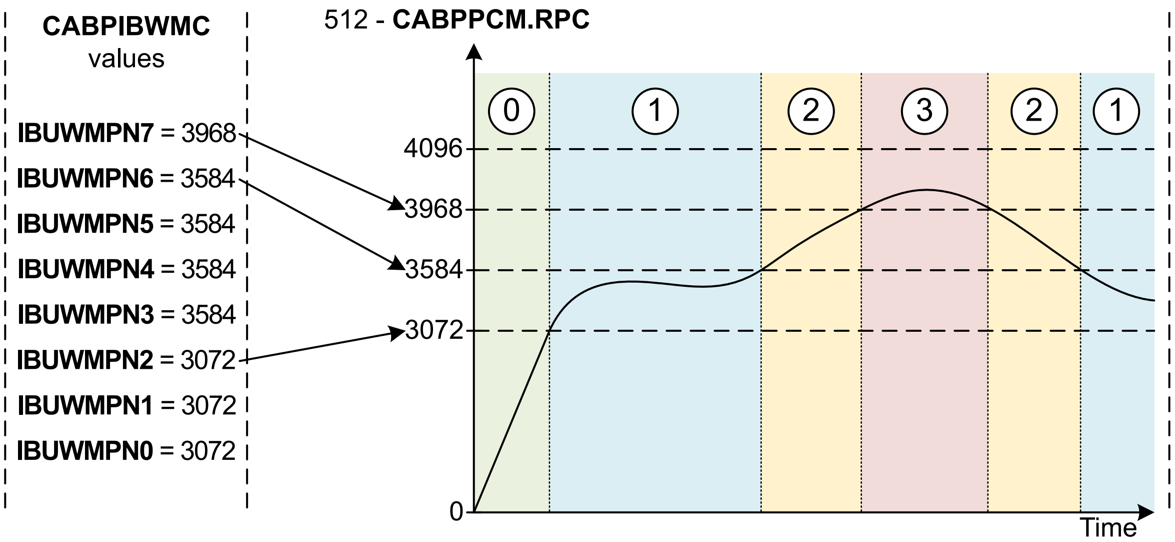
The watermark functions aim at controlling the switch behavior in case of overflow by generating data loss. All watermark functions are running in parallel and a frame can be rejected by one or more watermark functions.

31.5.1.1 IPV-based Watermark

The IPV-based watermark ensures the minimum QoS requirement for a switch. It is enabled by the default configurations for a switch maximum frame size of 256 bytes. This function is controlled by the Ethernet Common Agent, however, data loss is performed by the Forwarding Engine [FWD]. IPV-based watermark with a combination of other functions, is affected by the queue pause regardless of frame size (256).

Figure 31.11 shows an example of IPV-based unsecure watermark setting for Pointer number to address local RAM = = 4096. The IPV-based secure watermark works the same way but using CABPIBWMCi.IBSWMPN registers.

Note: The variations of 512 - CABPPCM.RPC in Figure 31.11 are arbitrary and do not correspond to anything relevant.



	Explanation
0	No descriptor is rejected because **CABPIBWMC.IBUWMPN2** is larger than 512 - **CABPPCM.RPC**
1	Descriptors with **FDESCR.IPV** 0, 1 or 2 are rejected because **CABPIBWMC.IBUWMPN2** is smaller than 512 - **CABPPCM.RPC**
2	Descriptors with **FDESCR.IPV** 0, 1, 2, 3, 4, 5 or 6 are rejected because **CABPIBWMC.IBUWMPN6** is smaller than 512 - **CABPPCM.RPC**
3	All descriptor are rejected because **CABPIBWMC.IBUWMPN7** is smaller than 512 - **CABPPCM.RPC**

Figure 31.11 Example for IPV-based watermark setting

Note: FDESCR.IPV refers to the internal priority value given to a frame by forwarding engine in the forwarding descriptor but before filtering. For more details, see the Forwarding Engine specification [FWD].

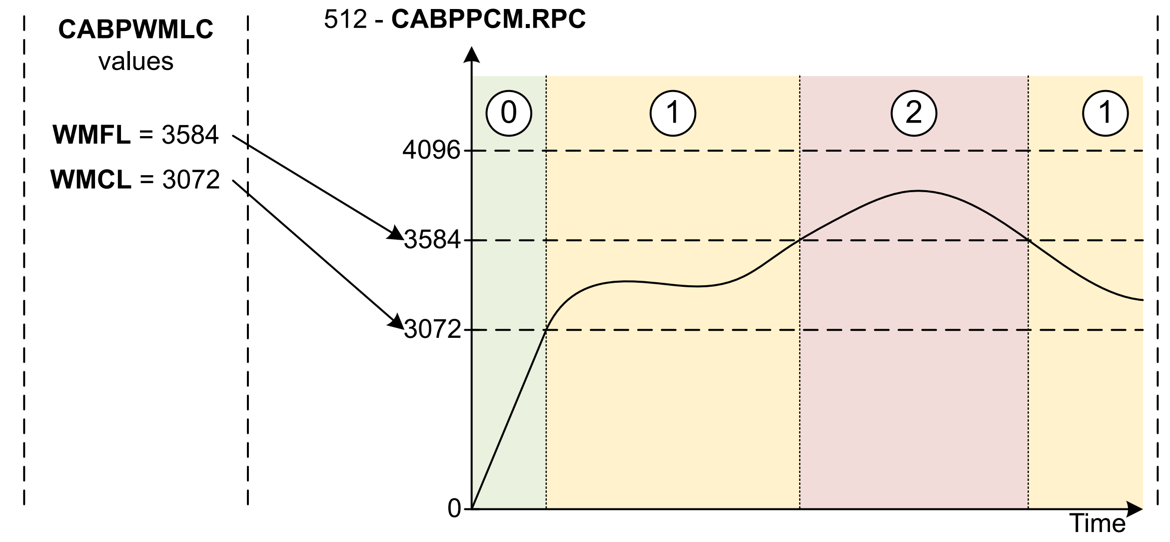
31.5.1.2 Global Level-based Watermark

The global level-based watermark aims at rejecting frames with targeted IPVs depending on the number of pointers currently used by the switch (CABPPCM.RPC register). This function is controlled by the Ethernet Common Agent (through WM.CRITICAL[3:0] and WM.FLUSH[3:0]), however, data loss is performed by the Forwarding Engine [FWD].

Note: WM.CRITICAL and WM.FLUSH interfaces can also be set by the per-port level-based watermark (one of these pins is set if the per-port level-based watermark or the global level-based watermark is in the required setting conditions).

Figure 31.12 shows an example for global level-based watermark setting for Pointer number to address local RAM = = 4096.

Note: The variations of 512 - CABPPCM.RPC in Figure 31.12 are arbitrary and do not correspond to anything relevant.



	Explanation	WM.CRITICAL/ WM.FLUSH values
0	No descriptor is rejected because **CABPWMLC.WMCL** is larger than 512 - **CABPPCM.RPC**	All0/All0
1	Descriptors associated with all port-critical level are rejected because 512 - **CABPPCM.RPC** is larger **CABPWMLC.WMCL**	All1/All0
2	Descriptors associated with all port-critical/flush level are rejected because 512 - **CABPPCM.RPC** is larger than **CABPWMLC.WMFL**	All1/All1

Figure 31.12 Example for global level-based watermark setting

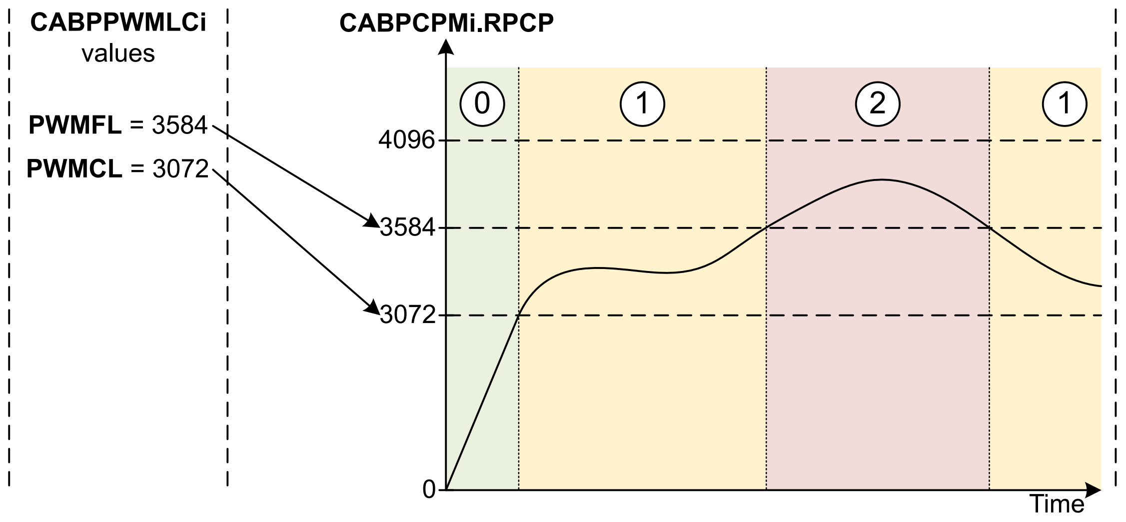
31.5.1.3 Per-port Level-based Watermark i (i = 0 to 3)

The per-port level-based watermark aims at rejecting frames with targeted IPVs depending on CABPPWMLCi.PWMCL. This function is controlled by the Ethernet Common Agent (through WM.CRITICAL[i] and WM.FLUSH[i]), however, data loss is performed by the Forwarding Engine [FWD].

Note: WM.CRITICAL and WM.FLUSH interfaces can also be set by the global level-based watermark (one of these pins is set if the per-port level-based watermark or the global level-based watermark is in the required setting conditions).

Figure 31.13 shows an example of per-port level-based watermark setting for port i when Pointer number to address local RAM = = 4096.

Note: The variations of CABPCPMi.RPCP in Figure 31.13 are arbitrary and do not correspond to anything relevant.



	Explanation	WM.CRITICAL[i]/ WM.FLUSH[i] values
0	No descriptor is rejected because **CABPPWMLCi.PWMCL** is larger than **CABPCPMi.RPCP**	0/0
1	Descriptors associated with all port-critical level are rejected **CABPPWMLCi.PWMCL** is smaller than **CABPCPMi.RPCP**	1/0
2	Descriptors associated with all port-critical/flush level are rejected **CABPPWMLCi.PWMFL** is smaller than **CABPCPMi.RPCP**	1/1

Figure 31.13 Example for per-port level-based watermark setting for port i

31.5.2 Pause Function

If a switch overflow occurs, frames in the switch are lost at the ingress part of agents [GWCA] [ETHA]. This result in a random loss of data is independent of the actual priority of frames. To avoid this, the pause function allows to selectively pause frame injection by agents [GWCA] [ETHA].

Ethernet Common Agent includes two pause functions, the global pause function and the per-port pause function. The pause functions aim at controlling the switch behavior in case of overflow by generating backpressure. All pause functions are running in parallel and backpressure can be applied by one or more pause functions.

31.5.2.1 Global Pause Function

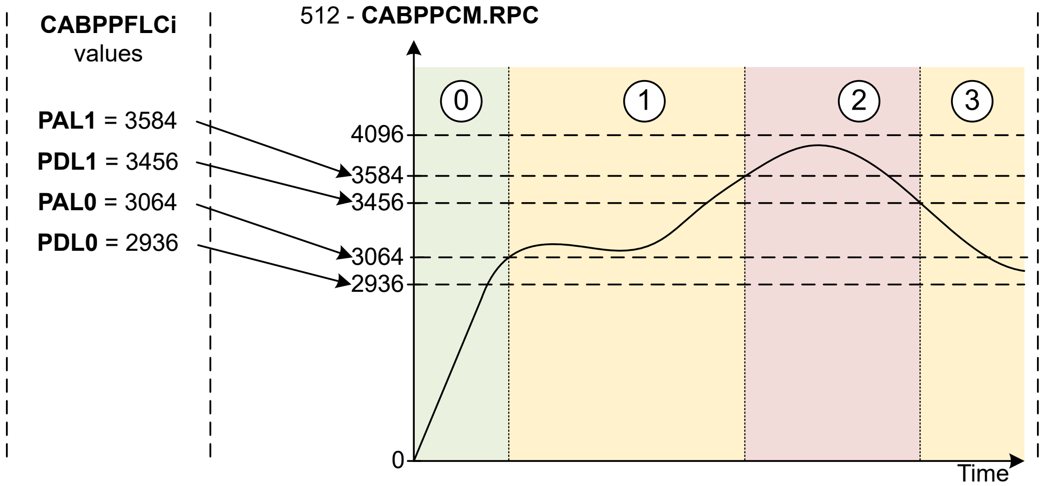
The global pause function aims at applying backpressure on agents for targeted IPVs depending on CABPPCM.RPC. The pause function can pause agents with two combinations of IPVs.

The global pause function is controlled by the Ethernet Common Agent PAUSE[3:0][1:0] interface, however, backpressure is performed by the agents [ETHA] [GWCA].

Note: The pause interface can also be set by the per-port pause function (one of these pins is set if the per-port pause function or the global pause function is in the required setting conditions).

Figure 31.14 shows an example of global pause function setting for Pause level number = = 2 and Pointer number to address local RAM = = 4096.

Note: The variations of 512 - CABPPCM.RPC in Figure 31.14 are arbitrary and do not correspond to anything relevant.



	Explanation	PAUSE[i][0]/ PAUSE[i][1] values (for all i = 0 to 2)
0	No agent paused because 512 - **CABPPCM.RPC** never exceeded **CABPPFLC0.PAL0** level nor **CABPPFLC1.PAL1** level	0/0
1	Agents are paused with level 0 because 512 - **CABPPCM.RPC** exceeded **CABPPFLC0.PAL0** level and not **CABPPFLC1.PAL1** level	1/0
2	Agents are paused with level 0 and 1 because 512 - **CABPPCM.RPC** exceeded **CABPPFLC0.PAL0** level and **CABPPFLC1.PAL1** level	1/1
3	Agents are paused with level 0 because 512 - **CABPPCM.RPC** exceeded **CABPPFLC0.PAL0** and dropped under **CABPPFLC1.PDL1**	1/0

Figure 31.14 Example for global pause function setting

31.5.2.2 Per-port Pause Function i (i = 0 to 3)

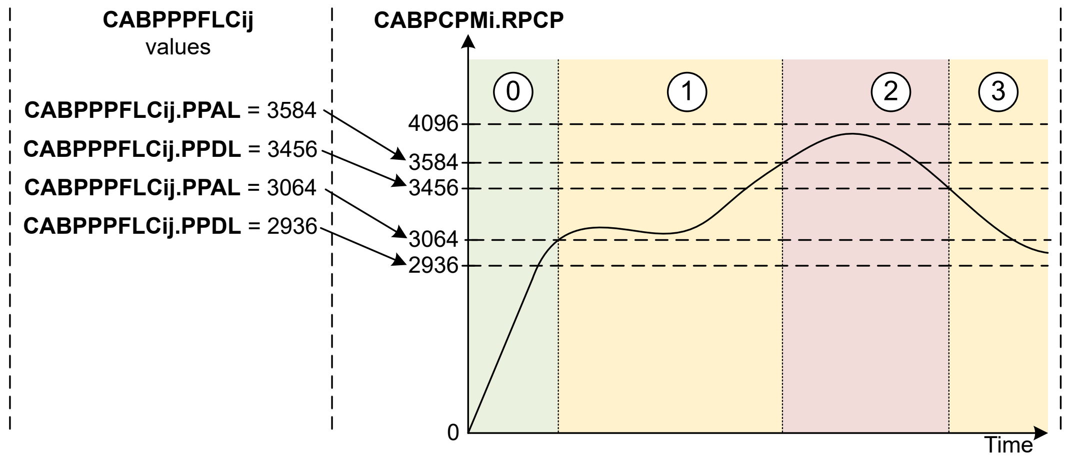
The per-port pause function aims at applying backpressure on agents for targeted IPVs depending on CABPPWMLCi.PWMCL. The pause function can pause agents with two combinations of IPVs.

This function is controlled by the Ethernet Common Agent PAUSE[i] interface, however, backpressure is performed by the agents [ETHA] [GWCA].

Note: The pause interface can also be set by the per-port pause function (one of these pins is set if the per-port pause function or the global pause function is in the required setting conditions).

Figure 31.15 shows an example of per-port pause function setting for port i when Pause level number = = 2 and Pointer number to address local RAM = = 4096.

Note: The variations of CABPPWMLCi.PWMCL in Figure 31.15 are arbitrary and do not correspond to anything relevant.



	Explanation	**PAUSE[i][0]**/ **PAUSE[i][0]** values
0	No agent paused because **CABPCPMi.RPCP** never exceeded **CABPPPFLCi0.PPAL** level nor **CABPPPFLCi1.PPAL** level	0/0
1	Agents are paused with level 0 because **CABPCPMi.RPCP** exceeded **CABPPPFLCi0.PPAL** level and not **CABPPPFLCi1.PPAL** level	1/0
2	Agents are paused with level 0 and level 1 because **CABPCPMi.RPCP** exceeded **CABPPPFLCi0.PPAL** level and **CABPPPFLCi1.PPAL** level	1/1
3	Agents are paused with level 0 because **CABPCPMi.RPCP** exceeded **CABPPPFLCi0.PPAL** and dropped under **CABPPPFLCi1.PPDL**	1/0

Figure 31.15 Example for per-port pause function setting for port i

31.5.3 Per-port Memory Allocation Function

It is possible to control the maximum and the minimum number of pointers that can be used per port due to CABPULCi.MNNPN and CABPULCi.MXNPN. Any agent can use at least its minimum number of pointers and at most its maximum number of pointers. This section provides some setting examples for the memory allocation function.

31.5.3.1 Shared Memory Setting (Default Setting)

In shared memory, the data RAM is shared by all the ports and is the reset state of the Ethernet Common Agent.

Figure 31.16 describes the shared memory setting for Pointer number to address local RAM = = 4096 and Port number on the switch = = 3.

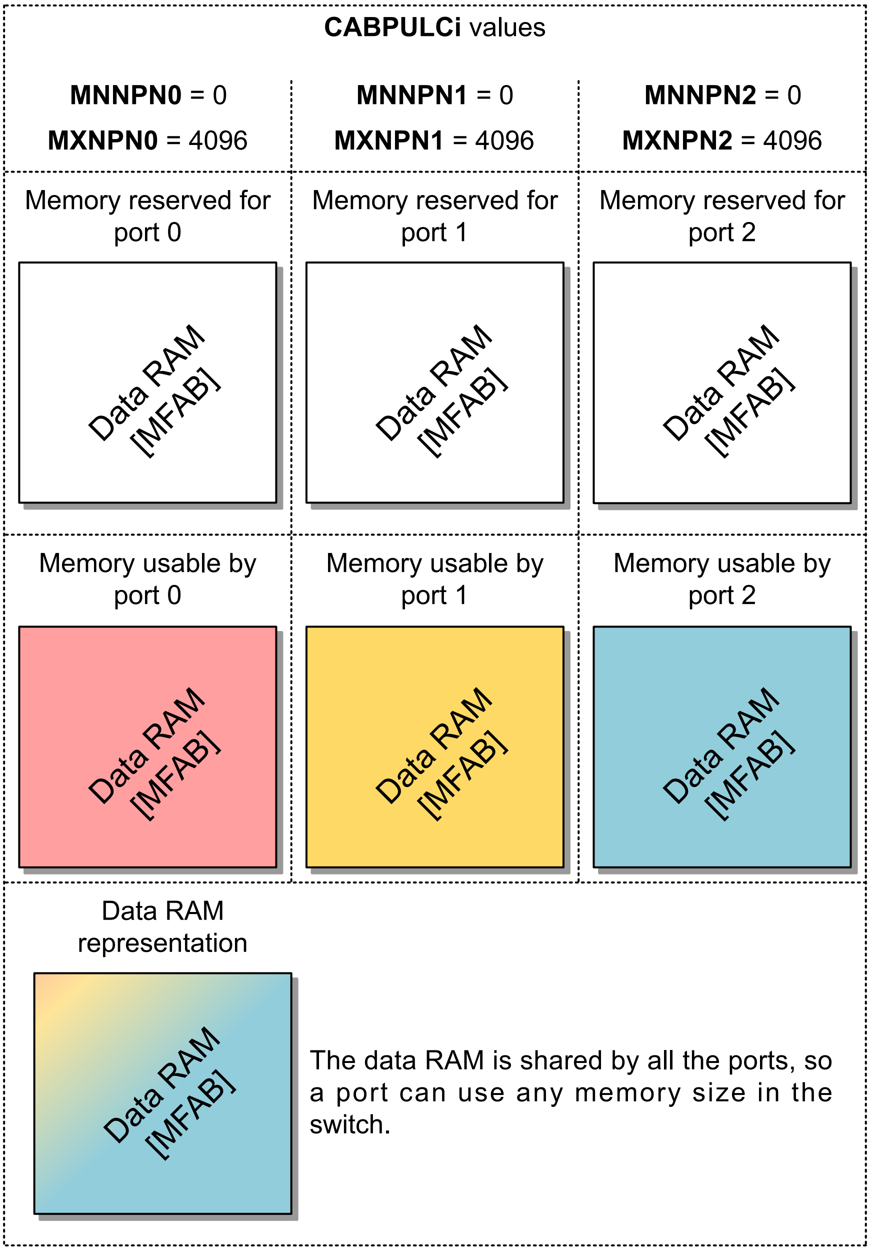


Figure 31.16 Shared memory setting

31.5.3.2 Reduced Memory Setting 1 (Recommended Setting)

In reduced memory, the data RAM is shared by all agents but some of them are not able to use all the memory because they can make the switch overflow (especially for CPU ports [GWCA]). To set this mode, use the CABPULCi.MXNPN register.

Figure 31.17 shows an example of reduced memory setting for Pointer number to address local RAM = = 4096 and Port number on the switch = = 3. For example, if port 2 is a fast CPU port, CABPULCi.MXNPN register can limit the number of reception frames.

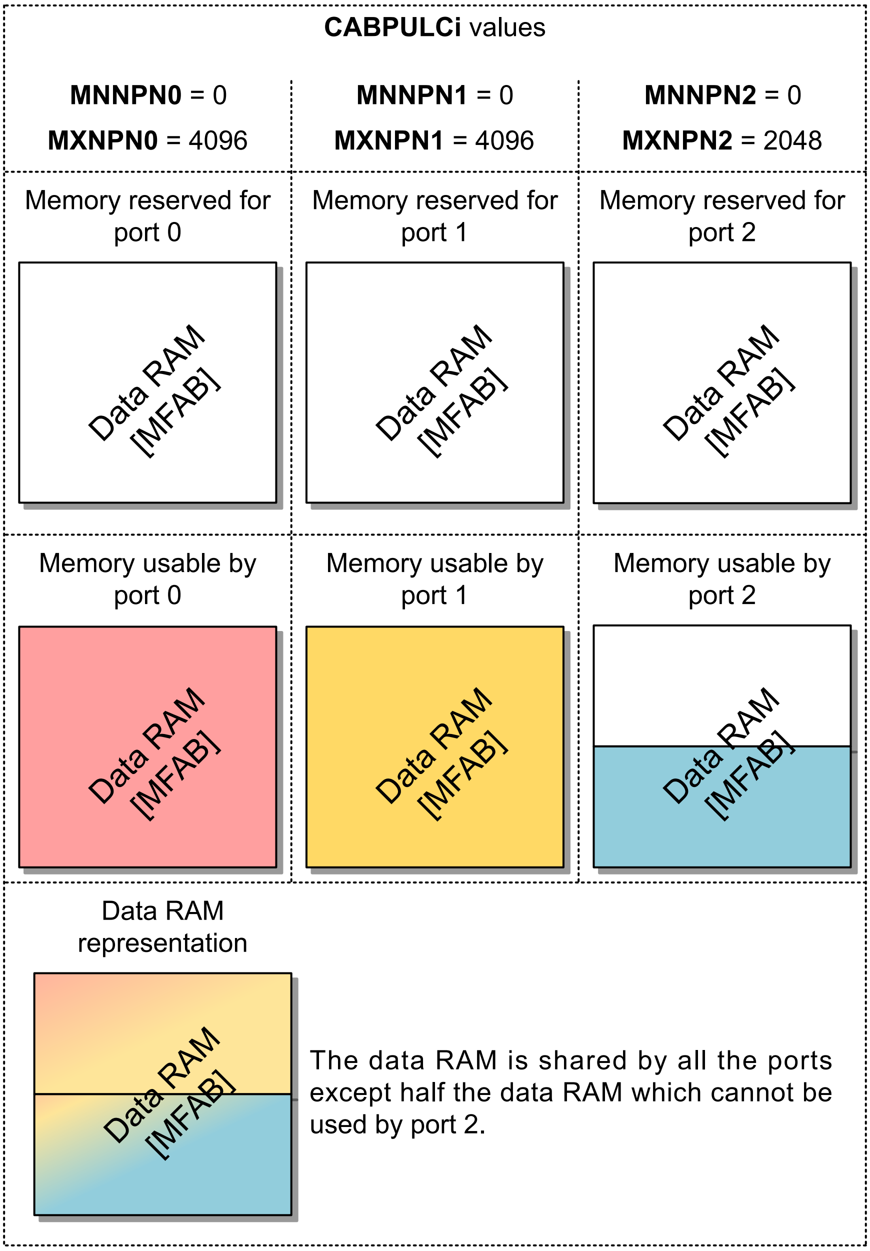


Figure 31.17 Reduced memory setting example

31.5.3.3 Reduced Memory Setting 2 (Recommended Setting)

In reduced memory, the data RAM is shared by all agents but some of them are not able to use all the memory because they can make the switch overflow (especially for CPU ports [GWCA]). To set this mode, use the CABPULCi.MNNPN register.

Figure 31.18 shows an example of reduced memory setting for Pointer number to address local RAM = = 4096 and Port number on the switch = = 3. For example, if port 0 or port 1 is a slow Ethernet port, CABPULCi.MNNPN register can protect the number of frame buffer.

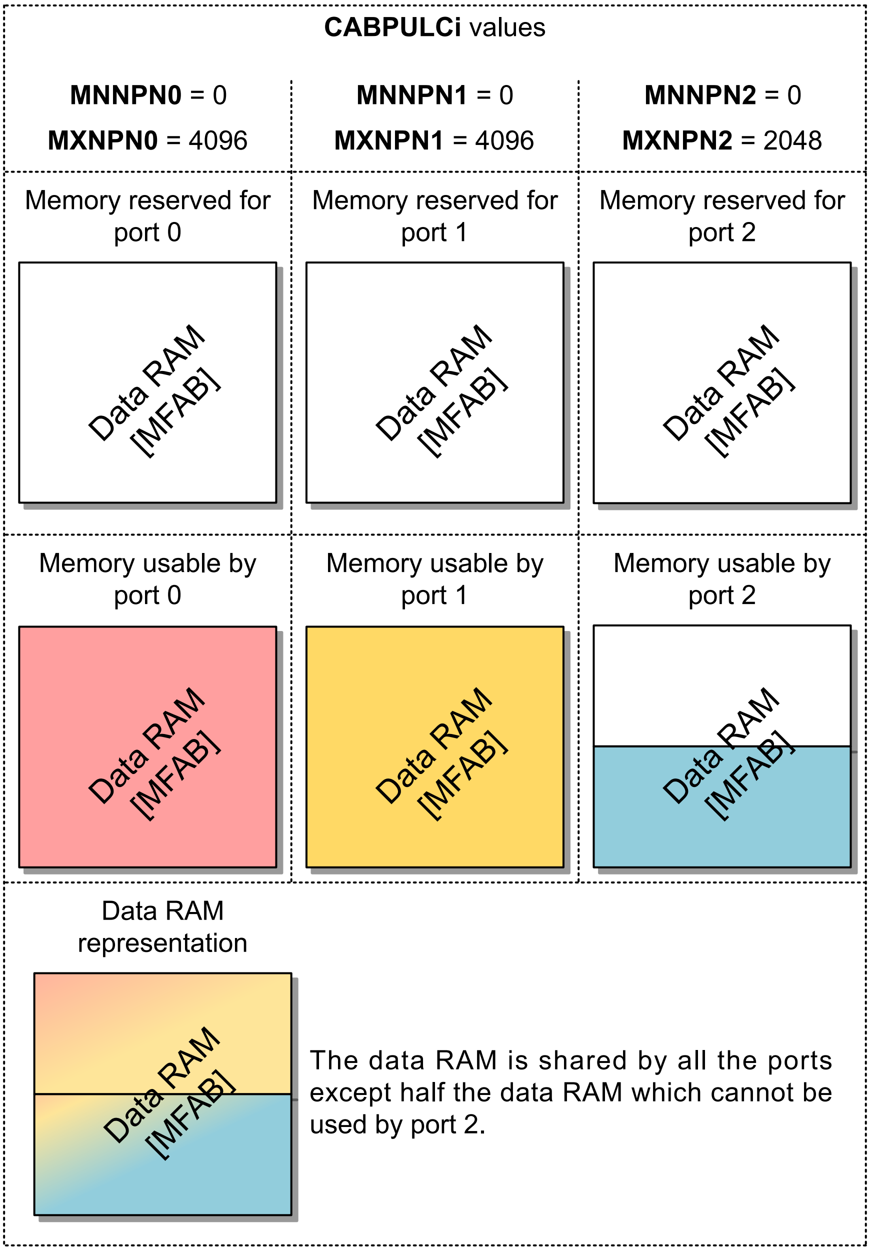


Figure 31.18 Reduced memory setting example

31.5.3.4 Hybrid Memory Setting

All the setting described in section 31.5.3.1. Shared Memory Setting (Default Setting), section 31.5.3.2. Reduced Memory Setting 1 (Recommended Setting) and section 31.5.3.3. Reduced Memory Setting 2 (Recommended Setting) can be used in combination to describe a complex memory such as the hybrid memory setting.

Figure 31.19 shows an example of hybrid memory setting for Pointer number to address local RAM = = 4096 and Port number on the switch = = 4.

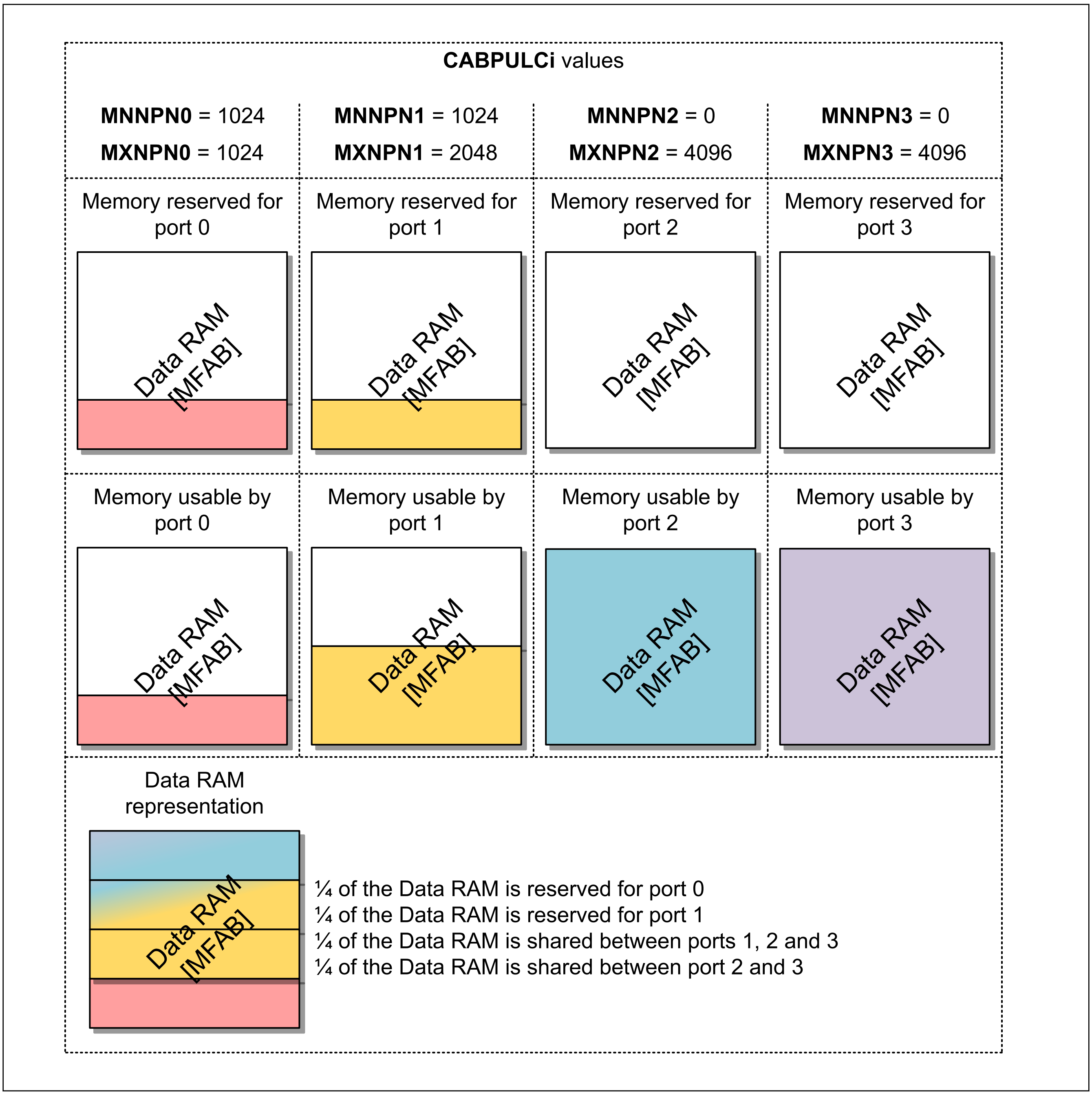


Figure 31.19 Hybrid memory setting example

31.6 Precautions

31.6.1 Precautions

Not applicable.

31.6.2 Restrictions (Including Known Problems)

Not applicable.

32. Ethernet Agent (ETHA)

32.1 Overview

Ethernet Agent (ETHA) consists of an agent interface module to allow communication within ESWM. ETHA handles data exchange between ESWM and an Ethernet PHY.

Table 32.1 lists the ETHA specifications and Figure 32.1 shows a ETHA block diagram.

Table 32.1 ETHA specifications

Item	Function	Description
Interfaces	Clock/Reset interface	Clock/Reset interface [TOP]
	Scan interface	Scan interface [TOP]
	Switch mode	Interface to receive the switch mode [MFWD]
	Pause interface	Interface to pause TX queues [COMA]
	SFR interface	Interface to access GWCA SFRs [COMA]
	Interrupts	GWCA interrupt to CPU
	Fabric interface	Interface to communicate with the data, TAG and pointer RAM [MFAB]
	Descriptor bus	Interface to receive frame information to send to CPU [MFWD]
	L2/L3 update bus	Interface to fetch the frame routing information [MFWD]
	BP request	Interface to receive BPs to store frames in the data, TAG and pointer RAM [COMA]
	BP release	Interface to release the BPs that has been used [COMA]
	TX timestamp capture interface	Interface to send TX timestamps to CPU [RMAC]
	RAM interfaces	Interface to communicate with RAMs [TOP]
Data provision	Ethernet frames	Corresponds to the IEEE 802.3, 802.1Q and 802.1CB standards [802.3] [802.1Q] [802.1CB]
	Descriptors	Local ethernet descriptor (ETHA to MFWD). Forwarding descriptor (MFWD to ETHA).

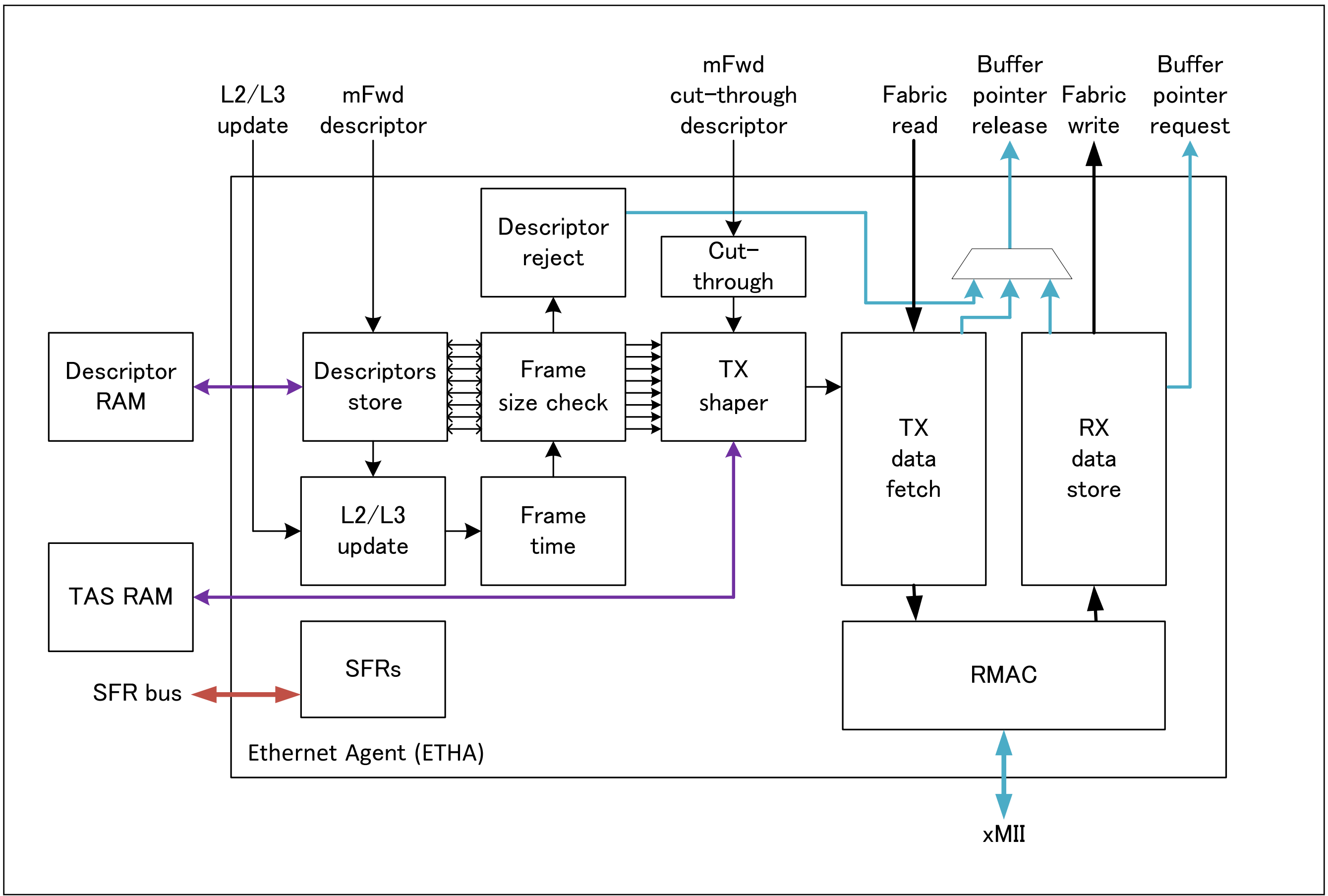


Figure 32.1 ETHA block diagram

Table 32.2 ETHA functional blocks

Block name	Function
RX data store	Transmits frames from RMAC to the local RAM
TX data fetch	Transmits frame from local RAM to RMAC
Cut-through	Receives cut-through descriptors from the forwarding engine
Descriptor store	Receives the descriptors from forwarding engine and controls the priority between descriptors
L2/L3 update	Fetches L2/L3 update information
Frame time	Calculates the frame time for TAS scheduling
Frame size check	Checks the frame size
Descriptor reject	Rejects descriptors
TX shaper	Shapes the traffic on TX side
RMAC	Mac interface [RMAC]
SFRs	ETHA SFRs

32.2 ETHA Register List

Table 32.3 ETHA registers (1 of 3)

Offset address	Register name	Abbreviation	Write access mode
0x0000	Mode Configuration Register	EAMC	Any
0x0004	Mode Status Register	EAMS	—
0x0010	IPV Remapping Configuration Register [802.1Q]	EAIRC	C
0x0014	TX Descriptor Queue Security Configuration Register	EATDQSC	C, O

Table 32.3 ETHA registers (2 of 3)

Offset address	Register name	Abbreviation	Write access mode
0x0018	TX Descriptor Queue Configuration Register	EATDQC	O
0x001C	TX Descriptor Queue Arbitration Configuration Register	EATDQAC	C
0x0020	TX Preemption Configuration Register	EATPEC	C
0x0040 + 0x04 × q	Transmission Maximum Frame Size Configuration Register q	EATMFSCq	C, O
0x0060 + 0x04 × q	Transmission Descriptor Queue Depth Configuration Register q	EATDQDCq	C
0x0080 + 0x04 × q	Transmission Descriptor Queue q Monitoring Register	EATDQMq	—
0x00A0 + 0x04 × q E: 0x00C0 + 0x04 × q	Transmission Descriptor Queue q Max Level Monitoring Register	EATDQMLMq	—
0x0100	Cut-Through Queue Configuration Register	EACTQC	C
0x0104	Cut-Through Descriptor Queue Depth Configuration Register	EACTDQDC	C
0x0108	Cut-Through Descriptor Queue Monitoring Register	EACTDQM	—
0x010C E: 0x0110	Cut-Through Descriptor Queue Max Level Monitoring Register	EACTDQMLM	—
0x0130	VLAN Control Configuration Register	EAVCC	C
0x0134	VLAN TAG Configuration Register	EAVTC	C
0x0138	Reception TAG Filtering Configuration Register	EARTFC	C
0x0200	CBS Admin Enable Configuration Register	EACAEC	C, O
0x0204	CBS Configuration Register	EACC	C, O
0x0220 + 0x04 × q	CBS Admin Increment Value Configuration Register q	EACAIVCq	C, O
0x0240 + 0x04 × q	CBS Admin Upper Limit Configuration Register q	EACAULCq	C, O
0x0260	CBS Oper Enable Monitoring Register	EACOEM	—
0x0280 + 0x04 × q	CBS Oper Increment Value Monitoring Register q	EACOIVMq	—
0x02A0 + 0x04 × q	CBS Oper Upper Limit Monitoring Register q	EACOULMq	—
0x02C0	CBS Gate State Monitoring Register	EACGSM	—
0x0300	TAS Configuration Register	EATASC	C, O
0x0304	TAS Initial Gate State Configuration Register	EATASIGSC	C, O
0x0320 + 0x04 × i	TAS Entry Number Configuration Register i	EATASENCi	C, O
0x0340	TAS Cut-Through Entry Number Configuration Register	EATASCTENC	C, O
0x0360 + 0x04 × i	TAS Entry Number Monitoring Register i	EATASENMi	C, O
0x0380	Cut-Through Entry Number Monitoring Register	EATASCTENM	C, O
0x03A0	TAS Cycle Start Time Configuration Register 0	EATASCSTC0	C, O
0x03A4	TAS Cycle Start Time Configuration Register 1	EATASCSTC1	C, O
0x03A8	TAS Cycle Start Time Monitoring Register 0	EATASCSTM0	C, O
0x03AC	TAS Cycle Start Time Monitoring Register 1	EATASCSTM1	C, O
0x03B0	TAS Cycle Time Configuration Register	EATASCTC	C, O
0x03B4	TAS Cycle Time Monitoring Register	EATASCTM	C, O
0x03C0	TAS Gate Learn Register 0	EATASGL0	C, O
0x03C4	TAS Gate Learn Register 1	EATASGL1	C, O
0x03C8	TAS Gate Learn Result Register	EATASGLR	—
0x03D0	TAS Gate Read Register	EATASGR	C, O
0x03D4	TAS Gate Read Result Register	EATASGRR	—
0x03E0	TAS Hardware Calibration Configuration Register	EATASHCC	C

Table 32.3 ETHA registers (3 of 3)

Offset address	Register name	Abbreviation	Write access mode
0x03E4	TAS RAM Initialization Register Monitoring Register	EATASRIRM	C, O
0x03E8	TAS Status Monitoring Register	EATASSM	—
0x0400 E: 0x0480	Under Minimum Frame Size Error Counter Register	EAUSMFSECN	—
0x0404 E: 0x0484	TAG Filtering Error Counter Register	EATFECN	—
0x0408 E: 0x0488	Frame Size Error Counter Register	EAFSECN	—
0x040C E: 0x048C	Descriptor Queue Overflow Error Counter Register	EADQOECN	—
0x0410 E: 0x0490	Descriptor Queue Security Error Counter Register	EADQSECN	—
0x0500	Error Interrupt Status Register 0	EAEIS0	C, O, D
0x0504	Error Interrupt Enable Register 0	EAEIE0	C, O, D
0x0508	Error Interrupt Disable Register 0	EAEID0	C, O, D
0x0510	Error Interrupt Status Register 1	EAEIS1	C, O, D
0x0514	Error Interrupt Enable Register 1	EAEIE1	C, O, D
0x0518	Error Interrupt Disable Register 1	EAEID1	C, O, D
0x0520	Error Interrupt Status Register 2	EAEIS2	C, O, D
0x0524	Error Interrupt Enable Register 2	EAEIE2	C, O, D
0x0528	Error Interrupt Disable Register 2	EAEID2	C, O, D

Note:
- Base address: ETHA0 = 0x403C_A000, ETHA1 = 0x403C_C000, ETHA0_NS = 0x503C_A000, ETHA1_NS = 0x503C_C000
- The address preceded by “E:” corresponds to an emulation address which allows a register to be read without modifying its content.
- All registers can be read in any mode
- Access modes:
  - Any: Register can be accessed in any mode
  - D: Register can be accessed in DISABLE mode
  - C: Register can be accessed in CONFIG mode
  - O: Register can be accessed in OPERATION mode.

32.3 ETHA Register Descriptions

32.3.1 ETHA Mode Function Registers

32.3.1.1 EAMC : Mode Configuration Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0000

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

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

Bit	Symbol	Function	R/W
1:0	OPC[1:0]	Operating Mode Command Used to set ETHA mode 0 0: Enter RESET mode 0 1: Enter DISABLE mode 1 0: Enter CONFIG mode 1 1: Enter OPERATION mode	R/W
31:2	—	These bits are read as 0.	R

Note: HW: This register is not writable if its value is different than EAMS.OPS.

32.3.1.2 EAMS : Mode Status Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0004

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

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

Bit	Symbol	Function	R/W
1:0	OPS[1:0]	Operating Mode Status Flag Indicates the current ETHA mode 0 0: RESET mode 0 1: DISABLE mode 1 0: CONFIG mode 1 1: OPERATION mode	R
31:2	—	These bits are read as 0.	R

32.3.2 Transmission Function Registers

32.3.2.1 EAIRC : IPV Remapping Configuration Register [802.1Q]

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	IPVR7[2:0]			—	IPVR6[2:0]			—	IPVR5[2:0]			—	IPVR4[2:0]		
Value after reset:	0	1	1	1	0	1	1	0	0	1	0	1	0	1	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	IPVR3[2:0]			—	IPVR2[2:0]			—	IPVR1[2:0]			—	IPVR0[2:0]		
Value after reset:	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
2:0	IPVR0[2:0]	IPV Remapping 0 Configure to which queue descriptor received with IPV 0 is stored (when a descriptor is received from forwarding engine with FDESCR.IPV [MFWD] equal to 0).	R/W

Bit	Symbol	Function	R/W
3	—	This bit is read as 0.	R
6:4	IPVR1[2:0]	IPV Remapping 1 Configure to which queue descriptor received with IPV 1 is stored (when a descriptor is received from forwarding engine with FDESCR.IPV [MFWD] equal to 1).	R/W
7	—	This bit is read as 0.	R
10:8	IPVR2[2:0]	IPV Remapping 2 Configure to which queue descriptor received with IPV 2 is stored (when a descriptor is received from forwarding engine with FDESCR.IPV [MFWD] equal to 2).	R/W
11	—	This bit is read as 0.	R
14:12	IPVR3[2:0]	IPV Remapping 3 Configure to which queue descriptor received with IPV 3 is stored (when a descriptor is received from forwarding engine with FDESCR.IPV [MFWD] equal to 3).	R/W
15	—	This bit is read as 0.	R
18:16	IPVR4[2:0]	IPV Remapping 4 Configure to which queue descriptor received with IPV 4 is stored (when a descriptor is received from forwarding engine with FDESCR.IPV [MFWD] equal to 4).	R/W
19	—	This bit is read as 0.	R
22:20	IPVR5[2:0]	IPV Remapping 5 Configure to which queue descriptor received with IPV 5 is stored (when a descriptor is received from forwarding engine with FDESCR.IPV [MFWD] equal to 5).	R/W
23	—	This bit is read as 0.	R
26:24	IPVR6[2:0]	IPV Remapping 6 Configure to which queue descriptor received with IPV 6 is stored (when a descriptor is received from forwarding engine with FDESCR.IPV [MFWD] equal to 6).	R/W
27	—	This bit is read as 0.	R
30:28	IPVR7[2:0]	IPV Remapping 7 Configure to which queue descriptor received with IPV 7 is stored (when a descriptor is received from forwarding engine with FDESCR.IPV [MFWD] equal to 7).	R/W
31	—	This bit is read as 0.	R

32.3.2.2 EATDQSC : TX Descriptor Queue Security Configuration Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0014

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	TDQSL7 to TDQSL0							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TDQSL7 to TDQSL0	TX Descriptor Queue n Security Level (n = 0 to 7) When a queue is secured, an unsecure descriptor cannot enter it (when a descriptor is from the forwarding engine with FDESCR.SEC [FWD] equal to 0). 0: Queue i unsecure 1: Queue i secure	R/W
31:8	—	These bits are read as 0.	R

32.3.2.3 EATDQC : TX Descriptor Queue Configuration Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0018

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	TDQP7 to TDQP0							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	TCTDQD	TDQD7 to TDQD0							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TDQD7 to TDQD0	TX Descriptor Queue n Disable (n = 0 to 7) 0: Queue i enabled 1: Queue i disabled	R/W
8	TCTDQD	TX Cut-Through Descriptor Queue Disable 0: Cut-through queue enabled 1: Cut-through queue disabled	R/W*1
15:9	—	These bits are read as 0.	R
23:16	TDQP7 to TDQP0*2	TX Descriptor Queue n Pause (n = 0 to 7) 0: Queue i active 1: Queue i paused	R/W*1
31:24	—	These bits are read as 0.	R

Note 1. Read value differs from written value.
Note 2. SW: Pausing a queue for a long time can result in switch overflow. In case of a switch overflow [COMA], set this register to 0.

TDQDn bits (TX Descriptor Queue n Disable (n = 0 to 7))

Avoid descriptor from MFWD to enter the corresponding queue. The forwarding engine rejects the corresponding descriptor.

If disabled queue data is no longer required, the corresponding EATMFSCq.MFS register can be set to 0 to avoid more frames going to the RMAC.

[Setting condition]

- SW: Writing 1 to this bit.

[Clearing conditions]

- SW: Writing 0 to this bit
- HW: Exiting OPERATION mode clears this register (EAMC.OPC ≠ 11).

TCTDQD bit (TX Cut-Through Descriptor Queue Disable)

Avoid descriptor from MFWD to enter the cut-through queue by deasserting eha_ct_ready pin. The forwarding engine rejects the corresponding descriptor using the Common Agent reject bus [COMA].
[Setting condition]

- SW: Writing 1 to this bit.

[Clearing conditions]

- SW: Writing 0 to this bit
- HW: Exiting OPERATION mode clears this register (EAMC.OPC ≠ 11).

TDQPn bits (TX Descriptor Queue n Pause (n = 0 to 7))

Avoid frames to be sent to the RMAC [RMAC] by stopping descriptor fetching from the descriptor RAM.
[Setting condition]

- SW: Writing 1 to this bit.

[Clearing conditions]

- SW: Writing 0 to this bit
- HW: Exiting OPERATION mode clears this register (EAMC.OPC ≠ 11).

Note:
- This register is used to stop a queue but MFWD does not stop sending descriptor to the corresponding queue. In this case the queue might overflow.
- When a queue is paused, the CBS module is still active, so the credit value continues to increase.

32.3.2.4 EATDQAC : TX Descriptor Queue Arbitration Configuration Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x001C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TDQA7[3:0]				TDQA6[3:0]				TDQA5[3:0]				TDQA4[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TDQA3[3:0]				TDQA2[3:0]				TDQA1[3:0]				TDQA0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	TDQA0[3:0]	TX Descriptor Queue Arbitration 0 0: Queue 0 strict arbitration Others: Queue 0 WRR arbitration	R/W
7:4	TDQA1[3:0]	TX Descriptor Queue Arbitration 1 0: Queue 1 strict arbitration Others: Queue 1 WRR arbitration	R/W
11:8	TDQA2[3:0]	TX Descriptor Queue Arbitration 2 0: Queue 2 strict arbitration Others: Queue 2 WRR arbitration	R/W
15:12	TDQA3[3:0]	TX Descriptor Queue Arbitration 3 0: Queue 3 strict arbitration Others: Queue 3 WRR arbitration	R/W
19:16	TDQA4[3:0]	TX Descriptor Queue Arbitration 4 0: Queue 4 strict arbitration Others: Queue 4 WRR arbitration	R/W
23:20	TDQA5[3:0]	TX Descriptor Queue Arbitration 5 0: Queue 5 strict arbitration Others: Queue 5 WRR arbitration	R/W
27:24	TDQA6[3:0]	TX Descriptor Queue Arbitration 6 0: Queue 6 strict arbitration Others: Queue 6 WRR arbitration	R/W
31:28	TDQA7[3:0]	TX Descriptor Queue Arbitration 7 0: Queue 7 strict arbitration Others: Queue 7 WRR arbitration	R/W

Note:
- For more details, see GWRDQAC register description [GWCA] in the GWCA specification
- SW: In hybrid arbitration mode, all the queues with an EATDQAC.TDQAi value other than 0 should have a consecutive priority

- SW: In all arbitration mode except strict priority arbitration mode, at least two queues should have an EATDQAC.TDQAi value other than 0
- SW: WRR arbitration is only possible between queues with the same type (e or p), see EATPEC.TTQ register description (for example, all queues with an EATDQAC.TDQAi value other than 0 should have the same setting in EATPEC.TTQi register)
- SW: Only queues set to strict arbitration can use CBS functions
- SW: When using TAS, all queues not set to strict arbitration should be opened simultaneously.

32.3.2.5 EATPEC : TX Preemption Configuration Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0020

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	TTQ7 to TTQ0							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TTQ7 to TTQ0	Transmission Type Queue n (n = 0 to 7) See section 32.5.1.7.2. Preemption. 0: Queue i contains e-frames 1: Queue i contains p-frames	R/W
15:8	—	These bits are read as 0.	R
17:16	AFS[1:0]	Additional Fragment Size Used for preemption. It defines the minimum size of a fragment which is not the last fragment of a frame. 0 0: 0 byte is added to minimum fragment size (minimum fragment size = 64 bytes) 0 1: 64 bytes are added to minimum fragment size (minimum fragment size = 128 bytes) 1 0: 128 bytes are added to minimum fragment size (minimum fragment size = 192 bytes) 1 1: 192 bytes are added to minimum fragment size (minimum fragment size = 256 bytes)	R/W
31:18	—	These bits are read as 0.	R

32.3.2.6 EATMFSCq : Transmission Maximum Frame Size Configuration Register q (q = 0 to 7)

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0040 + 0x4 × q

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

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

Bit	Symbol	Function	R/W
15:0	MFS[15:0]	Maximum Frame Size Maximum frame size for descriptor queue q. All bigger frames are rejected.	R/W
31:16	—	These bits are read as 0.	R

Note: HW: The MFS size comparison takes the frame size on the PHY interface.

32.3.2.7 EATDQDCq : Transmission Descriptor Queue Depth Configuration Register q (q = 0 to 7)

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0060 + 0x4 × q

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

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

Bit	Symbol	Function	R/W
9:0	DQD[9:0]	Descriptor Queue Depth Set the number of descriptors that can contain descriptor queue q. For details, see GWRDQDCq register description [GWCA] in the GWCA specification.	R/W
31:10	—	These bits are read as 0.	R

Note: SW: The sum of DQD fields should always be smaller or equal to 512.

32.3.2.8 EATDQMq : Transmission Descriptor Queue q Monitoring Register (q = 0 to 7)

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0080 + 0x4 × q

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

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

Bit	Symbol	Function	R/W
9:0	DNQ[9:0]	Descriptor Number in Queue These bits indicate the current number of descriptors stored in TX descriptor queue q. All descriptors stored in the TX descriptor queue q are rejected/dropped when exiting OPERATION mode.	R
31:10	—	These bits are read as 0.	R

DNQ[9:0] bits (Descriptor Number in Queue)

[Increment condition]

- HW: Incremented by 1 when a descriptor is received from the MFWD for the corresponding queue, the queue is not full and there is no descriptor security error.

[Decrement condition]

- HW: Decremented by 1 when a descriptor is read by Ethernet Agent to send data to RMAC [RMAC] or to reject data.

32.3.2.9 EATDQMLMq : Transmission Descriptor Queue q Max Level Monitoring Register (q = 0 to 7)

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x00A0 + 0x4 × q

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

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

Bit	Symbol	Function	R/W
9:0	DMLQ[9:0]	Descriptor Max Level in Queue These bits indicate the maximum number of descriptors that is stored in TX descriptor queue q.	R
31:10	—	These bits are read as 0.	R

DMLQ[9:0] bits (Descriptor Max Level in Queue)

[Clearing conditions]

- HW: RESET mode clears this register (EAMC.OPS == 00)
- SW: Reading this register clears to EATDQMq.DNQ.

[Increment condition]

- HW: Increments to EATDQMq.DNQ value when smaller than EATDQMq.DNQ.

32.3.2.10 EACTQC : Cut-Through Queue Configuration Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0100

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

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

Bit	Symbol	Function	R/W
15:0	CTQD[15:0]*1	Cut-Through Queue Delay (clk Cycle Number) Set the number of clocks to wait for descriptor transmission.	R/W

Bit	Symbol	Function	R/W
31:16	—	These bits are read as 0.	R

Note 1. Set to 0 for this product.

32.3.2.11 EACTDQDC : Cut-Through Descriptor Queue Depth Configuration Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0104

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

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

Bit	Symbol	Function	R/W
3:0	CTDQD[3:0]	Cut-Through Descriptor Queue Depth Number of descriptors that can contain cut-through descriptor queue.	R/W
31:4	—	These bits are read as 0.	R

Note: SW: The maximum value of this register is 8.

32.3.2.12 EACTDQM : Cut-Through Descriptor Queue Monitoring Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0108

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

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

Bit	Symbol	Function	R/W
3:0	CTQDN[3:0]	Cut-Though Queue Descriptor Number	R
31:4	—	These bits are read as 0.	R

CTQDN[3:0] bits (Cut-Though Queue Descriptor Number)

These bits indicate the current number of descriptors stored in the cut-through descriptor queue.

All cut-through descriptors stored in the queue are not rejected/dropped when exiting OPERATION mode. But the frame being received by ETHA is rejected/dropped, if ETHA transitioned to reset before the frame is completely received.

All cut-through descriptors stored in the queue are not rejected/dropped when exiting OPERATION mode but are rejected/dropped when transitioning to RESET mode.

[Increment condition]

- HW: Incremented by 1 when a descriptor is received from the forwarding engine for the cut-through queue and the queue is not full.

[Decrement condition]

- HW: Decremented by 1 when a descriptor is read by Ethernet Agent to send data to RMAC [RMAC].

32.3.2.13 EACTDQMLM : Cut-Through Descriptor Queue Max Level Monitoring Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x010C

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

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

Bit	Symbol	Function	R/W
3:0	CTDMLQ[3:0]	Cut-Through Descriptor Max Level in Queue These bits indicate the maximum number of descriptors that is stored in cut-through descriptor queue.	R
31:4	—	These bits are read as 0.	R

CTDMLQ[3:0] bits (Cut-Through Descriptor Max Level in Queue)

[Clearing conditions]

- HW: RESET mode clears this register (EAMC.OPS == 00)
- SW: Reading this register clears to EACTDQM.CTQDN.

[Increment condition]

- HW: Increments to EACTDQM.CTQDN value when smaller than EACTDQM.CTQDN.

32.3.3 TAG Function Registers

32.3.3.1 EAVCC : VLAN Control Configuration Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0130

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

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

Bit	Symbol	Function	R/W
0	VIM	VLAN Ingress Mode 0: Incoming VLAN mode 1: Port based VLAN mode	R/W
15:1	—	These bits are read as 0.	R
18:16	VEM[2:0]	VLAN Egress Mode 0 0 0: No VLAN mode 0 0 1: C-TAG VLAN mode 0 1 0: HW C-TAG VLAN mode 0 1 1: SC-TAG VLAN mode 1 0 0: HW SC-TAG VLAN mode	R/W
31:19	—	These bits are read as 0.	R

32.3.3.2 EAVTC : VLAN TAG Configuration Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0134

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	STD	STP[2:0]			STV[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
11:0	CTV[11:0]	C-TAG VLAN Set the C-TAG VID values.	R/W
14:12	CTP[2:0]	C-TAG PCP Set the C-TAG PCP values.	R/W
15	CTD	C-TAG DEI Set the C-TAG DEI values.	R/W
27:16	STV[11:0]	S-TAG VLAN Set the S-TAG VID values.	R/W
30:28	STP[2:0]	S-TAG PCP Set the S-TAG PCP values.	R/W
31	STD	S-TAG DEI Set the S-TAG DEI values.	R/W

32.3.3.3 EARTFC : Reception TAG Filtering Configuration Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0138

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	UT	SCRT	SCT	CRT	CT	CSRT	CST	RT	NT
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NT	No TAG 0: No TAG frame passed 1: No TAG frame rejected	R/W
1	RT	R-TAG 0: R-TAG frame passed 1: R-TAG frame rejected	R/W
2	CST	CoS-TAG 0: CoS-TAG frame passed 1: CoS-TAG frame rejected	R/W
3	CSRT	CoSR-TAG 0: CoSR-TAG frame passed 1: CoSR-TAG frame rejected	R/W
4	CT	C-TAG 0: C-TAG frame passed 1: C-TAG frame rejected	R/W
5	CRT	CR-TAG 0: CR-TAG frame passed 1: CR-TAG frame rejected	R/W
6	SCT	SC-TAG 0: SC-TAG frame passed 1: SC-TAG frame rejected	R/W
7	SCRT	SCR-TAG 0: SCR-TAG frame passed 1: SCR-TAG frame rejected	R/W
8	UT	Unknown TAG 0: Unknown TAG frame passed 1: Unknown TAG frame rejected	R/W
31:9	—	These bits are read as 0.	R

32.3.4 CBS Function Registers [802.1Qav]

32.3.4.1 EACAEC : CBS Admin Enable Configuration Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0200

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	CE7 to CE0							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CE7 to CE0	CBS Enable n (n = 0 to 7) This register is only used to configure CBS and is not directly used by CBS modules. 0: CBS for descriptor queue i disabled 1: CBS for descriptor queue i enabled	R/W
31:8	—	These bits are read as 0.	R

32.3.4.2 EACC : CBS Configuration Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0204

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	CC7 to CC0							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CC7 to CC0	Configuration Change n (n = 0 to 7) 0: No effect 1: The Admin settings of CBS i are copied to its Oper settings (EACAEC.CEi, EACAIVCq.CIV, and EACAULCq.CUL are respectively copied to EACOEM.CEi, EACOIVMq.CIV, and EACOULMq.CUL)	W*1
31:8	—	These bits are read as 0.	R

Note 1. Read value differs from written value.

32.3.4.3 EACAIVCq : CBS Admin Increment Value Configuration Register q (q = 0 to 7)

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0220 + 0x4 × q

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CIV[19:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
19:0	CIV[19:0]	Credit Increment Value	R/W
31:20	—	These bits are read as 0.	R

CIV bits (Credit Increment Value)

Configure CBS throughput.

CIV[3:0] is the credit increment in byte per clock and CIV[15:0] is the credit increment in sub-byte per clock.

This register is only used to configure CBS and is not directly used by CBS modules.

For more details, see section 32.5.1.5. CBS (Credit-based shaper) [802.1Qav].

Caution: SW: Setting this register to 0 or a value close to 0 sets the corresponding transmission queue to a very low throughput and can lead to a queue overflow.

32.3.4.4 EACAULCq : CBS Admin Upper Limit Configuration Register q (q = 0 to 7)

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0240 + 0x4 × q

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CUL[30:0]															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
30:0	CUL[30:0]	Credit Upper Limit	R/W
31	—	This bit is read as 0.	R

CUL bits (Credit Upper Limit)

Configure queue q interference time.

This register is only used to configure CBS and is not directly used by CBS modules.

For more details, see section 32.5.1.5. CBS (Credit-based shaper) [802.1Qav].

32.3.4.5 EACOEM : CBS Oper Enable Monitoring Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0260

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	CE7 to CE0							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CE7 to CE0	CBS Enable n (n = 0 to 7) [Update conditions] HW: This register is updated to EACACC.CEi when 1 is written to EACC.CCi. 0: CBS for descriptor queue i disabled 1: CBS for descriptor queue i enabled	R
31:8	—	These bits are read as 0.	R

32.3.4.6 EACOIVMq : CBS Oper Increment Value Monitoring Register q (q = 0 to 7)

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0280 + 0x4 × q

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CIV[19:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
19:0	CIV[19:0]	Credit Increment Value	R
31:20	—	These bits are read as 0.	R

CIV bits (Credit Increment Value)

Configure CBS throughput.

CIV[3:0] is the credit increment in byte per clock and CIV[15:0] is the credit increment in sub-byte per clock.

This register is only used to configure CBS and is not directly used by CBS modules.

[Update condition]

- HW: This register is updated to EACAIVq.CIV when 1 is written to EACC.CCi.

32.3.4.7 EACOULMq : CBS Oper Upper Limit Monitoring Register q (q = 0 to 7)

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x02A0 + 0x4 × q

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CUL[30:0]															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
30:0	CUL[30:0]	Credit Upper Limit	R
31	—	This bit is read as 0.	R

CUL bits (Credit Upper Limit)

Configure queue q interference time.

This register is only used to configure CBS and is not directly used by CBS modules.

[Update condition]

- HW: This register is updated to EACAULq.CUL when 1 is written to EACC.CCi.

32.3.4.8 EACGSM : CBS Gate State Monitoring Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x02C0

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	CGS7 to CGS0							
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	CGS7 to CGS0	CBS Gate State n (n = 0 to 7) 0: CBS does not authorize queue i transmission (CBS credit negative) 1: CBS authorizes queue i transmission (CBS credit positive)	R
31:8	—	These bits are read as 0.	R

32.3.5 TAS Function Registers [802.1Qbv]

32.3.5.1 EATASC : TAS Configuration Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0300

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	TASCA[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	TASE	TAS Enable Enables the TAS schedule See section 32.5.1.6.1. TAS General Behavior. 0: TAS disabled 1: TAS enabled	R/W*1
1	TASCC	TAS Configuration Change 0: TAS is not changing configuration 1: TAS is changing configuration	R/W*1
2	TASCI	TAS Configuration Impossible This bit aids the software in detecting when TAS configuration is possible, it is equal to (EATASC.TASE & ~ EATASSM.TASSO) | EATASC.TASCC. 0: TAS configuration is possible 1: TAS configuration is not possible	R
7:3	—	These bits are read as 0.	R
8	TASTS	TAS Timer Select Selects the gPTP to be used for TAS module.	R/W
15:9	—	These bits are read as 0.	R
23:16	TASCA[7:0]	TAS Configuration Address Shows the address from which entry learning occurs in TAS RAM for the next configuration.	R
31:24	—	These bits are read as 0.	R

Note 1. Read value differs from written value.

TASE bit (TAS Enable)

[Setting condition]

- SW: Writing 1 to this bit.

[Clearing conditions]

- SW: Writing 0 to this bit
- HW: When in RESET mode.

TASCC bit (TAS Configuration Change)

[Setting condition]

- SW: Writing 1 to this bit.

[Clearing conditions]

- HW: When TAS configuration change is complete

- HW: When in RESET mode.

See section 32.5.1.6.1. TAS General Behavior.

TASCI bit (TAS Configuration Impossible)

[Clearing condition]

- HW: When in RESET mode.

TASCA[7:0] bits (TAS Configuration Address)

[Clearing condition]

- HW: When in RESET mode.

32.3.5.2 EATASIGSC : TAS Initial Gate State Configuration Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0304

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	TASCTIGS	TASIGS7 to TASIGS0							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TASIGS7 to TASIGS0	TAS Initial Gate State n (n = 0 to 7) Bit q is the gate state used for initialization of descriptor queue q gate. 0: Initial gate state is closed 1: Initial gate state is opened	R/W
8	TASCTIGS	TAS Cut-Through Initial Gate State Gate state used for initialization of cut-through gate. 0: Initial gate state is closed 1: Initial gate state is opened	R/W
31:9	—	These bits are read as 0.	R

32.3.5.3 EATASENCi : TAS Entry Number Configuration Register i (i = 0 to 7)

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0320 + 0x4 × i

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

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

Bit	Symbol	Function	R/W
8:0	TASAEN[8:0]	TAS Admin Entry Number This register is only used to configure TAS and is not directly used by TAS module. Sets the number of TAS RAM entries used for gate i schedule.	R/W
31:9	—	These bits are read as 0.	R

Note:
- SW: If the TAS module is dynamically used (TAS setting flow in Figure 32.11 is used when EATASC.TASE is already set), the sum of EATASENCi.TASAEN should be smaller or equal to 119.
- SW: If the TAS module is statistically used (TAS setting flow in Figure 32.11 is not used when EATASC.TASE is already set), the sum of EATASENCi.TASAEN should be smaller or equal to 247.

32.3.5.4 EATASCTENC : TAS Cut-Through Entry Number Configuration Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0340

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

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

Bit	Symbol	Function	R/W
8:0	TASCTAEN[8:0]	TAS Admin Cut-Through Entry Number This register is only used to configure TAS and is not directly used by TAS module. Sets the number of TAS RAM entries used for cut-through gate schedule.	R/W
31:9	—	These bits are read as 0.	R

Note:
- SW: If the TAS module is dynamically used (TAS setting flow in Figure 32.11 is used when EATASC.TASE is already set), the sum of EATASENCi.TASAEN and EATASCTENC.TASCTAEN should be smaller or equal to 119.
- SW: If the TAS module is statistically used (TAS setting flow in Figure 32.11 is not used when EATASC.TASE is already set), the sum of EATASENCi.TASAEN and EATASCTENC.TASCTAEN should be smaller or equal to 247.

32.3.5.5 EATASENMi : TAS Entry Number Monitoring Register i (i = 0 to 7)

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0360 + 0x4 × i

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

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

Bit	Symbol	Function	R/W
8:0	TASOEN[8:0]	TAS Oper Entry Number	R
31:9	—	These bits are read as 0.	R

TASOEN[8:0] bits (TAS Oper Entry Number)

[Update condition]

- HW: This register is updated to EATASENCi.TASAEN when configuration change occurs or when schedule start occurs. See section 32.5.1.6.1. TAS General Behavior.

[Clearing condition]

- HW: When in RESET mode.

32.3.5.6 EATASCTENM : TAS Cut-Through Entry Number Monitoring Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0380

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

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

Bit	Symbol	Function	R/W
8:0	TASCTOEN[8:0]	TAS Cut-Through Oper Entry Number	R
31:9	—	These bits are read as 0.	R

TASCTOEN[8:0] bits (TAS Cut-Through Oper Entry Number)

[Update condition]

- HW: This register is updated to EATASCTENC.TASCTAEN when configuration change occurs or when schedule start occurs. See section 8.1.6.1.

[Clearing condition]

- HW: When in RESET mode.

32.3.5.7 EATASCSTC0 : TAS Cycle Start Time Configuration Register 0

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x03A0

Bit position:	31 … 0
Bit field:	TASACSTP0[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TASACSTP0[31:0]	TAS Admin Cycle Start Time Part 0 Time at which TAS scheduler should start/change configuration. This register is in ns.	R/W

32.3.5.8 EATASCSTC1 : TAS Cycle Start Time Configuration Register 1

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x03A4

Bit position: 31 … 0

Bit field: TASACSTP1[31:0]

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TASACSTP1[31:0]	TAS Admin Cycle Start Time Part 1 Time at which TAS scheduler should start/change configuration. This register is in 2^{32} × ns.	R/W

32.3.5.9 EATASCSTM0 : TAS Cycle Start Time Monitoring Register 0

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x03A8

Bit position: 31 … 0

Bit field: TASOCSTP0[31:0]

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TASOCSTP0[31:0]	TAS Oper Cycle Start Time Part 0	R

TASOCSTP0[31:0] bits (TAS Oper Cycle Start Time Part 0)

[Update conditions]

- HW: {EATASCSTM1, EATASCSTM0} is updated to {EATASCSTC1, EATASCSTC0} + EATASCTC when configuration change occurs or when schedule start occurs. See section 32.5.1.6.1. TAS General Behavior.
- HW: This register is updated to the next cycle start time every time a new cycle starts.

[Clearing condition]

- HW: When in RESET mode.

32.3.5.10 EATASCSTM1 : TAS Cycle Start Time Monitoring Register 1

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x03AC

Bit position: 31 … 0

Bit field: TASOCSTP1[31:0]

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TASOCSTP1[31:0]	TAS Oper Cycle Start Time Part 1	R

TASOCSTP1[31:0] bits (TAS Oper Cycle Start Time Part 1)

[Update conditions]

- HW: {EATASCSTM1, EATASCSTM0} is updated to the value of {EATASCSTC1, EATASCSTC0} + EATASCTC when configuration change occurs or when schedule start occurs. See section 32.5.1.6.1. TAS General Behavior.
- HW: This register is updated to the next cycle start time every time a new cycle starts.

[Clearing condition]

- HW: When in RESET mode.

32.3.5.11 EATASCTC : TAS Cycle Time Configuration Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x03B0

Bit position:	31–0
Bit field:	TASACT[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TASACT[31:0]	TAS Admin Cycle Time Configure the cycle time for TAS in ns.	R/W

Note: SW: Set this register to a value greater than 100 ns.

32.3.5.12 EATASCTM : TAS Cycle Time Monitoring Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x03B4

Bit position:	31–0
Bit field:	TASOCT[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TASOCT[31:0]	TAS Oper Cycle Time	R

TASOCT[31:0] bits (TAS Oper Cycle Time)

[Update condition]

- HW: This register is updated to EATASCTC.TASACT when configuration change occurs or when schedule start occurs. See section 32.5.1.6.1. TAS General Behavior.

[Clearing condition]

- HW: When in RESET mode.

32.3.5.13 EATASGL0 : TAS Gate Learn Register 0

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x03C0

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

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

Bit	Symbol	Function	R/W
7:0	TASGAL[7:0]	TAS Gate Address Learn Configures the address in which the TAS entry is learned.	R/W
31:8	—	These bits are read as 0.	R

32.3.5.14 EATASGL1 : TAS Gate Learn Register 1

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x03C4

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TASGTL[27:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
27:0	TASGTL[27:0]	TAS Gate Time Learn Configures gate time	R/W
28	TASGSL	TAS Gate State Learn Configures gate state	R/W
31:29	—	These bits are read as 0.	R

32.3.5.15 EATASGLR : TAS Gate Learn Result Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x03C8

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

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

Bit	Symbol	Function	R/W
30:0	—	These bits are read as 0.	R
31	GL	Gate Learn	R

GL bit (Gate Learn)

[Setting condition]

- HW: Writing to the EATASGL1 register.

[Clearing condition]

- HW: When learning is completed, this bit is deasserted.

32.3.5.16 EATASGR : TAS Gate Read Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x03D0

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

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

Bit	Symbol	Function	R/W
7:0	TASGAR[7:0]	TAS Gate Address Read Configures the address in which the TAS entry is read.	R/W
31:8	—	These bits are read as 0.	R

32.3.5.17 EATASGRR : TAS Gate Read Result Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x03D4

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TASGTR[27:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
27:0	TASGTR[27:0]	TAS Gate Time Read Displays gate time read value.	R
28	TASGSR	TAS Gate State Read Displays gate state read value.	R
30:29	—	These bits are read as 0.	R
31	GR	Gate Read	R

TASGTR bits (TAS Gate Time Read)

[Update condition]

- EATASGRR.GR clear event

TASGSR bit (TAS Gate State Read)

[Update condition]

- EATASGRR.GR clear event

GR bit (Gate Read)

[Setting condition]

- HW: Writing to the EATASGR register.

[Clearing condition]

- HW: When reading is completed, this bit is deasserted.

32.3.5.18 EATASHCC : TAS Hardware Calibration Configuration Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x03E0

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

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

Bit	Symbol	Function	R/W
15:0	TASJ[15:0]	TAS Jitter Configure the jitter between the TAS data transmission and RMAC PHY interface [RMAC]. See section 32.5.1.6.3. TAS Hardware Calibration.	R/W
31:16	—	These bits are read as 0.	R

32.3.5.19 EATASRIRM : TAS RAM Initialization Register Monitoring Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x03E4

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

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

Bit	Symbol	Function	R/W
0	TASRIOG	TAS RAM Initialization Ongoing	R/W*1
1	TASRR	TAS RAM Ready	R
31:2	—	These bits are read as 0.	R

Note 1. Read value differs from written value.

TASRIOG bit (TAS RAM Initialization Ongoing)

[Setting condition]

- SW: Writing 1 to this register starts TAS RAM initialization.

[Clearing condition]

- HW: When TAS RAM initialization is completed.

TASRR bit (TAS RAM Ready)

[Setting condition]

- When EATASRIRM.TASRIOG is cleared.

[Clearing condition]

- Writing 1 to EATASRIRM.TASRIOG.

32.3.5.20 EATASSM : TAS Status Monitoring Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x03E8

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	TASCTGS	TASGS7 to TASGS0							
Value after reset:	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	TASGS7 to TASGS0	TAS Gate n State (n = 0 to 7) 0: Gate i is closed 1: Gate i is opened	R
8	TASCTGS*1	TAS Cut-Through Gate State 0: Cut-through gate is closed 1: Cut-through gate is opened	R
15:9	—	These bits are read as 0.	R
16	TASSO	TAS Scheduler Ongoing 0: TAS scheduler is not ongoing 1: TAS scheduler is ongoing	R
31:17	—	These bits are read as 0.	R

Note 1. HW: This register only exists if UCIAPRACE020_CT_ON is defined.

TASSO bit (TAS Scheduler Ongoing)

[Clearing condition]

- HW: When in RESET mode.

32.3.6 ETHA Counter Registers

32.3.6.1 EAUSMFSECN : Under Minimum Frame Size Error Counter Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0400

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

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

Bit	Symbol	Function	R/W
15:0	USMFSEN[15:0]	Under Minimum Frame Size Error Number This register counts the number of received data lost because of under minimum size error.	R

Bit	Symbol	Function	R/W
31:16	—	These bits are read as 0.	R

USMFSEN[15:0] bits (Under Minimum Frame Size Error Number)

[Clearing conditions]

- HW: When in RESET mode
- SW: Reading this register.

[Increment condition]

- HW: Incremented by 1 when an Under Minimum Frame Size Error occurs and if this register has a value other than 0xFFFF.

32.3.6.2 EATFECN : TAG Filtering Error Counter Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0404

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

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

Bit	Symbol	Function	R/W
15:0	TFEN[15:0]	TAG Filtering Error Number This register counts the number of received data lost because of TAG filtering.	R
31:16	—	These bits are read as 0.	R

TFEN[15:0] bits (TAG Filtering Error Number)

[Clearing conditions]

- HW: When in RESET mode
- SW: Reading this register.

[Increment condition]

- HW: Incremented by 1 when a TAG filter error occurs and if this register has a value other than 0xFFFF.

32.3.6.3 EAFSECN : Frame Size Error Counter Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0408

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

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

Bit	Symbol	Function	R/W
15:0	FSEN[15:0]	Frame Size Error Number This register counts the number of received data lost because of a Frame Size Error.	R
31:16	—	These bits are read as 0.	R

FSEN[15:0] bits (Frame Size Error Number)

[Clearing conditions]

- HW: When in RESET mode
- SW: Reading this register.

[Increment condition]

- HW: Incremented by 1 when a Frame Size Error occurs and if this register has a value other than 0xFFFF.

32.3.6.4 EADQOECN : Descriptor Queue Overflow Error Counter Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x040C

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

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

Bit	Symbol	Function	R/W
15:0	DQOEN[15:0]	Descriptor Queue Overflow Error Number This register counts the number of transmit data lost because of a Descriptor Queue Overflow Error.	R
31:16	—	These bits are read as 0.	R

DQOEN[15:0] bits (Descriptor Queue Overflow Error Number)

[Clearing conditions]

- HW: When in RESET mode
- SW: Reading this register.

[Increment condition]

- HW: Incremented by 1 when a Descriptor Queue Overflow Error occurs and if this register has a value other than 0xFFFF.

32.3.6.5 EADQSECN : Descriptor Queue Security Error Counter Register

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0410

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

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

Bit	Symbol	Function	R/W
15:0	DQSEN[15:0]	Descriptor Queue Security Error Number This register counts the number of transmit data lost because of a Descriptor Queue Security Error.	R
31:16	—	These bits are read as 0.	R

DQSEN[15:0] bits (Descriptor Queue Security Error Number)

[Clearing conditions]

- HW: When in RESET mode
- SW: Reading this register.

[Increment condition]

- HW: Incremented by 1 when a Descriptor Queue Security Error occurs and if this register has a value other than 0xFFFF.

32.3.7 ETHA Interrupt Registers

32.3.7.1 Error Interrupt Registers

32.3.7.1.1 EAEIS0 : Error Interrupt Status Register 0

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0500

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FSES7 to FSES0								—	TFES	USMFSES	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	—	These bits are read as 0.	R
5	USMFSES	Under Switch Minimum Frame Size Error Status Flag	R/W*1
6	TFES	TAG Filtering Error Status Flag	R/W*1
7	—	This bit is read as 0.	R
15:8	FSES7 to FSES0	Frame Size Error Status Flag n (n = 0 to 7)	R/W*1
31:16	—	These bits are read as 0.	R

Note 1. Read value differs from written value.

USMFSES bit (Under Switch Minimum Frame Size Error Status Flag)

[Setting condition]

- HW: A frame smaller than 32 bytes has been received from RMAC.

This error occurs with [RMAC] MEIS.RPOES.

[Clearing conditions]

- HW: When in RESET mode
- SW: Writing 1 to one of these bits.

[Error recovery]

- HW: The frame is discarded
- SW: System-dependent, therefore cannot be defined.

TFES bit (TAG Filtering Error Status Flag)

[Setting condition]

- HW: An unauthorized TAG format has been detected. (See section 32.5.2.1. RX Data Store).

[Clearing conditions]

- HW: When in RESET mode
- SW: Writing 1 to one of these bits.

[Error recovery]

- HW: Frame is sent to MFWD with local descriptor DESCR.TFE bit set. (See section 32.5.2.1. RX Data Store).
- SW: System-dependent, therefore cannot be defined.

FSESn bits (Frame Size Error Status Flag n (n = 0 to 7))

[Setting condition]

- HW: Bit q of this register is set when a frame bigger than EATMFSC.MFS has been received for descriptor queue q.

[Clearing conditions]

- HW: When in RESET mode
- SW: Writing 1 to one of these bits.

[Error recovery]

- HW: The current frame is lost and the following data is processed normally
- SW: System-dependent, therefore it cannot be defined.

32.3.7.1.2 EAEIE0 : Error Interrupt Enable Register 0

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0504

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FSEE7 to FSEE0								—	TFEE	USMFSEE	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	—	These bits are read as 0.	R
5	USMFSEE	Under Switch Minimum Frame Size Error Enable 0: Interrupt disabled 1: Interrupt enabled	R/W*1
6	TFEE	TAG Filtering Error Enable 0: Interrupt disabled 1: Interrupt enabled	R/W*1
7	—	This bit is read as 0.	R
15:8	FSEE7 to FSEE0	Frame Size Error Enable n (n = 0 to 7) 0: Interrupt disabled for descriptor queue i 1: Interrupt enabled for descriptor queue i	R/W*1
31:16	—	These bits are read as 0.	R

Note 1. Read value differs from written value.

USMFSEE bit (Under Switch Minimum Frame Size Error Enable)

[Setting condition]

- Writing 1 to this bit.

[Clearing condition]

- Writing 1 to EAEID0.USMFSED register clears this bit.

TFEE bit (TAG Filtering Error Enable)

[Setting condition]

- Writing 1 to this bit.

[Clearing condition]

- Writing 1 to EAEID0.TFED register clears this bit.

FSEEn bits (Frame Size Error Enable n (n = 0 to 7))

[Setting condition]

- Writing 1 to one of these bits.

[Clearing condition]

- Writing 1 to one of EAEID0.FSED bits clears the corresponding bit in this register.

32.3.7.1.3 EAEID0 : Error Interrupt Disable Register 0

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0508

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FSED7 to FSED0								—	TFED	USMFSED	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	—	These bits are read as 0.	R
5	USMFSED	Under Switch Minimum Frame Size Error Disable Writing 1 to this bit clears EAEIE0.USMFSEE register.	R
6	TFED	TAG Filtering Error Disable Writing 1 to this bit clears EAEIE0.TFEE register.	R
7	—	This bit is read as 0.	R
15:8	FSED7 to FSED0	Frame Size Error Disable n (n = 0 to 7) Writing 1 to one of these bits clears the corresponding bit in EAEIE0.FSEE register.	R
31:16	—	These bits are read as 0.	R

32.3.7.1.4 EAEIS1 : Error Interrupt Status Register 1

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0510

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	TASCTGES	TASGES7 to TASGES0							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	CULES7 to CULES0							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CULES7 to CULES0	CBS n Upper Limit Error Status Flag (n = 0 to 7)	R/W*1
15:8	—	These bits are read as 0.	R
23:16	TASGES7 to TASGES0	TAS Gate n Error Status Flag (n = 0 to 7)	R/W*1*2
24	TASCTGES	TAS Cut-Through Gate Error Status Flag	R/W*1*2
31:25	—	These bits are read as 0.	R

Note 1. Read value differs from written value.
Note 2. This error can occur for the following reasons:

- TAS start time EATASCSTC0/1 is set in the past
- gPTP timer had an offset correction [gPTP]
- The minimum gate time has not been satisfied.

CULESn bits (CBS n Upper Limit Error Status Flag (n = 0 to 7))

[Setting condition]

- HW: Bit q of this register is set when descriptor queue q corresponding CBS credits has reached its upper limit EACOULMq.CUL.

[Clearing conditions]

- HW: When in RESET mode
- SW: Writing 1 to one of these bits.

[Error recovery]

- HW: Credit calculation stops at EACOULMq.CUL and no further credit are added until frame transmission for the corresponding queue. As a result, the queue throughput is lower than expected.
- SW: This is a software error and should not occur. (Setting TAS with gates that are partially overlapping each other can trigger this error).

TASGESn bits (TAS Gate n Error Status Flag (n = 0 to 7))

[Setting condition]

- HW: Bit q of this register is set when descriptor queue q corresponding TAS gate had no time to fetch the next gate state value until it starts.

[Clearing conditions]

- HW: When in RESET mode
- SW: Writing 1 to one of these bits.

[Error recovery]

- HW: Gate with restart during next cycle
- SW: If this error occurs frequently, TAS should be reconfigured with a different schedule (this is a software issue because the minimum gate time has not been satisfied). During reconfiguration, this error continues to occur until the new start time is reached and so, it should be ignored.

Caution: This error can occur for the following reasons:

- **TAS start time EATASCSTC0/1 is set in the past**
- **gPTP timer had an offset correction [gPTP]**
- **The minimum gate time has not been satisfied (a gate time is smaller than 50 ns + EATASHCC.TASJ).**

TASCTGES bit (TAS Cut-Through Gate Error Status Flag)

[Setting condition]

- HW: This register is set when cut-through corresponding TAS gate has no time to fetch the next gate state value until it starts.

[Clearing conditions]

- HW: When in RESET mode
- SW: Writing 1 to one of these bits.

[Error recovery]

- HW: Gate with restart during next cycle
- SW: If this error occurs frequently, TAS should be reconfigured with a different schedule (this is a software issue because the minimum gate time has not been satisfied). During reconfiguration, this error continues to occur until the new start time is reached and so, it should be ignored.

Caution: This error can occur for the following reasons:

- **TAS start time EATASCSTC0/1 is set in the past**

- **gPTP timer had an offset correction [gPTP]**
- **The minimum gate time has not been satisfied.**

32.3.7.1.5 EAEIE1 : Error Interrupt Enable Register 1

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0514

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	TASCTGEE	TASGEE7 to TASGEE0							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	CULEE7 to CULEE0							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CULEE7 to CULEE0	CBS n Upper Limit Error Enable (n = 0 to 7) 0: Interrupt disabled for descriptor queue i 1: Interrupt enabled for descriptor queue i	R/W*1
15:8	—	These bits are read as 0.	R
23:16	TASGEE7 to TASGEE0	TAS Gate n Error Enable (n = 0 to 7) 0: Interrupt disabled for descriptor queue i 1: Interrupt enabled for descriptor queue i	R/W*1
24	TASCTGEE	TAS Cut-through Gate Error Enable 0: Interrupt disabled 1: Interrupt enabled	R/W*1
31:25	—	These bits are read as 0.	R

Note 1. Read value differs from written value.

CULEEn bits (CBS n Upper Limit Error Enable (n = 0 to 7))

[Setting condition]

- Writing 1 to one of these bits.

[Clearing condition]

- Writing 1 to one of EAEID1.CULED bits clears the corresponding bit in this register.

TASEEn bits (TAS Gate n Error Enable (n = 0 to 7))

[Setting condition]

- Writing 1 to one of these bits.

[Clearing condition]

- Writing 1 to one of EAEID1.TASGED bits clears the corresponding bit in this register.

TASCTGEE bit (TAS Cut-through Gate Error Enable)

[Setting condition]

- Writing 1 to this bit.

[Clearing condition]

- Writing 1 to EAEID1.TASCTGED register.

32.3.7.1.6 EAEID1 : Error Interrupt Disable Register 1

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0518

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	TASCTGED	TASGED7 to TASGED0							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	CULED7 to CULED0							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CULED7 to CULED0	CBS n Upper Limit Error Disable (n = 0 to 7) Writing 1 to one of these bits clears the corresponding bit in EAEIE1.CULEE register.	R
15:8	—	These bits are read as 0.	R
23:16	TASGED7 to TASGED0	TAS Gate n Error Disable (n = 0 to 7) Writing 1 to one of these bits clears the corresponding bit in EAEIE1.TASGEE register.	R
24	TASCTGED	TAS Cut-through Gate Error Disable Writing 1 to this bit clears EAEIE1.TASCTGEE register.	R
31:25	—	These bits are read as 0.	R

32.3.7.1.7 EAEIS2 : Error Interrupt Status Register 2

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0520

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	DQSES7 to DQSES0							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CTDQOES	DQOES7 to DQOES0							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	DQOES7 to DQOES0	Descriptor Queue n Overflow Error Status Flag (n = 0 to 7)	R/W*1
8	CTDQOES	Cut-Through Descriptor Queue Overflow Error Status Flag	R/W*1
15:9	—	These bits are read as 0.	R
23:16	DQSES7 to DQSES0	Descriptor Queue n Security Error Status Flag (n = 0 to 7)	R/W*1
31:24	—	These bits are read as 0.	R

Note 1. Read value differs from written value.

DQOESn bits (Descriptor Queue n Overflow Error Status Flag (n = 0 to 7))

[Setting condition]

- HW: Bit q of this register is set.

[ETHA is operating] (EAMC.OPC == 11 and EAMC.OPS == 11)

Descriptor queue is "full (EATDQDCq.DQD == EATDQMq.DNQ)" or "not disabled (EATDQC.TDQDn is not set)" when a descriptor is received for queue q.

[ETHA is not operating] (EAMC.OPC != 11 and EAMC.OPS == 11)

When a descriptor is received for queue q.

[Clearing conditions]

- HW: When in RESET mode
- SW: Writing 1 to this bit.

[Error recovery]

- HW: Any descriptor received for a full descriptor queue is not accepted by ETHA and is not forwarded to RMAC [RMAC]
- SW: Too many descriptors are received for the corresponding queue. EATDQDCq.DQD setting should be reviewed.

CTDQOES bit (Cut-Through Descriptor Queue Overflow Error Status Flag)

[Setting conditions]

HW: This register has two exclusive set conditions:

- [ETHA is operating] (EAMC.OPC == 11 and EAMC.OPS == 11) Descriptor queue is "full (EACTDQDCq.CTDQD == EACTDQM.CTQDN)" and "not disabled (EATDQC.TCTDQD is not set)" when descriptor is received for cut-through descriptor
- [ETHA is not operating] (EAMC.OPC != 11 or EAMC.OPS != 11) When descriptor is received for cut-through descriptor.

[Clearing conditions]

- HW: When in RESET mode
- SW: Writing 1 to this bit.

[Error recovery]

- HW: Any descriptor received for a full descriptor queue is not accepted by ETHA and is not forwarded to RMAC [RMAC]
- SW: Too many descriptors are received for the corresponding queue. EACTDQDC.CTDQD setting should be reviewed.

DQSESn bits (Descriptor Queue n Security Error Status Flag (n = 0 to 7))

[Setting condition]

- HW: Bit q of this register is set when a non-secure descriptor is received (FDESCR.SEC is not set [FWD]) and queue q is a secure queue (EATDQSC.TDQSLn is set).

[Clearing conditions]

- HW: When in RESET mode
- SW: Writing 1 to this bit.

[Error recovery]

- HW: Any non-secure descriptor received for secure descriptor queue is not accepted by ETHA and is not forwarded to CPU
- SW: System-dependent, therefore cannot be defined.

32.3.7.1.8 EAEIE2 : Error Interrupt Enable Register 2

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0524

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	DQSEE7 to DQSEE0							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CTDQOEE	DQOEE7 to DQOEE0							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	DQOEE7 to DQOEE0	Descriptor Queue n Overflow Error Enable (n = 0 to 7) 0: Interrupt disabled for descriptor queue i 1: Interrupt enabled for descriptor queue i	R/W*1
8	CTDQOEE	Cut-Through Descriptor Queue Overflow Error Enable 0: Interrupt disabled 1: Interrupt enabled	R/W*1
15:9	—	These bits are read as 0.	R
23:16	DQSEE7 to DQSEE0	Descriptor Queue n Security Error Enable (n = 0 to 7) 0: Interrupt disabled for descriptor queue i 1: Interrupt enabled for descriptor queue i	R/W*1
31:24	—	These bits are read as 0.	R

Note 1. Read value differs from written value.

DQOEEn bits (Descriptor Queue n Overflow Error Enable (n = 0 to 7))

[Setting condition]

- Writing 1 to one of these bits.

[Clearing condition]

- Writing 1 to one of EAEID2.DQOED bits clears the corresponding bit in this register.

CTDQOEE bit (Cut-Through Descriptor Queue Overflow Error Enable)

[Setting condition]

- Writing 1 to this bit.

[Clearing condition]

- Writing 1 to EAEID2.CTDQOED register clears this bit.

DQSEEn bits (Descriptor Queue n Security Error Enable (n = 0 to 7))

[Setting condition]

- Writing 1 to one of these bits.

[Clearing condition]

- Writing 1 to one of EAEID2.DQSED bits clears the corresponding bit in this register.

32.3.7.1.9 EAEID2 : Error Interrupt Disable Register 2

Base address: ETHAm = 0x403C_A000 + 0x2000 × m (m = 0, 1)
ETHAm_NS = 0x503C_A000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0528

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	DQSED7 to DQSED0							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CTDQOED	DQOED7 to DQOED0							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	DQOED7 to DQOED0	Descriptor Queue n Overflow Error Disable (n = 0 to 7) Writing 1 to one of these bits clears the corresponding bit in EAEIE2.DQOEE register.	R
8	CTDQOED	Cut-Through Descriptor Queue Overflow Error Disable Writing 1 to this bit clears EAEIE2.CTDQOEE register.	R
15:9	—	These bits are read as 0.	R
23:16	DQSED7 to DQSED0	Descriptor Queue n Security Error Disable (n = 0 to 7) Writing 1 to one of these bits clears the corresponding bit in EAEIE2.DQSEE register.	R
31:24	—	These bits are read as 0.	R

32.4 Register Utilization

32.4.1 Operation Modes

Table 32.4 describes ETHA operation modes.

Table 32.4 ETHA operation modes

Operation mode	GWMS.OPS value	Description
DISABLE	1	• No ongoing transaction • Only status registers are accessible for writing when the agent clock is enabled.
RESET	0	• No ongoing transaction • An internal reset is asserted to reset ETHA logic with the status registers • No register is accessible for writing • RAM values are held.
CONFIG	2	• No ongoing transaction • Static and some dynamic registers are accessible for writing.
OPERATION	3	• Transactions are ongoing • Dynamic registers are accessible for writing.

32.4.1.1 Operation Mode Transitions

Figure 32.2 shows the operating mode transitions.

A mode transition can be triggered by:

- Hardware reset
- Software reset
- Configuration of EAMC.OPC. In this case, mode transition is confirmed by EAMS.OPS.

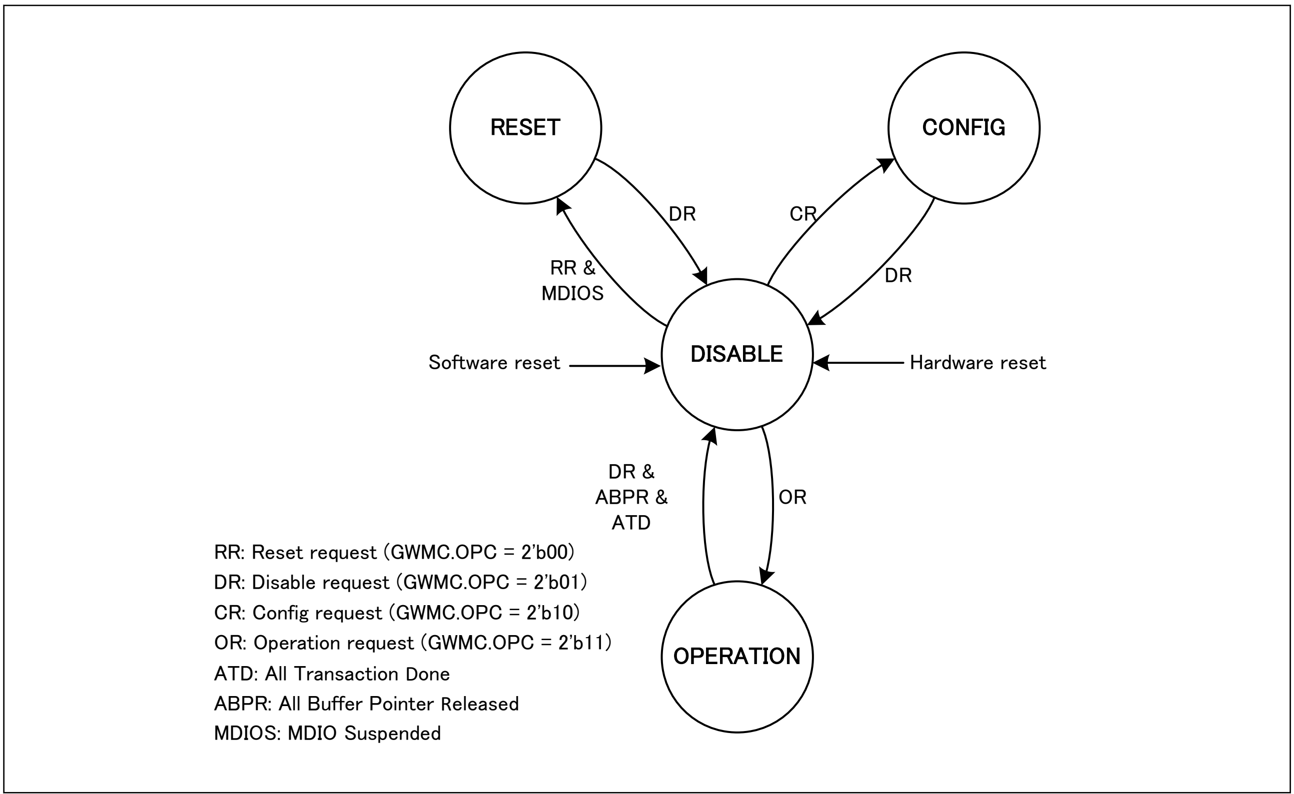


Figure 32.2 Operating mode transitions

Complementary information on transition conditions

ABPR:

- All the buffer pointers contained in the ETHA are released to the MFWD.

ATD:

- All the descriptors already received are processed and corresponding frames are transmitted to the Ethernet PHY
- All ongoing reception frames are fully sent to the MFWD.

MDIOS:

- MDIO has no ongoing access.

[Restrictions]

- Software shall only trigger transitions shown in Figure 32.2
  Do not directly transition from OPERATION to CONFIG.

32.4.2 Software Flows

SW: Follow the flows described in this section.

32.4.2.1 Software Flow Legend

Figure 32.3 describes the software flow legend.

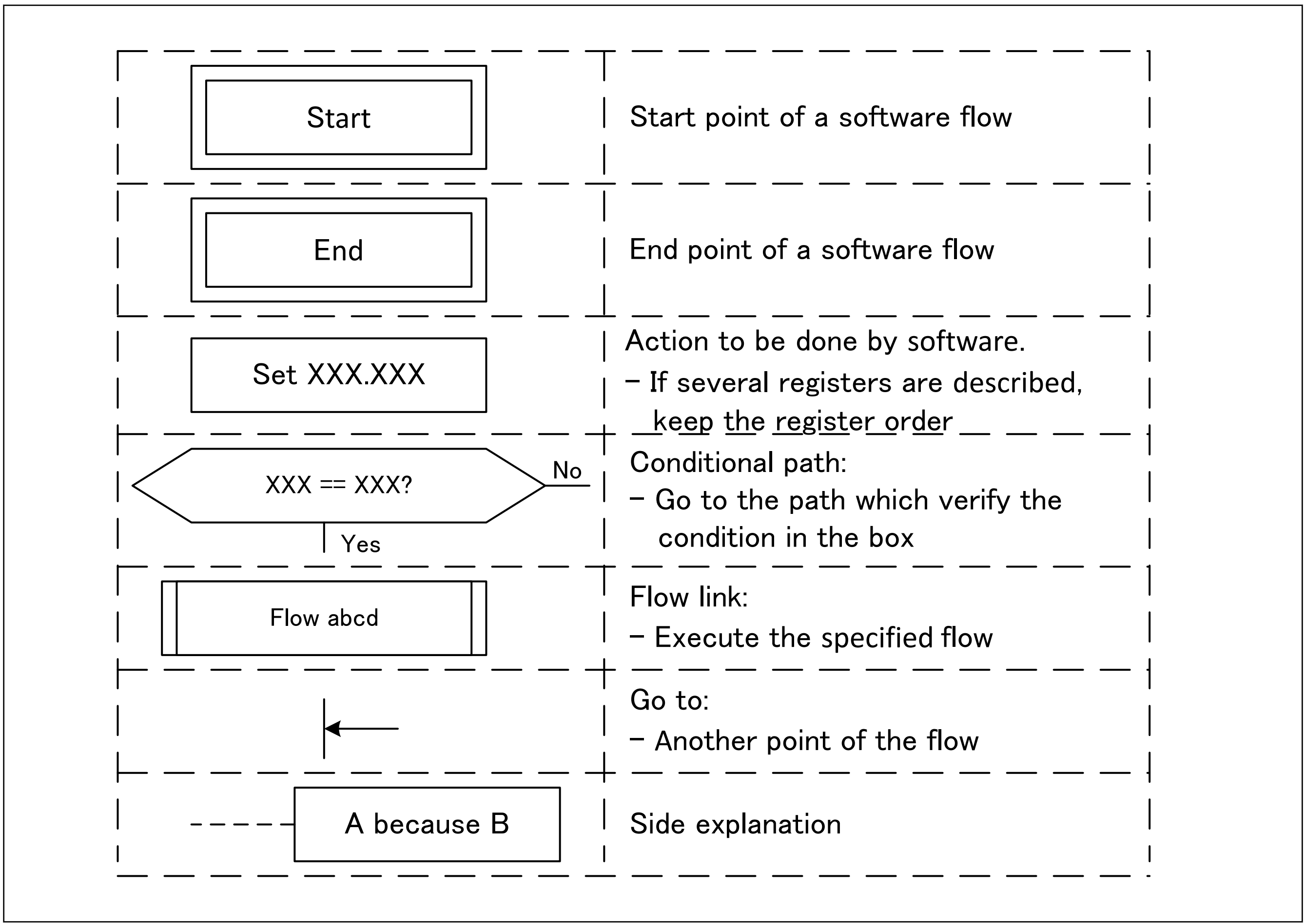


Figure 32.3 Software flow legend

32.4.2.2 Mode Transition Flow

Figure 32.4 describes the mode transition flow.

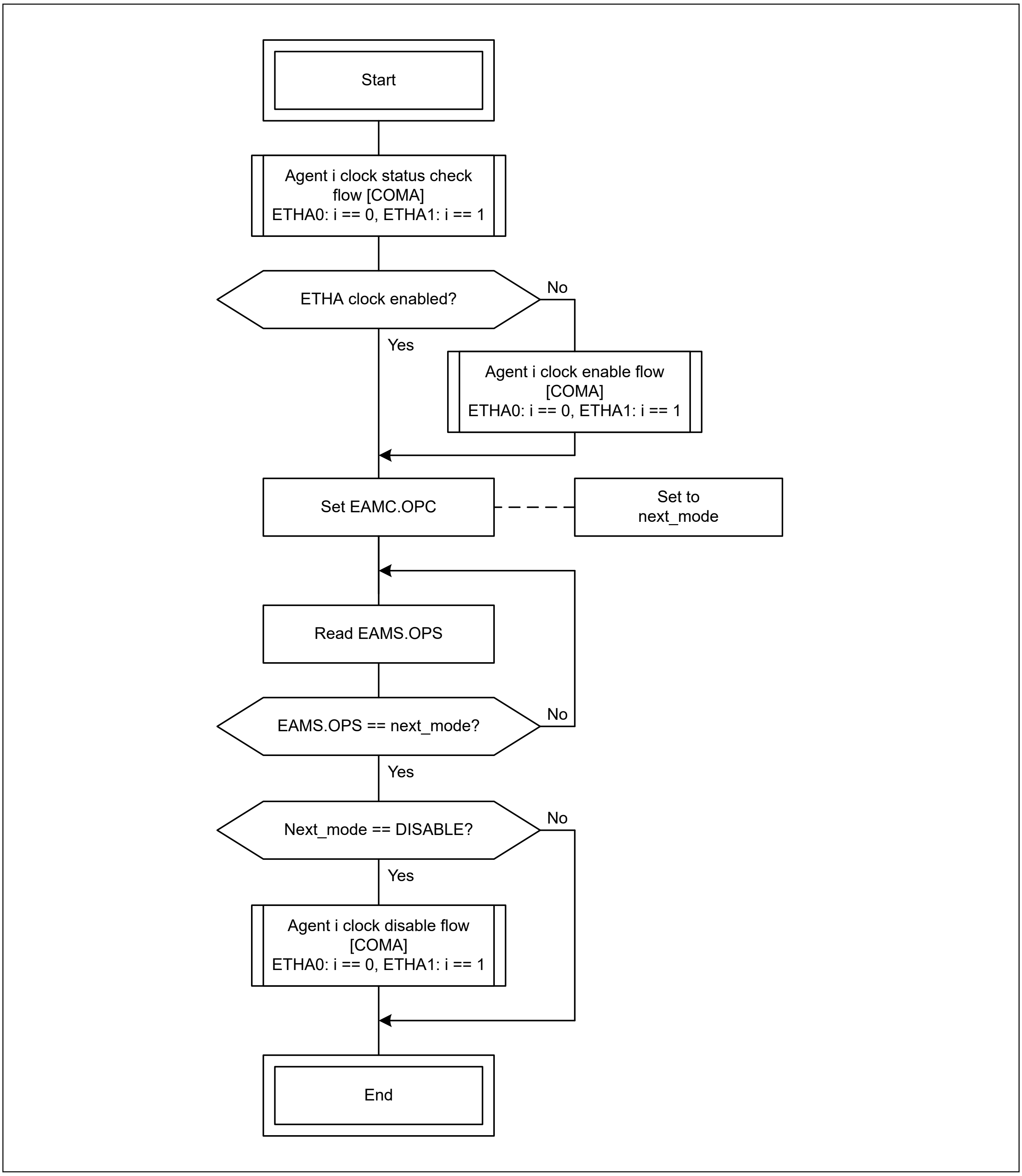


Figure 32.4 Mode transition flow

Note:
- If software already determines that the clock is enabled, the clock status check step can be skipped.
- If software does not need to disable the agent clock in DISABLE mode (because it goes directly to another mode or because the clock does not need to be disabled), the clock disable step can be skipped.

32.4.2.3 Reset Flow

Figure 32.5 describes the reset flow.

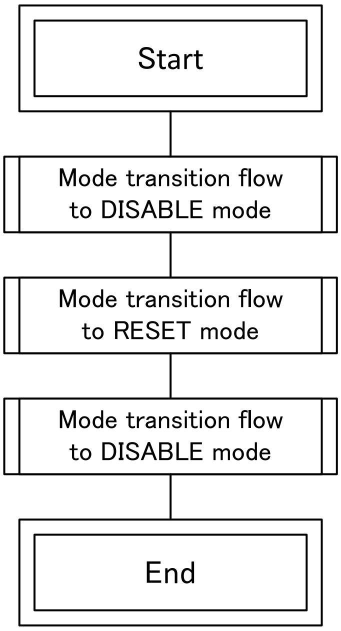


Figure 32.5 Reset flow

32.4.2.4 Initialization Flow

Figure 32.6 describes the initialization flow.

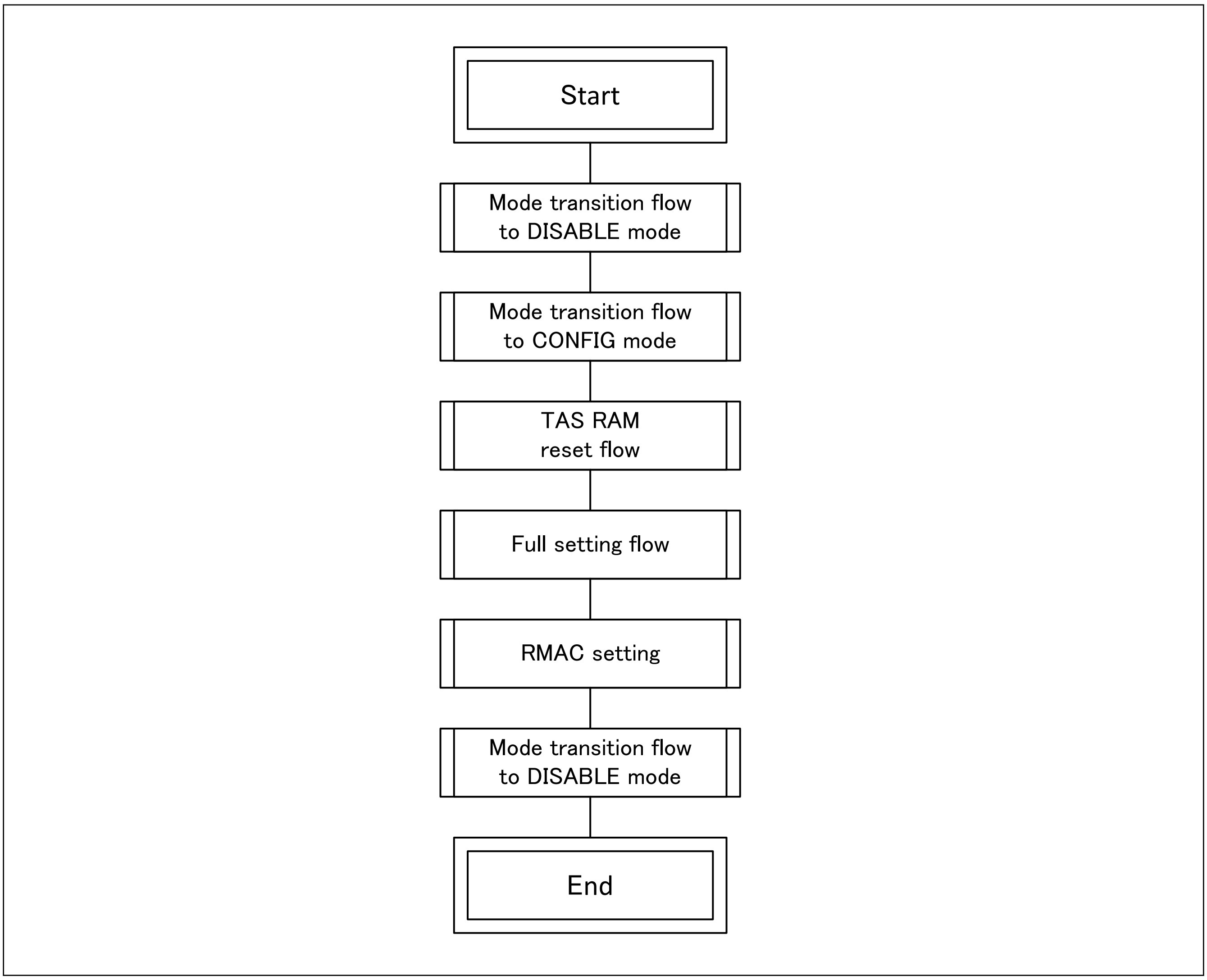


Figure 32.6 Initialization flow

Note: For RMAC setting, see RMAC specification [RMAC].

32.4.2.5 Reinitialization Flow

Figure 32.7 describes the reinitialization flow.

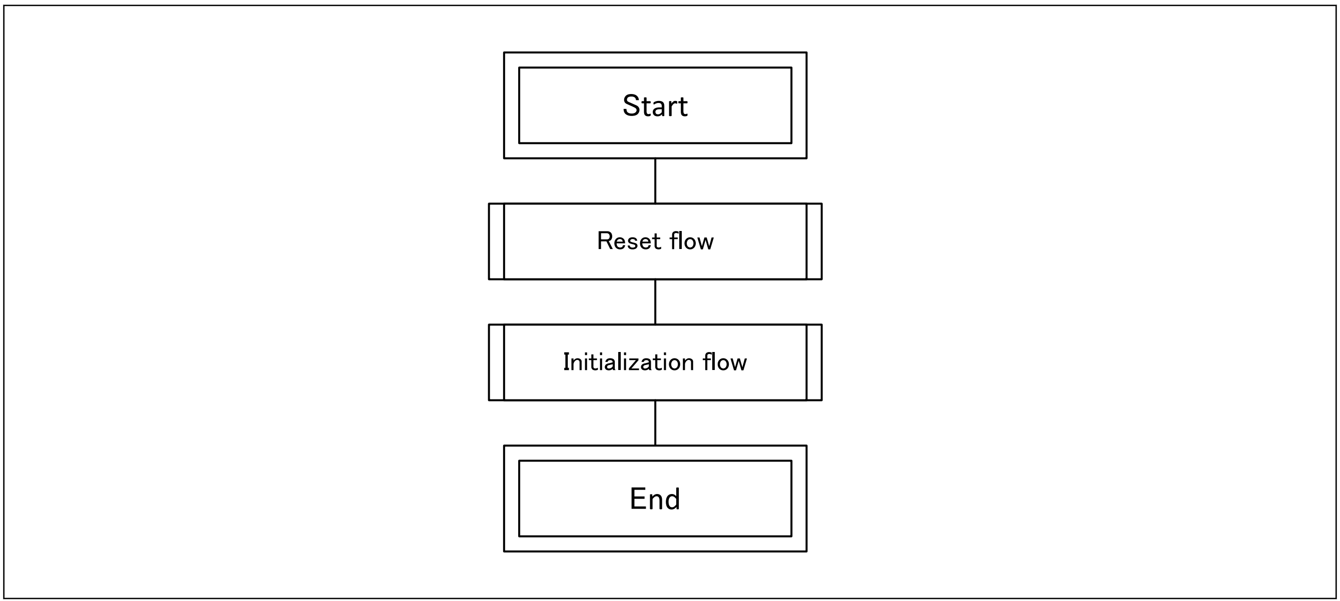


Figure 32.7 Reinitialization flow

32.4.2.6 TAS RAM Reset Flow

Figure 32.8 describes the TAS RAM reset flow.

Note: This flow is not usable in RESET and DISABLE modes.

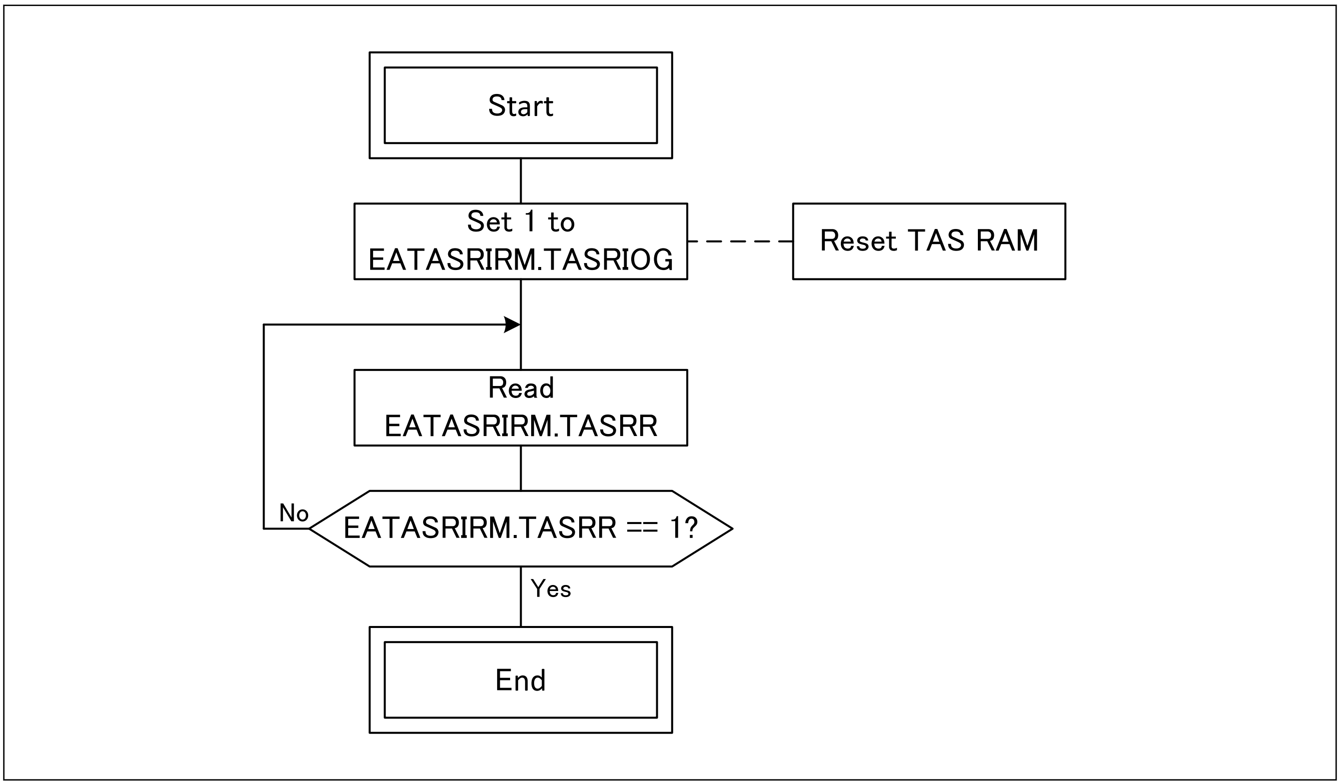


Figure 32.8 TAS RAM reset flow

32.4.2.7 CBS q Setting Flow (q = 0 to 7)

Figure 32.9 describes the CBS q setting flow.

Note: This flow is not usable in RESET and DISABLE modes.

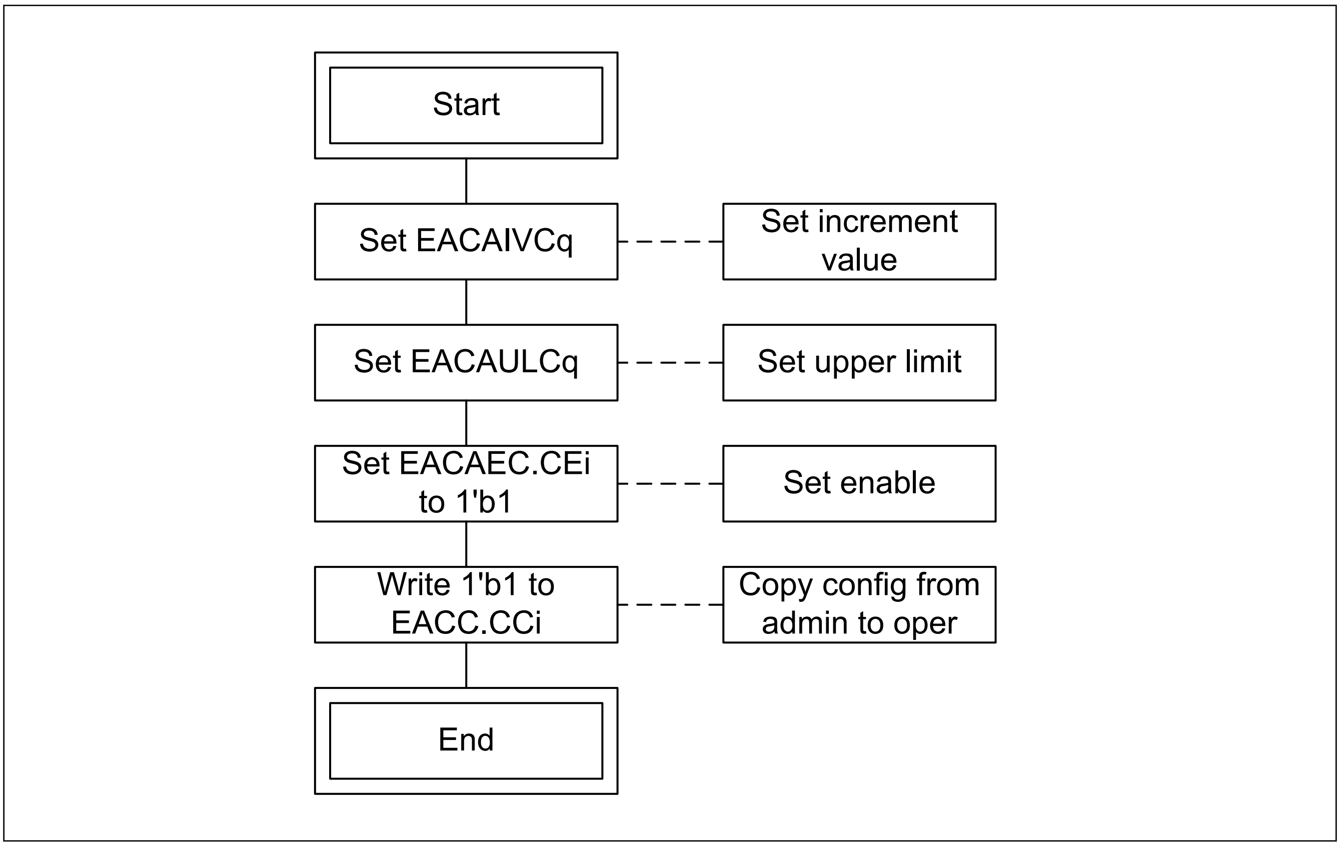


Figure 32.9 CBS q setting flow

32.4.2.8 CBS q Disabling Flow (q = 0 to 7)

Figure 32.10 describes the CBS q disabling flow.

Note: This flow is not usable in RESET and DISABLE modes.

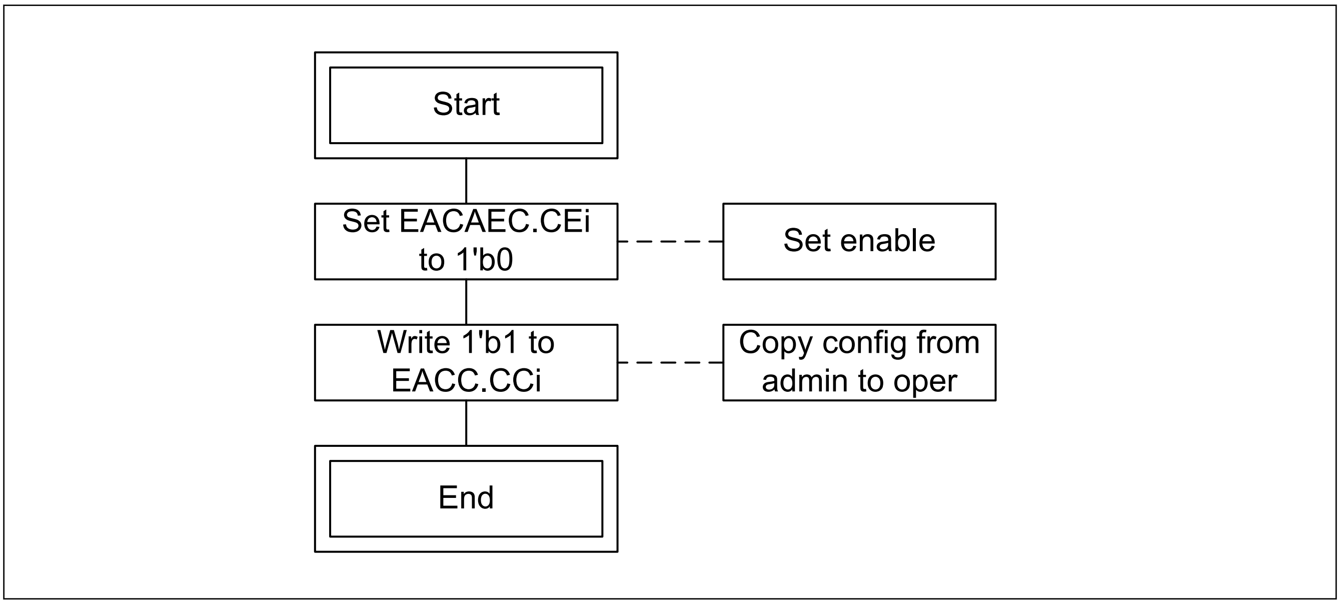


Figure 32.10 CBS q disabling flow

32.4.2.9 TAS Setting Flow

Figure 32.11 describes the TAS setting flow.

Note: This flow is not usable in RESET and DISABLE modes.

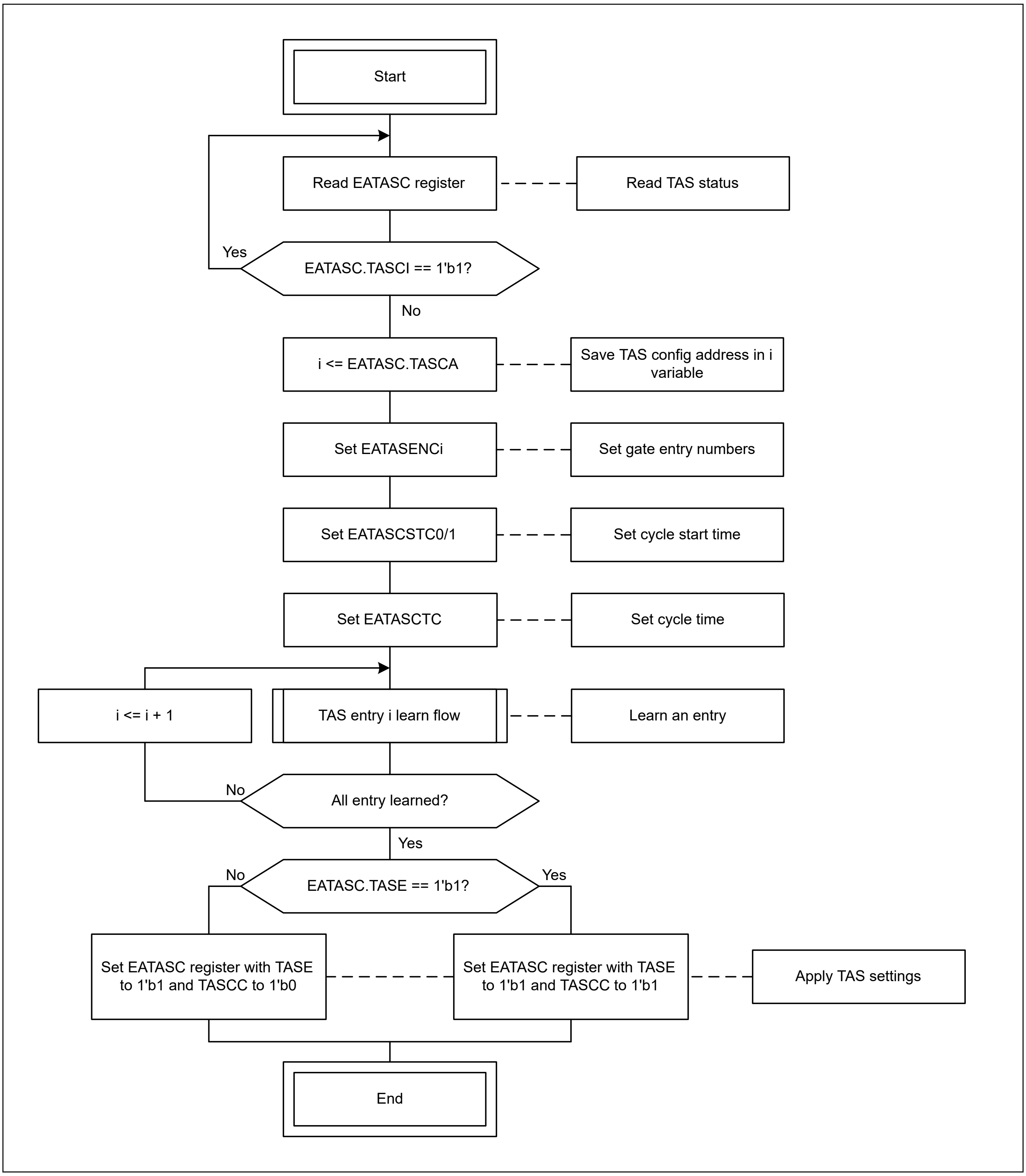


Figure 32.11 TAS setting flow

32.4.2.10 TAS Disabling Flow

Figure 32.12 describes the TAS disabling flow.

Note: This flow is not usable in RESET and DISABLE modes.

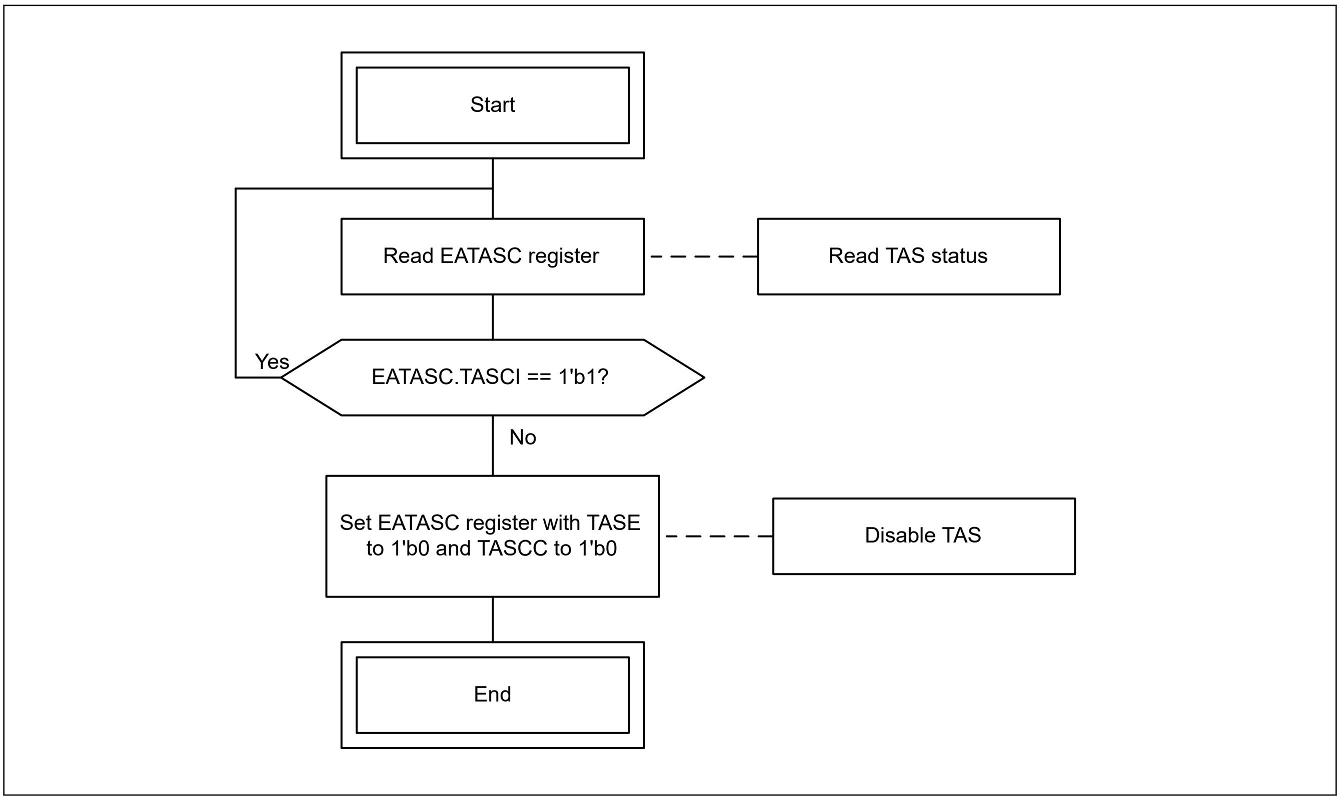


Figure 32.12 TAS disabling flow

32.4.2.11 TAS Enabling Flow

Figure 32.13 describes the TAS enabling flow.

Note: This flow is not usable in RESET and DISABLE modes.

The TAS enabling flow can only be applied after the TAS setting flow has been done at least once. By using the TAS enabling flow, TAS starts with the schedule previously set for the TAS setting flow.

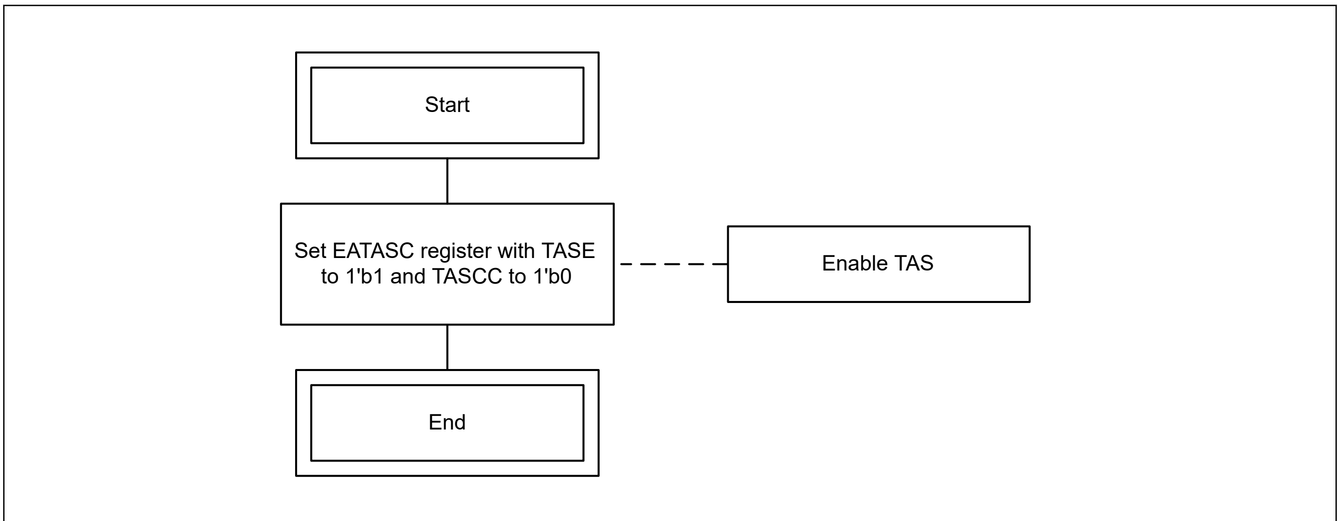


Figure 32.13 TAS enabling flow

32.4.2.12 TAS Entry i Learn Flow

Figure 32.14 describes the TAS entry i learn flow.

Note: This flow is not usable in RESET and DISABLE modes.

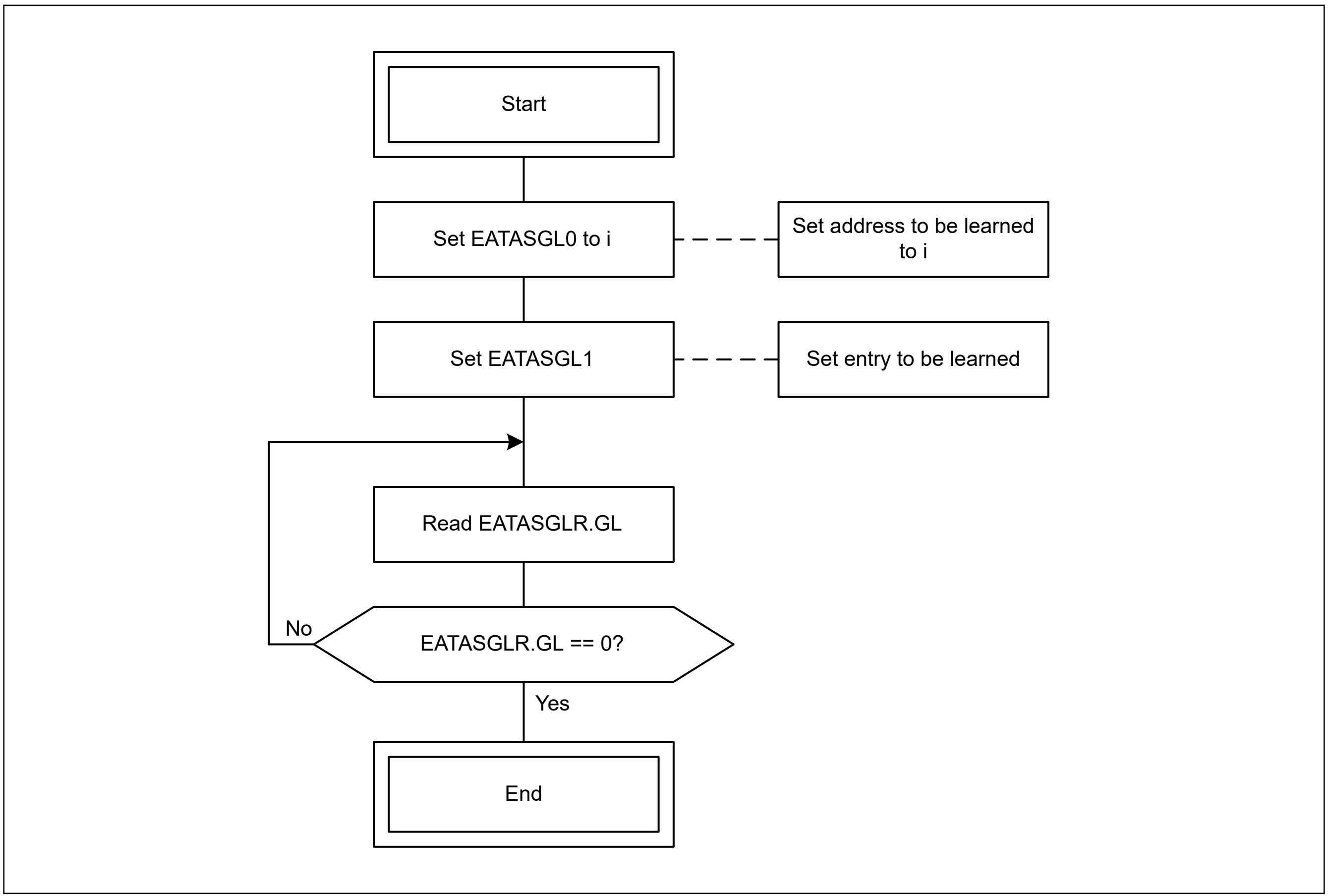


Figure 32.14 TAS entry i learn flow

32.4.2.13 TAS Entry i Read Flow

Figure 32.15 describes the TAS entry i read flow.

Note: This flow is not usable in RESET and DISABLE modes.

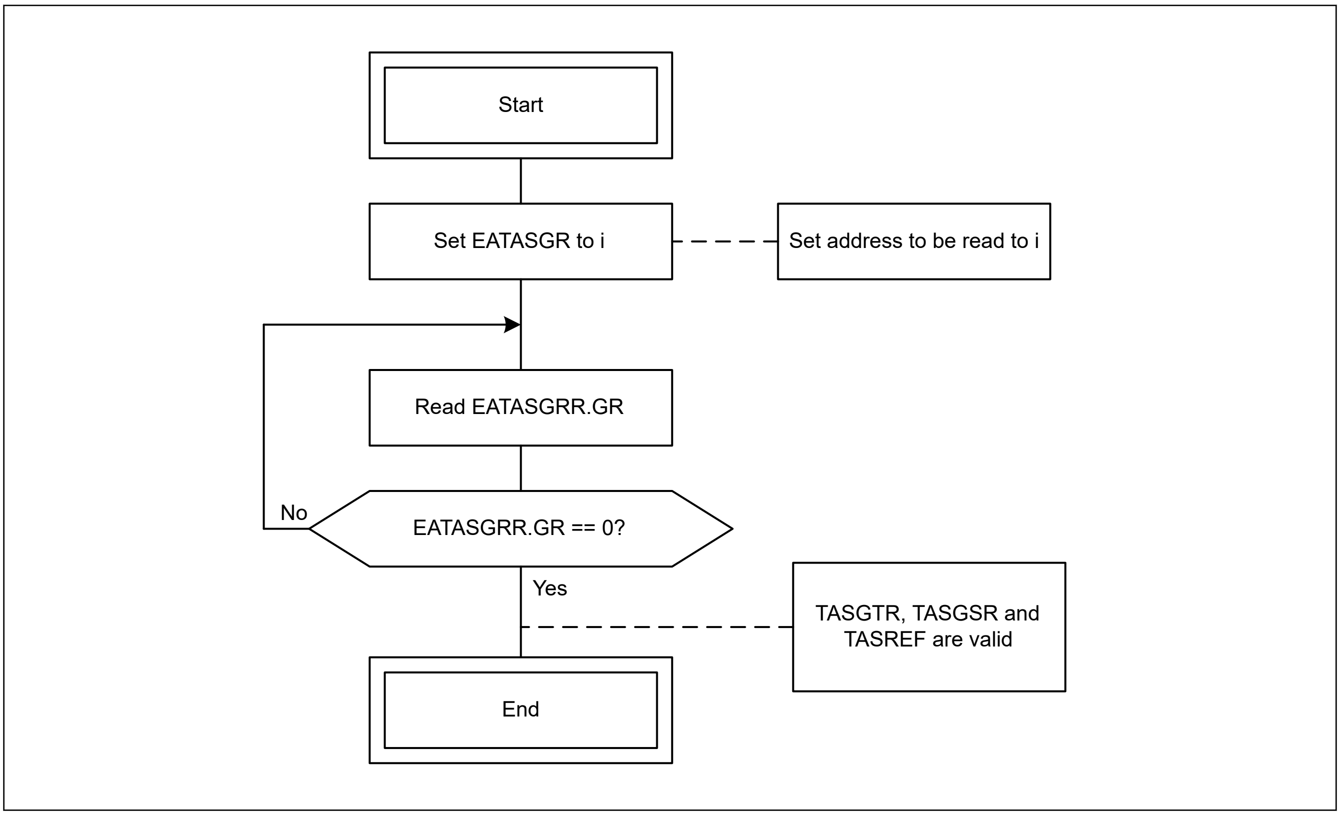


Figure 32.15 TAS entry i read flow

32.4.2.14 Interrupt Handling Flow

Figure 32.16 describes the interrupt handling flow.

Note: This flow is not usable in RESET mode.

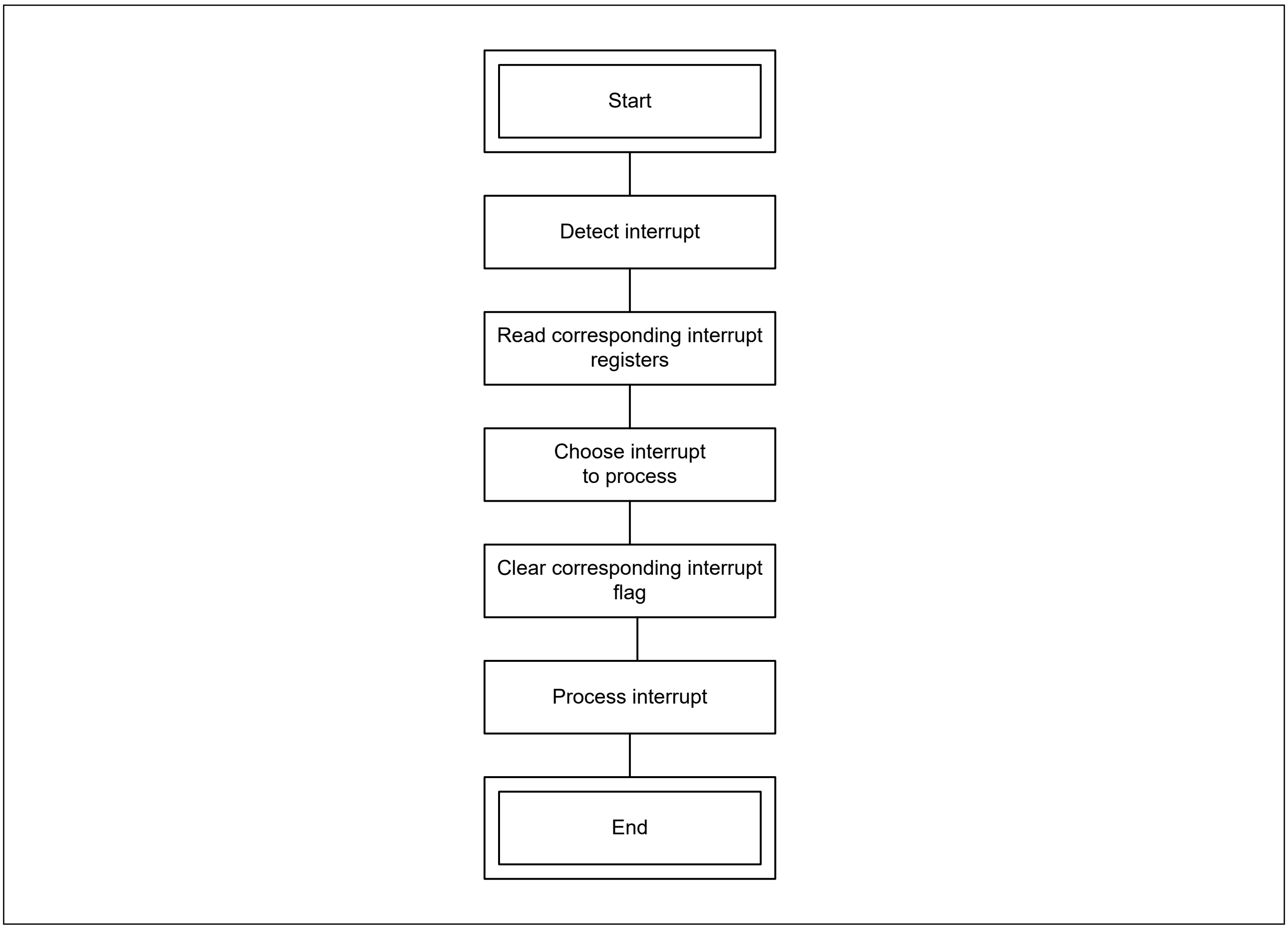


Figure 32.16 Interrupt handling flow

32.4.2.15 Called Software Flows

The flows described in this section can only be called from other flows as indicated in the flow link box in Figure 32.3 and cannot be used alone.

32.4.2.15.1 Full Setting Flow

Figure 32.17 shows the full setting flow.

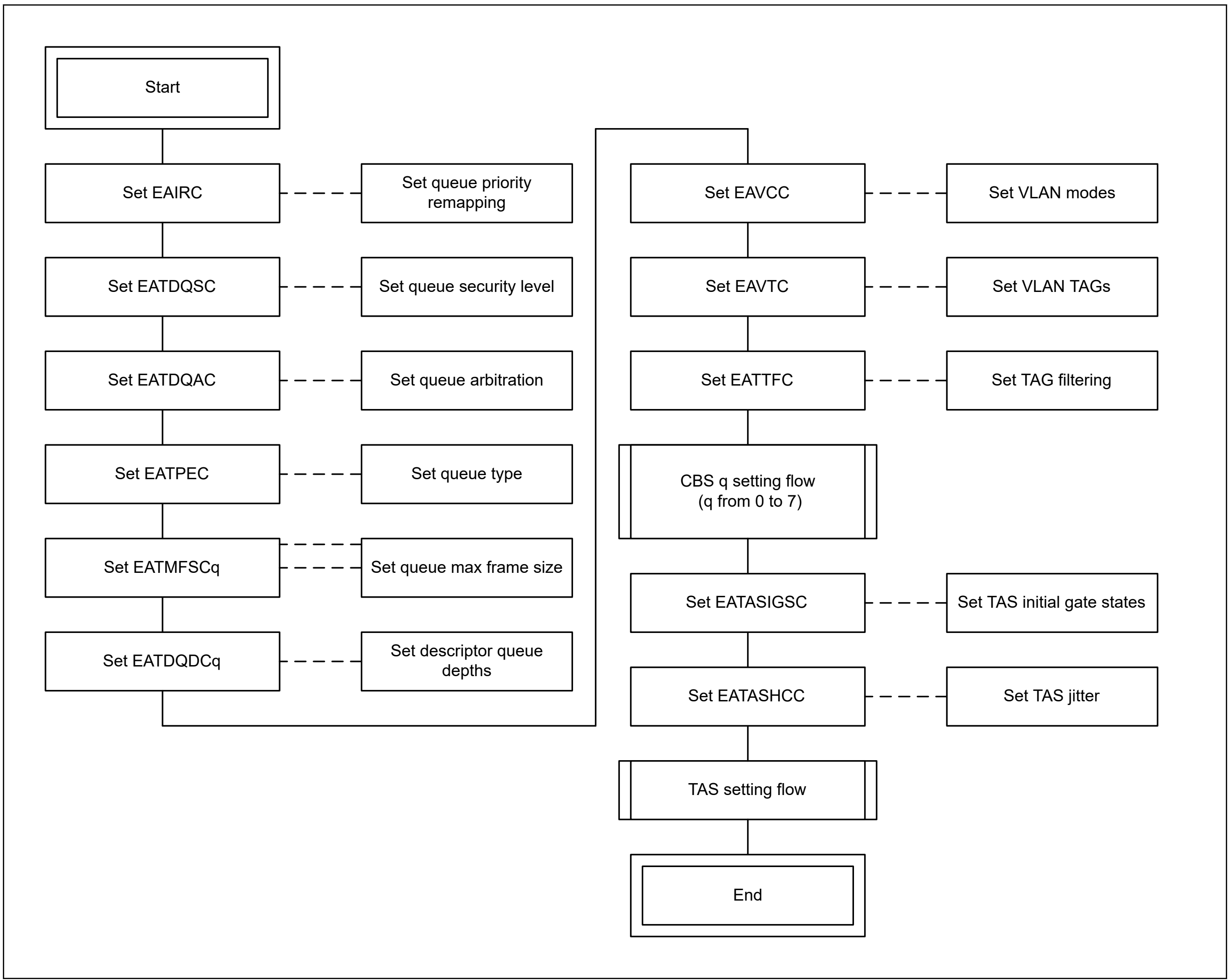


Figure 32.17 Full setting flow

32.4.2.16 Register Writable Without Software Flow

This section details registers that have not been described. These registers can be changed dynamically. However, it is necessary that the initial settings such as the clock enabling have been completed.

32.5 Functional Description

32.5.1 Data Transmission

Ethernet Agent allows data transmission through the ETHA TX data path. Figure 32.18 describes the TX data path.

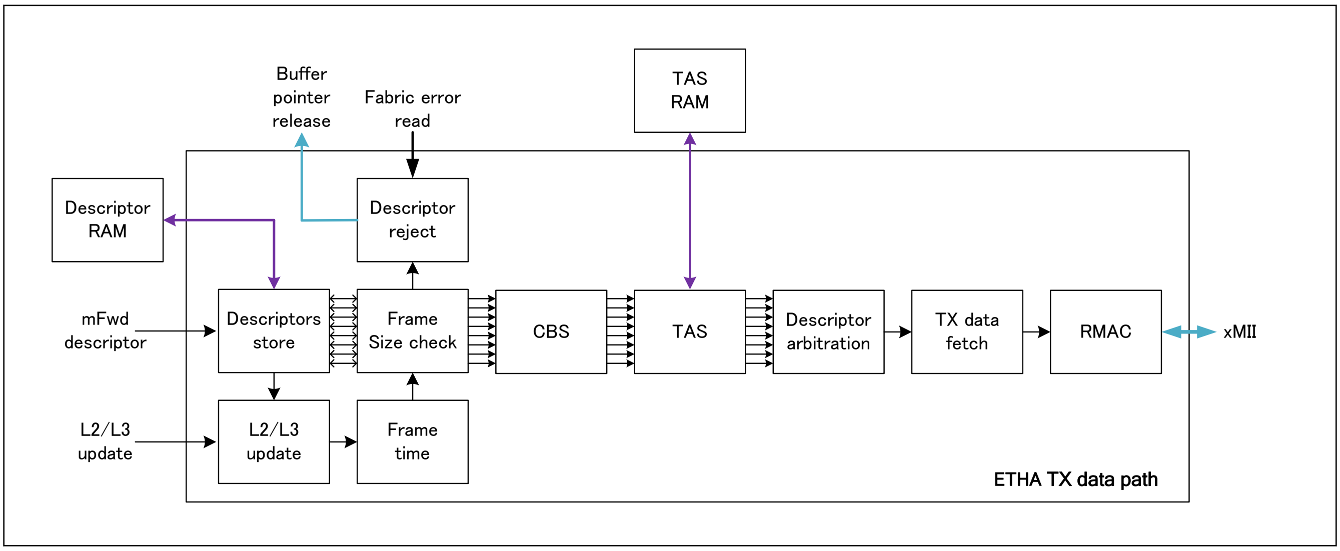


Figure 32.18 ETHA TX data path block diagram

The RX data path is separated in 11 blocks:

- Descriptor store: This block aims at storing the descriptor received from the MFWD in the descriptor RAM, reads it and sends it to the L2/L3 update.
- L2/L3 update: This block aims at fetching the L2/L3 update rules from the Forwarding Engine to update the frame while sending it.
- Frame time: This block aims at calculating how many nanosecond a frame is sent on the ethernet PHY for TAS arbitration. This block exists for TAS hardware purpose and its description is not required for switch utilization so it is not described.
- Frame size check: This block checks the frame size and can forward or discard frames.
- Descriptor reject: This block aims at releasing the pointer of the frames rejected by the frame size control block. This block exists for switch hardware purpose and its description is not required for switch utilization so it is not described.
- Cut-though: This block handles cut-through descriptor coming from MFWD.
- CBS: This block aims at shaping the data traffic per descriptor queue by controlling the data throughput per queue [802.1Qav].
- TAS: This block aims at shaping the data traffic per descriptor queue by controlling the data transmission with a schedule [802.1Qbv].
- Descriptor arbitration: Arbitrates between descriptors based on the descriptor queue using strict priority and/or WRR arbitration.
- TX data fetch: This block fetches the frame data from the local RAM, updates it depending on the information obtained by L2/L3 update module and on VLAN control information and releases the frame pointers. The pointer release exists for switch hardware purpose and its description is not required for switch utilization so it is not described.
- RMAC: This block handles the data exchange with the Ethernet PHY. The functionalities of this block are described in the RMAC specification [RMAC] so it is not described.

32.5.1.1 Descriptor Store

Descriptor store aims at storing the descriptors coming from the descriptor bus MFWD in the descriptor RAM. This function is the same as the descriptor storage function in the descriptor store block in GWCA. See GWCA specification for more details [GWCA]. Table 32.5 describes the association between ETHA and GWCA registers.

Table 32.5 Descriptor store ETHA/GWCA register association (1 of 2)

Register/signal name in GWCA [GWCA]	Register/signal name in ETHA
GWIRC.IPVRi (i = 0 to 7)	EAIRC.IPVRi (i = 0 to 7)
GWRDQSC.RDQSL	EATDQSC.TDQSL

Table 32.5 Descriptor store ETHA/GWCA register association (2 of 2)

Register/signal name in GWCA [GWCA]	Register/signal name in ETHA
GWRDQC.RDQD	EATDQC.TDQD
GWRDQC.RDQP	EATDQC.TDQP
GWRDQDCq.DQD (q = 0 to 7)	EATDQDCq.DQD (q = 0 to 7)
GWRDQMq.DNQ (q = 0 to 7)	EATDQMq.DNQ (q = 0 to 7)
GWRDQMLMq.DMLQ (q = 0 to 7)	EATDQMLMq.DMLQ (q = 0 to 7)
GWEIS1.DQOES	EAEIS2.DQOES
GWEIS1.DQSES	EAEIS2.DQSES

32.5.1.2 L2/L3 Update

This block is the same as the L2/L3 update block in GWCA. See GWCA specification for more details [GWCA].

32.5.1.3 Frame Size Check

Frame size check aims at checking if a frame has the right properties to be received by CPU using EATMFSCq (q = 0 to 7) registers.

EATMFSCq (q = 0 to 7) registers are used to set the maximum frame size accepted for each descriptor queue. Any frame received for queue q with a size larger than EATMFSCq.MFS is discarded and EAEIS0.FSESq flag is set. The frame size considered is the size of frame on the PHY.

32.5.1.4 Cut-through

Cut-through controls the cut-through descriptors queue to store cut-through descriptors received from MFWD using EATDQC.TCTDQD, EACTQC.CTQD and EACTDQDC.CTDQD registers and can be monitored using EACTDQM.CTQDN and EACTDQMLM.CTDMLQ registers.

- EATDQC.TCTDQD register is used to disable the cut-through queue. If the cut-through queue is disabled, eha_ct_ready bit is deasserted, and no descriptor is stored.
- EACTQC.CTQD is used to control the number of clocks that should be waited after receiving a cut-through descriptor to be able to send it for transmission.
- EACTDQDC.CTDQD register is used to set the maximum number of descriptors that can be stored in the cut-through descriptor queue. If the descriptor queue becomes full (EACTDQDC.CTDQD == EACTDQM.CTQDN) eha_ct_ready signal is deasserted. If a descriptor is received for cut-through descriptor queue while it is full, the descriptor is not stored and EAEIS2.CTDQOES flag is set.
- EACTDQM.CTQDN register is used to monitor the current number of descriptors in the cut-through descriptor queue.
- EACTDQMLM.CTDMLQ register is used to monitor the maximum number of descriptors that has been held in cut-through descriptor queue since the previous reset or since the previous time that this has been read.

32.5.1.5 CBS (Credit-based shaper) [802.1Qav]

CBS aims at shaping the data traffic per descriptor queue by controlling the data throughput per queue using EACAEC.CE, EACC.CC, EACAIVCq.CIV (q = 0 to 7) and EACAULCq.CUL (q = 0 to 7) registers and can be monitored using EACOEM.CE, EACOIVMq.CIV (q = 0 to 7), EACOULMq.CUL (q = 0 to 7) and EACGSM.CGS registers and, EAEIS1.CULES interrupt register.

- EACAEC.CE register is used to enable or disable CBS modules.
- EACAIVCq.CIV register is used to configure descriptor queue q CBS throughput.
- EACAULCq.CUL register is used to configure the maximum credit number that can be stored in descriptor queue q CBS module. It is used to monitor associated descriptor queue of CBS module with a long back pressure due to high priority non-CBS-shaped queues interference traffic, or due to queue open time partial overlapping in TAS. If too much back pressure has been applied to a CBS-related descriptor queue, the corresponding EAEIS1.CULES flag is set.
- EACC.CC register is used to apply configurations set in EACAEC.CE, EACC.CC, EACAIVCq.CIV and EACAULCq.CUL registers by copying them to EACOEM.CE, EACOIVMq.CIV and EACOULMq.CUL registers.

- EACOEM.CE register is used to monitor which descriptor queue is used as its CBS module enabled.
- EACOIVMq.CIV register is used to monitor the current throughput set for descriptor queue q.
- EACOULMq.CUL register is used to monitor the current maximum credit number for descriptor queue q.
- EACGSM.CGS register is used to monitor which queue is allowed by CBS to transmit.
- EAEIS1.CULES register is used to flag CBS Upper Limit Errors.

32.5.1.5.1 CBS Setting

CBS setting should follow equation (1) and equation (2). While setting descriptor queue q CBS, CIV should be set to EACAIVCq.CIV register and CUL should be set to EACAULCq.CUL register.

(1) CIV[byte/cycle] = ((portTransmitRate[bps] / 8) × (bandwidthFraction[%] / 100)) / clk_f[Hz] × (CycleTime / GateOpenTime).

For example, when *portTransmitRate* is 1 Gbps, *bandwidthFraction* is 25% and clk_f is 200 MHz, CIV is $\underline{0}$.15625.
CIV[19:16] = $\underline{0}$ CIV[15:0] = 0.15625 × 2^{16} = 10,240 = 2800 H.

(2) CUL[byte] = (maxInterferenceSize[bit] × clk_f[Hz] / portTransmitRate[bps] + requestDelay[cycle]) × CIV[byte/cycle].

Where:

- *portTransmitRate* is the ETHA link throughput [RMAC].
- *bandwidthFraction* is the percentage of bandwidth that the CBS corresponding should use.
- *clk_f* is the clock clk frequency.
- *(CycleTime / GateOpenTime)* is the ratio between the current TAS cycle time (EATASCTM.TASOCT) and the time the CBS corresponding descriptor queue TAS gate is opened during EATASCTM.TASOCT time. When TAS is disabled, *(CycleTime / GateOpenTime)* is equal to 1.
- *maxInterferenceSize* is the maximum size of any burst of traffic that can delay the transmission of a frame that is available for transmission for this traffic class (see section 32.5.1.5.2. Maximum Interference Size Calculation).
- *requestDelay* is the time required from descriptor reception to frame transmission ready and is fixed to 50 cycles.

Restrictions:

- SW: All CBS *bandwidthFraction* sum should be smaller than 70% for queues with simultaneously opened TAS gates.

32.5.1.5.2 Maximum Interference Size Calculation

The maximum interface *maxInterferenceSize* size can be calculated for a queue q using equation (3).

Each interference size includes preamble and inter frame gap (IFG).

(3) *maxInterferenceSize* = *queueInterferenceSizeLow* + *queueInterfereSize* + Σ_i(*queueInterferenceSizeHigh*[i])

Where:

- *queueInterferenceSizeLow* is the interference created by all the queues which have a priority smaller than q. This interface size is equal to (queueLowMaxSize + 20) × 8 where *queueLowMaxSize* is **EATMFSCi.MFSi** maximum values for all i smaller than q.
- *queueInterfereSize* is the interference created by queue q and is equal to (**EATMFSCq.MFS** + 20) × 8
- *queueInterferenceSizeHigh*[i] is the interference created by queue i with a higher priority than q. If CBS is enabled for queue i, *queueInterferenceSizeHigh*[i] is equal to (**EATMFSCi.MFSi** + *queueLowMaxSize* + 40) × 8 where *queueLowMaxSize* is **EATMFSCj.MFSj** maximum values for all j smaller than i. If CBS is disabled for queue i, *queueInterferenceSizeHigh*[i] is equal to (*MaxFrameBurst* × EATMFSCi.MFSi + *queueLowMaxSize* + 40) × 8 where *queueLowMaxSize* is EATMFSCj.MFSj maximum values for all j smaller than i and where *MaxFrameBurst* is the maximum number of frames that can be expected in a burst. *MaxFrameBurst* is a variable which is system-dependent and cannot be defined.

32.5.1.6 TAS (Time Aware Shaper) [802.1Qbv]

TAS aims at shaping the data traffic per descriptor queue by controlling a AVTP based schedule using EATASC, EATASIGSC, EATASENCi (i = 0 to 7), EATASCTENC, EATASCSTC0/1, EATASCTC, EATASGL0/1, EATASGLR, EATASHCC registers and can be monitored using ENTASENMi (q = 0 to 7), EATASCTENM, EATASCSTM0/1, EATASCTM, EATASGR, EATASGRR, EATASRIRM, EATASSM registers and, EAIEIS0.TASCTGEES, EAIEIS1.TASGES and EAIEIS1.TASCTGES interrupt registers.

- EATASC register is used for TAS configuration and enabling.
- EATASIGSC register is used to set the initial gate states.
- EATASENCi register is used to set the number of entries used in TAS RAM by gate i.
- EATASCTENC register is used to set the number of entries used in TAS RAM by cut-through gate.
- EATASCSTC0/1 registers are used to set the cycle start time.
- EATASCTC register is used to set the cycle time.
- EATASGL0/1 and EATASGLR registers are used to write entries in the TAS RAM.
- EATASHCC is used to set the TAS jitter caused by the switch architecture. ENTASENMi register is used to monitor the number of entries used in the TAS RAM by gate i for the ongoing schedule.
- EATASCTENM register is used to monitor the number of entries used in the TAS RAM by cut-through gate for the ongoing schedule.
- EATASCSTM0/1 registers are used to monitor the next cycle start time for the ongoing schedule.
- EATASCTM register is used to monitor the cycle time for the ongoing schedule.
- EATASGR and EATASGRR registers are used to read entries in the TAS RAM.
- EATASRIRM register is used to initialize TAS RAM entries to all 0s.
- EATASSM register is used to monitor TAS operation.
- EAIEIS1.TASGES and EAIEIS1.TASCTGES registers are used to monitor gates errors due to a lack of time to read the TAS RAM.

32.5.1.6.1 TAS General Behavior

TAS module is a scheduler which allows frames to be transmitted depending on their descriptor queue and time by allowing frame transmission when its corresponding gate is opened. TAS scheduler uses a time reference of 64-bit AVTP timer (derived from gPTP timer [gPTP]).

TAS general behavior can be summed up in the following four categories and is described in this section. The TAS schedule is described in section 8.1.6.2.

- TAS enabling
- TAS configuration change
- TAS disabling
- gPTP offset correction recovery [gPTP].

Note: HW: TAS module does not support 64-bit AVTP timer overflows.

1. TAS enabling
   TAS enabling happens when TAS is set while being disabled (EATASC.TASE not set).
   (a) TAS enabling in the future
   TAS module can be such that its schedule starts in the future. Figure 32.19 describes TAS behavior when its schedule starts in the future. The schedule used in this figure is just an example and does not correspond to anything relevant for TAS general behavior explanation.
   Restrictions:
      - SW: To enable TAS in the future, ensure that EATASC.TASE is set before the AVTP timer reaches the following time: {EATASCSTC1, EATASCSTC0} – 2 × EATASCTC.TASACT.

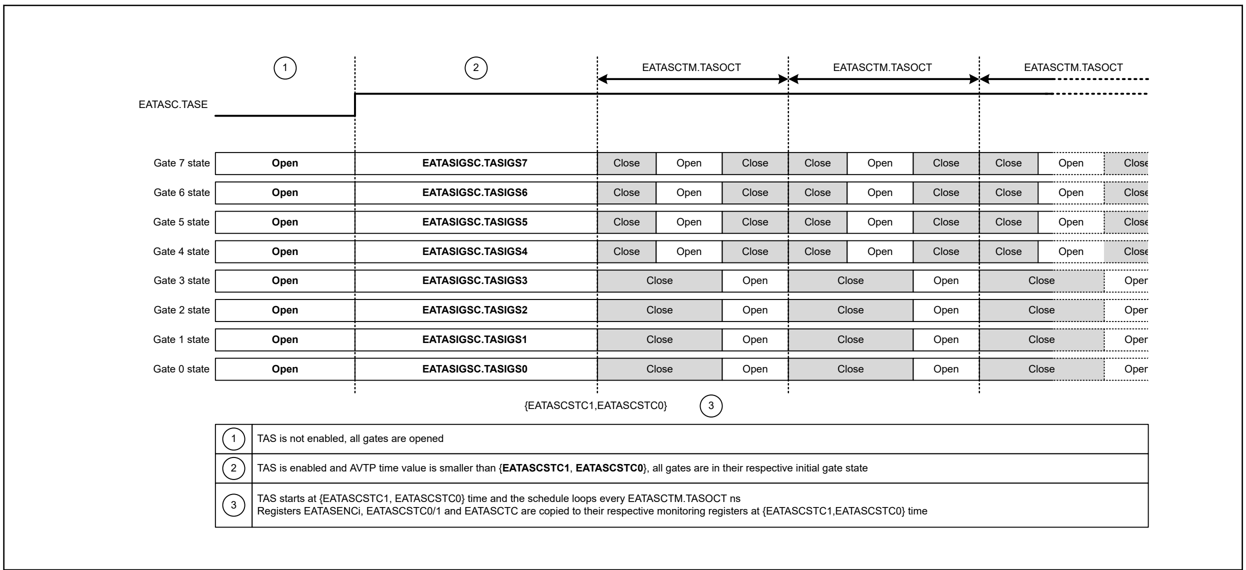


Figure 32.19 TAS behavior when its schedule starts in the future

Restrictions:

- HW: When EATASC.TASE is set, an ongoing frame crosses a gate close state if its initial gate state is closed
- HW: If EATASC.TASE is set such that an ongoing frame does not have the time to finish transmitting before {EATASCSTC1, EATASCSTC0} is reached, the ongoing frame can cross a gate close state in the schedule first cycle
- HW: If EATASC.TASE is set between {EATASCSTC1, EATASCSTC0} – 2 × EATASCTC.TASACT and {EATASCSTC1, EATASCSTC0}, there is a possibility that enabling cannot happen properly and TAS module recovers during 2 × cycle_time[ns] from the start of the new cycle where cycle_time is the schedule cycle time set in EATASCTC.

(b) TAS enabling in the past
TAS module can be such that its schedule should have started in the past. Figure 32.20 describes TAS behavior when its schedule should have started in the past. The schedule used in this figure is just an example and does not correspond to anything relevant for TAS general behavior explanation.

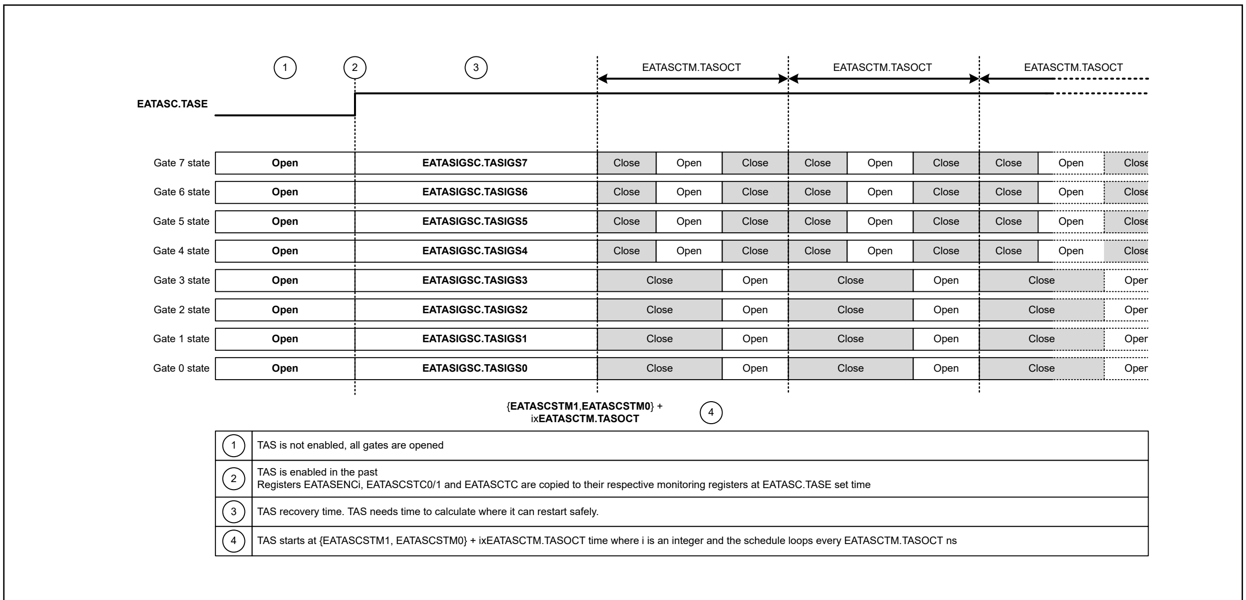


Figure 32.20 TAS behavior when its schedule should have started in the past

TAS recovery time (*tas_recovery_time*) maximum value can be calculated by equation (4).

(4) *tas_recovery_time[ns] = clk_period[ns]* × (*tase_set_time[ns]* – *start_time[ns]*) / *cycle_time[ns]* + 2 × *cycle_time[ns]*
Where:

- *clk_period* is the clock clk period
- *tase_set_time* is the AVTP timer time at which the EATASC.TASE has been set
- *start_time* is the AVTP timer time at which the schedule has been set to start ({EATASCSTC1, EATASCSTC0})
- *cycle_time* is the schedule cycle time set in EATASCTC.

Functions:

- Equation (4) comes from the fact that TAS switches its start time by *cycle_time* every clock during recovery. A problem resulting from that is that the recovery time increases proportionally with *tase_set_time[ns]* – *start_time[ns]*. To avoid this issue, TAS can switch its start time by 256^i × *cycle_time* (i = 0 to 6) every clock when start time is too far in the past.

Restrictions:

- HW: Because of the start setting in the past, the hardware guard band cannot guarantee from EATASC.TASE setting time to the end of the first cycle (frames can cross a gate close state)
- HW: Because of the start setting in the past, EAEIS1.TASGES gate errors can be set.

2. TAS configuration change
TAS configuration change happens when TAS is set while being enabled (EATASC.TASE already set).

(a) TAS configuration change in the future
TAS module can be such that its schedule is reconfigured in the future. Figure 32.21 describes TAS behavior when its schedule is reconfigured in the future.
The schedule used in this figure is just an example and does not correspond to anything relevant for TAS general behavior explanation.
Restrictions:

- SW: To enable TAS in the future, ensure that EATASC.TASCC is set before the AVTP timer reaches the following time: {EATASCSTC1, EATASCSTC0} – 2 × EATASCTC.TASACT.

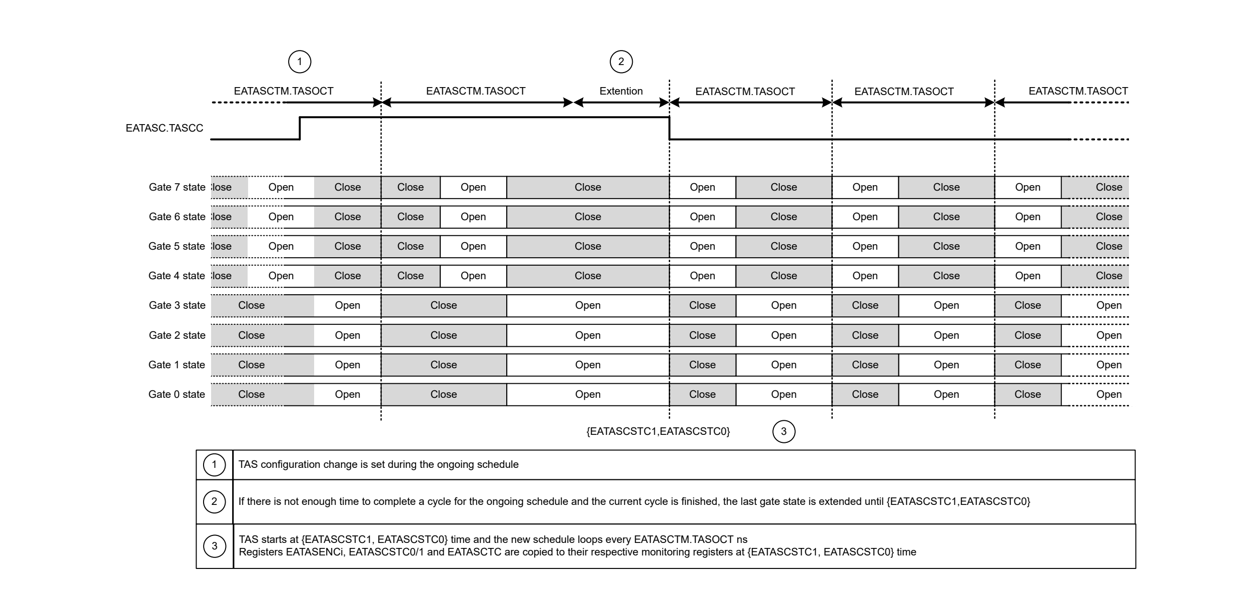


Figure 32.21 TAS behavior when its schedule is reconfigured in the future

Restrictions:

- HW: If EATASC.TASCC is set between {EATASCSTC1, EATASCSTC0} – 2 × EATASCTC.TASACT and {EATASCSTC1, EATASCSTC0} there is a possibility that configuration change cannot happen properly and

TAS module recovers during 2 × cycle_time[ns] from the start of the new cycle where cycle_time is the schedule cycle time set in EATASCTC.

(b) TAS configuration change in the past

TAS module can be such that its schedule reconfiguration should have happened in the past. Figure 32.22 describes TAS behavior when its schedule reconfiguration should have happened in the past. The schedule used in this figure is just an example and does not correspond to anything relevant for TAS general behavior explanation.

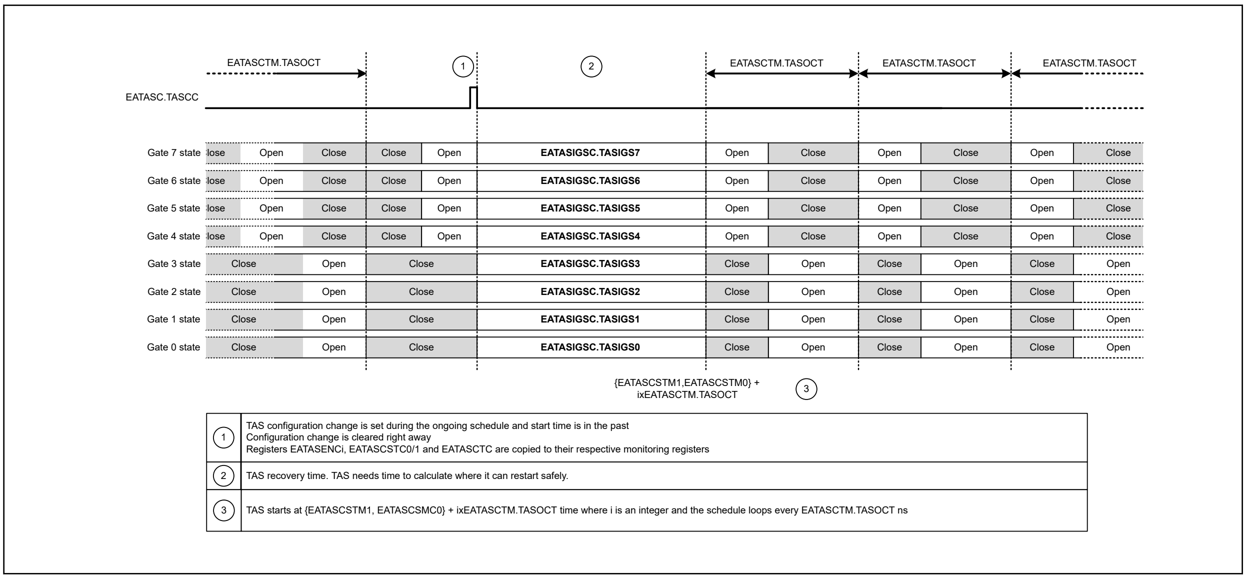


Figure 32.22 TAS behavior when its schedule reconfiguration should have happened in the past

TAS recovery time (*tas_recovery_time*) maximum value can be calculated by equation (5).

(5) $tas\_recovery\_time[ns] = clk\_period[ns] \times (tascc\_set\_time[ns] - restart\_time[ns]) / cycle\_time[ns] + 2 \times cycle\_time[ns]$

Where:

- *clk_period* is the clock clk period
- *tascc_set_time* is the AVTP timer time at which the EATASC.TASCC has been set
- *restart_time* is the AVTP timer time at which the schedule has been set to restart ({EATASCSTC1, EATASCSTC0})
- *cycle_time* is the schedule cycle time set in EATASCTC.

Functions:

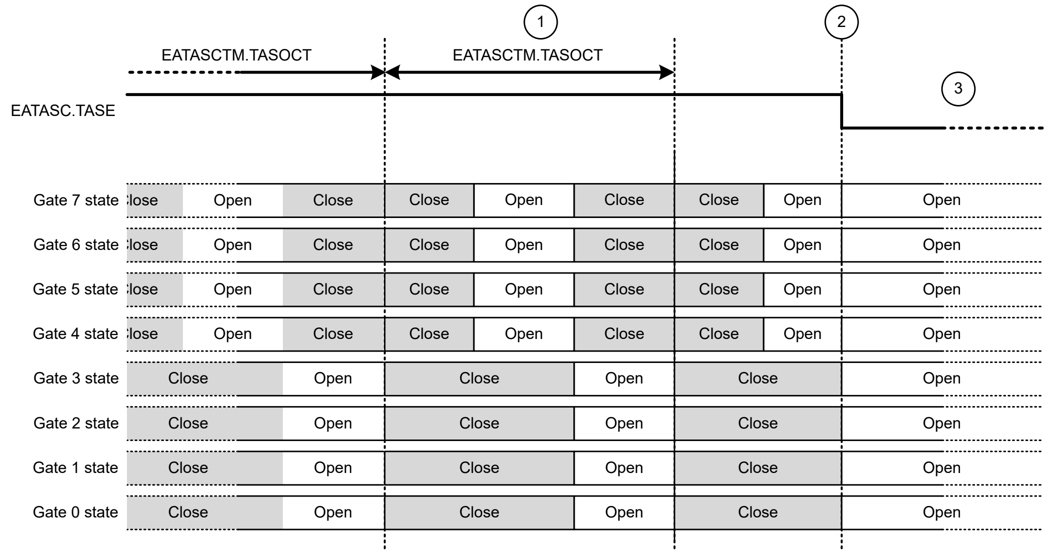
- Equation (5) comes from the fact that TAS switches its start time by *cycle_time* every clock during recovery. A problem resulting from that is that the recovery time increases proportionally with *tase_set_time[ns] – start_time[ns]*. To avoid this issue, TAS can switch its start time by 256^i × *cycle_time* (i = 0 to 6) every clock when start time is too far in the past.

Restrictions:

- HW: Because of the start setting in the past, the hardware guard band cannot guarantee from EATASC.TASCC setting time to the end of the first new cycle (frames can cross a gate close state)
- HW: Because of the start setting in the past, EAEIS1.TASGES gate errors can be set.

3. TAS disabling

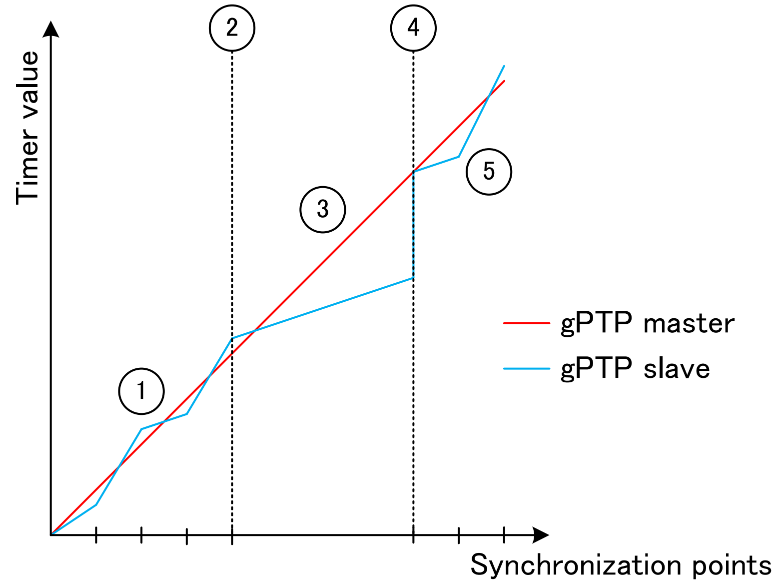
TAS module can be disabled during run-time. Figure 32.23 describes TAS behavior when its schedule is disabled. The schedule used in this figure is just an example and does not correspond to anything relevant for TAS general behavior explanation.



1	TAS is enabled and current schedule is ongoing
2	TAS is disabled
3	TAS is not enabled, all gates are opened

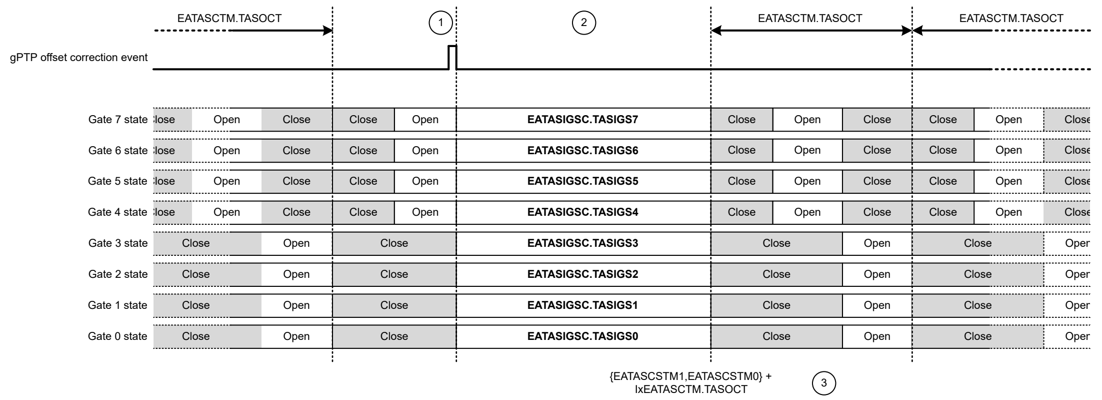
Figure 32.23 TAS behavior when its schedule is disabled

4. gPTP offset correction recovery [gPTP]
   gPTP timer control loop in normal condition corrects the gPTP timer using the gPTP incremental value. In this case the gPTP timer should be incremented for every clock by *clk_period* ns (clock clk period) for TAS to operate normally. However, when the gPTP timer is deviating from its corresponding master timer value, software might correct the gPTP offset value instead of the incremental value. Therefore, the gPTP timer and the AVTP timer can increase in time as described in Figure 32.24. In this case, TAS cannot operate normally and must recover as described in Figure 32.25.



1	The slave is synchronized to the master
2	Last successful synchronization point
3	Synchronization is lost. The gPTP slave is deriving
4	Synchronization is done successfully, the gPTP slave offset gets corrected.
5	The slave is synchronized to the master

Figure 32.24 gPTP offset correction



1	gPTP offset is corrected by the software so gPTP timer jumps in time
2	TAS recovery time. TAS needs time to calculate where it can restart safely.
3	TAS resynchronizes at {EATASCSTM1, EATASCSMC0} + IxEATASCTM.TASOCT time where I is an integer and the schedule loops every EATASCTM.TASOCT ns

Figure 32.25 TAS recovery during gPTP offset correction

TAS recovery time (*tas_recovery_time*) maximum value can be calculated by equation (6).

(6) $tas\_recovery\_time[ns] = clk\_period[ns] \times gptp\_jump\_time[ns] \,/\, cycle\_time[ns] + 2 \times cycle\_time[ns]$

Where:

- *clk_period* is the clock clk period

- *gptp_jump_time* is the time that has been jumped by the gPTP time
- *cycle_time* is the schedule cycle time contained in EATASCTM register.

Functions:

- For gPTP offset correction in the future, equation (6) comes from the fact that TAS switches its start time by *cycle_time* every clock during recovery. A problem resulting from that is that the recovery time increases proportionally with *tase_set_time[ns] – start_time[ns]*. To avoid this issue, TAS can switch its start time by $256^i \times$ *cycle_time* (i = 0 to 6) every clock when AVTP timer is too far in the future.

Restrictions:

- HW: Because of the gPTP offset correction, the hardware guard band cannot guarantee from the gPTP offset correction event time to the end of the first new cycle (frames can cross a gate close state)
- HW: Because of the gPTP offset correction, EAEIS1.TASGES gate errors can be set
- HW: If *gptp_jump_time* is too small (less than 2 × *cycle_time*), there is a possibility that the TAS cannot go back to its initial gate states but the recovery can still happen
- HW: For gPTP offset correction in the past, TAS cannot switch its start time by $256^i \times$ *cycle_time* (i = 0 to 6) every clock. Therefore, a gPTP offset correction too far in the past might result in a long recovery time.

32.5.1.6.2 TAS Schedule

TAS schedule defines the time of a cycle when a queue is allowed to transmit frames. A TAS schedule should be learned in the TAS RAM using TAS RAM learning functionality. This section describes the following information:

- TAS entry format
- TAS entry learning function
- TAS entry reading function
- Conversion between TAS schedule to a set of TAS entries.

1. TAS entry format
The TAS RAM is used to store TAS entries which compose the TAS schedule. A TAS entry contains the information of a gate state. All the fields in this table, if quoted, are written as TAS.{Field name}. Table 32.6 describes fields contained in a TAS entry.

Table 32.6 TAS entry format

Field name	Field size (bit)	Field description
GS	1	Gate state Values: ● 0: Gate state is closed ● 1: Gate state is opened
GT	28	Gate time Functions: ● Values of time in nanoseconds associated with the entry gate state TAS.GS

2. TAS entry learning
Learning is used to overwrite entries in the TAS RAM. Table 32.7 describes register used to learn an entry in the TAS RAM. There is no learning result because learning never fails.

Table 32.7 TAS entry learn registers (1 of 2)

Register name	Field name/corresponding field in TAS RAM	Field description
EATASGL0.TASGAL	Entry address	Not present in TAS RAM, new entry is written at this address
EATASGL1.TASGSL	TAS.GS	See section 32.5.1.6. TAS (Time Aware Shaper) [802.1Qbv].

Table 32.7 TAS entry learn registers (2 of 2)

Register name	Field name/corresponding field in TAS RAM	Field description
EATASGL1.TASGTL	TAS.GT	See section 32.5.1.6. TAS (Time Aware Shaper) [802.1Qbv].

3. TAS entry reading
Reading is used to read entries in the TAS RAM. Table 32.8 describes register used to read an entry in the TAS RAM and Table 32.9 describes the read results.

Table 32.8 TAS entry read registers

Register name	Field name/corresponding field in TAS RAM	Field description
EATASGR.TASGAR	Entry address	Not present in TAS RAM, entry is read from this address

Table 32.9 TAS entry read result

Register name	Field name/corresponding field in TAS RAM	Field description
EATASGRR.TASGSR	TAS.GS	See section 32.5.1.6. TAS (Time Aware Shaper) [802.1Qbv]
EATASGRR.TASGTR	TAS.GT	See section 32.5.1.6. TAS (Time Aware Shaper) [802.1Qbv]

4. Conversion between TAS schedule to a set of TAS entries

   (a) Schedule example
   Figure 32.26 describes a basic schedule example with basic gate behavior and recommendations while designing a schedule.

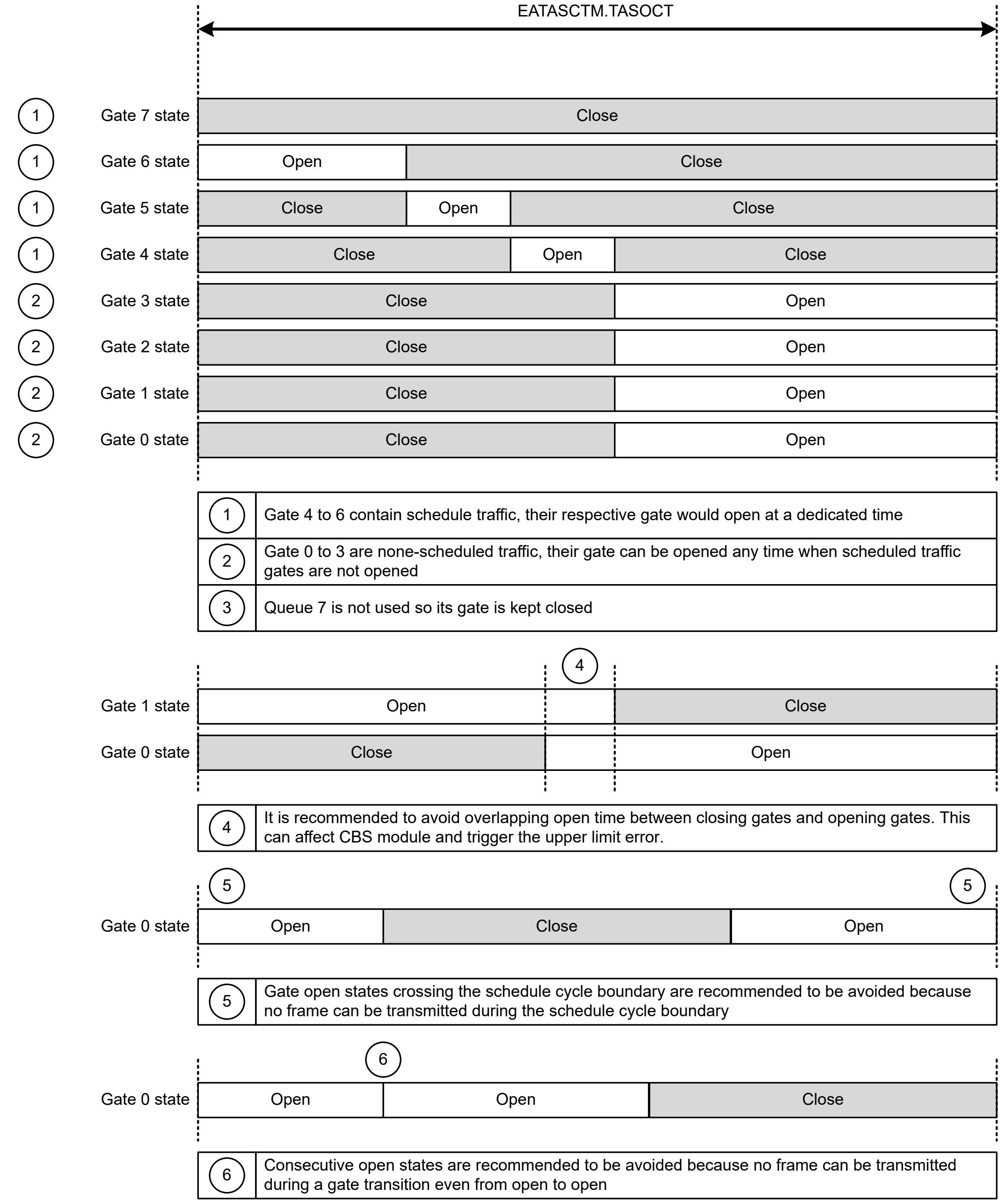


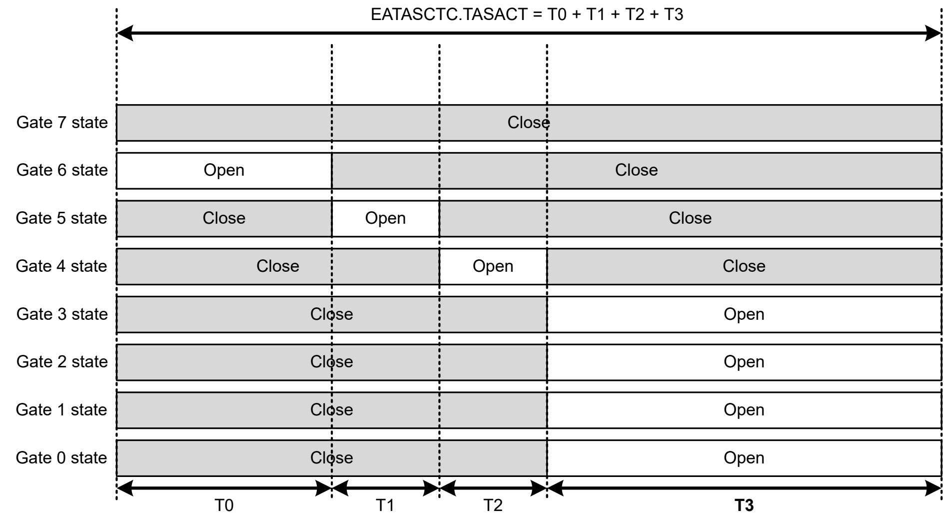
Figure 32.26 Schedule example and recommendations

Restrictions:

- SW: All p-frames queues (EATPEC.TTQ corresponding bit set to 1) should be opened simultaneously. For preemption, see section 32.5.1.7.2. Preemption.

(b) Schedule setting

Figure 32.27 describes how to convert a schedule into a set of TAS entries and how to set the schedule corresponding register through an example.



Corresponding gate	Entry number	TAS.GS	TAS.GT	Entry number
1 Gate 0 states	2 0	0	T0 + T1 + T2	EATASENC0.TASAEN0 = 1
	1	1	3 0	
Gate 1 states	2	0	T0 + T1 + T2	EATASENC1.TASAEN1 = 1
	3	1	0	
Gate 2 states	4	0	T0 + T1 + T2	EATASENC2.TASAEN2 = 1
	5	1	0	
Gate 3 states	6	0	T0 + T1 + T2	EATASENC3.TASAEN3 = 1
	7	1	0	
Gate 4 states	8	0	T0 + T1	4 EATASENC4.TASAEN4 = 2
	9	1	T2	
	10	0	0	
Gate 5 states	11	0	T0	EATASENC5.TASAEN5 = 2
	12	1	T1	
	13	0	0	
Gate 6 states	14	1	T0	EATASENC6.TASAEN6 = 1
	15	0	0	
Gate 7 states	16	0	5 0	EATASENC7.TASAEN7 = 0

1	Entries should be written in the TAS RAM from gate 0 to cut-through gate
2	For each gate, entries should be written along with their time in order of appearance
3	A gate corresponding to the last entry time should always be set to 0, the last gate is extended to the cycle end
4	A gate corresponding to an entry number register should be set as the number of entries - 1 used by the gate
5	If a gate has a constant state, the corresponding entry number should be set to 0 and the corresponding constant state should be set in TAS RAM with a gate time of 0

Figure 32.27 Schedule example

Restrictions:

- SW: Each gate time (T_0, T_1, T_2, T_3 in Figure 32.27) should be greater or equal to 50 ns + EATASHCC.TASJ.

5. Gate open time calculation
   Restrictions:

- SW: To avoid TAS module from clogging, gate open time should respect a minimum time value *minimumOpenTime* per queue. This time can be calculated by equation (7) for e-queues and equation (8) for p-queues (for preemption explanation, see section section 32.5.1.7.2. Preemption).
  (7) *minimumOpenTime[ns]* = (*maximumFrameSize[byte]* × 8 + 160) / *portTransmitRate[Gbps]* + *jitter[ns]* + *clk_period[ns]*
  (8) *minimumOpenTime[ns]* = (*MinimumFragmentSize[byte]* × *8* + 704) / *portTransmitRate[Gbps]* + *jitter[ns]* + *clk_period[ns]*

Where:

- *minimumOpenTime* is the smallest open time for a gate in the schedule.
- *maximumFrameSize* is the maximum frame size a queue can receive. Its value for descriptor queue q is equal to **EATMFSCq.MFS**.
- *portTransmitRate* is the ETHA link throughput [RMAC].
- *Jitter* is the internal jitter set in **EATASHCC.TASJ** register.
- *clk_period* is the clock clk period.
- *MinimumFragmentSize* is the minimum fragment size set in **EATPEC.AFS** register.

32.5.1.6.3 TAS Hardware Calibration

TAS transmission authorization is given to a frame before the frame read by TX data fetch block appends and before frame is sent to RMAC for transmission. Because of that, there is an internal minimum latency and an internal maximum jitter (Figure 32.28) that should be taken in account while setting TAS module. Latency calibration should be handled by software and jitter calibration should be set by software and handled by hardware. This section describes how to calibrate the TAS module in latency and jitter until the PHY interface [RMAC]. By adding, for example, the minimum latency and the maximum jitter until the next IP PSFP gate filtering module, it is possible to calibrate the TAS to it.

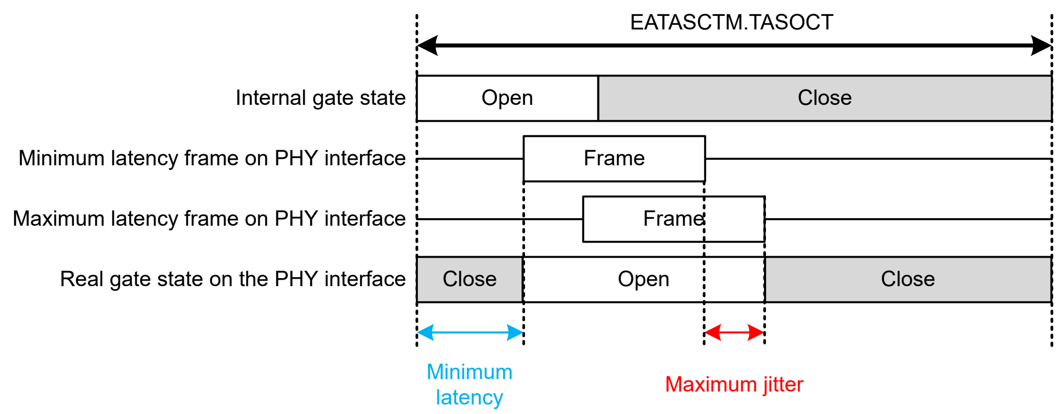


Figure 32.28 Minimum latency and maximum jitter

1. Latency calibration
   Because of the internal minimum latency, with a non-calibrated schedule, a frame always starts to transmit late on the PHY interface [RMAC]. To correct this, the TAS schedule should always be in advance compared to the expected schedule of the minimum latency between the TAS transmission decision and the actual transmission on the PHY interface (Figure 32.29). Table 32.10 describes this minimum latency. To calibrate the schedule, software should always subtract the minimum latency from the cycle start time before setting it in EATASCSTC0/1 registers.

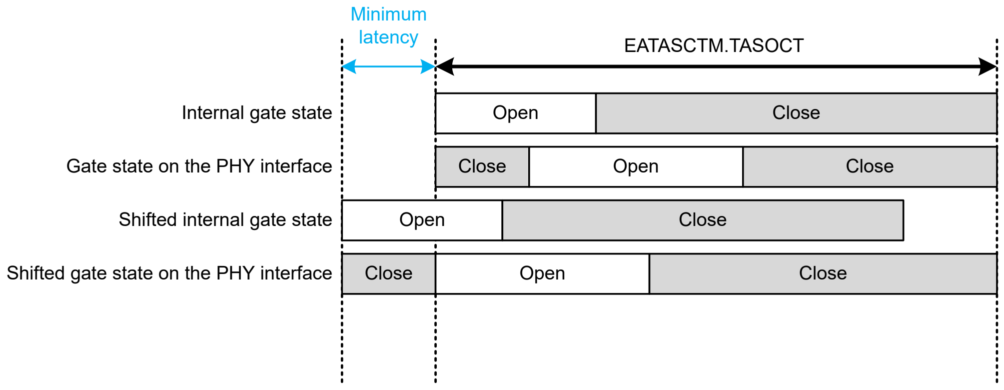


Figure 32.29 Minimum latency calibration

Table 32.10 Minimum latency values

PHY speed	Minimum latency when preemption disabled (EATPEC.TTQ set to All0)	Minimum latency when preemption enabled (at least one bit of EATPEC.TTQ set to 1)
10 Mbps	8 × *clk_period[ns]* + 10 × *clk_phy_tx_period[ns]* + *fabricLatency[ns]*	3 × *clk_period[ns]* + 10 × *clk_phy_tx_period[ns]* + *fabricLatency[ns]*
100 Mbps	8 × *clk_period[ns]* + 10 × *clk_phy_tx_period[ns]* + *fabricLatency[ns]*	3 × *clk_period[ns]* + 10 × *clk_phy_tx_period[ns]* + *fabricLatency[ns]*
1 Gbps	8 × *clk_period[ns]* + 10 × *clk_phy_tx_period[ns]* + *fabricLatency[ns]*	3 × *clk_period[ns]* + 10 × *clk_phy_tx_period[ns]* + *fabricLatency[ns]*

Where:

- *clk_period* is the clock clk period.
- *clk_phy_tx_period* is the clock clk_phy_tx period.
- *fabricLatency* is the latency induced by the fabric. See fabric specification document for calculation [MFAB].

2. Jitter calibration
   Because of the internal maximum jitter, with a non-calibrated schedule, a frame always finishes transmitting late on the PHY interface [RMAC]. To correct this, a TAS schedule opened gate should always be closed in advance compared to the expected close time of the maximum jitter (Figure 32.30). This maximum jitter should be calculated using equation (9). To calibrate the schedule, software should set the maximum jitter in EATASHCC.TASJ register.

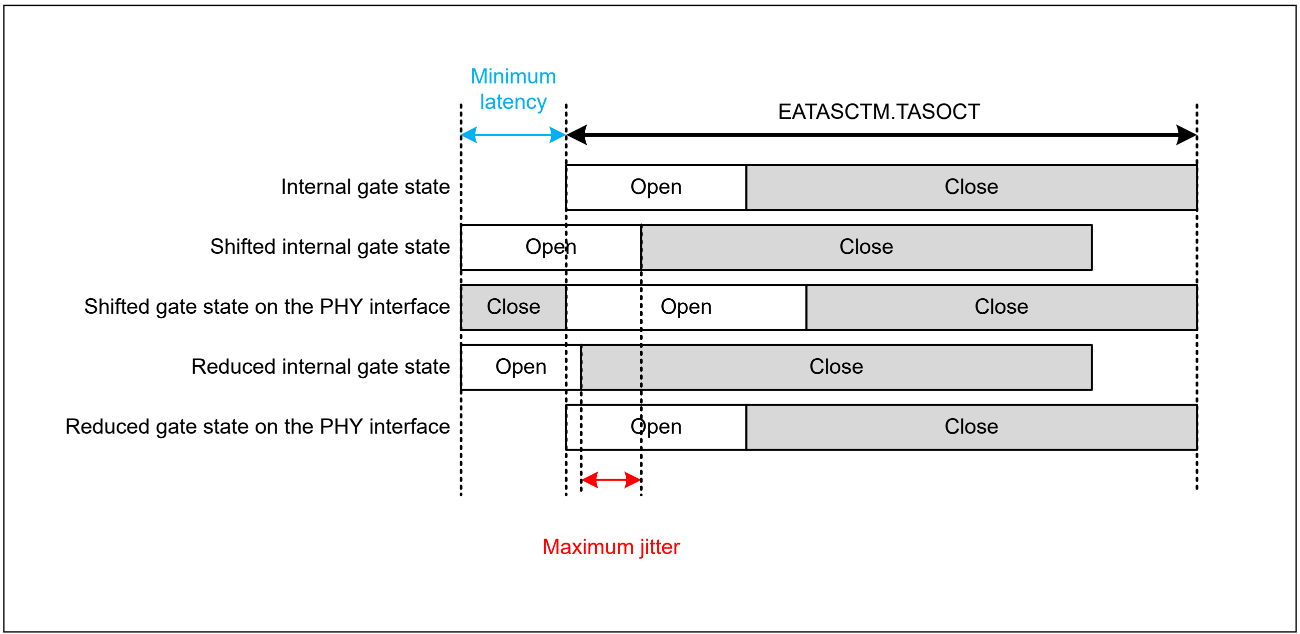


Figure 32.30 Maximum jitter calibration

(9) *jitter[ns] = tasInternalJitter[ns] + fabricJitter[ns] + preemptionJitter[ns] + asynchronousJitter[ns] + gptpSyncJitter[ns] + clkDerivationJitter[ns]*

Where:

- *tasInternalJitter* is the jitter induced by TAS module and is equal to 2 × *clk_period[ns]* where *clk_period* is the clock clk period.
- *fabricJitter* is the jitter induced by the fabric. See the Fabric specification for calculation [MFAB].
- *asynchronousJitter* is the jitter induced by the asynchronous conversion in RMAC [RMAC] and is equal to *clk_period[ns]* + *clk_phy_tx_period[ns]* where *clk_period* is the clock clk period and *clk_phy_tx_period*s the clock clk_phy_tx period.
- *gptpSyncJitter* is the jitter induced by gPTP synchronization between the gPTP master clock and the gPTP slave clock.
- *clkDerivationJitter* is the jitter induced by clock derivation between the gPTP clock clk and the PHY TX clock clk_phy_tx and is equal to *clkRelativeDerivation* × *maxGateTime[ns]* where *clkRelativeDerivation* is the clock derivation and *maxGateTime* is the longest opened gate in the TAS schedule. *clkRelativeDerivation* is equal to 0 when TAS corresponding gPTP timer is the master and the gPTP clock is phase synchronous to the TX PHY clock.
- *preemptionJitter* is the jitter induced by preemption and is equal to 5 × *clk_period[ns]* where *clk_period* is the clock clk period when preemption is enabled (at least one bit of EATPEC.TTQ register is set to 1) and 0 ns otherwise.

32.5.1.6.4 TAS Debug Interface

This interface aims at debugging the TAS gate states by giving it out of the IP.

Functions:

- eha_race_tas_gate_state[8] reflects the cut-through queue of the current TAS gate state.
- eha_race_tas_gate_state[7:0] reflects the current TAS gate states.

32.5.1.7 TX Data Fetch

TX data fetch is the same as RX data fetch in the GWCA except that it can handle preemption and also data fetching for cut-through queues. See GWCA specification for more details [GWCA] about data update for none cut-through frames. Table 32.11 describes the association between ETHA and GWCA registers/signals.

Table 32.11 TX data fetch ETHA/GWCA register correspondence

Register/signal name in GWCA [GWCA]	Register/signal name in ETHA
GWVCC.VEM	EAVCC.VEM
{GWMAC0.MAU, GWMAC1.MAL}	{MRMAC0.MAU, MRMAC1.MAL} [RMAC]
GWRGC.RCPT	Fixed to 1. RMAC adds an FCS to all frames sent without it.

32.5.1.7.1 Cut-through Frames

Cut-through frames handling by data fetched is transparent for TX data fetch. All cut-through frames has e-frames for transmission.

32.5.1.7.2 Preemption

Preemption functions allow express frames (e-frames) to interrupt a preemptable frame (p-frame) to start transmitting faster (a preemptable frame can also be interrupted by TAS gate close event) using the EATPEC register. Figure 32.31 describes a transmission between two e-queues, Figure 32.32 describes the same transmission but with the highest priority queue being a p-queue and Figure 32.33 describes how preemption occurs for a p-frame.

Functions:

- EATPEC.AFS is used to set the minimum fragment size
- EATPEC.TTQi is used to set descriptor queue q traffic to p-traffic or e-traffic.

Restrictions:

- HW: In Ethernet agent, preemption occurs on a 64-bit bus. As a result, preemption can only be performed every 16 bytes in a frame.
- HW: P-frames can only be preempted by e-frames with a higher priority. As a result, if p-frames are to be preempted by e-frames, always set low priority queues to p-frames and high priority queue to e-frames. The EAIRC register can be used to remap IPVs coming from MFWD to p-queues or e-queues.

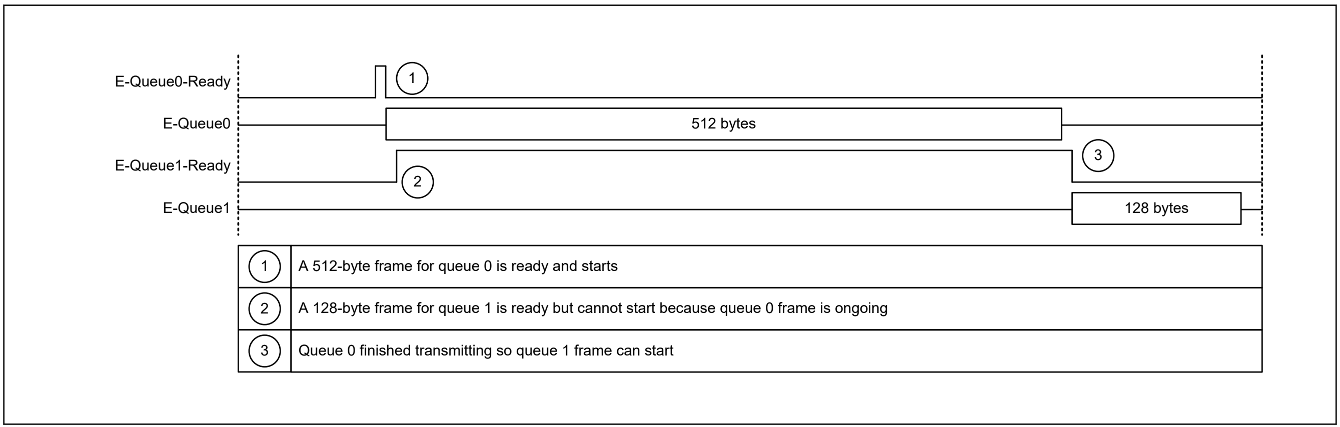


Figure 32.31 E-frame transmission

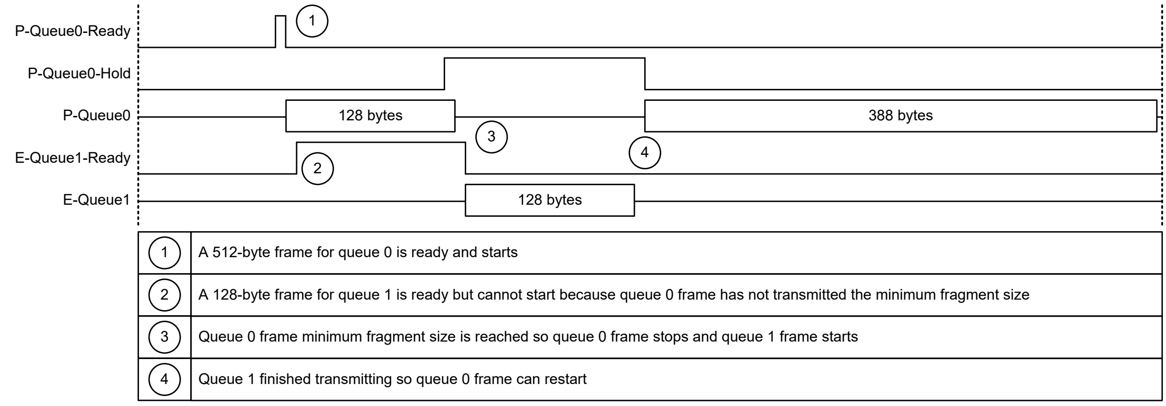


Figure 32.32 E-frame and p-frame transmission (EATPEC.AFS == 01)

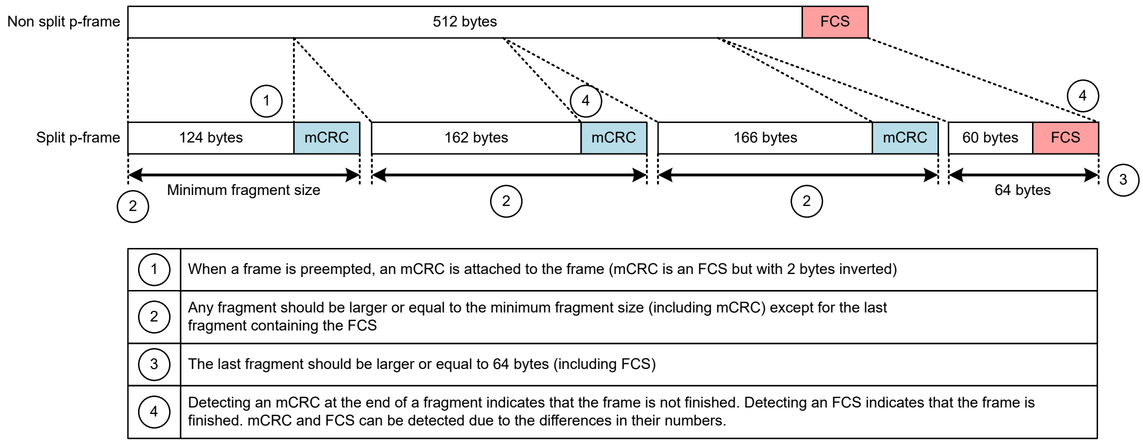


Figure 32.33 P-frame preemption (EATPEC.AFS == 01)

32.5.1.8 RMAC TX Interface Control

After TX data fetch, data is sent to the RMAC through MHD TX interface [RMAC]. Table 32.12 shows how MHD TX interface status information are provided to the RMAC.

Table 32.12 MHD TX interface status provided by ETHA to RMAC

Signal name in RMAC [RMAC]	Bus width	Value provided by ETHA
FCS in	1	Set to 1 if the corresponding transmit frame is an FCS (descriptor is received from forwarding engine with FDESCR.FI equal to 1) and if the frame has not been modified by the switch (no update due to VLAN tagging/ untagging and no update requested by L3 routing [FWD])
Timestamp capture	1	Set to LDESCR.TXC
Insert egress timestamp	1	Set to LDESCR.IET
Calculate residence time	1	Set to LDESCR.CRT
Timer number	1	Set to LDESCR.TN
Timestamp unique number	10	Set to {SPN, LDESCR.TSUN} where SPN is the source port number of the corresponding frame

32.5.2 Data Reception

Ethernet agent allows data reception through the ETHA RX data path. Figure 32.34 describes the RX data path.

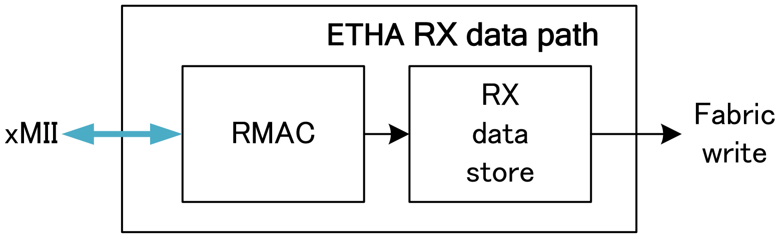


Figure 32.34 ETHA RX data path block diagram

The RX data path is separated in two blocks:

- RMAC: This block handles the data exchange with the Ethernet PHY. The functionalities of this block are described in the RMAC specification [RMAC].
- RX data store: This block extracts TAG information for frames, applies VLAN tagging and saves the frames in the local RAM. This block can also release pointers while exiting OPERATION mode. The pointer release exists for switch hardware purpose and its description is not required for switch utilization.

Restrictions:

- SW: The switch maximum input frame size is 60 KB. Set RMAC so it does not send frames larger than 60 KB (set MRFSCE.EMXS and MRFSCP.PMXS to a value smaller or equal to 61440 [RMAC]).

32.5.2.1 RX Data Store

The function of RX data store is the same as the function of TX data store in GWCA except for the local descriptor format. See GWCA specification for more details [GWCA]. Table 32.13 describes the association between ETHA and GWCA registers.

Table 32.13 Data store ETHA/GWCA register correspondence

Register/signal name in GWCA [GWCA]	Register/signal name in ETHA
GWTTFC	EARTFC
GWVCC.VIM	EAVCC.VIM
GWVTC	EAVTC
GWEIS0.USMFSES	EAEIS0.USMFSES

As local descriptor format, ETHA can only use ethernet local descriptors. Ethernet local descriptor is explained in GWCA specification and the field corresponding values for ethernet agent is described in Table 32.14.

Table 32.14 Ethernet local descriptor field description (1 of 2)

Field name	Bit width	Value for ETHA
FI	1	Set to RMAC MHD Rx Interface “FCS included” field [RMAC]
FMT	1	Set to 0 for ethernet descriptors
TXC	1	Set to RMAC MHD Rx Interface “Timestamp capture at TX side” field [RMAC]
IET	1	Set to 0
CRT	1	Set to 0
TN	1	Set to RMAC MHD Rx Interface “Time domain number” field [RMAC]
TSUN	8	Set to RMAC MHD Rx Interface “Timestamp unique number” field [RMAC]

Table 32.14 Ethernet local descriptor field description (2 of 2)

Field name	Bit width	Value for ETHA
SAEF	8	See Figure 32.35 and Table 32.15
VCTRL	3	VCTRL[2] values: ● Set to EAVCC.VIM VCTRL[1:0] values ● 00: For No TAG, R-TAG and Unknown TAG frames ● 01: For C-TAG and CR-TAG frames ● 10: For SC-TAG and SCR-TAG frames ● 11: For CoS-TAG and CoSR-TAG frames
RTGI	1	Values: ● 0: For No TAG, CoS-TAG, C-TAG, SC-TAG and Unknown TAG frames ● 1: For R-TAG, CoSR-TAG, CR-TAG and SCR-TAG frames
TPL	16	Payload length of frame. Only includes data saved in the data RAM [MFAB].
TSV	1	Values: ● 0: Timestamp has not been received along with the frame from RMAC [RMAC] ● 1: Timestamp has been received along with the frame from RMAC [RMAC]
TSD	1	Set to RMAC MHD Rx Interface “Timestamp is a default timestamp” field [RMAC]
TSNS	30	Set to 0 if timestamp has not been received. Set to timestamps nanosecond part if timestamp has not been received.
TSS	32	Set to 0 if timestamp has not been received. Set to timestamps second part if timestamp has not been received.
—	- -	Set to 0

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	EC1[1:0]		EC0[3:0]				—	TFE

Figure 32.35 Source agent error flag format

Table 32.15 Source agent error flag field description (1 of 2)

Field name	Bit width	Description
TFE	1	TAG Filtering Error Set conditions: ● HW: When a frame does not fit the required TAG ingress format. See section 32.3.3.3. EARTFC : Reception TAG Filtering Configuration Register.
—	- -	Reserved Set to 0

Table 32.15 Source agent error flag field description (2 of 2)

Field name	Bit width	Description
EC0	4	Error Code 0 [RMAC] Values: ● 0: No error ● 1: Frame received from RMAC with “PHY error” bit set ● 2: Frame received from RMAC with “PCH CRC error” bit set ● 3: Frame received from RMAC with “Nibble error” bit set ● 4: Frame received from RMAC with “FCS (mCRC) error” bit set ● 5: Frame received from RMAC with “Final fragment missing error” bit set ● 6: Frame received from RMAC with “C Fragment SMD Error” bit set ● 7: Frame received from RMAC with “C Fragment FRAG_COUNT Error” bit set ● 8: Frame received from RMAC with “RMAC Frame filtered” bit set ● 9: Frame received from RMAC with “Reception partially out of operation” bit set ● Others: Reserved Restrictions: ● HW: Because of the coding mechanism only one error can be flagged in this field ● HW: If several errors occur for the same frame, the error with the smallest EC0 value is flagged.
EC1	2	Error Code 1 [RMAC] Values: ● 0: No error ● 1: Frame received from RMAC with “Buffer overflow error” bit set ● 2: Frame received from RMAC with “Undersize error” bit set ● 3: Frame received from RMAC with “Oversize error” bit set Restrictions: ● HW: Because of the coding mechanism only one error can be flagged in this field ● HW: If several errors occur for the same frame, the error with the smallest EC0 value is flagged.
—	- -	Reserved Set to 0

33. Ethernet MAC (RMAC)

33.1 Overview

Ethernet MAC (RMAC) is an Ethernet controller that conforms to the definition of the MAC (Media Access Control) layer for Ethernet in the IEEE 802.3 standard. By connecting with a physical-layer LSI chip (PHY-LSI) that complies with the standard, RMAC can transmit and receive Ethernet (IEEE 802.3) frames. RMAC has a single MAC layer interface.

Table 33.1 lists the RMAC specifications and Figure 33.1 shows a block diagram.

Table 33.1 RMAC specifications

Item	Description
Protocol	Flow control conforming with the IEEE 802.3x standard
	Priority-based Flow Control with 802.1Qbb
	Interspersing Express Traffic (preemption) conforming with IEEE 802.3br
	Support for IEEE 1588 time synchronization, Support for IEEE AS and IEEE AS-Rev acceleration
Transfer speed	Supports transfer at 10,100,1000, 2500Mbps
Mode	Full-duplex mode
	Energy Efficient Ethernet (Low Power Idle)
Interface	Supports the IEEE 802.3 standard MII (Media Independent Interface), GMII (Gigabit Media Independent Interface)
	PHY MDIO programming Interface
	Supports internal register interface (32bit data width) Supports internal data interface (64bit data width)
Original function	Configurable frame data padding A variety of flexible RX address filtering modes Configurable CRC pass through for transmit and receive Network statistic counter (IEEE 802.3, ROMN RFC2819)
Magic Packet™	Detection of Magic Packet™*1 and output of a detected signal

Note 1. Magic Packet™ is a trademark of Advanced Micro Devices, Inc.

Table 33.2 References

Title	Contents
IEEE Std 802.3™-2008	Carrier sense multiple access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications
IEEE 802.3az™-2010	Media Access Control Parameters, Physical Layers, and Management Parameters for Energy-Efficient Ethernet
IEEE Std 802.3br™-2016	Specification and Management Parameters for Interspersing Express Traffic
P802.1AS-Rev™	Timing and Synchronization for Time Sensitive Applications
P802.1Qbu™	Frame Preemption
IEEE Std 1588™-2008	Precision Clock Synchronization Protocol for Networked Measurement and Control Systems
IEEE Std 802.1BA™-2011	Audio Video Bridging (AVB) Systems
IEEE802.1Qbb	Priority-based Flow Control

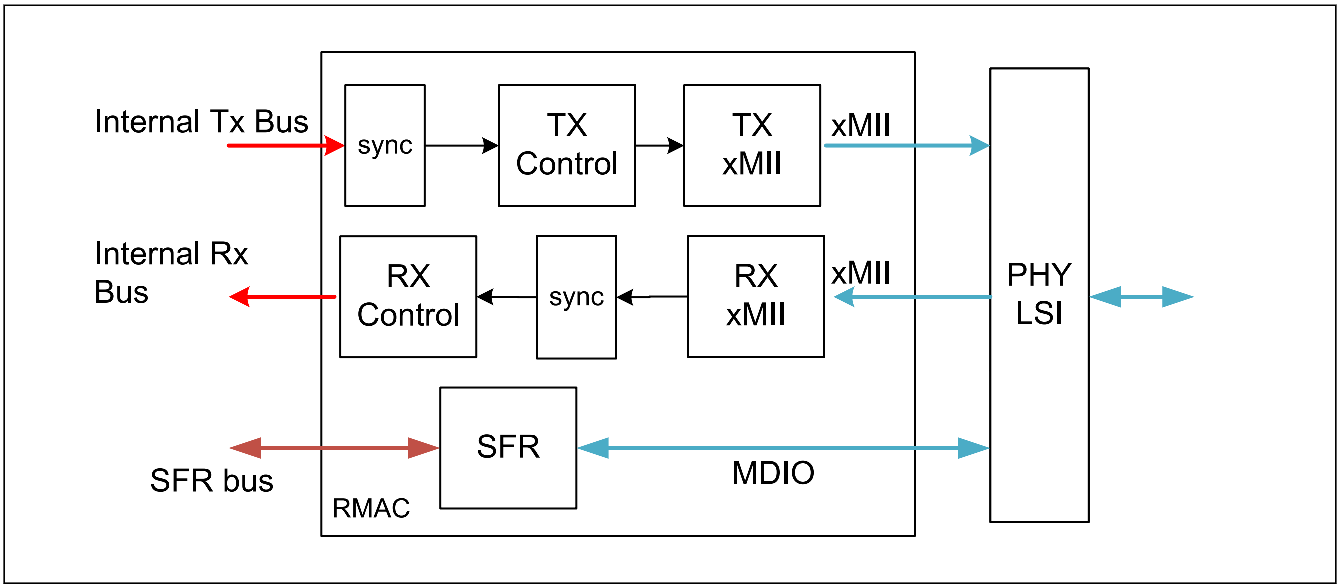


Figure 33.1 RMAC block diagram

Table 33.3 RAC functional blocks

Block name	Function
SFR	Contains all the RMAC SFRs and the bridging system to access PHY registers. RMAC registers are on clk.
Rx Control	Consists of the receive FSM and Data path logic, which disassembles the received Ethernet frame. It controls functions like checking and removing FCS, Destination Address filtering, Flow Control etc.
Tx Control	Consists of the transmission FSM and Data path logic, which facilitates the transmitted Ethernet frame. It controls functions like adding the Preamble and SFD, adding PAD bytes, calculating the FCS etc.
Rx xMII / Tx xMII	Provides the connectivity to the PHY. It supports several types of xMII (MII, GMII).
RMAC	IP described in this documentation.
Sync	Synchronized between bus clock domain and mii clock domain.

33.2 Pin Function

33.2.1 Pin Description

33.2.1.1 PHY Interfaces

PHY interface supports several Media Interfaces. Each media interface has its own interface signals.

PHY interface are described in “IEEE Std 802.3™-2008”

IPG (inter packet gap) of RMAC is defined as follows.

MII TYPE	Tx/Rx	IPG
GMII	Tx	Fixed 12 byte
	Rx	Minimum 12 byte
MII	Tx	Fixed 12 byte
	Rx	Minimum 12 byte

33.2.1.2 MDIO Interface

See MDIO standard “IEEE Std 802.3™-2018”.

33.2.1.3 MHD TX Timestamp Capture Interface

If a timestamp capture is request to RMAC on TX side (by SW or HW), this interface is used to transmit the timestamp.

This interface is based on a handshake, the interface time diagram is described in the figure bellow.

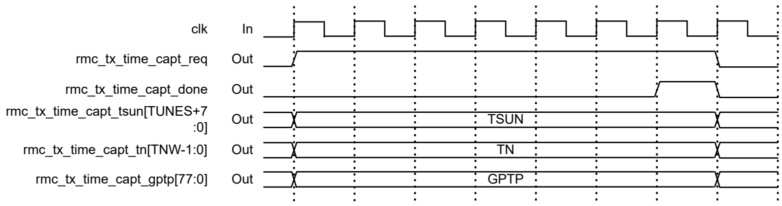


Figure 33.2 TX capture interface time chart

33.3 RMAC Register List

Table 33.4 RMAC registers (1 of 3)

Offset address	Register name	Register symbol	Write access mode
0x0000	PHY Station Management Register	MPSM	D, C, O
0x0004	PHY Interfaces Configuration Register	MPIC	C
0x0008	PHY Interfaces Monitoring Register	MPIM	—
0x0010	I/O Configuration Register	MIOC	D, C, O
0x0020	Transmission Frame Format Configuration Register	MTFFC	C
0x0024	Transmission Pause or PFC Frame Configuration Register	MTPFC	C
0x0028	Transmission Pause or PFC Frame Configuration Register 2	MTPFC2	C, O
0x0030 + 0x04 × t	Transmission Pause or PFC Frame Configuration Register 3 for Priority Group t	MTPFC3t	C
0x0080	Reception General Configuration Register	MRGC	C
0x0084	Reception MAC Address Configuration Register 0	MRMAC0	C
0x0088	Reception MAC Address Configuration Register 1	MRMAC1	C
0x008C	Reception Address Filter Configuration Register	MRAFC	C, O
0x0090	Reception Storm Configuration for E-Frames Register	MRSCE	C, O
0x0094	Reception Storm Configuration for P-Frames Register	MRSCP	C, O
0x0098	Reception Storm Counter Configuration Register	MRSCC	D, C, O
0x009C	Reception Frame Size Configuration for E-Frames Register	MRFSCE	C
0x00A0	Reception Frame Size Configuration for P-Frames Register	MRFSCP	C
0x00A4	Timestamp Reception Configuration Register	MTRC	C
0x00AC	Reception Pause or PFC Frame Monitoring Register	MRPFM	—
0x0100 + 0x04 × t	PTP Filtering Register Configuration Register t	MPFCt	C
0x0180	Link Verification Configuration Register	MLVC	O
0x0184	Energy Efficient Ethernet Configuration Register	MEEEC	D, C, O
0x0188	Loopback Configuration Register	MLBC	C
0x0200	Error Interrupt Status Register	MEIS	D, C, O

Table 33.4 RMAC registers (2 of 3)

Offset address	Register name	Register symbol	Write access mode
0x0204	Error Interrupt Enable Register	MEIE	D, C, O
0x0208	Error Interrupt Disable Register	MEID	D, C, O
0x0210	Monitoring Interrupt Status Register 0	MMIS0	D, C, O
0x0214	Monitoring Interrupt Enable Register 0	MMIE0	D, C, O
0x0218	Monitoring Interrupt Disable Register 0	MMID0	D, C, O
0x0220	Monitoring Interrupt Status Register 1	MMIS1	D, C, O
0x0224	Monitoring Interrupt Enable Register 1	MMIE1	D, C, O
0x0228	Monitoring Interrupt Disable Register 1	MMID1	D, C, O
0x0230	Monitoring Interrupt Status Register 2	MMIS2	D, C, O
0x0234	Monitoring Interrupt Enable Register 2	MMIE2	D, C, O
0x0238	Monitoring Interrupt Disable Register 2	MMID2	D, C, O
0x0300	Manual Pause Frame Transmit Counter Register	MMPFTCT	—
0x0304	Automatic Pause Frame Transmit Counter Register	MAPFTCT	—
0x0308	Pause Frame Receive Counter Register	MPFRCT	—
0x030C	False Carrier Indication Counter Register	MFCICT	—
0x0310	Energy Efficient Ethernet Counter Register	MEEECT	—
0x0320 + 0x04 × j	Manual PFC Frame Transmit Counter Register	MMPCFTCTt	—
0x0330 + 0x04 × j	Automatic PFC Frame Transmit Counter Register	MAPCFTCTt	—
0x0340 + 0x04 × j	PFC Frame Receive Counter Register	MPCFRCTt	—
0x0360	Receive Overflow Counter Register	MROVFC	—
0x0408	Received Good Frame Counter E-Frames Register	MRGFCE	—
0x040C	Received Good Frame Counter P-Frames Register	MRGFCP	—
0x0410	Received Good Broadcast Frame Counter Register	MRBFC	—
0x0414	Received Good Multicast Frame Counter Register	MRMFC	—
0x0418	Received Good Unicast Frame Counter Register	MRUFC	—
0x041C	Received PHY Error Frame Count Register	MRPEFC	—
0x0420	Received Nibble Error Frame Count Register	MRNEFC	—
0x0424	Received FCS/mCRC Error Frame Count Register	MRFMEFC	—
0x0428	Received Final Fragment Missing Error Frame Count Register	MRFFMEFC	—
0x042C	Received C-Fragment Count Error Frame Count Register	MRCFCEFC	—
0x0430	Received Fragment Count Error Frame Count Register	MRFCEFC	—
0x0434	Received RMAC Filter Error Frame Count Register	MRRCFEFC	—
0x0438	Received Frame Count Register	MRFC	—
0x043C	Received Good Undersize Error Frame Count Register	MRGUEFC	—
0x0440	Received Bad Undersize Error Frame Count Register	MRBUEFC	—
0x0444	Received Good Oversize Error Frame Count Register	MRGOEFC	—
0x0448	Received Bad Oversize Error Frame Count Register	MRBOEFC	—
0x044C	Received Byte Counter E-Frames Upper Side Register	MRXBCEU	—
0x0450	Received Byte Counter E-Frames Lower Side Register	MRXBCEL	—
0x0454	Received Byte Counter P-Frames Upper Side Register	MRXBCPU	—

Table 33.4 RMAC registers (3 of 3)

Offset address	Register name	Register symbol	Write access mode
0x0458	Received Byte Counter P-Frames Lower Side Register	MRXBCPL	—
0x0508	Transmitted Good Frame Counter E-Frames Register	MTGFCE	—
0x050C	Transmitted Good Frame Counter P-Frames Register	MTGFCP	—
0x0510	Transmitted Broadcast Frame Counter Register	MTBFC	—
0x0514	Transmitted Multicast Frame Counter Register	MTMFC	—
0x0518	Transmitted Unicast Frame Counter Register	MTUFC	—
0x051C	Transmitted Error Frame Counter Register	MTEFC	—
0x0520	Transmitted Byte Counter E-Frames Upper Side Register	MTXBCEU	—
0x0524	Transmitted Byte Counter E-Frames Lower Side Register	MTXBCEL	—
0x0528	Transmitted Byte Counter P-Frames Upper Side Register	MTXBCPU	—
0x052C	Transmitted Byte Counter P-Frames Lower Side Register	MTXBCPL	—

Note:
- Base address: RMAC0 = 0x403C_B000, RMAC1 = 0x403C_D000, RMAC0_NS = 0x503C_B000, RMAC1_NS = 0x503C_D000
- All registers can be read in any mode.
- Write Access Mode:
  - Any: Register can be accessed in any mode.
  - D: Register can be accessed in DISABLE mode.
  - C: Register can be accessed in CONFIG mode.
  - O: Register can be accessed in OPERATION mode.

33.4 RMAC Register Descriptions

33.4.1 RMAC Function Registers

33.4.1.1 MPSM : PHY Station Management Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PRD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	PSME*2*3*4	PHY Station Management Enable 0: PHY register access is not on progress. 1: PHY register access is requested and on progress.	R/W*1
1	—	This bit is read as 0.	R

Bit	Symbol	Function	R/W
2	MFF	Management Frame Format 0: Normal management frame format defined in Clause 22 of IEEE802.3 Describe "MDIO" in this document. 1: Extension management frame format defined in Clause 45 of IEEE802.3 Describe "eMDIO" in this document.	R/W
7:3	PDA[4:0]	PHY Device Address These bits specify following value according to MPSM.MFF. See 8.5 for more details. ● Case of MPSM.MFF = 0: The PHY Address field ● Case of MPSM.MFF = 1: The PHY Address field	R/W
12:8	PRA[4:0]	PHY Register Address These bits specify following value according to MPSM.MFF. See 8.5 for more details. ● Case of MPSM.MFF = 0: Register Address ● Case of MPSM.MFF = 1: Device address	R/W
14:13	POP[1:0]	PHY Operation Code These bits specify the operation code. See 8.5 for more details. These bits specify following value according to MPSM.MFF. ● Case of MPSM.MFF = 0 0 1: Write Frame 1 0: Read Frame Others: Reserved (0 0 or 1 1 should not be set when MPSM.MFF = 0) ● Case of MPSM.MFF = 1 0 0: Address Frame 0 1: Write Frame 1 0: Post-read-increment-address frame 1 1: Read Frame	R/W
15	—	This bit is read as 0.	R
31:16	PRD[15:0]	PHY Register Data These bits specify following value according to MPSM.MFF. See 8.5 for more details. ● Case of MPSM.MFF = 0: Data field of the PHY register ● Case of MPSM.MFF = 1: Data/Address filed of the PHY register	R/W*1

Note 1. Read value differs from written value.
Note 2. Accessing to MPSM register while this bit is asserted has no effect.
Note 3. This bit cannot be set when MPIC.PSMCS = 0.
Note 4. Before resetting RMAC, this register should always be checked. If this register is set, no reset should be performed.

Provides access to the PHY-LSI internal registers via the MDIO.

PSME bit (PHY Station Management Enable)

Set this bit to access PHY registers.

[Setting condition]

SW: By writing 1 to this register. It starts phy access.

[Clearing condition]

HW: When PHY register access is completed, this bit returns automatically to 0.

PRD[15:0] bits (PHY Register Data)

Write operation: HW send register value to PHY in write operation.

Read operation: HW write read data to this register from PHY in read operation.

33.4.1.2 MPIC : PHY Interfaces Configuration Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	PSMCT[2:0]			—	PSMHT[2:0]			PSMDP	PSMCS[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	PIS[2:0]	PHY Interface Select These bits select the xMII interface connected to the PHY. 0 0 0: MII 0 1 0: GMII 1 0 0: Reserved 1 0 1: Reserved 1 1 0: Reserved 1 1 1: Reserved	R/W
5:3	LSC[2:0]	Link Speed Configuration 0 0 0: 10mbps 0 0 1: 100mbps 0 1 0: 1gbps 0 1 1: Reserved 1 1 0: Reserved 1 1 1: Reserved	R/W
7:6	—	These bits are read as 0.	R
8	PIP	PHY Interrupt Polarity This bit sets the active level of the PHY interrupt pin (rmc_phy_int). For the active level, refer to the specifications of PHY-LSI to be connected. 0: PHY interrupt is active low (default). 1: PHY interrupt is active high.	R/W
9	PIPP	PHY Interrupt Pin Plugged 0: Unplugged 1: Plugged	R/W
10	PLSPP	PHY Link Status Pin Plugged 0: Unplugged 1: Plugged	R/W
15:11	—	These bits are read as 0.	R
22:16	PSMCS[6:0][*2]	PHY Station Management Clock Selection 0000000: MDC clock is off 0000001: clk / ((1+1)*2) 0000010: clk / ((2+1)*2) ⋮ 1111110: clk / ((126+1)*2) 1111111: clk / ((127+1)*2)	R/W[*1]
23	PSMDP[*2]	PHY Station Management Disable Preamble 0: PHY Station Management Preamble enabled 1: PHY Station Management Preamble disabled	R/W[*1]
26:24	PSMHT[2:0][*2*3]	PHY Station Management Hold Time Adjustment 0 0 0: No adjusted hold time (i.e. change MDO at the rising edge of MDC) 0 0 1: 1 clk cycle extra hold time ⋮ 1 1 1: 7 clk cycle extra hold time	R/W[*1]

Bit	Symbol	Function	R/W
27	—	This bit is read as 0.	R
30:28	PSMCT[2:0][*2*3]	PHY Station Management Capture Time Adjustment The configured value will shift the data capture timing of MDI with respect to the rising edge of MDC clock. 0 0 0: No adjusted capture time (i.e. capture MDI at the rising edge of MDC) 0 0 1: Capture before 1 clk cycle ⋮ 1 1 1: Capture before 7 clk cycle	R/W[*1]
31	—	This bit is read as 0.	R

Note 1. Read value differs from written value.
Note 2. SW: This register cannot be set if MPSM.PSME is set.
Note 3. SW: The configured value must be less than MDC clock cycle (less than MPIC.PSMCS*2)

Configures MDIO and xMII interface.

PSMCS[6:0] bits (PHY Station Management Clock Selection)

These bits configure the cycle of MDC clock. MDC clock is divided from clk clock. The duty cycle of MDC clock is 50%.

See to section 8.4 for detail.

PSMDP bit (PHY Station Management Disable Preamble)

This bit could cut the preamble pattern (sequence of 32 contiguous bits). When it is asserted, the preamble pattern is skipped and the management frame starts with the ST (Start of Frame) pattern.

PSMHT[2:0] bits (PHY Station Management Hold Time Adjustment)

The configured value will shift the data change timing of MDO with respect to the rising edge of MDC clock.

33.4.1.3 MPIM : PHY Interfaces Monitoring Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0008

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

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

Bit	Symbol	Function	R/W
0	PLS	PHY Link Status Flag 0: PHY link signal is at the low level. 1: PHY link signal is at the high level.	R
1	LPIA	LPI Active This bit indicates that LPI is active on RX side. 0: No LPI 1: LPI active	R
31:2	—	These bits are read as 0.	R

PLS bit (PHY Link Status Flag)

This bit indicates PHY link state. It could be used only when the link status pin (rmc_phy_link) is connected to the link signal from PHY-LSI.

For the active level, refer to the specifications of PHY-LSI to be connected.

LPIA bit (LPI Active)

[Setting condition]

HW: When LPI code is appeared on detected on xMII Rx data signals (rmc_rx_data_valid_in, rmc_rx_data_data_in[31:0], rmc_rx_data_err_in). For more details, see xMII specification.

[Clearing condition]

HW: When LPI code is not appeared on detected on xMII Rx data signals.

33.4.1.4 MIOC : I/O Configuration Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0010

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MIOC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MIOC0	I/O Configuration 0 0: Normal mode. clk_phy_tx and clk_phy_rx are provided. 1: Emergency clock recovery mode. clk_phy_tx and clk_phy_rx are provided free-run clock (clk_common). (So transmission is not possible in this state.)	R/W
31:1	—	These bits are read as 0.	R

Note: SW can write 1 to bit[0] when clock derivation from PHY is unexpectedly stopped.
In this case, write 1 to bit[0] to flush all internal condition in RMAC. After waiting to flush and change DISABLE mode, write 0 to bit[0] and change RESET mode.

33.4.1.5 MTFFC : Transmission Frame Format Configuration Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0020

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

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

Bit	Symbol	Function	R/W
0	DPAD*1*2	Data Padding Disable Disables the automatic padding of too small transmit frames (less than 60 bytes) with no FCS If a too small transmit frame (less than 60 bytes) with no FCS is received when this bit is set, RMAC adds FCS and sends such a frame. 0: Padding is inserted. 1: Padding is not inserted.	R/W

Bit	Symbol	Function	R/W
1	FCM	Flow Control Mode This bit specifies the flow control mode 0: PAUSE 1: PFC	R/W
31:2	—	These bits are read as 0.	R

Note 1. HW: A Verify frame, a Pause frame and a PFC frame are not affected by this bit.
Note 2. HW: Padding is not executed when “fcs included(FI==1)” indication is applied by upper layer.

The register is used to configure the frame format to be transmitted.

33.4.1.6 MTPFC : Transmission Pause or PFC Frame Configuration Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0024

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PFRLV[4:0]					PFM	—	—	PFRT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
15:0	PT[15:0]*1	Pause Time These bits are common bit for PAUSE and PFC. 0x0000 Pause frame disable : 0x0001 512 × 1 bit-time : : 0xFFFF 512 × 65,535 bit-time :	R/W
23:16	PFRT[7:0]*1*2	Pause Frame Retransmission Time These bits are common bit for PAUSE and PFC. 0x00: 512 × 0 bit-time 0x01: 512 × 1 bit-time : 0xFF: 512 × 255 bit-time	R/W
25:24	—	These bits are read as 0.	R
26	PFM*1	Pause Frame Mode This bit is common bit for PAUSE and PFC. 0: Automatic pause frame. Sending a pause frame will be triggered by a hardware pause request. 1: Manual pause frame. Sending a Pause frame will be triggered by MTPFC2.MPFR or MTPFC2.MPFCFR.	R/W
31:27	PFRLV[4:0]*1	Pause or PFC Frame Retry Limit Value 0x00: No interrupt 0x01: Interrupt at 1 pause frame : 0x1F: Interrupt at 31 pause frames	R/W

Note 1. SW: This register is only writable in CONFIG mode
Note 2. SW: This register should be set to a smaller value than one of MTPFC.PT.

The register is used to configure the TIME parameter value for pause frames.

PT[15:0] bits (Pause Time)

These bits set the TIME parameter value for pause frames.

- The value of these bits is set to Time field of a PAUSE or PFC frame.
- If the value of these bits is 0, PAUSE of PFC is not sent.
- The unit for the setting is 512 bit-time. 1 bit-time is equal to 1ns in 1000 Mbps, 10 ns in 100 Mbps, 100 ns in 10 Mbps.

PFRT[7:0] bits (Pause Frame Retransmission Time)

These bits set the TIME parameter value of pause frame retransmission.

- A pause frame will be retransmitted when the previous frame has been sent more than (PT-PFRT) ± 10 cycles bit-time ago.
- The unit for the setting is 512 bit-time. 1 bit-time is equal to 1ns in 1000 Mbps, 10 ns in 100 Mbps, 100 ns in 10 Mbps.

PFRLV[4:0] bits (Pause or PFC Frame Retry Limit Value)

Value of pause frame sent without pause frame request signal deassertion from which at which RMAC will emit an interrupt.

This bit is common bit for PAUSE and PFC.

The number of pause (pfc) frames in this field indicates the total number of transmission attempt.

33.4.1.7 MTPFC2 : Transmission Pause or PFC Frame Configuration Register 2

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0028

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	MPFCFR1 to MPFCFR0		—	—	—	—	—	—	PFCTTZ1 to PFCTTZ0	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PFCTTZ1 to PFCTTZ0*1*2	PFC Frame n Transmission with TIME = 0 (n = 0, 1) 0: The transmission of PFC frames with TIME value zero is disabled. 1: The transmission of PFC frames with TIME value zero is enabled.	R/W
7:2	—	These bits are read as 0.	R
9:8	MPFCFR1 to MPFCFR0*3	Manual PFC Frame n Request (n = 0, 1)	R/W
15:10	—	These bits are read as 0.	R
16	PFTTZ*2	Pause Frame Transmission with TIME = 0 0: The transmission of Pause frames with TIME value zero is disabled. 1: The transmission of Pause frames with TIME value zero is enabled.	R/W
17	MPFR*3	Manual Pause Frame Request	R/W
31:18	—	These bits are read as 0.	R

Note 1. SW: This bit has no effect for Manual PFC frame transmission.
Note 2. SW: This register is only writable in CONFIG mode.
Note 3. SW: This register is only writable in OPERATION mode.

The register is used to set the TIME parameter value for PFC frames.

PFCTTZn bits (PFC Frame n Transmission with TIME = 0 (n = 0, 1))

This bit enables the transmission of automatic PFC frames with TIME value zero.

It is transmitted when the MHD negated the PFC request in automatic PFC frame mode or when the software negate MTPFC2.

MPFCFR register in manual PFC frame mode and the remaining PFC time is more than PFRT.

MPFCFRn bits (Manual PFC Frame n Request (n = 0, 1))

Writing 1 to this bit request manual pfc frame.

If a pfc frame is requested so fast that the hardware did not have time to send the previous one (because a long frame is ongoing or because the software requests too many pfc frames), the hardware will ignore the pfc frame requests.

After completion of non-pause frame transmission the next attempt for PFC frame will be only when timer expires if request for PFC frame is still high.

[Setting condition]

SW: Writing 1 to this bit will set.

[Clearing condition]

SW: Writing 0 to this bit will clear.

HW: Going out of OPERATION mode will clear this register.

PFTTZ bit (Pause Frame Transmission with TIME = 0)

This bit is used for PAUSE frame only.

This bit enables the transmission of automatic Pause frames with TIME value zero. It is transmitted when the MHD negates the pause request in automatic pause frame mode or when the software negates MTPFC2.MPFR register in manual pause frame mode and the remaining pause time is more than PFRT.

This bit has no effect for Manual Pause frame transmission.

MPFR bit (Manual Pause Frame Request)

This bit is used for a PAUSE frame only.

Writing 1 to this bit requests a manual pause frame.

If a pause frame is requested so fast that the hardware has time to send the previous one (because a long frame is ongoing or because the software requests too many pause frames), the hardware will ignore the pause frame requests.

[Setting condition]

SW: Writing 1 to this bit will set the bit

[Clearing condition]

SW: Writing 0 to this bit will clear the bit.

HW: Going out of OPERATION mode will clear this register.

33.4.1.8 MTPFC3t : Transmission Pause or PFC Frame Configuration Register 3 for Priority Group t

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0030 + 0x4 × t

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	PFCPG7 to PFCPG0							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	PFCPG7 to PFCPG0*1	PFC Priority Enable n (n = 0 to 7) 0: Priority x is not assigned to priority group t. 1: Priority x is assigned to group t.	R/W
31:8	—	These bits are read as 0.	R

Note 1. SW: This register is only writable in CONFIG mode.

The register is used to set the TIME parameter value for PFC frames.

PFCPGn bits (PFC Priority Enable n (n = 0 to 7))

Function when MTFFC.FCM=1(PFC)

- When PFC frame transmission request (hardware request for automatic request, MTPFC2.MPFCFR for manual request) is set, PFC frame is generated. Then, the Field of Class-Enable vector is set to this register value.
- When all bits of PFCPGn = 0, the transmission request is ignored.

Function when MTFFC.FCM = 0 (PAUSE)

- PAUSE frame transmission request is set by hardware pause request or MTPFC2.MPFR.
- When PFCPG0 = 0, the transmission request is ignored.

33.4.1.9 MRGC : Reception General Configuration Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0080

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	PFCRC7 to PFCRC0							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	RCPT	Receive CRC Pass Through This bit selects to pass FCS field of the received frame. The CRC is checked regardless of this configuration. 0: Correct FCS is not passed to the MHD. Incorrect FCS is passed to the MHD. 1: Both Correct and Incorrect FCS are passed to the MHD.	R/W
1	PFRC*1*2	Pause Frame Reception Control This bit enables the Flow Control (Pause frame detection) function. 0: Flow control for the receiving port is disabled. 1: Flow control for the receiving port is enabled.	R/W
2	PFRTZ	Pause or PFC Frame Reception with Time = 0 When this bit is set to 1, the reception of Pause or PFC frame with the Timer value 0 releases the transmission wait state. 0: Reception of Pause or PFC frames with the TIME value 0 is disabled. 1: Reception of Pause or PFC frames with the TIME value 0 is enabled.	R/W
3	MPDE	Magic Packet™ Detection Enable This bit enables Magic Packet™ detection by hardware to allow activation via the Ethernet connection. (Only for e-frames) 0: Magic Packet™ detection is disabled. 1: Magic Packet™ detection is enabled.	R/W
4	RFCFE	Reception Flow Control Forwarding Enable This bit enables the Flow Control Frame (PAUSE or PFC frame) forwarding function. 0: Flow Control Frame (PAUSE or PFC frame) is not forward to Rx MHD I/F. 1: Flow Control Frame (PAUSE or PFC frame) is forward to Rx MHD I/F.	R/W
15:5	—	These bits are read as 0.	R
23:16	PFCRC7 to PFCRC0*1*3	PFC Frame Reception Control n (n = 0 to 7) This bit enables the Flow Control (Pause frame detection) function. 0: Flow control for the receiving priority level x (8) is disabled. 1: Flow control for the receiving priority level x (8) is enabled.	R/W
31:24	—	These bits are read as 0.	R

Note 1. HW: Only for e-frames
Note 2. SW: When any bits of MRGC.PFCRC is set, SW should not set this bit.
Note 3. SW: When MRGC.PFRC is set, SW should not set this bit.

The resister is used to configure the frame reception.

33.4.1.10 MRMAC0 : Reception MAC Address Configuration Register 0

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0084

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

Bit	Symbol	Function	R/W
15:0	MAU[15:0]	MAC Address Upper Part These bits are used to set the 16 higher-order bits of the MAC address. For example, if the MAC address is 01-23-45-67-89-AB (hexadecimal), set H'0123 in this register.	R/W
31:16	—	These bits are read as 0.	R

Specifies the 16 higher-order bits of the 48-bit unicast MAC address. This register is also used as source address filed when the IP transmits a Pause frame or a PFC frame.

33.4.1.11 MRMAC1 : Reception MAC Address Configuration Register 1

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0088

Bit position:	31–0
Bit field:	MAL[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	MAL[31:0]	MAC Address Lower Part These bits are used to set the 32 lower-order bits of the MAC address. For example, if the MAC address is 01-23-45-67-89-AB (hexadecimal), set H'456789AB in this register.	R/W

The register is to use to specify the 32 lower-order bits of the 48-bit unicast MAC address. This register is also used as source address filed when the IP transmits a Pause frame or a PFC frame.

33.4.1.12 MRAFC : Reception Address Filter Configuration Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x008C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	MSAREP	NSAREP	SDSFREP	NDAREP	BCACP	MCACP	BSTENP	MSTENP	BCENP	MCENP	UCENP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	MSAREE	NSAREE	SDSFREE	NDAREE	BCACE	MCACE	BSTENE	MSTENE	BCENE	MCENE	UCENE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	Symbol	Function	R/W
0	UCENE	Unicast Reception Enable E-Frames This bit enables the unicast filtering based on MAC address (MRMAC0/1). 0: Unicast reception disabled 1: Unicast reception enabled	R/W
1	MCENE	Multicast Reception Enable E-Frames This bit enables the multicast filtering. 0: Multicast reception disabled 1: Multicast reception enabled	R/W
2	BCENE	Broadcast Reception Enable E-Frames This bit enables the broadcast filtering. 0: Broadcast reception disabled 1: Broadcast reception enabled	R/W
3	MSTENE*1	Multicast Storm Filter Reception Enable E-Frames This bit enables the multicast storm filtering. When it is asserted, Multicast frames are rejected after the number of received multicast e-frames reaches the configured threshold (MRSCE.CMFE). 0: Multicast storm filter reception disabled 1: Multicast storm filter reception enabled	R/W

Bit	Symbol	Function	R/W
4	BSTENE*2	Broadcast Storm Filter Reception Enable E-Frames This bit enables the broadcast storm filtering. When it is asserted, Broadcast frames are rejected after the number of received broadcast e-frames reaches to the configured threshold (MRSCE.CBFE). 0: Broadcast storm filter reception disabled 1: Broadcast storm filter reception enabled	R/W
5	MCACE*3	Multicast Storm Autoclear E-Frames This bit enables auto-clear of multicast frame storm counter. When it is asserted, multicast frame storm counter is cleared by detecting a frame with non-multicast destination address. After the counter is cleared, the IP can receive multicast frame again. 0: Multicast Storm Autoclear disabled 1: Multicast Storm Autoclear enabled	R/W
6	BCACE	Broadcast Storm Autoclear E-Frames This bit enables auto-clear of broadcast frame storm counter. When it is asserted, broadcast frames storm counter is cleared by detecting a frame with non-broadcast destination address. After the counter is cleared, the IP can receive Broadcast frame again. 0: Broadcast Storm Autoclear disabled 1: Broadcast Storm Autoclear enabled	R/W
7	NDAREE	Null Destination Address Reception Enable E-Frames This bit enables the reception of e-frames with destination MAC address null. 0: disable 1: enable	R/W
8	SDSFREE	Same DA and SA Frames Reception Enable E-Frames This bit enables the reception of e-frame with equal destination address and source address. 0: disable 1: enable	R/W
9	NSAREE	Null Source Address Reception Enable E-Frames This bit enables the reception of e-frames with destination MAC address null. 0: disable 1: enable	R/W
10	MSAREE	Multicast Source Address Reception Enable E-Frames This bit enables the reception of e-frame with multicast and broadcast source address 0: disable 1: enable	R/W
15:11	—	These bits are read as 0.	R
16	UCENP	Unicast Reception Enable P-Frames This bit enables the unicast filtering based on MAC address (MRMAC0/1). 0: Unicast reception disabled 1: Unicast reception enabled	R/W
17	MCENP	Multicast Reception Enable P-Frames This bit enables the multicast filtering. 0: Multicast reception disabled 1: Multicast reception enabled	R/W
18	BCENP	Broadcast Reception Enable P-Frames This bit enables the broadcast filtering. 0: Broadcast reception disabled 1: Broadcast reception enabled	R/W
19	MSTENP*1	Multicast Storm Filter Reception Enable P-Frames This bit enables the multicast storm filtering. When it is asserted, Multicast frames are rejected after the number of received multicast p-frames reaches the configured threshold (MRSCP.CMFP). 0: Multicast storm filter reception disabled 1: Multicast storm filter reception enabled	R/W
20	BSTENP*2	Broadcast Storm Filter Reception Enable P-Frames This bit enables the broadcast storm filtering. When it is asserted, Broadcast frames are rejected after the number of received broadcast p-frames reaches the configured threshold (MRSCP.CBFP). 0: Broadcast storm filter reception disabled 1: Broadcast storm filter reception enabled	R/W

Bit	Symbol	Function	R/W
21	MCACP*3	Multicast Storm Autoclear P-Frames This bit enables auto-clear of multicast frame storm counter. When it is asserted, multicast frame storm counter is cleared by detecting a frame with non-multicast destination address. After the counter is cleared, the bit can receive multicast frame again. 0: Multicast Storm Autoclear disabled 1: Multicast Storm Autoclear enabled	R/W
22	BCACP	Broadcast Storm Autoclear P-Frames This bit enables auto-clear of broadcast frame storm counter. When it is asserted, broadcast frames storm counter is cleared by detecting a frame with non-broadcast destination address. After the counter is cleared, the IP can receive Broadcast frame again 0: Broadcast Storm Autoclear disabled 1: Broadcast Storm Autoclear enabled	R/W
23	NDAREP	Null Destination Address Reception Enable P-Frames This bit enables the reception of p-frames with destination MAC address null. 0: disable 1: enable	R/W
24	SDSFREP	Same DA and SA Frames Reception Enable P-Frames This bit enables the reception of p-frame with equal destination address and source address. 0: disable 1: enable	R/W
25	NSAREP	Null Source Address Reception Enable P-Frames This bit enables the reception of p-frames with a destination MAC address null. 0: disable 1: enable	R/W
26	MSAREP	Multicast Source Address Reception Enable P-Frames This bit enables the reception of p-frame with multicast and broadcast source address 0: disable 1: enable	R/W
31:27	—	These bits are read as 0.	R

Note 1. SW: It is valid only when Multicast reception is enabled (MRAFC.MCENE=1).
Note 2. SW: It is valid only when Broadcast reception is enabled (MRAFC.BCENE=1)
Note 3. HW: Broadcast frames are not taken in account by multicast storm counter.

33.4.1.13 MRSCE : Reception Storm Configuration for E-Frames Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0090

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CBFE[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
15:0	CMFE[15:0]	Consecutive Multicast Frame Reception Count Setting for E-Frames These bits specify the upper-limitation of received consecutive multicast frames. After the number of received consecutive multicast frames reaches the set value, the subsequent multicast frames are discarded. 0x0000: 1 Multicast frame accepted until multicast frame counter reset 0x0001: 2 Multicast frame accepted until multicast frame counter reset ⋮ 0xFFFE: 65535 Multicast frames accepted until frame counter reset 0xFFFF: Reserved	R/W
31:16	CBFE[15:0]	Consecutive Broadcast Frame Reception Count Setting for E-Frames These bits specify the upper-limitation of received consecutive broadcast frames. After the number of received consecutive broadcast frames reaches the set value, the subsequent broadcast frames are discarded. 0x0000: 1 Broadcast frame accepted until broadcast frame counter reset 0x0001: 2 Broadcast frame accepted until broadcast frame counter reset ⋮ 0xFFFE: 65535 Broadcast frames accepted until broadcast frame counter reset 0xFFFF: Reserved	R/W

The register is to use to configure the number of times how many broadcast/multicast frames can be received consecutively.

33.4.1.14 MRSCP : Reception Storm Configuration for P-Frames Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0094

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

Bit	Symbol	Function	R/W
15:0	CMFP[15:0]	Consecutive Multicast Frame Reception Count Setting for P-Frames These bits specify the upper-limitation of received consecutive multicast frames. After the number of received consecutive multicast frames reaches the set value, the subsequent multicast frames are discarded. 0x0000: 1 Multicast frame accepted until multicast frame counter reset 0x0001: 2 Multicast frame accepted until multicast frame counter reset ⋮ 0xFFFE: 65535 Multicast frames accepted until multicast frame counter reset 0xFFFF: Reserved	R/W
31:16	CBFP[15:0]	Consecutive Broadcast Frame Reception Count Setting for P-Frames These bits specify the upper-limitation of received consecutive Broadcast frames. After the number of received consecutive broadcast frames reaches the set value, the subsequent broadcast frames are discarded. 0x0000: 1 Broadcast frame accepted until broadcast frame counter reset 0x0001: 2 Broadcast frame accepted until broadcast frame counter reset ⋮ 0xFFFE: 65535 Broadcast frames accepted between two broadcast frame counter resets 0xFFFF: Reserved	R/W

Configures the number of times how many broadcast/multicast frames can be received consecutively.

33.4.1.15 MRSCC : Reception Storm Counter Configuration Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0098

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

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

Bit	Symbol	Function	R/W
0	MSCCE	Multicast Storm Counter Clear E-Frames This bit requests to clear the multicast frames storm counter. After the counter is cleared, the IP can receive multicast frames again until the upper-limitation of received consecutive multicast frames (MRSCE.CMFE). It should be used with multicast Storm Autoclear disabled (MRAFC.MCACE = 0). Writing 1 to this bit clear the multicast storm counter	R
1	BSCCE	Broadcast Storm Counter Clear E-Frames This bit requests to clear the broadcast frames storm counter. After the counter is cleared, the IP can receive Broadcast frames again until the upper-limitation of received consecutive Broadcast frames (MRSCE.CBFE). It should be used with Broadcast Storm Autoclear disabled (MRAFC.BCACE = 0). Writing 1 to this bit clear the broadcast storm counter	R
15:2	—	These bits are read as 0.	R
16	MSCCP	Multicast Storm Counter Clear P-Frames This bit requests to clear the multicast frames storm counter. After the counter is cleared, the IPt can receive multicast frames again until the upper-limitation of received consecutive multicast frames (MRSCP.CMFP). It should be used with multicast Storm Autoclear disabled (MRAFC.MCACP = 0). Writing 1 to this bit clears the multicast storm counter	R
17	BSCCP	Broadcast Storm Counter Clear P-Frames This bit requests to clear the broadcast frames storm counter. After the counter is cleared, the IP can receive broadcast frames again until the upper-limitation of received consecutive broadcast frames (MRSCP.CBFP). It should be used with Broadcast Storm Autoclear disabled (MRAFC.BCACP = 0). Writing 1 to this bit clears the broadcast storm counter	R
31:18	—	These bits are read as 0.	R

Register to clear Broadcast and Multicast Storm Counters.

33.4.1.16 MRFSCE : Reception Frame Size Configuration for E-Frames Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x009C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	EMNS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
15:0	EMXS[15:0]	E-Frame Maximum Size A part of received frame which exceeds the set value are truncated. Don’t set a value under 16 bytes. MRFSCE.EMXS have to set 3FFFH or less.	R/W
31:16	EMNS[15:0]	E-Frame Minimum Size	R/W

Note: The size limitations concern the frame size RMAC outputs to an upper layer (e.g. MHD), and so the FCS removal impacts the register values.

33.4.1.17 MRFSCP : Reception Frame Size Configuration for P-Frames Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x00A0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PMNS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
15:0	PMXS[15:0]	P-Frame Maximum Size A part of received frame which exceeds the set value are truncated. Don’t set a value under 16 bytes.	R/W
31:16	PMNS[15:0]	P-Frame Minimum Size	R/W

Note: The size limitations concern the frame size RMAC outputs to an upper layer (e.g. MHD), and so the FCS removal impacts the register values.

33.4.1.18 MTRC : Timestamp Reception Configuration Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x00A4

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TRHFME1 to TRHFME0	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	TRHFME1 to TRHFME0	Timestamp Reception Hardware Filter Match Enable n (n = 0 to 1) 0: disable 1: enable	R/W
23:2	—	These bits are read as 0.	R
24	TRDDE	Timestamp Reception Default Disable E-Frame A timestamp for every frame will be transmitted to the MHD. This timestamp will be taken on the default timer except when TRHFME is set and the hardware filter matches. 0: enable 1: disable	R/W
25	TRDDP	Timestamp Reception Default Disable P-Frame A timestamp for every frame will be transmitted to the MHD. This timestamp will be taken on the default timer. 0: enable 1: disable	R/W
26	TCTSE	Timestamp Capture on TX Side E-Frame The bit enables the timestamp capture on TX side for e-frames 0: disable 1: enable	R/W
27	TCTSP	Timestamp Capture on TX Side P-Frame The bit enables the timestamp capture on TX side for p-frames 0: disable 1: enable	R/W
29:28	DTN[1:0]	Default Timer Number The bit set Timer used for timestamp forwarding on rx mhd i/f when no timer is detected.	R/W
31:30	—	These bits are read as 0.	R

TRHFMEn bits (Timestamp Reception Hardware Filter Match Enable n (n = 0 to 1))

Setting bit i to 1 in this register will enable hardware filtering for timer i.

When bit i is set to 1, at least one of the bit i in MPFCn registers should be set to one.

When hardware filter matches, the timestamp from the matched timer will be sent along with the frame.

33.4.1.19 MRPFM : Reception Pause or PFC Frame Monitoring Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x00AC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	PFCTCA7 to PFCTCA0							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	PTCA	Pause Time Counting Active This bit indicates that the Pause time of a received Pause frame has not been completed. 0: No pause state 1: Pause state	R
15:1	—	These bits are read as 0.	R
23:16	PFCTCA7 to PFCTCA0	PFC Time Counting Active n (n = 0 to 7) This bit indicates that the Pause time of a received PFC frame has not been completed. 0: No pause state 1: Pause state	R
31:24	—	These bits are read as 0.	R

33.4.1.20 MPFCt : PTP Filtering Register Configuration Register t (t = 0 to 15)

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0100 + 0x4 × t

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TEF1 to TEF0	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
7:0	PFBN[7:0]	PTP Filtering Byte Number 0x00: Byte number 0 will be filtered. ⋮ 0xFF: Byte number 255 will be filtered.	R/W
15:8	PFBV[7:0]	PTP Filtering Byte Value When the filtered byte has a value equal to this register, the byte will be eligible for the corresponding timers.	R/W
17:16	TEF1 to TEF0	Timer Enable for Filtering n (n = 0 to 1) Setting bit i to 1 in this register will apply the filter to timer i.	R/W
31:18	—	These bits are read as 0.	R

Note: This register is used for RX time domain selection.
When an overlapping configuration is done, HW will always choose the timer with the smaller number.

The register is used to configure the PTP frame filtering for time domain selection.

33.4.1.21 MLVC : Link Verification Configuration Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0180

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

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

Bit	Symbol	Function	R/W
6:0	LVT[6:0]	Link Verification Timer These bits define the nominal wait time between verification attempts in unit of milliseconds. It means the time of retransmitting SMD-V frame. 000 0000: 1ms wait time 000 0001: 2ms wait time ⋮ 000 1001: 10ms wait time (default specified by IEEE 802.3) ⋮ 111 1111: 128ms wait time	R/W
7	—	This bit is read as 0.	R
8	PASE	Preemption Auto Response Enable This bit enables the automatic response to verify frames.	R/W
15:9	—	These bits are read as 0.	R
16	PLV	Preemption Link Verification 0: Preemption Link Verification not requested 1: Preemption Link Verification requested	R/W*1
31:17	—	These bits are read as 0.	R

Note: When PLV is set to 1, do not make write access continuously. Please wait completion of Link Verification Procedure for re-write access.

Note 1. Read value differs from written value.

PLV bit (Preemption Link Verification)

By asserting this bit, link verification is requested.

This bit requests the Link Verification process. When link verification is completed, this bit is cleared.

When Link Verification succeeded, Preemption status bit (MMIS0.LVSS) are set. While it failed, the status bit (MMIS0.LVFS) is set.

Going out of OPERATION mode will clear this register.

As described in 7.2.2, this bit should only be used in initialization.

SW set this bit when another transmission is not requested.

33.4.1.22 MEEEC : Energy Efficient Ethernet Configuration Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0184

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

Bit	Symbol	Function	R/W
0	LPITR*1	LPI Transmit Request This bit requests to enter Low Power Idle (LPI) mode. RMAC enters to LPI state after all ongoing transmissions have been completed. When this bit is negated, this IP leaves LPI state. LPI signal should be transmitted only when TX is disabled. Going out of OPERATION mode will clear this register. 0: LPI mode is not requested 1: LPI mode is requested	R/W
31:1	—	These bits are read as 0.	R

Note 1. SW: LPITR bit can only be set when transmission request to RMAC is stopped, and transmission request can only be allowed after LPITR is cleared.
It is prohibited to request transmission from HMD when LPITR bit is set.

33.4.1.23 MLBC : Loopback Configuration Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0188

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

Bit	Symbol	Function	R/W
0	LBME	Loopback Mode Enable Transmit xMII signals are connected to reception xMII signals internally. 0: disable 1: enable	R/W
31:1	—	These bits are read as 0.	R

Note: When loopback mode is enabled, frame received by external PHY Interface are ignored.
If this bit is set, clk_phy_tx and clk_phy_rx must be synchronized with each other.

33.4.2 RMAC Counter Registers

Counter registers

Note: All counter registers in RMAC are read clear registers.

When counters reach their own maximum value, they stop.

33.4.2.1 MMPFTCT : Manual Pause Frame Transmit Counter Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0300

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

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

Bit	Symbol	Function	R/W
15:0	MPFTC[15:0]	Manual Pause frame Transmit Counter	R
31:16	—	These bits are read as 0.	R

MPFTC[15:0] bits (Manual Pause frame Transmit Counter)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

HW: Incremented by 1 when manual Pause frame is transmitted.

33.4.2.2 MAPFTCT : Automatic Pause Frame Transmit Counter Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0304

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

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

Bit	Symbol	Function	R/W
15:0	APFTC[15:0]	Automatic pause Frame Counter	R
31:16	—	These bits are read as 0.	R

APFTC[15:0] bits (Automatic pause Frame Counter)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

HW: Incremented by 1 when automatic Pause frame is transmitted.

33.4.2.3 MPFRCT : Pause Frame Receive Counter Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0308

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

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

Bit	Symbol	Function	R/W
15:0	PFRC[15:0]*1	Pause Frame Receive Counter	R
31:16	—	These bits are read as 0.	R

Note 1. SW: Set condition is only available when Flow Control is enabled in reception. (MRGC.PFRC = 1)

PFRC[15:0] bits (Pause Frame Receive Counter)

[Clearing condition]

HW: Being in RESET mode will clear this register

SW: Reading this register clears it.

[Increment condition]

HW: Incremented by 1 when automatic Pause frame is received.

33.4.2.4 MFCICT : False Carrier Indication Counter Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x030C

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

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

Bit	Symbol	Function	R/W
15:0	FCIC[15:0]	False Carrier Indication Counter	R
31:16	—	These bits are read as 0.	R

Note: This register is counter of “False Carrier Indication” event. This event goes through the synchronizer between clk and clk_phy_rx. If the event occurs continuously, there is a possibility of miss counting.

FCIC[15:0] bits (False Carrier Indication Counter)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

HW: Incremented by 1 when False Carrier Indication pattern is detected.

33.4.2.5 MEEECT : Energy Efficient Ethernet Counter Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0310

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

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

Bit	Symbol	Function	R/W
15:0	EEERC[15:0]	Energy Efficient Ethernet Receive Counter	R
31:16	—	These bits are read as 0.	R

EEERC[15:0] bits (Energy Efficient Ethernet Receive Counter)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

HW: Incremented by 1 when LPI signal is detected on xMII reception bus.

33.4.2.6 MMPCFTCTt : Manual PFC Frame Transmit Counter Register (t = 0, 1)

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0320 + 0x4 × t

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

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

Bit	Symbol	Function	R/W
15:0	MPCFCTC[15:0]	Manual PFC frame Transmit Counter	R
31:16	—	These bits are read as 0.	R

MPCFCTC[15:0] bits (Manual PFC frame Transmit Counter)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

HW: Incremented by 1 when manual PFC frames is transmitted.

33.4.2.7 MAPCFTCTt : Automatic PFC Frame Transmit Counter Register (t = 0, 1)

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0330 + 0x4 × t

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

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

Bit	Symbol	Function	R/W
15:0	APCFCTC[15:0]	Automatic PFC Frame Counter	R
31:16	—	These bits are read as 0.	R

APCFCTC[15:0] bits (Automatic PFC Frame Counter)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

HW: Incremented by 1 when manual PFC frames is transmitted.

33.4.2.8 MPCFRCTt : PFC Frame Receive Counter Register (t = 0, 1)

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0340 + 0x4 × t

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

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

Bit	Symbol	Function	R/W
15:0	PCFCRC[15:0]*1	PFC Frame Receive Counter	R
31:16	—	These bits are read as 0.	R

Note 1. SW: Set condition is only available when Flow Control is enabled in reception (MRGC.PFRC = 1).

PCFCRC[15:0] bits (PFC Frame Receive Counter)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

HW: Incremented by 1 when manual PFC frames is received.

33.4.2.9 MROVFC : Receive Overflow Counter Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0360

Bit position:	31 ... 0
Bit field:	ROVFC[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	ROVFC[31:0]	Receive overflow counter	R

ROVFC[31:0] bits (Receive overflow counter)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

HW: Incremented by 1 when frame receive overflow occurred (when set condition for MEIS.REOES or MEIS.RPOES are met).

33.4.2.10 MRXBCEU : Received Byte Counter E-Frames Upper Side Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x044C

Bit position:	31 ... 0
Bit field:	RBNEU[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	RBNEU[31:0]	Received byte number E-frames upper side When the RMAC does not pass the FCS to the upper layer, FCS is not taken in account in total byte calculation. Read before reading MRXBCEL.	R

RBNEU[31:0] bits (Received byte number E-frames upper side)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Incremented each time the received data of e-Frame is sent to the rx mhd interface regardless error is detected.

33.4.2.11 MRXBCEL : Received Byte Counter E-Frames Lower Side Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0450

Bit position:	31	...	0
Bit field:	RBNEL[31:0]		
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		

Bit	Symbol	Function	R/W
31:0	RBNEL[31:0]	Received byte number E-frames lower side When the RMAC does not pass the FCS to the upper layer, FCS is not taken in account in total byte calculation. Read after reading MRXBCEU.	R

RBNEL[31:0] bits (Received byte number E-frames lower side)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: None. (Reading MRXBCEL register holds it.) Increment condition

[Increment condition]

Incremented each time the received data of e-frame is sent to the rx mhd interface regardless the error detection.

33.4.2.12 MRXBCPU : Received Byte Counter P-Frames Upper Side Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0454

Bit position:	31	...	0
Bit field:	RBNPU[31:0]		
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		

Bit	Symbol	Function	R/W
31:0	RBNPU[31:0]	Received byte number P-frames upper side If the RMAC does not pass the FCS to the upper layer, FCS is not taken in account in total byte calculation. Read before reading MRXBCPL.	R

RBNPU[31:0] bits (Received byte number P-frames upper side)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Incremented each time the received data of p-frame is sent to the rx mhd interface regardless the error detection.

33.4.2.13 MRXBCPL : Received Byte Counter P-Frames Lower Side Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0458

Bit position:	31 … 0
Bit field:	RBNPL[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	RBNPL[31:0]	Received byte number P-frames lower side When the RMAC does not pass the FCS to the upper layer, FCS is not taken in account in total byte calculation. Read after reading MRXBCPU.	R

RBNPL[31:0] bits (Received byte number P-frames lower side)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: None. (Reading MRXBCPL register holds it.)

[Increment condition]

Incremented each time the received data of p-frame is sent to the rx mhd interface regardless the error detection.

33.4.2.14 MRGFCE : Received Good Frame Counter E-Frames Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0408

Bit position:	31 … 0
Bit field:	RGFNE[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	RGFNE[31:0]	Received good frame number E-frames	R

RGFNE[31:0] bits (Received good frame number E-frames)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Incremented when the reception of e-frame is completed without error.

33.4.2.15 MRGFCP : Received Good Frame Counter P-Frames Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x040C

Bit position:	31–0
Bit field:	GFNP[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	GFNP[31:0]	Received good frame number P-frames	R

GFNP[31:0] bits (Received good frame number P-frames)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Incremented when the reception of p-frame is completed without error.

33.4.2.16 MRBFC : Received Good Broadcast Frame Counter Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0410

Bit position:	31–0
Bit field:	RBFN[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	RBFN[31:0]	Received good broadcast frame number	R

RBFN[31:0] bits (Received good broadcast frame number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Incremented when destination address filed of broadcast frame is sent to rx mhd interface without error.

33.4.2.17 MRMFC : Received Good Multicast Frame Counter Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0414

Bit position:	31–0
Bit field:	RMFN[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	RMFN[31:0]	Received good multicast frame number	R

RMFN[31:0] bits (Received good multicast frame number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Incremented when destination address filed of multicast frame is sent to rx mhd interface without error.

33.4.2.18 MRUFC : Received Good Unicast Frame Counter Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0418

Bit position:	31–0
Bit field:	RUFN[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	RUFN[31:0]	Received good unicast frame number	R

RUFN[31:0] bits (Received good unicast frame number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Incremented when destination address filed of unicast frame is sent to rx mhd interface without error.

33.4.2.19 MRPEFC : Received PHY Error Frame Count Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x041C

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

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

Bit	Symbol	Function	R/W
15:0	RPEFN[15:0]	Received PHY error frame number	R
31:16	—	These bits are read as 0.	R

RPEFN[15:0] bits (Received PHY error frame number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Incremented when received data is sent to rx mhd interface with phy error.

33.4.2.20 MRNEFC : Received Nibble Error Frame Count Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0420

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

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

Bit	Symbol	Function	R/W
15:0	RNEFN[15:0]	Received nibble error frame number	R
31:16	—	These bits are read as 0.	R

RNEFN[15:0] bits (Received nibble error frame number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Incremented when received data is sent to rx mhd interface with nibble error.

33.4.2.21 MRFMEFC : Received FCS/mCRC Error Frame Count Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0424

Bit position: 31 … 0

Bit field: RFMEFN[31:0]

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	RFMEFN[31:0]	Received FCS/mCRC error frame number	R

RFMEFN[31:0] bits (Received FCS/mCRC error frame number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Incremented when received data is sent to rx mhd interface with FCS/mCRC error.

33.4.2.22 MRFFMEFC : Received Final Fragment Missing Error Frame Count Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0428

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

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

Bit	Symbol	Function	R/W
15:0	RFFMEFN[15:0]	Received final fragment missing error frame number	R
31:16	—	These bits are read as 0.	R

RFFMEFN[15:0] bits (Received final fragment missing error frame number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Incremented when received data is sent to rx mhd interface with final fragment error.

33.4.2.23 MRCFCEFC : Received C-Fragment Count Error Frame Count Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x042C

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

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

Bit	Symbol	Function	R/W
15:0	RCFCEFN[15:0]	Received C-fragment count error frame number	R
31:16	—	These bits are read as 0.	R

RCFCEFN[15:0] bits (Received C-fragment count error frame number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Incremented when received data is sent to rx mhd interface with C-fragment count error.

33.4.2.24 MRFCEFC : Received Fragment Count Error Frame Count Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0430

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

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

Bit	Symbol	Function	R/W
15:0	RFCEFN[15:0]	Received fragment count error frame number	R
31:16	—	These bits are read as 0.	R

RFCEFN[15:0] bits (Received fragment count error frame number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Incremented when received data is sent to rx mhd interface with fragment count error.

33.4.2.25 MRRCFEFC : Received RMAC Filter Error Frame Count Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0434

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

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

Bit	Symbol	Function	R/W
15:0	RRCFEFN[15:0]	Received RMAC filter error frame number	R
31:16	—	These bits are read as 0.	R

RRCFEFN[15:0] bits (Received RMAC filter error frame number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Incremented when received data is sent to rx mhd interface with RMAC filtered.

33.4.2.26 MRFC : Received Frame Count Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0438

Bit position:	31–0
Bit field:	RFN[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	RFN[31:0]	Received frame number	R

RFN[31:0] bits (Received frame number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Incremented when received frame is sent to rx mhd interface.

33.4.2.27 MRGUEFC : Received Good Undersize Error Frame Count Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x043C

Bit position:	31–0
Bit field:	RUEFN[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	RUEFN[31:0]	Received good undersize error frame number	R

RUEFN[31:0] bits (Received good undersize error frame number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Incremented when received data is sent to rx mhd interface with undersize error but without fcs/mcrc error.

33.4.2.28 MRBUEFC : Received Bad Undersize Error Frame Count Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0440

Bit position:	31–0
Bit field:	RUEFN[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	RUEFN[31:0]	Received bad undersize error frame number	R

RUEFN[31:0] bits (Received bad undersize error frame number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Incremented when received data is sent to rx mhd interface with undersize error and fcs/mcrc error.

33.4.2.29 MRGOEFC : Received Good Oversize Error Frame Count Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0444

Bit position:	31 — 0
Bit field:	RGOEFN[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	RGOEFN[31:0]	Received good oversize error frame number	R

RGOEFN[31:0] bits (Received good oversize error frame number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Incremented when received data is sent to rx mhd interface with oversize error, but without fcs/mcrc error.

33.4.2.30 MRBOEFC : Received Bad Oversize Error Frame Count Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0448

Bit position:	31 — 0
Bit field:	RBOEFN[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	RBOEFN[31:0]	Received bad oversize error frame number	R

RBOEFN[31:0] bits (Received bad oversize error frame number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Incremented when received data is sent to rx mhd interface with oversize error and fcs/mcrc error.

33.4.2.31 MTXBCEU : Transmitted Byte Counter E-Frames Upper Side Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0520

Bit position:	31 ... 0
Bit field:	TBNEU[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TBNEU[31:0]	Transmitted byte number E-frames upper side	R

Note: If the FCS is not passed to the upper layer FCS is not taken in account in total byte calculation.

TBNEU[31:0] bits (Transmitted byte number E-frames upper side)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Increment every time transmit data for e-frame is received on tx mhd i/f.

Read before reading MTXBCEL.

33.4.2.32 MTXBCEL : Transmitted Byte Counter E-Frames Lower Side Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0524

Bit position:	31 ... 0
Bit field:	TBNEL[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TBNEL[31:0]	Transmitted byte number E-frames lower side	R

Note: If the FCS is not passed to the upper layer FCS is not taken in account in total byte calculation.

TBNEL[31:0] bits (Transmitted byte number E-frames lower side)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: None. (Reading MTXBCEL register holds it.)

[Increment condition]

Increment every time transmit data for e-frame is received on tx mhd i/f.

Read before reading MTXBCEU.

33.4.2.33 MTXBCPU : Transmitted Byte Counter P-Frames Upper Side Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0528

Bit position:	31 ... 0
Bit field:	TBNPU[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TBNPU[31:0]	Transmitted byte number P-frames upper side	R

Note: If the FCS is not passed to the upper layer, FCS is not taken in account in total byte calculation.

TBNPU[31:0] bits (Transmitted byte number P-frames upper side)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Increment every time transmit data for p-frame is received on tx mhd i/f.

Read before reading MTXBCPL.

33.4.2.34 MTXBCPL : Transmitted Byte Counter P-Frames Lower Side Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x052C

Bit position:	31 ... 0
Bit field:	TBNPL[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TBNPL[31:0]	Transmitted byte number P-frames lower side	R

Note: If the FCS is not passed to the upper layer, FCS is not taken in account in total byte calculation.

TBNPL[31:0] bits (Transmitted byte number P-frames lower side)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: None. (Reading MTXBCPL register holds it.)

[Increment condition]

Increment every time transmit data for p-frame is received on tx mhd i/f.

Read before reading MTXBCPU.

33.4.2.35 MTGFCE : Transmitted Good Frame Counter E-Frames Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0508

Bit position:	31 … 0
Bit field:	TGFNE[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TGFNE[31:0]	Transmitted good frame number E-frames	R

TGFNE[31:0] bits (Transmitted good frame number E-frames)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Increment when transmission of e-frame is completed on tx mhd i/f without error.

33.4.2.36 MTGFCP : Transmitted Good Frame Counter P-Frames Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x050C

Bit position:	31 … 0
Bit field:	TGFNP[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TGFNP[31:0]	Transmitted good frame number P-frames	R

TGFNP[31:0] bits (Transmitted good frame number P-frames)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Increment when transmission of p-frame is completed on tx mhd i/f without error.

33.4.2.37 MTBFC : Transmitted Broadcast Frame Counter Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0510

Bit position:	31 … 0
Bit field:	TBFN[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TBFN[31:0]	Transmitted broadcast frame number	R

TBFN[31:0] bits (Transmitted broadcast frame number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Increment when destination address field of broadcast frame is sent on tx mhd i/f.

33.4.2.38 MTMFC : Transmitted Multicast Frame Counter Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0514

Bit position:	31–0
Bit field:	TMFN[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TMFN[31:0]	Transmitted multicast frame number	R

TMFN[31:0] bits(Transmitted multicast frame number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Increment when destination address field of multicast frame is sent on tx mhd i/f.

33.4.2.39 MTUFC : Transmitted Unicast Frame Counter Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0518

Bit position:	31–0
Bit field:	TUFN[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TUFN[31:0]	Transmitted unicast frame number	R

TUFN[31:0] bits (Transmitted unicast frame number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Increment when destination address field of unicast frame is sent on tx mhd i/f.

33.4.2.40 MTEFC : Transmitted Error Frame Counter Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x051C

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

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

Bit	Symbol	Function	R/W
15:0	TEFN[15:0]	Transmitted error frame number	R
31:16	—	These bits are read as 0.	R

TEFN[15:0] bits (Transmitted error frame number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment condition]

Increment when frame transmission is completed on tx mhd interface with error.

33.4.3 RMAC Interrupt Registers

33.4.3.1 MEIS : Error Interrupt Status Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0200

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	FOES	FUES	FFS	RPOOMS	FRCES	CFCES	FFMES	FCMCES	PNAES	PDES	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	CTLES1 to CTLES0		—	RPOES	REOES	FCES	BFES	TBCIS	TCES	FCDS	PFRROS	PRES	—	TSLS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TSLS	Transmission Stream Lost Status Flag When using Cut through, this flag set condition may be due to the state ([COMA] CAEIS0.BPOPS) where all Buffer pointers are used. Please use the Watermark function so that Buffer Pool Out of Pointer Status does not occur. If it gets this error with Buffer Pool Out of Pointer Status using Cut through (it may get hung-up in the transmit state), initialize the state with Emergency reset flow [TOP].	R/W*1
1	—	This bit is read as 0.	R
2	PRES	Pause or PFC Frame Reception Error Status Flag	R/W*1
3	PFRROS	Pause or PFC Frame Retransmit Retry Over Status Flag	R/W*1

Bit	Symbol	Function	R/W
4	FCDS	False Carrier Detection Status Flag	R/W*1
5	TCES	TX CRC Error Status Flag	R/W*1
6	TBCIS	TX Bad CRC Insertion Status Flag	R/W*1
7	BFES	Bad Fragment Error Status Flag When C-fragment is received at very first time before any valid S-fragment is received or C-fragment is received after frame termination is occurred before any valid S-fragment is received, then RMAC will discard the fragment and this error status flag will not be set. For Error status flag to get set, RMAC should have received one valid S-fragment before error condition (bad fragment error) has occurred.	R/W*1
8	FCES	Frame Count Error Status Flag When C-fragment is received at very first time before any valid S-fragment is received or C-fragment is received after frame termination is occurred before any valid S-fragment is received, then RMAC will discard the fragment and this error status flag will not be set. For Error status flag to get set, RMAC should have received one valid S-fragment before error condition (frame count error) has occurred.	R/W*1
9	REOES*2	E-Frame Overflow Error Status Flag When frame reception is already ongoing and buffer overflow occurs i.e. CABPPCM.RPC = 0, RMAC adds one byte padding to the reception error Frames to terminate the protocol internally. When new frame is received, during buffer overflow i.e. CABPPCM.RPC = 0, the incoming frame will be rejected completely. 0: No overflow error 1: Overflow error occurred during e-frame reception.	R/W*1
10	RPOES*2	P-Frame Overflow Error Status Flag When frame reception is already ongoing and buffer overflow occurs i.e. CABPPCM.RPC = 0, RMAC adds one byte padding to the reception error Frames to terminate the protocol internally. When new frame is received, during buffer overflow i.e. CABPPCM.RPC = 0, the incoming frame will be rejected completely. 0: No overflow error 1: Overflow error occurred during p-frame reception.	R/W*1
11	—	This bit is read as 0.	R
13:12	CTLES1 to CTLES0	Captured Timestamp Lost Error Status Flag n (n = 0 to 1)	R/W*1
19:14	—	These bits are read as 0.	R
20	PDES	PHY Data Error Status Flag	R/W*1
21	PNAES	PHY Nibble Alignment Error Status Flag	R/W*1
22	FCMCES	FCS/mCRC Error Status Flag	R/W*1
23	FFMES	Final Fragment Missing Error Status Flag	R/W*1
24	CFCES	C Fragment Count Error Status Flag	R/W*1
25	FRCES	Fragment Count Error Status Flag	R/W*1
26	RPOOMS	Reception Partially out of Operation Mode Status Flag This error has nothing to do with partial rejection due to buffer overflow. E-Frames can only be fragmented/partialized by buffer overflows and it is not affected by partial out of operation.	R/W*1
27	FFS	Frame Filtered Status Flag	R/W*1
28	FUES	Undersize Error Status Flag	R/W*1
29	FOES	Oversize Error Status Flag	R/W*1
31:30	—	These bits are read as 0.	R

Note: This interrupt register is mapped to rmc_meis_int interrupt wire.
Note 1. Read value differs from written value.
Note 2. HW: Pulse is generated per 1 frame although multiple deassertion is generated.

TSLS bit (Transmission Stream Lost Status Flag)

[Setting condition]

HW: when MHD do not provide the transmitted data timely. In this case an invalid FCS is transmitted. (only for e-frames)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this register will clear it.

PRES bit (Pause or PFC Frame Reception Error Status Flag)

[Setting condition]

HW: When Pause or PFC frame is received with lower than minimum pause/pfc frame length.

Minimum PAUSE Length: 18Byte (MRGC.RCPT==0) or 22 Byte ((MRGC.RCPT==1)

Minimum PFC Length: 34Byte (MRGC.RCPT==0) or 38 Byte ((MRGC.RCPT==1)

HW: When PAUSE/PFC frame is received with fcs error in MRGC.RFCFE==0.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this register will clear it.

PFRROS bit (Pause or PFC Frame Retransmit Retry Over Status Flag)

[Setting condition]

HW: when the total number of Pause (pfc) transmit count reaches the configured upper limit and Pause time expires.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this register will clear it.

FCDS bit (False Carrier Detection Status Flag)

[Setting condition]

HW: When PHY-LSI has detected a false carrier on the line in case of MII and GMII.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this register will clear it.

TCES bit (TX CRC Error Status Flag)

[Setting condition]

HW: When CRC error occurred.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this register will clear it.

TBCIS bit (TX Bad CRC Insertion Status Flag)

[Setting condition]

HW: A bad CRC is inserted from the upper layer (e.g. MHD).

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this register will clear it.

BFES bit (Bad Fragment Error Status Flag)

[Setting condition]

HW: C fragment has been detected without receiving S fragment.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this register will clear it.

FCES bit (Frame Count Error Status Flag)

[Setting condition]

When S fragment has been received in the wrong order.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this register will clear it.

REOES bit (E-Frame Overflow Error Status Flag)

[Setting condition]

HW: The acknowledgment signal was not received successfully from an upper layer (e.g. MHD) during e-frame reception.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this register will clear it.

RPOES bit (P-Frame Overflow Error Status Flag)

[Setting condition]

HW: The acknowledgment signal was not received successfully from an upper layer (e.g. MHD) during p-frame reception.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this register will clear it.

CTLESn bits (Captured Timestamp Lost Error Status Flag n (n = 0 to 1))

[Setting condition]

HW: When captured timestamp is lost.

This bit is fixed to 0

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this register will clear it.

[Error recovery]

No recovery. Timestamp is lost.

PDES bit (PHY Data Error Status Flag)

[Setting condition]

HW: When frame is received with error.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this register will clear it.

PNAES bit (PHY Nibble Alignment Error Status Flag)

[Setting condition]

HW: When residual-bit is received.

[Clearing condition]

SW: Writing 1 to this register will clear it.

FCMCES bit (FCS/mCRC Error Status Flag)

[Setting condition]

HW: When unexpected FCS/mCRC are received.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this register will clear it.

FFMES bit (Final Fragment Missing Error Status Flag)

[Setting condition]

HW: When an S-fragment is received even though the previous fragment has been terminated by a mCRC.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this register will clear it.

CFCES bit (C Fragment Count Error Status Flag)

[Setting condition]

HW: When a C fragment not corresponding to the current frame is received For example: S0-C0-C0-C1.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this register will clear it.

FRCES bit (Fragment Count Error Status Flag)

[Setting condition]

HW: When a C-fragment is received with a bad fragment count field.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this register will clear it.

RPOOMS bit (Reception Partially out of Operation Mode Status Flag)

[Setting condition]

HW: When a frame is being received while RMAC is moving out of operation mode (suspending).

1. Initial or intermediate fragment of p-frame is receiving.
2. E-frame is receiving when initial p-frame is already received and last fragment of p-frame is not received.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this register will clear it.

FFS bit (Frame Filtered Status Flag)

[Setting condition]

HW: When frame did not pass the RMAC filters

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this register will clear it.

FUES bit (Undersize Error Status Flag)

[Setting condition]

HW: When too short frame is received. (When the received frame size is shorter than the setting in MRFSCE.EMNS and MRFSCP.PMNS)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this register will clear it.

FOES bit (Oversize Error Status Flag)

[Setting condition]

HW: When too big frame is received. (When the received frame size is bigger than the setting in MRFSCE.EMXS and MRFSCP.PMXS)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this register will clear it.

Table 33.5 Transmission error list

Error type	Event	Status Bit
TX Bad CRC Insertion Status	A packet has been received for transmission by RMAC with an error and a wrong FCS has been inserted.	MEIS.TBCIS
TX CRC Error Status	A packet which included a presupposed valid FCS had a faulty FCS.	MEIS.TCES
Pause frame Retransmit Retry Over Status	A Pause or PFC retransmit count reaches the configured upper limit(MTPFC.APFRLV) and Pause time expires.	MEIS.PFRROS
Transmission Stream Lost Status	An e-frame data has not been provided on time to RMAC.	MEIS.TSLS

Table 33.6 Reception error list

No	Error type	Event	Frame status bit	Status register
1	PHY data error	Received a data with error.	Bit 0	MEIS.PDES
2	PHY nibble alignment error	Received residual-bit.	Bit 1	MEIS.PNAES
3	FCS/mCRC error	Received unexpected FCS field value.	Bit 2	MEIS.FCMCES
4	Final fragment missing error	Occurs when an S-fragment is received even though the previous fragment has been terminated by a mCRC.	Bit 3	MEIS.FFMES
5	C fragment count error	Occurs when a C fragment not corresponding to the current frame is received. For example: S0-C0-C0-C1	Bit 4	MEIS.CFCES
6	Fragment count error	Occurs when a C-fragment is received with a bad fragment count field.	Bit 5	MEIS.FRCES
7	Reception partially out of operation mode	Occurs when a frame is being received while RMAC is moving out of operation mode.	Bit 6	MEIS.RPOOMS
8	Frame filtered	The frame did not pass the RMAC filters.	Bit 7	MEIS.FFS
9	Undersize error	The frame is too small.	Bit 8	MEIS.FUES
10	Oversize error	The frame is too big.	Bit 9	MEIS.FOES
11	Buffer overflow error	Occurs when the rx done signal has been deasserted during frame reception.	Bit 12	MEIS.REOES MEIS.RPOES
12	Frame Count Error Status	Occurs when a wrong S-frame order is received. For example: S0-S1-S2-S1	—	MEIS.FCES
13	Bad Fragment Error Status	Occurs when a C fragment has been detected without receiving S fragment.	—	MEIS.BFES
14	False Carrier Detection Status	Occurs when PHY-LSI has detected a false carrier on the line.	—	MEIS.FCDS

33.4.3.2 MEIE : Error Interrupt Enable Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0204

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	FOEE	FUEE	FFE	RPOOME	FRCEE	CFCEE	FFMEE	FCMCEE	PNAEE	PDEE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	CTLEE1 to CTLEE0		—	RPOEE	REOEE	FCEE	BFEE	TBCIE	TCEE	FCDE	PFRROE	PREE	—	TSLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TSLE	Transmission Stream Lost Enable	R/W*1
1	—	This bit is read as 0.	R
2	PREE	Pause or PFC Frame Reception Error Enable	R/W*1
3	PFRROE	Pause or PFC Frame Retransmit Retry Over Enable	R/W*1
4	FCDE	False Carrier Detection Enable	R/W*1
5	TCEE	Tx CRC Error Enable	R/W*1
6	TBCIE	TX Bad CRC Insertion Enable	R/W*1
7	BFEE	Bad Fragment Error Enable	R/W*1
8	FCEE	Frame Count Error Enable	R/W*1
9	REOEE	E-Frame Overflow Error Enable	R/W*1
10	RPOEE	P-Frame Overflow Error Enable	R/W*1
11	—	This bit is read as 0.	R
13:12	CTLEE1 to CTLEE0	Captured Timestamp Lost Error Enable n (n = 0 to 1)	R/W*1
19:14	—	These bits are read as 0.	R
20	PDEE	PHY Data Error Enable	R/W*1
21	PNAEE	PHY Nibble Alignment Error Enable	R/W*1
22	FCMCEE	FCS/mCRC Error Enable	R/W*1
23	FFMEE	Final Fragment Missing Error Enable	R/W*1
24	CFCEE	C Fragment Count Error Enable	R/W*1
25	FRCEE	Fragment Count Error Enable	R/W*1
26	RPOOME	Reception Partially out of Operation Mode Enable	R/W*1
27	FFE	Frame Filtered Enable	R/W*1
28	FUEE	Undersize Error Enable	R/W*1
29	FOEE	Oversize Error Enable	R/W*1
31:30	—	These bits are read as 0.	R

Note 1. Read value differs from written value.

TSLE bit (Transmission Stream Lost Enable)

[Setting condition]

Writing 1 to one of this bit will set it.

[Clearing condition]

Writing 1 to MEID.TSLD will clear this bit.

PREE bit (Pause or PFC Frame Reception Error Enable)

[Setting condition]

Writing 1 to one of this bit will set it.

[Clearing condition]

Writing 1 to MEID.PRED will clear this bit.

PFRROE bit (Pause or PFC Frame Retransmit Retry Over Enable)

[Setting condition]

Writing 1 to one of this bit will set it.

[Clearing condition]

Writing 1 to MEID.PFRROD will clear this bit.

FCDE bit (False Carrier Detection Enable)

[Setting condition]

Writing 1 to one of this bit will set it.

[Clearing condition]

Writing 1 to MEID.FCDD will clear this bit.

TCEE bit (Tx CRC Error Enable)

[Setting condition]

Writing 1 to one of this bit will set it.

[Clearing condition]

Writing 1 to MEID.TCED will clear this bit.

TBCIE bit (TX Bad CRC Insertion Enable)

[Setting condition]

Writing 1 to one of this bit will set it.

[Clearing condition]

Writing 1 to MEID.TBCID will clear this bit.

BFEE bit (Bad Fragment Error Enable)

[Setting condition]

Writing 1 to one of this bit will set it.

[Clearing condition]

Writing 1 to MEID.BFED will clear this bit.

FCEE bit (Frame Count Error Enable)

[Setting condition]

Writing 1 to one of this bit will set it.

[Clearing condition]

Writing 1 to MEID.FCED will clear this bit.

REOEE bit (E-Frame Overflow Error Enable)

[Setting condition]

Writing 1 to one of this bit will set it.

[Clearing condition]

Writing 1 to MEID.REOED will clear this bit.

RPOEE bit (P-Frame Overflow Error Enable)

[Setting condition]

Writing 1 to one of this bit will set it.

[Clearing condition]

Writing 1 to MEID.RPOED will clear this bit.

CTLEEn bits (Captured Timestamp Lost Error Enable n (n = 0 to 1))

[Setting condition]

Writing 1 to one of this bit will set it.

[Clearing condition]

Writing 1 to MEID.CTLED will clear this bit.

PDEE bit (PHY Data Error Enable)

[Setting condition]

Writing 1 to one of this bit will set it.

[Clearing condition]

Writing 1 to MEID.PDED will clear this bit.

PNAEE bit (PHY Nibble Alignment Error Enable)

[Setting condition]

Writing 1 to one of this bit will set it.

[Clearing condition]

Writing 1 to MEID.PNAED will clear this bit.

FCMCEE bit (FCS/mCRC Error Enable)

[Setting condition]

Writing 1 to one of this bit will set it.

[Clearing condition]

Writing 1 to MEID.FCMCED will clear this bit.

FFMEE bit (Final Fragment Missing Error Enable)

[Setting condition]

Writing 1 to one of this bit will set it.

[Clearing condition]

Writing 1 to MEID.FFMED will clear this bit.

CFCEE bit (C Fragment Count Error Enable)

[Setting condition]

Writing 1 to one of this bit will set it.

[Clearing condition]

Writing 1 to MEID.CFCED will clear this bit.

FRCEE bit (Fragment Count Error Enable)

[Setting condition]

Writing 1 to one of this bit will set it.

[Clearing condition]

Writing 1 to MEID.FRCED will clear this bit.

RPOOME bit (Reception Partially out of Operation Mode Enable)

[Setting condition]

Writing 1 to one of this bit will set it.

[Clearing condition]

Writing 1 to MEID.RPOOMD will clear this bit.

FFE bit (Frame Filtered Enable)

[Setting condition]

Writing 1 to one of this bit will set it.

[Clearing condition]

Writing 1 to MEID.FFD will clear this bit.

FUEE bit (Undersize Error Enable)

[Setting condition]

Writing 1 to one of this bit will set it.

[Clearing condition]

Writing 1 to MEID.FUED will clear this bit.

FOEE bit (Oversize Error Enable)

[Setting condition]

Writing 1 to one of this bit will set it.

[Clearing condition]

SW: Writing 1 to MEID.FOED will clear this bit.

33.4.3.3 MEID : Error Interrupt Disable Register

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0208

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	FOED	FUED	FFD	RPOOMD	FRCED	CFCED	FFMED	FCMCED	PNAED	PDED	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	CTLED1 to CTLED0		—	RPOED	REOED	FCED	BFED	TBCID	TCED	FCDD	PFRROD	PRED	—	TSLD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TSLD	Transmission Stream Lost Disable Writing 1 to this register will clear MEIE.TSLE register.	R/W*1
1	—	This bit is read as 0.	R
2	PRED	Pause or PFC Frame Reception Error Disable Writing 1 to this register will clear MEIE.PREE register.	R/W*1
3	PFRROD	Pause or PFC Frame Retransmit Retry Over Disable Writing 1 to this register will clear MEIE.PFRROE register.	R/W*1

Bit	Symbol	Function	R/W
4	FCDD	False Carrier Detection Disable Writing 1 to this register will clear MEIE.FCDE register.	R/W*1
5	TCED	Tx CRC Error Disable Writing 1 to this register will clear MEIE.TCEE register.	R/W*1
6	TBCID	TX Bad CRC Insertion Disable Writing 1 to this register will clear MEIE.TBCIE register.	R/W*1
7	BFED	Bad Fragment Error Disable Writing 1 to this register will clear MEIE.BFEE register.	R/W*1
8	FCED	Frame Count Error Disable Writing 1 to this register will clear MEIE.FCEE register.	R/W*1
9	REOED	E-Frame Overflow Error Disable Writing 1 to this register will clear MEIE.REOEE register.	R/W*1
10	RPOED	P-Frame Overflow Error Disable Writing 1 to this register will clear MEIE.RPOEE register.	R/W*1
11	—	This bit is read as 0.	R
13:12	CTLED1 to CTLED0	Captured Timestamp Lost Error Disable n (n = 0 to 1) Writing 1 to this register will clear MEIE.CTLEE register.	R/W*1
19:14	—	These bits are read as 0.	R
20	PDED	PHY Data Error Disable Writing 1 to this register will clear MEIE.PDEE register.	R/W*1
21	PNAED	PHY Nibble Alignment Error Disable Writing 1 to this register will clear MEIE.PNAEE register.	R/W*1
22	FCMCED	FCS/mCRC Error Disable Writing 1 to this register will clear MEIE.FCMCEE register.	R/W*1
23	FFMED	Final Fragment Missing Error Disable Writing 1 to this register will clear MEIE.FFMEE register.	R/W*1
24	CFCED	C Fragment Count Error Disable Writing 1 to this register will clear MEIE.CFCEE register.	R/W*1
25	FRCED	Fragment Count Error Disable Writing 1 to this register will clear MEIE.FCEE register.	R/W*1
26	RPOOMD	Reception Partially out of Operation Mode Disable Writing 1 to this register will clear MEIE.RPOOME register.	R/W*1
27	FFD	Frame Filtered Disable Writing 1 to this register will clear MEIE.FFE register.	R/W*1
28	FUED	Undersize Error Disable Writing 1 to this register will clear MEIE.FUEE register.	R/W*1
29	FOED	Oversize Error Disable Writing 1 to this register will clear MEID.FOEE register.	R/W*1
31:30	—	These bits are read as 0.	R

Note 1. Read value differs from written value.

33.4.3.4 MMIS0 : Monitoring Interrupt Status Register 0

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0210

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

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

Bit	Symbol	Function	R/W
0	PLSCS*2*3	PHY Link Signal Change Status Flag	R/W*1
1	PIDS*2*4	PHY Interrupt Detection Status Flag	R/W*1
2	LVSS*2*5	Link Verification Succeed Status Flag	R/W*1
3	LVFS*2*5	Link Verification Failed Status Flag	R/W*1
4	VFRS*2*5	Verify Frame Reception Status Flag	R/W*1
31:5	—	These bits are read as 0.	R

Note: This interrupt register is mapped to rmc_mmis0_int interrupt wire.
Note 1. Read value differs from written value.
Note 2. SW: Writing 0 to this bit has no effect.
Note 3. SW: To check the current link state, refer to the link status pin state bit in the PHY link status register (MPIM.PLS).
Note 4. HW: This flag could be asserted unintentionally before configuring PHY interrupt polarity (MPIC.PIP) into the right value. In the phase, this flag should be ignored.
Note 5. It’s no effect when suspending, e.g. an upper layer (e.g. MHD) is not in OPERATION state.

PLSCS bit (PHY Link Signal Change Status Flag)

[Setting condition]

HW: When the value of link status pin (rmc_phy_link port) is changed.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

PIDS bit (PHY Interrupt Detection Status Flag)

[Setting condition]

HW: PHY interrupt pin (rmc_phy_int port) is active level.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

LVSS bit (Link Verification Succeed Status Flag)

[Setting condition]

HW: When Link Verification process succeed.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

LVFS bit (Link Verification Failed Status Flag)

[Setting condition]

HW: When Link Verification process failed. It means that the remote node is not able to handle Preemptable frames.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

VFRS bit (Verify Frame Reception Status Flag)

[Setting condition]

HW: When valid verify frame has been received.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

33.4.3.5 MMIE0 : Monitoring Interrupt Enable Register 0

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0214

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

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

Bit	Symbol	Function	R/W
0	PLSCE	PHY Link Signal Change Enable Writing 1 to this register will set it. Writing 0 to this register has no effect.	R/W*1
1	PIDE	PHY Interrupt Detection Enable Writing 1 to this register will set it. Writing 0 to this register has no effect.	R/W*1
2	LVSE	Link Verification Succeed Enable Writing 1 to this register will set it. Writing 0 to this register has no effect.	R/W*1
3	LVFE	Link Verification Failed Enable Writing 1 to this register will set it. Writing 0 to this register has no effect.	R/W*1
4	VFRE	Verify Frame Reception Enable Writing 1 to this register will set it. Writing 0 to this register has no effect.	R/W*1
31:5	—	These bits are read as 0.	R

Note 1. Read value differs from written value.

33.4.3.6 MMID0 : Monitoring Interrupt Disable Register 0

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0218

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

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

Bit	Symbol	Function	R/W
0	PLSCD	PHY Link Signal Change Disable Writing 1 to this register will clear the corresponding bit in the corresponding enable register. Writing 0 to this register has no effect.	R/W *1
1	PIDD	PHY Interrupt Detection Disable Writing 1 to this register will clear the corresponding bit in the corresponding enable register. Writing 0 to this register has no effect.	R/W *1
2	LVSD	Link Verification Succeed Disable Writing 1 to this register will clear the corresponding bit in the corresponding enable register. Writing 0 to this register has no effect.	R/W *1
3	LVFD	Link Verification Failed Disable Writing 1 to this register will clear the corresponding bit in the corresponding enable register. Writing 0 to this register has no effect.	R/W *1
4	VFRD	Verify Frame Reception Disable Writing 1 to this register will clear the corresponding bit in the corresponding enable register. Writing 0 to this register has no effect.	R/W *1
31:5	—	These bits are read as 0.	R

Note 1. Read value differs from written value.

33.4.3.7 MMIS1 : Monitoring Interrupt Status Register 1

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0220

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

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

Bit	Symbol	Function	R/W
0	PRACS*2	PHY Read Access Completed Status Flag	R/W*1
1	PWACS*2	PHY Write Access Completed Status Flag	R/W*1
2	PAACS*2	PHY Address Access Completed Status Flag	R/W*1
3	PPRACS*2	PHY Post-Read Access Completed Status Flag	R/W*1

Bit	Symbol	Function	R/W
31:4	—	These bits are read as 0.	R

Note: This interrupt register is mapped to rmc_mdio_int interrupt wire.
Note 1. Read value differs from written value.
Note 2. Writing 0 to this bit has no effect.

PRACS bit (PHY Read Access Completed Status Flag)

[Setting condition]

HW: When PHY read access has completed. See section 8.4.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

PWACS bit (PHY Write Access Completed Status Flag)

[Setting condition]

HW: When PHY write access has completed. See section 8.4.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

PAACS bit (PHY Address Access Completed Status Flag)

[Setting condition]

HW: When PHY address access has completed. See section 8.4.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

PPRACS bit (PHY Post-Read Access Completed Status Flag)

[Setting condition]

HW: When PHY post-read access has completed. See section 8.4.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

33.4.3.8 MMIE1 : Monitoring Interrupt Enable Register 1

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0224

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

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

Bit	Symbol	Function	R/W
0	PRACE*2	PHY Read Access Completed Enable	R/W*1
1	PWACE*2	PHY Write Access Completed Enable	R/W*1
2	PAACE*2	PHY Address Access Completed Enable	R/W*1
3	PPRACE*2	PHY Post-Read Access Completed Enable	R/W*1
31:4	—	These bits are read as 0.	R

Note 1. Read value differs from written value.
Note 2. SW: Writing 0 to this bit has no effect.

PRACE bit (PHY Read Access Completed Enable)

[Setting condition]

HW: Writing 1 to this bit will set it.

[Clearing condition]

HW: Writing 1 to MMID1.PRACD register will clear this bit.

PWACE bit (PHY Write Access Completed Enable)

[Setting condition]

HW: Writing 1 to this bit will set it.

[Clearing condition]

HW: Writing 1 to MMID1.PWACD register will clear this bit.

PAACE bit (PHY Address Access Completed Enable)

[Setting condition]

HW: Writing 1 to this bit will set it.

[Clearing condition]

HW: Writing 1 to MMID1.PAACD register will clear this bit.

PPRACE bit (PHY Post-Read Access Completed Enable)

[Setting condition]

HW: Writing 1 to this bit will set it.

[Clearing condition]

HW: Writing 1 to MMID1.PPRACD register will clear this bit.

33.4.3.9 MMID1 : Monitoring Interrupt Disable Register 1

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0228

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

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

Bit	Symbol	Function	R/W
0	PRACD*1	PHY Read Access Completed Disable Writing 1 to one of this bit will clear MMIE1.PRACE register.	W
1	PWACD*1	PHY Write Access Completed Disable Writing 1 to one of this bit will clear MMIE1.PWACE register.	W
2	PAACD*1	PHY Address Access Completed Disable Writing 1 to one of this bit will clear MMIE1.PAACE register.	W
3	PPRACD*1	PHY Post-Read Access Completed Disable Writing 1 to one of this bit will clear MMIE1.PPRACE register.	W
31:4	—	These bits are read as 0.	R

Note 1. Writing 0 to this register has no effect.

33.4.3.10 MMIS2 : Monitoring Interrupt Status Register 2

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0230

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

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

Bit	Symbol	Function	R/W
0	MPDIS*2*3*4*5	Magic Packet Detection Interrupt Status Flag	R/W*1
1	LPIAIS*2*5	LPI Assertion Interrupt Status Flag	R/W*1
2	LPIDIS*2*5	LPI De-Assertion Interrupt Status Flag	R/W*1
31:3	—	These bits are read as 0.	R

Note:
- This interrupt register is mapped to rmc_mp_int, rmc_lpi_start_int and rmc_lpi_stop_int interrupt wire.
- This register's value goes through the synchronizer between clk and clk_phy_rx.
  The result of write access is reflected to each SFR bit after "5clk + 4clk_phy_rx" in Worst Case in RMAC.
  Do not make write access continuously. Please wait following time for re-write access.

Note 1. Read value differs from written value.
Note 2. SW: Writing 0 to this bit has no effect.
Note 3. HW: Magic Packet in Preemptable frames is ignored.
Note 4. HW: CRC of magic packet is not checked.
Note 5. HW: It's no effect when suspending.

MPDIS bit (Magic Packet Detection Interrupt Status Flag)

[Setting condition]

HW: When a Magic Packet has been detected.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

LPIAIS bit (LPI Assertion Interrupt Status Flag)

[Setting condition]

HW: When LPI request is detected. This register is continuously asserted when LPI request continues.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

LPIDIS bit (LPI De-Assertion Interrupt Status Flag)

[Setting condition]

HW: When LPI request deassertion.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

33.4.3.11 MMIE2 : Monitoring Interrupt Enable Register 2

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0234

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

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

Bit	Symbol	Function	R/W
0	MPDIE	Magic Packet™ Detection Interrupt Enable Writing 1 to this register will set it. Writing 0 to this register has no effect.	R/W*1
1	LPIAIE	LPI Assertion Interrupt Enable Writing 1 to this register will set it. Writing 0 to this register has no effect.	R/W*1
2	LPIDIE	LPI De-Assertion Interrupt Enable Writing 1 to this register will set it. Writing 0 to this register has no effect.	R/W*1
31:3	—	These bits are read as 0.	R

Note: This register’s value goes through the synchronizer between clk and clk_phy_rx.
The result of write access is reflected to each SFR bit after “5clk + 4clk_phy_rx” in Worst Case in RMAC.
Do not make write access continuously. Please wait following time for re-write access.

Note 1. Read value differs from written value.

33.4.3.12 MMID2 : Monitoring Interrupt Disable Register 2

Base address: RMACm = 0x403C_B000 + 0x2000 × m (m = 0, 1)
RMACm_NS = 0x503C_B000 + 0x2000 × m (m = 0, 1)

Offset address: 0x0238

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

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

Bit	Symbol	Function	R/W
0	MPDID	Magic Packet™ Detection Interrupt Disable Writing 1 to this register will clear the corresponding bit in the corresponding enable register. Writing 0 to this register has no effect.	W
1	LPIAID	LPI Assertion Interrupt Disable Writing 1 to this register will clear the corresponding bit in the corresponding enable register. Writing 0 to this register has no effect.	W
2	LPIDID	LPI De-Assertion Interrupt Disable Writing 1 to this register will clear the corresponding bit in the corresponding enable register. Writing 0 to this register has no effect.	W
31:3	—	These bits are read as 0.	R

Note: This register’s value goes through the synchronizer between clk and clk_phy_rx.
The result of write access is reflected to each SFR bit after “5clk + 4clk_phy_rx” in Worst Case in RMAC.
Do not make write access continuously. Please wait following time for re-write access.

33.5 Register Utilizations

33.5.1 Operation Modes

Operational mode is controlled. Please refer to IPS. RMAC received a mode control signal for controlling operational mode of RMAC.

33.5.1.1 Frame Handling

In transmission, this IP transmits all frames from an upper module (e.g. MHD) when this IP is in operation state, e.g. when MHD is in OPERATION state. This IP receives an e-frame and a p-frame from the MHD.

In reception, this IP only receives frames from PHY when this IP is in operation state, e.g. when MHD is in OPERATION state., and outputs them to the upper module (e.g. MHD). In other modes, this IP discards all received frames from PHY internally, and does not output them to the upper module.

When the system supports preemption and the Mode change is executed, this IP of the Rx side behaves like the following figures.

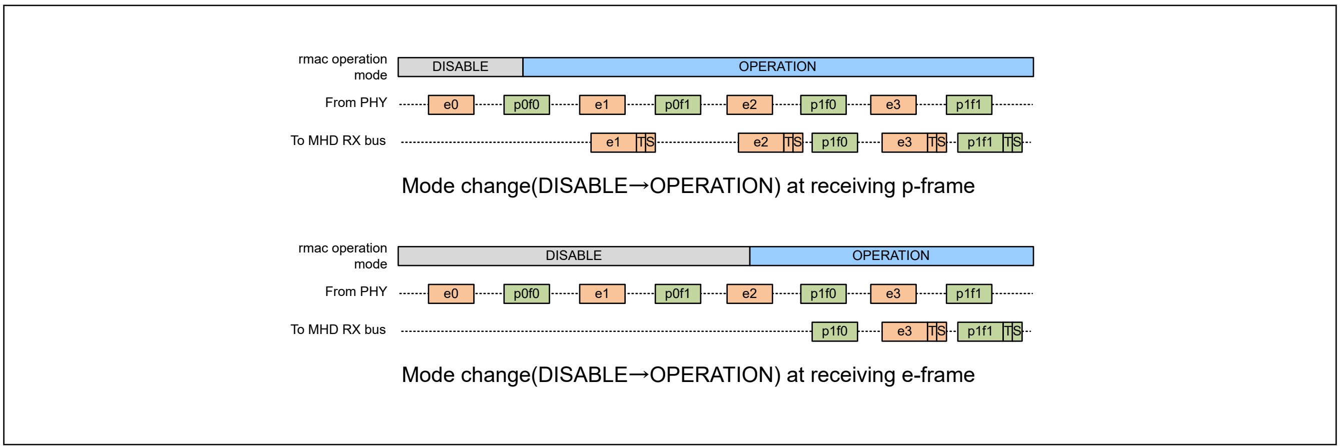


Figure 33.3 Mode change (DISABLE→OPERATION) on Rx

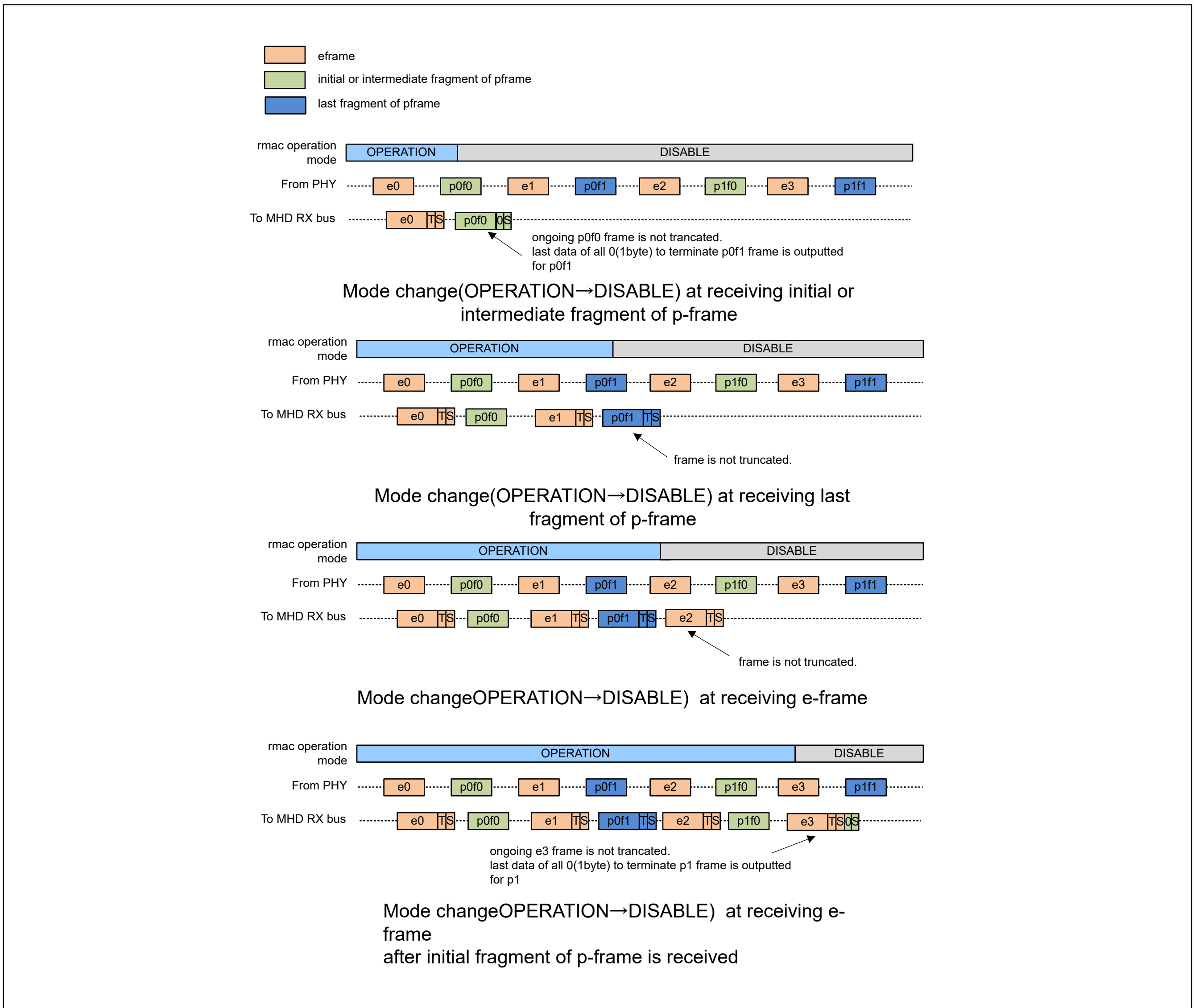


Figure 33.4 Mode change (OPERATION→DISABLE) on Rx

33.5.2 Software Flows

Restrictions:

SW: Please follow to the flow in this section.

33.5.2.1 Software Flow Legend

Software flow legend is described in Figure 33.5.

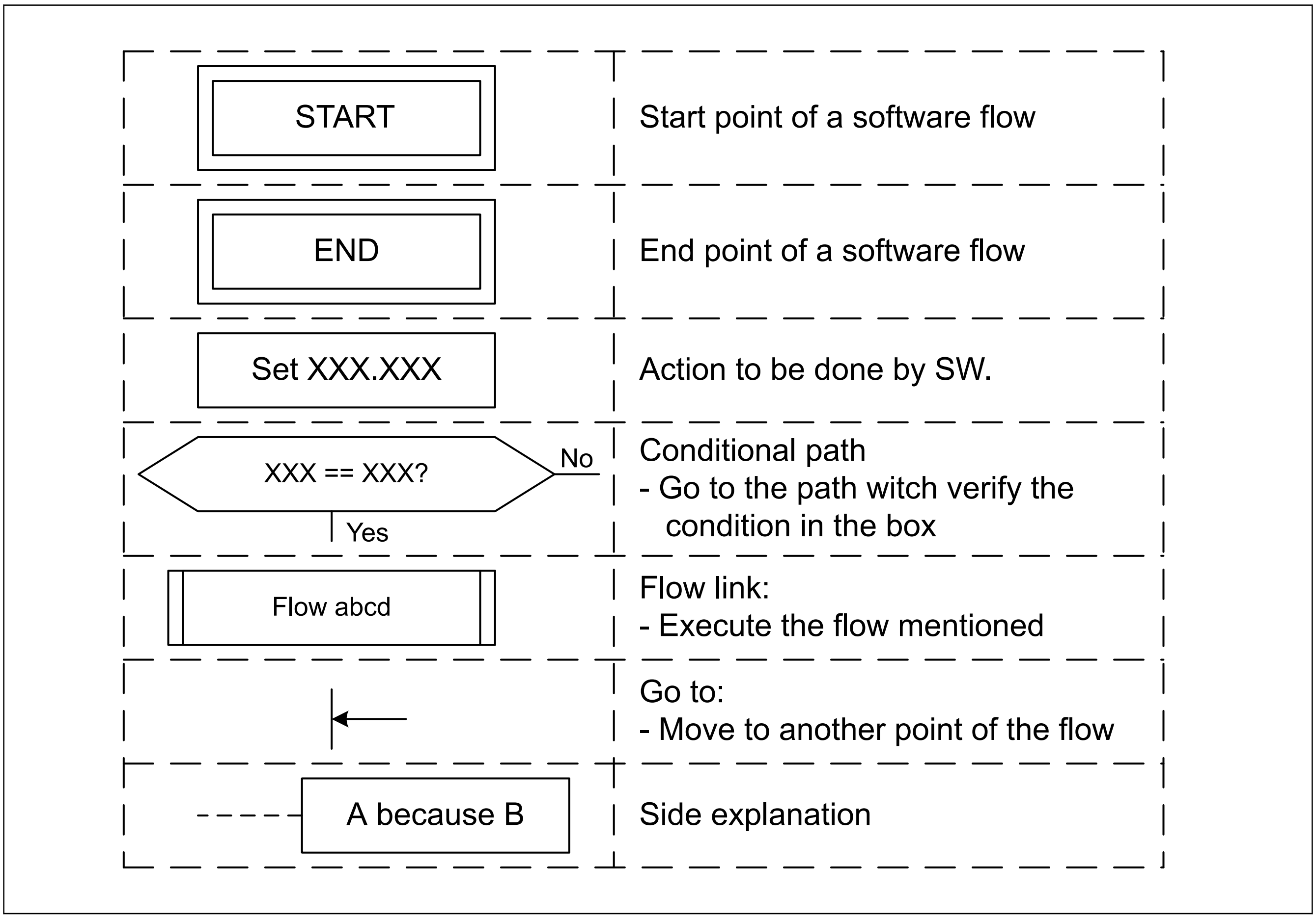


Figure 33.5 Software flow legend

33.5.2.2 Initialization Flow

The figure below shows the initialization flow.

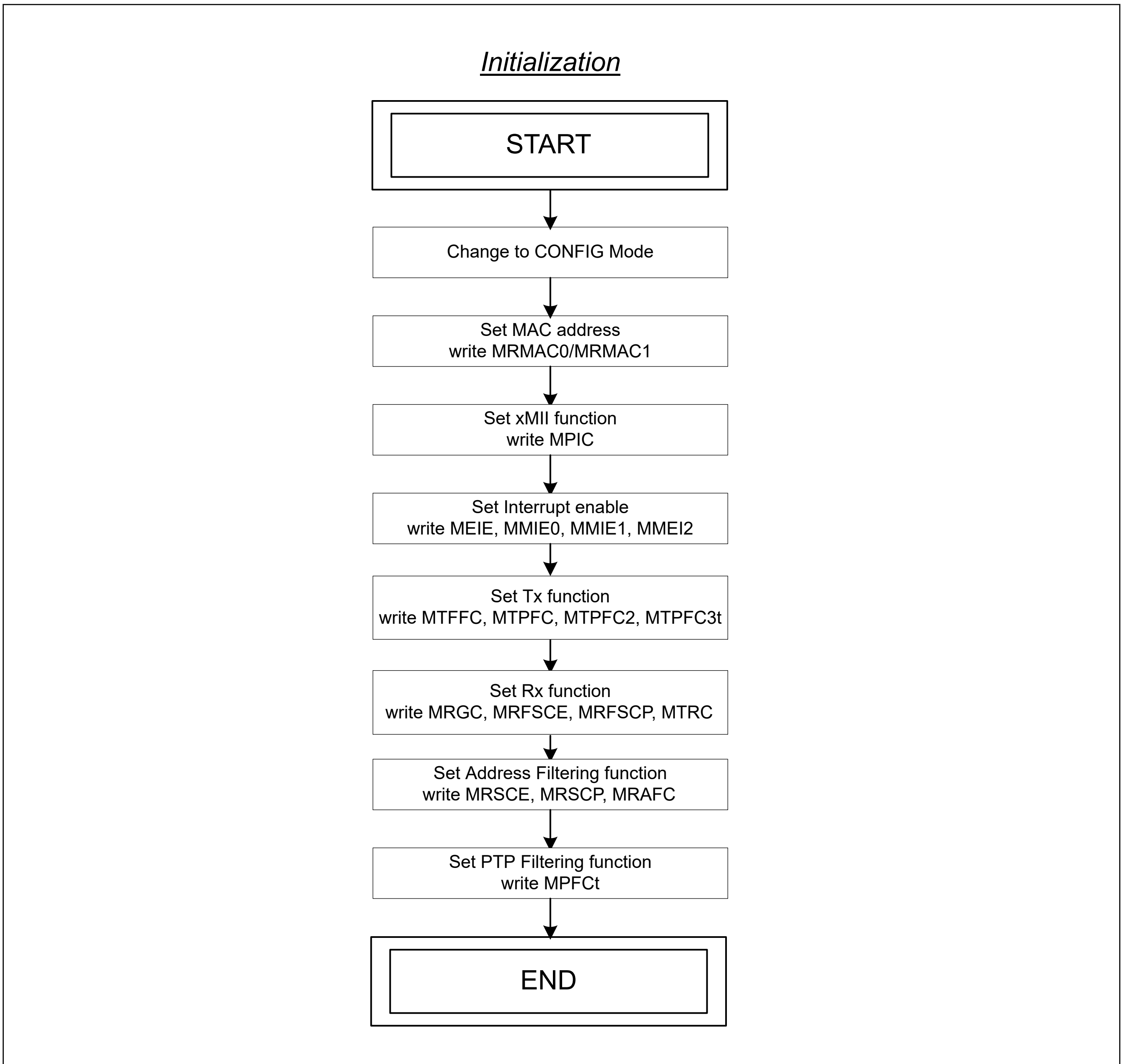


Figure 33.6 Initialization flow

33.5.2.3 Interrupt Handling Flow

Interrupt handling flow can be used in any mode except RESET mode.

The figure below shows the interrupt handling flow.

Restrictions:

This flow is not usable in RESET modes.

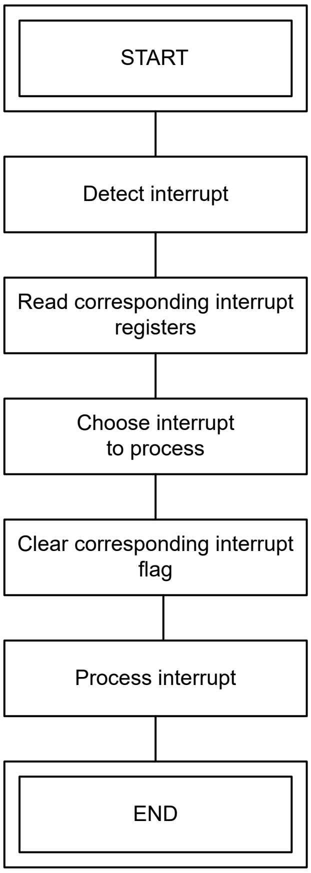


Figure 33.7 Interrupt handling flow

33.5.2.4 PHY MDIO and eMDIO Access Flows

The figure below shows the PHY MDIO and eMDIO access flow.

Restrictions:

This flow is not usable in RESET modes.

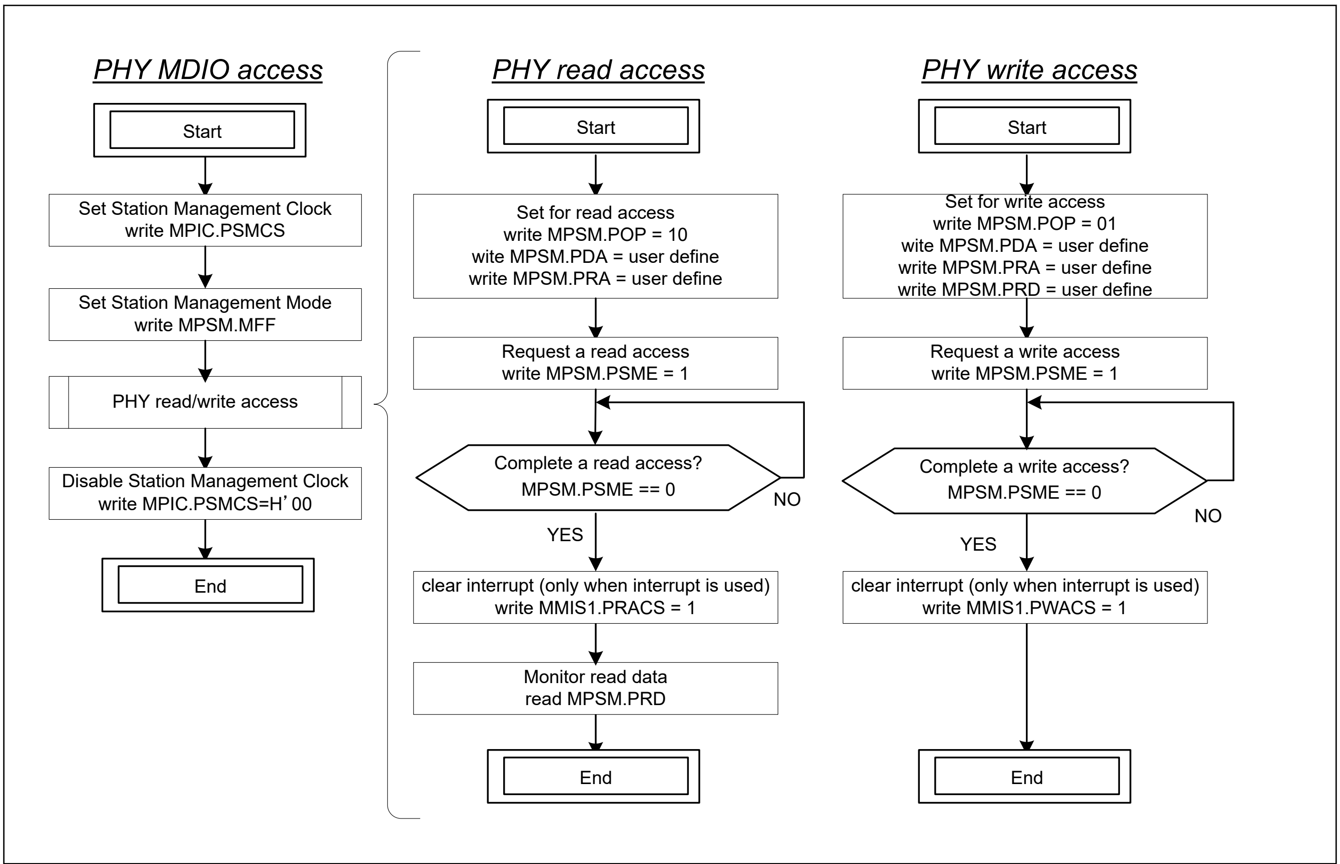


Figure 33.8 PHY MDIO access flow

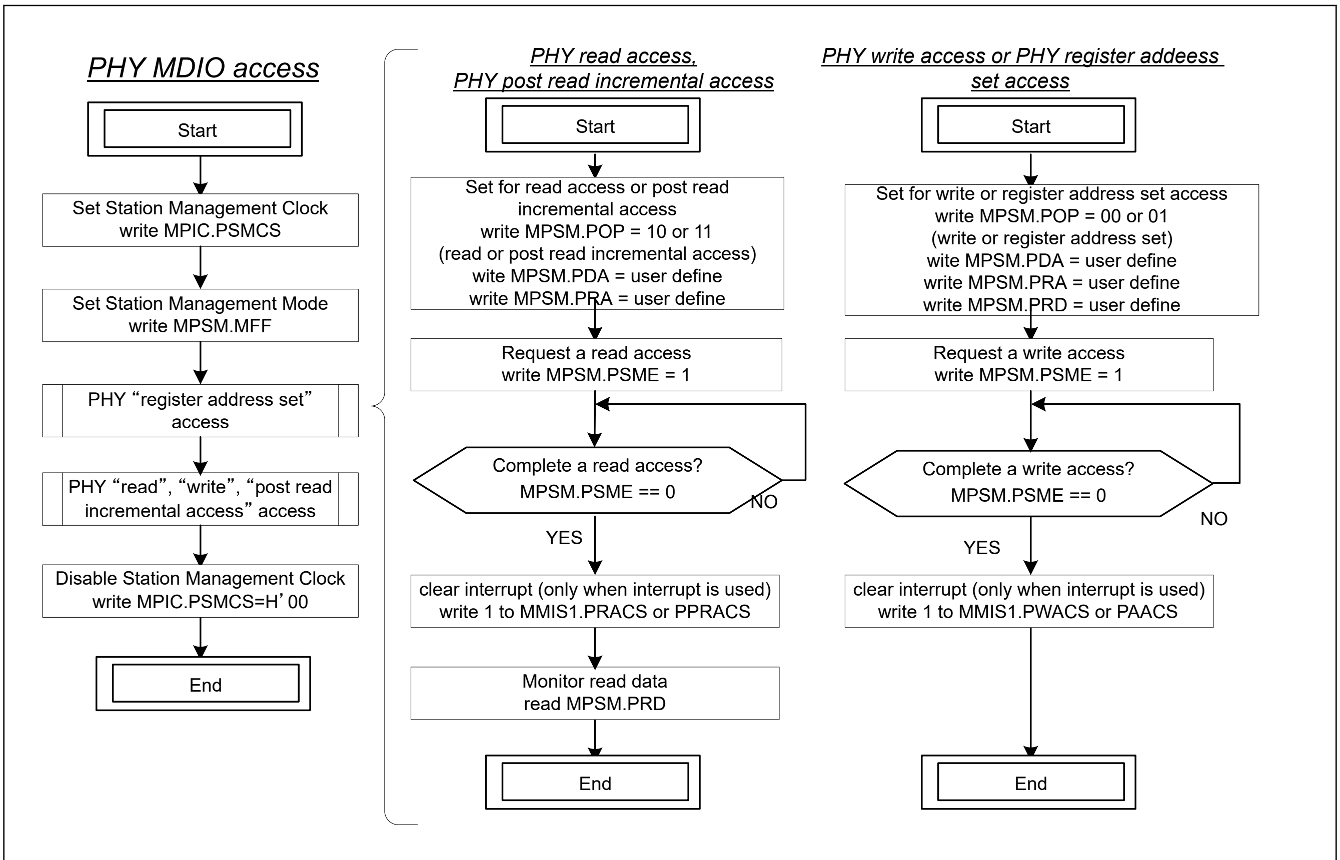


Figure 33.9 PHY eMDIO access flow

33.5.2.5 LPI Transition for Tx Port

Figure 33.10 shows the LPI transition flow. When resumed from LPI, the transmission should be requested until PHY wake-up completed.

Restrictions:

This flow is not usable in RESET modes.

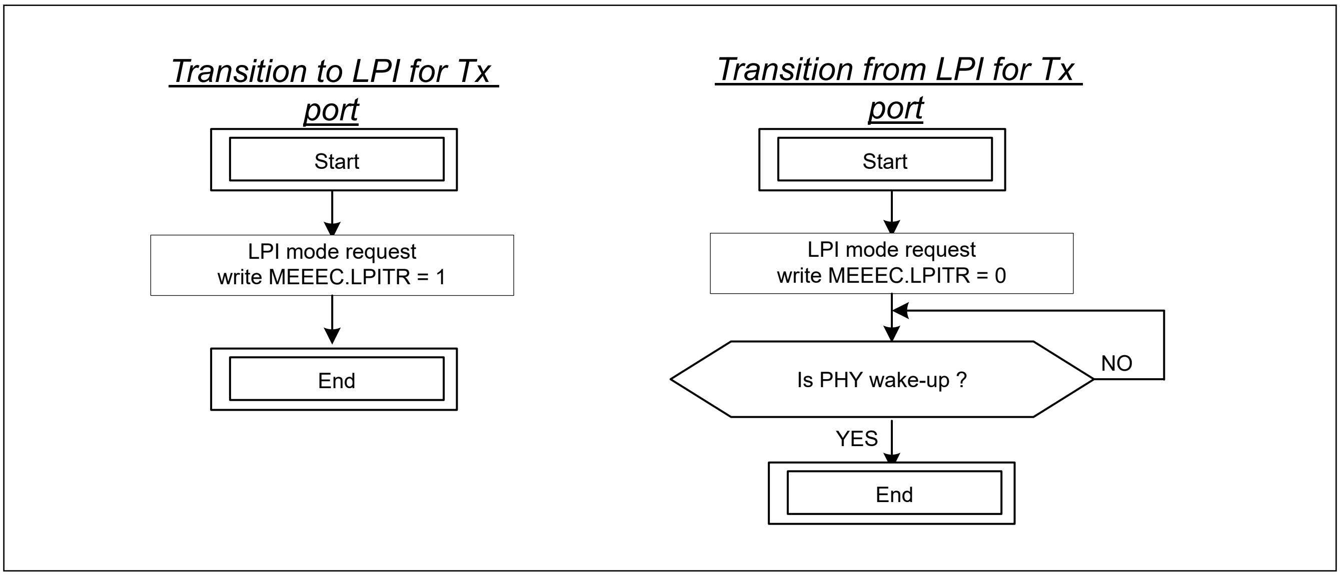


Figure 33.10 LPI transition for Tx port

33.5.2.6 LINK Verification

The figure below shows Link Verification flow.

Restrictions:

This flow is usable in OPERATIONAL mode.

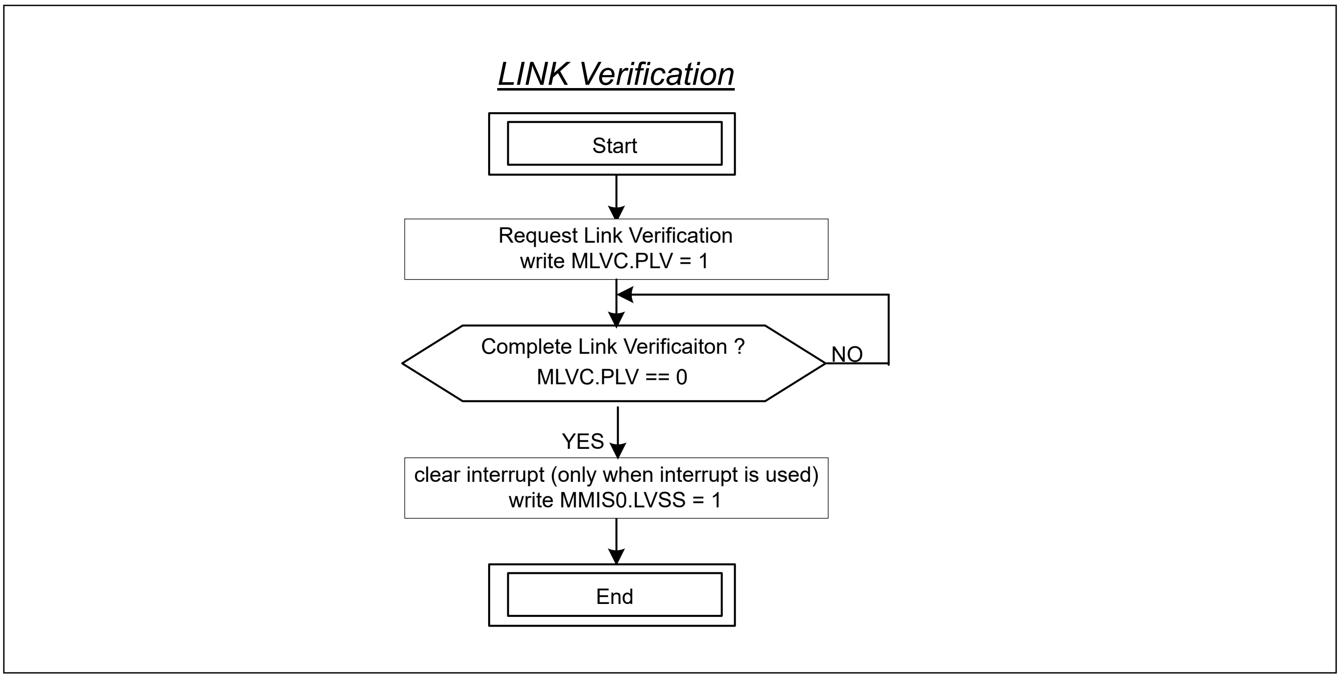


Figure 33.11 Link Verification flow

33.5.2.7 Called Software Flows

The flows described in this section can only be called from other flows thanks to a "flow link" box (Figure 33.5) and cannot be used alone.

33.5.3 Register Writable Without Software Flow

This section describes registers that have not been described so far. These registers can be changed dynamically. (However, it is necessary that the initial settings such as the clock enabling have been completed.)

33.6 Functional Details

33.6.1 Preemption

"Preemption" is the function specified in IEEE802.3.br. This function separates an ethernet frame to several fragments like Figure 33.12. This figure shows an example of preemption of 3 fragments. Preemption for only two fragments or over three fragments is possible.

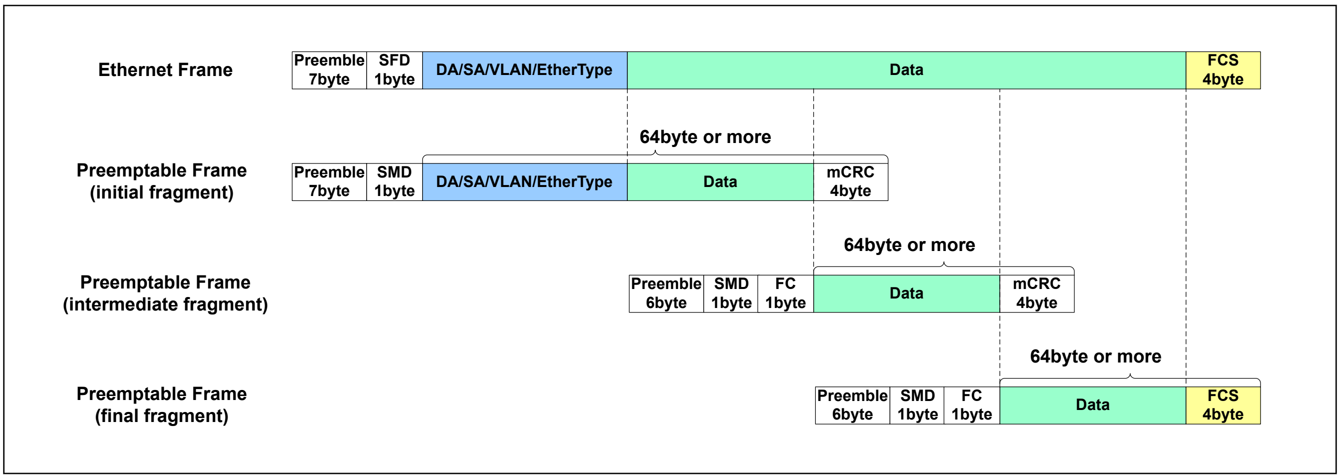


Figure 33.12 Outline of preemption

This function is handled by this IP and an upper module (e.g. MHD).

On the Tx Side, This IP only generates the fields of "Preamble", "SMD", "FC" and "mCRC" of the Preemptable frame complying with the rule of IEEE802.3.br. The upper module (e.g. MHD) has to separate a transmit frame appropriately.

On the Rx Side, this IP removes the fields of "Preamble", "SMD", "FC" and "mCRC" of the Preemptable frame and outputs the fields of "DA/SA/VLAN/EtherType", "Data" and "FCS" (To output FCS part whether or not it is selectable by MRGC.RCPT) to upper module. The upper module (e.g. MHD) has to re-assemble a received frame.

"Preemption" is effective to prevent decreasing of throughput for high priority frame by transferring of low priority frames. Figure 33.13 shows one of examples. The software does not care the length of Low priority frames.

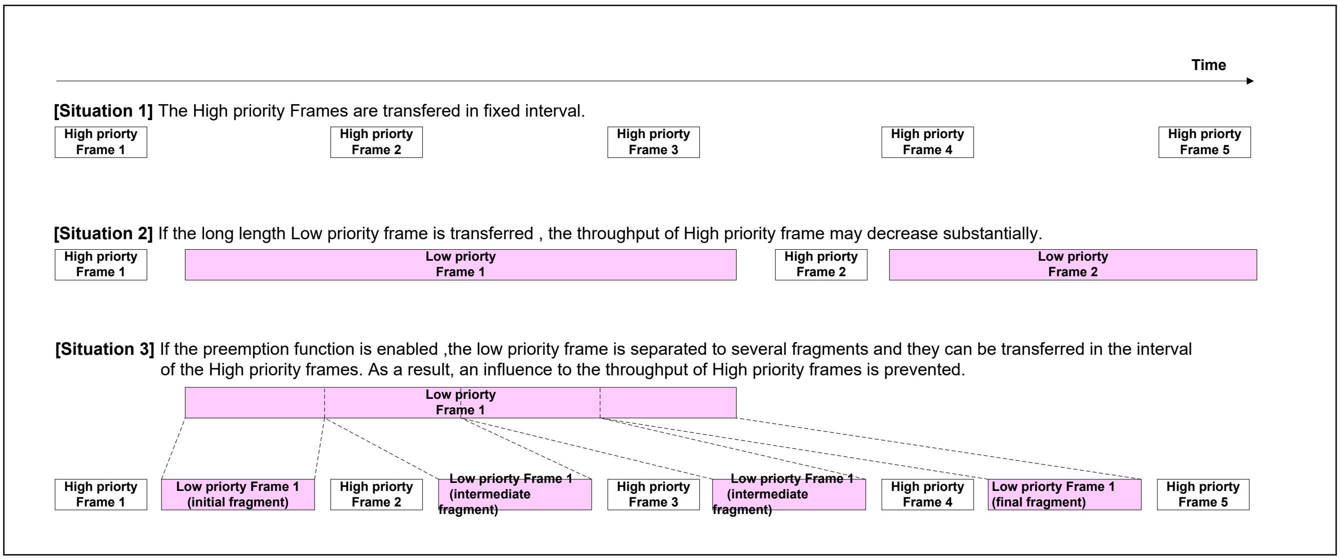


Figure 33.13 Effective of Preemption

To use “Preemption”, the Link Verification process should be executed on the system level.

Note: This IP does not hold an information whether the preemption function is valid or not in the system.

33.6.1.1 Link Verification

This IP supports Link Verification process of Figure 33.14.

By setting MLVC.PASE this IP answers to a Verify frame by a Response frame to authorize the linked IP that preemption is possible. When a valid Verify frame is received, MMIS.VFRS is asserted as an interrupt.

When MLVC.PLV is set to 1, this IP starts to transmit Verify frame. The transmission attempts 3 times with the configured interval (MLVC.LVT) until Response frame is received from remote node. If the received Response frame is valid, MMIS.LVSS is asserted as an interrupt.

If not receiving Response frame after attempted 3 times, MMIS.LVFS is asserted as an interrupt.

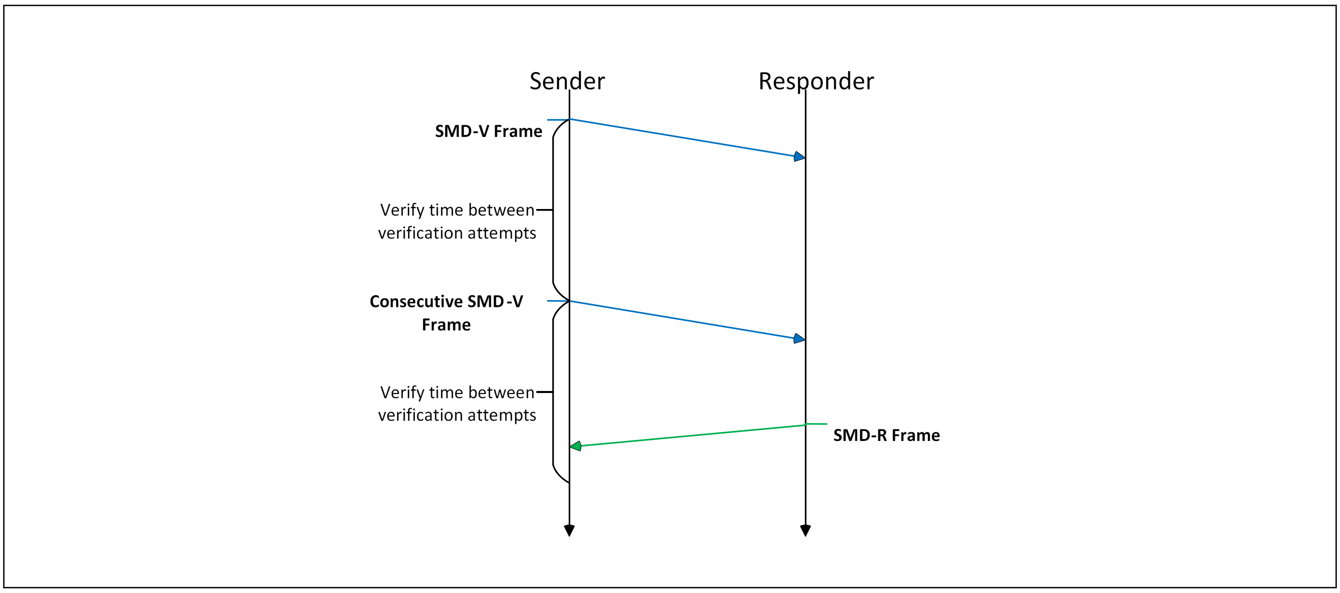


Figure 33.14 Example of link verification flow

33.6.2 Flow Control

This IP supports the following two types of Flow Control functions.

1. Flow Control specified in IEEE 802.3 for a full-duplex operation using a PAUSE frame.

2. Priority Based Flow Control specified in IEEE 802.1Qbb for a full-duplex operation using a PFC frame.

33.6.2.1 Flow Control by PAUSE Frame

This Flow Control is only for a full-duplex operation and this consists of Transmission and Reception of a PAUSE frame.

33.6.2.1.1 Transmission of PAUSE Frame

1. While MTFFC.FCM = 0 and MTPFC.PT[15:0] ≠ 0, the transmission function of a PAUSE frame is enabled. This IP transmits a PAUSE frame as described in Figure 33.15 by asserting a transmission request. The value of MTPFC.PT[15:0] is used for "PAUSE_TIME" field, and MRMAC0 and MRMAC1 are used for "SA" field.

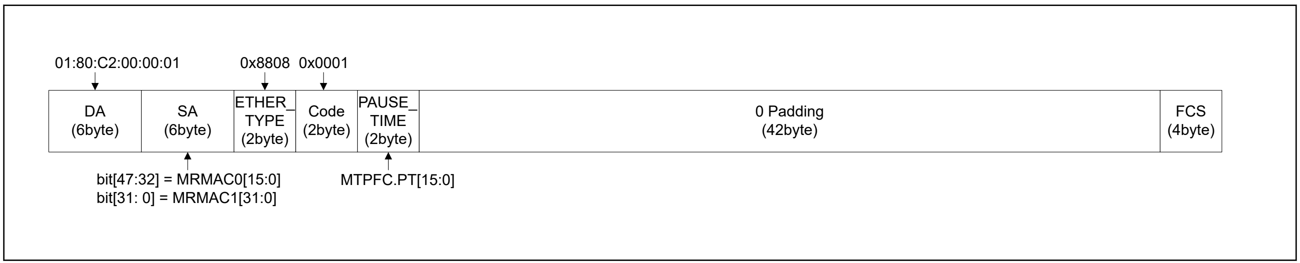


Figure 33.15 PAUSE frame format

- Auto PAUSE Frame
  When MTPFC.PFM = 0, this IP uses a hardware pause transmission request as a transmit request. After transmitting the PAUSE Frame, MAPFTCT is incremented.
- Manual PAUSE Frame
  When MTPFC.PFM = 1, this IP uses MTPFC2.MPFR as a transmit request. After transmitting the PAUSE Frame, MMPFTCT is incremented.

When the transmit request of the PAUSE frame is asserted while the IP is transmitting the other frame, the PAUSE frame is transmitted after the transmitting frame.

2. While the transmit request is asserted, a PAUSE frame is continuously re-transmitted at a fixed interval of time configured by MTPFC.PFRT. If the number of re-transmission reaches the value of MTPFC.PFRLV, this IP sets MEIS.PFRROS to 1 for an interrupt. Even if the interrupt is detected, this IP continues re-transmission of a PAUSE frame while the transmit request is asserted.
3. After deasserting the transmission request, this IP stops a transmission of PAUSE frame. At that time, when MTPFC2.PFTTZ = 1 and an internal counter to count a pausing-period exceeds the value of MTPFC.PFRT (*), this IP transmits one more a PAUSE frame with "PAUSE_TIME" field as 0.

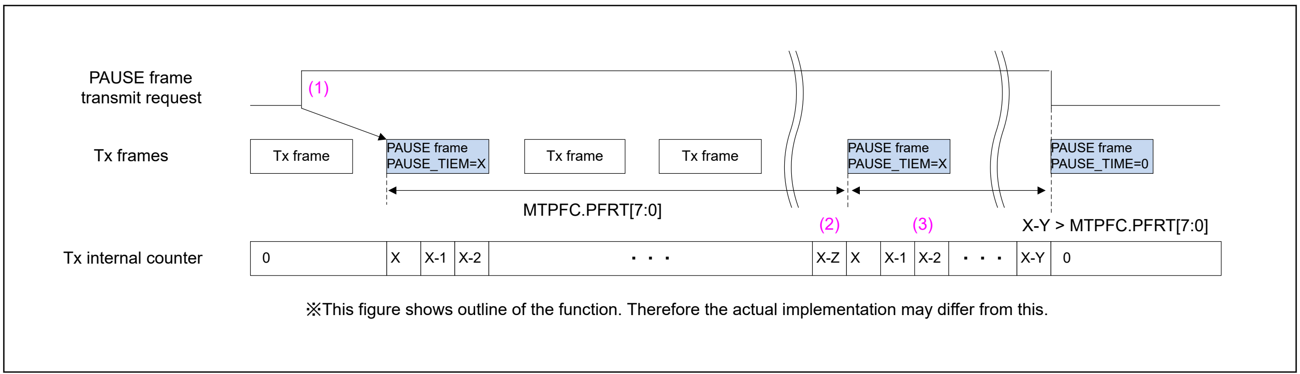


Figure 33.16 Transmission of PAUSE frame

33.6.2.1.2 Reception of PAUSE Frame

1. While MRGC.PFRC = 1, this IP receives a PAUSE frame and it starts internally counting a pausing-period specified by "PAUSE_TIME" field of a received PAUSE frame. While counting the pausing-period, notification of pause frame reception to an upper layer (e.g. MHD) and MRPFM.PTCA is asserted. And any transmissions except for a

Verify_packet, a Respond_packet, and a PAUSE frame are inhibited. When this IP receives a PAUSE frame while transmitting any frames, transmissions are inhibited after the transmission completion.

2. When a PAUSE frame with "PAUSE_TIME ≠ 0" is received again before expiring the internal pausing-period, the internal counter is updated to that value.
3. When the internal pausing-period is expired, notification of pause frame reception to an upper layer (e.g. MHD) and MRPFM.PTCA is deasserted.
4. When MRGC.PFRTZ = 1 and PAUSE frame with "PAUSE_TIME = 0" is received before expiring the internal pause-period, the internal counter is cleared. Then notification of pause frame reception to an upper layer (e.g. MHD) and MRPFM.PTCA is deasserted.
In any case, MPFRCT is incremented after receiving a PAUSE frame in MRGC.PFRC = 1. And the received PAUSE frames are discarded internally.

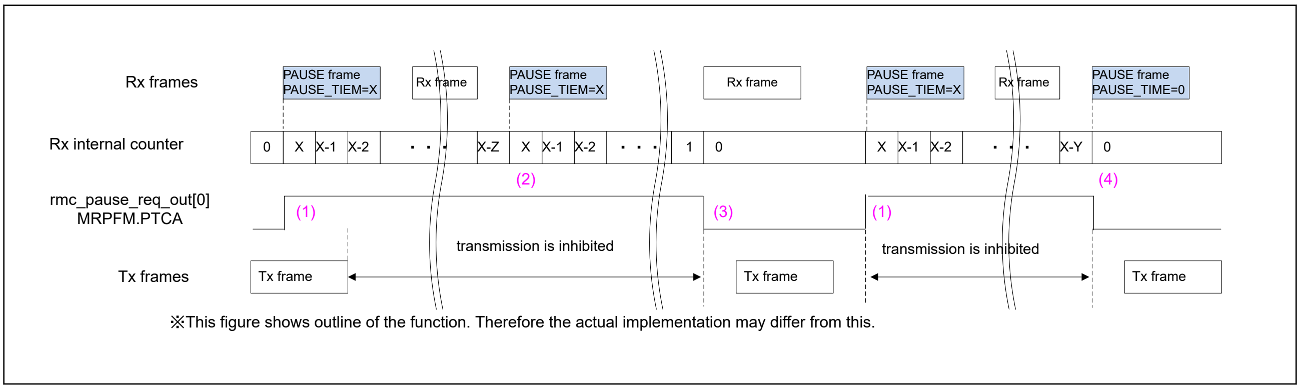


Figure 33.17 Reception of PAUSE frame

33.6.2.2 Flow Control by PFC Frame

This Flow Control is for a full-duplex operation and this consists of Transmission and Reception of a PFC frame.

33.6.2.2.1 Transmission of PFC Frame

When MTFFC.FCM = 1 and MTPFC.PT[15:0] ≠ 0, the transmission function of a PFC frame is enabled.

This IP treats several priorities as one priority-group to transmission of the PFC frame. The number of priority-group is 2. Each priority-group is configured by MTPFC30 and MTPFC31.

This IP transmits a PFC frame as described in Figure 33.18 for each priority-group by asserting a transmission request. The value of MTPFC.PT[15:0] is used for all "PAUSE_TIME" fields. "Priority_Enable" field value is different for each priority-group and the values of MTPFC30.PFCPG and MTPFC31.PFCPG are used for these. And MRMAC0 and MRMAC1 are used for "SA" field. When MTPFC30.PFCPG (or MTPFC31.PFCPG) is set to all 0, this IP does not transmit a PFC frame for the priority-group 0 (or priority-group 1) even when a transmission request is asserted.

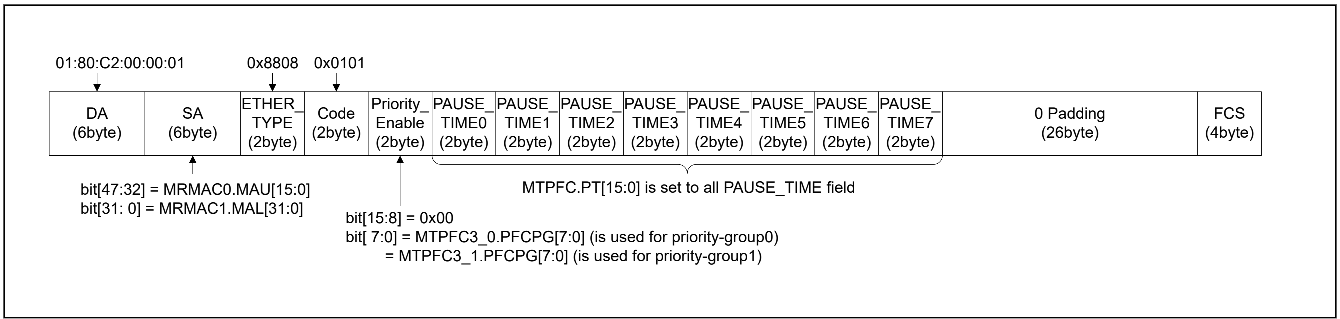


Figure 33.18 PFC frame Format

- Auto PFC Frame
  When MTPFC.PFM = 0, this IP uses a hardware pause transmission request as a transmit request. The bit[0] is for priority-group0 and the bit[1] is for priority-group1. MAPCFTCT0 and MAPCFTCT1 are incremented respectively after transmitting a PFC Frame of each priority-group.

- Manual PFC Frame
  When MTPFC.PFM = 1, this IP uses MTPFC2.MPFCFRn as a transmit request. The bit[0] is for priority-group 0 and the bit[1] is for priority-group 1. MMPCFTCT0 and MMPCFTCT1 are incremented respectively after transmitting a PFC Frame of each priority-group.

If several transmit requests are asserted at same time, one PFC frame which has merged priority_enable information is transmitted.

If several transmit requests are asserted while transmitting other frames, an internal counter is cleared and one PFC frame which has merged priority_enable information is transferred.

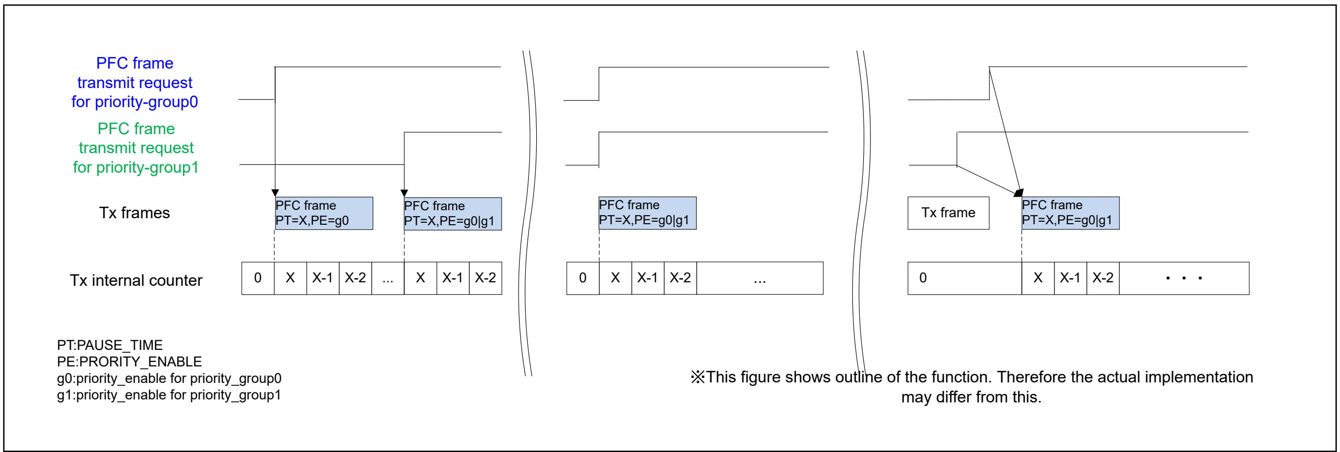


Figure 33.19 Transmitting PFC frame (part 1)

While the transmit request is asserted, a PFC frame corresponding to that request is continuously re-transmitted at a fixed interval configured by MTPFC.PFRT. When the number of re-transmissions for one of priority-groups reaches the value of MTPFC.PFRLV, this IP sets MEIS.PFRROS to 1 for an interrupt. Even when the interrupt is detected, this IP continues re-transmission of a PFC frame while the transmit request is asserted.

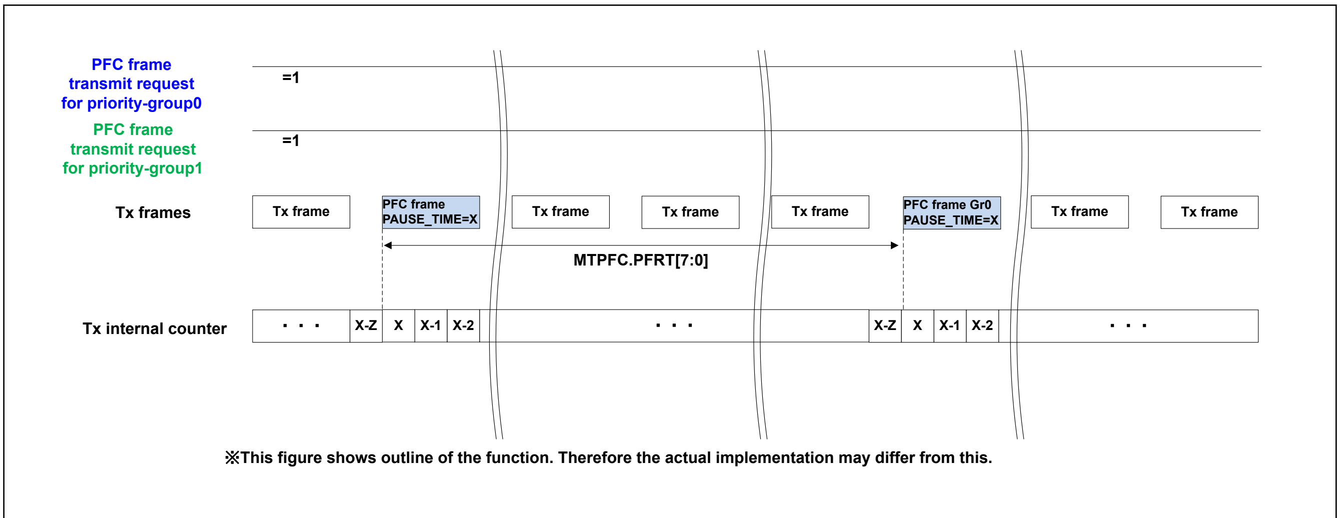


Figure 33.20 Transmitting PFC frame (part 2)

After deasserting the transmission request, this IP stops the transmission of the PFC frame corresponding to that request. At that time, if the corresponding bit of MTPFC2.PFCTTZn for deasserted priority-group request is 1 and an internal counter to count the pausing-period exceeds MTPFC.PFRT (*), this IP transmits one more PFC frame with all “PAUSE_TIME” fields as 0 for that priority-group.

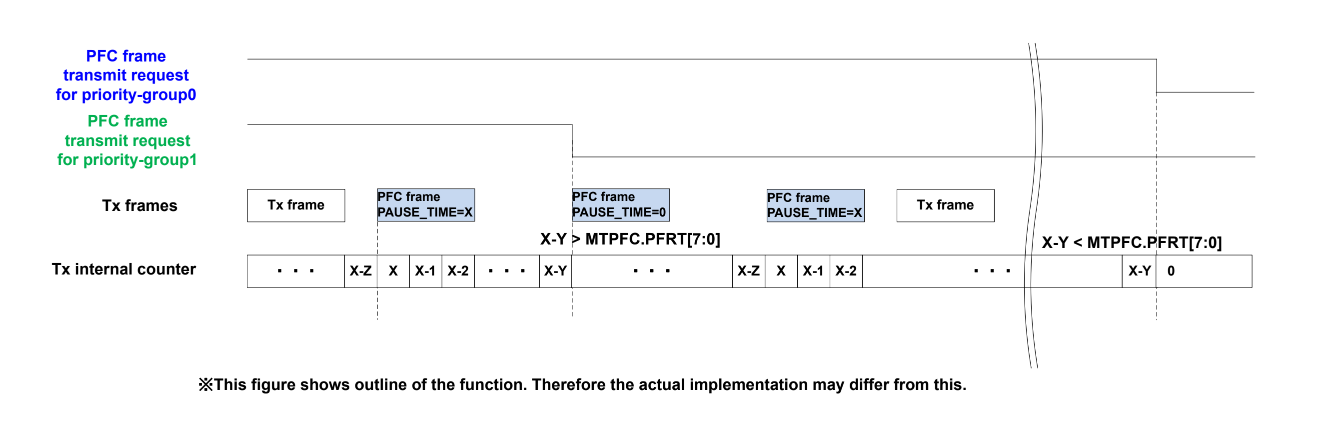


Figure 33.21 Transmitting PFC frame (part 3)

33.6.2.2.2 Reception of PFC Frame

This IP treats several priorities specified in received PFC frames. The number of supported priorities is 8.

1. When this IP receives a PFC frame with a priority-enable field which is corresponding to a priority configured by MRGC.PFCRCn, it starts internally counting a pausing-period for each priority respectively specified by each “PAUSE_TIME” fields of a received PFC frame. During counting the pausing-period, notification of PFC frame reception to an upper layer (e.g. MHD) and MRPFM.PFCTCAn corresponding to each priority are asserted. The bit number is corresponding to priority number.
2. When PFC frame with “PAUSE_TIME ≠ 0” with corresponding to valid “priority_enable” field is received again before expiring the internal pausing-period, the internal counter for that priority is updated to that value.
3. When MRGC.PFRTZ = 1 and PFC frame with “PAUSE_TIME = 0” with corresponding to valid “priority_enable” field is received before expiring the internal pause-period, the internal counter for that priority is cleared and notification of pause frame reception to an upper layer (e.g. MHD) and MRPFM.PFCTCAn for that priority is deasserted.
4. When the internal pausing-period is expired, notification of pause frame reception to an upper layer (e.g. MHD) and MRPFM.PFCTCAn for that priority are deasserted.
   In any case, MPCFRCT0 to MPCFRCT7 are incremented respectively after receiving a PFC frame for enabled priority by MRGC.PFCRCn. And all received PFC frames are discarded internally.

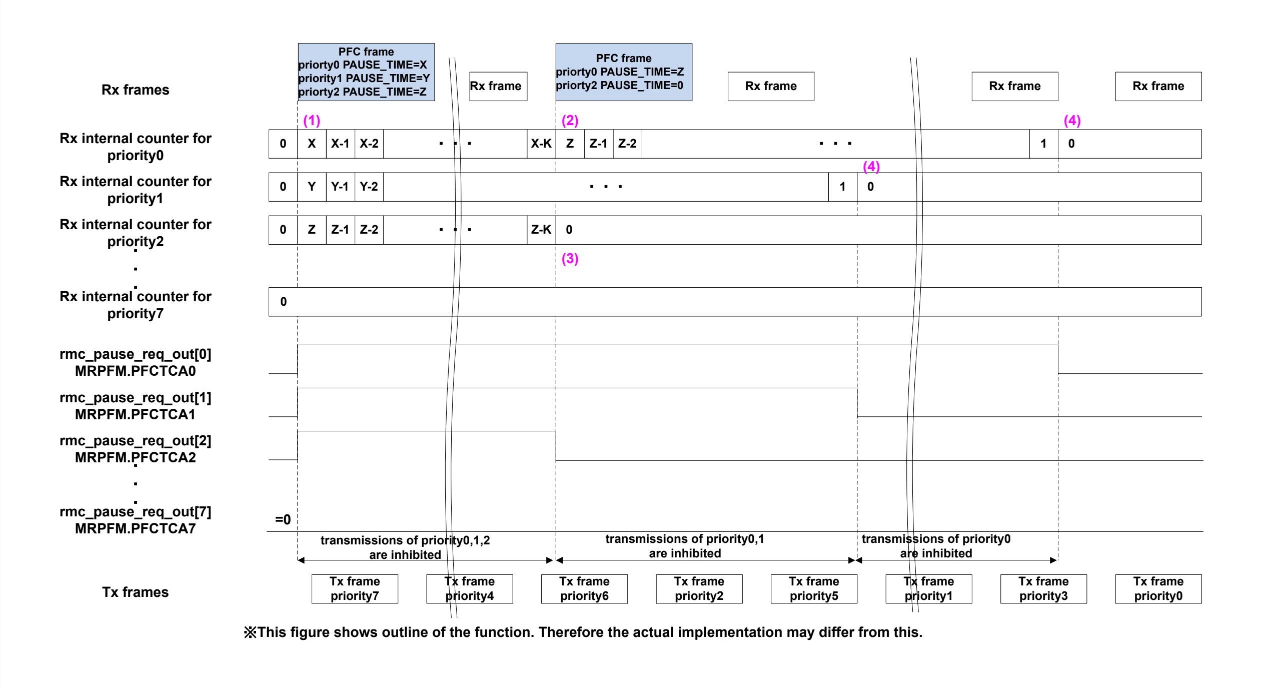


Figure 33.22 Reception of PAUSE frame

33.6.3 Timestamping

This IP supports time synchronization for IEEE 1588. It supports time synchronization accuracy in the sub-microsecond range.

33.6.3.1 Timestamp Capture

This IP can capture timestamps on both TX and RX side. gPTP timer value includes two 62bit timers, bit[61:0] is a Timer0, bit[123:62] is a Timer1.

[Timestamp capture on TX side]

“Timestamp capture” and “Timer number” fields in [GWCA] control this function for both e-frame and p-frame. These are configured by a Tx Descriptor. If “Timestamp capture” =1, the timer specified by “Timer number” field is captured.

The captured timestamp is sent to an upper module(e.g. MHD) with “Timestamp unique number” in [GWCA] via MHD Tx timestamp capture Interface.

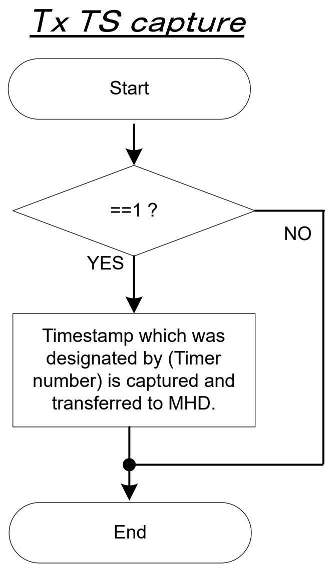


Figure 33.23 Tx Timestamp capture flow

Note: On the TX Side, the timestamp capture function by this IP and PHY can be enabled at same time.

[Timestamp capture on RX side]

MTRC.TRDDE, MTRC.TRDDP, and MTRC.TRHFMEn control this function.

The configuration of the timestamp capture is following.

- MTRC.TRDDE controls this function for e-frames.
  - MTRC.TRDDE = 0
    The default timer configured by MTRC.DTN is captured for a received e-frame.
- MTRC.TRDDP controls this function for p-frames.
  - MTRC.TRDDP = 0
    the default timer configured by MTRC.DTN is captured for a received p-frame.
- MTRC.TRHFMEn controls this function for e-frames using the HW PTP message filtering.
  The bit[0] is for Timer0 and bit[1] is for Timer1.
  See Figure 33.26 to use the HW PTP message filter.
  - TRHFMEn = x1 and the received e-frame matched the HW PTP messaged filter for Timer0
    The Timer0 is captured for that received e-frames.
  - TRHFMEn = 10 and the received e-frame matched the HW PTP messaged filter for Timer1
    The Timer1 is captured for that received e-frames.
  - TRHFMEn = 11 and the received e-frame matches the HW PTP messaged filter for Timer0 and 1
    The Timer0 is captured for that received e-frames.

A captured timestamp is outputted from this IP with the received frame.

Figure 33.25 shows Rx Timestamp capture flow.

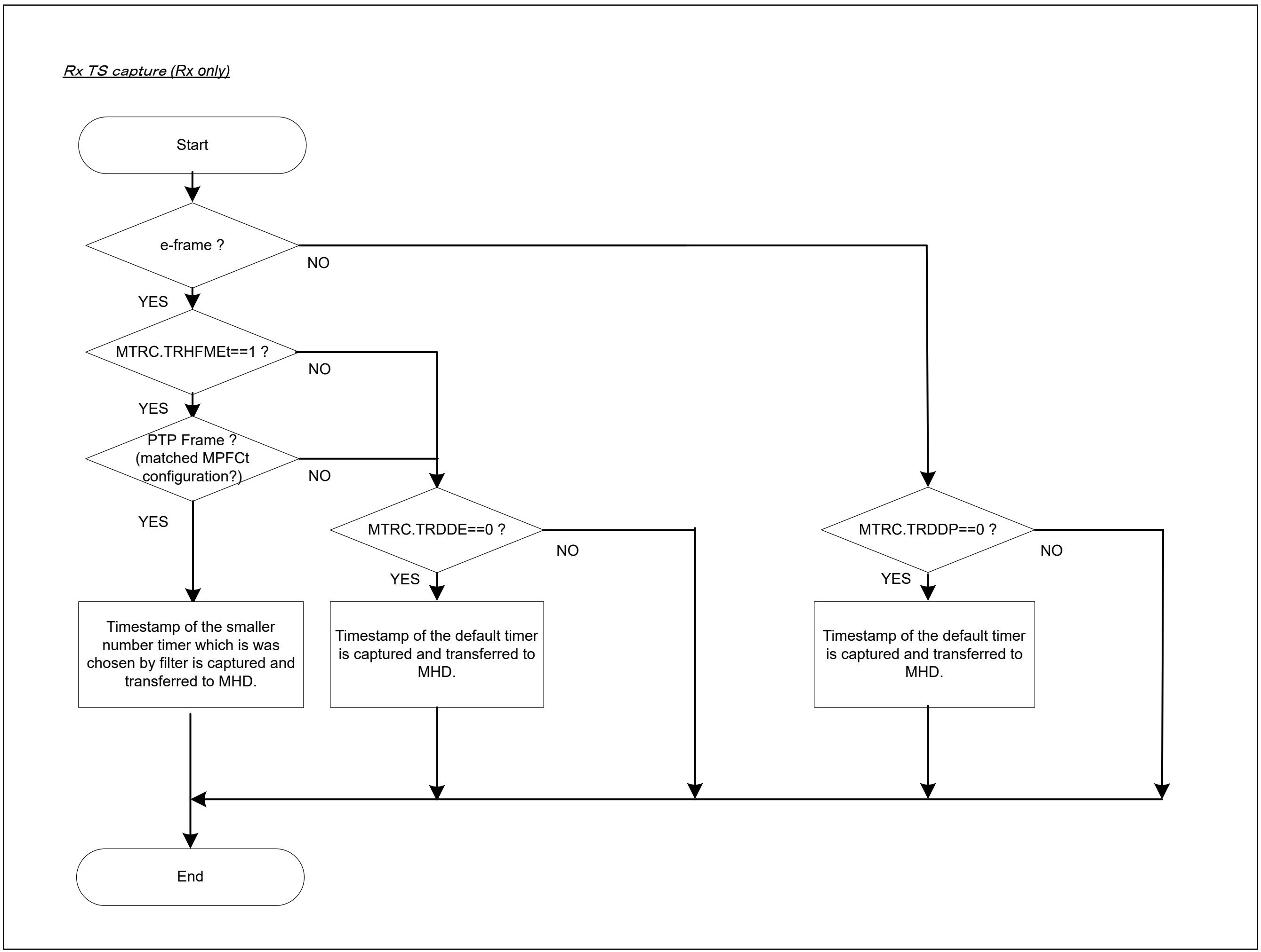


Figure 33.24 Rx TS capture (Rx only)

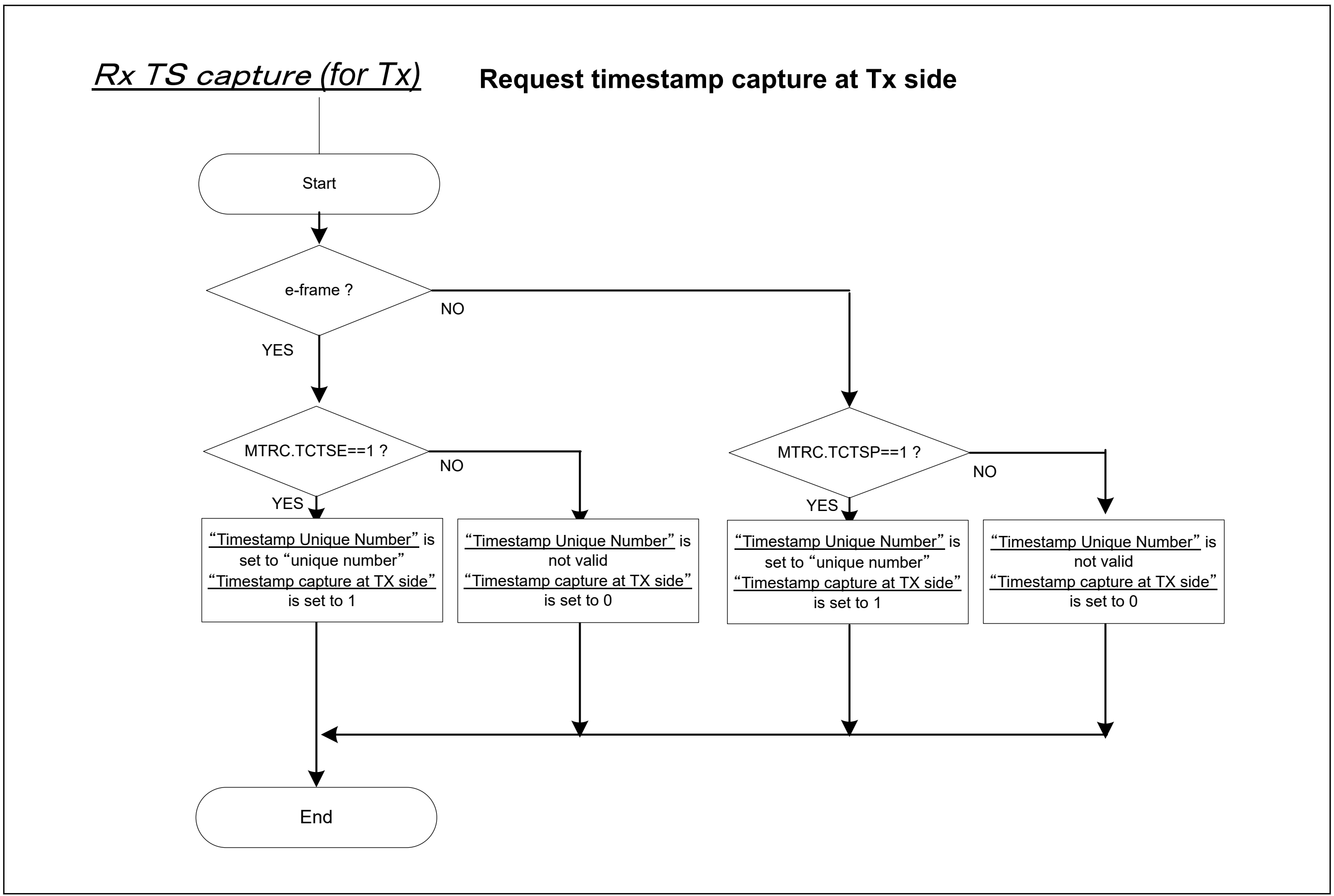


Figure 33.25 Rx timestamp capture flow

[Configuration of HW PTP Filter]

HW PTP message filter is configured by using MPFC0-15. The Figure 33.26 shows an example configuration of PTP Message Filter. This is a case of MTRC.TRHFMEn = 11.

In this case:

MPFC0-7 are used as the filters for both Timer0 and Timer1. Because TEF field is set to 0x03.

MPFC8 is used as the filter for Timer0. Because TEF field is set to 0x01.

MPFC9 is used as the filter for Timer1. Because TEF field is set to 0x02.

MPFC10-15 are not used. Because TEF field is set to 0x00.

If an e-frame matches all MPFC0-7 and MPFC8, it is judged as a frame of Timer0 domain.

If an e-frame matches all MPFC0-7 and MPFC9, it is judged as a frame of Timer1 domain.

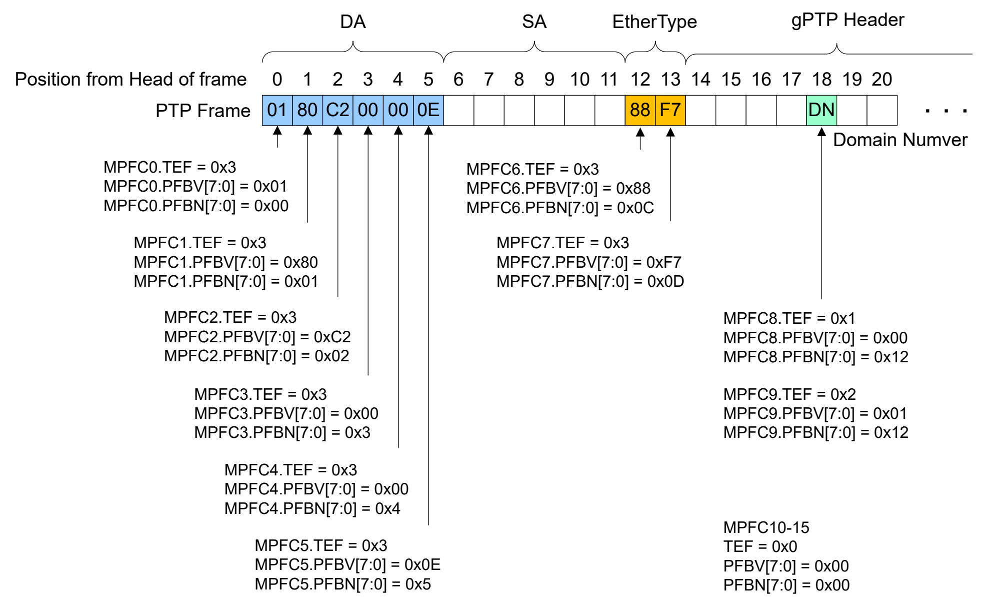


Figure 33.26 Example configuration of PTP message filter

Note:
- Depending on the configuration of the filter, a frame may match both filters. In this case, Timer0 is selected.
- PFBN should not select a position that exceeds the end of frame. Therefore, this value less than or equal to the minimum frame size is recommended.

33.6.4 Station Management

The Station Management controls the access to Management registers in PHY.

rmc_mdi, rmc_mdo, rmc_mde, rmc_mdc and rmc_mdio_int are used for this function.

33.6.4.1 Management Register Access

This IP has two types of access methods, one is specified by clause 22 the other is specified by clause 45 in IEEE802.3.

33.6.4.1.1 Access Method Specified in Clause 22

To write or read of Management register of PHY which supports this access method using management frame format, execute the following steps.

[write access]

Set MPSM as following.

PRD[15:0]="Write Data"

POP[1:0]=01

PRA[4:0]="Register Address"

PDA[4:0]="PHY Address"

MFF=0

PSME=1

After setting MPSM.PSME is kept to 1 during the access. Wait an interrupt is detected (MMIS1.PWACS=1) or MPSM.PSME turns to 0 as a sign of an access completion.

At this time, when MMIE1.PWACE is set to 1, the interrupt appears on rmc_mdio_int.

Set MMIS1.PWACS to 0 to clear the interrupt.

[Read access]

Set MPSM as following.

PRD[15:0]="0x00"

POP[1:0]=10

PRA[4:0]="Register Address"

PDA[4:0]="PHY Address"

MFF=0

PSME=1

After setting MPSM.PSME is kept to 1 during the access. Wait an interrupt is detected (MMIS1.PRACS=1) or MPSM.PSME turns to 0 as a sign of an access completion.

At this time, when MMIE1.PRACE is set to 1, the interrupt appears on rmc_mdio_int.

After the completion of read access, the read data is stored in MPSM.PRD[15:0]. So read it.

Set MMIS1.PRACS to 0 to clear the interrupt.

Note: When MPSM.PSME is set to 1, this IP outputs preamble code of fixed length specified in standard on rmc_mdo before write or read access. IEEE802.3 clause 22 allows skipping preamble code. When the preamble code is not needed and PHY also supports to skip the preamble code, this IP does not output the preamble code by setting MPIC.PSMDP to 1.

33.6.4.1.2 Access Method Specified in Clause 45

To write or read Management register of PHY which supports this access method using extended management frame format, execute the following steps.

[write access]

1. Set MPSM as following. ("Register Address" is set to a PHY)
   PRD[15:0] = "Register Address"
   POP[1:0] = 00
   PRA[4:0] = "Device Address"
   PDA[4:0] = "Port Address"
   MFF = 1
   PSME = 1
   After setting MPSM.PSME is kept to 1 during the access. Wait an interrupt is detected (MMIS1.PAACS = 1) or MPSM.PSME turns to 0 as a sign of an access completion.
   At this time, when MMIE1.PAACE is set to 1, the interrupt appears on rmc_mdio_int.
   Set MMIS1.PWACS to 0 to clear interrupt.
2. Set MPSM as following. ("Write Data" is set to the "Register Address" which was configured by step 1.)
   PRD[15:0] = "Write Data"
   POP[1:0] = 01
   PRA[4:0] = "Device Address"
   PDA[4:0] = "Port Address"
   MFF = 1
   PSME = 1
   After setting MPSM.PSME is kept to 1 during the access. Wait an interrupt is detected (MMIS1.PWACS = 1) or MPSM.PSME turns to 0 as a sign of an access completion.
   At this time, if MMIE1.PWACE is set to 1, the interrupt appears on rmc_mdio_int.
   Set MMIS1.PWACS to 0 to clear interrupt.

[Read access]

1. Set MPSM as following. ("Register Address" is set to a PHY)
   PRD[15:0] = "Register Address"
   POP[1:0] = 00
   PRA[4:0] = "Device Address "

PDA[4:0] = “Port Address”
MFF = 1
PSME = 1
After setting MPSM.PSME is kept to 1 during the access. Wait an interrupt is detected (MMIS1.PAACS = 1) or MPSM.PSME turns to 0 as a sign of an access completion.
At this time, when MMIE1.PAACE is set to 1, the interrupt appears on rmc_mdio_int.
Set MMIS1.PWACS to 0 to clear interrupt.

2. Set MPSM as following. (Read access of the Register Address which was pointed by previous step)
PRD[15:0] = 0x00
POP[1:0] = 11
PRA[4:0] = “Device Address”
PDA[4:0] = “Port Address”
MFF = 1
PSME = 1
After setting MPSM.PSME is kept to 1 during the access. Wait an interrupt is detected (MMIS1.PRACS = 1) or MPSM.PSME turns to 0 as a sign of an access completion.
At this time, when MMIE1.PRACE is set to 1, the interrupt appears on rmc_mdio_int.
After the completion of read access, the read data is stored in MPSM.PRD[15:0]. So read it.
Set MMIS1.PRACS to 0 to clear interrupt.

[Post read incremental access]

Read access needs setting “Register Address” every time. Only when continuous Register Address fields are read continuously, using this access (POP[1:0] = 10) is efficient.

1. Set MPSM as following. (“Register Address” is set to a PHY)
PRD[15:0] = “Register Address”
POP[1:0] = 00
PRA[4:0] = “Device Address ”
PDA[4:0] = “Port Address”
MFF = 1
PSME = 1
After setting MPSM.PSME is kept to 1 during the access. Wait an interrupt is detected (MMIS1.PAACS = 1) or MPSM.PSME turns to 0 as a sign of an access completion.
At this time, when MMIE1.PAACE is set to 1, the interrupt appears on rmc_mdio_int.
Set MMIS1.PWACS to 0 to clear the interrupt.

2. Set MPSM as following. (Read access of the Register Address which was pointed by previous step)
PRD[15:0] = 0x00
POP[1:0] = 10
PRA[4:0] = “Device Address ”
PDA[4:0] = “Port Address”
MFF = 1
PSME = 1
After setting MPSM.PSME is kept to 1 during the access. Wait an interrupt is detected (MMIS1.PPRACS = 1) or MPSM.PSME turns to 0 as a sign of an access completion.
At this time, when MMIE1.PPRACE is set to 1, the interrupt appears on rmc_mdio_int.
After the completion of read access, the read data is stored in MPSM.PRD[15:0]. So read it. And “Register Address” set by step 1 is automatically changed in the PHY to point a next register address field. So don’t need to re-configuration of “Register Address” for next read access.
Set MMIS1.PPRACS to 0 to clear the interrupt.

3. Repeat step 2 for the read access from the Next Register Address field.

Note: When MPSM.PSME set to 1, this IP outputs preamble code of fixed length specified in standard on rmc_mdo before write or read access. IEEE802.3 clause 45 does not allow skipping the preamble code. But skipping preamble code is also supported in this access method by setting MPIC.PSMDP to 1. To use this function, check a PHY specification.

33.6.4.2 Management Data Clock (MDC) Setting

To access Management registers in PHY, this IP outputs MDC. This is generated by dividing CHI clock (clk) in this IP and outputted from rmc_mdc. The division ratio can be set by the MPIC.PSMCS.

The following table shows the programmed MDC frequency depending on MPIC.PSMCS. MDC frequency is calculated by clk[MHz] / ((MPIC.PSMCS +1)*2).

Table 33.7 Programming example for MPIC.PSMCS

clk frequency	MPIC.PSMCS	MDC frequency
100MHz	0x13	2.5MHz
200MHz	0x27	2.5MHz

Note: IEEE 802.3 specifies as the clock should be slower than 2.5MHz. But some PHY could operate with a higher frequency. See the PHY specification for detail.

33.6.4.3 Management Data Timing Adjustment

This IP can adjust the drive timing of MDO (rmc_mdo and rmc_mde) and the capture timing of MDI (rmc_mdi). It may be used for the propagation delay on the PCB.

The drive timing could be adjusted by PHY Station Management Hold Time Adjustment (MPIC.PSMHT). The capture timing could be adjusted by PHY Station Management Capture Time Adjustment (MPIC.PSMCT). The figure below is an example shifted by 2clk cycles.

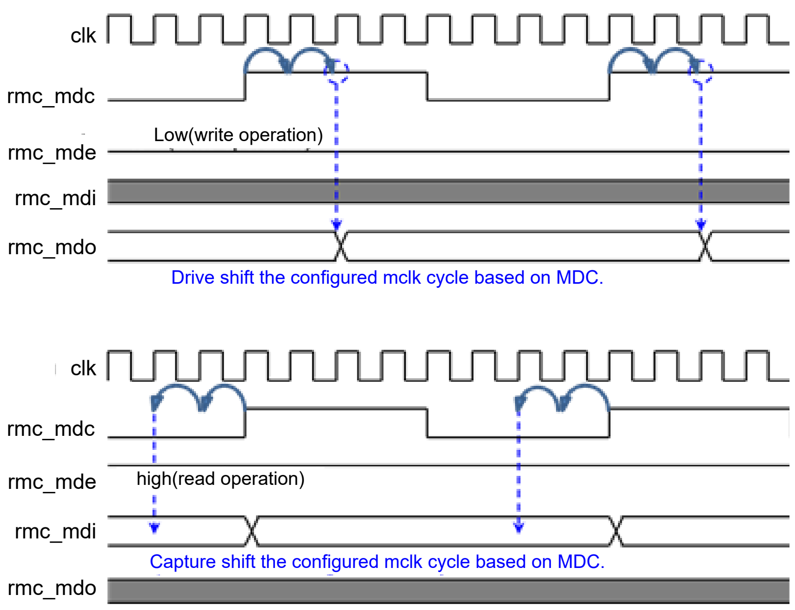


Figure 33.27 Management data timing adjustment

33.6.5 Energy Efficient Ethernet

This IP supports Energy Efficient Ethernet (EEE) which is standardized in IEEE 802.3. Tx-LPI mode and Rx-LPI mode can be activated independently.

33.6.5.1 LPI Control

This IP can request LPI mode for Tx port to the PHY when the LPI request bit (MEEEC.LPITR) is asserted. LPI mode is released when the bit is negated. The operation flow is shown in section 33.5.2.5. LPI Transition for Tx Port in detail.

The LPI mode for Rx port can be recognized by the reception LPI interrupt status bit (MMIS2.LPIAIS). When the bit (MMIS2.LPIDIS) is asserted, it means RMAC detected deassertion of LPI.

Table 33.8 and Table 33.9 show LPI specification on each xMII interface. The specification for MII or GMII is specified by IEEE802.3.

Table 33.8 Rx LPI specification on each xMII interface

PHY IF	RX_DV	RX_ER	RXD
MII	rmc_rx_valid_gmii = 0	rmc_rx_err_gmii = 1	rmc_rx_data_gmii[3:0] = 0001
GMII	rmc_rx_valid_gmii= 0	rmc_rx_err_gmii = 1	rmc_rx_data_gmii[7:0] = 00000001

Table 33.9 Tx LPI specification on each xMII interface

PHY IF	TX_EN	TX_ER	TXD
MII	rmc_tx_valid_gmii = 0	rmc_tx_err_gmii = 1	rmc_tx_data_gmii [3:0] = 0001
GMII	rmc_tx_valid_gmii = 0	rmc_tx_err_gmii = 1	rmc_tx_data_gmii [7:0] = 00000001

33.6.6 MAC Address Filtering on Reception

This IP supports the frame reception in a full-duplex operation mode.

This IP can pass all received Ethernet frames data to an upper layer via xMII Interface.

This IP strips off automatically SFDs/SMDs, preambles, FRAG_COUNTs and mCRCs from all the frames.

The FCS reception by the upper layer can be set though MRGC.RCPT register.

This IP identifies the frames with the special format and informs to the upper layer. It is indicated with the dedicated flag in status information followed payload data. The table below indicates the frames with the special format.

Table 33.10 Operation for special format frame

Frame	Format	Operation
Broadcast frame	DA: FF_FF_FF_FF_FF_FF	Filtering possible.
Multicast frame	DA: B'xxxx_xxx1_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx_xxxx	Filtering possible.
Pause frame	DA: 01_80_C2_00_00_01 TYPE: 88_08 OPCODE: 00_01	This IP handles Flow Control, refer to section 8.2. Pause information is not sent to upper layer.
PFC frame	DA: 01_80_C2_00_00_01 TYPE: 88_08 OPCODE: 01_01	This IP handles Flow Control, refer to section 8.2. Pause information is not sent to upper layer.
Magic packet	When the MAC address (MRMAC.MA) is 00-11-22-33-44-55h, this IP would scan for the data sequence: Destination-Address Source-Address ....MISC....... FF FF FF FF FF FF 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 ....MISC...... FCS This sequence can be located anywhere in the Magic Packet payload. The Magic Packet is formed with a standard Ethernet header and FCS.	Interrupt. Provide information to upper layer. Provide magic packet interrupt (MMIS2.MPDIS). Magic packets are only detected as E-frame.

This IP supports the MAC address filtering for Unicast, Multicast, Broadcast, Broadcast storm (continuous Broadcast frame reception) and Multicast storm (continuous Multicast frame reception). It uses the destination address (DA) and source Address(SA) of a received frame for the address filtering. When the filter condition is satisfied, the received frames are transferred to an upper layer. If not satisfied, the received frames are transferred to the upper layer with the error flag of “Frame filtered” in [ETHA].

The filter condition is configured in Reception filter configuration register (MRAFC). It can be individually configured for Unicast address, Multicast address, and Broadcast address.

And this IP also supports Promiscuous mode.

When MRAFC.UCEN*, MCEN*, BCEN*, NDARE*, SDSFRE*, NSARE* and MSARE* are set to 1, all frames are transferred to the upper layer without error flag of “Frame filtered”.

Note : Regardless of MRAFC configuration, Verify and Respond are not transferred to the upper layer.

The flow of Figure 33.28 shows the MAC Address filtering.

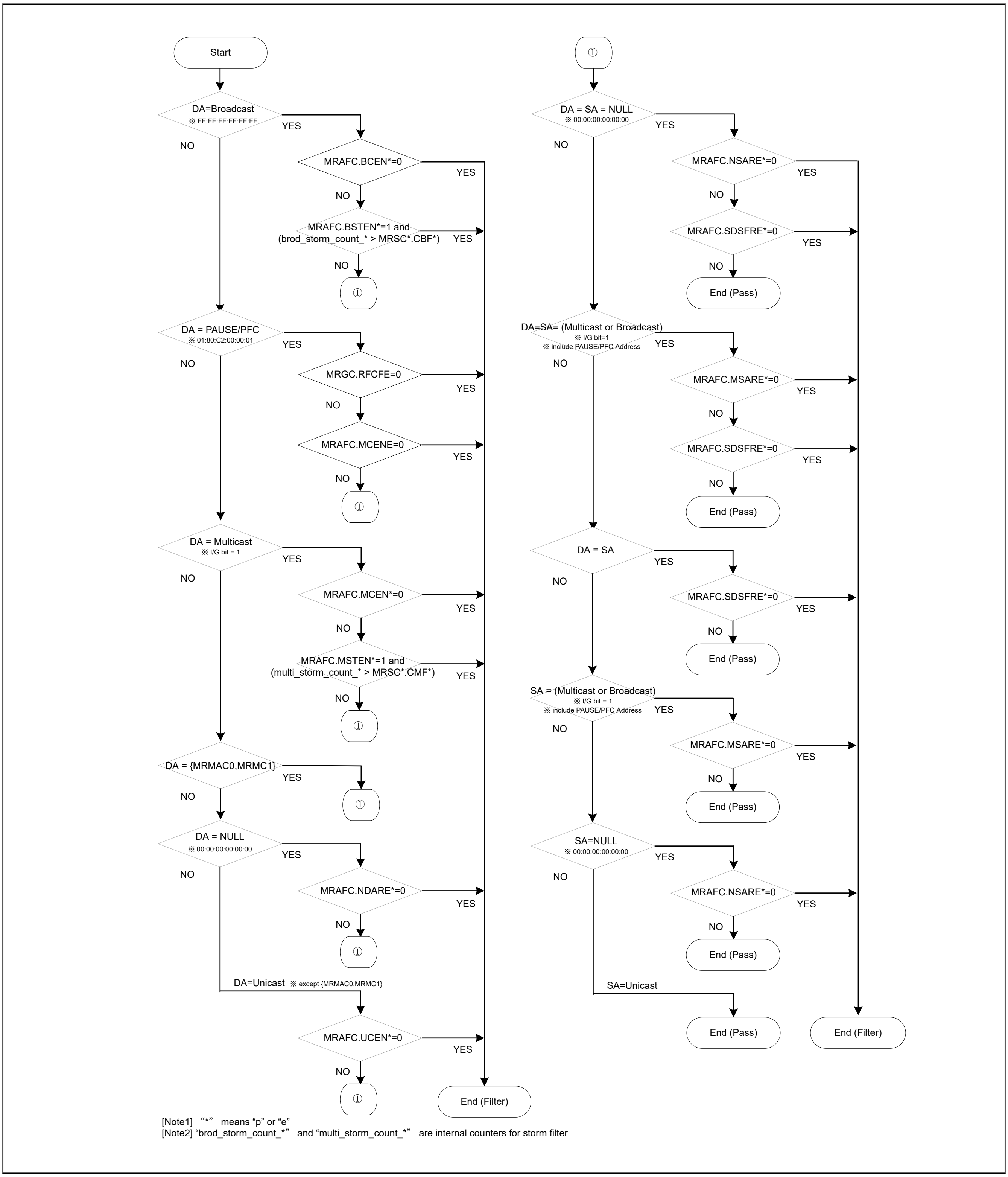


Figure 33.28 MAC address filtering

34. Ethernet CPU Agent (GWCA)

34.1 Overview

Ethernet CPU Agent (GWCA) is a CPU interface that consists of an agent interface module allowing communication within ESWM. GWCA handles data exchange between ESWM and the GWCPU subsystem. Software is required for the configuration of Gateway AXI bus interface.

Table 34.1 lists the GWCA specifications and Figure 34.1 shows a block diagram.

.

Table 34.1 GWCA Specifications

Function		Description
Interfaces	Switch mode	Interface to receive the switch mode [MFWD]
	Pause interface	Interface to pause TX queues [COMA]
	AXI Master interface	Master interface conforming to AXI 3 AMBA protocol. It is used for data transfer between GWCPU and GWCA. ● Data bit width: 128 bits ● Number of outstanding read: 8 ● Number of outstanding write: 8 ● Number of ID read: 8 (from 0 to 7) ● Number of ID write: 8 (from 0 to 8-1) ● Incremental Access only ● Supports unaligned transactions ● Supports out of order ● Supports interleave on read bus
	SFR interface	Interface to access GWCA SFRs [COMA]
	Interrupts	GWCA interrupt to CPU
	Fabric interface	Interface to communicate with the data, TAG and pointer RAM [MFAB]
	Descriptor bus	Interface to receive frame information to send them to CPU [MFWD]
	L2/L3 update bus	Interface to fetch the frame routing information [MFWD]
	BP Request	Interface to receive BPs to store frames in the data, TAG and pointer RAM [COMA]
	BP Release	Interface to release the BPs that has been used [COMA]
	TX Timestamp Capture interface	Interface to receive TX timestamps and send it to CPU [RMAC]
	RAM interfaces	Interface to communication with RAMs [TOP]
Data provision	Ethernet Frames	Corresponds to the IEEE 802.3, 802.1Q and 802.1CB standards [802.3] [802.1Q] [802.1CB]
	Descriptors	AXI TX Descriptor (CPU to GWCA) AXI RX Descriptor (GWCA to CPU) Local Ethernet Descriptor, Local Direct Descriptor (GWCA to Forwarding engine) Forwarding descriptor (Forwarding engine to GWCA)
	Timestamp	TX timestamp for transmission to CPU RAM [RMAC]

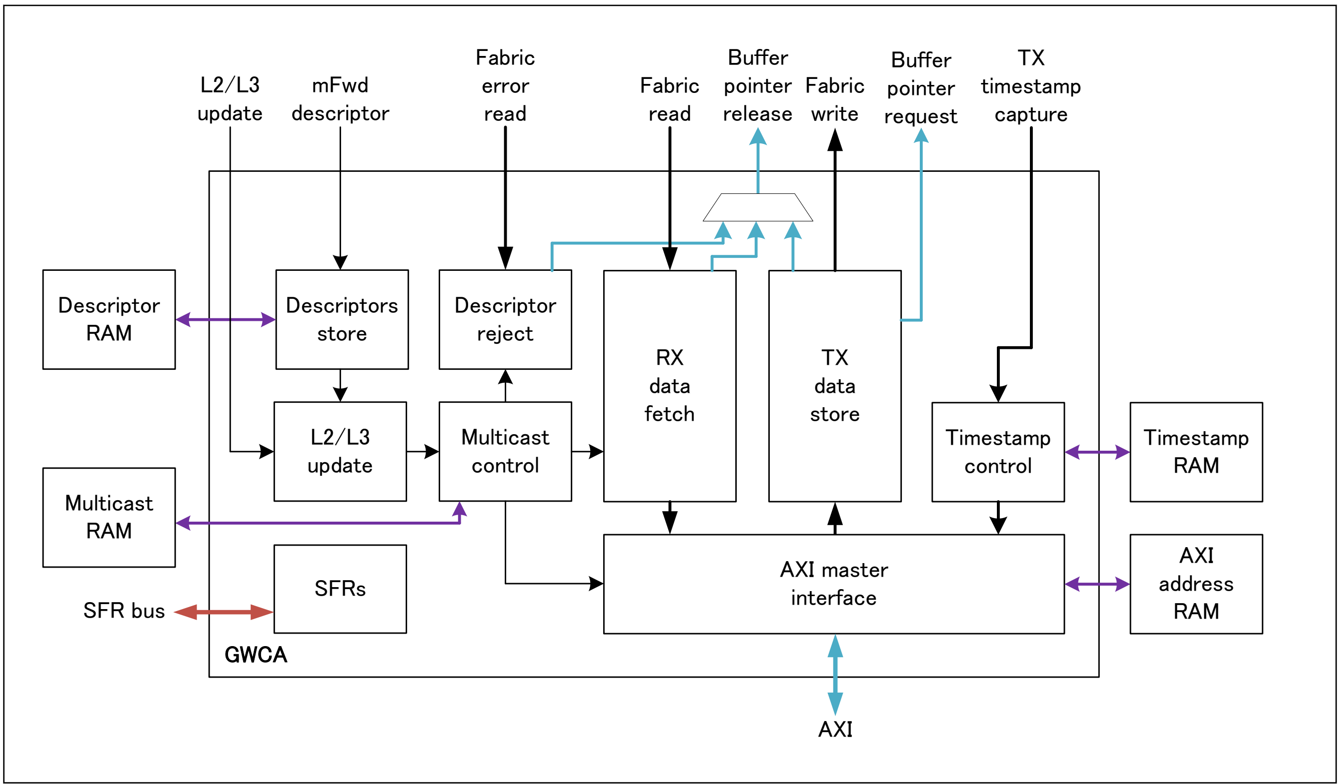


Figure 34.1 GWCA block diagram

Table 34.2 GWCA functional blocks

Block name	Function
TX data store	Transmits frames from AXI master to the local RAM.
RX data fetch	Transmits frame from local RAM to AXI master.
Descriptor store	Receives the descriptors from forwarding engine and controls the priority between descriptors.
L2/L3 update	Fetches L2/L3 update information.
Descriptor reject	Reject descriptors.
Multicast control	Controls multicast transactions.
Timestamp control	Receives timestamps for Ethernet Agents, saves them and transmits them to the AXI master.
AXI master interface	AXI master for data exchange with the CPU
SFRs	GWCA SFRs

34.2 GWCA Register List

Table 34.3 GWCA registers (1 of 4)

Offset address	Register symbol	Register name	Write access mode
0x0000	GWMC	Mode Configuration Register	Any
0x0004	GWMS	Mode Status Register	—
0x0010	GWIRC	IPV Remapping Configuration Register [802.1Q]	C
0x0014	GWRDQSC	RX Descriptor Queue Security Configuration Register	C, O
0x0018	GWRDQC	RX Descriptor Queue Control Register	O
0x001C	GWRDQAC	RX Descriptor Queue Arbitration Control Register	C
0x0020	GWRGC	RX General Configuration Register	C

Table 34.3 GWCA registers (2 of 4)

Offset address	Register symbol	Register name	Write access mode
0x0040 + 0x04 × q	GWRMFSCq	Reception Maximum Frame Size Configuration Register q (q = 0 to 7)	C, O
0x0060 + 0x04 × q	GWRDQDCq	Reception Descriptor Queue q Depth Configuration Register (q = 0 to 7)	C
0x0080 + 0x04 × q	GWRDQMq	RX Descriptor Queue q Monitoring Register (q = 0 to 7)	—
0x00A0 + 0x04 × q E: 0x00C0 + 0x04 × q	GWRDQMLMq	RX Descriptor Queue q Max Level Monitoring Register (q = 0 to 7)	—
0x0100	GWMTIRM	Multicast Table Initialization Register Monitoring Register	C, O
0x0104	GWMSTLS	Multicast Table Learning Setting Register	C, O
0x0108	GWMSTLR	Multicast Table Learning Result Register	—
0x010C	GWMSTSS	Multicast Table Searching Setting Register	C, O
0x0110	GWMSTSR	Multicast Table Searching Result Register	—
0x0120	GWMAC0	MAC Address Configuration Register 0	C
0x0124	GWMAC1	MAC Address Configuration Register 1	C
0x0130	GWVCC	VLAN Control Configuration Register	C
0x0134	GWVTC	VLAN TAG Configuration Register	C
0x0138	GWTTFC	Transmission TAG Filtering Configuration Register	C
0x0140 + 0x08 × s	GWTDCACs0	Timestamp Descriptor Chain s Address Configuration Register 0 (s = 0, 1)	C, O
0x0144 + 0x08 × s	GWTDCACs1	Timestamp Descriptor Chain s Address Configuration Register 1 (s = 0, 1)	C, O
0x0160 + 0x04 × s	GWTSDCCs	Timestamp Descriptor Chain s Configuration Register (s = 0, 1)	C
0x0180	GWTSNM	Timestamp Number Monitoring Register	C, O
0x0184 E: 0x0188	GWTSMNM	Timestamp Maximum Number Monitoring Register	C, O
0x0190	GWAC	AXI Control Register	O
0x0194	GWDCBAC0	Descriptor Chain Base Address Configuration Register 0	C
0x0198	GWDCBAC1	Descriptor Chain Base Address Configuration Register 1	C
0x01A0	GWMDNC	Maximum Descriptor Number Configuration Register	C
0x0200 + 0x4 × i	GWTRCi	Transmission Request Configuration Register i (i = 0, 1)	O
0x0300 + 0x4 × p	GWTPCp	Transmission Pause Configuration Register p (p = 0, 1)	C
0x0380	GWARIRM	AXI RAM Initialization Register Monitoring Register	C, O
0x0400 + 0x4 × i	GWDCCi	Descriptor Chain i Configuration Register (i = 0 to 63)	C, O
0x0800	GWAARSS	AXI Address RAM Searching Setting Register	C, O
0x0804	GWAARSR0	AXI Address RAM Searching Result Register 0	—
0x0808	GWAARSR1	AXI Address RAM Searching Result Register 1	—
0x0840 + 0x4 × i	GWIDAUASi	Incremental Data Area i Used Area Size Register (i = 0 to 3)	C, O, D
0x0880 + 0x4 × i	GWIDASMi	Incremental Data Area i Size Monitoring Register (i = 0 to 3)	—
0x0900 + 0x8 × i	GWIDASAMi0	Incremental Data Area i Start Address Monitoring Register 0 (i = 0 to 3)	—

Table 34.3 GWCA registers (3 of 4)

Offset address	Register symbol	Register name	Write access mode
0x0904 + 0x8 × i	GWIDASAMi1	Incremental Data Area i Start Address Monitoring Register 1 (i = 0 to 3)	—
0x0980 + 0x8 × i	GWIDACAMi0	Incremental Data Area i Current Address Monitoring Register 0 (i = 0 to 3)	—
0x0984 + 0x8 × i	GWIDACAMi1	Incremental Data Area i Current Address Monitoring Register 1 (i = 0 to 3)	—
0x0A00	GWGRLC	Global Rate Limiter Configuration Register	C, O
0x0A04	GWGRLULC	Global Rate Limiter Upper Limit Configuration Register	C, O
0x0A80 + 0x8 × i	GWRLCi	Rate Limiter i Configuration Register (i = 0 to 7)	C, O
0x0A84 + 0x8 × i	GWRLULCi	Rate Limiter i Upper Limit Configuration Register (i = 0 to 7)	C, O
0x0B80	GWIDPC	Interrupt Delay Prescaler Configuration Register	C
0x0C00 + 0x4 × i	GWIDCi	Interrupt Delay Configuration Register i (i = 0 to 63)	C, O
0x1000 E: 0x1080	GWRDCN	Received Data Counter Register	—
0x1004 E: 0x1084	GWTDCN	Transmitted Data Counter Register	—
0x1008 E: 0x1088	GWTSCN	Timestamp Counter Register	—
0x100C E: 0x108C	GWTSOVFECN	Timestamp Overflow Error Counter Register	—
0x1010 E: 0x1090	GWUSMFSECN	Under Minimum Frame Size Error Counter Register	—
0x1014 E: 0x1094	GWTFECN	TAG Filtering Error Counter Register	—
0x1018 E: 0x1098	GWSEQECN	Sequence Error Counter Register	—
0x1020 E: 0x10A0	GWTXDNECN	TX Descriptor Number Error Counter Register	—
0x1024 E: 0x10A4	GWFSECN	Frame Size Error Counter Register	—
0x1028 E: 0x10A8	GWTDFECN	Timestamp Descriptor Full Error Counter Register	—
0x102C E: 0x10AC	GWTSDNECN	Timestamp Descriptor Number Error Counter Register	—
0x1030 E: 0x10B0	GWDQOECN	Descriptor Queue Overflow Error Counter Register	—
0x1034 E: 0x10B4	GWDQSECN	Descriptor Queue Security Error Counter Register	—
0x1038 E: 0x10B8	GWDFECN	Descriptor Full Error Counter Register	—
0x103C E: 0x10BC	GWDSECN	Descriptor Security Error Counter Register	—
0x1040 E: 0x10C0	GWDSZECN	Data Size Error Counter Register	—
0x1044 E: 0x10C4	GWDCTECN	Descriptor Chain Type Error Counter Register	—
0x1048 E: 0x10C8	GWRXDNECN	RX Descriptor Number Error Counter Register	—
0x1100 + 0x10 × i	GWDISi	Data Interrupt Status Register i (i = 0, 1)	C, O, D

Table 34.3 GWCA registers (4 of 4)

Offset address	Register symbol	Register name	Write access mode
0x1104 + 0x10 × i	GWDIEi	Data Interrupt Enable Register i (i = 0, 1)	C, O, D
0x1108 + 0x10 × i	GWDIDi	Data Interrupt Disable Register i (i = 0, 1)	C, O, D
0x110C + 0x10 × i	GWDIDSi	Data Interrupt Delayed Status Register i (i = 0, 1)	—
0x1180	GWTSDIS	Timestamp Data Interrupt Status Register	C, O, D
0x1184	GWTSDIE	Timestamp Data Interrupt Enable Register	C, O, D
0x1188	GWTSDID	Timestamp Data Interrupt Disable Register	C, O, D
0x1190	GWEIS0	Error Interrupt Status Register 0	C, O, D
0x1194	GWEIE0	Error Interrupt Enable Register 0	C, O, D
0x1198	GWEID0	Error Interrupt Disable Register 0	C, O, D
0x11A0	GWEIS1	Error Interrupt Status Register 1	C, O, D
0x11A4	GWEIE1	Error Interrupt Enable Register 1	C, O, D
0x11A8	GWEID1	Error Interrupt Disable Register 1	C, O, D
0x1200 + 0x10 × i	GWEIS2i	Error Interrupt Status Register 2i (i = 0, 1)	C, O, D
0x1204 + 0x10 × i	GWEIE2i	Error Interrupt Enable Register 2i (i = 0, 1)	C, O, D
0x1208 + 0x10 × i	GWEID2i	Error Interrupt Disable Register 2i (i = 0, 1)	C, O, D
0x1280	GWEIS3	Error Interrupt Status Register 3	C, O, D
0x1284	GWEIE3	Error Interrupt Enable Register 3	C, O, D
0x1288	GWEID3	Error Interrupt Disable Register 3	C, O, D
0x1290	GWEIS4	Error Interrupt Status Register 4	C, O, D
0x1294	GWEIE4	Error Interrupt Enable Register 4	C, O, D
0x1298	GWEID4	Error Interrupt Disable Register 4	C, O, D
0x12A0	GWEIS5	Error Interrupt Status Register 5	C, O, D
0x12A4	GWEIE5	Error Interrupt Enable Register 5	C, O, D
0x12A8	GWEID5	Error Interrupt Disable Register 5	C, O, D

Note:
- Base address: GWCA0 = 0x403C_E000, GWCA0_NS = 0x503C_E000
- All registers can be read in any mode.
- The address preceded by “E:” correspond to an emulation address which allows a register to be read without modifying its content.
- Write Access Mode:
  - Any: Register can be written in any mode.
  - R: Register can be written in RESET mode
  - D : Register can be written in DISABLE mode
  - C: Register can be written in CONFIG mode
  - O: Register can be written in OPERATION mode

34.3 GWCA Register Descriptions

34.3.1 GWCA Mode Function Registers

34.3.1.1 GWMC : Mode Configuration Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0000

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

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

Bit	Symbol	Function	R/W
1:0	OPC[1:0]	Operating Mode Command[*1] Used to set GWCA mode. For more details, see section 34.4.1. Operation Modes. 0 0: Enter RESET mode 0 1: Enter DISABLE mode 1 0: Enter CONFIG mode 1 1: Enter OPERATION mode	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. HW: This register is not writable if its value is different than GWMS.OPS.

34.3.1.2 GWMS : Mode Status Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0004

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

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

Bit	Symbol	Function	R/W
1:0	OPS[1:0]	Operating Mode Status Indicates the current GWCA mode. For more details, see section 34.4.1. Operation Modes. 0 0: RESET mode 0 1: DISABLE mode 1 0: CONFIG mode 1 1: OPERATION mode	R
31:2	—	These bits are read as 0.	R

OPS[1:0] bits (Operating Mode Status)

[Update conditions]

HW: Mode change completed.

This register is change at “Maximum frame communication time (exp: 64 KB / 100 Mbps = 5.12 ms)” from OPERATION to DISABLE.

34.3.2 Reception Function Registers

34.3.2.1 GWIRC : IPV Remapping Configuration Register [802.1Q]

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	IPVR7[2:0]			—	IPVR6[2:0]			—	IPVR5[2:0]			—	IPVR4[2:0]		
Value after reset:	0	1	1	1	0	1	1	0	0	1	0	1	0	1	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	IPVR3[2:0]			—	IPVR2[2:0]			—	IPVR1[2:0]			—	IPVR0[2:0]		
Value after reset:	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
2:0	IPVR0[2:0]	IPV remapping 0	R/W
3	—	These bits are read as 0. The write value should be 0.	R/W
6:4	IPVR1[2:0]	IPV remapping 1	R/W
7	—	These bits are read as 0. The write value should be 0.	R/W
10:8	IPVR2[2:0]	IPV remapping 2	R/W
11	—	These bits are read as 0. The write value should be 0.	R/W
14:12	IPVR3[2:0]	IPV remapping 3	R/W
15	—	These bits are read as 0. The write value should be 0.	R/W
18:16	IPVR4[2:0]	IPV remapping 4	R/W
19	—	These bits are read as 0. The write value should be 0.	R/W
22:20	IPVR5[2:0]	IPV remapping 5	R/W
23	—	These bits are read as 0. The write value should be 0.	R/W
26:24	IPVR6[2:0]	IPV remapping 6	R/W
27	—	These bits are read as 0. The write value should be 0.	R/W
30:28	IPVR7[2:0]	IPV remapping 7	R/W
31	—	These bits are read as 0. The write value should be 0.	R/W

IPVRi (i = 0 to 7) bits (IPV remapping i)

Configure to which descriptor queue descriptor received with IPV i will be stored (when a descriptor is received from forwarding engine with FDESCR.IPV [MFWD] equal to i).

34.3.2.2 GWRDQSC : RX Descriptor Queue Security Configuration Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0014

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	RDQSL7	RDQSL6	RDQSL5	RDQSL4	RDQSL3	RDQSL2	RDQSL1	RDQSL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RDQSL7 to RDQSL0	RX Descriptor Queue i Security Level (The variable i corresponds to the bit number.) When a queue is secured, an unsecure descriptor cannot enter it (when a descriptor is from MFWD with FDESCR.SEC equal to 0). 0: Queue i unsecure 1: Queue i secure	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

34.3.2.3 GWRDQC : RX Descriptor Queue Control Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0018

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	RDQP7	RDQP6	RDQP5	RDQP4	RDQP3	RDQP2	RDQP1	RDQP0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	RDQD7	RDQD6	RDQD5	RDQD4	RDQD3	RDQD2	RDQD1	RDQD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RDQD7 to RDQD0	RX Descriptor Queue i Disable (The variable i corresponds to the bit position number.) Avoid descriptor from MFWD to enter the corresponding queue. The Forwarding Engine will reject the corresponding descriptor. If disabled queue data is not needed anymore, corresponding GWRMFSCq.MFS register can be set to 0 in order to avoid more frames to go to the CPU RAM. 0: Queue i enabled 1: Queue i disabled	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
23:16	RDQP7 to RDQP0	RX Descriptor Queue i Pause*1 (The variable i corresponds to the bit position number.) Avoid frames to be sent to the CPU by stopping descriptor fetching from the descriptor RAM. 0: Queue i active 1: Queue i paused	R/W*2
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: Pausing a queue during a long time could result in switch overflow. In case of switch overflow [COMA] this register should be set to 0.

Note 2. Read value differs from written value.

RDQDi bits (RX Descriptor Queue i Disable)

[Setting condition]

SW: Writing 1 to this bit will set it.

[Clearing condition]

SW: Writing 0 to this bit will clear it.

HW: Going out of OPERATION mode will clear this register (GWMC.OPC != 11b).

RDQPi bits (RX Descriptor Queue i Pause)

[Setting condition]

SW: Writing 1 to this bit will set it.

[Clearing condition]

SW: Writing 0 to this bit will clear it.

HW: Going out of OPERATION mode will clear this register (GWMC.OPC != 11b).

[Cautions]

This register is used to stop a queue but MFWD does not stop sending descriptor to the corresponding queue. In this case the queue might overflow.

34.3.2.4 GWRDQAC : RX Descriptor Queue Arbitration Control Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x001C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RDQA7[3:0]				RDQA6[3:0]				RDQA5[3:0]				RDQA4[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RDQA3[3:0]				RDQA2[3:0]				RDQA1[3:0]				RDQA0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	RDQA0[3:0]	RX Descriptor Queue 0 Arbitration*1 For more details, see section 34.5.3.1.2. Descriptor Arbitration. 0: Queue i strict arbitration Others: Queue i WRR arbitration	R/W
7:4	RDQA1[3:0]	RX Descriptor Queue 1 Arbitration*1 For more details, see section 34.5.3.1.2. Descriptor Arbitration. 0: Queue i strict arbitration Others: Queue i WRR arbitration	R/W
11:8	RDQA2[3:0]	RX Descriptor Queue 2 Arbitration*1 For more details, see section 34.5.3.1.2. Descriptor Arbitration. 0: Queue i strict arbitration Others: Queue i WRR arbitration	R/W
15:12	RDQA3[3:0]	RX Descriptor Queue 3 Arbitration*1 For more details, see section 34.5.3.1.2. Descriptor Arbitration. 0: Queue i strict arbitration Others: Queue i WRR arbitration	R/W
19:16	RDQA4[3:0]	RX Descriptor Queue 4 Arbitration*1 For more details, see section 34.5.3.1.2. Descriptor Arbitration. 0: Queue i strict arbitration Others: Queue i WRR arbitration	R/W

Bit	Symbol	Function	R/W
23:20	RDQA5[3:0]	RX Descriptor Queue 5 Arbitration*1 For more details, see section 34.5.3.1.2. Descriptor Arbitration. 0: Queue i strict arbitration Others: Queue i WRR arbitration	R/W
27:24	RDQA6[3:0]	RX Descriptor Queue 6 Arbitration*1 For more details, see section 34.5.3.1.2. Descriptor Arbitration. 0: Queue i strict arbitration Others: Queue i WRR arbitration	R/W
31:28	RDQA7[3:0]	RX Descriptor Queue 7 Arbitration*1 For more details, see section 34.5.3.1.2. Descriptor Arbitration. 0: Queue i strict arbitration Others: Queue i WRR arbitration	R/W

Note 1.
- SW: In hybrid arbitration mode, all the queues with a GWRDQAC.RDQAi value other than 0 should have a consecutive priority (priority refers to the queue number i).
- SW: In all arbitration mode except strict priority arbitration mode, at least two queues should have a GWRDQAC.RDQAi other than 0.

34.3.2.5 GWRGC : RX General Configuration Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0020

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

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

Bit	Symbol	Function	R/W
0	RCPT	Receive CRC Pass Through [802.3]*1 This bit selects to pass FCS field on the reception frame. 0: FCS is not passed to GWCPU. If a descriptor is received from forwarding engine with FDESCR.FI equal to 1, GWCA remove the last 4 bytes of the Frame Data (FCS). 1: FCS is passed to GWCPU GWCA don't remove FCS from the Frame which has one (a descriptor is received from forwarding engine with FDESCR.FI equal to 1) if the frame has not been modified by the switch (No update due to VLAN tagging/untagging and no update requested by L3 routing [FWD]).	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. HW: GWCA does not check FCS integrity.

34.3.2.6 GWRMFSCq : Reception Maximum Frame Size Configuration Register q (q = 0 to 7)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0040 + 0x04 × q

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

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

Bit	Symbol	Function	R/W
15:0	MFS[15:0]	Maximum Frame Size*1 Maximum frame size for descriptor queue q. All bigger frames will be rejected.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. HW: The maximum frame size comparison takes in account the frame size which will actually be written by the AXI master (after VLAN untagging, R-TAG insertion and L2L3 update) but always counts in this size FCS even if not received by CPU.

34.3.2.7 GWRDQDCq : Reception Descriptor Queue q Depth Configuration Register (q = 0 to 7)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0060 + 0x04 × q

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

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

Bit	Symbol	Function	R/W
9:0	DQD[9:0]	Descriptor Queue Depth*1 Number of descriptors that can contain descriptor queue q. For details see section 34.5.3.1.1. Descriptor Storage.	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: The sum of DQD fields should always be smaller or equal to 512.

34.3.2.8 GWRDQMq : RX Descriptor Queue q Monitoring Register (q = 0 to 7)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0080 + 0x04 × q

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

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

Bit	Symbol	Function	R/W
9:0	DNQ[9:0]	Descriptor Number in Queue Indicate the current number of descriptors stored in RX Descriptor Queue q.	R
31:10	—	These bits are read as 0.	R

DNQ[9:0] bits (Descriptor Number in Queue)

[Increment conditions]

HW: Incremented by 1 when a descriptor is received from MFWD for the corresponding queue and, the queue is not full, there is no descriptor security error and the queue is not disabled.

[Decrement conditions]

HW: Decremented by 1 when a descriptor is read by GWCA to send data to AXI Master Interface or to reject data.

34.3.2.9 GWRDQMLMq : RX Descriptor Queue q Max Level Monitoring Register (q = 0 to 7)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x00A0 + 0x04 × q

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

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

Bit	Symbol	Function	R/W
9:0	DMLQ[9:0]	Descriptor Max Level in Queue Indicates the maximum number of descriptors that has been stored in RX Descriptor Queue q.	R
31:10	—	These bits are read as 0.	R

DMLQ[9:0] bits (Descriptor Max Level in Queue)

[Clearing condition]

HW: Being in RESET mode will clear this register (GWMC.OPS == 00).

SW: By reading this register, it is cleared to GWRDQMq.DNQ.

[Increment conditions]

HW: Increment to GWRDQMq.DNQ value when smaller than GWRDQMq.DNQ.

34.3.3 Reception Multicast Function Registers

34.3.3.1 GWMTIRM : Multicast Table Initialization Register Monitoring Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0100

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

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

Bit	Symbol	Function	R/W
0	MTIOG	Multicast Table Initialization Ongoing [Setting condition] SW: By writing 1 to this register. It starts Multicast Table initialization. [Clearing condition] HW: This bit is cleared when Multicast Table initialization is finished. Min clk_period * 64 = [exp] (1000/320) * 128 = 400 ns	R/W*1
1	MTR	Multicast Table Ready [Setting condition] When GWMTIRM.MTIOG is getting cleared. [Clearing condition] By writing 1 to GWMTIRM.MTIOG.	R
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

34.3.3.2 GWMSTLS : Multicast Table Learning Setting Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0104

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	MSENL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
5:0	MNRCNL[5:0]	Multicast Next Descriptor Chain Number Learn This register is used to set the next address in the multicast table to be learnt in GWMSTLS.MSENL multicast table entry, see section 34.5.3.3.3. Multicast Learning.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
10:8	MNL[2:0]	Multicast Number Learn This register is used to set the multicast number that should be learnt in GWMSTLS.MSENL multicast table entry. 0 0 0: Multicast 0 (1 target, no multicast) 0 0 1: Multicast 1 (2 targets) : 1 1 1: Multicast 7 (8 targets)	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
21:16	MSENL[5:0]	Multicast Setting Entry Number Learn This register is used to set in which address the current learning should happen.	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

34.3.3.3 GWMSTLR : Multicast Table Learning Result Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0108

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

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

Bit	Symbol	Function	R/W
0	MTLF	Multicast Table Learning Fail [Update Conditions] HW: GWMSTLR.MTL clear event. 0: Entry learning did not fail because the Multicast table is not ready. 1: Entry learning failed the Multicast table is not ready.	R
30:1	—	These bits are read as 0.	R
31	MTL	Multicast Table Learning [Setting condition] SW: Writing GWMSTLS register will set this bit. [Clearing condition] HW: This bit will be deasserted when the Multicast RAM write is completed.	R

34.3.3.4 GWMSTSS : Multicast Table Searching Setting Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x010C

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

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

Bit	Symbol	Function	R/W
5:0	MSENS[5:0]	Multicast Setting Entry Number Search This register is used to read multicast table.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

34.3.3.5 GWMSTSR : Multicast Table Searching Result Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0110

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

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

Bit	Symbol	Function	R/W
5:0	MNRCNR[5:0]	Multicast Next RX Descriptor Chain Number Result This register is used to read multicast table. [Update conditions] HW: GWMSTSR.MTS clear event.	R
7:6	—	These bits are read as 0.	R
10:8	MNR[2:0]	Multicast Number Result This register is used to read multicast table. [Update conditions] HW: GWMSTSR.MTS clear event.	R
30:11	—	These bits are read as 0.	R
31	MTS	Multicast Table Searching This register is used to read multicast table. [Setting condition] SW: Writing GWMSTSS register will set this bit. [Clearing condition] HW: This bit will be deasserted when the Multicast RAM read is completed.	R

34.3.4 MAC Function Registers

34.3.4.1 GWMAC0 : MAC Address Configuration Register 0

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0120

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

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

Bit	Symbol	Function	R/W
15:0	MAU[15:0]	MAC Address Upper Part These bits are used to set the 16 higher-order bits of the MAC address. For example, if the MAC address is 01-23-45-67-89-AB (hexadecimal), set 0x0123 in this register.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

34.3.4.2 GWMAC1 : MAC Address Configuration Register 1

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0124

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

Bit	Symbol	Function	R/W
31:0	MAL[31:0]	MAC Address Lower Part These bits are used to set the 32 lower-order bits of the MAC address. For example, if the MAC address is 01-23-45-67-89-AB (hexadecimal), set 0x456789AB in this register.	R/W

34.3.5 TAG Function Registers

34.3.5.1 GWVCC : VLAN Control Configuration Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0130

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

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

Bit	Symbol	Function	R/W
0	VIM	VLAN Ingress Mode*1 0: Incoming VLAN mode 1: Port based VLAN mode	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W
18:16	VEM[2:0]	VLAN Egress Mode 000: No VLAN mode, frames are transmitted with no VLAN. 001: C-TAG VLAN mode, frames are transmitted with ingress C-TAG if there is one stored in the Local RAM. 010: HW C-TAG VLAN mode, frames are transmitted with the C-TAG stored in local RAM. 011: SC-TAG VLAN mode, frames are transmitted with ingress C-TAG and S-TAG if they were stored in the Local RAM. 100: HW SC-TAG VLAN mode, frames are transmitted with the C-TAG and S-TAG stored in local RAM.	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: This register should not be set to 1 is the switch is in no VLAN mode (In forwarding engine, FWGC.SVM is set to 00 [FWD])

34.3.5.2 GWVTC : VLAN TAG Configuration Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0134

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	STD	STP[2:0]			STV[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CTD	CTP[2:0]			CTV[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	CTV[11:0]	C-TAG VLAN	R/W
14:12	CTP[2:0]	C-TAG PCP	R/W
15	CTD	C-TAG DEI	R/W
27:16	STV[11:0]	S-TAG VLAN	R/W
30:28	STP[2:0]	S-TAG PCP	R/W
31	STD	S-TAG DEI	R/W

34.3.5.3 GWTTFC : Transmission TAG Filtering Configuration Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0138

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	UT	SCRT	SCT	CRT	CT	CSRT	CST	RT	NT
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NT	No Tag 0: No Tag frame passed 1: No Tag frame rejected	R/W
1	RT	R-TAG 0: R-TAG frame passed 1: R-TAG frame rejected	R/W
2	CST	CoS-TAG 0: CoS-TAG frame passed 1: CoS-TAG frame rejected	R/W
3	CSRT	CoSR-TAG 0: CoSR-TAG frame passed 1: CoSR-TAG frame rejected	R/W

Bit	Symbol	Function	R/W
4	CT	C-TAG 0: C-TAG frame passed 1: C-TAG frame rejected	R/W
5	CRT	CR-TAG 0: CR-TAG frame passed 1: CR-TAG frame rejected	R/W
6	SCT	SC-TAG 0: SC-TAG frame passed 1: SC-TAG frame rejected	R/W
7	SCRT	SCR-TAG 0: SCR-TAG frame passed 1: SCR-TAG frame rejected	R/W
8	UT	Unknown TAG 0: Unknown Tag frame passed 1: Unknown Tag frame rejected	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

34.3.6 Timestamp Function Registers

34.3.6.1 GWTDCACs0 : Timestamp Descriptor Chain s Address Configuration Register 0 (s = 0, 1)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0140 + 0x08 × s

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	TSCCAU[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TSCCAU[7:0]	TS Descriptor Chain s Current Address Upper Part*1	R/W *2
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: To overwrite the current address, GWTDCACs0 and GWTDCACs1 should always both be written stating by GWTDCACs1 and finishing by GWTDCACs0.
Note 2. Read value differs from written value.

TSCCAU[7:0] bits (TS Descriptor Chain s Current Address Upper Part)

[Update conditions]

SW: Writing to this field with update it to the write value.

HW: When an FEMPTY_ND descriptor is read for descriptor queue s, this field value is automatically updated to the next descriptor address upper part.

HW: When a LINK or LINKFIX descriptor is read, this register value is updated to DESCR.PTR[39:32].

34.3.6.2 GWTDCACs1 : Timestamp Descriptor Chain s Address Configuration Register 1 (s = 0, 1)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0144 + 0x08 × s

Bit position:	31–0
Bit field:	TSCCAL[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TSCCAL[31:0]	TS Descriptor Chain s Current Address Lower Part*1	R/W*2

Note 1. SW: To overwrite the current address, GWTDCACs0 and GWTDCACs1 should always both be written stating by GWTDCACs1 and finishing by GWTDCACs0.

Note 2. Read value differs from written value.

TSCCAL[31:0] bits (TS Descriptor Chain s Current Address Lower Part)

[Update conditions]

SW: Writing GWTDCACs0 register with update this field to the previous value written on it by the same APB.

HW: When an FEMPTY_ND descriptor is read for descriptor queue s, this field value is automatically updated to the next descriptor address downer part.

HW: When a LINK or LINKFIX descriptor is read, this register value is updated to DESCR.PTR[31:0].

34.3.6.3 GWTSDCCs : Timestamp Descriptor Chain s Configuration Register (s = 0, 1)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0160 + 0x04 × s

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	OSID[2:0]			—	—	—	—	—	—	DCS	TE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TE	Timer Enable*1 Enable timestamp reception for timestamps taken with timer s.	R/W
1	DCS	Descriptor Chain Select Select to which chain timestamps taken with timer s will be received.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
10:8	OSID[2:0]	OS ID When a memory access is done for timer number s, the OSID set in this register will be used.	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. HW: If a timestamp is received from RMAC for a timer which is disabled for timestamp reception, the timestamp will be ignored and lost. This is a valid setting (Another GWCA could have taken the timestamp) so there is no flag to monitor this event.

34.3.6.4 GWTSNM : Timestamp Number Monitoring Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0180

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	TNTR[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	TNTR[4:0]	Timestamp Number in Timestamp RAM [Increment conditions] HW: Incremented by 1 when a timestamp is received from Ethernet Agents. [Decrement conditions] HW: Decremented by 1 when a timestamp is read from timestamp RAM.	R
31:5	—	These bits are read as 0.	R

34.3.6.5 GWTSMNM : Timestamp Maximum Number Monitoring Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0184

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	TMNTR[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	TMNTR[4:0]	Timestamp Maximum Number in Timestamp RAM [Update conditions] HW: Set to GWTSNM.TNTR when GWTSNM.TNTR> GWTSNM.TMNTR. [Clearing condition] HW: Being in RESET mode will clear this register. SW: Reading this register clears it to GWTSNM.TNTR value.	R
31:5	—	These bits are read as 0.	R

34.3.7 AXI Master Registers

34.3.7.1 GWAC : AXI Control Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0190

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AMP	AMPR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AMPR	AXI Master Pause Request This register blocks the AXI master from making new accesses (Because the AXI master has register slices for timing purpose few AXI accesses can be emitted before stopping). 0: AXI master pause not requested 1: AXI master pause requested	R/W
1	AMP	AXI Master Paused “AXI Master Pause” stop only the final processing of the AXI interface. (Internal processing cannot be stopped.) For example, the AXI address for chain [i] is loaded immediately before “AXI Master Paused”. This address will used after “AXI Master not paused” regardless of AXI address re-configuration (by GWDCCi) during “AXI Master Pause”. 0: AXI master not paused (transaction ongoing) 1: AXI master paused (no transaction ongoing)	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

AMP bit (AXI Master Paused)

[Setting condition]

HW: When GWAC.AMPR is set to 1’b1 and all ongoing AXI transactions are completed, this bit will be set.

[Clearing condition]

HW: When GWAC.AMPR is set to 0, this bit will be set to 0 (it can take time because of asynchronous modules).

34.3.7.2 GWDCBAC0 : Descriptor Chain Base Address Configuration Register 0

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0194

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	DCBAU[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	DCBAU[7:0]	Descriptor Chain Base Address Upper Part For further setting information, see section 34.5.1.3.1. LINKFIX Table Initialization.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

34.3.7.3 GWDCBAC1 : Descriptor Chain Base Address Configuration Register 1

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0198

Bit position:	31–0
Bit field:	DCBAL[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	DCBAL[31:0]	Descriptor Chain Base Address Lower Part For further setting information, see section 34.5.1.3.1. LINKFIX Table Initialization.	R/W

34.3.7.4 GWMDNC : Maximum Descriptor Number Configuration Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x01A0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSDMN[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TXDMN[4:0]					—	—	—	RXDMN[4:0]				
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
4:0	RXDMN[4:0]	RX Descriptor Maximum Number Configures the maximum number of descriptors that can be processed at once for an RX queue (It limits to GWMDNC.RXDMN+1 the number of descriptors read that can be done for an RX frame). See GWEIS0.RXDNES register explanation.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
12:8	TXDMN[4:0]	TX Descriptor Maximum Number*1 Configures the maximum number of descriptors that can be processed at once for a TX queue (It limits to GWMDNC.TXDMN+1 the number of descriptors read that can be done for a TX frame). See GWEIS0.TXDNES.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
17:16	TSDMN[1:0]	Timestamp Descriptor Maximum Number Configures the maximum number of descriptors that can be processed at once for a TS queue (It limits to GWMDNC.TSDMN+1 the number of descriptors read that can be done for a timestamp). See GWEIS0.TSDNES.	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: The switch maximum input frame size is 60 KB so this register should be set to a value smaller or equal to 30. To avoid any unexpected frame loss due to this register, any software sending frames bigger than 58 KB should avoid having more than one consecutive LINK/LINKFIX descriptor in a descriptor chain.

RXDMN[4:0] bits (RX Descriptor Maximum Number)

[Cautions]

SW: This maximum descriptor number takes in account all descriptors including LINK, LINKFIX descriptors and descriptors remaining from a previous error frame like FMID_EMPTY and FEND_EMPTY descriptors. It should be taken in account in account while setting this register.

TXDMN[4:0] bits (TX Descriptor Maximum Number)

[Cautions]

SW: This maximum descriptor number takes in account all descriptors including LINK, LINKFIX descriptors and descriptors remaining from a previous error frame like FMID and FEND descriptors. It should be taken in account while setting this register.

34.3.7.5 GWTRCi : Transmission Request Configuration Register i (i = 0, 1)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0200 + 0x04 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TSRj	TSRj	TSRj	TSRj	TSRj	TSRj	TSRj	TSRj	TSRj	TSRj	TSRj	TSRj	TSRj	TSRj	TSRj	TSRj
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TSRj	TSRj	TSRj	TSRj	TSRj	TSRj	TSRj	TSRj	TSRj	TSRj	TSRj	TSRj	TSRj	TSRj	TSRj	TSRj
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
b (b = 0 to 31)	TSRj (j = 32 × i + b)	Transmission Start Request j (j = 0 to 63)*1 This register is used to start transfer from CPU to switch for its corresponding TX queue. 0: TX Descriptor Chain is empty. 1: Transmission request is ongoing.	R/W*2

Note 1. HW: For RX queues, this bit cannot be set.
Note 2. Read value differs from written value.

TSRj bits (Transmission Start Request j (j = 0 to 63))

[Setting condition (only for TX queues)]

SW: Writing 1 to this bit will this bit set if queue t is a transmission queue (GWDCCi.DQT is set).

HW: a transmission request received through INTER_TX[t] interface will set this bit if queue t is a transmission queue (GWDCCi.DQT is set).

[Clearing condition]

HW: When HW read a descriptor other than FSINGLE, FSTART, FMID, FEND, LINK and LINKFIX. *

HW: When an AXI error occurs while reading a descriptor in the corresponding AXI descriptor queue. *

HW: When a descriptor number error occurs while reading the corresponding AXI descriptor queue. *

HW: Going out of OPERATION mode will clear this register.

34.3.7.6 GWTPCp : Transmission Pause Configuration Register p (p = 0 to 1)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0300 + 0x04 × p

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	PPPL7	PPPL6	PPPL5	PPPL4	PPPL3	PPPL2	PPPL1	PPPL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	PPPL7 to PPPL0	Per Priority Pause Level i (The variable i corresponds to the bit position number.) 0: Transmission queues with their priority GWDCCi.DCP set to i are not paused when pause level p is set (PAUSE[2][p] is set [COMA]). 1: Transmission queues with the priority GWDCCi.DCP set to i are paused when pause level p is set (PAUSE[2][p] is set [COMA]).	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

34.3.7.7 GWARIRM : AXI RAM Initialization Register Monitoring Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0380

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARR	ARIOG
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ARIOG	AXI RAM Initialization Ongoing [Setting condition] SW: By writing 1 to this register. It starts AXI RAM initialization. [Clearing condition] HW: This bit is cleared when AXI RAM initialization is finished. Min clk_period * 64 = [exp] (1000/320) * 128 = 400 ns	R/W*1
1	ARR	AXI RAM Ready [Setting condition] When GWARIRM.ARIOG is getting cleared. [Clearing condition] By writing 1 to GWARIRM.ARIOG.	R
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

34.3.7.8 GWDCCi : Descriptor Chain i Configuration Register (i = 0 to 63)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0400 + 0x04 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	OSID[2:0]			—	—	—	BALR	—	—	—	—	—	DCP[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	DQT	SL	ETS	EDE	—	—	—	—	—	—	SM[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	SM[1:0]	Synchronization Mode 0 0: Normal mode (full descriptor write back) 0 1: No-write-back mode (no descriptor write back) 1 0: Keep-DT mode (no update of DT field at descriptor write back) 1 1: Reserved	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	EDE	Extended Descriptor Enable 0: Basic Descriptor 1: Extended Descriptor	R/W
9	ETS	Enable Timestamp Storage*1 0: Timestamp is not added in the Descriptor. 1: Timestamp is added in the Descriptor.	R/W
10	SL	Security Level*1 When a chain is secured, an unsecure descriptor cannot enter it (when a descriptor is received from MFWD with FDESCR.SEC equal to 0). 0: Chain i unsecure 1: Chain i secure	R/W
11	DQT	Descriptor Queue Type 0: Descriptor queue i is a reception queue. 1: Descriptor queue i is a transmission queue.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
18:16	DCP[2:0]	Descriptor Chain Priority*2 0 0 0: Lowest priority 1 1 1: Highest priority	R/W*3
23:19	—	These bits are read as 0. The write value should be 0.	R/W
24	BALR	Base Address Load Request This bit is used to reset current_address field in the AXI address RAM to {{GWDCBAC.DCBAU, GWDCBAC.DCBAL} + I × 8} [Setting condition] SW: Writing 1 to this bit will set it. [Clearing condition] HW: This bit will be cleared when the current_address field in the AXI address RAM is reset.	R/W*3
27:25	—	These bits are read as 0. The write value should be 0.	R/W
30:28	OSID[2:0]	OS ID When a memory access is done for descriptor queue number i the OSID set in this register will be used.	R/W
31	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. HW: This register is only valid for RX queues.
Note 2. • HW: This register is only valid for TX queues.

- SW: This register value should not be changed for a TX queue when GWTRCj.TSRj is set. (When transmission is ongoing for the corresponding queue)

Note 3. Read value differs from written value.

34.3.7.9 GWAARSS : AXI Address RAM Searching Setting Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0800

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	AARA[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	AARA[5:0]	AXI Address RAM Address*1 This register is used to read the current address used in AXI master for GWAARSS.AARA descriptor queue.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. HW: Due to the pipeline architecture of the AXI master, the AXI read can just give an idea of which address is processing the AXI master and is not precise enough to be used for HW/SW synchronization. This register is only here for debug purpose.

34.3.7.10 GWAARSR0 : AXI Address RAM Searching Result Register 0

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0804

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	AARS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	ACARU[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	ACARU[7:0]	AXI Current Address Result Upper Part*1 Displays AXI address read value. [Update conditions] HW: GWAARSR0.AARS clear event.	R
30:8	—	These bits are read as 0.	R
31	AARS	AXI Address RAM Searching This register is used to read an in AXI address RAM. [Setting condition] SW: Writing GWAARSS will set this bit. [Clearing condition] HW: This bit will be deasserted when the AXI address RAM search is completed.	R

Note 1. AXI address read will not restart from the base address of chain after RESET mode (Because being in RESET mode will not clear this register).

34.3.7.11 GWAARSR1 : AXI Address RAM Searching Result Register 1

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0808

Bit position:	31–0
Bit field:	ACARL[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	ACARL[31:0]	AXI Current Address Result Lower Part*1 Displays AXI address read value. [Update conditions] HW: GWAARSR0.AARS clear event.	R

Note 1. AXI address read will not restart from the base address of chain after RESET mode (Because being in RESET mode will not clear this register).

34.3.7.12 GWIDAUASi : Incremental Data Area i Used Area Size Register (i = 0 to 3)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0840 + 0x04 × i

Bit position:	31								23–0
Bit field:	—	—	—	—	—	—	—	—	IDAUAS[23:0]
Value after reset:	0	0	0	0	0	0	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
23:0	IDAUAS[23:0]	Incremental Data Area Used Size of RX Descriptor Chain i*1 Represents the byte numbers that have been used in the corresponding incremental area. [Clearing condition] HW: Being in RESET mode will clear this register. HW: When a FEMPTY_IS descriptor is read in the corresponding descriptor queue, this register gets cleared. [Increment conditions] HW: When a frame has been stored in the corresponding incremental area, this register is incremented by the frame size. [Decrement conditions] SW: By writing a value, SW will decrement this register by this value.	R/W*2
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: should always write a value smaller than the current register value.
Note 2. Read value differs from written value.

34.3.7.13 GWIDASMi : Incremental Data Area i Size Monitoring Register (i = 0 to 3)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0880 + 0x04 × i

Bit position:	31								23–0
Bit field:	—	—	—	—	—	—	—	—	IDAS[23:0]
Value after reset:	0	0	0	0	0	0	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
23:0	IDAS[23:0]	Incremental Data Area Size of RX Descriptor Chain i This register indicates the size of RX descriptor chain i incremental area in bytes. [Clearing condition] HW: Being in RESET mode will clear this register. [Update conditions] HW: When a FEMPTY_IS descriptor is read in the corresponding descriptor queue, this register is updated to DESCR.DS × 4096.	R
31:24	—	These bits are read as 0.	R

34.3.7.14 GWIDASAMi0 : Incremental Data Area i Start Address Monitoring Register 0 (i = 0 to 3)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0900 + 0x08 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	IDASAU[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	IDASAU[7:0]	Incremental Data Area Start Address Upper Part of RX Descriptor Chain i [Clearing condition] HW: Being in RESET mode will clear this register. [Update conditions] HW: When a FEMPTY_IS descriptor is read in the corresponding descriptor queue, this register is updated to DESCR.PTR[39:32].	R
31:8	—	These bits are read as 0.	R

34.3.7.15 GWIDASAMi1 : Incremental Data Area i Start Address Monitoring Register 1 (i = 0 to 3)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0904 + 0x08 × i

Bit position:	31–0
Bit field:	IDASAL[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	IDASAL[31:0]	Incremental Data Area Start Address Lower Part of RX Descriptor Chain i [Clearing condition] HW: Being in RESET mode will clear this register. [Update conditions] HW: When a FEMPTY_IS descriptor is read in the corresponding descriptor queue, this register is updated to DESCR.PTR[31:0].	R

34.3.7.16 GWIDACAMi0 : Incremental Data Area i Current Address Monitoring Register 0 (i = 0 to 3)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0980 + 0x08 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	IDACAU[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	IDACAU[7:0]	Incremental Data Area Current Address Upper Part of RX Descriptor Chain i [Clear conditions] HW: Being in RESET mode will clear this register. [Update conditions] HW: When a FEMPTY_IS descriptor is read in the corresponding descriptor queue, this register is updated to DESCR.PTR[39:32]. HW: When a burst is decided to be sent in the incremental area, {GWIDACAMi0.IDACAU, GWIDACAMi1.IDACAL} is incremented by the burst size. HW: {GWIDACAMi0.IDACAU, GWIDACAMi1.IDACAL} overflows at {GWIDASAMi0.IDASAU, GWIDASAMi1.IDASAL} + GWIDASMi.IDAS-1.	R
31:8	—	These bits are read as 0.	R

34.3.7.17 GWIDACAMi1 : Incremental Data Area i Current Address Monitoring Register 1 (i = 0 to 3)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0984 + 0x08 × i

Bit position: 31 … 0

Bit field: IDACAL[31:0]

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	IDACAL[31:0]	Incremental Data Area Current Address Lower Part of RX Descriptor Chain i [Clear conditions] HW: Being in RESET mode will clear this register. [Update conditions] HW: When a FEMPTY_IS descriptor is read in the corresponding descriptor queue, this register is updated to DESCR.PTR[31:0]. HW: When a burst is decided to be sent in the incremental area, {GWIDACAMi0.IDACAU, GWIDACAMi1.IDACAL} is incremented by the burst size. HW: {GWIDACAMi0.IDACAU, GWIDACAMi1.IDACAL} overflows at {GWIDASAMi0.IDASAU, GWIDASAMi1.IDASAL} + GWIDASMi.IDAS-1.	R

34.3.8 Rate Limiter Function Registers

34.3.8.1 GWGRLC : Global Rate Limiter Configuration Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0A00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GRLULRS	GRLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	GRLIV[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	GRLIV[15:0]	Global Rate Limiter Incremental Value For further setting information see section 34.5.2.1.2. Rate Limiters. [Cautions] SW: Setting this register to 0 or a value close to 0 would set GWCA to a very low throughput or could stuck GWCA in transmission.	R/W
16	GRLE	Global Rate Limiter Enable The global rate limiter limits the throughput through all TX queues to avoid the switch to take in too much data at once. 0: Global Rate limiter disabled 1: Global Rate limiter enabled	R/W
17	GRLULRS	Global Rate Limiter Upper Limit Reached Status Flag If this bit gets set when only the global rate limiter is enabled (per port rate limiters disabled), it means that the AXI master is too slow to provide the throughput set in the global rate limiter or that the AXI latency has been underestimated. This bit can get set while using per port rate limiters. In this case it should be ignored. [Setting condition] HW: When the global rate limit reaches its upper limit GWGRLULC.GRLUL, this bit gets set. [Clearing condition] HW: Being in RESET mode will clear this register. SW: Writing one to this bit will clear it.	R/W*1
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

34.3.8.2 GWGRLULC : Global Rate Limiter Upper Limit Configuration Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0A04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	GRLUL[23:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	GRLUL[23:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
23:0	GRLUL[23:0]	Global Rate Limiter Upper Limit*1 Set the maximum number of credits in bit rate limiter that can accumulate before stopping. For further setting information see section 34.5.2.1.2. Rate Limiters.	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: This register should not be set to a value bigger than 0x800000.

34.3.8.3 GWRLCi : Rate Limiter i Configuration Register (i = 0 to 7)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0A80 + 0x8 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	RLIV[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	RLIV[11:0]	Rate Limiter Incremental Value For further setting information see section 34.5.2.1.2. Rate Limiters. [Cautions] SW: Setting this register to 0 or a value close to 0 would set corresponding queue to a very low throughput or could stuck the corresponding queue.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
16	RLE	Rate Limiter Enable Rate limiter i limit the through put for AXI descriptor queue number 63-I. For details, see section 34.5.1.1. AXI Descriptor Queue Mapping. 0: Rate limiter i disabled 1: Rate limiter i enabled	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

34.3.8.4 GWRLULCi : Rate Limiter i Upper Limit Configuration Register (i = 0 to 7)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0A84 + 0x8 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	RLUL[23:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RLUL[23:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
23:0	RLUL[23:0]	Rate Limiter Upper Limit*1 Set the maximum number of credits in bit rate limiter i can accumulate before stopping. This register can be used if a lot of traffic happens on the AXI bus so burst on queues i make the switch overflow. If not required, set this register to 0x80_0000. For further setting information see section 34.5.2.1.2. Rate Limiters.	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: This register should not be set to a value bigger than 0x80_0000.

34.3.9 Interrupt Function Registers

34.3.9.1 GWIDPC : Interrupt Delay Prescaler Configuration Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0B80

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	IDPV[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	IDPV[9:0]	Interrupt Delay Prescaler Value This register is used to create an internal clock. This internal clock period will be clk_delay_period = clk_period*(GWIDPC.IDPV+1) where clk_period is the period of clk. For further information, see section 34.5.6. Interrupt Delay Function.	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

34.3.9.2 GWIDCi : Interrupt Delay i Configuration Register (i = 0 to 63)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x0C000 + 0x4 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	IDV[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	IDV[11:0]	Interrupt Delay Value Configures the delay between the interrupt status flag GWDISi.DISt setting and the actual interrupt event in us. If this register is set to 0, data interrupt will occur without delay. If value X is set to this register, the interrupt will be delayed by a time between X-1*clk_delay_period us and X*clk_delay_period us where clk_delay_period is the internal clock created using GWIDPC register period. For further information, see section 34.5.6. Interrupt Delay Function.	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

34.3.10 GWCA Counter Registers

34.3.10.1 GWRDCN : Received Data Counter Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1000

Bit position:	31 ... 0
Bit field:	RDN[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	RDN[31:0]	Received Data Number This register counts data that has been received by GWCA (data aborted by AXI error are accounted here).	R

RDN[31:0] bits (Received Data Number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment conditions]

HW: Incremented by 1 when a frame is passed from multicast control to AXI master interface and if this register has a value other than 32.

34.3.10.2 GWTDCN : Transmitted Data Counter Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1004

Bit position:	31 ... 0
Bit field:	TDN[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TDN[31:0]	Transmitted Data Number This register counts data that has been transmit by GWCA (this includes data transmitted with an error and frames smaller than 32 bytes).	R

TDN[31:0] bits (Transmitted Data Number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment conditions]

HW: Incremented by 1 when a frame is passed from AXI master interface to TX data store and if this register has a value different than 32.

34.3.10.3 GWTSCN : Timestamp Counter Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1008

Bit position:	31–0
Bit field:	TN[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TN[31:0]	Timestamp Number This register counts timestamp processed by GWCA (TS lost because of AXI errors are accounted here).	R

TN[31:0] bits (Timestamp Number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment conditions]

HW: Incremented by 1 when a Timestamp is saved in timestamp RAM and if this register has a value other than 32.

34.3.10.4 GWTSOVFECN : Timestamp Overflow Error Counter Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x100C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TSOVFEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TSOVFEN[15:0]	Timestamp Overflow Error Number This register counts the number of timestamps lost because of timestamp RAM overflow.	R
31:16	—	These bits are read as 0.	R

TSOVFEN[15:0] bits (Timestamp Overflow Error Number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment conditions]

HW: Incremented by 1 when a timestamp overflow error happens and if this register has a value other than 16.

34.3.10.5 GWUSMFSECN : Under Minimum Frame Size Error Counter Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	USMFSEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	USMFSEN[15:0]	Under Switch Minimum Frame Size Error Number This register counts the number of transmit data lost because of under switch minimum size error.	R
31:16	—	These bits are read as 0.	R

USMFSEN[15:0] bits (Under Switch Minimum Frame Size Error Number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment conditions]

HW: Incremented by 1 when an Under Switch Minimum Frame Size Error happens and if this register has a value other than 16.

34.3.10.6 GWTFECN : TAG Filtering Error Counter Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TFEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TFEN[15:0]	TAG Filtering Error Number This register counts the number of transmit data lost because of TAG filtering.	R
31:16	—	These bits are read as 0.	R

TFEN[15:0] bits (TAG Filtering Error Number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment conditions]

HW: Incremented by 1 when a TAG filter error happens and if this register has a value different than 16.

34.3.10.7 GWSEQECN : Sequence Error Counter Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1018

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SEQEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	SEQEN[15:0]	Sequence Error Number This register counts the number of transmit data lost because of a sequence error.	R
31:16	—	These bits are read as 0.	R

SEQEN[15:0] bits (Sequence Error Number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment conditions]

HW: Incremented by 1 when a sequence error happens and if this register has a value different than 16.

34.3.10.8 GWTXDNECN : TX Descriptor Number Error Counter Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1020

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TXDNEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TXDNEN[15:0]	TX Descriptor Number Error Number This register counts the number of transmit data lost because of a TX Descriptor Number Error.	R
31:16	—	These bits are read as 0.	R

TXDNEN[15:0] bits (TX Descriptor Number Error Number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment conditions]

HW: Incremented by 1 when a TX Descriptor Number Error happens and if this register has a value other than 16.

34.3.10.9 GWFSECN : Frame Size Error Counter Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1024

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FSEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	FSEN[15:0]	Frame Size Error Number This register counts the number of receive data lost because of a Frame Size Error.	R
31:16	—	These bits are read as 0.	R

FSEN[15:0] bits (Frame Size Error Number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment conditions]

HW: Incremented by 1 when a Frame Size Error happens and if this register has a value different than 16.

34.3.10.10 GWTDFECN : Timestamp Descriptor Full Error Counter Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1028

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TDFEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TDFEN[15:0]	Timestamp Descriptor Full Error Number This register counts the number of timestamps lost because of a Timestamp Descriptor Full Error.	R
31:16	—	These bits are read as 0.	R

TDFEN[15:0] bits (Timestamp Descriptor Full Error Number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment conditions]

HW: Incremented by 1 when a Timestamp Descriptor Full Error happens and if this register has a value other than 16.

34.3.10.11 GWTSDNECN : Timestamp Descriptor Number Error Counter Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x102C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TSDNEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TSDNEN[15:0]	Timestamp Descriptor Number Error Number This register counts the number of timestamps loss because of a Timestamp Descriptor Number Error.	R
31:16	—	These bits are read as 0.	R

TSDNEN[15:0] bits (Timestamp Descriptor Number Error Number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment conditions]

HW: Incremented by 1 when a Timestamp Descriptor Number Error happens and if this register has a value other than 16.

34.3.10.12 GWDQOECN : Descriptor Queue Overflow Error Counter Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1030

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DQOEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DQOEN[15:0]	Descriptor Queue Overflow Error Number This register counts the number of receive data lost because of a Descriptor Queue Overflow Error.	R
31:16	—	These bits are read as 0.	R

DQOEN[15:0] bits (Descriptor Queue Overflow Error Number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment conditions]

HW: Incremented by 1 when a Descriptor Queue Overflow Error happens and if this register has a value other than 16.

34.3.10.13 GWDQSECN : Descriptor Queue Security Error Counter Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1034

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DQSEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DQSEN[15:0]	Descriptor Queue Security Error Number This register counts the number of receive data lost because of a Descriptor Queue Security Error.	R
31:16	—	These bits are read as 0.	R

DQSEN[15:0] bits (Descriptor Queue Security Error Number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment conditions]

HW: Incremented by 1 when a Descriptor Queue Security Error happens and if this register has a value other than 16.

34.3.10.14 GWDFECN : Descriptor Full Error Counter Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1038

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DFEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DFEN[15:0]	Descriptor Full Error Number This register counts the number of receive data lost because of a Descriptor Full Error.	R
31:16	—	These bits are read as 0.	R

DFEN[15:0] bits (Descriptor Full Error Number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment conditions]

HW: Incremented by 1 when a Descriptor Full Error happens and if this register has a value other than 16.

34.3.10.15 GWDSECN : Descriptor Security Error Counter Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x103C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DSEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DSEN[15:0]	Descriptor Security Error Number This register counts the number of receive data lost because of a Descriptor Security Error.	R
31:16	—	These bits are read as 0.	R

DSEN[15:0] bits (Descriptor Security Error Number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment conditions]

HW: Incremented by 1 when a Descriptor Security Error happens and if this register has a value other than 16.

34.3.10.16 GWDSZECN : Data Size Error Counter Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1040

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DSZEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DSZEN[15:0]	Data Size Error Number This register counts the number of receive data lost because of a Data Size Error.	R
31:16	—	These bits are read as 0.	R

DSZEN[15:0] bits (Data Size Error Number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment conditions]

HW: Incremented by 1 when a Data Size Error happens and if this register has a value other than 16.

34.3.10.17 GWDCTECN : Descriptor Chain Type Error Counter Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1044

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DCTEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DCTEN[15:0]	Descriptor Chain Type Error Number This register counts the number of receive data lost because of a Descriptor Chain Type Error.	R
31:16	—	These bits are read as 0.	R

DCTEN[15:0] bits (Descriptor Chain Type Error Number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment conditions]

HW: Incremented by 1 when a Descriptor Chain Type Error happens and if this register has a value different than 16.

34.3.10.18 GWRXDNECN : RX Descriptor Number Error Counter Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1048

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RXDNEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RXDNEN[15:0]	RX Descriptor Number Error Number This register counts the number of receive data lost because of a RX Descriptor Number Error.	R
31:16	—	These bits are read as 0.	R

RXDNEN[15:0] bits (RX Descriptor Number Error Number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Reading this register clears it.

[Increment conditions]

HW: Incremented by 1 when a RX Descriptor Number Error happens and if this register has a value different than 16.

34.3.11 GWCA Interrupt Registers

34.3.11.1 GWDISi : Data Interrupt Status Register i (i = 0, 1)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1100 + 0x10 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DISt	DISt	DISt	DISt	DISt	DISt	DISt	DISt	DISt	DISt	DISt	DISt	DISt	DISt	DISt	DISt
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DISt	DISt	DISt	DISt	DISt	DISt	DISt	DISt	DISt	DISt	DISt	DISt	DISt	DISt	DISt	DISt
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
b (b = 0 to 31)	DISt (t = 32 × i + b)	Data t Interrupt Status Flag (t = 0 to 63)	R/W*1

Note 1. Read value differs from written value.

DISt bits (Data t Interrupt Status Flag (t = 0 to 63))

[Setting condition TX]

HW: A descriptor has been processed in the corresponding queue and DESCR.DIE was set in the corresponding read descriptor. [About set timing information for normal TX] A descriptor is considered as processed when written back. All descriptor which are not written back are considered as processed at the timing it should have been written back so that SW can know that all the previous at the actual descriptor has been fully processed.

[Setting condition RX]

HW: A frame has been processed in the corresponding queue and DESCR.DIE was set in at least one of the corresponding read descriptors. [About set timing information for normal RX] A frame is considered as processed when frame FSTART or FSINGLE descriptor is processed. A descriptor is considered as processed when written back. All descriptor which are not written back are considered as processed at the timing it should have been written back so that SW can know that all the previous at the actual descriptor has been fully processed. All unknown descriptors are never considered as processed (See GWEIS2i.DFESt).

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

34.3.11.2 GWDIEi : Data Interrupt Enable Register i (i = 0, 1)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1104 + 0x10 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DIEt	DIEt	DIEt	DIEt	DIEt	DIEt	DIEt	DIEt	DIEt	DIEt	DIEt	DIEt	DIEt	DIEt	DIEt	DIEt
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DIEt	DIEt	DIEt	DIEt	DIEt	DIEt	DIEt	DIEt	DIEt	DIEt	DIEt	DIEt	DIEt	DIEt	DIEt	DIEt
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
b (b = 0 to 31)	DIEt (t = 32 × i + b)	Data t Interrupt Enable (t = 0 to 63) [Setting condition] Writing 1 to this bit will set it. [Clearing condition] Writing 1 to GWDIDi.DIDt register will clear this bit. 0: Interrupt disabled 1: Interrupt Enabled	R/W*1

Note 1. Read value differs from written value.

34.3.11.3 GWDIDi : Data Interrupt Disable Register i (i = 0, 1)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1108 + 0x10 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DIDt	DIDt	DIDt	DIDt	DIDt	DIDt	DIDt	DIDt	DIDt	DIDt	DIDt	DIDt	DIDt	DIDt	DIDt	DIDt
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DIDt	DIDt	DIDt	DIDt	DIDt	DIDt	DIDt	DIDt	DIDt	DIDt	DIDt	DIDt	DIDt	DIDt	DIDt	DIDt
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
b (b = 0 to 31)	DIDt (t = 32 × i + b)	Data t Interrupt Disable (t = 0 to 63) Writing 1 to this bit will clear GWDIEi.DIEt register.	R

34.3.11.4 GWDIDSi : Data Interrupt Delayed Status Register i (i = 0, 1)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x110C + 0x10 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DIDSt	DIDSt	DIDSt	DIDSt	DIDSt	DIDSt	DIDSt	DIDSt	DIDSt	DIDSt	DIDSt	DIDSt	DIDSt	DIDSt	DIDSt	DIDSt
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DIDSt	DIDSt	DIDSt	DIDSt	DIDSt	DIDSt	DIDSt	DIDSt	DIDSt	DIDSt	DIDSt	DIDSt	DIDSt	DIDSt	DIDSt	DIDSt
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
b (b = 0 to 31)	DIDSt (t = 32 × i + b)	Data t Interrupt Delayed Status Flag (t = 0 to 63) Displays the interrupt GWDISi.DISt delayed by interrupt delay function.	R

34.3.11.5 GWTSDIS : Timestamp Data Interrupt Status Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1180

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSDIS1	TSDIS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	TSDIS1 to TSDIS0	Timestamp Data i Interrupt Status Flag (The variable i corresponds to the bit position number.)	R/W*1
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

TSDISi bits (Timestamp Data i Interrupt Status Flag)

[Setting condition]

HW: A descriptor is processed in the corresponding queue and DESCR.DIE was set in the corresponding read descriptor. A descriptor is considered as processed when written back. All descriptor which are not written back are considered as processed at the timing it should have been written back so that SW can know that all the previous at the actual descriptor has been fully processed. All unknown descriptors are never considered as processed (See GWEIS0.TDFES).

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

34.3.11.6 GWTSDIE : Timestamp Data Interrupt Enable Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1184

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSDIE 1	TSDIE 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	TSDIE1 to TSDIE0	Timestamp Data i Interrupt Enable (The variable i corresponds to the bit position number.) [Setting condition] Writing 1 to this bit will set it. [Clearing condition] Writing 1 to GWTSDID.TSDIDs register will clear this bit. 0: Interrupt disabled 1: Interrupt Enabled	R/W*1
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

34.3.11.7 GWTSDID : Timestamp Data Interrupt Disable Register

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1188

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSDID1	TSDID0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	TSDID1 to TSDID0	Timestamp Data i Interrupt Disable (The variable i corresponds to the bit position number.) Writing 1 to this bit will clear GWTSDIE.TSDIEs register.	R
31:2	—	These bits are read as 0. The write value should be 0.	R

34.3.11.8 GWEIS0 : Error Interrupt Status Register 0

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1190

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	TSDNES1	TSDNES0	—	—	TDFES1	TDFES0	FSES7	FSES6	FSES5	FSES4	FSES3	FSES2	FSES1	FSES0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TSHES	TXDNES	—	SEQES	TFES	USMFSES	TSOVFES	—	—	—	—	—	—	—	—	AES
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AES	AXI Error Status Flag	R/W*1
8:1	—	These bits are read as 0. The write value should be 0.	R/W
9	TSOVFES	Timestamp Overflow Error Status Flag	R/W*1
10	USMFSES	Under Switch Minimum Frame Size Error Status Flag	R/W*1
11	TFES	TAG Filtering Error Status Flag	R/W*1
12	SEQES	Sequence Error Status Flag*2	R/W*1
13	—	This bit is read as 0. The write value should be 0.	R/W
14	TXDNES	TX Descriptor Number Error Status Flag*2	R/W*1
15	TSHES	Timestamp Hardware Error Status Flag	R/W*1
23:16	FSES7 to FSES0	Frame Size Error Status Flag i (The variable i corresponds to the bit position number.)	R/W*1
25:24	TDFES1 to TDFES0	Timestamp Descriptor i Full Error Status Flag (The variable i corresponds to the bit position number.)	R/W*1
27:26	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
29:28	TSDNES1 to TSDNES0	Timestamp Descriptor i Number Error Status Flag (The variable i corresponds to the bit position number.)	R/W*1
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.
Note 2. HW: This flag is only available for TX descriptor queues.

AES bit (AXI Error Status Flag)

[Setting condition]

HW: When an error is detected on an AXI access (xRESP != 00)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

[Error recovery]

HW: For reception, if an AXI error happens during descriptor read before a frame/timestamp transfer started, the frame/timestamp is rejected.

HW: For reception, if an AXI error happens during descriptor read after a frame transfer started, the remaining part of the frame is rejected and DESCR.AXIE is set in the frame corresponding first written back data descriptor, which is FSTART in normal synchronization mode (for GWDCCi.SM == 00).

HW: For reception, if an AXI error happens during data write, DESCR.AXIE is set in the frame corresponding first written back data descriptor, which a FSTART or a FSINGLE descriptor in normal synchronization mode (for GWDCCi.SM == 00).

HW: For reception, if an AXI error happens during RX/TS descriptor write, nothing happens.

HW: For transmission, if an AXI error happens during descriptor read and no frame is ongoing, corresponding GWTRCi.TSRj bit is cleared and no frame is transmitted.

HW: For transmission, if an AXI error happens during descriptor read and a frame is ongoing, the ongoing frame is padded with 0, sent to MFWD with AXE error set ((3) SAEF format) and corresponding GWTRCi.TSRj bit is cleared.

HW: For transmission, if an AXI error happens during data read, the ongoing frame is sent to MFWD with AXE error set ((3) SAEF format).

HW: For transmission, if an AXI error happens during descriptor write, nothing happens.

SW: System reset.

TSOVFES bit (Timestamp Overflow Error Status Flag)

[Setting condition]

HW: When a timestamp is received when the timestamp RAM is full.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

[Error recovery]

HW: A timestamp is lost.

SW: Nothing if it is acceptable to lose a timestamp else switch reset.

USMFSES bit (Under Switch Minimum Frame Size Error Status Flag)

[Setting condition]

HW: A frame smaller than 32 bytes has been received from CPU for transmission.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: The frame is discarded.

SW: Software error, software should be reviewed.

TFES bit (TAG Filtering Error Status Flag)

[Setting condition]

HW: An unauthorized TAG format has been detected. (See section 34.5.2.3.2. VLAN Tagging).

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Frame is sent to MFWD with local descriptor DESCR.TFE bit set (See section 34.5.2.3.3. Frame Saving in Local RAM).

SW: Software error, software should be reviewed.

SEQES bit (Sequence Error Status Flag)

[Setting condition]

HW: A FMID, FEND has been received before a frame transfer started.

HW: A descriptor different than FMID, FEND, LINK and LINKFIX has been received after a frame transfer started.

HW: An FSTART, FMID, FEND or FSINGLE descriptor has been received with DESCR.DS set to 0.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: If a SEQuence Error happens before a frame transfer started, nothing happens.

HW: If a SEQuence Error happens after a frame transfer started, The ongoing frame is padded with 0 and sent to MFWD with SEQE error set ((3) SAEF format).

SW: Software error, software should be reviewed.

TXDNES bit (TX Descriptor Number Error Status Flag)

[Setting condition]

HW: GWMDNC.TXDMN+1 descriptor has already been used to process one frame and the frame processing is not finished.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: If a TX Descriptor Number Error happens before a frame transfer started, GWTRCi.TSRj bit is cleared.

HW: If an TX Descriptor Number Error happens after a frame transfer started, the ongoing frame is padded with 0, sent to MFWD with DNE error set ((3) SAEF format) and corresponding GWTRCi.TSRj bit is cleared.

SW: Software error, software should be reviewed.

TSHES bit (Timestamp Hardware Error Status Flag)

[Setting condition]

HW: This error happens when timestamps are coming too fast to GWCA to be stored in time, so timestamp are lost even if the timestamp RAM is not full. This error should never happen and tells that the number of captured timestamps is too high or the bus clock clk has a too low frequency.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: timestamp lost.

SW: No recovery, this error should not happen.

FSESi bits (Frame Size Error Status Flag i)

[Setting condition]

HW: Bit q of this register is set when a frame bigger than GWRMFSC.MFS has been received for descriptor queue q.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: The current frame will be lost. Following data will be processed normally.

SW: System dependent, cannot be defined here.

TDFESi bits (Timestamp Descriptor i Full Error Status Flag)

[Setting condition]

HW: Bit i sets when a timestamp has been received for descriptor chain i which is full (read any other descriptor than a FEMPTY_ND, LINKFIX or LINK descriptor).

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Timestamp lost.

SW: Nothing if it is acceptable to lose a timestamp else switch reset.

SW: AXI is too slow to save all timestamps in the CPU RAM. Less timestamps should be captured or more bandwidth on AXI should be reserved for ESWM.

TSDNESi bits (Timestamp Descriptor i Number Error Status Flag)

[Setting condition]

HW: Bit i sets when GWMDNC.TSDMN+1 descriptor has already been used to process one timestamp for descriptor chain i and the timestamp processing is not finished.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: Timestamp lost.

SW: Software error, software should be reviewed.

34.3.11.9 GWEIE0 : Error Interrupt Enable Register 0

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1194

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	TSDNEE1	TSDNEE0	—	—	TDFEE1	TDFEE0	FSEE7	FSEE6	FSEE5	FSEE4	FSEE3	FSEE2	FSEE1	FSEE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TSHEE	TXDNEE	—	SEQEE	TFEE	USMFSEE	TSOVFEE	—	—	—	—	—	—	—	—	AEE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AEE	AXI Error Enable [Setting condition] Writing 1 to this bit will set it. [Clearing condition] Writing 1 to GWEID0.AED register will clear this bit. 0: Interrupt disabled 1: Interrupt Enabled	R/W*1
8:1	—	These bits are read as 0. The write value should be 0.	R/W
9	TSOVFEE	Timestamp Overflow Error Enable [Setting condition] Writing 1 to this bit will set it. [Clearing condition] Writing 1 to GWEID0.TSOVFED register will clear this bit. 0: Interrupt disabled 1: Interrupt Enabled	R/W*1
10	USMFSEE	Under Minimum Frame Size Error Enable [Setting condition] Writing 1 to this bit will set it. [Clearing condition] Writing 1 to GWEID0.USMFSED register will clear this bit. 0: Interrupt disabled 1: Interrupt Enabled	R/W*1
11	TFEE	TAG Filtering Error Enable [Setting condition] Writing 1 to this bit will set it. [Clearing condition] Writing 1 to GWEID0.TFED register will clear this bit. 0: Interrupt disabled 1: Interrupt Enabled	R/W*1
12	SEQEE	Sequence Error Enable [Setting condition] Writing 1 to this bit will set it. [Clearing condition] Writing 1 to GWEID0.SEQED register will clear this bit. 0: Interrupt disabled 1: Interrupt Enabled	R/W*1
13	—	This bit is read as 0. The write value should be 0.	R/W
14	TXDNEE	TX Descriptor Number Error Enable [Setting condition] Writing 1 to one of these bits will set it. [Clearing condition] Writing 1 to GWEID0.TXDNED register will clear this bit. 0: Interrupt disabled 1: Interrupt Enabled	R/W*1

Bit	Symbol	Function	R/W
15	TSHEE	Timestamp Hardware Error Enable [Setting condition] Writing 1 to one of these bits will set it. [Clearing condition] Writing 1 to GWEID0.TSHED register will clear this bit. 0: Interrupt disabled 1: Interrupt Enabled	R/W*1
23:16	FSEE7 to FSEE0	Frame Size Error Enable i (The variable i corresponds to the bit position number.) [Setting condition] Writing 1 to one of these bits will set it. [Clearing condition] Writing 1 to one of GWEID0.FSED bits will clear the corresponding bit in this register. 0: Interrupt disabled for error q 1: Interrupt enabled for error q	R/W*1
25:24	TDFEE1 to TDFEE0	Timestamp Descriptor i Full Error Enable (The variable i corresponds to the bit position number.) [Setting condition] Writing 1 to one of these bits will set it. [Clearing condition] Writing 1 to one of GWEID0.TDFEID bits will clear the corresponding bit in this register. 0: Interrupt disabled for descriptor queue i 1: Interrupt enabled for descriptor queue i	R/W*1
27:26	—	These bits are read as 0. The write value should be 0.	R/W
29:28	TSDNEE1 to TSDNEE0	Timestamp Descriptor i Number Error Enable (The variable i corresponds to the bit position number.) [Setting condition] Writing 1 to one of these bits will set it. [Clearing condition] Writing 1 to one of GWEID0.TSDNED bits will clear the corresponding bit in this register. 0: Interrupt disabled 1: Interrupt enabled	R/W*1
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

34.3.11.10 GWEID0 : Error Interrupt Disable Register 0

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1198

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	TSDNED1	TSDNED0	—	—	TDFED1	TDFED0	FSED7	FSED6	FSED5	FSED4	FSED3	FSED2	FSED1	FSED0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TSHED	TXDNED	—	SEQED	TFED	USMFSED	TSOVFED	—	—	—	—	—	—	—	—	AED
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AED	AXI Error Disable Writing 1 to this bit will clear GWEIE0.AEE register.	R
8:1	—	These bits are read as 0.	R
9	TSOVFED	Timestamp Overflow Error Disable Writing 1 to this bit will clear GWEIE0.TSOVFEE register.	R
10	USMFSED	Under Minimum Frame Size Error Disable Writing 1 to this bit will clear GWEIE0.USMFSEE register.	R

Bit	Symbol	Function	R/W
11	TFED	TAG Filtering Error Disable Writing 1 to this bit will clear GWEIE0.TFEE register.	R
12	SEQED	Sequence Error Disable Writing 1 to this bit will clear GWEIE0.SEQEE register.	R
13	—	This bit is read as 0.	R
14	TXDNED	TX Descriptor Number Error Disable Writing 1 to this bit will clear GWEIE0.TXDNEE register.	R
15	TSHED	Timestamp Hardware Full Error Disable Writing 1 to this bit will clear GWEIE0.TSHEE register.	R
23:16	FSED7 to FSED0	Frame Size Error Disable i (The variable i corresponds to the bit position number.) Writing 1 to one of these bits will clear the corresponding bit in GWEIE0.FSEE register.	R
25:24	TDFED1 to TDFED0	Timestamp Descriptor i Full Error Disable (The variable i corresponds to the bit position number.) Writing 1 to one of these bits will clear the corresponding bit in GWEIE0.TDFEIE register.	R
27:26	—	These bits are read as 0.	R
29:28	TSDNED1 to TSDNED0	Timestamp Descriptor i Number Error Disable (The variable i corresponds to the bit position number.) Writing 1 to one of these bits will clear the corresponding bit in GWEIE0.TSDNED register.	R
31:30	—	These bits are read as 0.	R

34.3.11.11 GWEIS1 : Error Interrupt Status Register 1

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x11A0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	DQSES7	DQSES6	DQSES5	DQSES4	DQSES3	DQSES2	DQSES1	DQSES0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	DQOES7	DQOES6	DQOES5	DQOES4	DQOES3	DQOES2	DQOES1	DQOES0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	DQOES7 to DQOES0	Descriptor Queue i Overflow Error Status Flag (The variable i corresponds to the bit position number.)	R/W*1
15:8	—	These bits are read as 0. The write value should be 0.	R/W
23:16	DQSES7 to DQSES0	Descriptor Queue i Security Error Status Flag (The variable i corresponds to the bit position number.)	R/W*1
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

DQOESi bits (Descriptor Queue i Overflow Error Status Flag)

[Setting condition]

HW: Bit q of this register is set when a descriptor is received for queue q when it is full (GWRDQDCq.DQD == GWRDQMq.DNQ), not disabled (GWRDQC.RDQDi is not set) and GWCA is not going out or out of OPERATION mode (GWMC.OPC == 11).

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

[Error recovery]

HW: Any descriptor received for a full Descriptor queue will not be accepted by GWCA and will not be forwarded to CPU.

SW: Too many descriptors are received for the corresponding queue. GWRDQDCq.DQD setting should be reviewed.

DQSESi bits (Descriptor Queue i Security Error Status Flag)

[Setting condition]

HW: Bit q of this register is set when a non-secure descriptor is received (FDESCR.SEC is not set [FWD]) and queue q is a secure queue (GWRDQSC.RDQSLn is set)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

[Error recovery]

HW: Any non-secure descriptor received for secure Descriptor queue will not be accepted by GWCA and will not be forwarded to CPU.

SW: System dependent, cannot be defined here.

34.3.11.12 GWEIE1 : Error Interrupt Enable Register 1

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x11A4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	DQSEE7	DQSEE6	DQSEE5	DQSEE4	DQSEE3	DQSEE2	DQSEE1	DQSEE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	DQOEE7	DQOEE6	DQOEE5	DQOEE4	DQOEE3	DQOEE2	DQOEE1	DQOEE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	DQOEE7 to DQOEE0	Descriptor Queue i Overflow Error Enable (The variable i corresponds to the bit position number.) [Setting condition] Writing 1 to one of these bits will set it. [Clearing condition] Writing 1 to bit q in GWEID1.DQOED register will clear the bit q in this register. 0: Interrupt disabled for error q 1: Interrupt enabled for error q	R/W*1
15:8	—	These bits are read as 0. The write value should be 0.	R/W
23:16	DQSEE7 to DQSEE0	Descriptor Queue i Security Error Enable (The variable i corresponds to the bit position number.) [Setting condition] Writing 1 to one of these bits will set it. [Clearing condition] Writing 1 to bit q in GWEID1.DQSED register will clear the bit q in this register. 0: Interrupt disabled for error q 1: Interrupt enabled for error q	R/W*1
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

34.3.11.13 GWEID1 : Error Interrupt Disable Register 1

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x11A8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	DQSED7	DQSED6	DQSED5	DQSED4	DQSED3	DQSED2	DQSED1	DQSED0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	DQOED7	DQOED6	DQOED5	DQOED4	DQOED3	DQOED2	DQOED1	DQOED0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	DQOED7 to DQOED0	Descriptor Queue i Overflow Error Disable (The variable i corresponds to the bit position number.) Writing 1 to bit q in this register will clear bit q in GWEIE1.DQOEE register.	R
15:8	—	These bits are read as 0.	R
23:16	DQSED7 to DQSED0	Descriptor Queue i Security Error Disable (The variable i corresponds to the bit position number.) Writing 1 to bit q in this register will clear bit q in GWEIE1.DQSEE register.	R
31:24	—	These bits are read as 0.	R

34.3.11.14 GWEIS2i : Error Interrupt Status Register 2i (i = 0, 1)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1200 + 0x10 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DFESt	DFESt	DFESt	DFESt	DFESt	DFESt	DFESt	DFESt	DFESt	DFESt	DFESt	DFESt	DFESt	DFESt	DFESt	DFESt
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DFESt	DFESt	DFESt	DFESt	DFESt	DFESt	DFESt	DFESt	DFESt	DFESt	DFESt	DFESt	DFESt	DFESt	DFESt	DFESt
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
b (b = 0 to 31)	DFESt (t = 32 × i + b)	Descriptor t Full Error Status Flag (t = 0 to 63)*1	R/W*2

Note 1. HW: This register is only valid for RX queues.
Note 2. Read value differs from written value.

DFESt bits (Descriptor t Full Error Status Flag (t = 0 to 63))

[Setting condition]

HW: When a frame has been received for descriptor queue t which is full (read a descriptor other than FEMPTY, FEMPTY_IS, FEMPTY_IC, FEMPTY_ND, FEMPTY_START, FEMPTY_MID, FEMPTY_END, LINKFIX and LINK). This flag is set at the timing the frame last descriptor write-back should have been completed.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

[Error recovery]

HW: set of the DESCR.ERR bit in RX descriptor if a part of the frame has been already sent.

HW: Current frame or remaining part of the frame will be discarded, and next frames will be processed normally (as long as software does not insert new descriptors for reception the queue will continue overflowing).

SW: Discard data which have DESCR.ERR bit set.

34.3.11.15 GWEIE2i : Error Interrupt Enable Register 2i (i = 0, 1)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1204 + 0x10 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DFEEt	DFEEt	DFEEt	DFEEt	DFEEt	DFEEt	DFEEt	DFEEt	DFEEt	DFEEt	DFEEt	DFEEt	DFEEt	DFEEt	DFEEt	DFEEt
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DFEEt	DFEEt	DFEEt	DFEEt	DFEEt	DFEEt	DFEEt	DFEEt	DFEEt	DFEEt	DFEEt	DFEEt	DFEEt	DFEEt	DFEEt	DFEEt
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
b (b = 0 to 31)	DFEEt (t = 32 × i + b)	Descriptor t Full Error Enable (t = 0 to 63) 0: Interrupt disabled 1: Interrupt Enabled	R/W*1

Note 1. Read value differs from written value.

DFEEt bits (Descriptor t Full Error Enable (t = 0 to 63))

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to GWEID2i.DFEDt register will clear this bit.

34.3.11.16 GWEID2i : Error Interrupt Disable Register 2i (i = 0, 1)

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1208 + 0x10 × i

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DFEDt	DFEDt	DFEDt	DFEDt	DFEDt	DFEDt	DFEDt	DFEDt	DFEDt	DFEDt	DFEDt	DFEDt	DFEDt	DFEDt	DFEDt	DFEDt
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DFEDt	DFEDt	DFEDt	DFEDt	DFEDt	DFEDt	DFEDt	DFEDt	DFEDt	DFEDt	DFEDt	DFEDt	DFEDt	DFEDt	DFEDt	DFEDt
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
b (b = 0 to 31)	DFEDt (t = 32 × i + b)	Descriptor t Full Error Disable (t = 0 to 63) Writing 1 to this bit will clear GWEIE2i.DFEEt register.	R

34.3.11.17 GWEIS3 : Error Interrupt Status Register 3

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1280

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	IAOES3	IAOES2	IAOES1	IAOES0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IAOES3 to IAOES0	Incremental Area i Overflow Error Status Flag*1 (The variable i corresponds to the bit position number.)	R/W*2
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. HW: This register is only valid for RX incremental queues.
Note 2. Read value differs from written value.

IAOESi bits (Incremental Area i Overflow Error Status Flag)

[Setting condition]

HW: When incremental area i overflows, for further information, see section 34.5.3.5.3. Single Page Incremental Data Reception. This flag is only set when GWIDAUASi.IDAUAS crosses GWIDASMi.IDAS value (if incremental area is already overflowing, this flag will not be set again except if GWIDAUASi.IDAUAS overflows).

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

[Error recovery]

HW: The data coming for this incremental area will still be written.

SW: Set a new incremental area.

34.3.11.18 GWEIE3 : Error Interrupt Enable Register 3

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1284

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	IAOEE3	IAOEE2	IAOEE1	IAOEE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IAOEE3 to IAOEE0	Incremental Area i Overflow Error Enable (The variable i corresponds to the bit position number.) 0: Interrupt disabled 1: Interrupt Enabled	R/W*1
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

IAOEEi bits (Incremental Area i Overflow Error Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to GWEID3.IAOEDi register will clear this bit.

34.3.11.19 GWEID3 : Error Interrupt Disable Register 3

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1288

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	IAOED3	IAOED2	IAOED1	IAOED0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IAOED3 to IAOED0	Incremental Area i Overflow Error Disable (The variable i corresponds to the bit position number.) Writing 1 to this bit will clear GWEIE3.IAOEDi register.	R
31:4	—	These bits are read as 0.	R

34.3.11.20 GWEIS4 : Error Interrupt Status Register 4

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1290

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	DSECN[5:0]						—	—	—	—	—	—	DSEIOS	DSES
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	DSSECN[5:0]						—	—	—	—	—	—	DSSEIOS	DSSES
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DSSES	Descriptor Security Error Status Flag	R/W*1
1	DSSEIOS	Descriptor Security Error Interrupt Overflow Status Flag*2	R/W*1
7:2	—	These bits are read as 0. The write value should be 0.	R/W
13:8	DSSECN[5:0]	Descriptor Security Error Chain Number*2	R
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	DSES	Data Size Error Status Flag*2	R/W*1
17	DSEIOS	Data Size Error Interrupt Overflow Status Flag*2	R/W*1
23:18	—	These bits are read as 0. The write value should be 0.	R/W
29:24	DSECN[5:0]	Data Size Error Chain Number*2	R
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.
Note 2. HW: This register is only valid for RX queues.

DSSES bit (Descriptor Security Error Status Flag)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

[Setting condition]

HW: During reception, when an unsecure descriptor has been received for a secure queue (GWDCCi.SL is set).

[Error recovery]

HW: During reception, the current frame will be lost. Following data will be processed normally.

SW: Secure CPU should read the unsecure forwarding engine registers to find which entry is trying to use a secure queue for unsecure traffic and suppress it.

SW: Secure CPU could block access to switch from unsecure CPU.

DSSEIOS bit (Descriptor Security Error Interrupt Overflow Status Flag)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

[Setting condition]

When a Descriptor Security Error occurs and GWEIS4.DSSES is already set to one. In this case, the descriptor chain number in which the new error occurred is lost.

[Error recovery]

SW: Switch reset.

DSSECN[5:0] bits (Descriptor Security Error Chain Number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

[Update conditions]

This field is update to the descriptor chain number in which a Descriptor Security Error occurred when GWEIS4.DSSES is being set.

DSES bit (Data Size Error Status Flag)

[Setting condition]

HW: When a frame has been received with an unexpected size, for more information see section 34.5.3.5.2. Size-controlled Data Reception. This flag is set after the frame last descriptor write back is completed.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

[Error recovery]

HW: set of DESCR.DSE bit in FSTART RX descriptor.

HW: If the frame was an undersized frame the frame is written with the normal descriptor sequence (FSINGLE or FSTART/FEND or FSTART/n*FMID/ FEND)

HW: If the frame was an undersized frame, when a new frame comes for the corresponding queue, the AXI master will skip all following descriptors until an empty descriptor which is not FEMPTY_MID or FEMPTY_END (at start of a frame, FEMPTY_MID and FEMPTY_END descriptors are always ignored).

If the frame is an oversized frame, the frame will be truncated.

SW: Discard data which have DESCR.DSE bit set.

SW: Because DESCR.DSE is set in RX descriptors with a size error, it is not mandatory to read this flag for error recovery.

DSEIOS bit (Data Size Error Interrupt Overflow Status Flag)

[Setting condition]

When a Data Size Error occurs and GWEIS4.DSES is already set to one. In this case, the descriptor chain number in which the new error occurred is lost.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

[Error recovery]

SW: Switch reset.

DSECN[5:0] bits (Data Size Error Chain Number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

[Update conditions]

This field is update to the descriptor chain number in which a Data Size Error occurred when GWEIS4.DSEIS is being set.

34.3.11.21 GWEIE4 : Error Interrupt Enable Register 4

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1294

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DSEIOE	DSEE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DSSEIOE	DSSEE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DSSEE	Descriptor Security Error Enable 0: Interrupt disabled 1: Interrupt Enabled	R/W*1
1	DSSEIOE	Descriptor Security Error Interrupt Overflow Enable 0: Interrupt disabled 1: Interrupt Enabled	R/W*1
15:2	—	These bits are read as 0. The write value should be 0.	R/W
16	DSEE	Data Size Error Enable 0: Interrupt disabled 1: Interrupt Enabled	R/W*1
17	DSEIOE	Data Size Error Interrupt Overflow Interrupt Enable 0: Interrupt disabled 1: Interrupt Enabled	R/W*1
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

DSSEE bit (Descriptor Security Error Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to GWEID4.DSSED register will clear this bit.

DSSEIOE bit (Descriptor Security Error Interrupt Overflow Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to GWEID4.DSSEIOD register will clear this bit.

DSEE bit (Data Size Error Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to GWEID4.DSED register will clear this bit.

DSEIOE bit (Data Size Error Interrupt Overflow Interrupt Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to GWEID4.DSEIOD register will clear this bit.

34.3.11.22 GWEID4 : Error Interrupt Disable Register 4

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x1298

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DSEIOD	DSED
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DSSEIOD	DSSED
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DSSED	Descriptor Security Error Disable Writing 1 to this bit will clear GWEIE4.DSSEE register.	R
1	DSSEIOD	Descriptor Security Error Interrupt Overflow Disable Writing 1 to this bit will clear GWEIE4.DSSEIOE register.	R
15:2	—	These bits are read as 0.	R
16	DSED	Data Size Error Disable Writing 1 to this bit will clear GWEIE4.DSEE register.	R
17	DSEIOD	Data Size Error Interrupt Overflow Disable Writing 1 to this bit will clear GWEIE4.DSEIOE register.	R
31:18	—	These bits are read as 0.	R

34.3.11.23 GWEIS5 : Error Interrupt Status Register 5

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x12A0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	RXDNECN[5:0]						—	—	—	—	—	—	RXDNEIOS	RXDNES
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	DCTECN[5:0]						—	—	—	—	—	—	DCTEIOS	DCTES
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DCTES	Descriptor Chain Type Error Status Flag	R/W*1
1	DCTEIOS	Descriptor Chain Type Error Interrupt Overflow Status Flag	R/W*1
7:2	—	These bits are read as 0. The write value should be 0.	R/W
13:8	DCTECN[5:0]	Descriptor Chain Type Error Chain Number	R
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
16	RXDNES	RX Descriptor Number Error Status Flag	R/W*1
17	RXDNEIOS	RX Descriptor Number Error Interrupt Overflow Status Flag	R/W*1
23:18	—	These bits are read as 0. The write value should be 0.	R/W
29:24	RXDNECN[5:0]	RX Descriptor Number Error Chain Number	R
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

DCTES bit (Descriptor Chain Type Error Status Flag)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

[Setting condition]

HW: When a descriptor is received for a transmission descriptor queue (GWDCCi.DQT is set).

[Error recovery]

HW: When a descriptor is received for a transmission descriptor queue, the corresponding frame is discarded.

SW: It is a software error. Software should be reviewed.

DCTEIOS bit (Descriptor Chain Type Error Interrupt Overflow Status Flag)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

[Setting condition]

Descriptor Chain Type Error occurs and GWEIS5.DCTES is already set to one. In this case, the descriptor chain number in which the new error occurred is lost.

[Error recovery]

SW: Switch reset.

DCTECN[5:0] bits (Descriptor Chain Type Error Chain Number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

[Update conditions]

This field is update to the descriptor chain number in which a Descriptor Chain Type Error occurred when GWEIS5.DCTES is being set.

RXDNES bit (RX Descriptor Number Error Status Flag)

[Setting condition]

HW: GWMDNC.RXDMN+1 descriptor has already been used to process one frame and the frame processing is not finished.

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to one of these bits will clear it.

[Error recovery]

HW: set of the DESCR.ERR bit in RX descriptor if a part of the frame has been already sent.

HW: Current frame or remaining part of the frame will be discarded, and next frames will be processed normally.

SW: Discard data which have DESCR.ERR bit set.

SW: Software error, software should be reviewed.

RXDNEIOS bit (RX Descriptor Number Error Interrupt Overflow Status Flag)

[Clearing condition]

HW: Being in RESET mode will clear this register.

SW: Writing 1 to this bit will clear it.

[Setting condition]

RX Descriptor Number Error occurs and GWEIS5.RXDNES is already set to one. In this case, the descriptor chain number in which the new error occurred is lost.

[Error recovery]

SW: Switch reset.

RXDNECN[5:0] bits (RX Descriptor Number Error Chain Number)

[Clearing condition]

HW: Being in RESET mode will clear this register.

[Update conditions]

This field is update to the descriptor chain number in which a RX Descriptor Number Error occurred when GWEIS5.RXDNES is being set.

34.3.11.24 GWEIE5 : Error Interrupt Enable Register 5

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x12A4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RXDNEIOE	RXDNEE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DCTEIOE	DCTEE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DCTEE	Descriptor Chain Type Error Enable 0: Interrupt disabled 1: Interrupt Enabled	R/W*1
1	DCTEIOE	Descriptor Chain Type Error Interrupt Overflow Enable 0: Interrupt disabled 1: Interrupt Enabled	R/W*1
15:2	—	These bits are read as 0. The write value should be 0.	R/W
16	RXDNEE	RX Descriptor Number Error Enable 0: Interrupt disabled 1: Interrupt Enabled	R/W*1
17	RXDNEIOE	RX Descriptor Number Error Interrupt Overflow Enable 0: Interrupt disabled 1: Interrupt Enabled	R/W*1
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

DCTEE bit (Descriptor Chain Type Error Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to GWEID5.DCTED register will clear this bit.

DCTEIOE bit (Descriptor Chain Type Error Interrupt Overflow Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to GWEID5.DCTEIOD register will clear this bit.

RXDNEE bit (RX Descriptor Number Error Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to GWEID5.RXDNED register will clear this bit.

RXDNEIOE bit (RX Descriptor Number Error Interrupt Overflow Enable)

[Setting condition]

Writing 1 to this bit will set it.

[Clearing condition]

Writing 1 to GWEID5.RXDNEIOD register will clear this bit.

34.3.11.25 GWEID5 : Error Interrupt Disable Register 5

Base address: GWCA0 = 0x403C_E000
GWCA0_NS = 0x503C_E000

Offset address: 0x12A8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RXDNEIOD	RXDNED
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DCTEIOD	DCTED
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DCTED	Descriptor Chain Type Error Disable Writing 1 to this bit will clear GWEIE5.DCTEE register.	R
1	DCTEIOD	Descriptor Chain Type Error Interrupt Overflow Disable Writing 1 to this bit will clear GWEIE5.DCTEIOE register.	R
15:2	—	These bits are read as 0.	R
16	RXDNED	RX Descriptor Number Error Disable Writing 1 to this bit will clear GWEIE5.RXDNEE register.	R
17	RXDNEIOD	RX Descriptor Number Error Interrupt Overflow Disable Writing 1 to this bit will clear GWEIE5.RXDNEIOE register.	R
31:18	—	These bits are read as 0.	R

34.4 Register Utilization

34.4.1 Operation Modes

Table 34.4 describes GWCA operation modes.

Table 34.4 GWCA operation modes

Operation mode	GWMS.OPS value	Description
DISABLE	1	• No transaction is ongoing. But there can be tolerance (left behind) of transactions*1 from GWCA when the agent has moved from OPERATION to DISABLE. • Only status registers are accessible for writing when the agent clock is enabled.
RESET	0	• No transaction is ongoing. • An internal Reset is asserted to reset GWCA logic with status registers (When a register is reset in RESET mode, it is mentioned in its description). • No register is accessible for writing. • RAM values are held.
CONFIG	2	• No transaction is ongoing. • Static and some dynamic registers are accessible for writing.
OPERATION	3	• Transactions are ongoing. • Dynamic registers are accessible for writing.

Note 1. max 8 read transactions or max 8 write transactions.

34.4.1.1 Operation Mode Transitions

Figure 34.2 shows the operating mode transitions.

A mode transition can be trigger by:

- Hardware reset.
- Software reset.
- Configuration of **GWMC.OPC**. In that case, mode transition will be confirmed by GWMS.OPS.

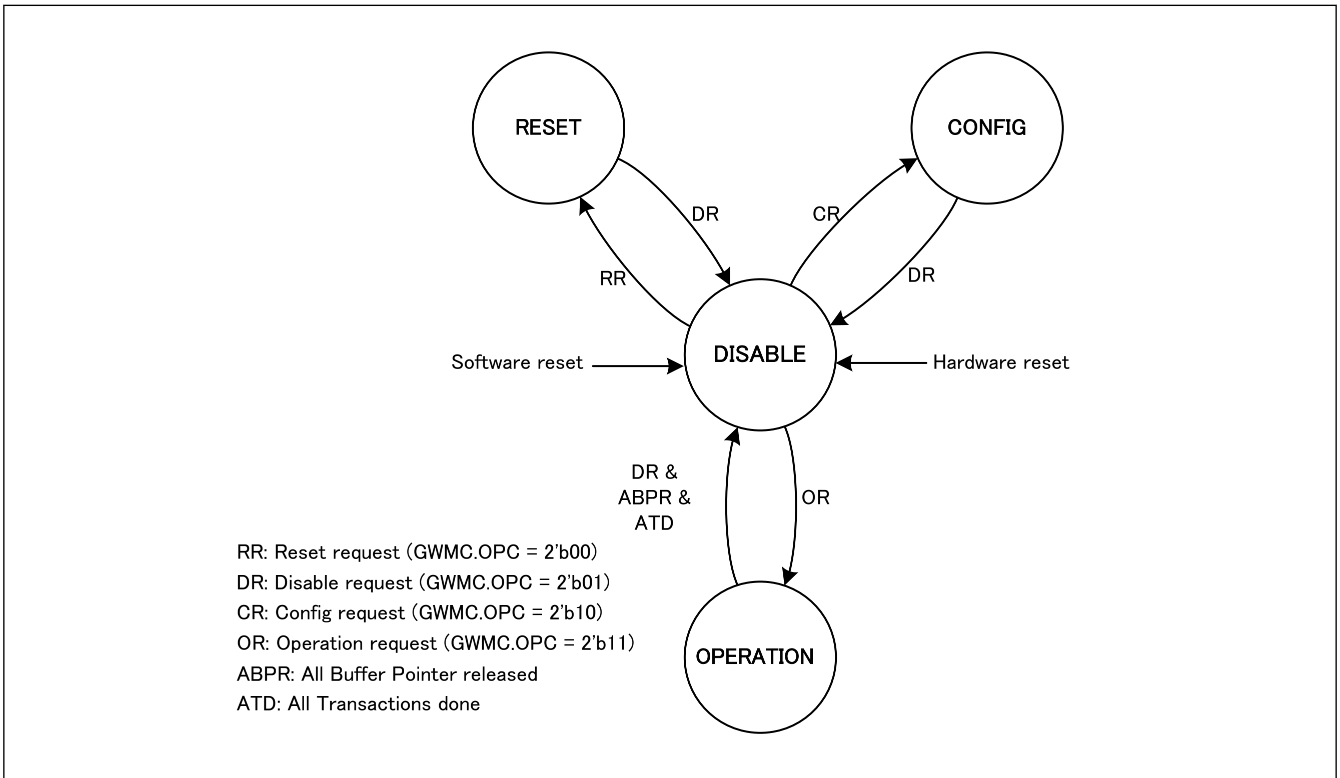


Figure 34.2 Operating mode transitions

Complementary information on transition conditions

ABPR:

- All the buffer pointers contained in the GWCA are release to the forwarding engine [FWD].

ATD:

- All the descriptors already received are processed and corresponding frames are sent to CPU and AXI descriptors are written back.

All ongoing transmit frames are fully sent to MFWD and AXI descriptors are written back.

[Restrictions]

- Software shall only trigger transitions shown to Figure 34.2.
  Exp : Do not directory transition from OPERATION to CONFIG.

34.4.2 Software Flows

Restrictions:

- SW: Please follow to the flow in this section.

34.4.2.1 Software Flows Legend

Software flow legend is described in Figure 34.3.

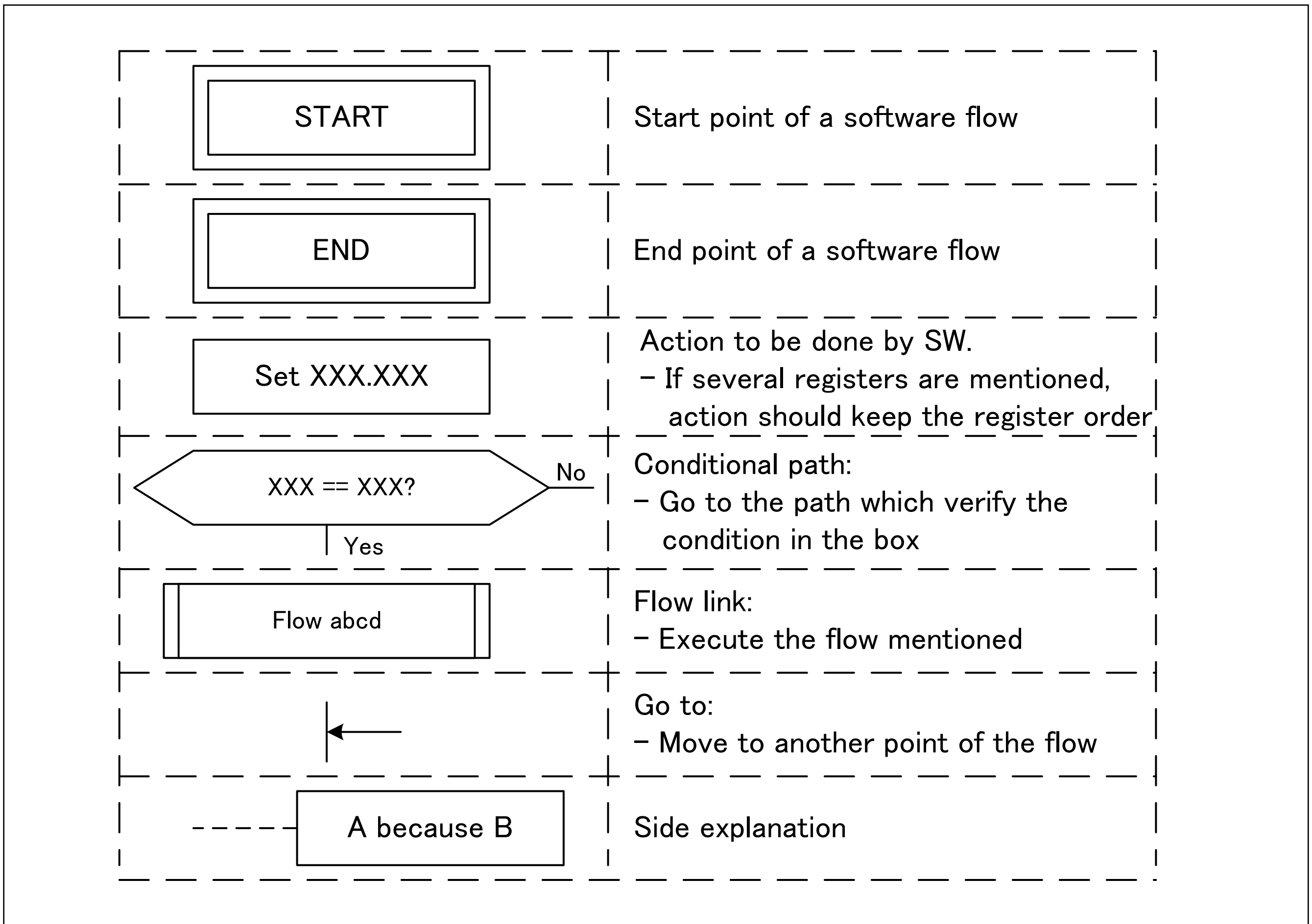


Figure 34.3 Software flow legend

34.4.2.2 Mode Transition Flow

The mode transition flow is described in Figure 34.4.

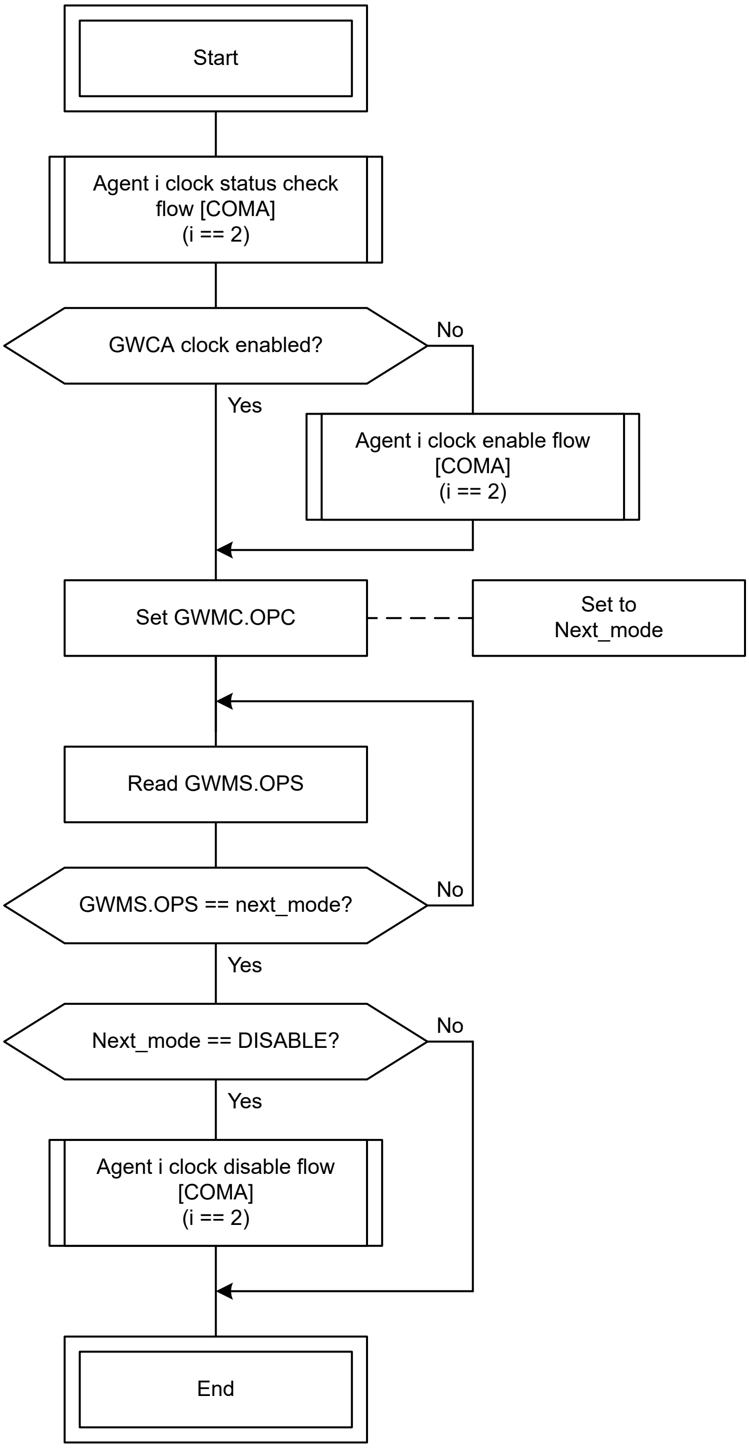


Figure 34.4 Mode transition flow

Note:
- If SW already knows that the clock is enabled, clock status check step can be skipped.
- If SW does not need to disable the agent clock in DISABLE mode (because it will go directly to another mode or because the clock never needs to be disabled), clock disable step can be skipped.

34.4.2.3 Reset Flow

The reset flow is described in Figure 34.5.

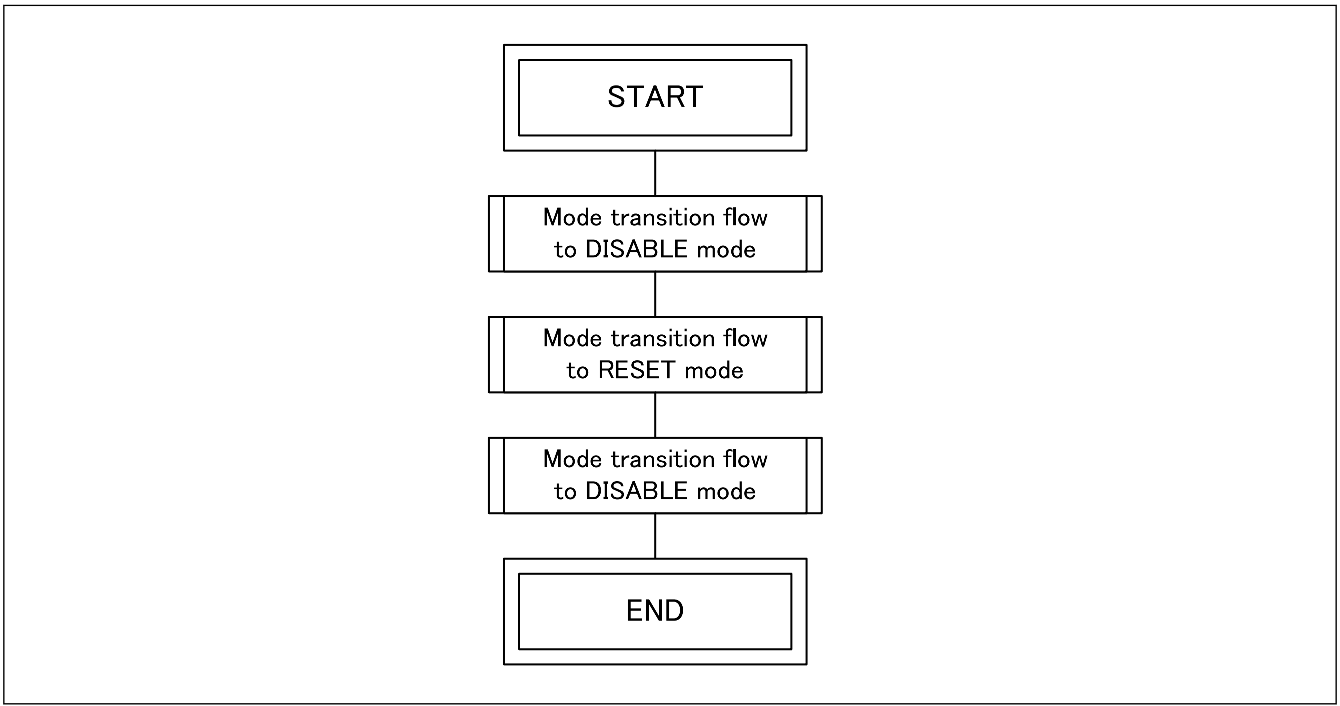


Figure 34.5 Reset flow

34.4.2.4 Initialization Flow

The initialization flow is described in Figure 34.6.

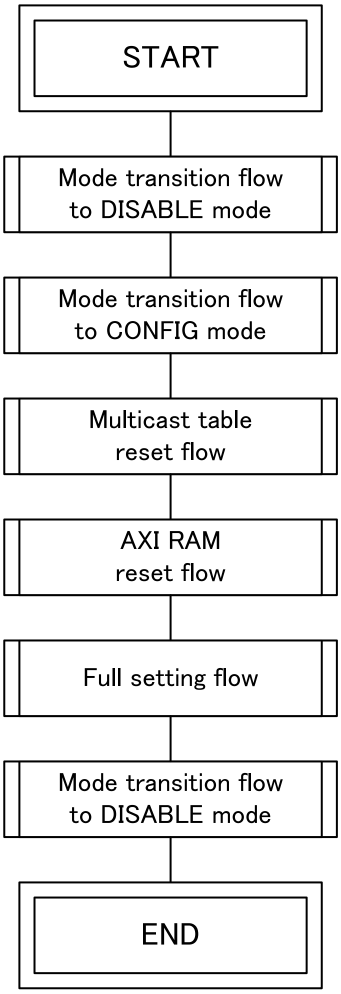


Figure 34.6 Initialization flow

34.4.2.5 Reinitialization Flow

The reinitialization flow is described in Figure 34.7.

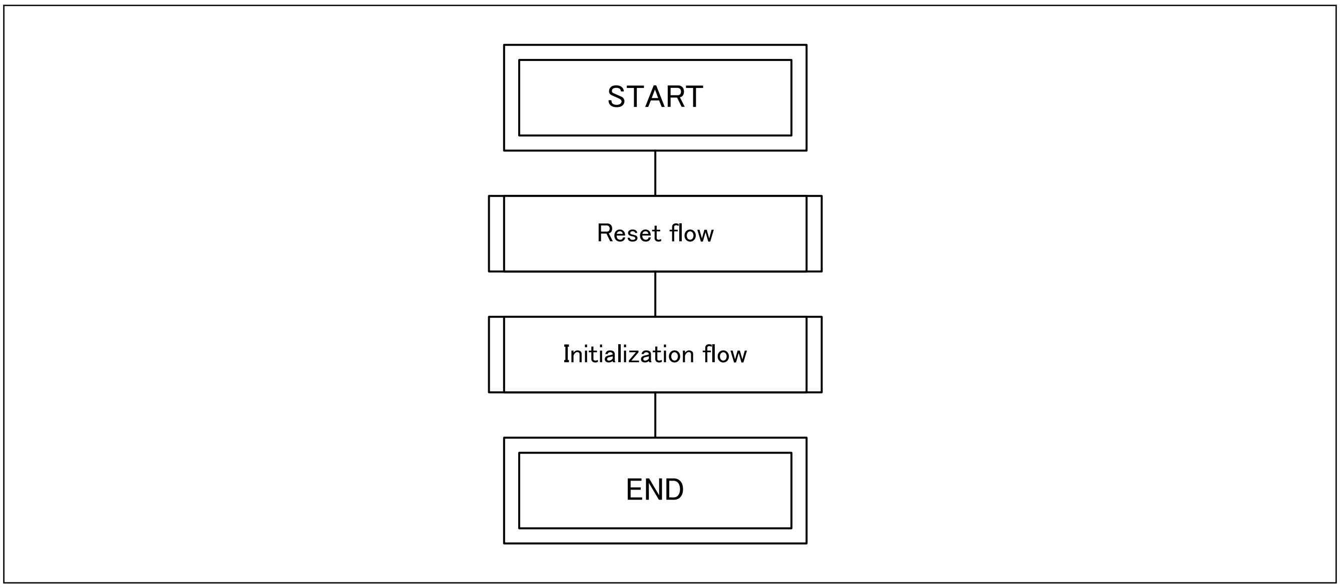


Figure 34.7 Reinitialization flow

34.4.2.6 Multicast Table Reset Flow

The multicast table reset flow is described in Figure 34.8.

Restrictions:

- This flow is not usable in RESET and DISABLE modes.

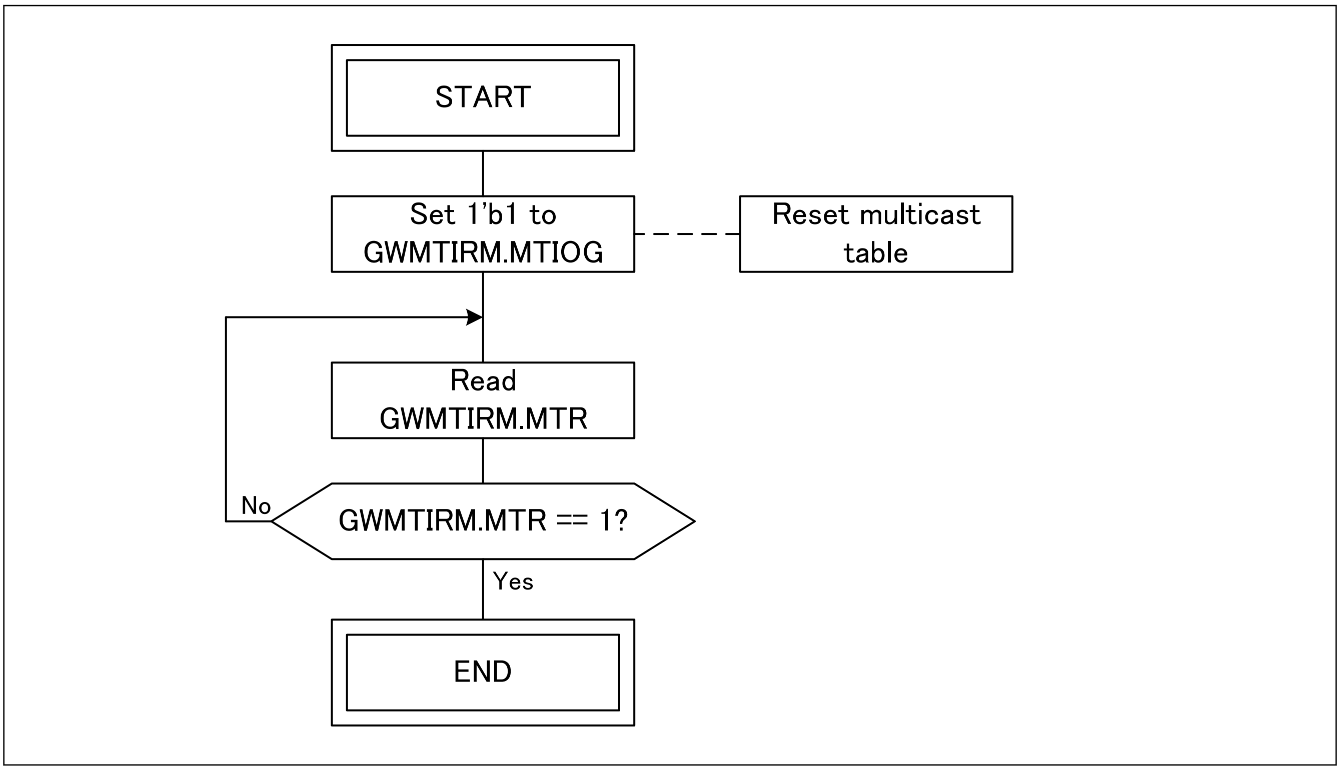


Figure 34.8 Multicast table reset flow

34.4.2.7 Multicast Table Setting Flow

The multicast table setting flow is described in Figure 34.9.

Restrictions:

- This flow is not usable in RESET and DISABLE modes.

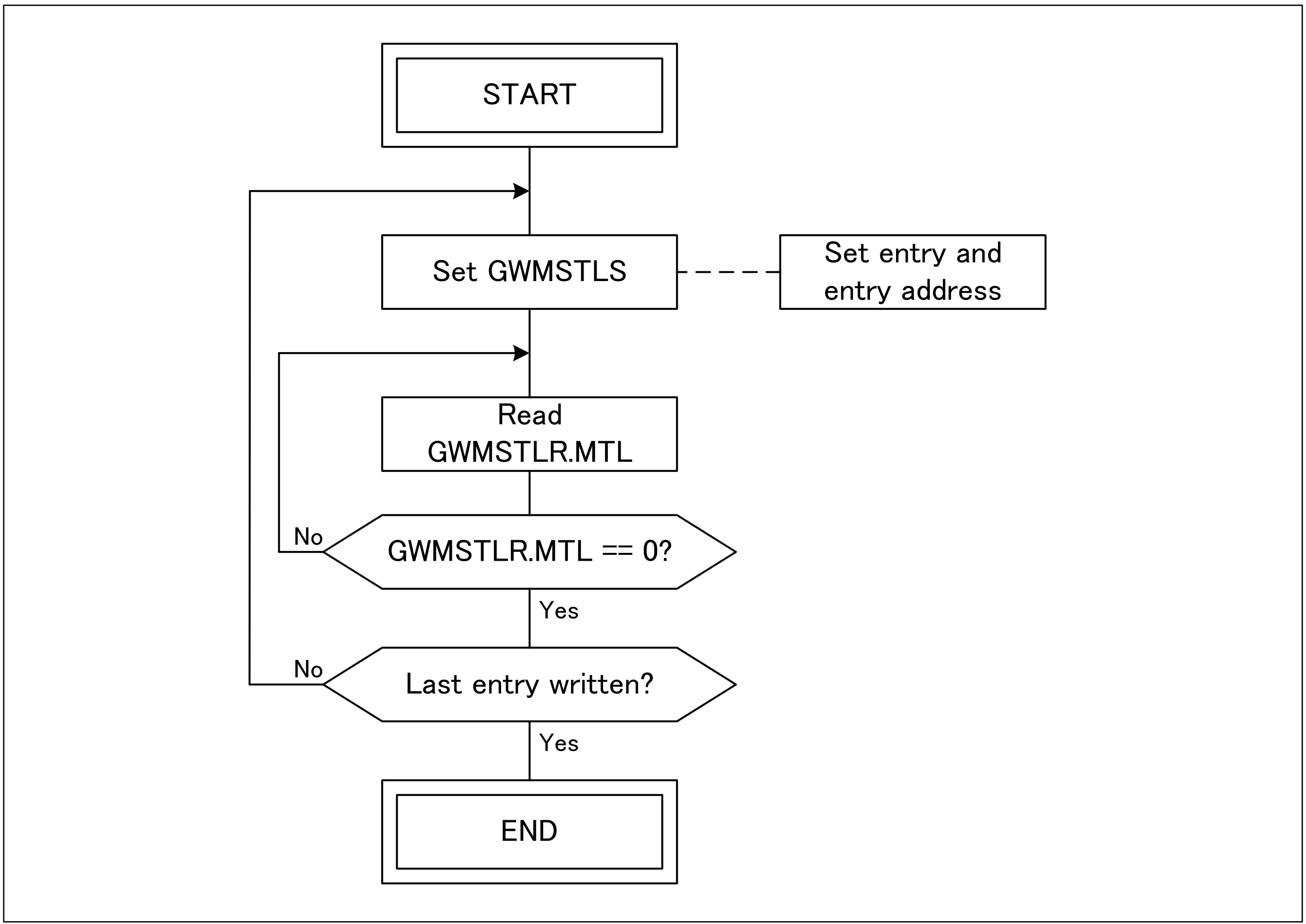


Figure 34.9 Multicast table setting flow

34.4.2.8 Multicast Table Read Flow

The multicast table read flow is described in Figure 34.10.

Restrictions:

- This flow is not usable in RESET and DISABLE modes.

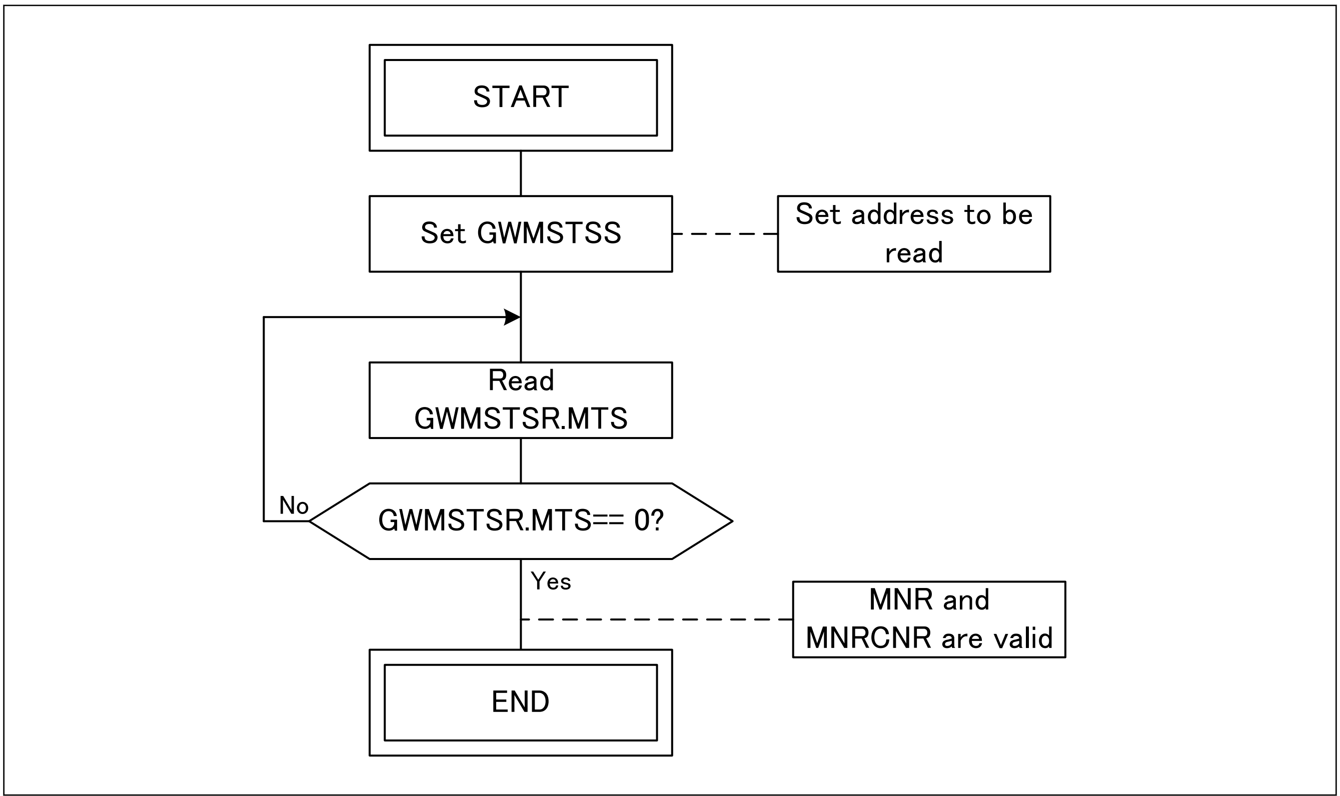


Figure 34.10 Multicast table read flow

34.4.2.9 AXI RAM Reset Flow

The AXI RAM reset flow is described in Figure 34.11.

Restrictions:

- This flow is not usable in RESET, DISABLE and OPERATION modes.

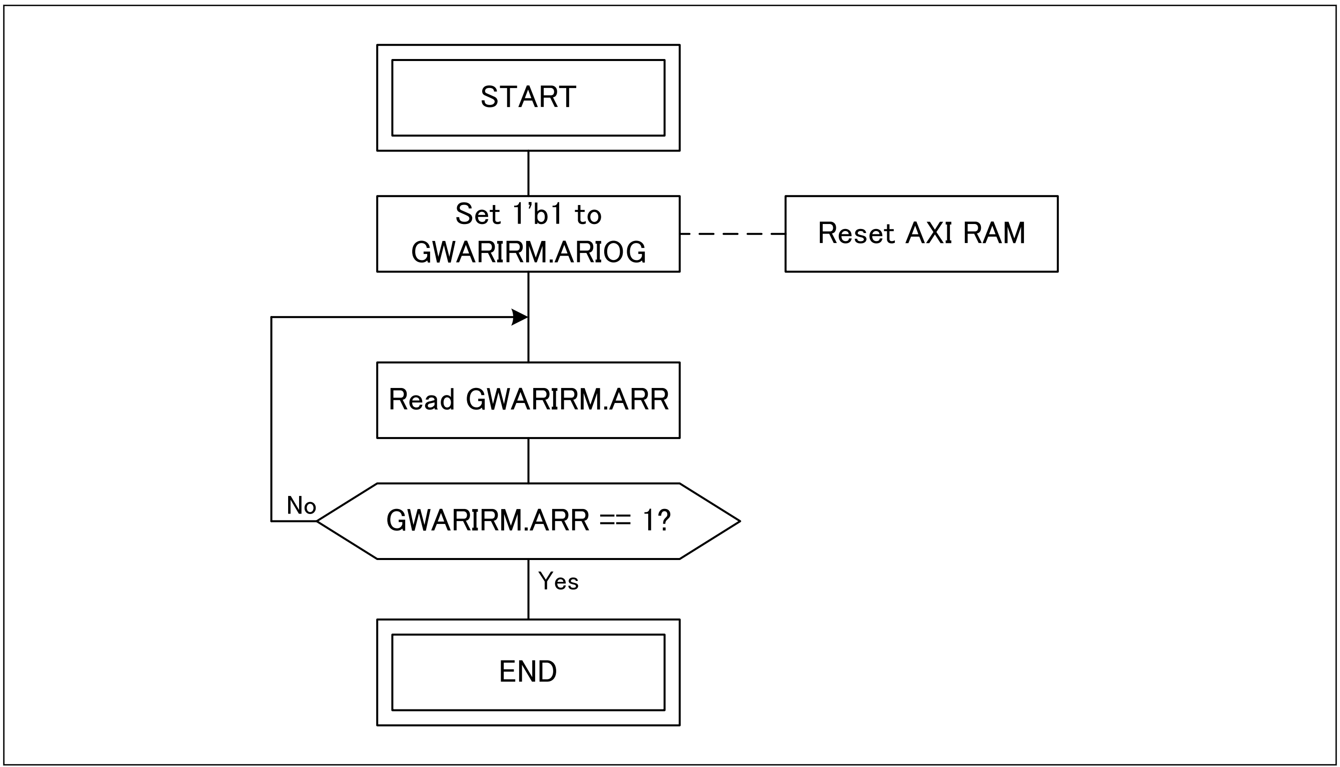


Figure 34.11 AXI RAM reset flow

34.4.2.10 AXI RAM Read Flow

The AXI RAM read flow is described in Figure 34.12.

Restrictions:

- This flow is not usable in RESET and DISABLE modes.

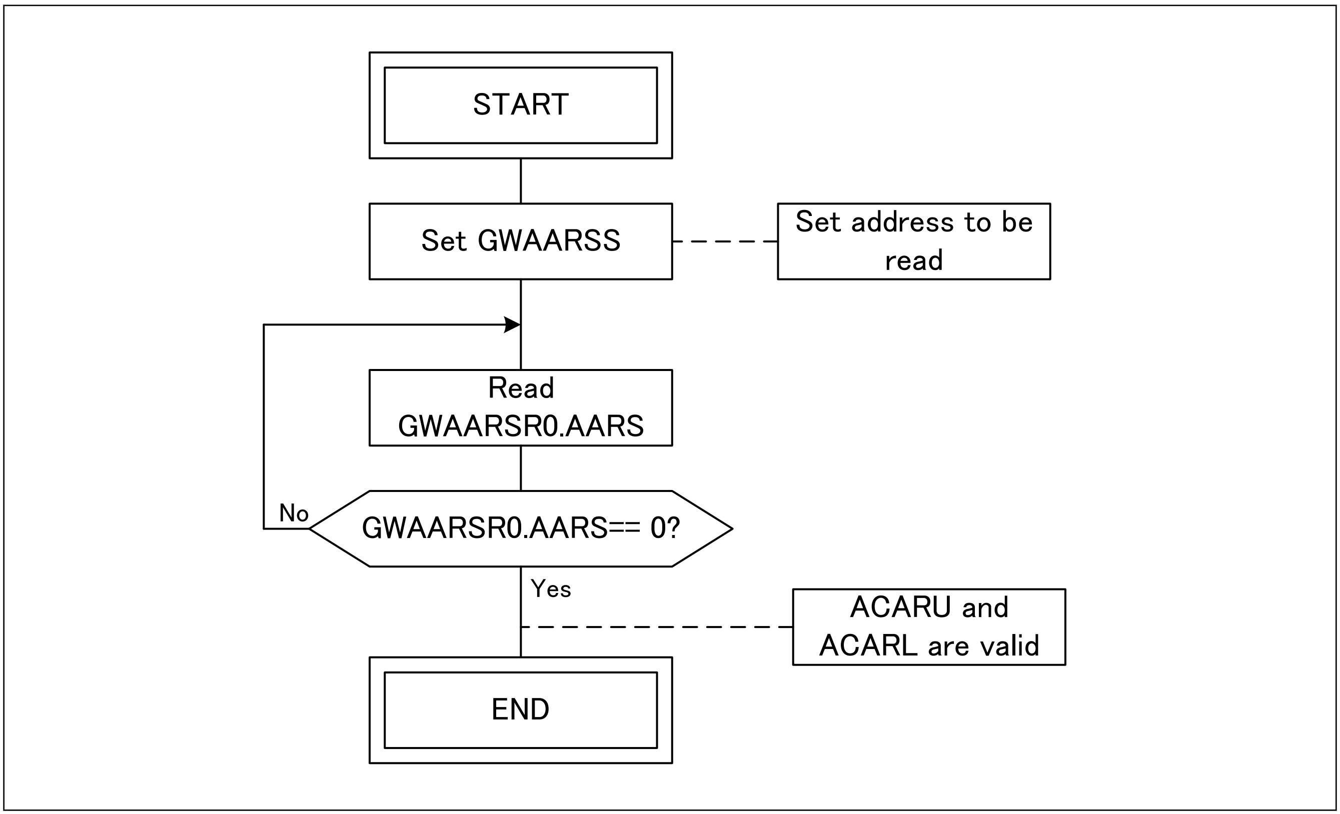


Figure 34.12 AXI RAM read flow

34.4.2.11 Global Rate Limiter Setting Flow

The global rate limiter setting flow is described in Figure 34.13.

Restrictions:

- This flow is not usable in RESET and DISABLE modes.

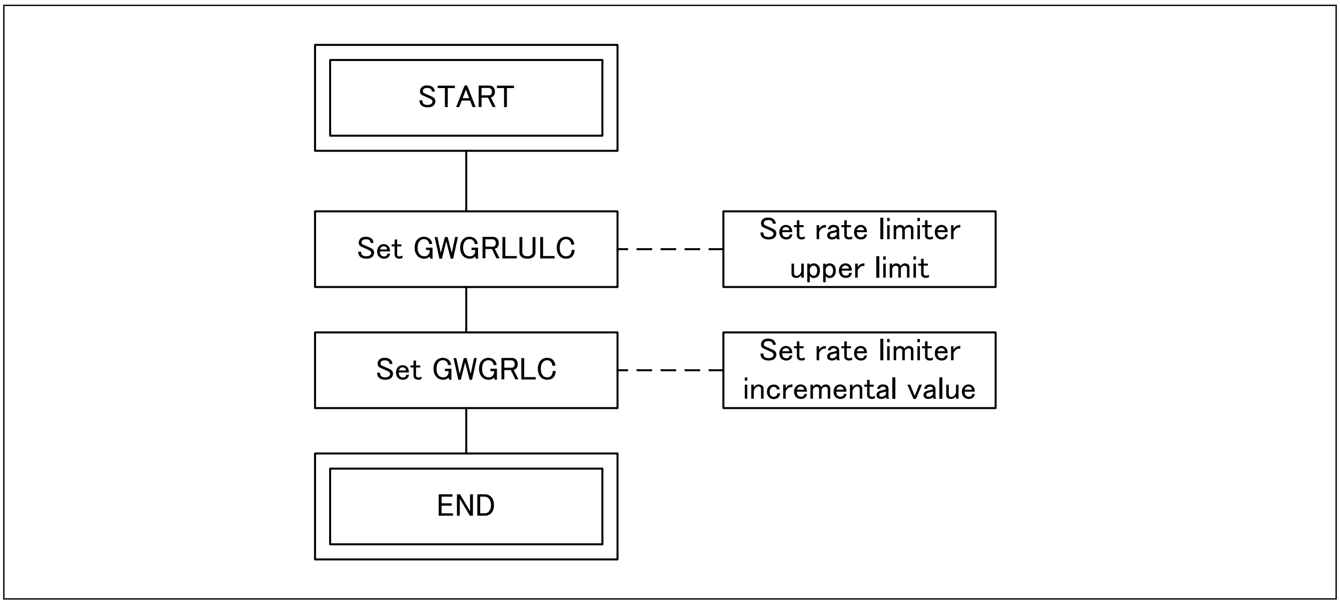


Figure 34.13 Global Rate limiter setting flow

34.4.2.12 Global Rate Limiter Disabling Flow

The global rate limiter disabling flow is described in Figure 34.14.

Restrictions:

- This flow is not usable in RESET and DISABLE modes.

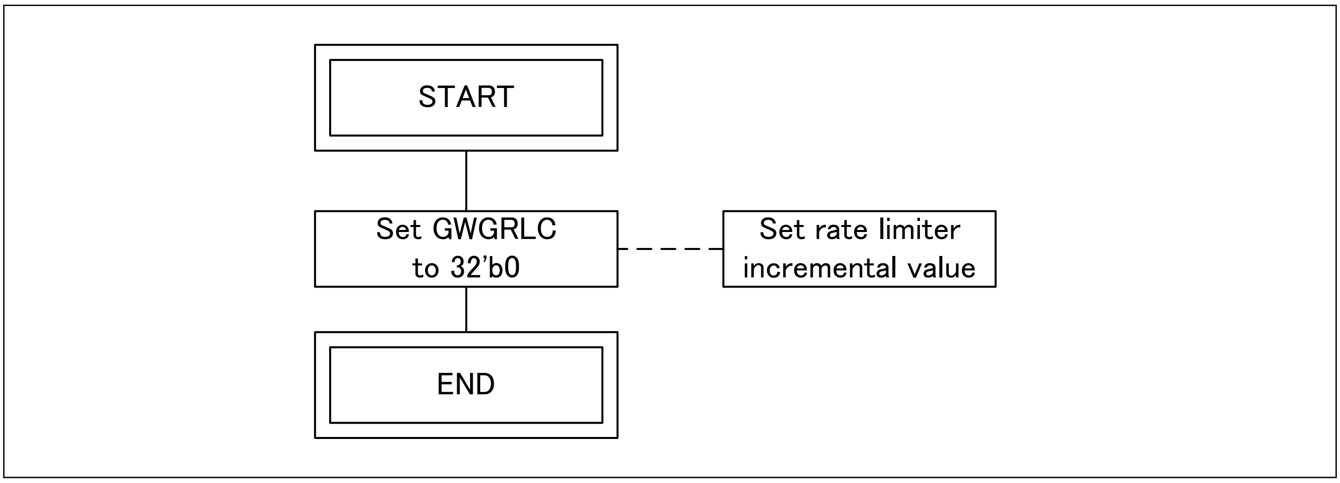


Figure 34.14 Global Rate limiter disabling flow

34.4.2.13 Rate Limiter i Setting Flow

The rate limiter i setting flow is described in Figure 34.15.

Restrictions:

- This flow is not usable in RESET and DISABLE modes.

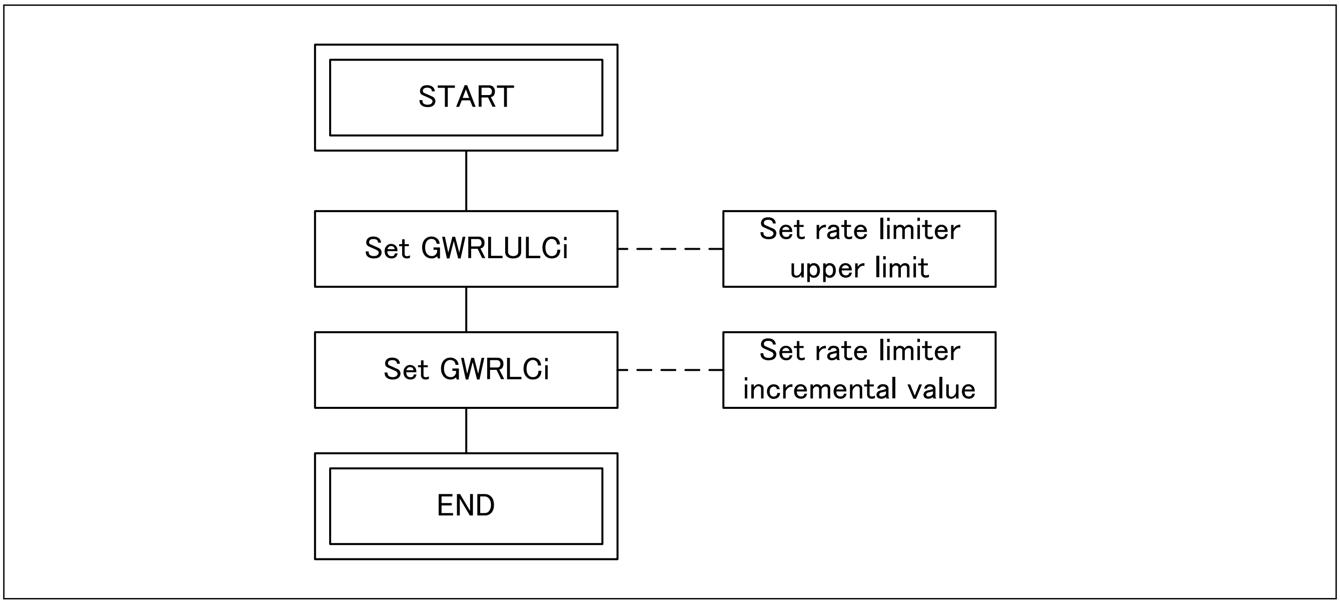


Figure 34.15 Rate limiter i setting flow

34.4.2.14 Rate Limiter i Disabling Flow

The rate limiter i disabling flow is described in Figure 34.16.

Restrictions:

- This flow is not usable in RESET and DISABLE modes.

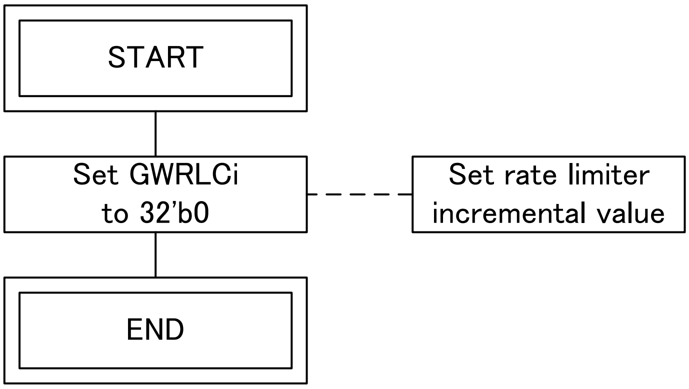


Figure 34.16 Rate limiter i disabling flow

34.4.2.15 AXI Table Read Flow

The AXI table read flow is described in Figure 34.17.

Restrictions:

- This flow is not usable in RESET and DISABLE modes.

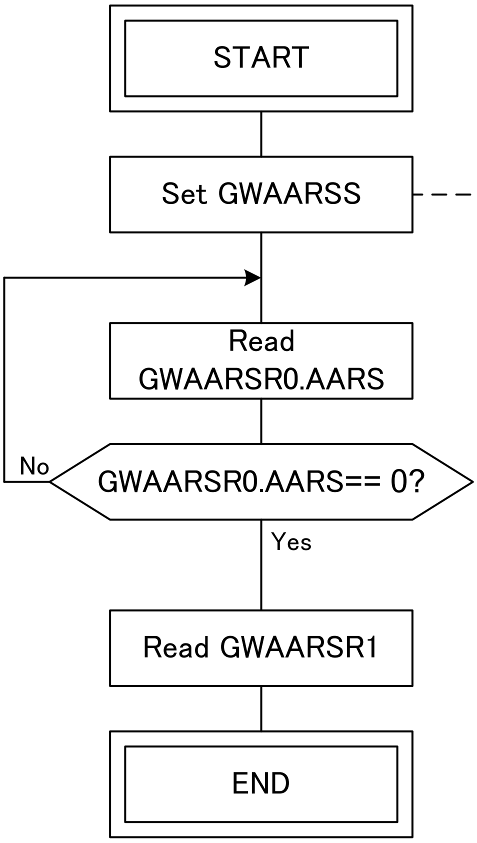


Figure 34.17 AXI table read flow

34.4.2.16 Interrupt Handling Flow

The interrupt handling flow is described in Figure 34.18.

Restrictions:

- This flow is not usable in RESET mode.

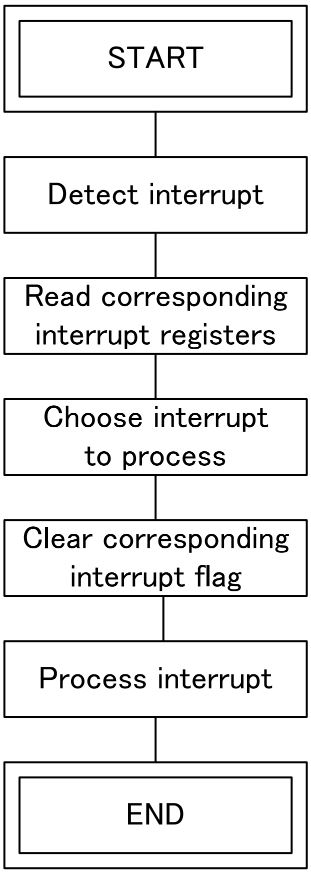


Figure 34.18 Interrupt handling flow

34.4.2.17 Called Software Flows

The flows described in this section can only be called from other flows thanks to a “flow link” box (Figure 34.3) and cannot be used alone.

34.4.2.17.1 Full Setting Flow

The full setting flow is described in Figure 34.19.

For LINKFIX table setting see section 34.5.1.3.1. LINKFIX Table Initialization.

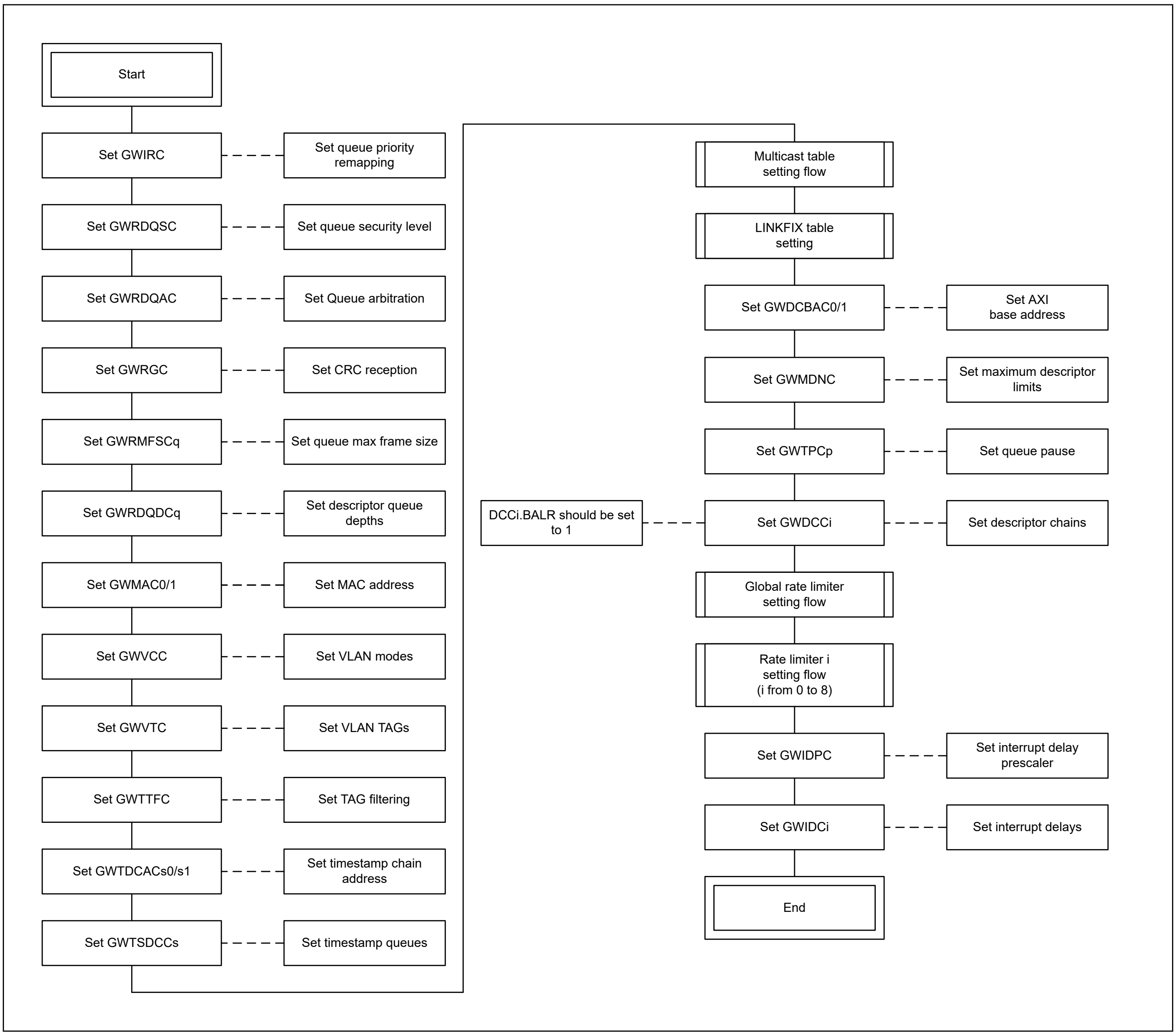


Figure 34.19 Full setting flow

34.4.2.17.2 AXI Emergency Stop Flow

The AXI emergency stop flow is described in Figure 34.20.

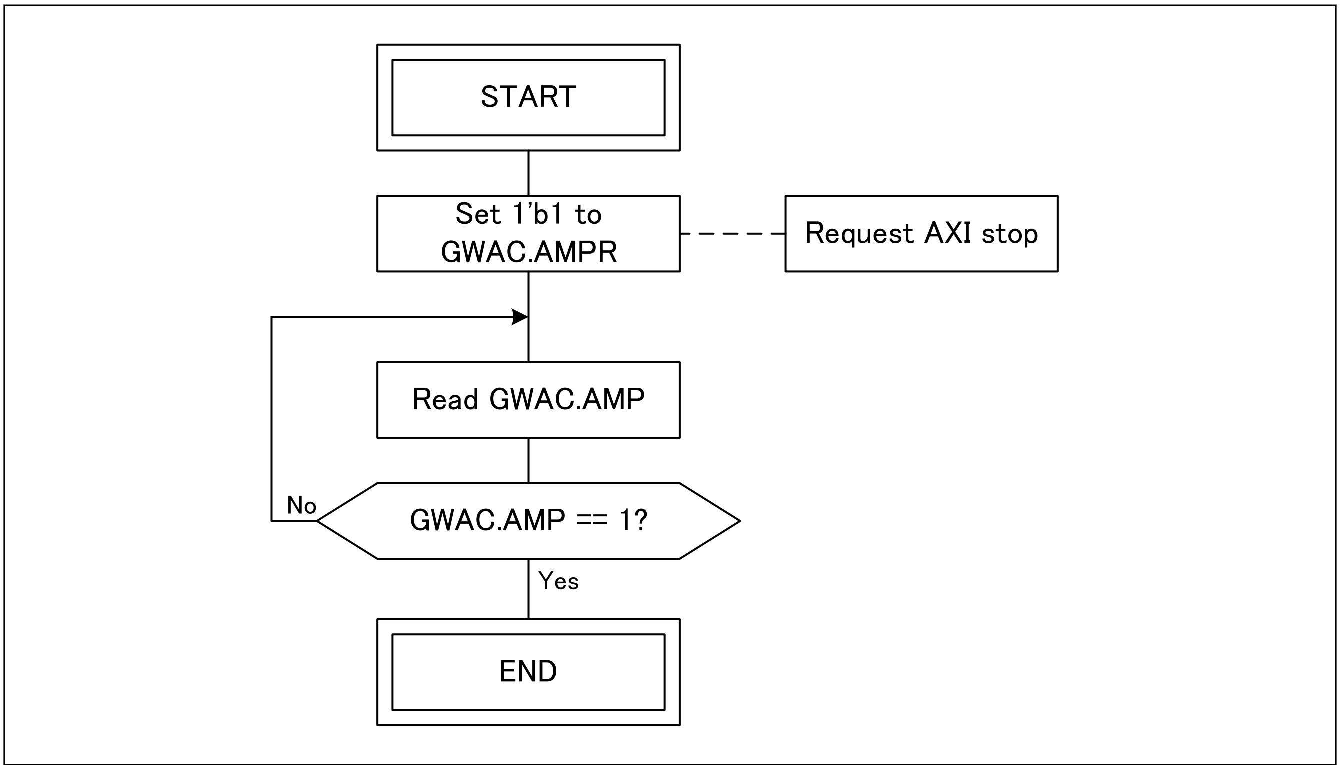


Figure 34.20 AXI emergency stop flow

34.4.2.18 Register Writable Without Software Flow

This section describes registers that have not been described so far. These registers can be changed dynamically. (However, it is necessary that the initial settings such as the clock enabling have been completed.)

Note: When a frame has reached GWCA and before the frame has been written into URAM by AXI BMI if GWDCC configurations of target chain is reconfigured by the user then the frame will be handled using the GWDCC register settings before reconfiguration and not using the reconfigured GWDCC settings.

34.5 Functional Details

34.5.1 AXI Master General Information

In this section is described the knowledge required to utilize the GWCA AXI master. It contains the AXI address table handling, the AXI descriptor general description, and the AXI IDs utilization description.

34.5.1.1 AXI Descriptor Queue Mapping

The AXI descriptor queue mapping is described along with its mapping to the **INTER_TX** interface, the incremental queues and the rate limiters in Figure 34.21. The global rate limiter is not represented because it limits throughput for all the TX queues.

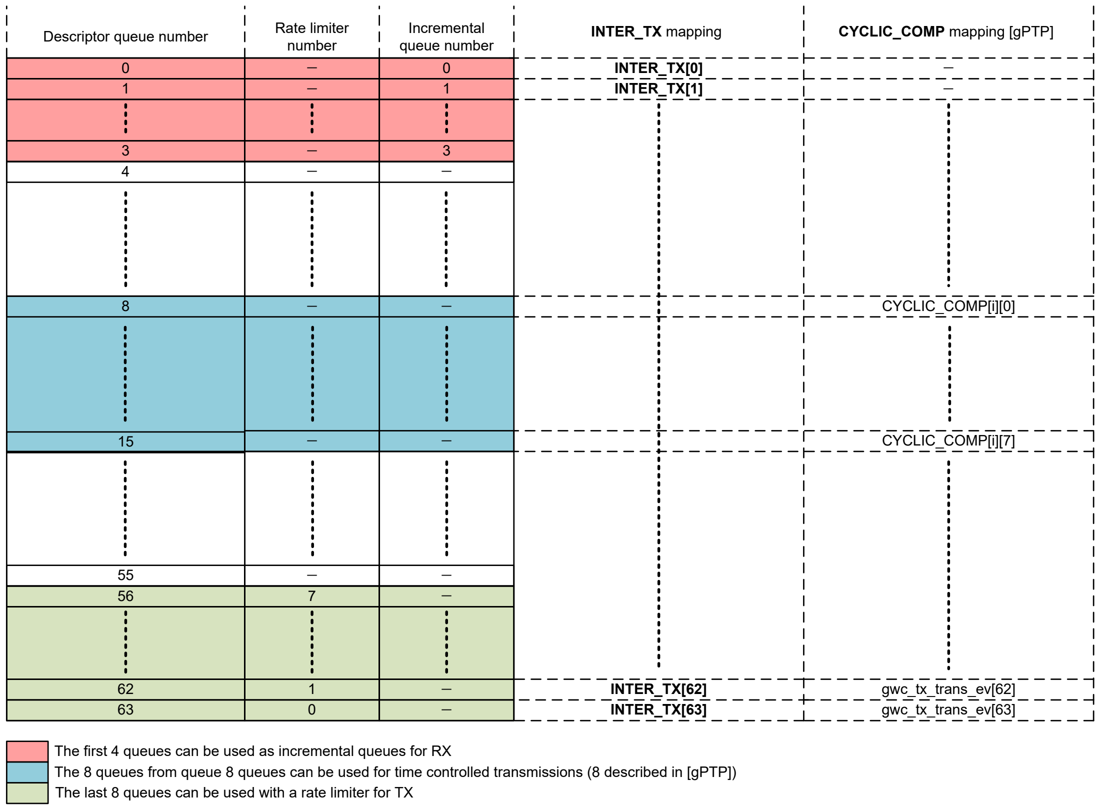


Figure 34.21 AXI descriptor queue mapping

34.5.1.2 AXI Descriptor General Description

34.5.1.2.1 General Formats

AXI descriptor general formats are described in Figure 34.22, Figure 34.23, Figure 34.24 and Figure 34.25. The fields are explained in Table 34.5. The fields utilization will be explained per transaction type (Data reception, Data transmission and Timestamp reception).

<table>
<tr><th>Byte</th><th>Bit7</th><th>Bit6</th><th>Bit5</th><th>Bit4</th><th>Bit3</th><th>Bit2</th><th>Bit1</th><th>Bit0</th></tr>
<tr><td>0</td><td colspan="8">DS[7:0]</td></tr>
<tr><td>1</td><td colspan="4">INFO0[3:0]</td><td colspan="4">DS[11:8]</td></tr>
<tr><td>2</td><td colspan="4">DT[3:0]</td><td>DIE</td><td>AXIE</td><td>DSE</td><td>ERR</td></tr>
<tr><td>3</td><td colspan="8">PTR[39:32]</td></tr>
<tr><td>4</td><td colspan="8" rowspan="4">PTR[31:0]</td></tr>
<tr><td>5</td></tr>
<tr><td>6</td></tr>
<tr><td>7</td></tr>
</table>

Figure 34.22 Basic descriptor (GWDCCi.ETS == 0 and GWDCCi.EDE == 0)

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	DS[7:0]							
1	INFO0[3:0]				DS[11:8]			
2	DT[3:0]				DIE	AXIE	DSE	ERR
3	PTR[39:32]							
4	PTR[31:0]							
5								
6								
7								
8	TS[63:0]							
9								
10								
11								
12								
13								
14								
15								

Figure 34.23 Timestamp descriptor (GWDCCi.ETS == 1 and GWDCCi.EDE == 0)

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	DS[7:0]							
1	INFO0[3:0]				DS[11:8]			
2	DT[3:0]				DIE	AXIE	DSE	ERR
3	PTR[39:32]							
4	PTR[31:0]							
5								
6								
7								
8	INFO1[63:0]							
9								
10								
11								
12								
13								
14								
15								

Figure 34.24 Extended descriptor (GWDCCi.ETS == 0 and GWDCCi.EDE == 1)

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	DS[7:0]							
1	INFO0[3:0]				DS[11:8]			
2	DT[3:0]				DIE	AXIE	DSE	ERR
3	PTR[39:32]							
4–7	PTR[31:0]							
8–15	INFO1[63:0]							
16–23	TS[63:0]							

Figure 34.25 Extended timestamp descriptor (GWDCCi.ETS == 1 and GWDCCi.EDE == 1)

Table 34.5 General format descriptor field explanation

Field name	Bit width	Description
DS	12	Descriptor size Restrictions: ● SW: DS field maximum value is 2048. ● SW: For other descriptors that LINKFIX, LEMPTY, EEMPTY, LINK and EOS, DS should not be set to 0.
INFO0	4	Information 0
ERR	1	Error
DSE	1	Data Size Error
AXIE	1	AXI Bus Error
DIE	1	Descriptor Interrupt Enable This (DESCR.DIE) field should not be set for LINK/LINKFIX descriptors in Rx chain.
DT	4	Descriptor Type, see section 34.5.1.2.2. Descriptor Types.
PTR	40	Pointer
INFO1	64	Information 1
TS	64	Timestamp

Note: General format descriptor field utilization is described per use-case later in the following sections.

34.5.1.2.2 Descriptor Types

The table below gives an overview of all descriptor types used by GWCA. The first column shows the descriptor type name and the second column the 4-bit number to be written to DESCR.DT.

Table 34.6 Descriptor Type

Name	DT	Description
Empty data descriptors		
FEMPTY_IS	1	Frame Empty Incremental Start Frame data related descriptor without valid frame data. Used to set an incremental area.
FEMPTY_IC	2	Frame Empty Incremental Continue Frame data related descriptor without valid frame data. Used to continue using an incremental area.
FEMPTY_ND	3	Frame Empty No Data Storage Frame data related descriptor without valid frame data. Used to reject data.
FEMPTY	4	Frame Empty Frame complete data related descriptor without valid frame data. Used to save frame data.
FEMPTY_START	5	Frame Empty start Frame start data related descriptor without valid frame data. Used to save first part of a frame.
FEMPTY_MID	6	Frame Empty Middle Frame continues data related descriptor without valid frame data. Used to save middle part of a frame.
FEMPTY_END	7	Frame Empty End Frame end data related descriptor without valid frame data. Used to save end part of a frame.
Data descriptors		
FSINGLE	8	Frame Single Storage Element has valid frame data of a complete frame.
FSTART	9	Frame Start Storage Element has valid frame data. Frame starts with this descriptor and continues in next descriptor.
FMID	10	Frame Middle Storage Element has valid frame data. Frame has started with a previous descriptor and continues in next descriptor.
FEND	11	Frame End Storage Element has valid frame data. Frame has started with a previous descriptor and ends with this descriptor.
Chain control / HW/SW arbitration		
LINKFIX	0	Linkfix Control element pointing to next descriptor in chain.
LEMPTY	12	Link Empty Used control element pointing to next descriptor in chain.
EEMPTY	13	EOS Empty Used control element pausing a transmit chain.
LINK	14	Link Control element pointing to next descriptor in chain.
EOS	15	End Of Set Control element pausing a transmit chain.

34.5.1.3 LINKFIX Table

34.5.1.3.1 LINKFIX Table Initialization

The usage of AXI address table requires the setting of a LINKFIX table in the CPU USER RAM at {GWDCBAC.DCBAU, GWDCBAC.DCBAL} location. This table will be used to set the first address of descriptor chains. It can be used any time to change a chain base address or to return to the previous base address.

The LINKXFIX table setting is described in Figure 34.26 through an example. Note that SW and HW initialization are happening simultaneously.

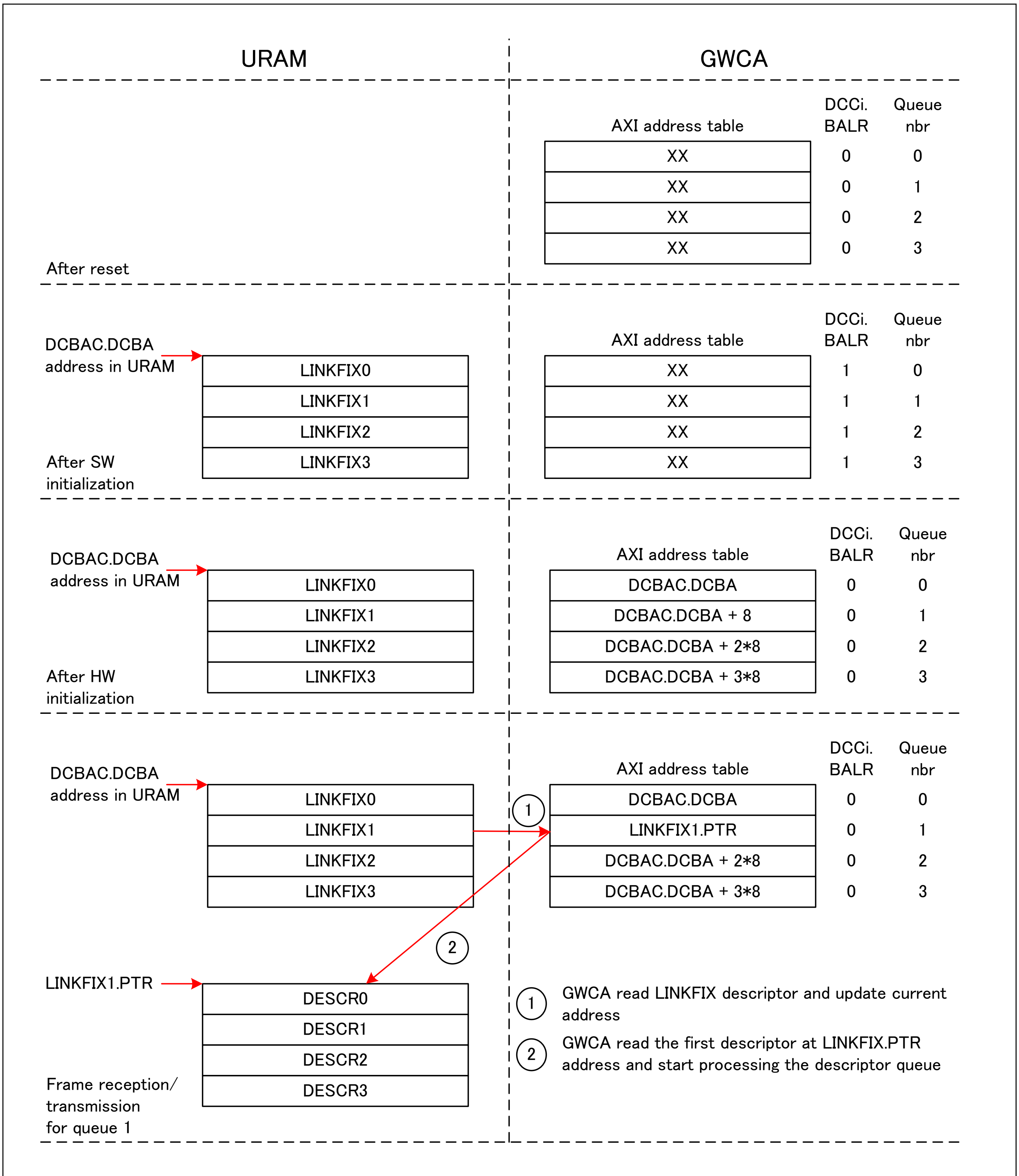


Figure 34.26 LINKXFIX table setting example

Restrictions:

- SW: A LEMPTY descriptor queues must be written for preventing them from starting.
- HW: Added descriptors in the LINKFIX table are 8 byte-descriptors which are independent of GWDCCi.EDE and GWDCCi.ETS register settings.

Note: LINKFIX and LINK descriptors are interchangeable. The only difference is that the LINK descriptor can be written back by HW.

34.5.1.3.2 LINKFIX Table Descriptors

For the LINKFIX table, a LINKFIX descriptor is used to start a descriptor queue, a LEMPTY descriptor is used to disable a TX queues (GWDCCi.DQT == 1) and a LEMPTY descriptor is used to disable a RX queues (GWDCCi.DQT == 0). In the LINKFIX table, only basic descriptors can be used (section 34.5.1.2.1. General Formats). The descriptor formats for LINKFIX table are describes in Figure 34.27 and Table 34.7. If a reception disabled queue is started despite being disabled in the LINKFIX table, the descriptor queue be considered as full (see register GWEIS2i explanation).

<table>
<tr><th>Byte</th><th>Bit7</th><th>Bit6</th><th>Bit5</th><th>Bit4</th><th>Bit3</th><th>Bit2</th><th>Bit1</th><th>Bit0</th></tr>
<tr><td>0</td><td colspan="8">8'b0</td></tr>
<tr><td>1</td><td colspan="4">4'b0</td><td colspan="4">4'b0</td></tr>
<tr><td>2</td><td colspan="4">4'd0</td><td>1'b0</td><td>1'b0</td><td>1'b0</td><td>1'b0</td></tr>
<tr><td>3</td><td colspan="8">PTR[39:32]</td></tr>
<tr><td>4</td><td colspan="8" rowspan="4">PTR[31:0]</td></tr>
<tr><td>5</td></tr>
<tr><td>6</td></tr>
<tr><td>7</td></tr>
</table>

LINKFIX for LINKFIX table

<table>
<tr><th>Byte</th><th>Bit7</th><th>Bit6</th><th>Bit5</th><th>Bit4</th><th>Bit3</th><th>Bit2</th><th>Bit1</th><th>Bit0</th></tr>
<tr><td>0</td><td colspan="8">8'b0</td></tr>
<tr><td>1</td><td colspan="4">4'b0</td><td colspan="4">4'b0</td></tr>
<tr><td>2</td><td colspan="4">4'd12</td><td>1'b0</td><td>1'b0</td><td>1'b0</td><td>1'b0</td></tr>
<tr><td>3</td><td colspan="8">PTR[39:32]</td></tr>
<tr><td>4</td><td colspan="8" rowspan="4">PTR[31:0]</td></tr>
<tr><td>5</td></tr>
<tr><td>6</td></tr>
<tr><td>7</td></tr>
</table>

LEMPTY for LINKFIX table

Figure 34.27 LINKFIX table descriptor formats

Table 34.7 LINKFIX table descriptor field description

Field name	LINKFIX	LEMPTY
DS	0	0
INFO0	0	0
ERR	0	0
DSE	0	0
AXIE	0	0
DIE	0	0
DT	0	12
PTR	PTR Pointer pointing to the first descriptor of the descriptor chain	0
INFO1 (GWDCCi.EDE == 1)	NA for LINKFIX table	NA for LINKFIX table
TS (GWDCCi.ETS == 1)	NA for LINKFIX table	NA for LINKFIX table

34.5.1.4 AXI Common Descriptor Utilization

AXI common descriptors are the descriptors which are not reserved to a special use-case and can be inserted in any descriptor chain at any time. It includes LINKFIX and LINK descriptors.

34.5.1.4.1 LINKFIX Descriptor Utilization

A LINKFIX descriptor is used to change a descriptor queue pointer in the GWCA to move it to another place and is described in Figure 34.28. The LINKFIX descriptor is never written back to decrease the AXI load.

The LINKFIX descriptor format is described in Figure 34.29 (for basic descriptor, section 34.5.1.2.1. General Formats) and Table 34.8.

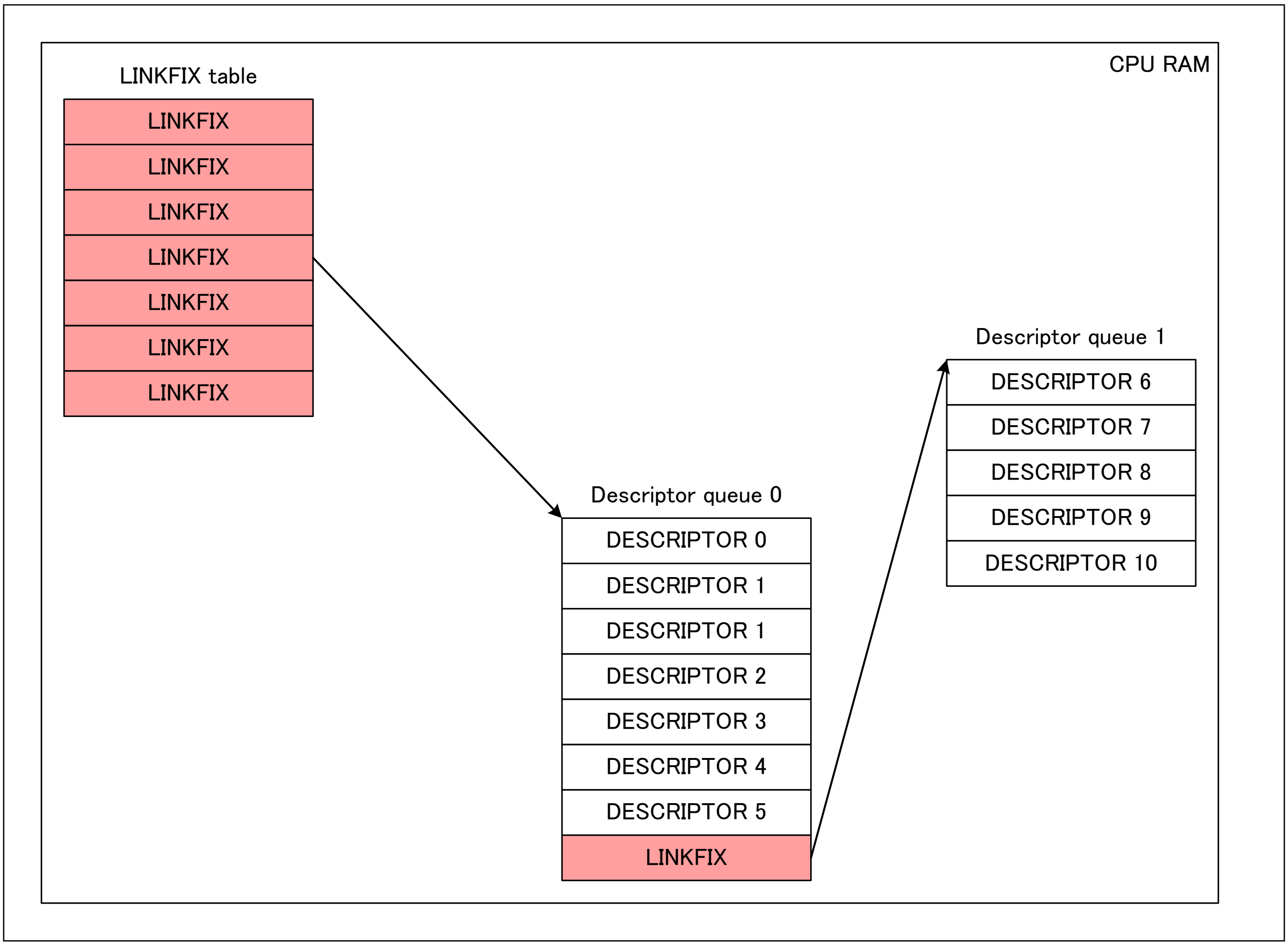


Figure 34.28 LINKFIX descriptor utilization

Note:
- The LINKFIX table is described in section 34.5.1.3. LINKFIX Table.
- A LINKFIX descriptor can be placed anywhere in a descriptor queue.
- LINKFIX and LINK descriptors are interchangeable. The only difference is that the LINK descriptor can be written back by HW.

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	8'b0							
1	4'b0				4'b0			
2	4'd0				1'b0	1'b0	1'b0	1'b0
3	PTR[39:32]							
4	PTR[31:0]							
5								
6								
7								

LINKFIX written by SW

Figure 34.29 LINKFIX utilization format

Table 34.8 LINKFIX utilization field description

Field name	Field description written by SW	Field description after HW write-back
DS	0	NA: LINKFIX descriptors are never written back
INFO0	0	NA: LINKFIX descriptors are never written back
ERR	0	NA: LINKFIX descriptors are never written back
DSE	0	NA: LINKFIX descriptors are never written back
AXIE	0	NA: LINKFIX descriptors are never written back
DIE	0	NA: LINKFIX descriptors are never written back
DT	0	NA: LINKFIX descriptors are never written back
PTR	PTR Pointer pointing to the new descriptor queue	NA: LINKFIX descriptors are never written back
INFO1 (GWDCCi.EDE == 1)	0	NA: LINKFIX descriptors are never written back
TS (GWDCCi.ETS == 1)	0	NA: LINKFIX descriptors are never written back

34.5.1.4.2 LINK Descriptor Utilization

A LINK descriptor has the same usage as a LINKFIX descriptor (see section 34.5.1.4.1. LINKFIX Descriptor Utilization). Contrary to the LINKFIX descriptor the LINK descriptor can be written back by HW. The LINK descriptor format is described in Figure 34.30 (for basic descriptor, section 34.5.1.2.1. General Formats) and Table 34.9.

Note: LINKFIX and LINK descriptors are interchangeable. The only difference is that the LINK descriptor can be written back by HW.

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	8'b0							
1	4'b0				4'b0			
2	4'd14				1'b0	1'b0	1'b0	1'b0
3	PTR[39:32]							
4	PTR[31:0]							
5								
6								
7								

LINK written by SW

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	8'b0							
1	4'b0				4'b0			
2	4'd12				1'b0	1'b0	1'b0	1'b0
3	PTR[39:32]							
4	PTR[31:0]							
5								
6								
7								

LEMPTY written back by HW

Figure 34.30 LINK utilization format

Table 34.9 LINK utilization field description

Field name	Field description written by SW	Field description after HW write-back
DS	0	0 (Value is copied from read descriptor)
INFO0	0	0 (Value is copied from read descriptor)
ERR	0	0 (Value is copied from read descriptor)
DSE	0	0 (Value is copied from read descriptor)
AXIE	0	0 (Value is copied from read descriptor)
DIE	0	0 (Value is copied from read descriptor)
DT	14	12 (Value is copied from read descriptor)
PTR	PTR (user setting) Pointer pointing to the new descriptor queue	PTR (Value is copied from read descriptor)
INFO1 (GWDCCi.EDE == 1)	0	NA: INFO1 not written back for LINK descriptors
TS (GWDCCi.ETS == 1)	0	NA: TS not written back for LINK descriptors

34.5.1.5 AXI Descriptor Queue Format

The AXI master accepts two kinds of descriptor queues: linear descriptor queues and cyclic descriptor queues. A linear descriptor queue is a queue which is ended by a terminate descriptor. A linear descriptor queue example is described in figure Figure 34.31. A cyclic descriptor queue is a queue which is looping on itself. A cyclic descriptor queue example is described in figure Figure 34.32.

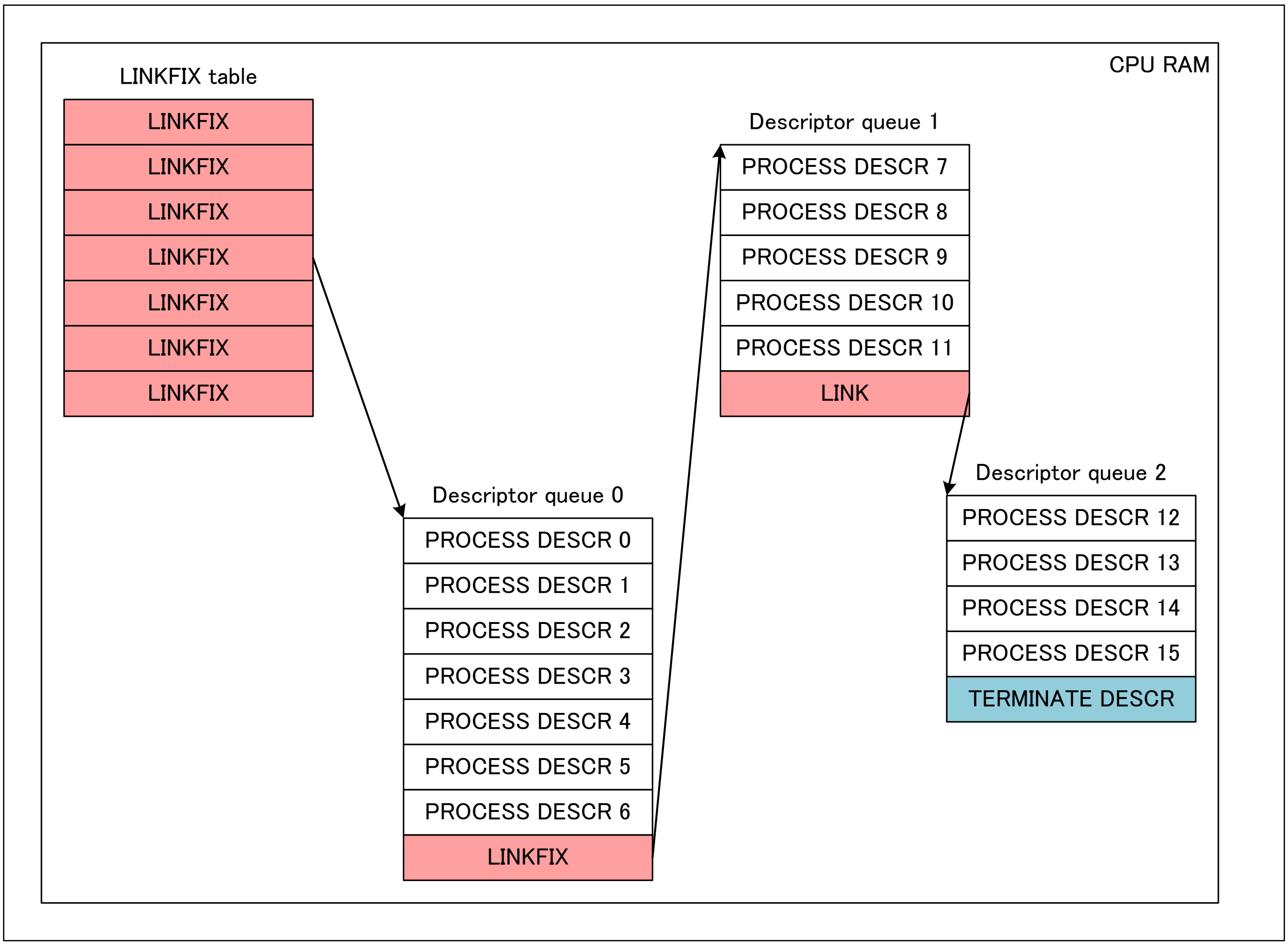


Figure 34.31 Linear descriptor queue example

Restrictions:

- SW: To add data to a linear queue, the SW should write the data and process/terminate descriptor and overwrite the previous terminate descriptor with a LINK/LINKFIX descriptor pointing to the new descriptor chain.

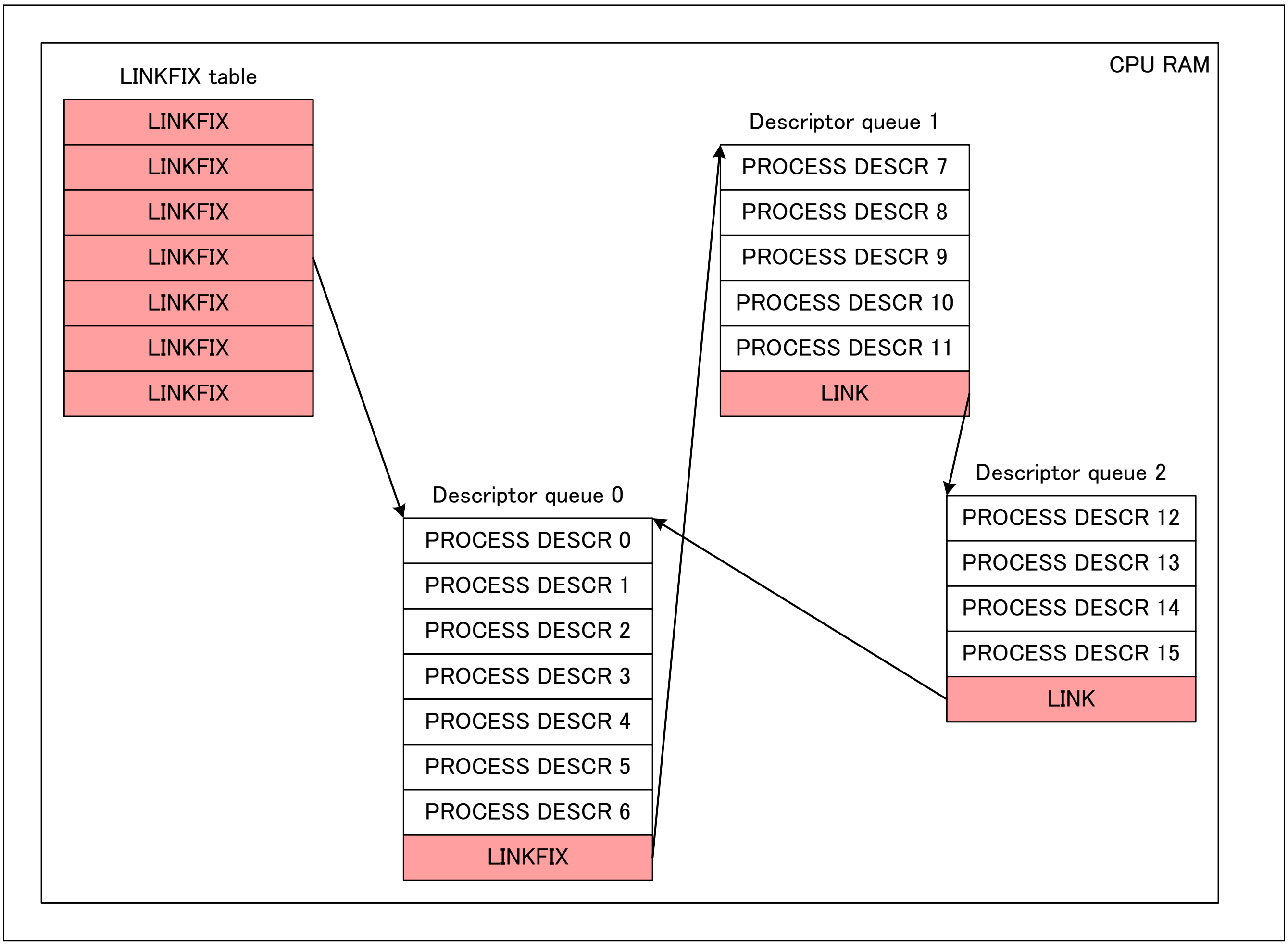


Figure 34.32 Cyclic descriptor queue example

Note:
- The LINKFIX table is described in section 34.5.1.3. LINKFIX Table.
- The LINK, LINKFIX utilization is described in section 34.5.1.4. AXI Common Descriptor Utilization.
- A terminate descriptor (TERMINATE DESCR in Figure 34.31) correspond to a FSINGLE or a LEMPTY descriptor for an RX queue, a FEMPTY or a LEMPTY descriptor for a TX queue and a FSINGLE or a LEMPTY descriptor for a TS queue.
- A process descriptor (PROCESS DESCR in Figure 34.31 and Figure 34.32) correspond to a FEMPTY_IS, FEMPTY_IC, FEMPTY_ND, FEMPTY, FEMPTY_START, FEMPTY_MID or FEMPTY_END descriptor for an RX queue (see section 34.5.3.5. AXI Master Interface), to a FSINGLE, FSTART, FMID, FEND or EOS descriptor for a TX queue (see section 34.5.2.2. AXI Master Interface) and to a FEMPTY_ND for a TS queue (see section 34.5.4.2. AXI Master Interface).

34.5.1.6 AXI Address RAM Searching

Searching functions is used to read entries in the AXI address RAM. Table 34.10 describes register used to read an entry in the TAS RAM. Table 34.11 describes the read results.

Table 34.10 AXI Address read registers

Register name	Field name/corresponding field in MAC table	Field explanation
GWAARSS.AARA	Entry address Set the descriptor queue number from which AXU address should be read.	Not present in AXI address RAM, entry will be read from this address.

Table 34.11 AXI Address read result

Register name	Field name/corresponding field in MAC table	Field explanation
{GWAARSR0.ACARU, GWAARSR1.ACARL}	AXI address value	Address of the next descriptor to be processed for AXI descriptor queue number GWAARSS.AARA.

Note: {GWAARSR0.ACARU, GWAARSR1.ACARL} indicates the address of the next descriptor to be processed for AXI descriptor queue number but it does not mean that the previous descriptor process is completed. That’s why this register only has debugging purpose and not HW/SW synchronization purposes.

34.5.2 Data Transmission

GWCA allows data transmission through the GWCA TX data path, The TX data path is described in Figure 34.33.

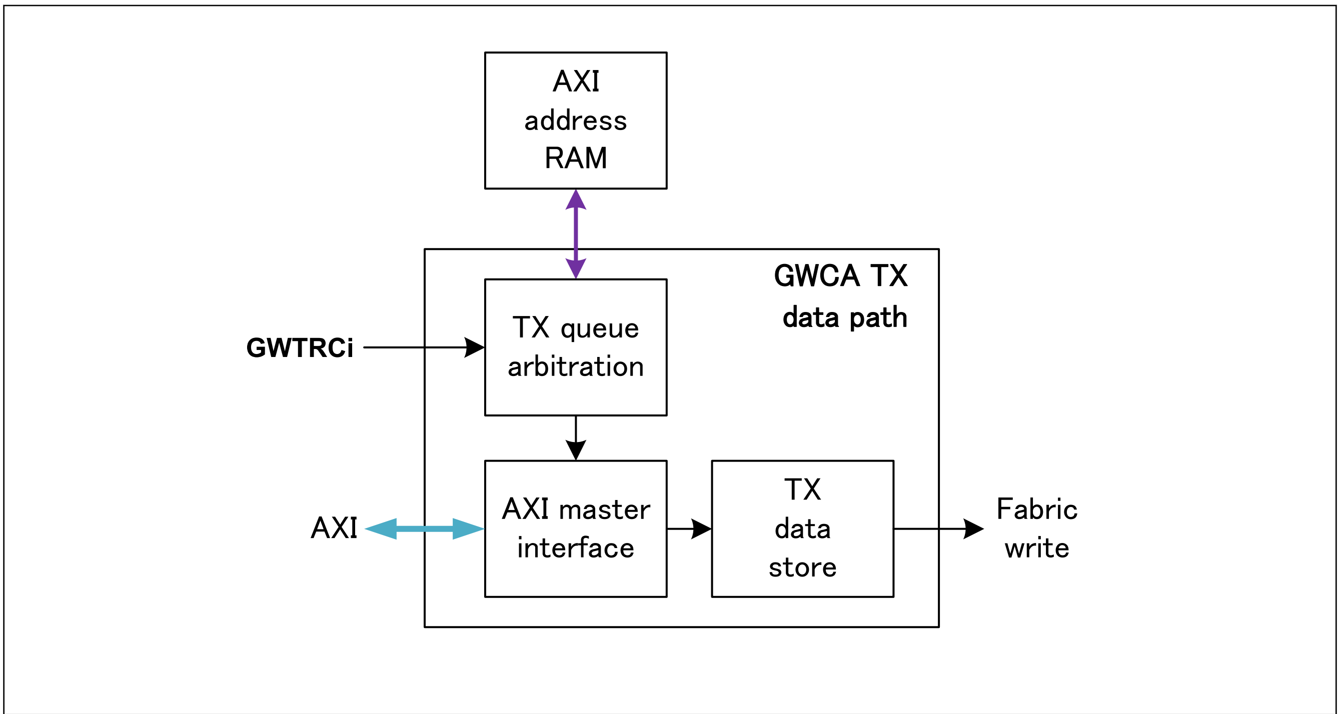


Figure 34.33 GWCA TX data path block diagram

The TX data path is separated in three blocks:

- TX queue arbitration: This block arbitrates between TX queues when several queues are requesting for transmission.
- AXI master interface: This block handles the data/AXI descriptor exchange with the CPU.
- TX data store: This block extracts TAG information for frames, applies VLAN tagging and save the frames in the local RAM. This block can also release pointers while going out of OPERATION mode. The pointer release is here for switch hardware purpose and its description is not required for switch utilization so it will not be described.

34.5.2.1 TX Queue Arbitration

TX queue arbitration has four functions. Pausing certain queues when switch is about to overflow, limiting TX queue data rates, arbitrating between TX descriptor queue with a strict priority algorithm and arbitrating between same priority queues with a round-robin algorithm.

34.5.2.1.1 Per Priority Pause

Per priority pause aims at pausing descriptor queue transmission depending on their priority using PAUSE[2] values [COMA] and GWTPCp register. It avoids the AXI master to inject uncritical data in the switch when it is about to overflow.

Functions:

- GWTPCp sets the priorities that should be paused when PAUSE[2][p] is set.

34.5.2.1.2 Rate Limiters

Rate limiters are controlling the input flow of the switch. Using these limiters could prevent the switch from overflowing. GWCA has two types of rate limiters, a global rate limiter and 8 per-queue rate limiters. By setting the rate limiter configurations correctly, expected bandwidth with a tolerance value of less than ± 2% can be achieved.

(1) Per-queue rate-limiter

A per-queue rate-limiter aims at limiter the data throughput per one descriptor queue using GWRLCi.RLE, GWRLCi.RLIV and GWRLULCi.RLUL registers. For per-port rate limiter/descriptor queue mapping information see section 34.5.1.1. AXI Descriptor Queue Mapping.

Functions:

- GWRLCi.RLE register enables rate-limiter number i.
- GWRLCi.RLIV register sets the maximum throughput accepted by rate limiter number i and should be set respecting equation (1).
- GWRLULCi.RLUL register sets the maximum collision time accepted by rate limiter number i and should be set respecting equation (2).

1. GWRLCi.RLIV = 256 x *ACLK_period[ns]* x *maxBandwidth[Gbps]*
2. GWRLULCi.RLUL = *maximumBurst[bit]*

Where:

- *ACLK_period* is ACLK clock period.
- *maxBandwidth* is the expected maximum throughput accepted by rate limiter number i.
- *maximumBurst* is the maximum burst in bit accepted for corresponding queue.

(2) Global rate limiter

The global rate limiter aims at limiting all AXI data transmission throughput using GWGRLC.GRLE, GWGRLC.GRLIV and GWGRLULC.GRLUL registers and can be monitored using GWGRLC.GRLULRS.

Functions:

- GWGRLC.GRLE register enables the global rate limiter.
- GWGRLC.GRLIV register sets the maximum throughput accepted by AXI data transmission and should be set respecting equation (1).
- GWGRLULC.GRLUL register sets the maximum number of credits that the global rate limiter can stack and should be set respecting equation (2).
- GWGRLC.GRLULRS register shows if some credit has been lost because of a too high AXI latency or because of a too slow AXI.

1. GWGRLC.GRLIV = 256 x *ACLK_period[ns]* x *max_bandwidth[Gbps]*
2. GWGRLULC.GRLUL = *ACLK_period[ns]* x *max_bandwidth[Gbps]* x 2**maximum_axi_latency[cycle]*

Where:

- ACLK_period is ACLK clock period in ns.
- max_bandwidth is the expected maximum throughput accepted by GWCA.
- maximum_axi_latency is the AXI maximum latency in ACLK clock number.

Restrictions:

- SW: The sum of all GWRLCi.RLIV registers should always be smaller or equal to GWGRLC.GRLIV.

34.5.2.1.3 Strict Priority

Strict priority arbitrates between TX queue which are authorized by rate limiters to transmit using GWDCCi.DCP. The strict priority algorithm operates as the one in the data reception descriptor store. See section (1) Strict priority (SP) for more information.

Functions:

- GWDCCi.DCP register sets descriptor queue i priority. The highest priority is the biggest value.

34.5.2.1.4 Round Robin

Round robin arbitrates between the highest priority queues selected by the strict arbiter. The round robin operates as the one in the data reception descriptor store. See (2) Round Robin (RR) for more information.

34.5.2.2 AXI Master Interface

The AXI master interface handles the data/AXI descriptor exchange with the CPU. It receives the queue number of the descriptor queue number authorized to transmit by the TX queue arbitration, reads the corresponding data from the CPU RAM, and sends the data to TS data store. To do so, it uses GWTRCi.TSRj, GWDCCi.EDE, GWDCCi.ETS, and GWDCCi.SM, and can be monitored using GWAARSS, GWAARSR0, and GWAARSR1 interrupt registers.

Functions:

- GWTRCi.TSRj register is used to start transmission when a descriptor queue is ready.
- GWDCCi.EDE and GWDCCi.ETS registers set the descriptor format that will be used for descriptor queue number i. See section 34.5.1.2.1. General Formats.
- GWDCCi.SM register sets how GWCA will write back descriptors. In this section all explanation will use GWDCCi.SM equal to 00. For GWDCCi.SM equal to 01 or 10 see the register explanation.
- For GWAARSS, GWAARSR0, and GWAARSR1 registers, see section 34.5.1.6. AXI Address RAM Searching.

Restrictions:

- SW: Only basic descriptors and extended descriptors are authorized for data transmission. See section 34.5.1.2.1. General Formats.

Data transmission can be done using linear or cyclic descriptor queues (see section 34.5.1.5. AXI Descriptor Queue Format). Data transmission process descriptors can be set two different ways, through the basic data transmission descriptor chain (section 34.5.2.2.1. Basic Data Transmission) and the time-controlled data transmission descriptor chain (section 34.5.2.2.2. Time-controlled Data Transmission).

34.5.2.2.1 Basic Data Transmission

Basic data transmission process descriptor utilization is described in Figure 34.34. Basic data transmission process descriptor format is described in Figure 34.35 (for basic descriptor, section 34.5.1.2.1. General Formats) and Table 34.12.

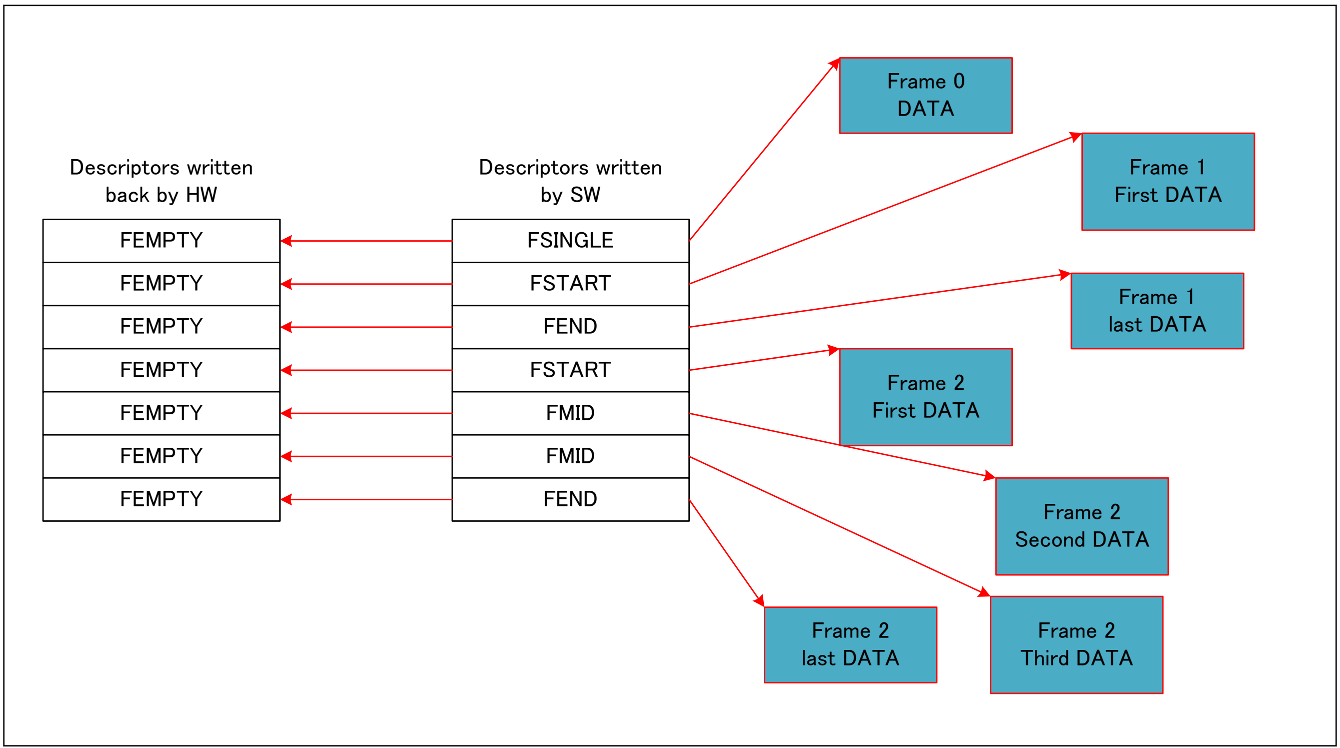


Figure 34.34 Data transmission process descriptor utilization

Note:
- If needed, the SW can split a frame between several descriptors. The first descriptor should be a FSTART descriptor, the last descriptor should be a FEND descriptor and all other descriptors should be FMID descriptors.
- When data is ready, CPU should set the corresponding GWTRCi.TSRj bit to start transfer. See GWTRCi.TSRj description.
- GWTRCi.TSRj can be set anytime even if data is not available. The GWCA will in this case read the terminate descriptor and clear GWTRCi.TSRj bit.

Restrictions:

- SW: When a frame is split, the FSTART descriptor should be written after FMIDs and FEND descriptor for a given frame.

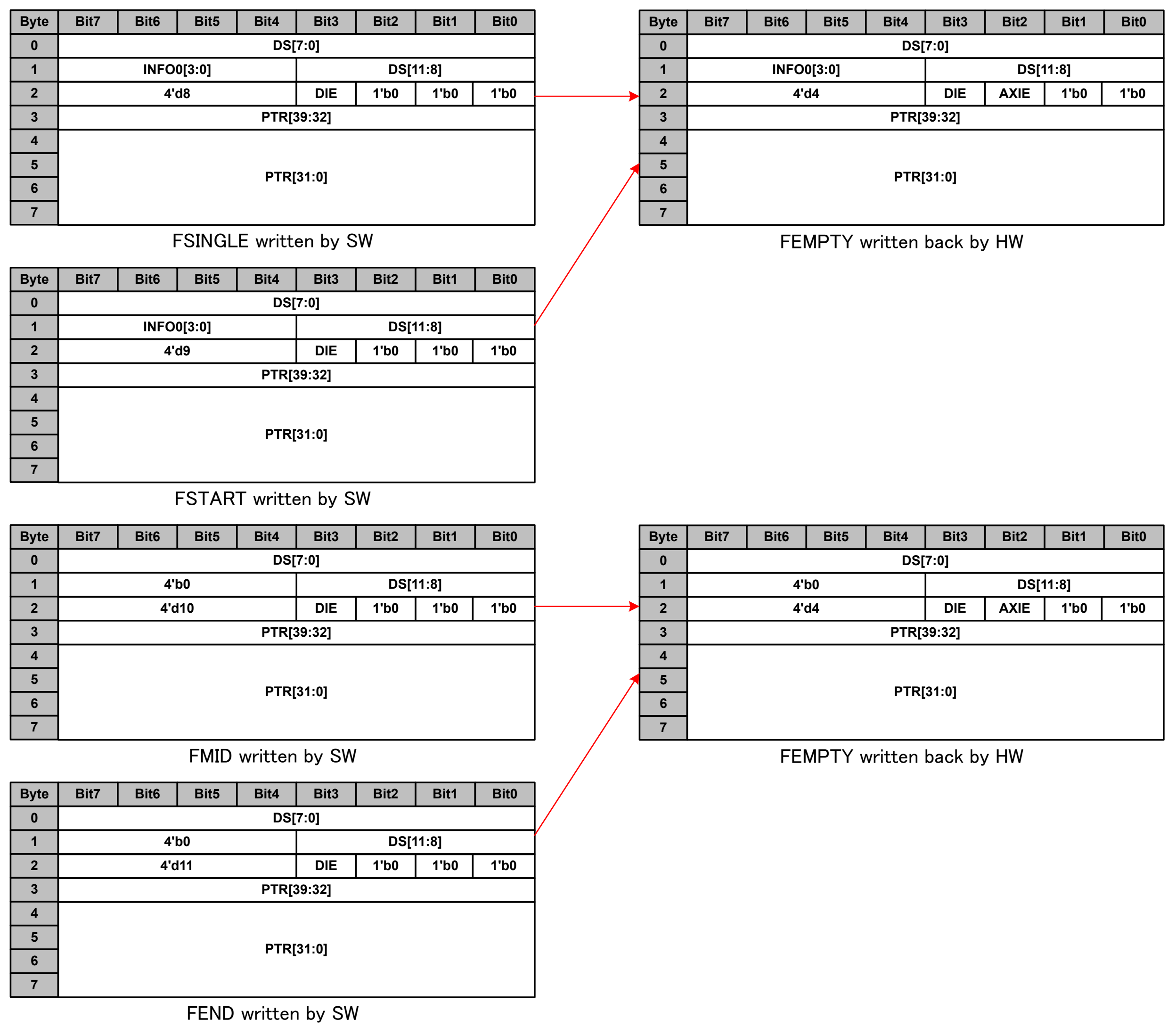


Figure 34.35 Data transmission process descriptor utilization format

Table 34.12 Data transmission process descriptor utilization field description (1 of 2)

Field name	Field description written by SW	Field description after HW write-back
DS	Data what has been written to the URAM by SW	DS (Value is copied from read descriptor)
INFO0	For FSINGLE and FSTART see section 34.5.2.2.3. Data Transmission Common Descriptor Format 0 for FMID and FEND.	INFO0 (Value is copied from read descriptor)
ERR	0	0
DSE	0	0
AXIE	0	Set conditions: ● HW: An AXI error happened while reading a data (RRESP ! = 00).
DIE	DIE	DIE (Value is copied from read descriptor)
DT	8 for FSINGLE 9 for FSTART 10 for FMID 11 for FEND	4 for FEMPTY
PTR	Address where data is written in the URAM	PTR (Value is copied from read descriptor)

Table 34.12 Data transmission process descriptor utilization field description (2 of 2)

Field name	Field description written by SW	Field description after HW write-back
INFO1 (GWDCCi.EDE == 1)	For FSINGLE and FSTART see section 34.5.2.2.3. Data Transmission Common Descriptor Format 0 for FMID and FEND.	NA: INFO1 not written back for transmission
TS (GWDCCi.ETS == 1)	NA: No timestamp descriptor for transmission	NA: No timestamp descriptor for transmission

34.5.2.2.2 Time-controlled Data Transmission

Time-controlled data transmission process descriptor utilization is described in Figure 34.36. Time-controlled data transmission process descriptor format is described in section 34.4.2.11. Global Rate Limiter Setting Flow (for basic descriptor, section 34.5.1.2.1. General Formats) and Table 34.13. FSINGLE, FSTART, FMID, FEND and FEMPTY formats are not described in this section, for their description see Figure 34.35 and Table 34.12.

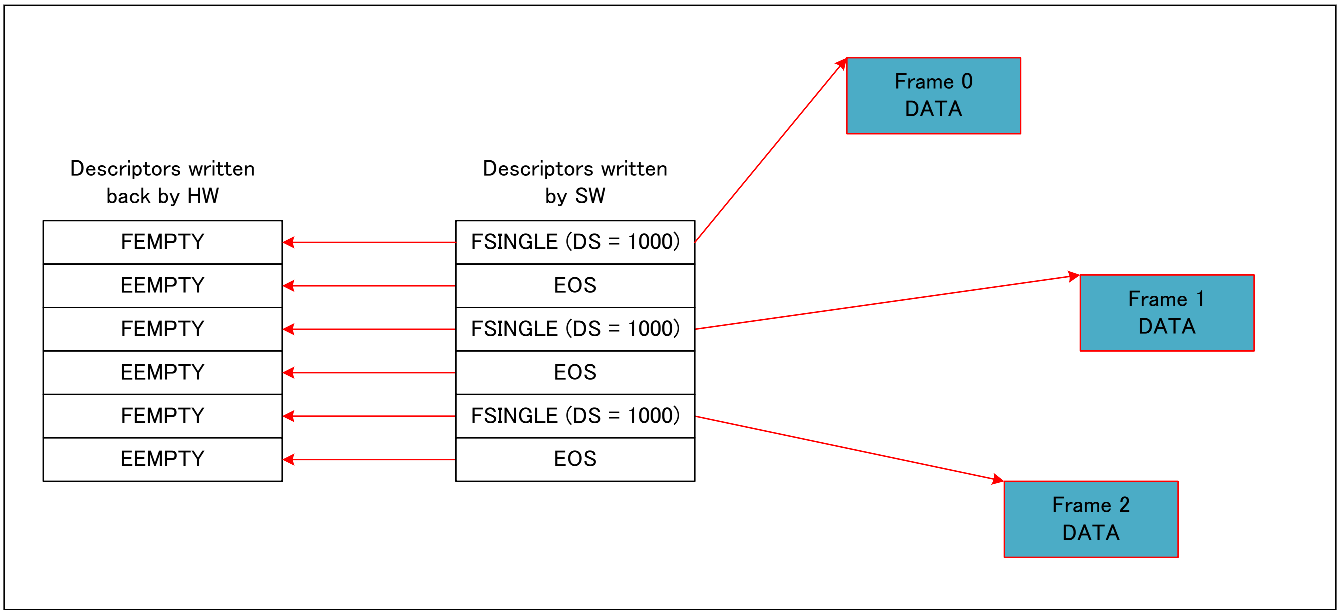


Figure 34.36 Data transmission process descriptor utilization

Note:
- If needed, the SW can split a frame between several descriptors. The first descriptor should be a FSTART descriptor, the last descriptor should be a FEND descriptor and all other descriptors should be FMID descriptors.
- When data is ready, CPU should set the corresponding GWTRCi.TSRj bit to start transfer. See GWTRCi.TSRj description.
- Reading an EOS descriptor will have per effect to clear GWTRCi.TSRj bit. So, if the CPU set GWTRCi.TSRj bit at a cyclic timing, CPU can control the data throughput (If TSR is already set when the CPU tries to set it the next time, only one frame will be sent, so a decrease in throughput can happen).
- GWTRCi.TSRj bit can be set automatically using gPTP timer [gPTP]. See section 34.5.5. gPTP Triggered Transmission.
- GWTRCi.TSRj can be set anytime even if data is not available. The GWCA will in this case read the terminate descriptor and clear GWTRCi.TSRj bit.

Restrictions:

- SW: When a frame is split, the FSTART descriptor should be written after FMIDs and FEND descriptor for a given frame.

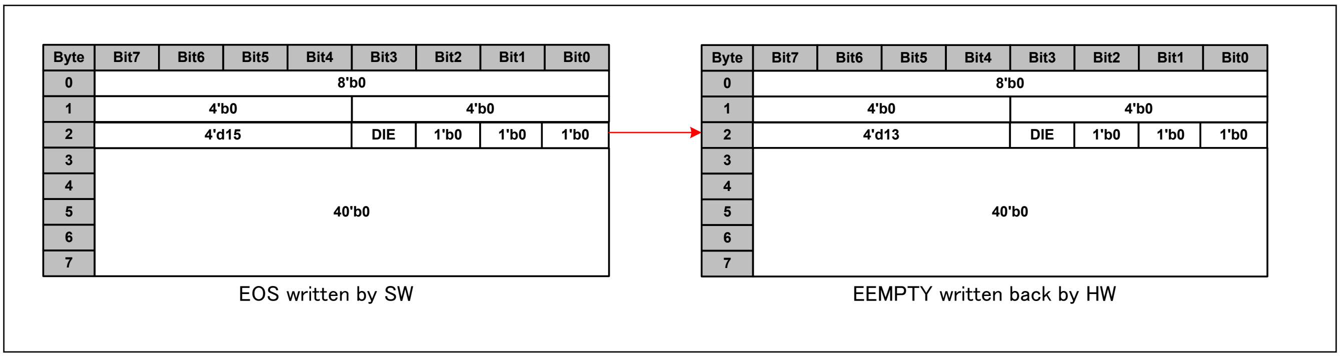


Figure 34.37 Time-controlled data transmission process descriptor utilization format

Table 34.13 Time-controlled data transmission process descriptor utilization field description

Field name	Field description written by SW	Field description after HW write-back
DS	0	DS (Value is copied from read descriptor)
INFO0	0	INFO0 (Value is copied from read descriptor)
ERR	0	0
DSE	0	0
AXIE	0	0
DIE	DIE	DIE (Value is copied from read descriptor)
DT	15	13
PTR	0	PTR (Value is copied from read descriptor)
INFO1 (GWDCCi.EDE == 1)	0	NA: INFO1 not written back for EOS descriptors
TS (GWDCCi.ETS == 1)	NA: No timestamp descriptor for transmission	NA: No timestamp descriptor for transmission

34.5.2.2.3 Data Transmission Common Descriptor Format

In this section are described the fields which are common for any data transmission (INFO0 and INFO1). There are two types of formats, the transmission direct descriptor and the transmission ethernet descriptor formats. The transmission direct descriptor could be used, for example, for non-routable protocol forwarding (network control) or for CPU-CPU communication and the transmission ethernet descriptor could be used for all normal ethernet transmissions. All the fields contained in AXI descriptors, if quoted, will be written ADESCR.{Field name}.

(1) Transmission direct descriptor

Transmission direct descriptor is used by a CPU to send frames by deciding their target directly (Forwarding engine processing will be by-passed [FWD]). Transmission direct descriptor INFO formats are described in Figure 34.38 and Table 34.14.

Restrictions:

- SW: The Transmission direct descriptor can only be and extended descriptor, see section 34.5.1.2.1. General Formats.

INFO0

<table>
<tr><th>Byte</th><th>Bit3</th><th>Bit2</th><th>Bit1</th><th>Bit0</th></tr>
<tr><td>0</td><td colspan="2">—</td><td>SEC</td><td>FI</td></tr>
</table>

INFO1

<table>
<tr><th>Byte</th><th>Bit7</th><th>Bit6</th><th>Bit5</th><th>Bit4</th><th>Bit3</th><th>Bit2</th><th>Bit1</th><th>Bit0</th></tr>
<tr><td>0</td><td>—</td><td>TN</td><td>—</td><td>—</td><td>TXC</td><td>FMT</td><td>SEC</td><td>FI</td></tr>
<tr><td>1</td><td colspan="8">TSUN[7:0]</td></tr>
<tr><td>2</td><td colspan="8">RN[7:0]</td></tr>
<tr><td>3</td><td>FW</td><td colspan="3">IPV[2:0]</td><td>RV</td><td colspan="3">—</td></tr>
<tr><td>4</td><td>—</td><td colspan="7">CSD0[6:0]</td></tr>
<tr><td>5</td><td colspan="8">—</td></tr>
<tr><td>6</td><td>—</td><td colspan="7">DV[6:0]</td></tr>
<tr><td>7</td><td colspan="8">—</td></tr>
</table>

Figure 34.38 Transmission direct descriptor INFO formats

Table 34.14 Transmission direct descriptor INFO field descriptions

Field name	Bit width	Description
FI	1	FCS in.
SEC	1	Secure descriptor
FMT	1	Descriptor format Restrictions: ● SW: Fixed to 1 for Direct descriptor.
TXC	1	TX Timestamp capture [RMAC]. Used to set corresponding bit in RMAC TX interface.
TN	1	Timer utilized for capture [RMAC]. Used to set corresponding bit in RMAC TX interface.
TSUN	8	Timestamp unique number [RMAC]. Used to set corresponding bit in RMAC TX interface.
RV	1	Routing valid. See section 34.5.3.2. L2/L3 Update, to ethernet agent L2/3 update [ETHA] and to forwarding engine L3 routing table [FWD].
RN	5	Routing valid. See section 34.5.3.2. L2/L3 Update, to ethernet agent L2/3 update [ETHA] and to forwarding engine L3 routing table [FWD].
IPV	3	Internal priority value Target priority of the frame.
FW	1	The FCS contained in the frame is wrong (It will be corrected in the switch).
CSD0	6	CPU sub destination for GWCA0
DV	4	Destination vector [MFWD]
—	—	Reserved area Restrictions: ● SW: these fields should be set to 0.

(2) Transmission ethernet descriptor

Transmission ethernet descriptor is used by a CPU to send frames that will be forwarded by the Forwarding engine [FWD]. Transmission ethernet descriptor INFO formats are described in Figure 34.39 and Table 34.15.

Restrictions:

- SW: Because FW field (see (1) Transmission direct descriptor) is not present in an Ethernet descriptor. Any transmit frame should have either a correct FCS or no FCS.

INFO0

Byte	Bit3	Bit2	Bit1	Bit0
0	—			FI

INFO1

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	—	TN	—	—	TXC	FMT	—	FI
1	TSUN[7:0]							
2	—							
3								
4								
5								
6								
7								

Figure 34.39 Transmission ethernet descriptor INFO formats

Table 34.15 Transmission ethernet descriptor INFO field descriptions

Field name	Bit width	Description
FI	1	FCS in.
FMT	1	Descriptor format Restrictions: ● SW: Fixed to 0 for Ethernet descriptors.
TXC	1	TX Timestamp capture [RMAC]. Used to set corresponding bit in RMAC TX interface.
TN	1	Timer utilized for capture [RMAC]. Used to set corresponding bit in RMAC TX interface.
TSUN	8	Timestamp unique number [RMAC]. Used to set corresponding bit in RMAC TX interface.
—	—	Reserved Restrictions: ● SW: these fields should be set to 0.

34.5.2.3 TX Data Store

TX data store has follow functions, it extracts TAG information for frames, applies VLAN tagging, and save the frames in the local RAM. These functions are applied to all frames independently from their descriptor format.

34.5.2.3.1 TAG Information Extraction

TAG information extraction is done in HW without setting intervention. The extracted format can be filtered using GWTTFC register. TAG information extraction is described in Figure 34.40 along with GWTTFC register bit explanation for TAG filtering. The extracted fields are SN, CoS-TCI, C-TCI and S-TCI.

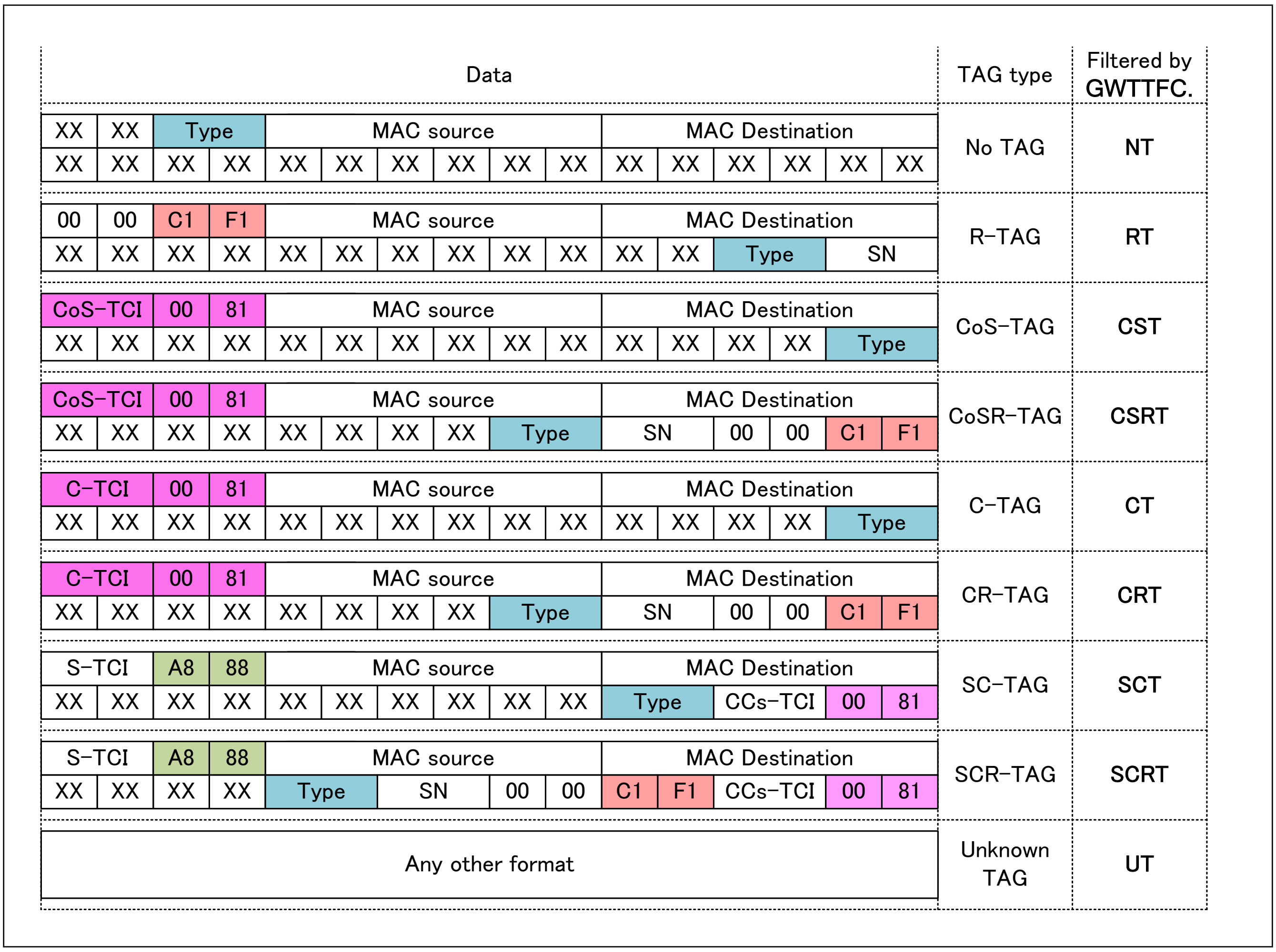


Figure 34.40 TAG information extraction

Note:
- Only colored part in are considered for TAG information extraction.
- HF1C1 corresponds to R-TAG TPID and SN correspond to R-TAG sequence number [802.1CB].
- H8100 corresponds to C-TAG TPID, C-TCI corresponds to a C-TAG TCI with a non-null VID, CoS-TCI corresponds to a C-TAG TCI with a null VID and CCs-TCI corresponds to C-TCI or CoS-TCI [802.1Q].
- H88A8 corresponds to S-TAG TPID and S-TCI corresponds to an S-TAG TCI [802.1Q].
- Type corresponds to any Ethernet type value which is not HF1C1, H88A8 or H8100.
- TPID values can be configured in Forwarding Engine [FWD]. Values for TPIDs can be set in forwarding engine (in FWTTC0 and FWTTC1 registers).

Restrictions:

- HW: Because 32 bytes are needed for TAG extraction, GWCA will reject transmit frames smaller than 32 bytes and set GWEIS0.USMFSES interrupt register.
- SW: In R-TAG, the reserved part should always be set to 0. If the reserved part is not set to 0 and corresponding frame contains an FCS, the frame could be forwarded with an error.

34.5.2.3.2 VLAN Tagging

VLAN tagging replaces or overwrites present TAGs in a frame using the format given by TAG information extraction (section 34.5.2.3.1. TAG Information Extraction), switch VLAN mode (FWGC.SVM register in MFWD) and, GWVCC and GWVTC registers. After VLAN tagging, any frame has three TAGs, one R-TAG, one C-TAG and one S-TAG which will be directly saved in the TAG RAM [MFAB].

Functions:

- GWVCC.VIM register sets the TX data path in Port based VLAN (all incoming frames are sent to the forwarding engine with the same VLAN ID). GWVCC.VIM setting is ignored when the switch in is No VLAN mode (FWGC.SVM register in forwarding engine set to 00) [MFWD].
- GWVTC register is used to set the HW TAGs. There are three possible HW TAGs. The HW C-TAG made from GWVTC.CTV as VID, GWVTC.CTP as PCP and GWVTC.CTD as DEI. The HW S-TAG made from GWVTC.STV as VID, GWVTC.STP as PCP and GWVTC.STD as DEI. The HW CoS-TAG made from GWVTC.CTV as VID, the incoming frame PCP as PCP and the incoming frame DEI as DEI.

TAGs after VLAN tagging are described in Figure 34.41, Figure 34.42 and Figure 34.43 depending on Switch VLAN mode (FWGC.SVM register in MFWD) and the type of frame given by TAG information extraction (section 34.5.2.3.1. TAG Information Extraction). In the figures, the data extracted from the incoming frames are in blue and the added HW TAGs are in red.

TAG type	R-TAG	C-TAG	S-TAG
No TAG	0	0	0
R-TAG	SN	0	0
CoS-TAG	0	CoS-TCI	0
CoSR-TAG	SN	CoS-TCI	0
C-TAG	0	C-TCI	0
CR-TAG	SN	C-TCI	0
SC-TAG	0	C-TCI	S-TCI
SCR-TAG	SN	C-TCI	S-TCI
Unknown TAG	0	0	0

Figure 34.41 TAGs after VLAN tagging: Switch in No-VLAN mode [MFWD]

	Incoming VLAN mode (**GWVCC.VIM** == 1'b0)			Port based VLAN mode (**GWVCC.VIM** == 1'b1)		
TAG type	R-TAG	C-TAG	S-TAG	R-TAG	C-TAG	S-TAG
No TAG	0	HW C-TAG	0	0	HW C-TAG	0
R-TAG	SN	HW C-TAG	0	SN	HW C-TAG	0
CoS-TAG	0	HW CoS-TAG	0	0	HW CoS-TAG	0
CoSR-TAG	SN	HW CoS-TAG	0	SN	HW CoS-TAG	0
C-TAG	0	C-TCI	0	0	HW C-TAG	0
CR-TAG	SN	C-TCI	0	SN	HW C-TAG	0
SC-TAG	0	C-TCI	S-TCI	0	HW C-TAG	0
SCR-TAG	SN	C-TCI	S-TCI	SN	HW C-TAG	0
Unknown TAG	0	HW C-TAG	0	0	HW C-TAG	0

Figure 34.42 TAGs after VLAN tagging: Switch in C-TAG mode [MFWD]

	Incoming VLAN mode (**GWVCC.VIM** == 1'b0)			Port based VLAN mode (**GWVCC.VIM** == 1'b1)		
TAG type	R-TAG	C-TAG	S-TAG	R-TAG	C-TAG	S-TAG
No TAG	0	HW C-TAG	HW S-TAG	0	HW C-TAG	HW S-TAG
R-TAG	SN	HW C-TAG	HW S-TAG	SN	HW C-TAG	HW S-TAG
CoS-TAG	0	HW CoS-TAG	HW S-TAG	0	HW CoS-TAG	HW S-TAG
CoSR-TAG	SN	HW CoS-TAG	HW S-TAG	SN	HW CoS-TAG	HW S-TAG
C-TAG	0	C-TCI	HW S-TAG	0	C-TCI	HW S-TAG
CR-TAG	SN	C-TCI	HW S-TAG	SN	C-TCI	HW S-TAG
SC-TAG	0	C-TCI	S-TCI	0	C-TCI	HW S-TAG
SCR-TAG	SN	C-TCI	S-TCI	SN	C-TCI	HW S-TAG
Unknown TAG	0	HW C-TAG	HW S-TAG	0	HW C-TAG	HW S-TAG

Figure 34.43 TAGs after VLAN tagging: Switch in SC-TAG mode [MFWD]

34.5.2.3.3 Frame Saving in Local RAM

For frame saving in the local RAM, see Fabric specification document [MFAB], section “Data representation in RAMs”. Only the frame descriptor mentioned in Fabric specification document [MFAB] will be described in this section.

There are two types of Local RAM descriptors, direct local descriptors and the ethernet local descriptors. They are respectively created from direct descriptors and ethernet descriptors, see section 34.5.2.2.3. Data Transmission Common Descriptor Format.

All the fields in these descriptors, if quoted, will be written LDESCR.{Field name}.

(1) Direct local descriptor

Direct local descriptor format is described in Figure 34.44 and Table 34.16.

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	—	TN	CRT	IET	TXC	FMT	SEC	FI
1	TSUN[7:0]							
2	SAEF[7:0]							
3	—				RTGI	VCTRL[2:0]		
4	RN[7:0]							
5	FW	IPV[2:0]			RV	—		
6	TPL[15:0]							
7								
8	—	CSD0[6:0]						
9	—	CSD1[6:0]						
10	—	DV[6:0]						
11	—							
12								
13								
14								
15								

Figure 34.44 Direct local descriptor format

Table 34.16 Direct local descriptor field description

Field name	Bit width	Description	Value for GWCA
FI	1	See Table 34.14.	Copied from transmission direct descriptor. See (1) Transmission direct descriptor.
SEC	1	See Table 34.14.	Copied from transmission direct descriptor. See (1) Transmission direct descriptor.
FMT	1	See Table 34.14.	Copied from transmission direct descriptor. See (1) Transmission direct descriptor.
TXC	1	See Table 34.14.	Copied from transmission direct descriptor. See (1) Transmission direct descriptor.
IET	1	See Table 34.14.	Copied from transmission direct descriptor. See (1) Transmission direct descriptor.
CRT	1	See Table 34.14.	Copied from transmission direct descriptor. See (1) Transmission direct descriptor.
TN	1	See Table 34.14.	Copied from transmission direct descriptor. See (1) Transmission direct descriptor.
TSUN	8	See Table 34.14.	Copied from transmission direct descriptor. See (1) Transmission direct descriptor.
SAEF	8	Source agent error flags	See (3) SAEF format.
VCTRL	3	VLAN control	VCTRL[2] values: ● Set to GWVCC.VIM. VCTRL[1:0] values ● 00: For No TAG, R-TAG and Unknown TAG frames, see section 34.5.2.3.1. TAG Information Extraction. ● 01: For C-TAG and CR-TAG frames, see section 34.5.2.3.1. TAG Information Extraction. ● 10: For SC-TAG and SCR-TAG frames, see section 34.5.2.3.1. TAG Information Extraction. ● 11: For CoS-TAG and CoSR-TAG frames, see section 34.5.2.3.1. TAG Information Extraction.
RTGI	1	R-TAG in	Values: ● 0: For No TAG, CoS-TAG, C-TAG, SC-TAG and Unknown TAG frames, see section 34.5.2.3.1. TAG Information Extraction. ● 1: For R-TAG, CoSR-TAG, CR-TAG and SCR-TAG frames, see section 34.5.2.3.1. TAG Information Extraction.
RV	1	See Table 34.14.	Copied from transmission direct descriptor. See (1) Transmission direct descriptor.
RN	5	See Table 34.14.	Copied from transmission direct descriptor. See (1) Transmission direct descriptor.
IPV	3	See Table 34.14.	Copied from transmission direct descriptor. See (1) Transmission direct descriptor.
FW	1	See Table 34.14.	Copied from transmission direct descriptor except when set by checksum handling function. See (1) Transmission direct descriptor for transmission direct descriptor and see 8.2.3.4.
TPL	16	Total payload length	Payload length of frame. It only includes the data saved in the data RAM [MFAB].
CSD0	6	See Table 34.14.	Copied from transmission direct descriptor. See (1) Transmission direct descriptor.
DV	4	See Table 34.14.	Copied from transmission direct descriptor. See (1) Transmission direct descriptor.
—	—	Reserved area	Set to 0

(2) Ethernet local descriptor

Ethernet local descriptor format is described in Figure 34.45 and Table 34.17.

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	—	TN	CRT	IET	TXC	FMT	—	FI
1	TSUN[7:0]							
2	SAEF[7:0]							
3	—				RTGI	VCTRL[2:0]		
4	—							
5								
6	TPL[15:0]							
7								
8	TSNS[29:0]							
9								
10								
11	TSD	TSV						
12	TSS[31:0]							
13								
14								
15								

Figure 34.45 Ethernet local descriptor format

Table 34.17 Ethernet local descriptor field description (1 of 2)

Field name	Bit width	Description	Value for GWCA
FI	1	See Table 34.15.	Copied from transmission ethernet descriptor. See (2) Transmission ethernet descriptor.
FMT	1	See Table 34.15.	Copied from transmission ethernet descriptor. See (2) Transmission ethernet descriptor.
TXC	1	See Table 34.15.	Copied from transmission ethernet descriptor. See (2) Transmission ethernet descriptor.
IET	1	See Table 34.15.	Copied from transmission ethernet descriptor. See (2) Transmission ethernet descriptor.
CRT	1	See Table 34.15.	Copied from transmission ethernet descriptor. See (2) Transmission ethernet descriptor.
TN	1	See Table 34.15.	Copied from transmission ethernet descriptor. See (2) Transmission ethernet descriptor.
TSUN	8	See Table 34.15.	Copied from transmission ethernet descriptor. See (2) Transmission ethernet descriptor.
SAEF	8	Source agent error flags	See (3) SAEF format.
VCTRL	3	VLAN control	VCTRL[2] values: ● Set to GWVCC.VIM. VCTRL[1:0] values ● 00: For No TAG, R-TAG and Unknown TAG frames, see section 34.5.2.3.1. TAG Information Extraction. ● 01: For C-TAG and CR-TAG frames, see section 34.5.2.3.1. TAG Information Extraction. ● 10: For SC-TAG and SCR-TAG frames, see section 34.5.2.3.1. TAG Information Extraction. ● 11: For CoS-TAG and CoSR-TAG frames, see section 34.5.2.3.1. TAG Information Extraction.

Table 34.17 Ethernet local descriptor field description (2 of 2)

Field name	Bit width	Description	Value for GWCA
RTGI	1	R-TAG in	Values: ● 0: For No TAG, CoS-TAG, C-TAG, SC-TAG and Unknown TAG frames, see section 34.5.2.3.1. TAG Information Extraction. ● 1: For R-TAG, CoSR-TAG, CR-TAG and SCR-TAG frames, see section 34.5.2.3.1. TAG Information Extraction.
TPL	16	Total payload length	Payload length of frame. It only includes the data saved in the data RAM [MFAB].
TSV	1	Timestamp valid	Set to 0. Unused in GWCA.
TSD	1	Timestamp default	Set to 0. Unused in GWCA.
TSNS	30	Timestamp nanosecond [gPTP]	Set to 0. Unused in GWCA.
TSS	32	Timestamp second [gPTP] Restrictions: ● HW: The 16-upper bits of the gPTP second part are truncated.	Set to 0. Unused in GWCA.
—	—	Reserved area	Set to 0

(3) SAEF format

Source agent error flag format is described in Figure 34.46 and Table 34.18.

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	—	DNE	—	SEQE	AXE	—	—	TFE

Figure 34.46 Source agent error flag format

Table 34.18 Source agent error flag field description

Field name	Bit width	Description
TFE	1	TAG Filtering Error Set conditions: ● HW: When a frame does not fit the required TAG ingress format. See GWTTFC register explanation.
AXE	1	AXI Error Set conditions: ● HW: When an AXIE error happened during a split frame, the frame is sent incomplete.
SEQE	1	Sequence Error Set conditions: ● HW: When a FSINGLE, FSTART, EOS or unknown descriptor is received during a split frame, the frame is sent incomplete.
DNE	1	Descriptor Number Error ● HW: When a number GWMDNC.TXDMN+1 of descriptor has been already read to fetch a frame during a spit frame and the frame is not finished. The frame is sent incomplete.
—	—	Reserved Set to 0

34.5.3 Data Reception

GWCA allows data reception through the GWCA RX data path, The RX data path is described in section 34.4.2.11. Global Rate Limiter Setting Flow.

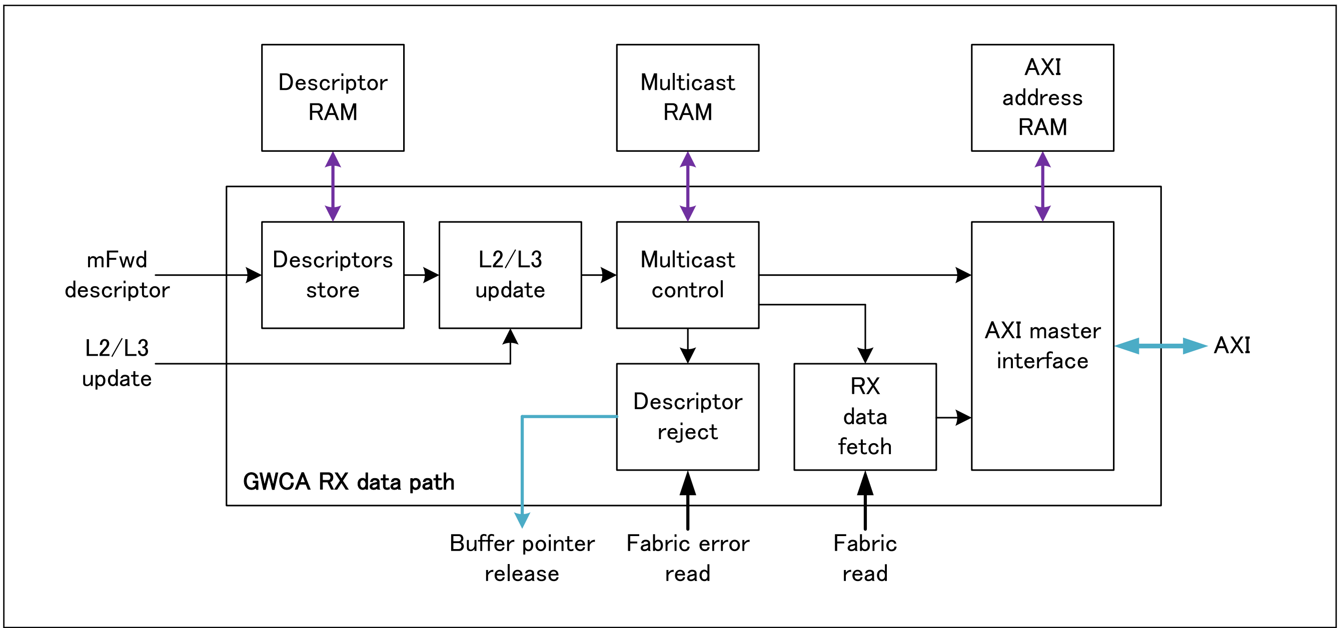


Figure 34.47 GWCA RX data path block diagram

The RX data path is separated in seven blocks:

- Descriptor store: This block aims at storing the descriptor received from the MFWD in the Descriptor RAM and controls the arbitration between descriptors.
- L2/L3 update: This block aims at fetching the L2/L3 update rules from the Forwarding Engine to update the frame while sending it.
- Frame size check: This block aims at checking the frame size and security level and decide to forward or discard frames.
- Multicast control: This block aims at checking the frame size and security level and decide to forward or discard frames and at copying the descriptors to be able to send the data at several CPU sub-destinations.
- Descriptor reject: This block aims at releasing the pointer of the frames rejected by Frame size control block. This block is here for switch hardware purpose and its description is not required for switch utilization so it will not be described.
- RX data fetch: This block fetches frames data from the local RAM, update frames depending on the information obtained by L2/L3 update module, on VLAN control information and the R-TAG information, can remove frame FCSs, and release frame pointers. The pointer release is here for switch hardware purpose and its description is not required for switch utilization so it will not be described.
- AXI master interface: This block handles the data/AXI descriptor exchange with the CPU.

34.5.3.1 Descriptor Store

Descriptor store has two functions. Storing the descriptors coming from the descriptor bus MFWD in the descriptor RAM and arbitrating descriptor read from the descriptor RAM to decide in which order the descriptors will be sent to the CPU.

34.5.3.1.1 Descriptor Storage

Descriptor storage controls the descriptor storage using GWIRC.IPVRi, GWRDQSC.RDQSL, GWRDQC.RDQD, GWRDQC.RDQP and GWRDQDCq.DQD registers and can be monitored using GWRDQMq.DNQ and GWRDQMLMq.DMLQ registers, GWEIS1.DQOES and GWEIS1.DQSES interrupt registers. Figure 34.48 describes an example for Descriptor storage setting.

Functions:

- GWIRC.IPVRi register is used to map descriptor received from forwarding engine with a given FDESCR.IPV [MFWD] value to a determined descriptor queue. For example, by setting GWIRC.IPVR6 to 1 all the descriptors received with FDESCR.IPV equal to 6 will be stored in descriptor queue number 1.

- GWRDQSC.RDQSLn is used to set a queue's security level. If a queue is secure (GWRDQSC.RDQSLn is set) and a non-secure descriptor is received (FDESCR.SEC [MFWD] is not set) for this queue, the descriptor will not be stored and GWEIS1.DQSESi flag will be set.
- GWRDQC.RDQP register is used to pause descriptor queues. For paused descriptor queues, the descriptor transmission to CPU will stop but the descriptor storage in the descriptor RAM will continue.
- GWRDQC.RDQD register is used to disable descriptor queues. For disabled descriptor queues and no descriptor will be stored in the Descriptor RAM.
- GWRDQDCq.DQD registers are used to set the maximum number of descriptors that can be stored in each descriptor queue. If a descriptor is received for descriptor q while queue q is full (GWRDQDCq.DQD == GWRDQMq.DNQ), the descriptor will not be stored and GWEIS1.DQOESi flag will be set.
- GWRDQMq.DNQ registers are used to monitor the current number of descriptors in each descriptor queue.
- GWRDQMLMq.DMLQ registers are used to monitor the maximum number of descriptors that has been held in each queue since the previous reset or the previous time corresponding register has been read.
- GWEIS1.DQOES register signals that descriptors have been discarded because of a descriptor queue overflow.
- GWEIS1.DQSES register signals that descriptors have been discarded because of a descriptor queue security error.

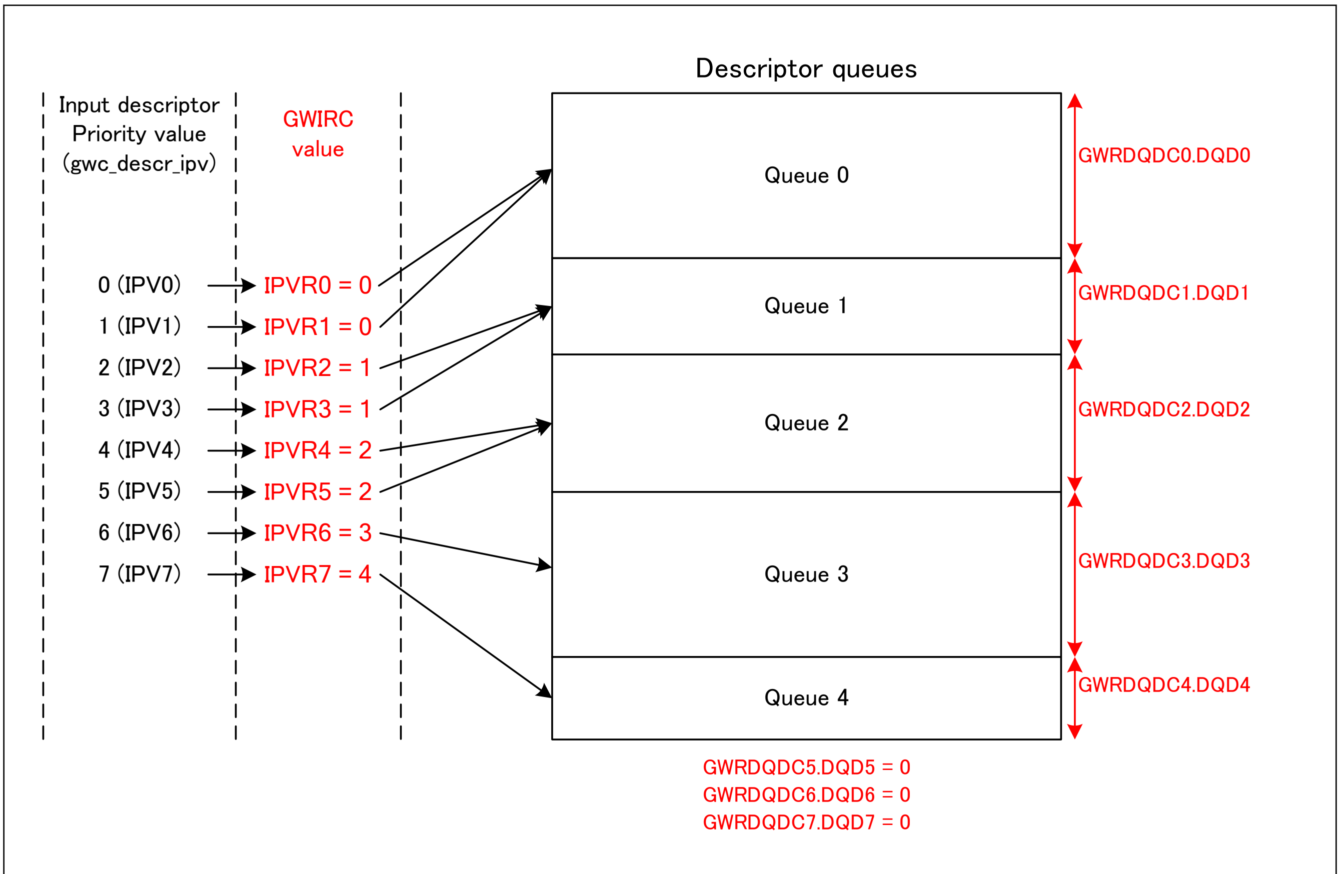


Figure 34.48 Descriptor storage setting example

34.5.3.1.2 Descriptor Arbitration

Descriptor arbitration controls the descriptor reading order for reception using GWRDQAC register. It has 4 modes. The Strict Priority, the Round Robin, the Weight Round Robin and the Hybrid mode. This function is used to ensure QoS when the AXI access allows a throughput which is too small to allow all the data to be stored in the user RAM.

(1) Strict priority (SP)

To set the descriptor arbitration in strict priority mode, all the GWRDQAC.RDQAq (q = 0 to 7) registers should be set to 0. In this case a strict priority algorithm will be applied for descriptor arbitration. The strict priority algorithm is described through an example in Figure 34.49.

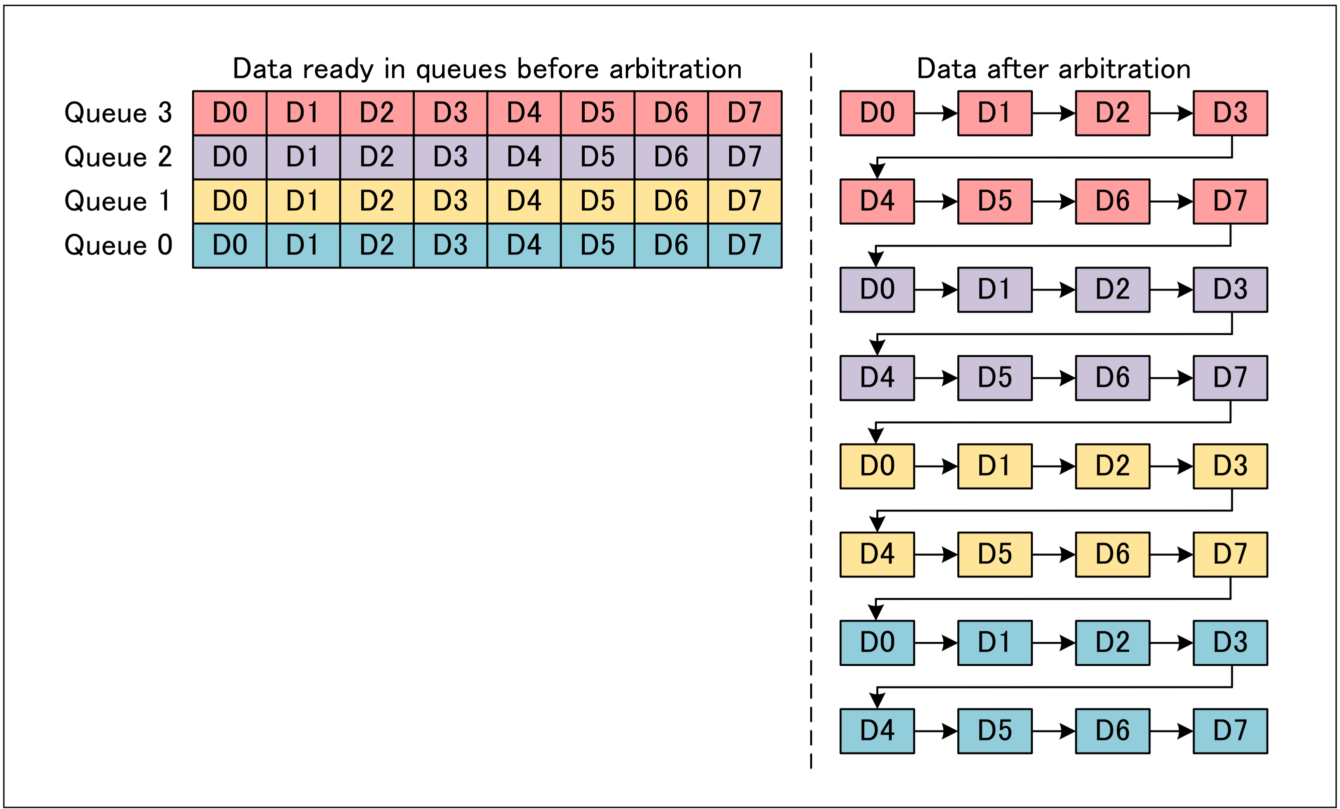


Figure 34.49 Strict priority arbitration example

(2) Round Robin (RR)

To set the descriptor arbitration in round robin mode, all the GWRDQAC.RDQAq (q = 0 to 7) registers should be set to 1. In this case a round robin algorithm will be applied for descriptor arbitration. The round robin algorithm is described through an example in Figure 34.50.

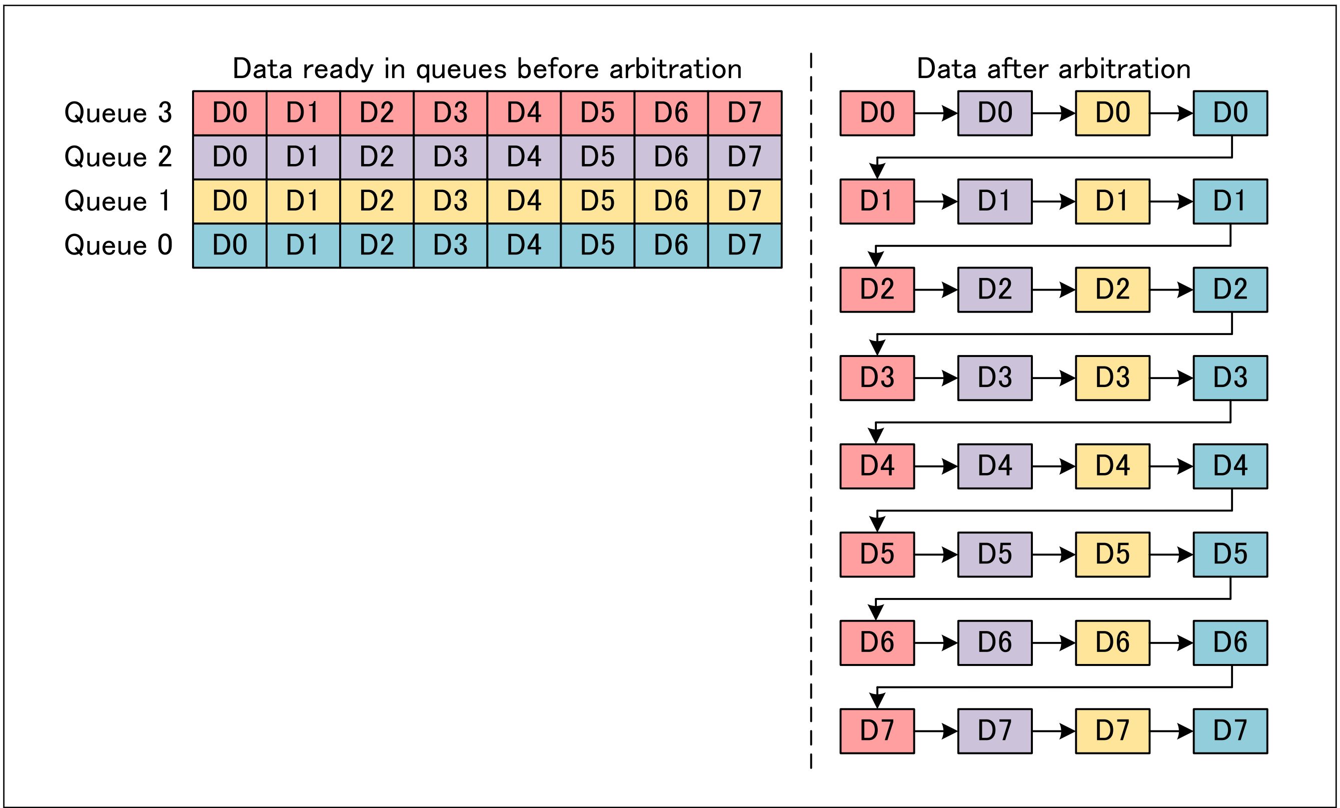


Figure 34.50 Round robin arbitration example

(3) Weight Round Robin (WRR)

To set the descriptor arbitration in weight round robin mode, all the GWRDQAC.RDQAq (q = 0 to 7) registers should be set to a value different than 0 with at least one queue set to a value different than 1. In this case a weight round robin algorithm will be applied for descriptor arbitration. The weight round robin algorithm is described through an example in Figure 34.51 (RDQA0 = 2, RDQA1 = 1, RDQA2 = 3 and RDQA3 = 4). In this arbitration GWRDQAC.RDQAq register represents the weight of queue q. the heavier a queue is the more chance it will have to transmit data.

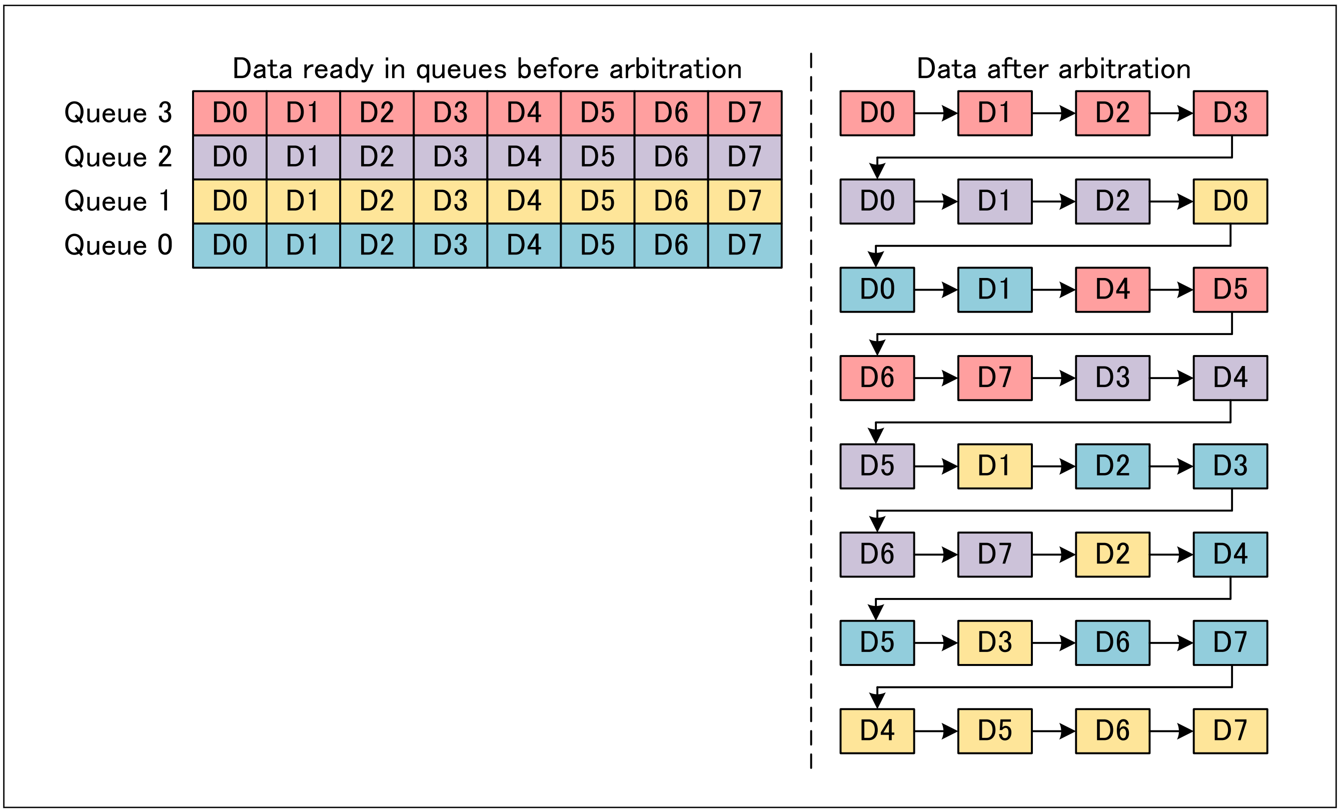


Figure 34.51 Weight round robin arbitration example

(4) Hybrid arbitration

It is allowed to use at the same time SP and RR or SP and WRR. This mode is called the hybrid arbitration mode. To set the descriptor arbitration in hybrid arbitration mode, some of the GWRDQAC.RDQAq (q = 0 to 7) registers should be set to 0 and some others to a value different than 0 (For restrictions see GWRDQAC.RDQAq register explanation). The hybrid arbitration algorithm is described through an example in Figure 34.52 (RDQA0 = 0, RDQA1 = 1, RDQA2 = 2 and RDQA3 = 0).

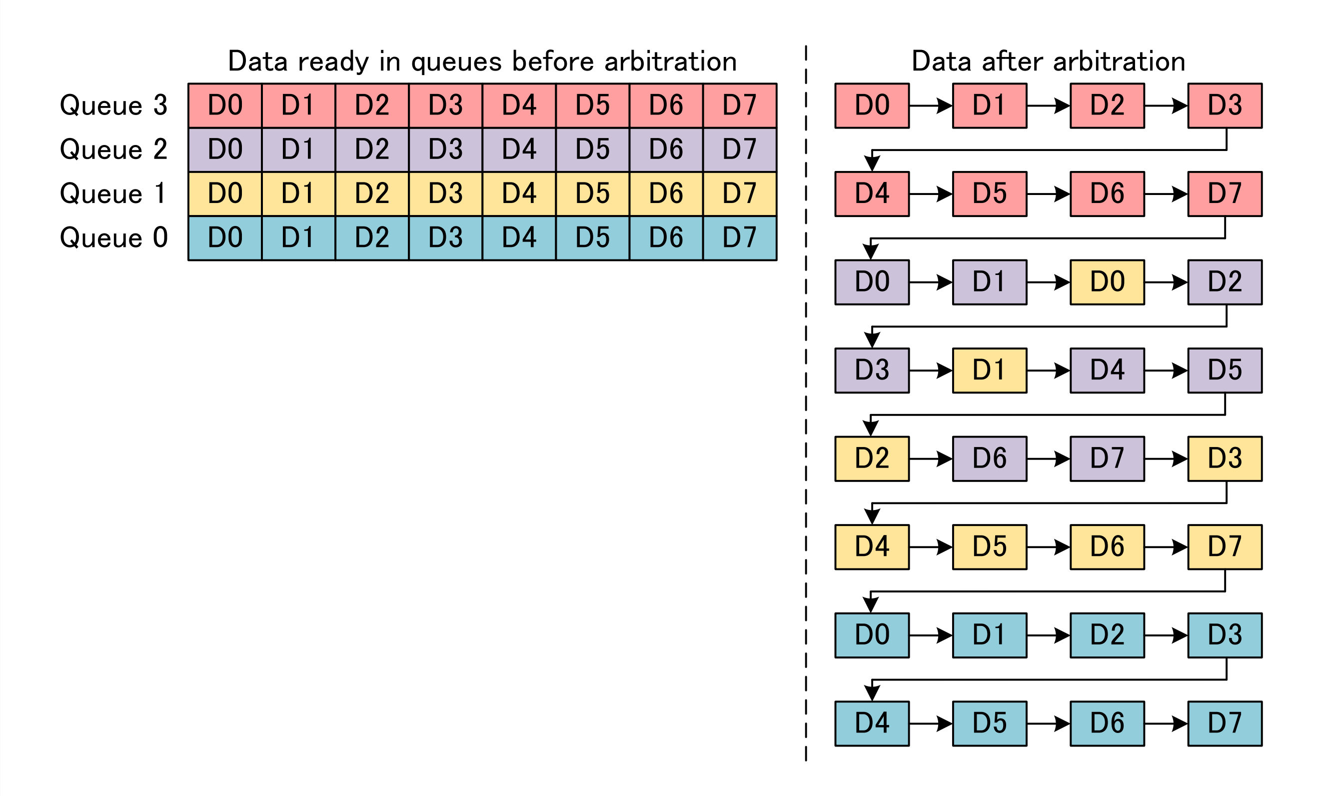


Figure 34.52 Hybrid arbitration example

34.5.3.2 L2/L3 Update

L2/L3 update aims at fetching the update rules thanks to L2/L3 update bus [MFWD].

The L2/L3 update rules will be fetched for any descriptor received with FDESCR.RV set [MFWD]. Rule number FDESCR.RN [MFWD] will be used for L2/L3 update.

Functions:

- For rule fetching setting details see Forwarding Engine specification document [MFWD].
- For frame update information see RX data fetch block explanation in section 34.5.3.4. RX Data Fetch.

34.5.3.3 Multicast Control

Multicast function aims at copying the same frame to several AXI descriptor queue. It allows different CPU to process same frames at the same time without interfering with each other. If nothing is set, all the descriptor queues are initialized as unicast queues.

Multicast control has four functions. Checking the frame size, security level and queue type (TX or RX), copying the descriptor coming from L2/L3 update using the multicast table, modifying the multicast table using multicast learning and reading the multicast table using multicast searching.

34.5.3.3.1 Frame Check

Frame check aims at checking if a frame has the right properties to be received by CPU using GWRMFSCq, GWDCCi.SL, GWDCCi.DQT registers and can be monitored using GWEIS0.FSES, GWEIS4.DSSES, GWEIS4.DSSEIOS, GWEIS4.DSSECN, GWEIS5.DCTES, GWEIS5.DCTEIOS, GWEIS5.DCTECN interrupt registers.

Functions:

- GWRMFSCq registers are used to set the maximum frame size accepted for each descriptor queue. Any frame received for queue q with a size bigger than GWRMFSCq.MFS will be discarded and GWEIS0.FSES flag will be set. The frame size considered is the size after VLAN update and with FCS. In this case, multicast will not happen and the frame will totally be discarded.

- GWDCCi.SL registers are used to set the security level of a CPU sub destination. Any frame received unsecure (FDESCR.SEC [MFWD] is not set) for a secure CPU sub destination (GWDCCi.SL is set) discarded and GWEIS4.DSSES flag will be set (during multicast, for all CPU sub destination for which GWDCCi.SL is set frame will be discarded and for all CPU sub destination for which GWDCCi.SL is not set, frame will be received).
- GWDCCi.DQT is used to control is the received frame is going to a reception descriptor queue. Any frame received for a transmission descriptor queue will be discarded and GWEIS5.DCTES flag will be set. In case of multicast, only the targeted transmission queue will see their data lost.
- GWEIS0.FSES interrupt register signals that a descriptor has been rejected because the corresponding frame where oversized.
- GWEIS4.DSSES, GWEIS4.DSSEIOS and GWEIS4.DSSECN interrupt registers signal that a descriptor has been rejected because of an AXI descriptor queue security error.
- GWEIS5.DCTES, GWEIS5.DCTEIOS, GWEIS5.DCTECN interrupt registers signal that a descriptor has been rejected because of a descriptor chain type error.

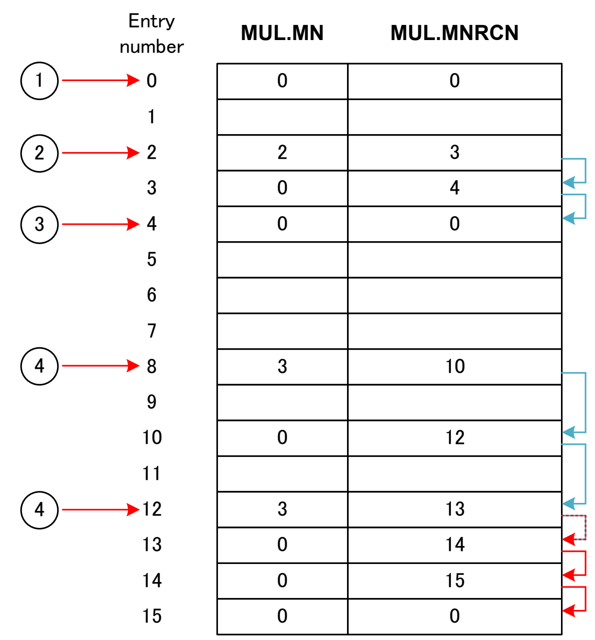
34.5.3.3.2 Descriptor Copy

Descriptor copy aims at multicasting descriptor to send them to the AXI master interface and the RX data fetch using the multicast table.

The multicast table is divided in two fields, MUL.MN[2:0] which is the multicast number and MUL.MNRCN[5:0] which is the next descriptor chain and is used to link AXI descriptor chains together. AXI descriptor chains linked together form a multicast chain to which a frame will be multicast.

Because the multicast table is based on links all the multicast patterns are not possible.

The multicast table utilization is described in Figure 34.53 through an example.



Case	Explanation
1	Unicast: − Input descriptor with CPUSD = 0 will be sent to AXI descriptor chain 0
2	Multicast: − Input descriptor with CPUSD = 2 will be sent to AXI descriptor chain 2,3 and 4.
3	Unicast in a multicast chain: − Input descriptor with CPUSD = 4 will be sent to AXI descriptor chain 4.
4	Multicast chain in a multicast chain: − Input descriptor with CPUSD = 8 will be sent to AXI descriptor chain 8,10,12,13. − Input descriptor with CPUSD = 12 will be sent to AXI descriptor chain 12,13,14,15.

Figure 34.53 Multicast table utilization example

34.5.3.3.3 Multicast Learning

Multicast learning aims at updating entries in the multicast table using GWMSTLS and GWMSTLR registers. Table 34.19 describes mapping between learning registers and fields in multicast table. Table 34.20 describes the learn results, their mapping with the multicast table and how to handle them. It is possible to multicast a frame to 8 AXI descriptor chains.

Functions:

- GWMSTLS register is used to set the information needed to overwrite an entry.

- GWMSTLR register is used to check if the previously set entry has been overwritten.
- To set entries in the multicast table, see software flow in section 34.4.2.7. Multicast Table Setting Flow.
- If GWAC.AMP = 1 (the AXI transaction is pending), learning will not complete. Release the AXI master pose.

Table 34.19 Learn registers/Multicast table mapping registers

Register name	Field name in L3 table
GWMSTLS.MSENL	Not present in multicast table, used to indicate which entry should be overwritten.
GWMSTLS.MNL	MUL.MN, see section 34.5.3.3.2. Descriptor Copy
GWMSTLS.MNRCNL	MUL.MNRCN, see section 34.5.3.3.2. Descriptor Copy

Table 34.20 Learn result

Register name	Field name	Field explanation
GWMSTLR.MTLF	Learning fail	Learning fail will happen in one of the following conditions: ● When an APB tries to learn a multicast entry and the multicast table is not ready (GWMTIRM.MTR is not set)

(1) Static learning

When a system aims at using the multicast table in a static way (set in CONFIG mode only) there is no restriction regarding the entry format, the entry link pointers and the learning order of entries.

(2) Dynamic learning

When a system aims at using the multicast table in a dynamic way (set in OPERATION mode), the restrictions described in Figure 34.54 should be respected.

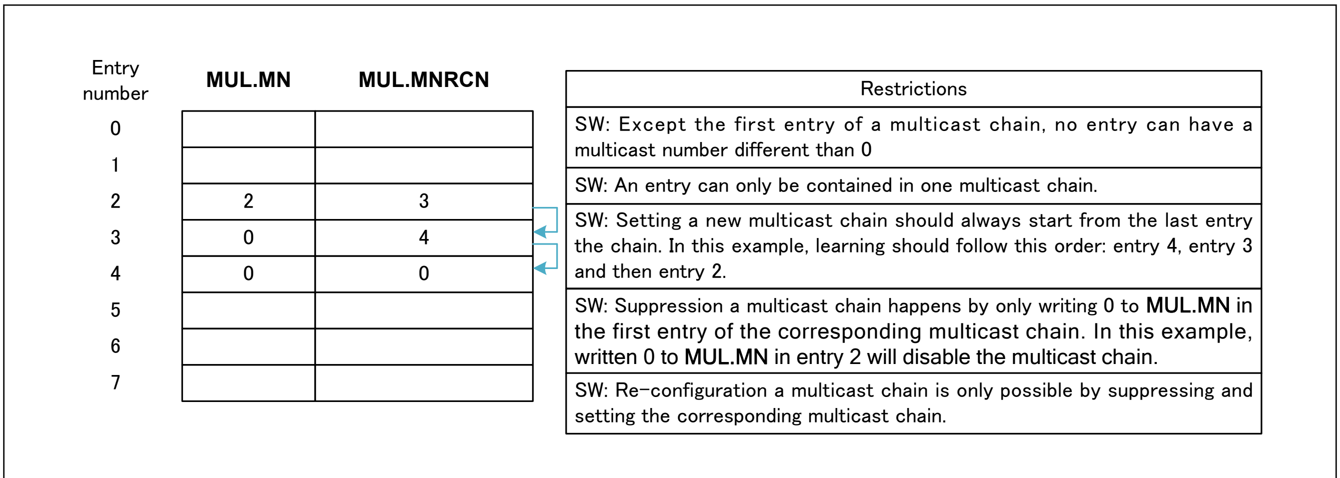


Figure 34.54 Multicast table dynamic utilization restrictions

34.5.3.3.4 Multicast Searching

Multicast searching aims at reading entries in the multicast table using GWMSTSS and GWMSTSR registers. Table 34.21 describes mapping between learning registers and fields in multicast table. Table 34.22

Functions:

- GWMSTSS register is used to set the entry that should be read.
- GWMSTSR register is used to read the read result.
- To read an entry in the multicast table, see software flow in section 34.4.2.9. AXI RAM Reset Flow.

Table 34.21 Search registers/Multicast table mapping registers

Register name	Field name in L3 table
GWMSTSS.MSENS	Not present in multicast table, used to indicate which entry should be read.

Table 34.22 Search result

Register name	Field name/ corresponding field in multicast table	Field explanation
GWMSTSR.MNR	MUL.MN	Multicast table read value
GWMSTSR.MNRCNR	MUL.MNRCN	Multicast table read value

34.5.3.4 RX Data Fetch

RX data fetch has the following functions. Fetching the data from the Local RAM, untagging/ VLANs depending on VLAN settings, inserting an R-TAG, updating the data depending on L2/L3 update information fetched in L2/L3 update, handling FCS, sending the data the AXI master interface and releasing the buffer pointer of the read frames.

The data fetching from Local RAM, the data sending to the AXI master interface and the pointer release are here for switch hardware purpose and its description is not required for switch utilization so it will not be described.

34.5.3.4.1 VLAN (SC-TAG)

Fetching the data from the Local RAM, updating the data depending on “L23 update” and “TAG function registers”. Updating VLANs data path is described in Figure 34.55.

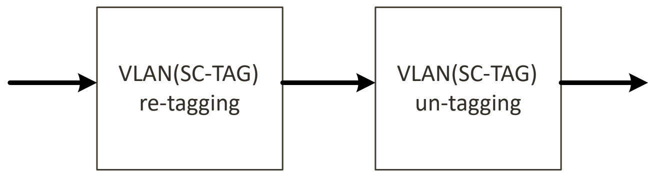


Figure 34.55 VLAN data path

(1) VLAN(SC-TAG) re-tagging

Layer 2 update can happen for the following fields:

- Destination MAC address updated to L23U.MDA when L23U.MDAU set.
- Source MAC address is updated to {GWMAC0.MAU, GWMAC1.MAL} when L23U.MSAU set.
- C-TAG VID is updated to L23U.CVID when L23U.CVIDU set.
- C-TAG PCP is updated to L23U.CPCP when L23U.CPCPU set.
- C-TAG VID is updated to L23U.CDEI when L23U.CDEIU set.
- S-TAG VID is updated to L23U.SVID when L23U.SVIDU set.
- S-TAG PCP is updated to L23U.SPCP when L23U.SPCPU set.
- S-TAG VID is updated to L23U.SDEI when L23U.SDEIU set.

Restrictions:

- HW: To update the C-TAG/S-TAG fields, the C-TAG/S-TAG should be present in the frame after VLAN untagging ((2) VLAN(SC-TAG) untagging). If not preset, the update will not happen (e.g. L2 update does not change the frame format and does not permit TAG insertion).

(2) VLAN(SC-TAG) untagging

VLAN update aims at modifying VLAN frame format depending on switch VLAN mode (FWGC.SVM register in MFWD), the VLAN control information given by the forwarding engine descriptor (FDESCR.VCTRL [FWD]) and GWVCC.VEM register.

Functions:

- GWVCC.VEM register sets the VLAN reception mode.

The frames format after VLAN untagging are described in Figure 34.56, Figure 34.57 and Figure 34.58 Switch VLAN mode (FWGC.SVM register in MFWD) and VLAN control information. The conversion between the frame stored in the Local RAM and the frame format after VLAN untagging is described in Figure 34.59.

	GWVCC.VEM				
VCTRL	3’b000	3’b001	3’b010	3’b011	3’b100
0	No TAG	No TAG	No TAG	No TAG	No TAG
1	No TAG	C−TAG	C−TAG	C−TAG	C−TAG
2	No TAG	C−TAG	C−TAG	SC−TAG	SC−TAG
3	No TAG	CoS−TAG	CoS−TAG	CoS−TAG	CoS−TAG
4	NA	NA	NA	NA	NA
5	NA	NA	NA	NA	NA
6	NA	NA	NA	NA	NA
7	NA	NA	NA	NA	NA

Figure 34.56 Frame format after VLAN untagging: Switch in No-VLAN mode [MFWD]

	GWVCC.VEM				
VCTRL	3’b000	3’b001	3’b010	3’b011	3’b100
0	No TAG	No TAG	C−TAG	No TAG	C−TAG
1	No TAG	C−TAG	C−TAG	C−TAG	C−TAG
2	No TAG	C−TAG	C−TAG	SC−TAG	SC−TAG
3	No TAG	CoS−TAG	C−TAG	CoS−TAG	C−TAG
4	No TAG	No TAG	C−TAG	No TAG	C−TAG
5	No TAG	No TAG	C−TAG	No TAG	C−TAG
6	No TAG	No TAG	C−TAG	No TAG	C−TAG
7	No TAG	CoS−TAG	C−TAG	CoS−TAG	C−TAG

Figure 34.57 Frame format after VLAN untagging: Switch in C-VLAN mode [MFWD]

	GWVCC.VEM				
VCTRL	3'b000	3'b001	3'b010	3'b011	3'b100
0	No TAG	No TAG	C-TAG	No TAG	SC-TAG
1	No TAG	C-TAG	C-TAG	C-TAG	SC-TAG
2	No TAG	C-TAG	C-TAG	SC-TAG	SC-TAG
3	No TAG	CoS-TAG	C-TAG	CoS-TAG	SC-TAG
4	No TAG	No TAG	C-TAG	No TAG	SC-TAG
5	No TAG	C-TAG	C-TAG	C-TAG	SC-TAG
6	No TAG	C-TAG	C-TAG	C-TAG	SC-TAG
7	No TAG	CoS-TAG	C-TAG	CoS-TAG	SC-TAG

Figure 34.58 Frame format after VLAN untagging: Switch in SC-VLAN mode [MFWD]

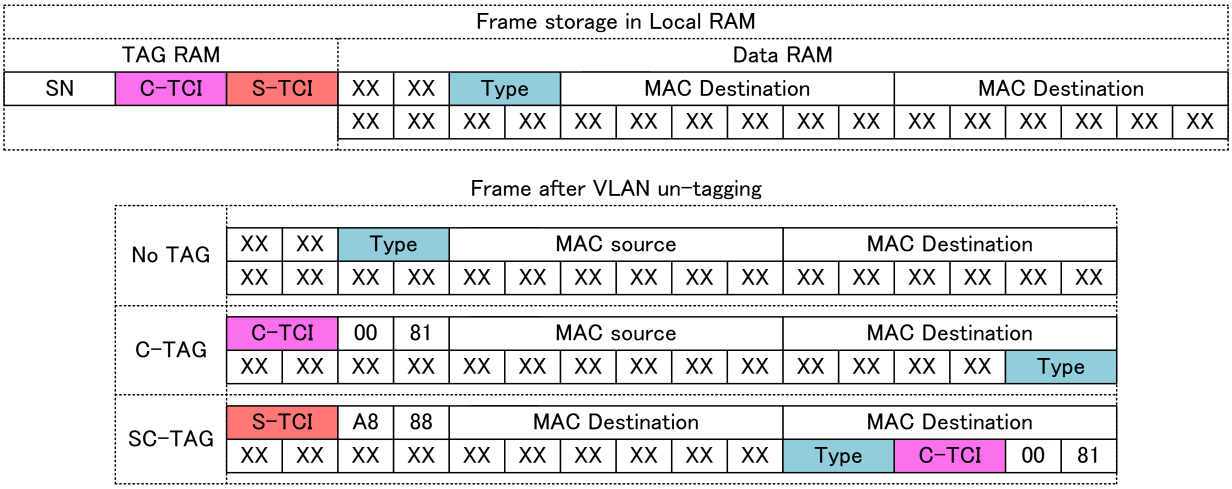


Figure 34.59 Local RAM frame to frame format after VLAN untagging conversion

Note:
- The Local RAM frame to frame format after VLAN untagging conversion for CoS-TAG is not described because it is the same as C-TAG except that the VID in the C-TCI will be fixed to 0.
- The R-TAG insertion is not described here because part of the L2/L3 update function, see section 34.5.3.4.2. R-TAG Insertion.
- H8100 corresponds to C-TAG TPID, C-TCI corresponds to a C-TAG TCI with a non-null VID, CoS-TCI corresponds to a C-TAG TCI with a null VID and CCs-TCI corresponds to C-TCI or CoS-TCI [802.1Q].
- H88A8 corresponds to S-TAG TPID and S-TCI corresponds to an S-TAG TCI [802.1Q].
- Type corresponds to any Ethernet type value which is not HF1C1, H88A8 or H8100.
- TPID values can be changed in Forwarding Engine [FWD]. Values for TPIDs can be set in forwarding engine (in FWTTC0 and FWTTC1 registers).

34.5.3.4.2 R-TAG Insertion

The R-TAG insertion happens for the two following cases:

- For frames received with an R-TAG (LDESCR.RTGI was set during reception, see section 34.5.2.3.3. Frame Saving in Local RAM, so, in descriptor received for forwarding, FDESCR.RTGI was also set) and for which routing is not requested (FDESCR.RV was not set in received descriptor [MFWD]) or for which routing is requested but the R-TAG stripping is not request by routing (L23U.RTU [MFWD] was not set to 10 in rule number FDESCR.RN [MFWD] during routing rule fetching by L2/L3 update block, see section 34.5.3.2. L2/L3 Update).

- For frames received without an R-TAG and for which routing and the R-TAG insertion is requested (L23U.RTU [MFWD] was not set to 01 in rule number FDESCR.RN [MFWD] during routing rule fetching by L2/L3 update block, see section 34.5.3.2. L2/L3 Update).

In both cases, the R-TAG sequence number (SN in Figure 34.60) will be set to the sequence number FDESCR.SEQN [MFWD] received in received descriptor.

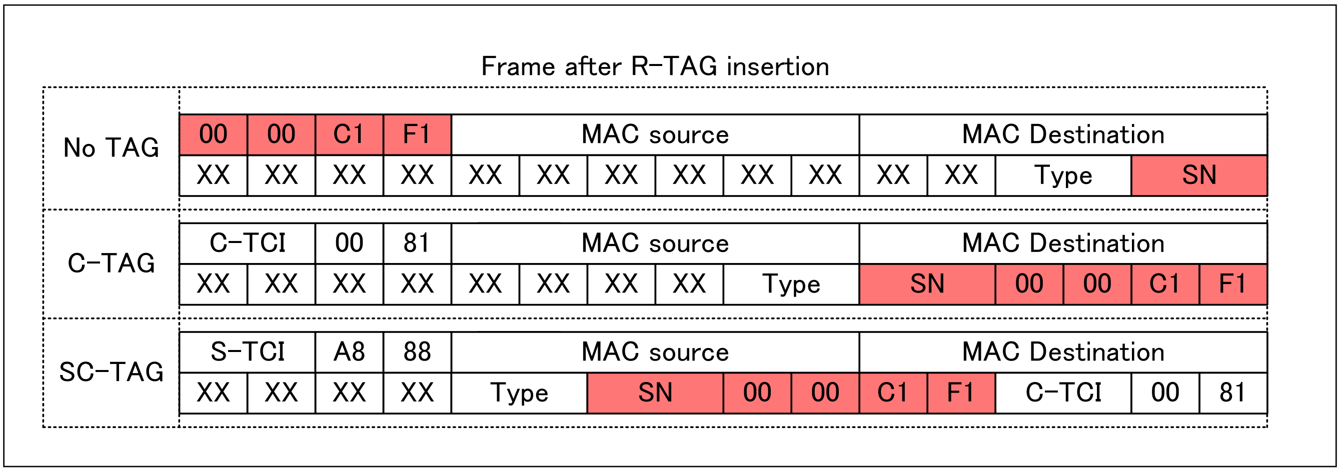


Figure 34.60 R-TAG inserted frame format

Note:
- HF1C1 corresponds to R-TAG TPID and SN correspond to R-TAG sequence number[802.1CB].
- H8100 corresponds to C-TAG TPID, C-TCI corresponds to a C-TAG TCI with a non-null VID, CoS-TCI corresponds to a C-TAG TCI with a null VID and CCs-TCI corresponds to C-TCI or CoS-TCI [802.1Q].
- H88A8 corresponds to S-TAG TPID and S-TCI corresponds to an S-TAG TCI [802.1Q].
- Type corresponds to any Ethernet type value which is not HF1C1, H88A8 or H8100.
- TPID values can be changed in Forwarding Engine [FWD]. Values for TPIDs can be set in forwarding engine (in FWTTC0 and FWTTC1 registers).

34.5.3.4.3 L2 /L3 Update

L2/L3 updates is used to the layer 2 and the layer 3 of frames. L2/L3 update happens for frames for which routing is requested (received descriptor with FDESCR.RV set [FWD]) and depending on the update information fetched by the L2/L3 update block (see section 34.5.3.2. L2/L3 Update).

(1) L2 update

See (1) VLAN(SC-TAG) re-tagging.

(2) L3 update

Layer 3 updates happens when L23U.TTLU was set. L3 update will happen in two steps, the frame decoding and the L3 update.

The decoder decodes the type field (see section 34.5.2.3.1. TAG Information Extraction) of the Layer 3 update frame and separate frames in the following formats:

- IPv4: When type field is equal to H0800.
- IPv6: When type field is equal to H86DD.
- Unknown: When type field is other than H0800 and H86DD.

Depending on the decoding result, the L3 update will happen has following:

- For IPv4: TimeToLive will be decremented by 1 if it is other than 0 and HeaderChecksum will be corrected using a relative equation (If the checksum were wrong in the input frame, the checksum will also be wrong for the output L3 updated frame).

- For IPv6: HopLimit will be decremented by 1 if it is other than 0.
- For Unknown: L3 update will be ignored.

34.5.3.4.4 FCS Handling

Ethernet frame FCS can be forwarded to the CPU using GWRGC.RCPT register.

Functions:

- GWRGC.RCPT is used to pass frame FCSs to the CPU.

However, following conditions should be met for GWCA to forward a frame FCS to the CPU:

- For Ethernet agent ETHA, RMAC should not have removed the FCS [RMAC] at the input port.
- For GWCA, the FCS should be present in the ingress frame and valid (FI set and FW not set in input descriptor, see section 34.5.2.2.3. Data Transmission Common Descriptor Format).
- The VLANs should be the same at the egress than at the ingress port.
- The frame should not be a routed frame (received descriptor FDESCR.RV field not set MFWD).

Note:
- If one of the above conditions is not met by a frame, the FCS will be removed from it even if GWRGC.RCPT register is set to 1.
- If a frame is received by CPU with an FCS, FI flag will be set in the reception descriptor. If not, it will not be set.

34.5.3.5 AXI Master Interface

The AXI master interface handles the data/AXI descriptor exchange with the CPU. It receives the updated data and the information related to the data for RX data fetch block and stores them in the CPU RAM. To do so, the AXI master interface uses GWDCCi.EDE, GWDCCi.ETS, GWDCCi.SM, and can be monitored using GWAARSS, GWAARSR0 and GWAARSR1, GWIDAUASi, GWIDASMi, GWIDASAMi0, GWIDASAMi1, GWIDACAMi0, GWIDACAMi1 registers and GWEIS2i.DFESt, GWEIS3.IAOESi, GWEIS4.DSES, GWEIS4.DSEIOS, GWEIS4.DSECN interrupt registers.

Functions:

- GWDCCi.EDE and GWDCCi.ETS registers set the descriptor format that are used for descriptor queue number i. See section 34.5.1.2.1. General Formats.
- GWDCCi.SM register sets how GWCA writes back descriptors. In this section, all explanation will use GWDCCi.SM equal to 00b. For GWDCCi.SM equal to 01b or 10b, see the register explanation.
- For GWAARSS, GWAARSR0 and GWAARSR1 registers, see section 34.5.1.6. AXI Address RAM Searching.
- GWIDAUASi, GWIDASMi, GWIDASAMi0, GWIDASAMi1, GWIDACAMi0, GWIDACAMi1 registers and GWEIS3.IAOESi interrupt register are used for incremental area monitoring, see section 34.5.3.5.3. Single Page Incremental Data Reception.
- GWEIS2i.DFESt interrupt register signals that frames have been lost because a full descriptor queue.
- GWEIS4.DSES, GWEIS4.DSEIOS, GWEIS4.DSECN interrupt registers signal that frames have been received with an unexpected size, see section 34.5.3.5.2. Size-controlled Data Reception.

Data reception can be done using linear or cyclic descriptor queues (see section 34.5.1.5. AXI Descriptor Queue Format). Data reception process descriptors can be set six different ways, through the basic data reception descriptor chain (section 34.5.3.5.1. Basic Data Reception), the Size-controlled data reception descriptor chain (section 34.5.3.5.2. Size-controlled Data Reception), the Single page incremental data reception descriptor chain (section 34.5.3.5.3. Single Page Incremental Data Reception), the Interrupt based multi-page incremental data reception descriptor chain (section 34.5.3.5.4. Interrupt Based on Multi-page Incremental Data Reception), the Header removal incremental data reception descriptor chain (section 34.5.3.5.5. Header Removal Incremental Data Reception).

34.5.3.5.1 Basic Data Reception

Basic data reception process descriptor utilization is described in Figure 34.61. Basic data reception process descriptor format is described in Figure 34.62 (for basic descriptor, section section 34.5.1.2.1. General Formats) and Table 34.23.

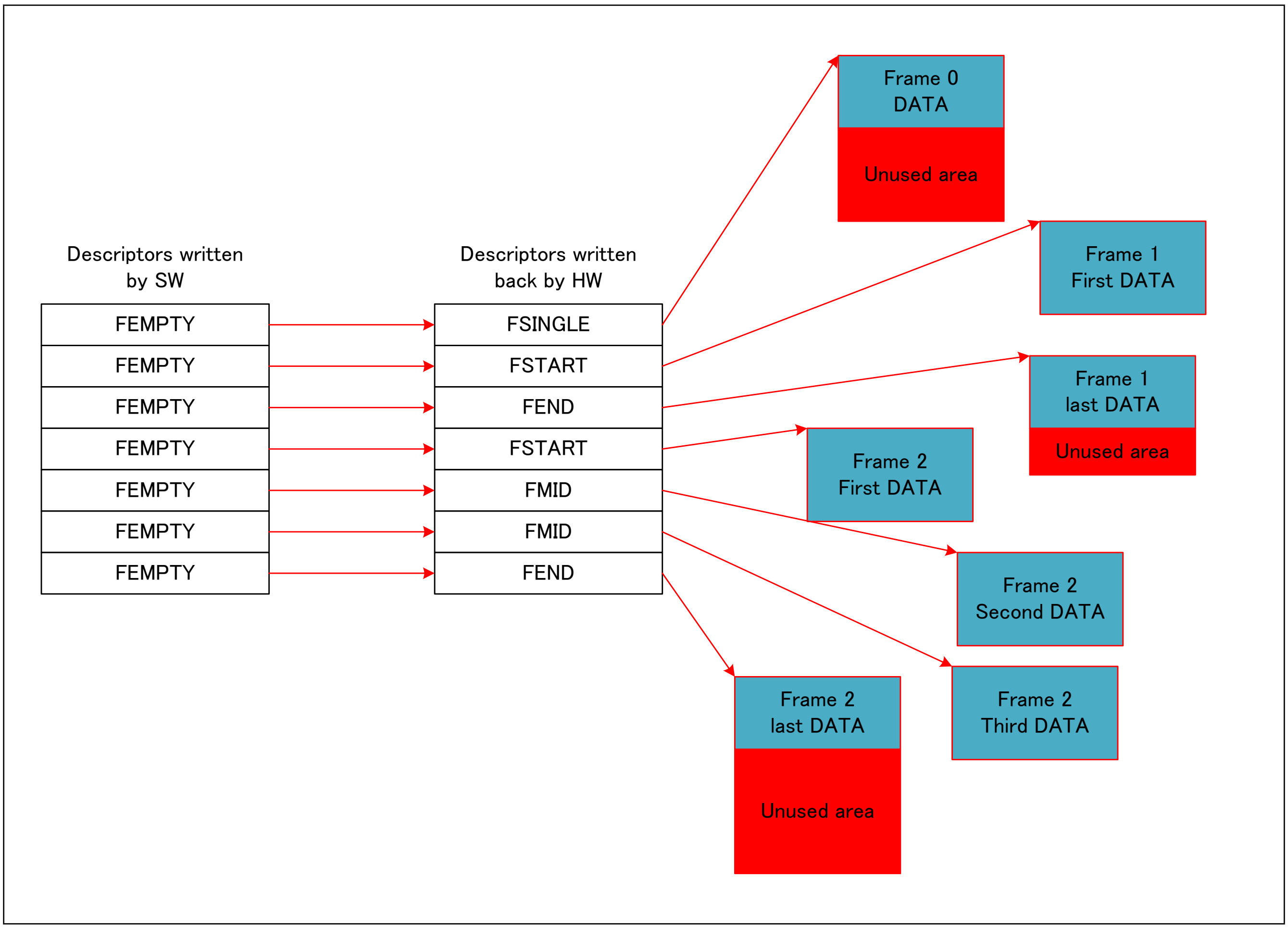


Figure 34.61 Basic data reception process descriptor utilization

Note:
- If a frame is too big to fit in an area reserved by one descriptor, it will be automatically split. The first descriptor will be FSTART, the last descriptor will be FEND and all other descriptors will be FMID.
- When a frame is split the hardware will write back the FSTART descriptor last. Even if the queue is set in keep DT mode, this order is kept.

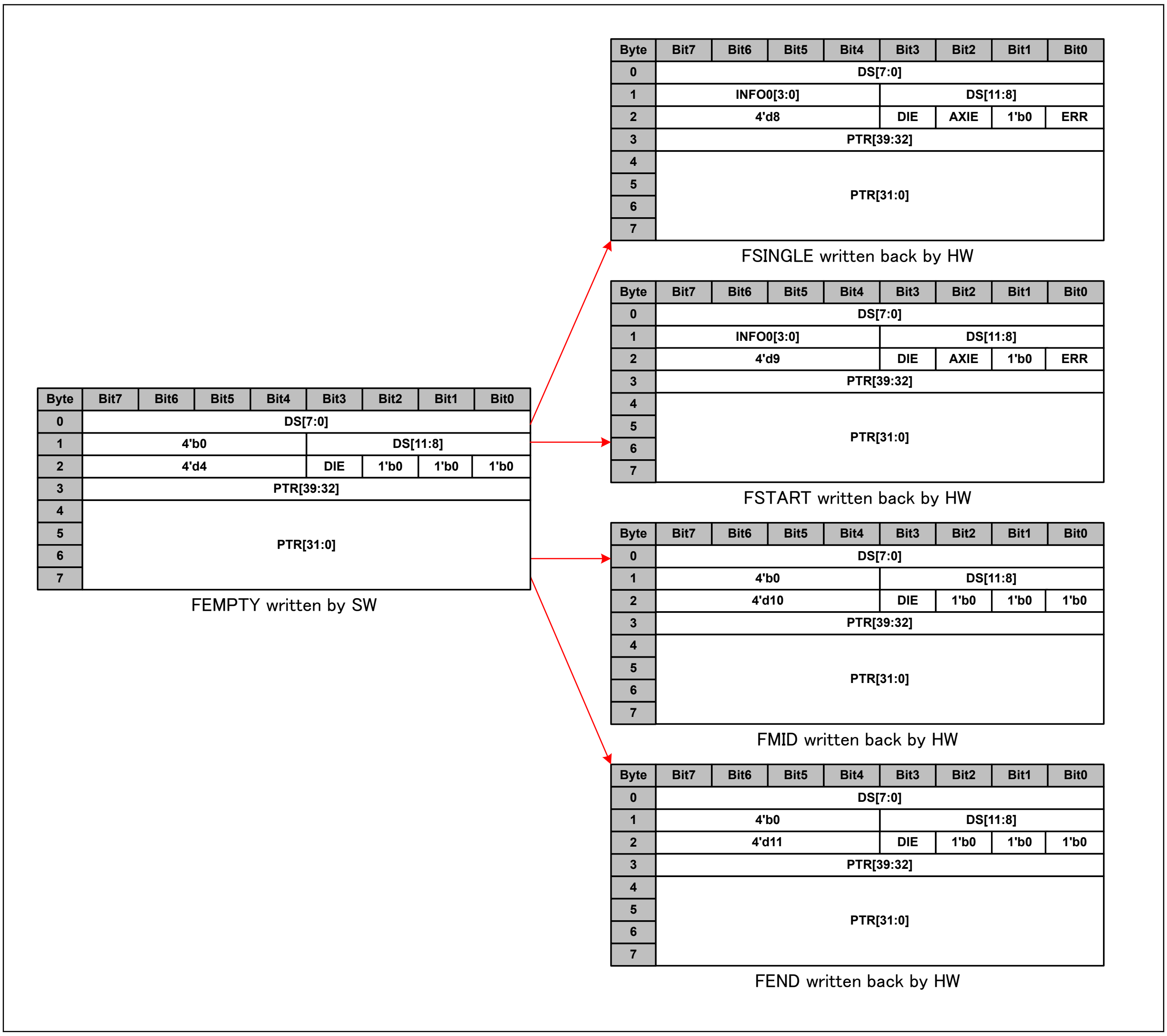


Figure 34.62 Basic data reception process descriptor utilization format

Table 34.23 Basic data reception process descriptor utilization field description (1 of 2)

Field name	Field description before write-back	Field description after write-back
DS	Size of available URAM memory associated to the pointer for data storage	Actual size that has been written by the AXIBMI in the URAM.
INFO0	0	0 for FEND and FMID. For FSTART and FSINGLE, see section 34.5.3.5.6. Data Reception Common Descriptor Format.
ERR	0	Set conditions: ● HW: A descriptor queue full error happened while reading an AXI address during a split frame (The first descriptor has been processed normally but the last part of the frame has no descriptor to be written) and a FSINGLE or a FSTART descriptor written back. ● HW: A descriptor number error happened while reading an AXI address during a split frame (The first descriptor has been processed normally but the last part of the frame will not be written) and a FSINGLE or a FSTART descriptor written back.
DSE	0	0

Table 34.23 Basic data reception process descriptor utilization field description (2 of 2)

Field name	Field description before write-back	Field description after write-back
AXIE	0	Set conditions: ● HW: An AXI error happened while writing a data (BRESP != 00) and a FSINGLE or a FSTART descriptor written back. ● HW: An AXI error happened while reading a descriptor during a split frame (The first descriptor has been processed normally but the last part of the frame has no descriptor to be written) and a FSINGLE or a FSTART descriptor written back.
DIE	DIE	DIE (Value is copied from read descriptor).
DT	4 for FEMPTY	8 for FSINGLE 9 for FSTART 10 for FMID 11 for FEND
PTR	Address where data should be written in the URAM	PTR (Value is copied from read descriptor).
INFO1 (GWDCCi.EDEi == 1)	0	0 for FEND and FMID, For FSTART and FSINGLE, see section 34.5.3.5.6. Data Reception Common Descriptor Format.
TS (GWDCCi.ETSi == 1)	0	0 for FEND and FMID, For FSTART and FSINGLE, see section 34.5.3.5.6. Data Reception Common Descriptor Format.

34.5.3.5.2 Size-controlled Data Reception

Size-controlled data reception aims at controlling the frame size at reception while purposely splitting the header and the payload of a frame. By size-controlled data reception the SW ensure that descriptor order is maintained even if a too short or too long frame is received. Size-controlled data reception process descriptor utilization is described in Figure 34.63. Size-controlled data reception process descriptor format is described Figure 34.64 (for basic descriptor, section 34.5.1.2.1. General Formats) in and Table 34.24. Size-controlled data reception can be monitored using GWEIS4.DSES, GWEIS4.DSEIOS, GWEIS4.DSECN interrupt registers.

Function:

- GWEIS4.DSES, GWEIS4.DSEIOS, and GWEIS4.DSECN interrupt registers signal a frame received with an unexpected size.

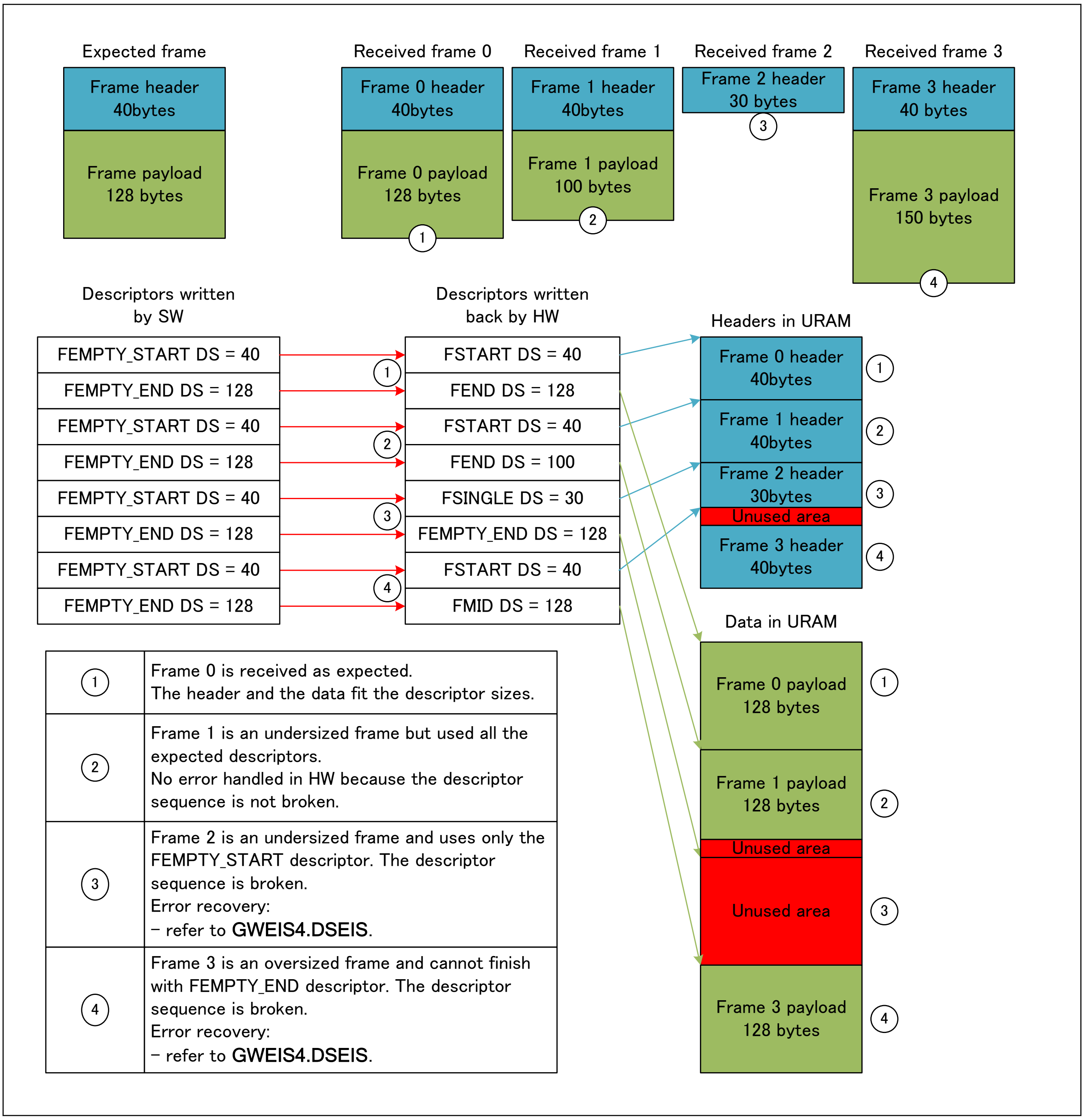


Figure 34.63 Size-controlled data reception process descriptor utilization

Note:
- When a frame is split the hardware will write back the FSTART descriptor last. Even if the queue is set in keep DT mode, this order is kept.
- A frame can be split in more than two descriptors by adding FEMPTY_MID descriptors between FEMPTY_START and FEMPTY_END descriptors.

Restrictions:

- SW: The process descriptor queue should always repeat the pattern of the same descriptors from the first descriptor to the descriptor before the termination descriptor. This pattern should start with an FEMPTY_START descriptor, finish by a FEMPTY_END descriptor and in between be only composed by FEMPTY_MID descriptors. If this is violated (for example, FEMPTY_START is detected in the middle), the frame will be discarded/rejected without notification.

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	DS[7:0]							
1	4'b0				DS[11:8]			
2	4'd4				DIE	1'b0	1'b0	1'b0
3	PTR[39:32]							
4	PTR[31:0]							
5								
6								
7								

FEMPTY_START written by SW

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	DS[7:0]							
1	INFO0[3:0]				DS[11:8]			
2	4'd9				DIE	AXIE	DSE	ERR
3	PTR[39:32]							
4	PTR[31:0]							
5								
6								
7								

FSTART written back by HW

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	DS[7:0]							
1	4'b0				DS[11:8]			
2	4'd5				DIE	1'b0	1'b0	1'b0
3	PTR[39:32]							
4	PTR[31:0]							
5								
6								
7								

FEMPTY_MID written by SW

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	DS[7:0]							
1	4'b0				DS[11:8]			
2	4'd10				DIE	1'b0	1'b0	1'b0
3	PTR[39:32]							
4	PTR[31:0]							
5								
6								
7								

FMID written back by HW

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	DS[7:0]							
1	4'b0				DS[11:8]			
2	4'd6				DIE	1'b0	1'b0	1'b0
3	PTR[39:32]							
4	PTR[31:0]							
5								
6								
7								

FEMPTY_END written by SW

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	DS[7:0]							
1	4'b0				DS[11:8]			
2	4'd11				DIE	1'b0	1'b0	1'b0
3	PTR[39:32]							
4	PTR[31:0]							
5								
6								
7								

FEND written back by HW

Figure 34.64 Size-controlled data reception process descriptor utilization format

Table 34.24 Size-controlled data reception process descriptor utilization field description (1 of 2)

Field name	Field description before write-back	Field description after write-back
DS	Size of available URAM memory associated to the pointer for data storage	Actual size that has been written by the AXIBMI in the URAM.
INFO0	0	0 for FEND and FMID. For FSTART and FSINGLE, see section 34.5.3.5.6. Data Reception Common Descriptor Format.
ERR	0	Set condition: ● HW: A descriptor queue full error happened while reading an AXI address during a split frame (The first descriptor has been processed normally but the last part of the frame has no descriptor to be written) and a FSINGLE or a FSTART descriptor written back. (This case should not happen due to the process descriptor order) ● HW: A descriptor number error happened while reading an AXI address during a split frame (The first descriptor has been processed normally but the last part of the frame will not be written) and a FSINGLE or a FSTART descriptor written back.
DSE	0	Set conditions: ● HW: The frame has been finished with a FEMPTY_START or a FEMPTY_MID descriptor (undersized frame) ● HW: The frame is too big to finish with a FEMPTY_END descriptor (oversized frame)
AXIE	0	Set conditions: ● HW: An AXI error happened while writing a data (BRESP != 00) and a FSINGLE or a FSTART descriptor written back. ● HW: An AXI error happened while reading a descriptor during a split frame (The first descriptor has been processed normally but the last part of the frame has no descriptor to be written) and a FSINGLE or a FSTART descriptor written back.
DIE	DIE	DIE (Value is copied from read descriptor).

Table 34.24 Size-controlled data reception process descriptor utilization field description (2 of 2)

Field name	Field description before write-back	Field description after write-back
DT	5 for FEMPTY_START 6 for FEMPTY_MID 7 for FEMPTY_END	9 for FSTART 10 for FMID 11 for FEND
PTR	Address where data should be written in the URAM	PTR (Value is copied from read descriptor).
INFO1 (GWDCCi.EDEi == 1)	0	0 for FEND and FMID. For FSTART and FSINGLE, see section 34.5.3.5.6. Data Reception Common Descriptor Format.
TS (GWDCCi.ETSi == 1)	0	0 for FEND and FMID, For FSTART and FSINGLE, see section 34.5.3.5.6. Data Reception Common Descriptor Format.

34.5.3.5.3 Single Page Incremental Data Reception

Single page incremental data reception aims at saving the received frames back to back in a given memory area. By single page incremental data reception, the SW avoid wasting CPU power to stack data together. Single page incremental data reception process descriptor utilization is described in Figure 34.65 (for basic descriptor, section 34.5.1.2.1. General Formats). Single page incremental data reception process descriptor format is described Figure 34.66, Table 34.25, Figure 34.67 and Table 34.26. Single page incremental data reception HW/SW synchronization should be done by GWIDAUASi register and can be monitored by GWIDASMi, GWIDASAMi0, GWIDASAMi1, GWIDACAMi0, GWIDACAMi1 registers and GWEIS3.IAOESi interrupt register.

Functions:

- GWIDAUASi register shows the number of bytes that has been written in the incremental area. After reading the data from the CPU RAM, SW should write to this register the quantity of data that has been read. The CPU data handling can be done, either based on the written back descriptor in the corresponding descriptor queue, or on the byte value contained in the increment area than can be read in GWIDAUASi register.
- GWIDASMi register is used to monitor the total size of the incremental area.
- {GWIDASAMi0.IDASAU, GWIDASAMi1.IDASAL} register array is used to monitor the start address of the incremental area.
- {GWIDACAMi0.IDACAU, GWIDACAMi1.IDACAL} register array is used to monitor the next used address in the incremental area.
- GWEIS3.IAOESi interrupt register signals data loss due to incremental area overflow.

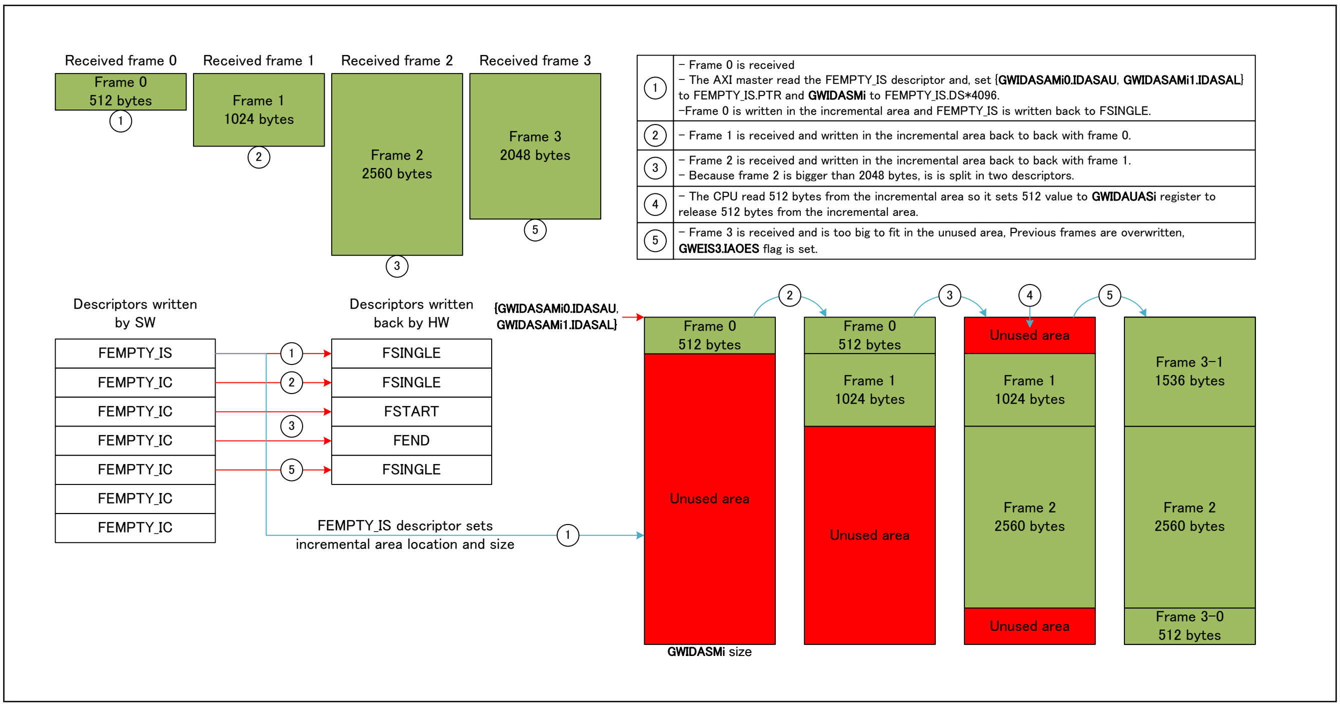


Figure 34.65 Single page incremental data reception process descriptor utilization

Restrictions:

- SW: This reception type can only be used for the first 4 RX chains, see section 34.5.1.1. AXI Descriptor Queue Mapping.

Note:
- If a frame is too big to fit in an area reserved by one descriptor, it will be automatically split. The first descriptor will be FSTART, the last descriptor will be FEND and all other descriptors will be FMID.
- When a frame is split the hardware will write back the FSTART descriptor last. Even if the queue is set in keep DT mode, this order is kept.
- When GWCA reset has been applied (GWMS.OPS set to 00), all the incremental registers (i.e. incremental page start address, current address, page size and used page size) were cleared. Therefore, when the GWCA has moved back again to OPERATION mode (GWMS.OPS set to 11) and a frame has been received by GWCA incremental chain, it is not able to write frame using FEMPTY_IC descriptor. Descriptor of AXI current address have to be (set up) a FEMPTY_IS.

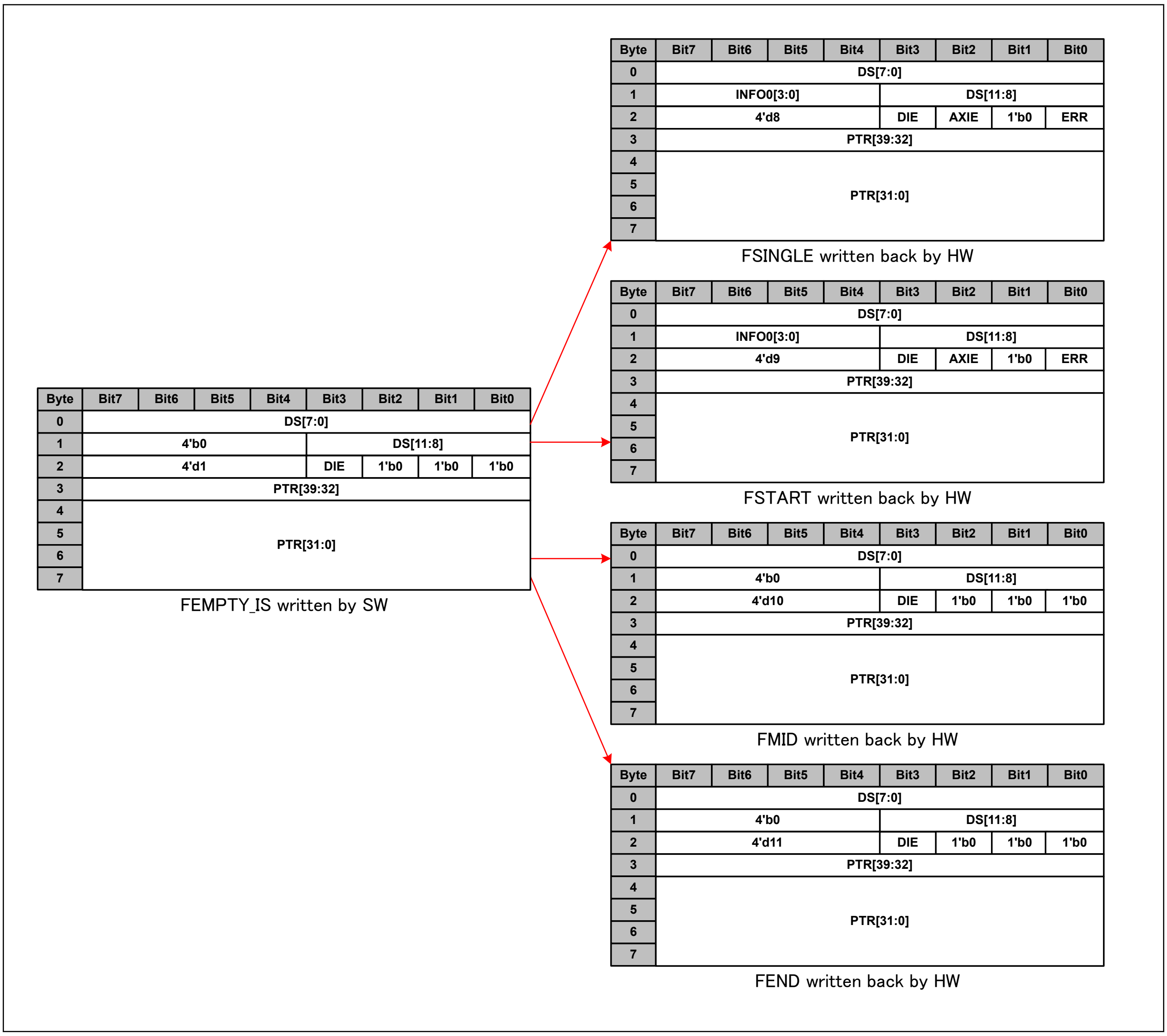


Figure 34.66 Single page incremental data reception FEMPTY_IS process descriptor utilization format

Table 34.25 Single page incremental data reception FEMPTY_IS process descriptor utilization field description (1 of 2)

Field name	Field description before write-back	Field description after write-back
DS	Size of the incremental area in 4 KB unity.	Size of the data stored in the URAM associated tot the descriptor.
INFO0	0	0 for FEND and FMID. For FSTART and FSINGLE, see section 34.5.3.5.6. Data Reception Common Descriptor Format.
ERR	0	Set conditions: ● HW: A descriptor queue full error happened while reading an AXI address during a split frame (The first descriptor has been processed normally but the last part of the frame has no descriptor to be written) and a FSINGLE or a FSTART descriptor written back. ● HW: A descriptor number error happened while reading an AXI address during a split frame (The first descriptor has been processed normally but the last part of the frame will not be written) and a FSINGLE or a FSTART descriptor written back.
DSE	0	0

Table 34.25 Single page incremental data reception FEMPTY_IS process descriptor utilization field description (2 of 2)

Field name	Field description before write-back	Field description after write-back
AXIE	0	Set conditions: ● HW: An AXI error happened while writing a data (BRESP != 00) and a FSINGLE or a FSTART descriptor written back. ● HW: An AXI error happened while reading a descriptor during a split frame (The first descriptor has been processed normally but the last part of the frame has no descriptor to be written) and a FSINGLE or a FSTART descriptor written back.
DIE	DIE	DIE (Value is copied from read descriptor).
DT	1 for FEMPTY_IS	8 for FSINGLE 9 for FSTART 10 for FMID 11 for FEND
PTR	Address where the incremental area starts in the URAM.	PTR (Value is copied from read descriptor).
INFO1 (GWDCCi.EDE == 1)	0	0 for FEND and FMID. For FSTART and FSINGLE, see section 34.5.3.5.6. Data Reception Common Descriptor Format.
TS (GWDCCi.ETS == 1)	0	0 for FEND and FMID, For FSTART and FSINGLE, see section 34.5.3.5.6. Data Reception Common Descriptor Format.

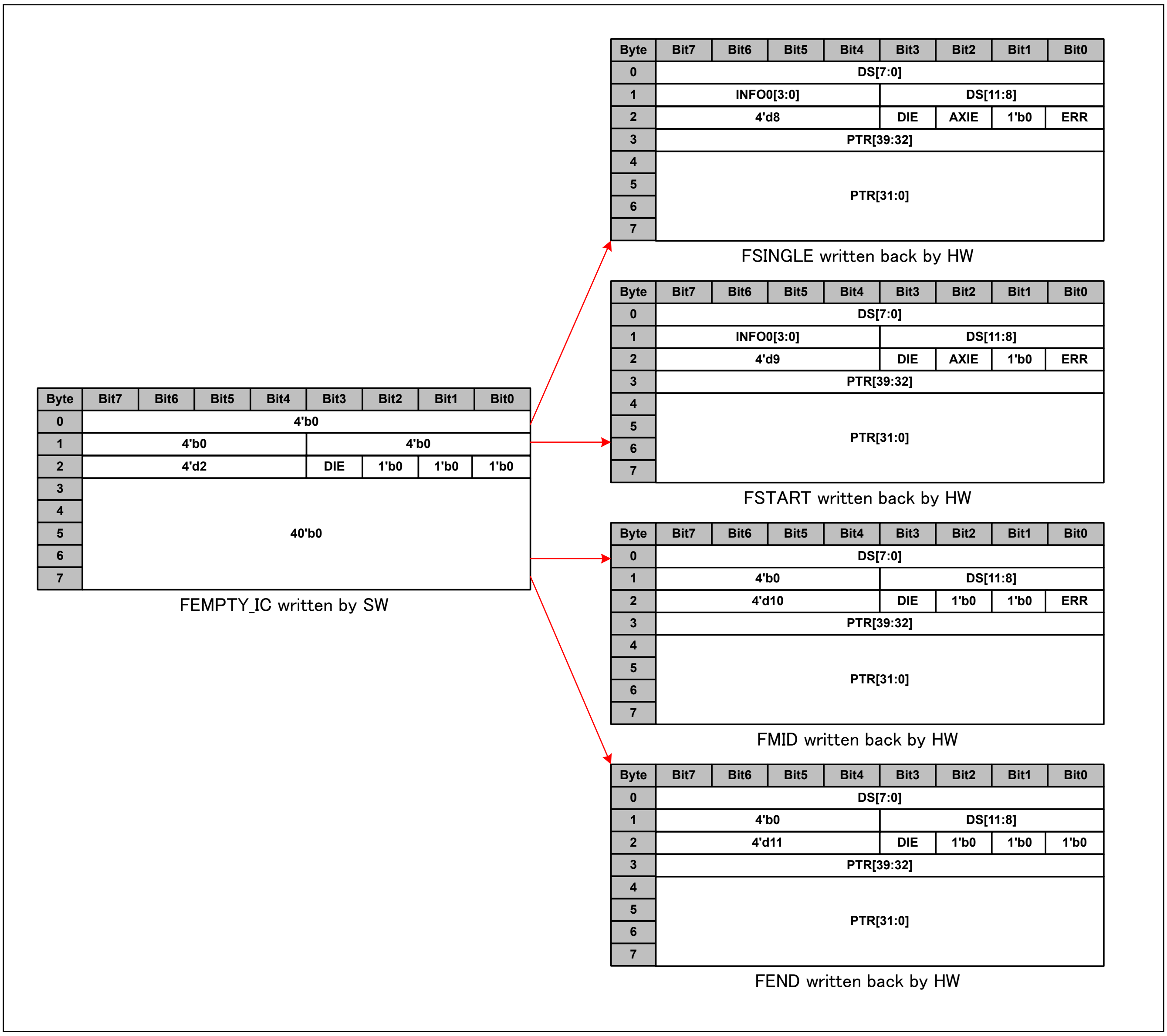


Figure 34.67 Single page incremental data reception FEMPTY_IC process descriptor utilization format

Table 34.26 Single page incremental data reception FEMPTY_IC process descriptor utilization field description (1 of 2)

Field name	Field description before write-back	Field description after write-back
DS	0	Size of the data stored in the URAM associated tot the descriptor.
INFO0	0	0 for FEND and FMID. For FSTART and FSINGLE, see section 34.5.3.5.6. Data Reception Common Descriptor Format.
DSE	0	0
AXIE	0	Set conditions: ● HW: An AXI error happened while writing a data (BRESP != 00) and a FSINGLE or a FSTART descriptor written back. ● HW: An AXI error happened while reading a descriptor for a split frame and the descriptor was not the first.
DIE	DIE	DIE (Value is copied from read descriptor).

Table 34.26 Single page incremental data reception FEMPTY_IC process descriptor utilization field description (2 of 2)

Field name	Field description before write-back	Field description after write-back
DT	2 for FEMPTY_IC	8 for FSINGLE 9 for FSTART 10 for FMID 11 for FEND
PTR	0	Address where the data corresponding to the descriptor has been written in the URAM
INFO1 (GWDCCi.EDE == 1)	0	0 for FEND and FMID. For FSTART and FSINGLE, see section 34.5.3.5.6. Data Reception Common Descriptor Format.
TS (GWDCCi.ETS == 1)	0	0 for FEND and FMID. For FSTART and FSINGLE, see section 34.5.3.5.6. Data Reception Common Descriptor Format.

34.5.3.5.4 Interrupt Based on Multi-page Incremental Data Reception

An interrupt based on multi-page incremental data reception aims at saving the received frames back to back in a given memory area. By an interrupt based on multi-page incremental data reception, the SW avoids wasting CPU power to stack data together. An interrupt based on multi-page incremental data reception process descriptor utilization is described in Figure 34.68. An interrupt based on multi-page incremental data reception process descriptor format is the same as single page incremental data reception. For details, see section 34.5.3.5.3. Single Page Incremental Data Reception.

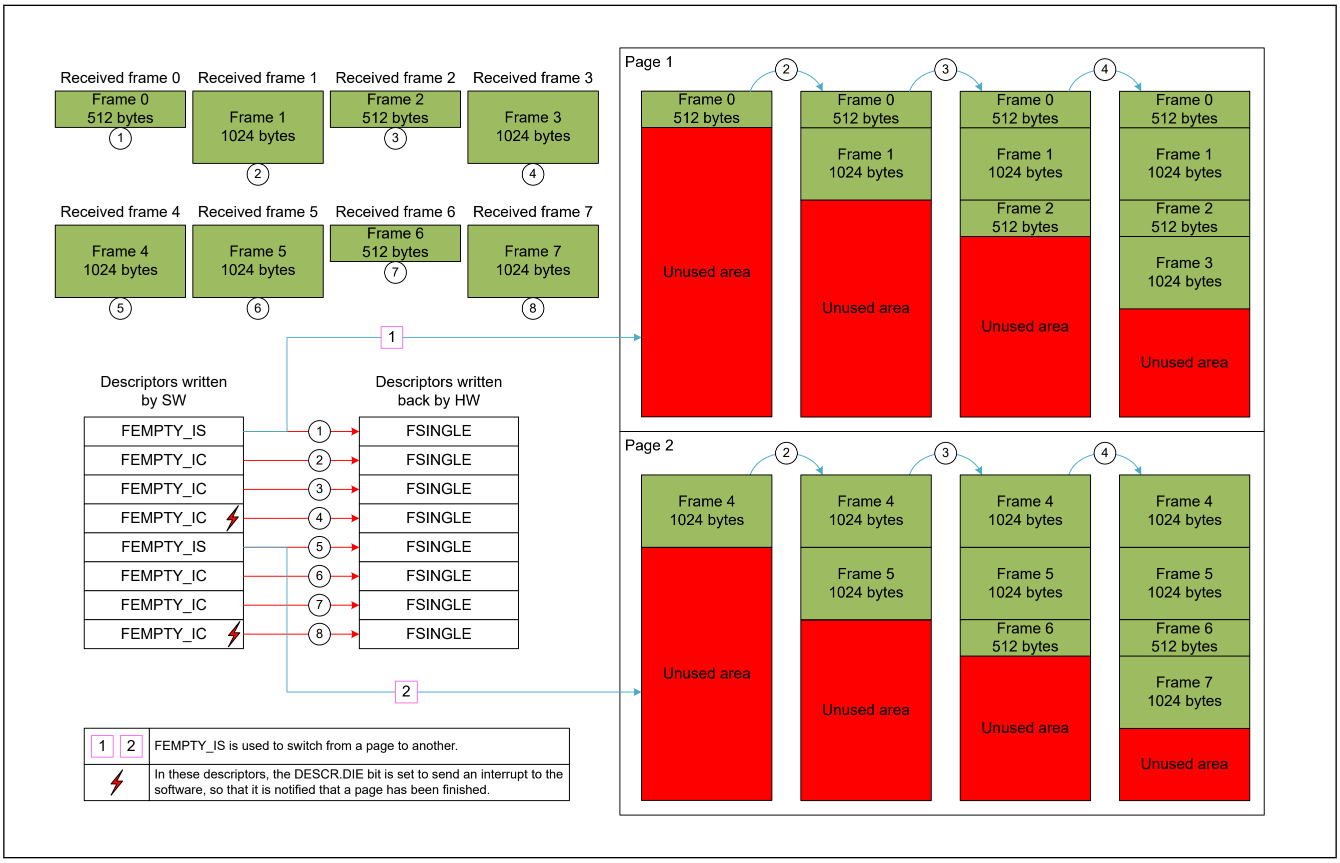


Figure 34.68 Interrupt based on multi-page incremental data reception process descriptor utilization

Restrictions:

- SW: It should be ensured that, recognizing the expected data and/or controlling page sizes and the number of descriptors for a page, a page cannot overflow.
- SW: This reception type can only be used for the first 4 RX chains.

Note:
- If a frame is too big to fit in an area reserved by one descriptor, it will be automatically split. The first descriptor will be FSTART, the last descriptor will be FEND, and all other descriptors will be FMID.
- When a frame is split, the hardware will write back the FSTART descriptor last. Even if the queue is set in keeping DT mode, this order is kept.
- An interrupt based on multi-page incremental data reception allows any number of pages to be used.
- When GWCA reset has been applied (GWMS.OPS is set to 00), all the incremental registers (i.e. incremental page start address, current address, page size, and used page size) were cleared. Therefore, when the GWCA has moved back again to OPERATION mode (GWMS.OPS is set to 11) and a frame has been received by GWCA incremental chain, it is not able to write a frame using FEMPTY_IC descriptor. A descriptor of AXI current address has to be (set up) a FEMPTY_IS.

34.5.3.5.5 Header Removal Incremental Data Reception

Header removal incremental data reception aims at suppressing header from incremental data. It is an update of single page incremental data reception (section 34.5.3.5.3. Single Page Incremental Data Reception) and interrupt based multi-page incremental data reception (section 34.5.3.5.4. Interrupt Based on Multi-page Incremental Data Reception) by adding a FEMPTY_ND between each incremental descriptor. Header removal incremental data reception process descriptor utilization is described in Figure 34.69. Header removal process descriptor format is described in Figure 34.70 (for basic descriptor, section 34.5.1.2.1. General Formats) and Table 34.27.

Restrictions:

- SW: A FEMPTY_IS/FEMPTY_IC descriptor can only contain 2048 bytes. It should be taken in account when setting the descriptor in the descriptor queue because the HW does not ensure that the descriptor sequence is correct.

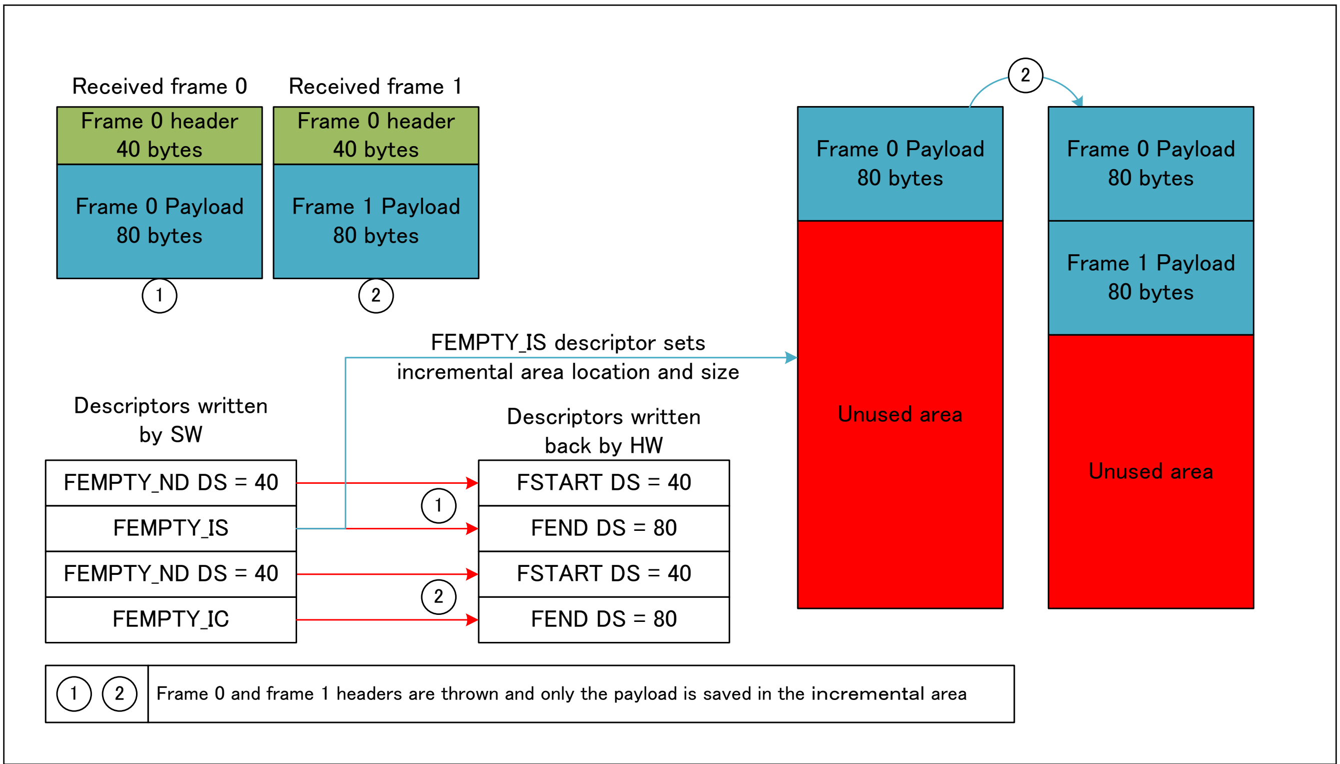


Figure 34.69 Header removal incremental data reception process descriptor utilization

Note:
- When using Size controlled data reception, to get rid of the header part of a frame and still keep the descriptor order, it is possible to start every frame with a FEMPTY_START descriptor always pointing toward the same memory address (by setting always the same pointer).
- When GWCA reset has been applied (GWMS.OPS set to 00), all the incremental registers (i.e incremental page start address, current address, page size and used page size) were cleared. Therefore, when the GWCA has moved back again to OPERATION mode (GWMS.OPS set to 11) and a frame has been received by GWCA incremental chain, it is not able to write frame using FEMPTY_IC descriptor. Descriptor of AXI current address have to be (set up) a FEMPTY_IS.

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	DS[7:0]							
1	4'b0				DS[11:8]			
2	4'd3				DIE	1'b0	1'b0	1'b0
3	40'b0							
4								
5								
6								
7								

FEMPTY_ND written by SW

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	DS[7:0]							
1	INFO0[3:0]				DS[11:8]			
2	4'd8				DIE	AXIE	1'b0	ERR
3	40'b0							
4								
5								
6								
7								

FSINGLE written back by HW

Figure 34.70 Header removal utilization format

Table 34.27 Header removal process descriptor utilization field description

Field name	Field description before write-back	Field description after write-back
DS	Size to be rejected	Actual size that has been rejected (it should be the same as DS before write-back).
INFO0	0	0 for FEND and FMID. For FSTART and FSINGLE, see section 34.5.3.5.6. Data Reception Common Descriptor Format.
ERR	0	Set conditions: ● HW: A descriptor queue full error happened while reading an AXI address during a split frame (The first descriptor has been processed normally but the last part of the frame has no descriptor to be written) and a FSINGLE or a FSTART descriptor written back. ● HW: A descriptor number error happened while reading an AXI address during a split frame (The first descriptor has been processed normally but the last part of the frame will not be written) and a FSINGLE or a FSTART descriptor written back.
DSE	0	0
AXIE	0	Set conditions: ● HW: An AXI error happened while writing a data (BRESP != 00) and a FSINGLE or a FSTART descriptor written back. ● HW: An AXI error happened while reading a descriptor during a split frame (The first descriptor has been processed normally but the last part of the frame has no descriptor to be written) and a FSINGLE or a FSTART descriptor written back.
DIE	DIE	DIE (Value is copied from read descriptor).
DT	3 for FEMPTY_ND	8 for FSINGLE 9 for FSTART 10 for FMID 11 for FEND Restrictions: ● SW: Because the HW turns a FEMPTY_ND descriptor in a FSINGLE, FSTART, FMID or FEND descriptor, by reading the descriptor it is impossible to know that is contains no data. Consequently, SW should remember where the FEMPTY_ND descriptors have been written.
PTR	0	0
INFO1 (GWDCCi.EDE == 1)	0	0 for FEND and FMID, For FSTART and FSINGLE, see section 34.5.3.5.6. Data Reception Common Descriptor Format.
TS (GWDCCi.ETS == 1)	0	0 for FEND and FMID, For FSTART and FSINGLE, see section 34.5.3.5.6. Data Reception Common Descriptor Format.

34.5.3.5.6 Data Reception Common Descriptor Format

In this section are described the fields which are common for any data transmission (INFO0, INFO1 and TS). There are two types of formats, the reception direct descriptor and the reception ethernet descriptor formats. They are respectively created from direct local descriptors and ethernet local descriptors, see section 34.5.2.3.3. Frame Saving in Local RAM. All the fields contained in AXI descriptors, if quoted, will be written ADESCR.{Field name}.

(1) Reception direct descriptor

Reception direct descriptor corresponds to frames sent by a CPU and by deciding their target directly (Forwarding engine processing has been by-passed [FWD]). Reception direct descriptor INFO and TS formats are described in Figure 34.71 and Table 34.28.

INFO0

Byte	Bit3	Bit2	Bit1	Bit0
0	—		SEC	FI

INFO1

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	—	TN	CRT	IET	TXC	FMT	SEC	FI
1	TSUN[7:0]							
2	SAEF[7:0]							
3	RN[7:0]							
4	—	SPN[2:0]			RV	—		
5	FESF[23:0]							
6								
7								

TS

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	—	CSD0[6:0]						
1	—	CSD1[6:0]						
2	—	DV[6:0]						
3	—							
4								
5								
6								
7								

Figure 34.71 Reception direct descriptor INFO and TS formats

Table 34.28 Reception direct descriptor INFO and TS field descriptions (1 of 2)

Field name	Bit width	Description	Value for GWCA
FI	1	See Table 34.14.	See section 8.3.4.5.
SEC	1	See Table 34.14.	Copied from direct local descriptor. See (1) Direct local descriptor.
FMT	1	See Table 34.14.	Copied from direct local descriptor. See (1) Direct local descriptor.
TXC	1	See Table 34.14.	Copied from direct local descriptor. See (1) Direct local descriptor.
IET	1	See Table 34.14.	Copied from direct local descriptor. See (1) Direct local descriptor.
CRT	1	See Table 34.14.	Copied from direct local descriptor. See (1) Direct local descriptor.
TN	1	See Table 34.14.	Copied from direct local descriptor. See (1) Direct local descriptor.
TSUN	8	See Table 34.14.	Copied from direct local descriptor. See (1) Direct local descriptor.
SAEF	8	See Table 34.16.	Copied from direct local descriptor. See (1) Direct local descriptor.
RV	1	See Table 34.14.	Copied from received descriptor FDESCR.RV field [MFWD]

Table 34.28 Reception direct descriptor INFO and TS field descriptions (2 of 2)

Field name	Bit width	Description	Value for GWCA
RN	5	See Table 34.14.	Copied from received descriptor FDESCR.RN field [MFWD]
SPN	2	Source port number	Port number from which the frame entered the switch [MFAB] Copied from received descriptor FDESCR.SPN field [MFWD]
FESF	24	Forwarding engine status flags	Copied from received descriptor FDESCR.MINFO field [MFWD]
CSD0	6	See Table 34.14.	Copied from direct local descriptor. See (1) Direct local descriptor.
DV	4	See Table 34.14.	Copied from direct local descriptor. See (1) Direct local descriptor.
—	—	Reserved area	Set to 0

(2) Reception ethernet descriptor

Reception ethernet descriptor corresponds to frames that has been by the MFWD. Reception ethernet descriptor INFO and TS formats are described in Figure 34.72 and Table 34.29.

INFO0

Byte	Bit3	Bit2	Bit1	Bit0
0	—		SEC	FI

INFO1

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	—	TN	CRT	IET	TXC	FMT	SEC	FI
1	TSUN[7:0]							
2	SAEF[7:0]							
3	RN[7:0]							
4	—	SPN[2:0]			RV	—		
5	FESF[23:0]							
6								
7								

TS

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	TSNS[29:0]							
1								
2								
3	TSD	TSV						
4	TSS[31:0]							
5								
6								
7								

Figure 34.72 Reception ethernet descriptor format INFO and TS formats

Table 34.29 Reception ethernet descriptor format INFO and TS field descriptions (1 of 2)

Field name	Bit width	Description	Value for GWCA
FI	1	See Table 34.15.	See section 8.3.4.5.

Table 34.29 Reception ethernet descriptor format INFO and TS field descriptions (2 of 2)

Field name	Bit width	Description	Value for GWCA
SEC	1	See Table 34.15.	Copied from ethernet local descriptor from [GWCA Data transmission] or [ETHA Data reception]
FMT	1	See Table 34.15.	Copied from ethernet local descriptor from [GWCA Data transmission] or [ETHA Data reception]
TXC	1	See Table 34.15.	Copied from ethernet local descriptor from [GWCA Data transmission] or [ETHA Data reception]
IET	1	See Table 34.15.	Copied from ethernet local descriptor from [GWCA Data transmission] or [ETHA Data reception]
CRT	1	See Table 34.15.	Copied from ethernet local descriptor from [GWCA Data transmission] or [ETHA Data reception]
TN	1	See Table 34.15.	Copied from ethernet local descriptor from [GWCA Data transmission] or [ETHA Data reception]
TSUN	8	See Table 34.15.	Copied from ethernet local descriptor from [GWCA Data transmission] or [ETHA Data reception]
SAEF	8	See Table 34.17.	Copied from ethernet local descriptor from [GWCA Data transmission] or [ETHA Data reception]
RV	1	See Table 34.15.	Copied from received descriptor FDESCR.RV field [MFWD]
RN	5	See Table 34.15.	Copied from received descriptor FDESCR.RN field [MFWD]
SPN	2	Source port number	Port number from which the frame entered the switch [MFAB] Copied from received descriptor FDESCR.SPN field [MFWD]
FESF	24	Forwarding engine status flags	Copied from received descriptor **FDESCR.MINFO** field [MFWD]
TSV	1	Timestamp valid	Copied from ethernet local descriptor from [GWCA Data transmission] or [ETHA Data reception]
TSD	1	Timestamp default	Copied from ethernet local descriptor from [GWCA Data transmission] or [ETHA Data reception]
TSNS	30	Timestamp nanosecond [gPTP] PCH header timestamp [29:0]	Copied from ethernet local descriptor from [GWCA Data transmission] or [ETHA Data reception]
TSS	32	Timestamp second [gPTP] PCH header timestamp [31:30] Restrictions: ● HW: The 16-upper bits of the gPTP second part are truncated.	Copied from ethernet local descriptor from [GWCA Data transmission] or [ETHA Data reception]
—	—	Reserved area	Set to 0

34.5.4 Timestamp Reception [gPTP]

GWCA allows Timestamp reception through the GWCA TS path, The TS path is described in Figure 34.73.

Timestamps received through this function are only Ethernet agent TX timestamps. Ethernet agent RX timestamps are stored in the local RAM along with their corresponding data and can only be received by CPU through RX data path with the corresponding data.

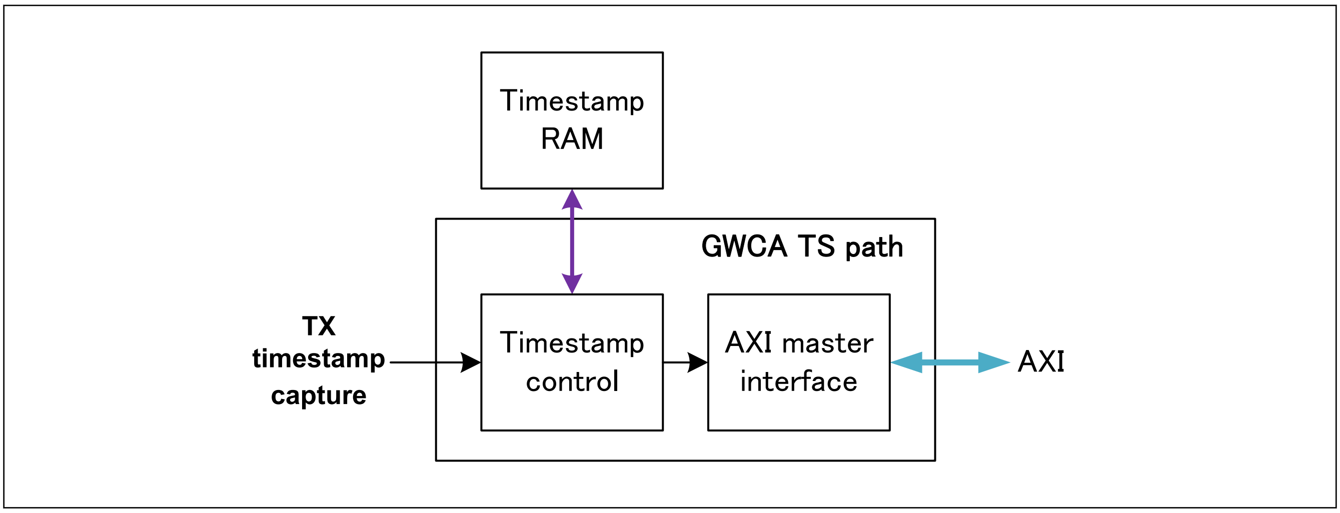


Figure 34.73 GWCA TS path block diagram

The TS path is separated in two blocks:

- Timestamp control: This block aims are receiving the timestamps from TX Timestamp Capture interface [RMAC], storing them in the Timestamp RAM and sending them to the AXI master interface.
- AXI master interface: This block handles the timestamp/AXI descriptor exchange with the CPU.

34.5.4.1 Timestamp Control

Timestamp control controls the descriptor storage/fetching to/from the Timestamp RAM using GWTSDCCs.TE register and can be monitored using GWTSNM and GWTSMNM registers and GWEIS0.TSOVFES interrupt registers.

Functions:

- GWTSDCCs.TE is used to enable timestamp reception for timer s. If enabled, all timestamps received from TX Timestamp Capture interfaces [RMAC] for timer s will be stored in the timestamp RAM and later given to the AXI master interface.
- GWTSNM register is used to monitor the current number of timestamps present in the timestamp RAM.
- GWTSMNM register is used to monitor the maximum number of timestamps that has been held in the timestamp RAM since the previous reset or the previous time corresponding register has been read.
- GWEIS0.TSOVFES interrupt register is used to flag a timestamp loss because of Timestamp RAM overflow.

34.5.4.2 AXI Master Interface

The AXI master interface handles the timestamp/AXI descriptor exchange with the CPU. It will receive the timestamps and the information related to the timestamps and store them in the CPU RAM. To do so, it uses GWTDCACs0, WTDCACs1, and GWTSDCCs.DCS registers and it can be monitored using GWTSDIS.TSDIS and GWEIS0.TDFES interrupt registers.

Functions:

- GWTDCACs0 and GWTDCACs1 registers form the AXI address {GWTDCACs0.TSCCAU, GWTDCACs1.TSCCAL} with correspond to the current address being processed by the GWCA for TS descriptor queue s. It can be used to set the first address of the descriptor queue during GWCA initialization, move an under-use TS descriptor queue to another location in the CPU RAM or monitor the currently processed descriptor address.
- GWTSDCCs.DCS register is used to map timer s timestamps to a TS descriptor queue number.
- GWTSDIS.TSDIS interrupt register is used to flag some timestamps are ready in the CPU RAM.
- GWEIS0.TDFES interrupt register is used to flag a timestamp loss because the timestamp descriptor queue is full.

Timestamp reception can be done using linear or cyclic descriptor queues (see section 34.5.1.5. AXI Descriptor Queue Format). Timestamp reception process descriptor utilization is described in Figure 34.74. Timestamp reception process descriptor format is described in Figure 34.75 and Table 34.30.

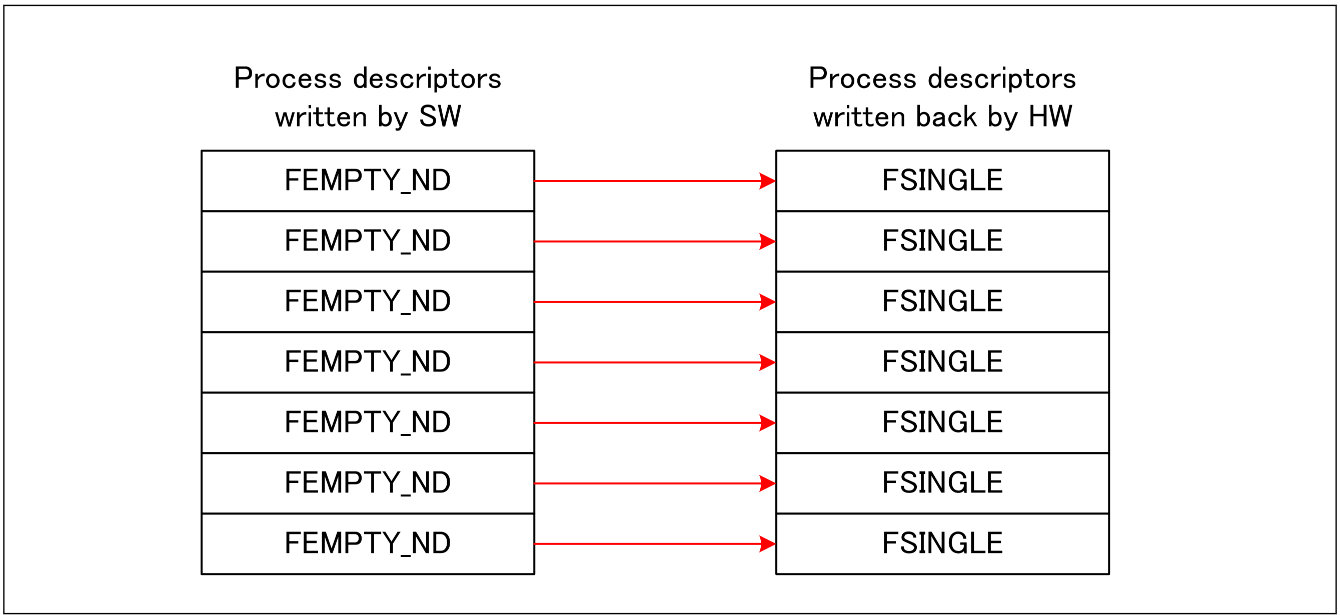


Figure 34.74 TS reception process descriptor utilization

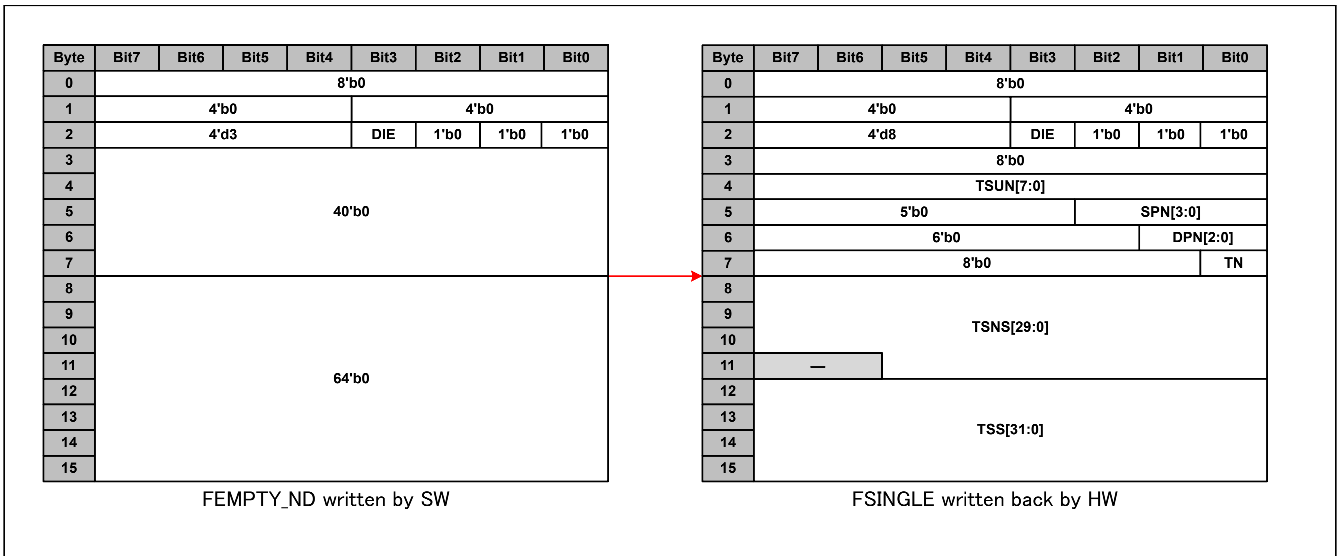


Figure 34.75 TS reception process descriptor utilization format

Table 34.30 TS process descriptor utilization field description (1 of 2)

Field name	Field description written by SW	Field description after HW write-back
DS	0	0
INFO0	0	0
ERR	0	0
DSE	0	0
AXIE	0	0
DIE	DIE	DIE (Value is copied from read descriptor).
DT	3 for FEMPTY_ND	8 for FSINGLE
PTR	0	Used as timestamp related information field See Table 34.31
INFO1	NA: Info1 is never present for TS reception	NA: Info1 is never present for TS reception

Table 34.30 TS process descriptor utilization field description (2 of 2)

Field name	Field description written by SW	Field description after HW write-back
TS	0	See Table 34.32

Table 34.31 TS reception process descriptor PTR field explanation

Field name	Bit width	Description
TSUN	8	Timestamp unique number [RMAC]
SPN	2	Port number from which the timestamp corresponding frame entered the switch [MFAB]
DPN	1	Port number by which the timestamp has been taken
TN	1	Timer Number

Table 34.32 TS reception process descriptor TS field explanation

Field name	Bit width	Description
TSNS	30	Timestamp nanosecond [gPTP]
TSS	32	Timestamp second [gPTP] Restrictions: ● HW: The 16-upper bits of the gPTP second part are truncated.

34.5.5 gPTP Triggered Transmission

It is possible to use gPTP timer to trig a periodic transmission start using time-controlled data transmission (see Table 34.13). To do so, CYCLIC_COMP[i][c] (Where i is 0 for GWCA0) functional interface is linked to INTER_TX[8+c] (see Figure 34.21).

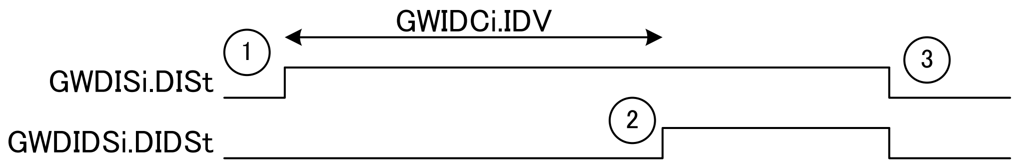
For 8 parameter explanation, see gPTP specification [gPTP].

34.5.6 Interrupt Delay Function

Interrupt delay function aims at delaying data interrupts (GWDISi.DISt) to accumulate data during a certain among of time before requesting software to process the data. As a result, software load is diminished, and data latency is still guaranteed. To operate, interrupt delay function uses GWIDPC and GWIDCi registers and can be monitored using GWDIDSi interrupt register. Interrupt delay function is described in Figure 34.76.

Functions:

- GWIDPC register sets an internal clock of period *clk_delay_period* used as base clock for interrupt delay function.
- GWIDCi register sets the delay time for descriptor queue i in *clk_delay_period*.
- GWDIDSi.DIDSt register is the delayed GWDISi.DISt interrupt register and triggers the interrupt gwc_gwdis_int[t].
- For interrupt handling, software can decide to read GWDIDSi register and to process only interrupts that have been set until their corresponding delay time is archived (allows a certain amount of data surely came before processing) or can decide to read GWDISi register and to process all the interrupts even the ones which did not reach their delay time (allows a certain amount of data surely came before processing at least for one queue can diminish the interrupt load).



1	GWDISi.DISt flag is set
2	After GWIDCi.IDV**clk_delay_period* GWDIDSi.DIDSt is asserted with the interrupt if GWDIEi.DIEt is set to 1
3	SW clears GWDISi.DISt flag, it de-asserts the interrupt

Figure 34.76 Interrupt delay function

Restrictions:

- HW: The maximum delay achievable by the interrupt delay function is clk_period*4194304. For example, it will be around 10 ms for a clock clk with a frequency of 400 MHz.

34.6 Precautions

- AXI address bus width
  The AXI address bus width degenerated from 40 bits to 32 bits by products connection. Please use AXI descriptor address with only 32 bits. Please fix to all 0 for upper 8 bits for AXI address. (For example, cannot use GWDCBAC0.)

35. Ethernet Generic PTP Timer (GPTP)

35.1 Overview

Ethernet Generic PTP timer (GPTP) enables accurate synchronization of the clock in the control system.

Table 35.1 lists the GPTP specifications and Figure 35.1 shows a block diagram.

Table 35.1 GPTP specifications

Item	Function	Description
Overall function	Timer	• GPTP and AVTP timer
Data provision	—	• GPTP timer Data width: 78 bits ([77:30] = second, [29:0] = nanosecond) • AVTP timer Data width: 64 bits/32 bits
Functions	Timer control	• GPTP timer incremental value control • GPTP timer offset control • Generation of GPTP and AVTP timer
	PPS	• PPS signal generation
	Media clock	• Media clock capture • Media clock recovery
	Cyclic compare	• Cyclic comparison to emit pulses

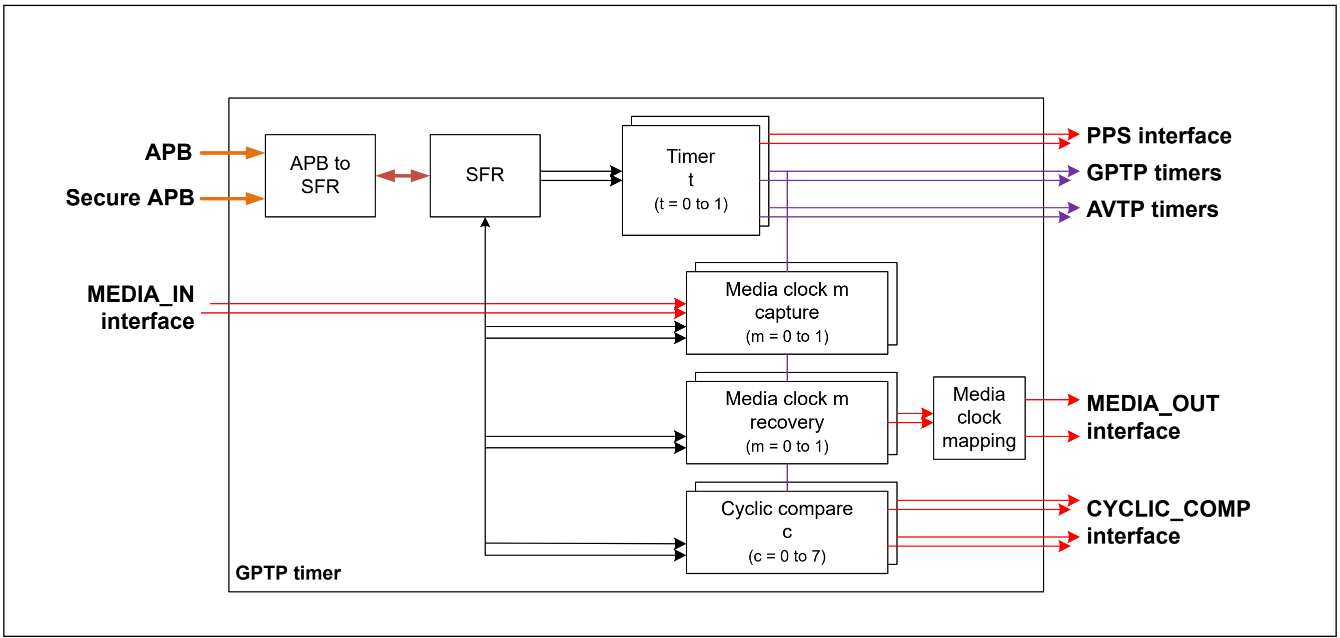


Figure 35.1 GPTP block diagram

Table 35.2 GPTP functional blocks

Block name	Function
APB to SFR	Handle SFR accesses
SFR	GPTP SFRs
Timer	Generate GPTP an AVTP timers
Media clock capture	Capture input media clocks
Media clock recovery	Recover output media clocks
Cyclic compare	Generate a cyclic pulse

35.2 gPTP Register List

Table 35.3 gPTP timer registers

Offset address	Register name	Register symbol	Write access mode
0x0000	IP Version Register	PTPIPV	—
0x0010	Timer Enable Configuration Register	PTPTMEC	—
0x0014	Timer Disable Configuration Register	PTPTMDC	—
0x0020 + 0x40 × t	Timer t Increment Value Configuration Register (t = 0, 1)	PTPTIVCt	—
0x0030 + 0x40 × t	Timer t Offset Value Configuration Register L (t = 0, 1)	PTPTOVCtL	—
0x0034 + 0x40 × t	Timer t Offset Value Configuration Register M (t = 0, 1)	PTPTOVCtM	—
0x0038 + 0x40 × t	Timer t Offset Value Configuration Register U (t = 0, 1)	PTPTOVCtU	—
0x0040 + 0x40 × t	AVTP Timer t Monitoring Register L (t = 0, 1)	PTPAVTPTMtL	—
0x0044 + 0x40 × t	AVTP Timer t Monitoring Register U (t = 0, 1)	PTPAVTPTMtU	—
0x0050 + 0x40 × t	GPTP Timer t Monitoring Register L (t = 0, 1)	PTPGPTPTMtL	—
0x0054 + 0x40 × t	GPTP Timer t Monitoring Register M (t = 0, 1)	PTPGPTPTMtM	—
0x0058 + 0x40 × t	GPTP Timer t Monitoring Register U (t = 0, 1)	PTPGPTPTMtU	—
0x0200 + 0x10 × m	Media Clock m Capture Configuration Register (m = 0, 1)	PTPMCCCm	—
0x0204 + 0x10 × m	Media Clock m Capture Monitoring Register L (m = 0, 1)	PTPMCCMmL	—
0x0208 + 0x10 × m	Media Clock m Capture Monitoring Register M (m = 0, 1)	PTPMCCMmM	—
0x020C + 0x10 × m	Media Clock m Capture Monitoring Register U (m = 0, 1)	PTPMCCMmU	—
0x0300 + 0x10 × m	Media Clock m Recovery Configuration Register (m = 0, 1)	PTPMCRCm	—
0x0304 + 0x10 × m	Media Clock m Recovery Time Configuration Register L (m = 0, 1)	PTPMCRTCmL	—
0x0308 + 0x10 × m	Media Clock m Recovery Time Configuration Register M (m = 0, 1)	PTPMCRTCmM	—
0x030C + 0x10 × m	Media Clock m Recovery Time Configuration Register U (m = 0, 1)	PTPMCRTCmU	—
0x0400 + 0x04 × m	Media Clock m Pin Configuration Register (m = 0, 1)	PTPMCPCm	—
0x0500 + 0x08 × c	Cyclic Compare c Configuration Register 0 (c = 0 to 7)	PTPCCCc0	—
0x0504 + 0x08 × c	Cyclic Compare c Configuration Register 1 (c = 0 to 7)	PTPCCCc1	—
0x0700	Interrupt Status Register 0	PTPIS0	—
0x0704	Interrupt Enable Register 0	PTPIE0	—
0x0708	Interrupt Disable Register 0	PTPID0	—
0x0710	Interrupt Status Register 1	PTPIS1	—
0x0714	Interrupt Enable Register 1	PTPIE1	—
0x0718	Interrupt Disable Register 1	PTPID1	—
0x0780	Security Configuration Register 0	PTPSCR0	—
0x0784	Security Configuration Register 1	PTPSCR1	—
0x0788	Security Configuration Register 2	PTPSCR2	—

Note: Base address: GPTP = 0x403E_0000, GPTP_NS = 0x503E_0000

35.3 gPTP Register Descriptions

35.3.1 Version Register

35.3.1.1 PTPIPV : IP Version Register

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0000

Bit position:	31 … 0
Bit field:	IPV[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1

Bit	Symbol	Function	R/W
31:0	IPV[31:0]	IP Version	R

35.3.2 Timer Configuration Registers

35.3.2.1 PTPTMEC : Timer Enable Configuration Register

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TE1 to TE0	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	TE1 to TE0	Timer q Enable (q = 0, 1) Bit q enables timer q. 0: Timer q disabled 1: Timer q enabled	R/W*1
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

TEq bits (Timer q Enable (q = 0, 1))

When timer q is disabled, all counters are fixed to 0.

[Setting condition]

- SW: Writing 1 to one of these bits.

[Clearing condition]

- SW: Writing 1 to PTPTMDC.TDq clears this register bit q.

35.3.2.2 PTPTMDC : Timer Disable Configuration Register

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TD1 to TD0	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	TD1 to TD0	Timer n Disable (n = 0, 1) 0: No effect 1: Bit q of this register clears PTPTMEC.TEq	W*1
31:2	—	These bits are read as 0.	R

Note 1. Read value differs from written value.

TDn bits (Timer n Disable (n = 0, 1))

[Clearing condition]

- SW: Writing 1 to bit q of this register clears PTPTMEC.TEq.

35.3.2.3 PTPTIVCt : Timer t Increment Value Configuration Register (t = 0, 1)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0020 + 0x40 × t

Bit position:	31 ... 0
Bit field:	TIV[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TIV[31:0]	Timer Increment Value	R/W

TIV[31:0] bits (Timer Increment Value)

Timer increment value in ns per clock clk. PTPTIVCt.TIV[31:27] is the nanosecond part and PTPTIVCt.TIV[26:0] is the nanosecond sub-part.

[Examples]

- If clk clock frequency is 50 MHz, 100 MHz, 200 MHz or 250 MHz, PTPTIVCt.TIV should be respectively set to a value around 0xA000_0000, 0x5000_0000, 0x2800_0000 or 0x2000_0000.

35.3.2.4 PTPTOVCtL : Timer t Offset Value Configuration Register L (t = 0, 1)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0030 + 0x40 × t

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	TOVL[29:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TOVL[29:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
29:0	TOVL[29:0]	Timer Offset Value Lower Part*1	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. SW: Only value from 0x0000_0000 to 0x3b9a_c9ff can be set to this register.

TOVL[29:0] bits (Timer Offset Value Lower Part)

- Timer offset value in GPTP format. PTPTOVCtL.TOVL represents its nanosecond part.
- Writing to this register loads the new offset to timer t.
- {PTPTOVCtU.TOVU, PTPTOVCtM.TOVM, 0, PTPTOVCtL.TOVL} is timer t offset.

35.3.2.5 PTPTOVCMt : Timer t Offset Value Configuration Register M (t = 0, 1)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0034 + 0x40 × t

Bit position:	31 ... 0
Bit field:	TOVM[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	TOVM[31:0]	Timer Offset Value Middle Part	R/W*1

Note 1. Read value differs from written value.

TOVM[31:0] bits (Timer Offset Value Middle Part)

- Timer offset value in GPTP format. PTPTOVCtM.TOVM represents its second lower 32-bit part.
- {PTPTOVCtU.TOVU, PTPTOVCtM.TOVM, 0, PTPTOVCtL.TOVL} is timer t offset.

[Update condition]

- SW: Writing to the PTPTOVCtL register updates this register to the latest value

35.3.2.6 PTPTOVCtU : Timer t Offset Value Configuration Register U (t = 0, 1)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0038 + 0x40 × t

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TOVU[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TOVU[15:0]	Timer Offset Value Upper Part	R/W*1
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

TOVU[15:0] bits (Timer Offset Value Upper Part)

- Timer offset value in GPTP format. PTPTOVCtU.TOVU represents its second higher 16-bit part.
- {PTPTOVCtU.TOVU, PTPTOVCtM.TOVM, 0, PTPTOVCtL.TOVL} is timer t offset.

[Update condition]

- SW: Writing to PTPTOVCtL register updates this register to its latest value.

35.3.3 Timer Monitoring Registers

35.3.3.1 PTPAVTPTMtL : AVTP Timer t Monitoring Register L (t = 0, 1)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0040 + 0x40 × t

Bit position:	31 … 0
Bit field:	AVTPL[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	AVTPL[31:0]	AVTP Timer Value Lower Part	R

AVTPL[31:0] bits (AVTP Timer Value Lower Part)

- AVTP time is in nanoseconds and derives from GPTP timer t. PTPAVTPTMtL.AVTPL represents either timer t 32-bit AVTP timer or timer t 64-bit AVTP timer lower bits.
- This register is always equal to timer t 32-bit AVTP timer or timer t 64-bit AVTP timer lower bits.

35.3.3.2 PTPAVTPTMtU : AVTP Timer t Monitoring Register U (t = 0, 1)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0044 + 0x40 × t

Bit position:	31	0
Bit field:	AVTPU[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	AVTPU[31:0]	AVTP Timer Value Upper Part	R

AVTPU[31:0] bits (AVTP Timer Value Upper Part)

AVTP time is in nanoseconds and derives from GPTP timer t. PTPAVTPTMtU.AVTPU represents timer t 64-bit AVTP timer higher bits.

[Update condition]

- HW: Reading PTPAVTPTMtL register updates this register to timer t 64-bit AVTP timer higher bits.

35.3.3.3 PTPGPTPTMtL : GPTP Timer t Monitoring Register L (t = 0, 1)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0050 + 0x40 × t

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	GPTPL[29:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	GPTPL[29:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
29:0	GPTPL[29:0]	GPTP Timer Value Lower Part ● PTPGPTPTMtL.GPTPL represents timer t GPTP timer nanosecond part ● This register is always equal to timer t GPTP timer nanosecond part.	R
31:30	—	These bits are read as 0.	R

35.3.3.4 PTPGPTPTMtM : GPTP Timer t Monitoring Register M (t = 0, 1)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0054 + 0x40 × t

Bit position:	31	0
Bit field:	GPTPM[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	GPTPM[31:0]	GPTP Timer Value Middle Part	R

GPTPM[31:0] bits (GPTP Timer Value Middle Part)

PTPGPTPTMtM.GPTPM represents timer t GPTP timer second lower 32 bit part.

[Update condition]

- HW: Reading PTPGPTPTMtL register updates this register to timer t GPTP timer second lower 32-bit part.

35.3.3.5 PTPGPTPTMtU : GPTP Timer t Monitoring Register U (t = 0, 1)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0058 + 0x40 × t

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	GPTPU[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	GPTPU[15:0]	GPTP Timer Value Upper Part	R
31:16	—	These bits are read as 0.	R

GPTPU[15:0] bits (GPTP Timer Value Upper Part)

PTPGPTPTMtU.GPTPU represents timer t GPTP timer second higher 16-bit part.

[Update condition]

- HW: Reading PTPGPTPTMtL register updates this register to timer t GPTP timer second higher 16-bit part.

35.3.4 Media Clock Capture Registers

35.3.4.1 PTPMCCCm : Media Clock m Capture Configuration Register (m = 0, 1)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0200 + 0x10 × m

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCCR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MCTNS	MCTTS	MCNEE	MCPEE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MCPEE	Media Clock Capture Positive Edge Enable 0: Timer value is not captured on positive edge of MEDIA_IN[m] signal 1: Timer value is captured on positive edge of MEDIA_IN[m] signal	R/W

Bit	Symbol	Function	R/W
1	MCNEE	Media Clock Capture Negative Edge Enable 0: Timer value is not captured on negative edge of MEDIA_IN[m] signal 1: Timer value is captured on negative edge of MEDIA_IN[m] signal	R/W
2	MCTTS	Media Clock Capture Timer Type Select 0: Timer value is captured on GPTP timer 1: Timer value is captured on AVTP timer	R/W
3	MCTNS	Media Clock Capture Timer Number Select Selects from which timer the timer values are taken.	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W
16	MCCR	Media Clock Capture Request Writing 1 to this register triggers timer capture.	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

35.3.4.2 PTPMCCMmL : Media Clock m Capture Monitoring Register L (m = 0, 1)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0204 + 0x10 × m

Bit position:	31 ... 0
Bit field:	MCCTVL[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	MCCTVL[31:0]	Media Clock Captured Timer Value Lower Part	R

MCCTVL[31:0] bits (Media Clock Captured Timer Value Lower Part)

- If a GPTP timer is captured, PTPMCCMmL.MCCTVL[29:0] represent the GPTP timer nanosecond part
- If an AVTP timer is captured, this register represents either the 32-bit AVTP timer or the 64-bit AVTP lower 32 bits.

[Update condition]

- Reading PTPMCCMmL register updates PTPMCCMmL, PTPMCCMmM and PTPMCCMmU to the next captures value
- Capturing a timer while the capture buffer is empty updates PTPMCCMmL, PTPMCCMmM and PTPMCCMmU to the captured value.

35.3.4.3 PTPMCCMmM : Media Clock m Capture Monitoring Register M (m = 0, 1)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0208 + 0x10 × m

Bit position:	31 ... 0
Bit field:	MCCTVM[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	MCCTVM[31:0]	Media Clock Captured Timer Value Middle Part	R

MCCTVM[31:0] bits (Media Clock Captured Timer Value Middle Part)

- If a GPTP timer is captured, this register represents the GPTP timer second part 32 lower bits

- If an AVTP timer is captured, this register represents the 64-bit AVTP higher 32 bits.

[Update condition]

- Reading PTPMCCMmL register updates PTPMCCMmL, PTPMCCMmM and PTPMCCMmU to the next captured value
- Capturing a timer while the capture buffer is empty updates PTPMCCMmL, PTPMCCMmM and PTPMCCMmU to the captured value.

35.3.4.4 PTPMCCMmU : Media Clock m Capture Monitoring Register U (m = 0, 1)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x020C + 0x10 × m

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	MCCN[1:0]		—	—	—	—	—	MCSWC	MCNEC	MCPEC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MCCTVU[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	MCCTVU[15:0]	Media Clock Captured Timer Value Upper Part	R
16	MCPEC	Media Clock Positive Edge Captured	R
17	MCNEC	Media Clock Negative Edge Captured	R
18	MCSWC	Media Clock Software Captured	R
23:19	—	These bits are read as 0.	R
25:24	MCCN[1:0]	Media Clock Capture Number	R
31:26	—	These bits are read as 0.	R

MCCTVU[15:0] bits (Media Clock Captured Timer Value Upper Part)

- If a GPTP timer is captured, this register represents the GPTP timer second part higher 16 bits
- If an AVTP timer is captured, this register is unused.

[Update condition]

- Reading PTPMCCMmL register updates PTPMCCMmL, PTPMCCMmM and PTPMCCMmU to the next captured value
- Capturing a timer while the capture buffer is empty updates PTPMCCMmL, PTPMCCMmM and PTPMCCMmU to the captured value.

MCPEC bit (Media Clock Positive Edge Captured)

[Update condition]

- Reading PTPMCCMmL register updates PTPMCCMmL, PTPMCCMmM and PTPMCCMmU to the next captured value
- Capturing a timer while the capture buffer is empty updates PTPMCCMmL, PTPMCCMmM and PTPMCCMmU to the captured value.

MCNEC bit (Media Clock Negative Edge Captured)

[Update condition]

- Reading PTPMCCMmL register updates PTPMCCMmL, PTPMCCMmM and PTPMCCMmU to the next captured value
- Capturing a timer while the capture buffer is empty updates PTPMCCMmL, PTPMCCMmM and PTPMCCMmU to the captured value.

MCSWC bit (Media Clock Software Captured)

[Update condition]

- Reading PTPMCCMmL register updates PTPMCCMmL, PTPMCCMmM and PTPMCCMmU to the next captured value
- Capturing a timer while the capture buffer is empty updates PTPMCCMmL, PTPMCCMmM and PTPMCCMmU to the captured value.

MCCN[1:0] bits (Media Clock Capture Number)

[Increment condition]

- HW: Incremented by one when Media Capture Unit m captures a timer other than 2.

[Decrement condition]

- SW: Decremented by one when reading by software is executed other than 0.

35.3.5 Media Clock Recovery Registers

35.3.5.1 PTPMCRCm : Media Clock m Recovery Configuration Register (m = 0, 1)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0300 + 0x10 × m

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MRPL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	MRTNS	MRAMS	MRTTS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MRTTS	Media Clock Recovery Timer Type Select*1 0: Timer value is compared to GPTP timer 1: Timer value is compared to AVTP timer	R/W
1	MRAMS	Media Clock Recovery AVTP Mode Select*2 0: 64-bit AVTP is used for comparison 1: 32-bit AVTP is used for comparison	R/W
2	MRTNS	Media Clock Recovery Timer Number Select Selects from which timer the timer values are taken.	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W
31:16	MRPL[15:0]	Media Clock Recovery Pulse Length Pulse length for clock recovery in clk clocks - 1.	R/W

Note 1. Restrictions:

- HW: For GPTP timer, overflow is not supported. If a time smaller than the current GPTP timer is set, the comparison instantaneously matches.

Note 2. Restrictions:

- HW: For 64-bit AVTP timer, overflow is not supported. If a time smaller than the current AVTP timer is set, the comparison instantaneously matches.

35.3.5.2 PTPMCRTCmL : Media Clock m Recovery Time Configuration Register L (m = 0, 1)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0304 + 0x10 × m

Bit position:	31 ... 0
Bit field:	MRTVL[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	MRTVL[31:0]	Media Clock Recovery Timer Value Lower Part	R/W

MRTVL[31:0] bits (Media Clock Recovery Timer Value Lower Part)

- If the compared timer is a GPTP timer, this register represents the GPTP timer nanosecond part
- If the compared timer is a 32-bit AVTP timer, this register represents the 32-bit AVTP timer
- If the compared timer is a 64-bit AVTP timer, this register represents the 64-bit AVTP lower 32 bits
- Writing to this register loads MCCTVU, MCCTVM and MCCTVL registers for comparison in the recovery buffer.

35.3.5.3 PTPMCRTCmM : Media Clock m Recovery Time Configuration Register M (m = 0, 1)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0308 + 0x10 × m

Bit position:	31 ... 0
Bit field:	MRTVM[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	MRTVM[31:0]	Media Clock Recovery Timer Value Middle Part	R/W

MRTVM[31:0] bits (Media Clock Recovery Timer Value Middle Part)

- If the compared timer is a GPTP timer, this register represents the GPTP timer second part 32 lower bits
- If the compared timer is a 32-bit AVTP timer, this register represents nothing relevant
- If the compared timer is a 64-bit AVTP timer, this register represents the 64-bit AVTP higher 32 bits.

35.3.5.4 PTPMCRTCmU : Media Clock m Recovery Time Configuration Register U (m = 0, 1)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x030C + 0x10 × m

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MRBCR	—	—	—	—	—	—	—	—	—	—	MCRN[2:0]			MRTT[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MRTVU[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	MRTVU[15:0]	Media Clock Recovery Timer Value Upper Part	R/W
17:16	MRTT[1:0]	Media Clock Recovery Trigger Type 0 0: When timer match, time recovery pin is set to 1 0 1: When timer match, time recovery pin is set to 0 1 0: When timer match, time recovery pin is inverted 1 1: When timer match, a positive pulse of length PTPMCRCm.MRPL is output on time recovery pin	R/W
20:18	MCRN[2:0]	Media Clock Recovery Number Displays the number on entry in the recovery buffer.	R
30:21	—	These bits are read as 0. The write value should be 0.	R/W
31	MRBCR	Media Clock Recovery Buffer Clear Request Writing one to this register clears the internal timer buffer along with PTPMCRTCmU.MCRN register.	R

MRTVU[15:0] bits (Media Clock Recovery Timer Value Upper Part)

- If the compared timer is a GPTP timer, this register represents the GPTP timer second part 16 higher bits
- If the compared timer is a 32-bit AVTP timer, this register represents nothing relevant
- If the compared timer is a 64-bit AVTP timer, this register represents nothing relevant.

35.3.5.5 PTPMCPCm : Media Clock m Pin Configuration Register (m = 0, 1)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0400 + 0x04 × m

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MRS	PE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PE	Pin Enable*1 0: MEDIA_OUT[p] fixed to 0 1: MEDIA_OUT[p] outputs the comparison result of PTPMCPCm.MRS Recovery logic	R/W
1	MRS	Media Clock Recovery Select*1 Select Recovery logic for MEDIA_OUT[p].	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Restrictions:

- Do not change this setting on any timer enabled (PTPTMEC.TEq = 1).

35.3.6 Cyclic Compare Registers

35.3.6.1 PTPCCCc0 : Cyclic Compare c Configuration Register 0 (c = 0 to 7)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0500 + 0x8 × c

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	CCOPS	—	—	—	CCTNS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CCTNS	Cyclic Compare Timer Number Select Selects from which timer the cyclic compare is based on.	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	CCOPS	Cyclic Compare Output Pin Select CYCLIC_COMP[PTPCCCc0.CCOPS][c] is selected as output for cyclic compare c.	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

35.3.6.2 PTPCCCc1 : Cyclic Compare c Configuration Register 1 (c = 0 to 7)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0504 + 0x8 × c

Bit position:	31 … 0
Bit field:	CCV[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	CCV[31:0]	Cycle Compare Value ● Setting this bit to a value smaller than 32 disables the compare function ● Setting this bit to a value larger or equal to 32 enables the compare function.	R/W

35.3.7 Interrupt Registers

35.3.7.1 PTPIS0 : Interrupt Status Register 0

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0700

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCCOES1 to MCCOES0	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCCS1 to MCCS0	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	MCCS1 to MCCS0	Media Clock n Capture Status Flag (n = 0, 1)	R/W*1
15:2	—	These bits are read as 0. The write value should be 0.	R/W
17:16	MCCOES1 to MCCOES0	Media Clock n Capture Overflow Error Status Flag (n = 0, 1)	R/W*1
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

MCCSn bits (Media Clock n Capture Status Flag (n = 0, 1))

[Setting condition]

- HW: Media Capture Unit m captured a timer.

[Clearing condition]

- SW: Writing 1 to one of these bits.

MCCOESn bits (Media Clock n Capture Overflow Error Status Flag (n = 0, 1))

[Setting condition]

- HW: Bit m is set when Media Capture Unit m captured a timer, but it cannot be stored because the timer FIFO is full (PTPMCCMmU.MCCN equal to 2).

[Clearing condition]

- SW: Writing 1 to one of these bits.

[Error recovery]

- HW: The captured timer is lost, and the next capture continues normally
- SW: The timer reading is too slow, and software should be reviewed.

35.3.7.2 PTPIE0 : Interrupt Enable Register 0

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0704

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCCOEE1 to MCCOEE0	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCCE1 to MCCE0	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	MCCE1 to MCCE0	Media Clock m Capture Enable (m = 0, 1) Bit m enables media clock m capture 0: Interrupt m disabled 1: Interrupt m enabled	R/W*1
15:2	—	These bits are read as 0. The write value should be 0.	R/W
17:16	MCCOEE1 to MCCOEE0	Media Clock m Capture Overflow Error Enable (m = 0, 1) Bit m enables media clock m capture overflow error. 0: Interrupt m disabled 1: Interrupt m enabled	R/W*1
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

MCCEm bits (Media Clock m Capture Enable (m = 0, 1))

[Setting condition]

- SW: Writing 1 to one of these bits.

[Clearing condition]

- SW: Writing 1 to bit m of PTPID0.MCCD register clears the corresponding bit of this register.

MCCOEEm bits (Media Clock m Capture Overflow Error Enable (m = 0, 1))

[Setting condition]

- SW: Writing 1 to one of these bits.

[Clearing condition]

- SW: Writing 1 to bit m of PTPID0.MCCOED register clears the corresponding bit of this register.

35.3.7.3 PTPID0 : Interrupt Disable Register 0

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0708

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCCOED1 to MCCOED0	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCCD1 to MCCD0	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	MCCD1 to MCCD0	Media Clock m Capture Disable (m = 0, 1)	R
15:2	—	These bits are read as 0.	R
17:16	MCCOED1 to MCCOED0	Media Clock m Capture Overflow Error Disable (m = 0, 1)	R
31:18	—	These bits are read as 0.	R

MCCDm bits (Media Clock m Capture Disable (m = 0, 1))

[Clearing condition]

- SW: Writing 1 to one of these bits clears the corresponding bit of PTPIE0.MCCE register.

MCCOEDm bits (Media Clock m Capture Overflow Error Disable (m = 0, 1))

[Clearing condition]

- SW: Writing 1 to one of these bits clears the corresponding bit of PTPIE0.MCCOEE register.

35.3.7.4 PTPIS1 : Interrupt Status Register 1

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0710

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCRMS1 to MCRMS0	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	MCRMS1 to MCRMS0	Media Clock m Recovery Match Status Flag (m = 0, 1)	R/W*1
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

MCRMSm bits (Media Clock m Recovery Match Status Flag (m = 0, 1))

[Setting condition]

- HW: The oldest timer in Media Capture Unit m is matched.

[Clearing condition]

- SW: Writing 1 to one of these bits.

35.3.7.5 PTPIE1 : Interrupt Enable Register 1

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0714

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCRME1 to MCRME0	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	MCRME1 to MCRME0	Media Clock m Recovery Match Enable (m = 0, 1) Bit m enables media clock m recovery match. 0: Interrupt m disabled 1: Interrupt m enabled	R/W*1
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Read value differs from written value.

MCRMEm bits (Media Clock m Recovery Match Enable (m = 0, 1))

[Setting condition]

- SW: Writing 1 to one of these bits.

[Clearing condition]

- SW: Writing 1 to bit m of PTPID1.MCRMD register clears the corresponding bit of this register.

35.3.7.6 PTPID1 : Interrupt Disable Register 1

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x0718

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCRMD1 to MCRMD0	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	MCRMD1 to MCRMD0	Media Clock m Recovery Match Disable (m = 0, 1)	R
31:2	—	These bits are read as 0.	R

MCRMDm bits (Media Clock m Recovery Match Disable (m = 0, 1))

[Clearing condition]

- SW: Writing 1 to one of these bits clears the corresponding bit of PTPIE0.MCRME register.

35.4 Register Utilization

35.4.1 Software Flows

SW: Follow the flows described in this section.

35.4.1.1 Software Flow Legend

Figure 35.2 describes the software flow legend.

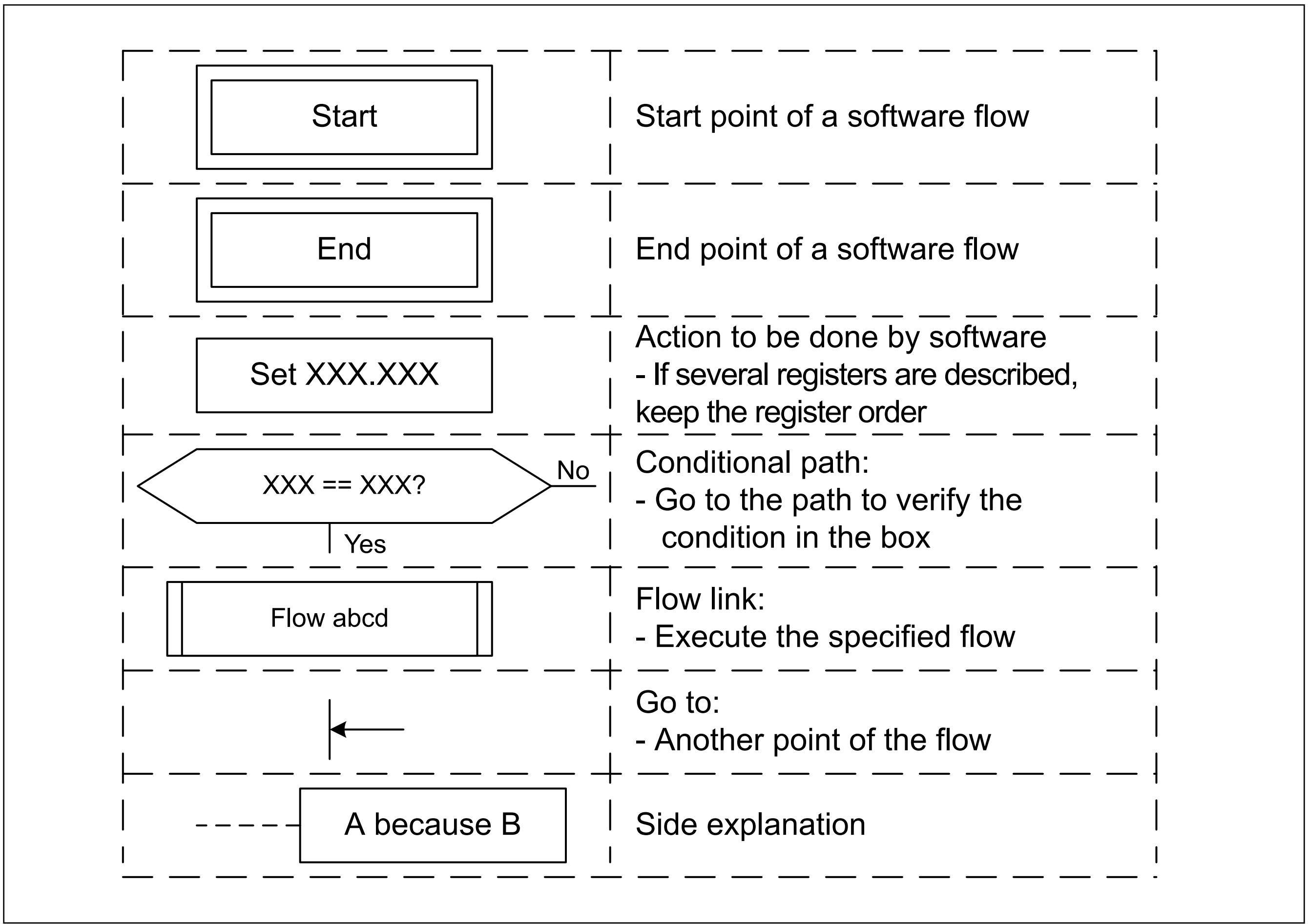


Figure 35.2 Software flow legend

35.4.1.2 Interrupt Handling Flow

Figure 35.3 describes the interrupt handling flow.

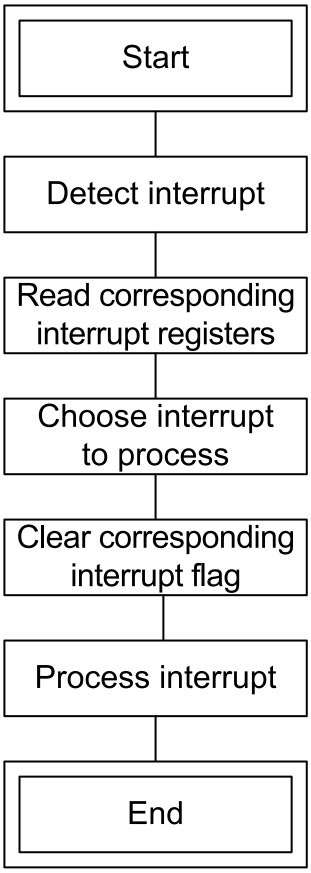


Figure 35.3 Interrupt handling flow

35.4.1.3 Timer t Flows (t = 0, 1)

This section describes the flows to set and monitor timer t.

35.4.1.3.1 Timer t Offset Setting Flow

Figure 35.4 describes the timer t offset setting flow.

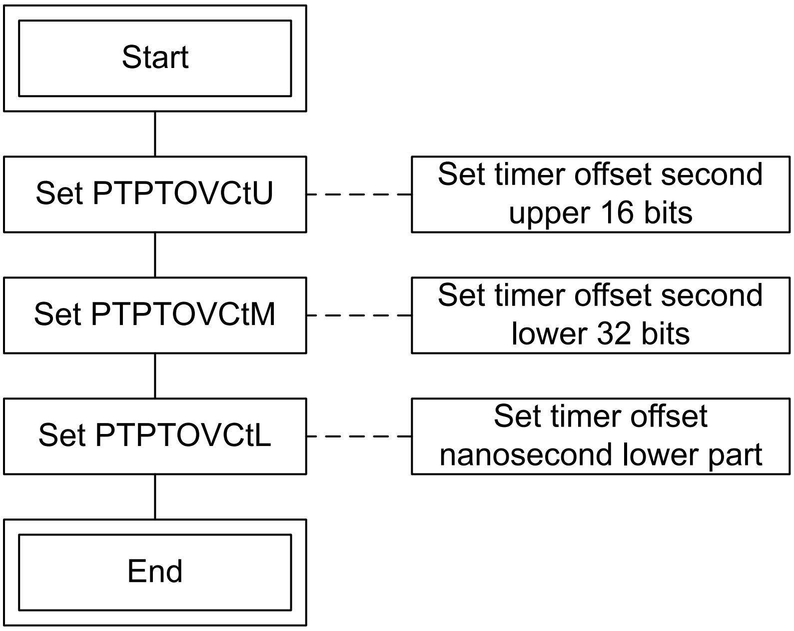


Figure 35.4 Timer t offset setting flow

35.4.1.3.2 32-bit AVTP Timer t Reading Flow

Figure 35.5 describes the 32-bit AVTP timer t reading flow.

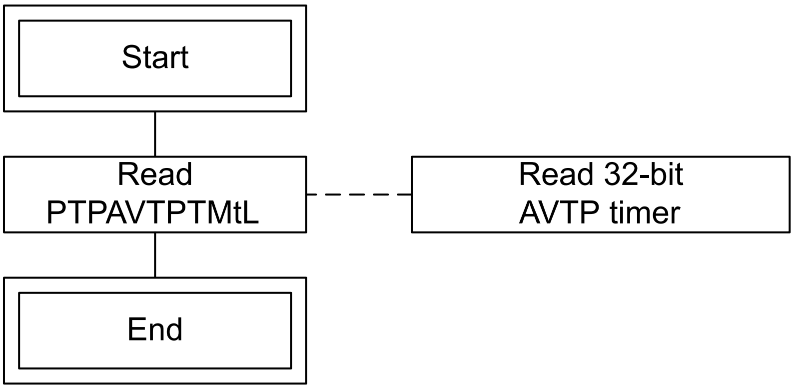


Figure 35.5 32-bit AVTP timer t reading flow

35.4.1.3.3 64-bit AVTP Timer t Reading Flow

Figure 35.6 describes the 64-bit AVTP timer t reading flow.

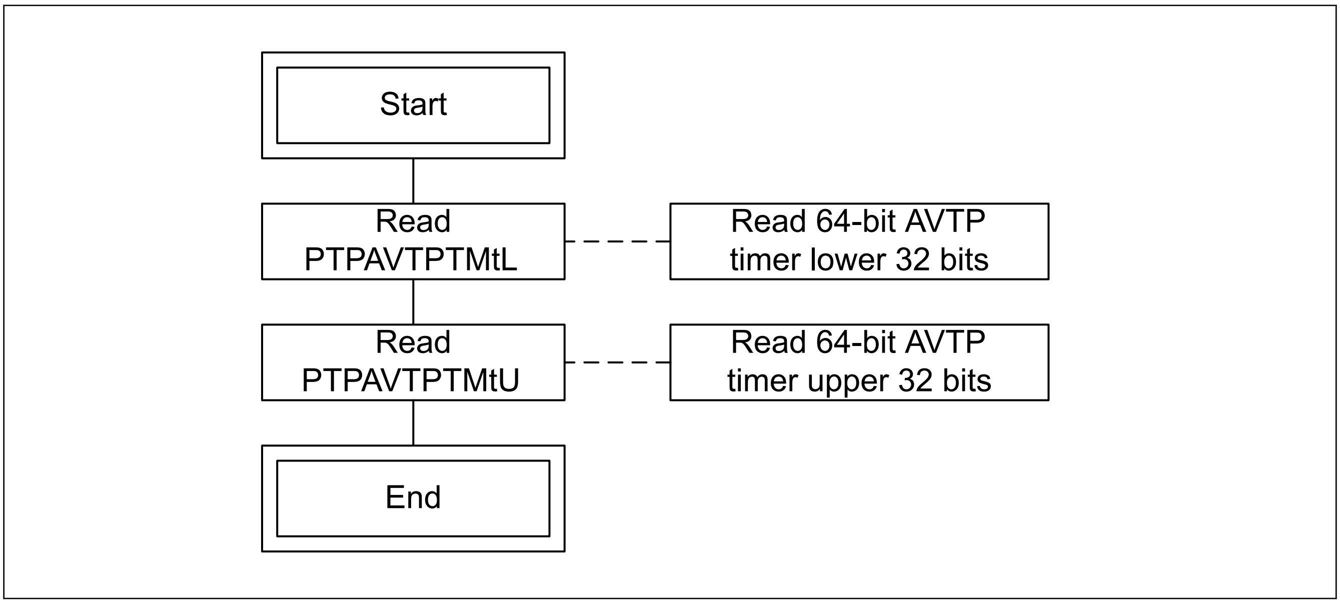


Figure 35.6 64-bit AVTP timer t reading flow

35.4.1.3.4 GPTP Timer t Reading Flow

Figure 35.7 describes the GPTP timer t reading flow.

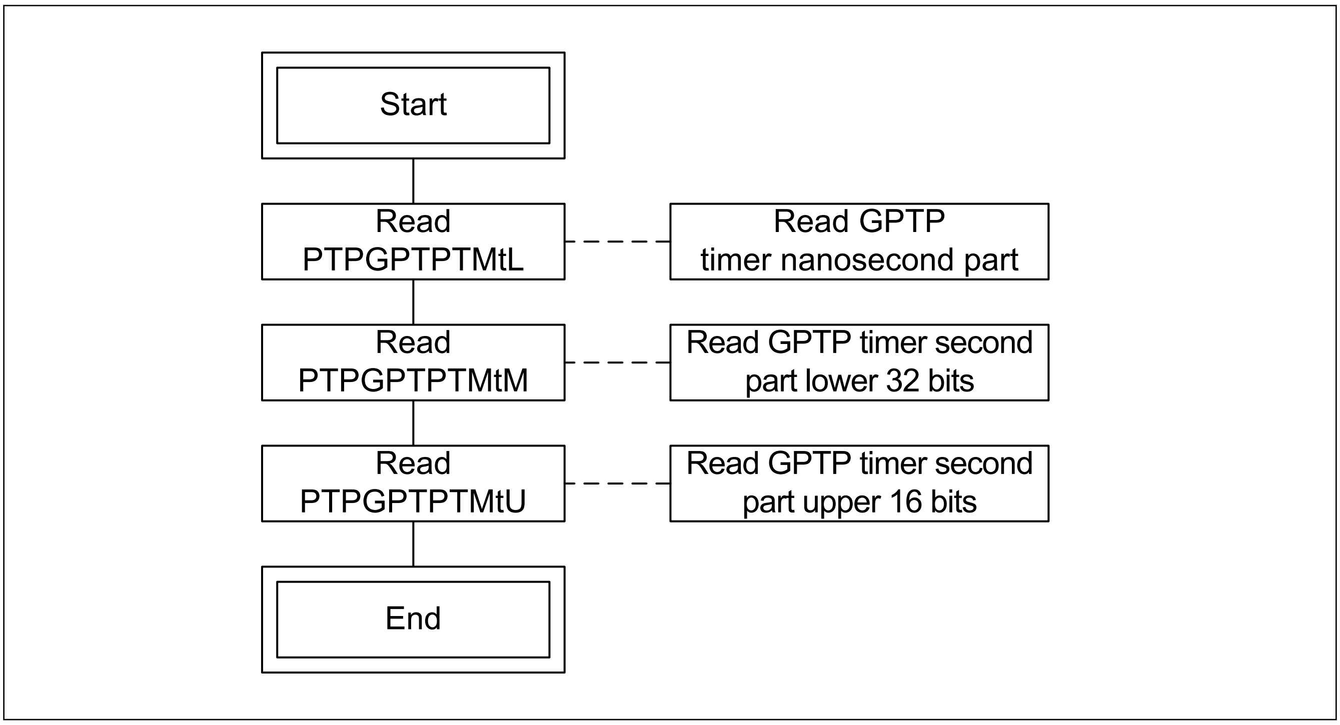


Figure 35.7 GPTP timer t reading flow

35.4.1.3.5 Register Writable Without Software Flow for Timer t

Table 35.4 describes the registers writable without software flow and in which mode they can be written.

Table 35.4 Writable without software flow registers (1 of 2)

(✓ = writable, × = not writable)

Register name	Writable anytime
PTPTMEC	✓

Table 35.4 Writable without software flow registers (2 of 2)

(✓ = writable, × = not writable)

Register name	Writable anytime
PTPTMDC	✓
PTPTIVCt	✓

35.4.1.4 Media Clock m Capture Flows (m = 0, 1)

This section describes the flows to set and monitor media clock m capture.

35.4.1.4.1 32-bit AVTP Timer Captured Value Reading Flow

Figure 35.8 describes the 32-bit AVTP timer captured value reading flow.

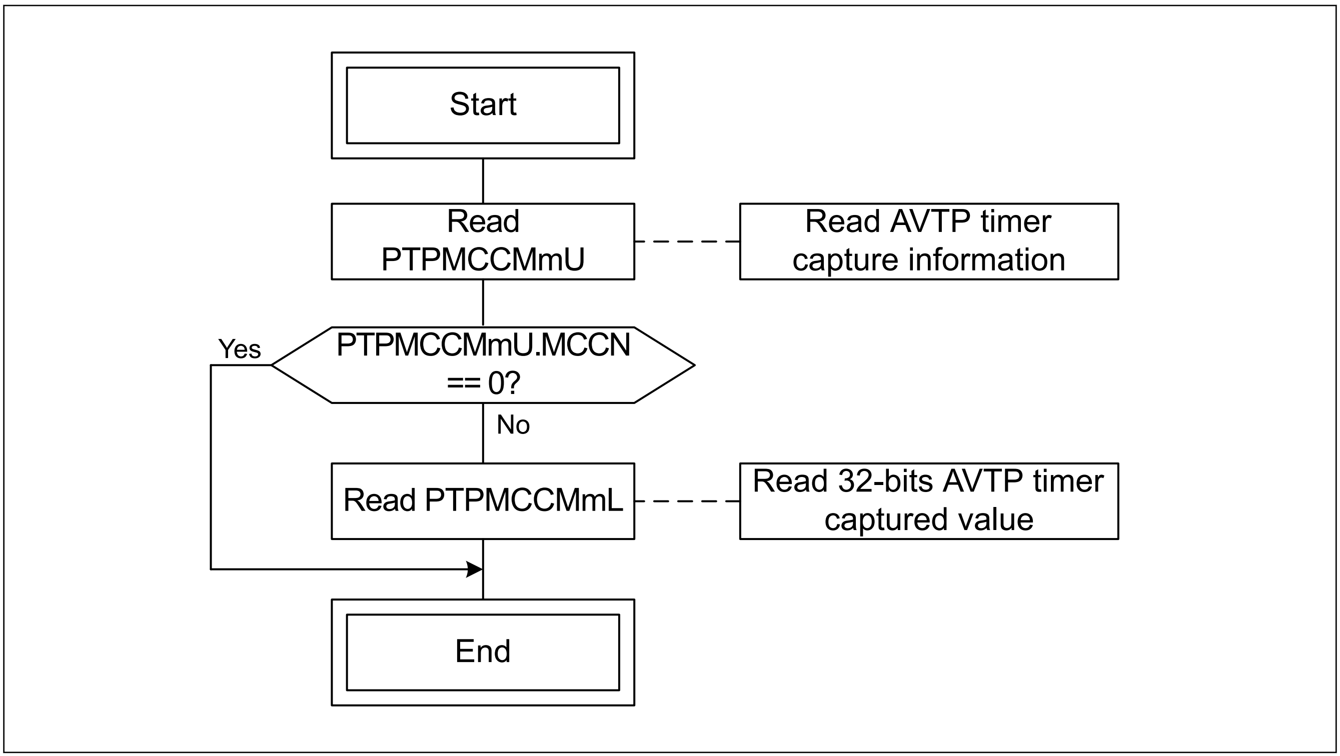


Figure 35.8 32-bit AVTP timer captured value reading flow

35.4.1.4.2 64-bit AVTP Timer Captured Value Reading Flow

Figure 35.9 describes the 64-bit AVTP timer captured value reading flow.

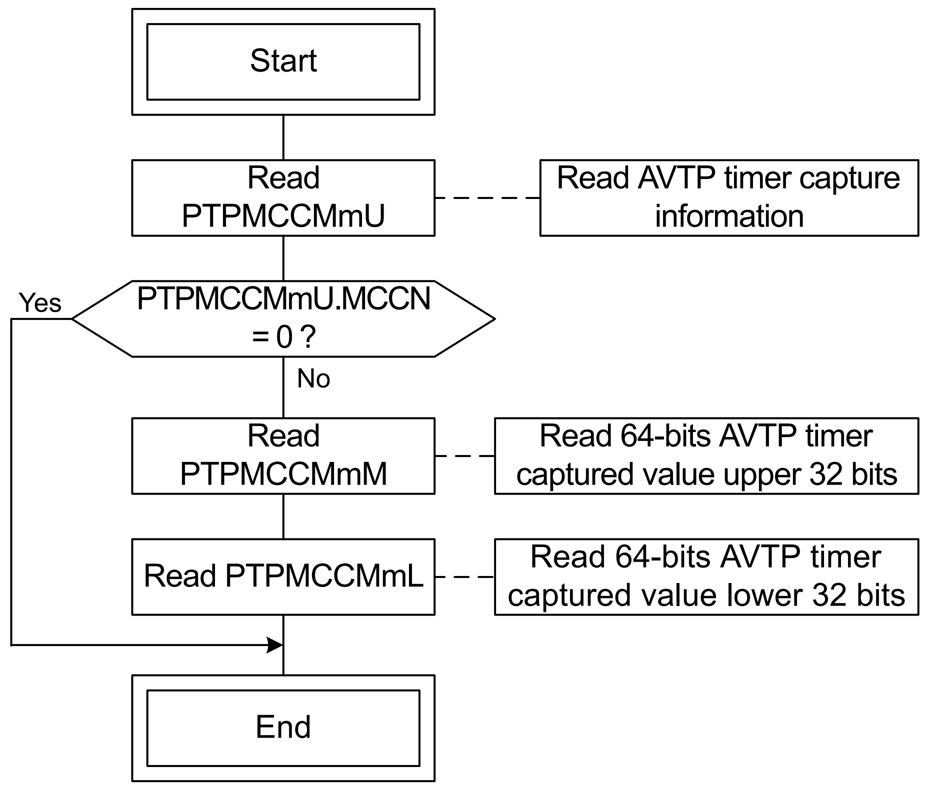


Figure 35.9 64-bit AVTP timer captured value reading flow

35.4.1.4.3 GPTP Timer Captured Value Reading Flow

Figure 35.10 describes the GPTP timer captured value reading flow.

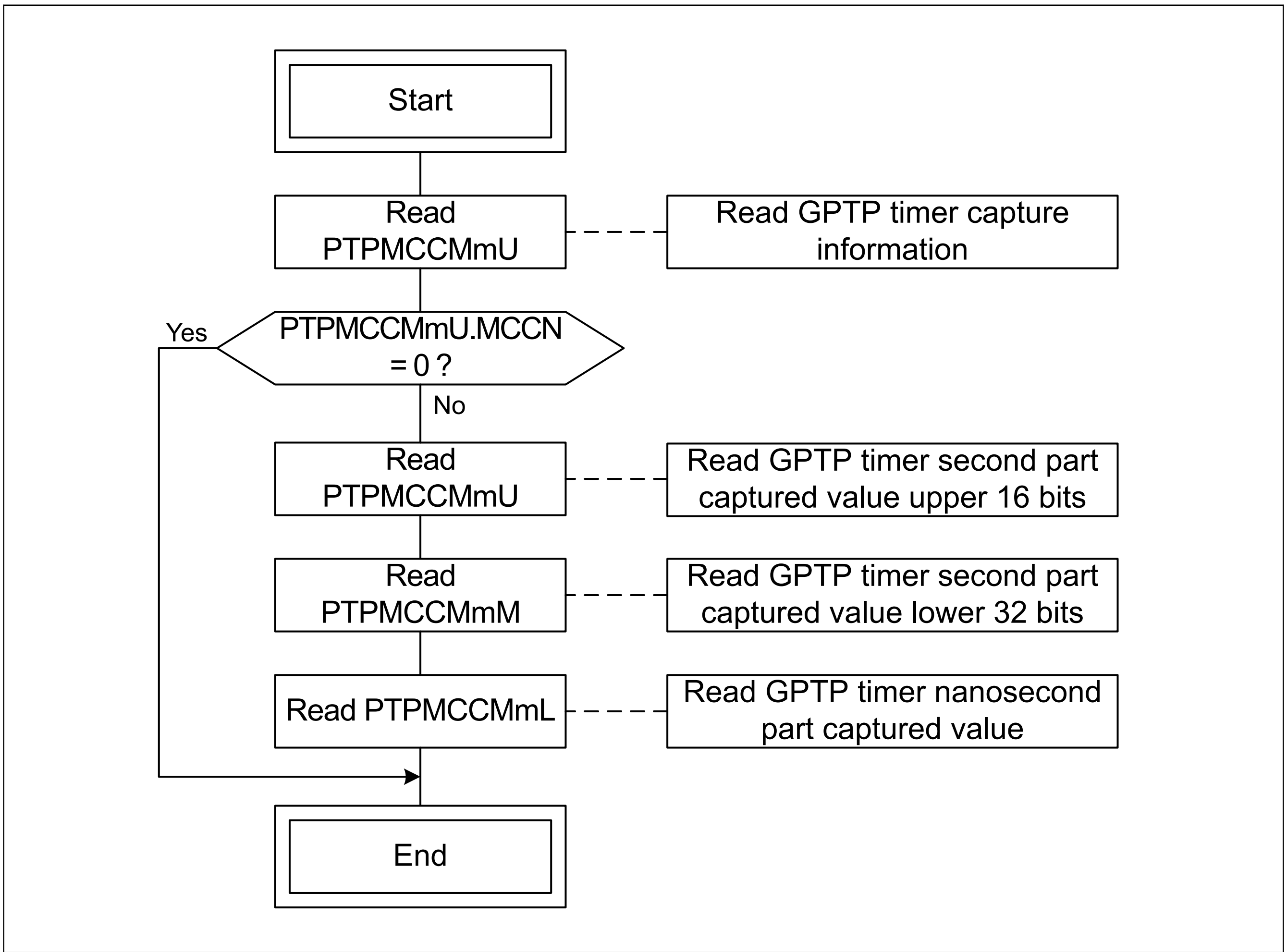


Figure 35.10 GPTP timer captured value reading flow

35.4.1.4.4 Register Writable Without Software Flow for Media Clock m Capture

Table 35.5 describes the registers writable without software flow and in which mode they can be written.

Table 35.5 Writable without software flow registers

(✓ = writable, × = not writable)

Register name	Writable anytime
PTPMCCCm	✓

35.4.1.5 Media Clock m Recovery Flows (m = 0, 1)

This section describes the flows to set and monitor media clock m recovery.

35.4.1.5.1 Media Clock Recovery Time Value Setting Flow

Figure 35.11 describes the media clock recovery time value setting flow.

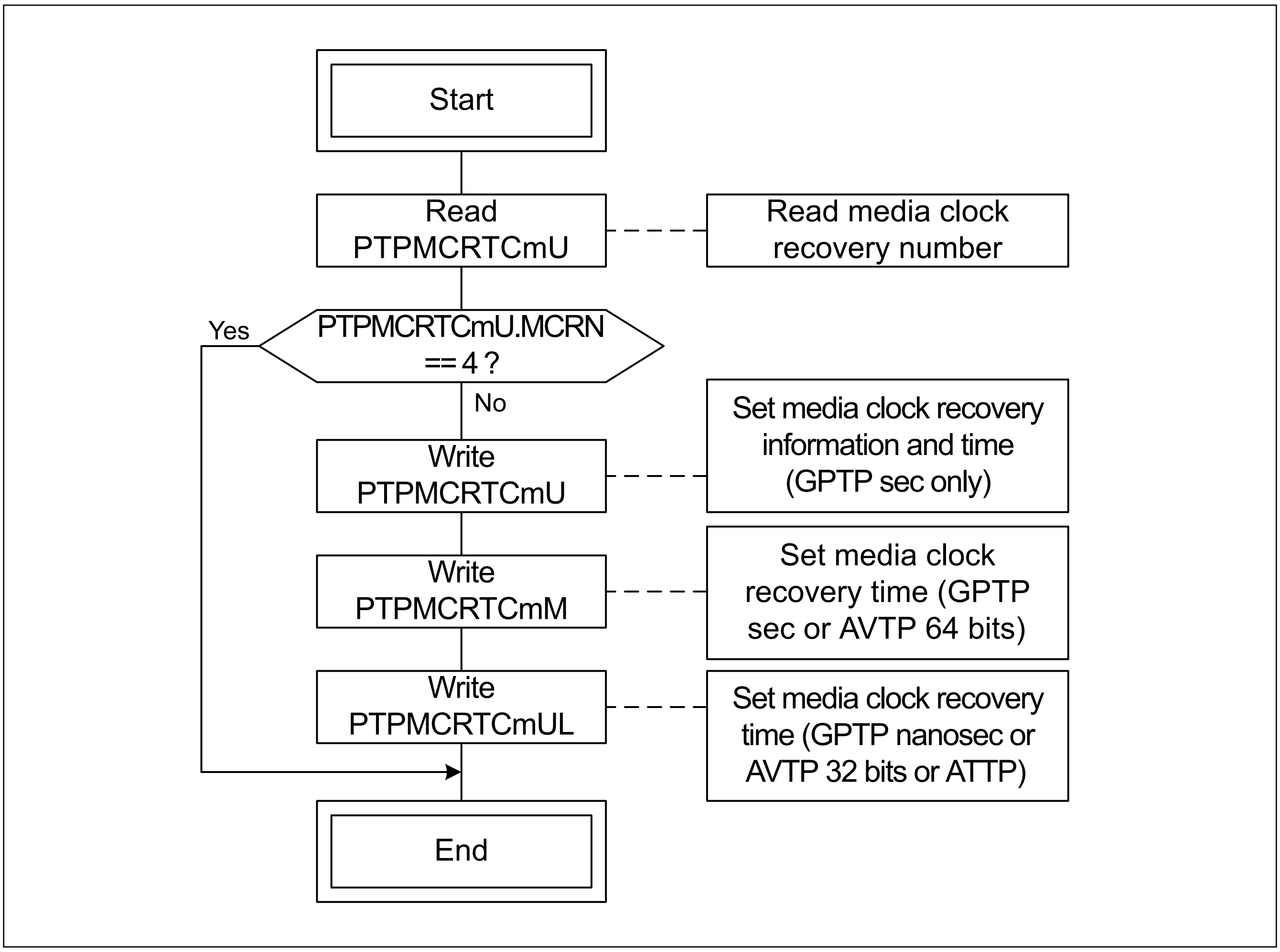


Figure 35.11 Media clock recovery time value setting flow

35.4.1.5.2 Media Clock Recovery Buffer Clear Flow

Figure 35.12 describes the media clock recovery buffer clear flow.

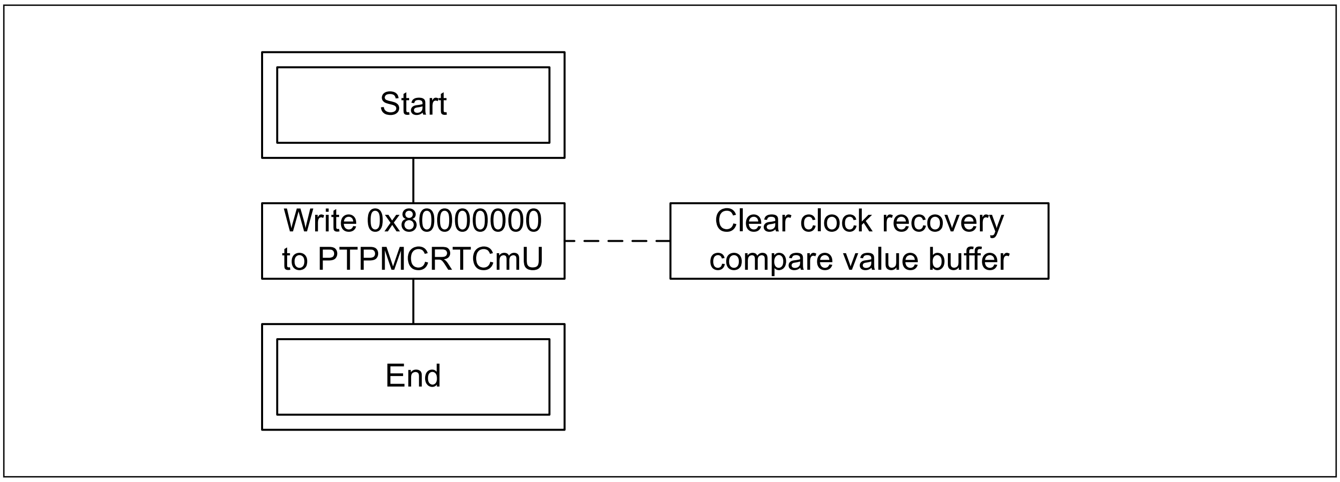


Figure 35.12 Media clock recovery buffer clear flow

35.4.1.5.3 Register Writable Without Software Flow for Media Clock m Recovery

Table 35.6 describes the registers writable without software flow and in which mode they can be written.

Table 35.6 Writable without software flow registers

(✓ = writable, × = not writable)

Register name	Writable anytime
PTPMCRCm	✓

35.4.1.6 Cyclic Compare c Flows (c = 0 to 7)

This section describes the flows to set and monitor cyclic compare c.

35.4.1.6.1 Register Writable Without Software Flow for Cyclic Compare c

Table 35.7 describes the registers writable without software flow and in which mode they can be written.

Table 35.7 Writable without software flow registers

(✓ = writable, × = not writable)

Register name	Writable anytime
PTPCCCc0	✓
PTPCCCc1	✓

35.4.1.7 Register Writable Without Software Flow

This section describes registers that have not been described. These registers can be changed dynamically. However, it is necessary that the initial settings such as the clock enabling have been completed.

35.5 Functional Description

35.5.1 Timer t (t = 0, 1) [gPTP]

Each timer instance includes three timers:

- A GPTP timer
- A 64-bit AVTP timer
- A 32-bit AVTP timer.

35.5.1.1 GPTP Timer t

The GPTP timer t GPTP[t] can be enabled by setting PTPTMEC.TE to 1. When enabled, GPTP timer t automatically calculates its value using equation (1).

(1) gptpTimer[t] = TimerOffset[t] + FreeRunningTimerValue[t]

Where:

- gptpTimer[t] is the calculated timer value GPTP[t] and is divided in two parts. The second part gptpSecond[t] is GPTP[t][77:30] and the nanosecond part gptpNano[t] is GPTP[t][29:0].
- TimerOffset[t] is {PTPTOVCtU.TOVU, PTPTOVCtM.TOVM, PTPTOVCtL.TOVL}
- FreeRunningTimerValue[t] is the value of a free running timer starting from 0 (when timer is disabled) and is incremented by PTPTIVCt.TIV every cycle (the free running timer has a 27-bit sub-nanosecond part not described here so it should be set to clk_period[ns] × 2^{27} where clk_period is clock clk period).

Functions:

- When GPTP timer t is a master, PTPTIVCt.TIV and {PTPTOVCtU.TOVU, PTPTOVCtM.TOVM, PTPTOVCtL.TOVL} registers can be kept to fixed values.
- When GPTP timer t is a slave, PTPTIVCt.TIV register can be used in priority for time resynchronization. If GPTP timer t exceeds the corresponding master value, {PTPTOVCtU.TOVU, PTPTOVCtM.TOVM, PTPTOVCtL.TOVL} registers can be used for time resynchronization.

35.5.1.2 64-bit AVTP Timer t

The 64-bit AVTP timer t AVTP64[t] is used as a reference to synchronize any necessary media clocks and to determine when the payload of a stream is to be presented to the time-sensitive application. Media clocks vary with audio/video types therefore the exact usage of the AVTP presentation time is media-format dependent [1722].

The 64-bit AVTP timer t is generated from the GPTP timer t and can be calculated using equation (2).

(2) $avtpTimer64[t] = (gptpSecond[t] \times 10^9 + gptpNano[t]) \bmod 2^{64}$

Where:

- avtpTimer64[t] is the calculated timer value AVTP64[t]
- gptpSecond[t] is gptpTimer[t] second part GPTP[t][77:30]
- gptpNano[t] is gptpTimer[t] nanosecond part GPTP[t][29:0].

35.5.1.3 32-bit AVTP Timer t

The 32-bit AVTP timer t AVTP32[t] is the 32-bit version of 64-bit AVTP timer t and can be calculated using equation (3).

(3) $avtpTimer32[t] = (gptpSecond[t] \times 10^9 + gptpNano[t]) \bmod 2^{32}$

Where:

- avtpTimer32[t] is the calculated timer value AVTP32[t] (AVTP64[t][31:0])
- gptpSecond[t] is gptpTimer[t] second part GPTP[t][77:30]
- gptpNano[t] is gptpTimer[t] nanosecond part GPTP[t][29:0].

35.5.1.4 PPS Signal t

The PPS[t] signal is used to check if GPTP synchronization is successful. As per GPTP standard [PTP], PPS[t] signal should be output as pins at the chip level to allow synchronization checking on oscilloscope. PPS[t] is the PPS signal corresponding to timer number t. For PPS signal details, see the GPTP standard [gPTP].

35.5.2 Media Clock m Capture (m = 0, 1)

Media clock m capture aims at capturing timer values on MEDIA_IN[m] positive edge and/or negative edge.

Functions:

- PTPMCCCm.MCPEE is used to enable timer value capture on MEDIA_IN[m] positive edge
- PTPMCCCm.MCNEE is used to enable timer value capture on MEDIA_IN[m] negative edge
- PTPMCCCm.MCTNS is used to select timer number for timer value capture
- PTPMCCCm.MCTTS is used to select between GPTP and AVTP timer for timer value capture
- PTPMCCMmL/mM/mU are used to read the captured timer values
- PTPIS0.MCCSn interrupt is used to signal to the CPU that a timer value has been captured
- PTPIS0.MCCOESn interrupt is used to signal to the CPU that a captured timer value has been lost because the timer capture value FIFO was full.

35.5.3 Media Clock m Recovery (m = 0, 1)

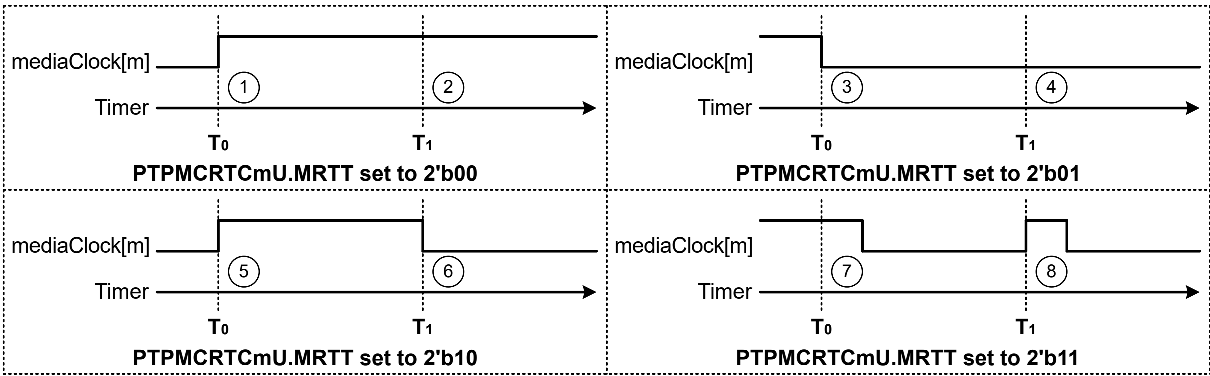
Media clock m recovery aims at creating an internal clock mediaClock[m] which can be mapped to an output pin, (see section 35.5.4. Media Clock m Mapping to Output Pins).

Functions:

- PTPMCRCm.MRTNS is used to select timer number for clock recovery
- PTPMCRCm.MRTTS is used to select between GPTP and AVTP timers for clock recovery
- PTPMCRCm.MRAMS is used to select between 32-bit AVTP timer and 64-bit AVTP timer for clock recovery when PTPMCRCm.MRTTS is set to AVTP timer
- PTPMCRCmL/mM/mU are used to add a timer value and command to execute to mediaClock[m]

- PTPIS0.MCRMS[m] interrupt is used to signal to the CPU that the oldest timer value (which has not matched yet) set in PTPMCRCmL/mM/mU matched the selected timer therefore mediaClock[m] value has been updated.

Figure 35.13 describes mediaClock[m] behavior when a timer value matches depending on the timer value type setting and the initial value of mediaClock[m].



1	Compare entry set with **PTPMCRTCmU.MRTT** set to 2'b00 matched at Timer T_0 when mediaClock[m] is not set. mediaClock[m] gets set.
2	Compare entry set with **PTPMCRTCmU.MRTT** set to 2'b00 matched at Timer T_1 when mediaClock[m] is set. mediaClock[m] stays set.
3	Compare entry set with **PTPMCRTCmU.MRTT** set to 2'b01 matched at Timer T_0 when mediaClock[m] is set. mediaClock[m] gets cleared.
4	Compare entry set with **PTPMCRTCmU.MRTT** set to 2'b01 matched at Timer T_1 when mediaClock[m] is not set. mediaClock[m] stays cleared.
5	Compare entry set with **PTPMCRTCmU.MRTT** set to 2'b10 matched at Timer T_0 when mediaClock[m] is not set. mediaClock[m] gets inverted.
6	Compare entry set with **PTPMCRTCmU.MRTT** set to 2'b10 matched at Timer T_0 when mediaClock[m] is set. mediaClock[m] gets inverted.
7	Compare entry set with **PTPMCRTCmU.MRTT** set to 2'b11 matched at Timer T0 when mediaClock[m] is set. mediaClock[m] stays set for **PTPMCRCm.MRPL**+1 clock cycles.
8	Compare entry set with **PTPMCRTCmU.MRTT** set to 2'b11 matched at Timer T0 when mediaClock[m] is not set. mediaClock[m] gets set for **PTPMCRCm.MRPL**+1 clock cycles.

Figure 35.13 mediaClock[m] behavior

Note: When timer matched an entry with PTPMCRTCmU.MRTT set to 2'b11 and the next match occurs before mediaClock[m] deassertion, mediaClock[m] deassertion command is overwritten and does not occur.

35.5.4 Media Clock m Mapping to Output Pins

The mediaClock[m] signal created by media clock m recovery (section 35.5.3. Media Clock m Recovery (m = 0, 1)) can be output to be used inside the chip or as a chip pin.

Functions:

- By setting PTPMCPCm.PE to 1 and PTPMCPCm.MRS to m, mediaClock[m] signal is assigned to MEDIA_OUT[p] output pin.

35.5.5 Cyclic Compare c (c = 0 to 7)

Cyclic compare c aims at creating a pulse cyclically on CYCLIC_COMP[i][c] for i = 0, 1 interface to trigger an event. It can be used, for example, to trigger a periodic data transmission for GPTP synchronization (network control frame transmission).

Functions:

- PTPCCCc0.CCTNS is used to select timer number for cyclic compare. During cyclic compare, timer offset is ignored, and only incremental value is taken in account.
- PTPCCCc1.CCV is used to enable cycle compare. When cycle compare c is enabled, a pulse is sent every PTPCCCc1.CCV to CYCLIC_COMP[PTPCCCc0.CCOPS][c].
- When this feature is enabled (when the PTPCCCc1.CCV configuration is updated to a value greater than 32), the first pulse from comparator c is output immediately (after one system clock), regardless of PTPCCCc1.CCV configuration.

35.6 Pulse Output Timer Module (plsotim)

35.6.1 Overview of plsotim

The pulse output timer module (plsotim), a submodule of gPTP, produces a pulsed output according to the base period of a gPTP timer.

Table 35.8 lists the plsotim specifications.

Table 35.8 plsotim specifications

Item	Function	Description
Overall function	Pulse generator	• Pulse generation timer, pulsed output, and interrupt output.
Data provision		• Pulsed output channel: 4 ch The pulse cycle: 80 bits ([79:32] = second and [31:0] = ns) Pulse width: 20 bits ([19:0] = counter) • Interruption output channel: 4 ch.
Functions	PTP timer select	• The PTP timer to be used as the reference time can be selected.
	PTP_PULSE_GEN	• Uses the reference time of the GPTP timer to generate pulse outputs of any period and any pulse width.
	Interrupt pulse output	• The rising edge of the pulse output is detected, and the interrupt pulse is output.

35.6.2 Plsotim Register List

Table 35.9 Plsotim registers

Offset address	Register name	Register symbol	Write access mode
0x1000	Pulse Output Timer Configuration Register	POTCFGR	—
0x1004 + 0x30 × n	Pulse Output Timer Control Register n (n = 0 to 3)	POTCRn	—
0x1008 + 0x30 × n	Pulse Output Start Time Setting Register nU (n = 0 to 3)	POTSTRnU	—
0x100C + 0x30 × n	Pulse Output Start Time Setting Register nM (n = 0 to 3)	POTSTRnM	—
0x1010 + 0x30 × n	Pulse Output Start Time Setting Register nL (n = 0 to 3)	POTSTRnL	—
0x1014 + 0x30 × n	Period Setting Register nU (n = 0 to 3)	POTPERnU	—
0x1018 + 0x30 × n	Period Setting Register nM (n = 0 to 3)	POTPERnM	—
0x101C + 0x30 × n	Period Setting Register nL (n = 0 to 3)	POTPERnL	—
0x1020 + 0x30 × n	Pulse Width Setting Register n (n = 0 to 3)	POTPWRn	—
0x1028 + 0x30 × n	Time Capture Register nU (n = 0 to 3)	POTCPRnU	—
0x102C + 0x30 × n	Time Capture Register nM (n = 0 to 3)	POTCPRnM	—
0x1030 + 0x30 × n	Time Capture Register nL (n = 0 to 3)	POTCPRnL	—

Note: Base address: GPTP = 0x403E_0000, GPTP_NS = 0x503E_0000

35.6.3 PIsotim Register Descriptions

35.6.3.1 POTCFGR : Pulse Output Timer Configuration Register

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x1000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	REFSEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	REFSEL	Reference Timer Select 0: Select and use the reference time of GPTP timer 0 1: Select and use the reference time of GPTP timer 1	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

35.6.3.2 POTCRn : Pulse Output Timer Control Register n (n = 0 to 3)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x1004 + 0x30 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	START
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	START	Pulse Output Enable Pulse generation is performed by taking in the set value of the pulse generation register when this register becomes 1. If set to 0, no pulse is output. By writing 0 during pulse output, pulse output stops (0 output). 0: Pulse output stop 1: Pulse output permission	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

35.6.3.3 POTSTRnU : Pulse Output Start Time Setting Register nU (n = 0 to 3)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x1008 + 0x30 × n

Bit position:	31																15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	n/a	Seconds (upper 16 bits) of pulse output start time. Functions: ● Sets the lower 16 bits second of the time to start pulse output ● For comparison with the reference time of the GPTP timer. Valid value ● The register value to generate the pulse output uses the value when the POTCRn.START bit changes from 0 to 1 ● Even if the register value is changed after the POTCRn.START bit becomes 1, it is not reflected in the pulse generation.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

35.6.3.4 POTSTRnM : Pulse Output Start Time Setting Register nM (n = 0 to 3)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x100C + 0x30 × n

Bit position:	31																															0
Bit field:																																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Seconds (upper 32 bits) of pulse output start time. Functions: ● Sets the lower 32 bits second of the time to start pulse output ● For comparison with the reference time of the GPTP timer. Valid value ● The register value to generate the pulse output uses the value when the POTCRn.START bit changes from 0 to 1 ● Even if the register value is changed after the POTCRn.START bit becomes 1, it is not reflected in the pulse generation.	R/W

35.6.3.5 POTSTRnL : Pulse Output Start Time Setting Register nL (n = 0 to 3)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x1010 + 0x30 × n

Bit position:	31		29																													0
Bit field:	—	—																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
29:0	n/a	Nanoseconds (29 bits) of pulse output start time. Functions: ● Sets the nanosecond of the time to start pulse output ● For comparison with the reference time of the GPTP timer. Valid value ● The register value to generate the pulse output uses the value when the POTCRn.START bit changes from 0 to 1 ● Even if the register value is changed after the POTCRn.START bit becomes 1, it is not reflected in the pulse generation. Restrictions: ● Set the value between 0 and 0x3B9AC9FF. 0x3B9ACA00 is 1 second, therefore set using the Second register.	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

35.6.3.6 POTPERnU : Period Setting Register nU (n = 0 to 3)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x1014 + 0x30 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	n/a	Seconds (upper 16 bits) of pulse output period. Functions: ● Sets the lower 16 bit seconds of the time to start pulse output ● Calculates the reference time of the start time to be compared next. Valid value ● The register value to generate the pulse output uses the value when the POTCRn.START bit changes from 0 to 1 ● Even if the register value is changed after the POTCRn.START bit becomes 1, it is not reflected in the pulse generation.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

35.6.3.7 POTPERnM : Period Setting Register nM (n = 0 to 3)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x1018 + 0x30 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Seconds (upper 32 bits) of pulse output period. Functions: ● Sets the lower 32-bit seconds period of the pulse output ● Calculates the reference time of the start time to be compared next. Valid value: ● The register value to generate the pulse output uses the value when the POTCRn.START bit changes from 0 to 1 ● Even if the register value is changed after the POTCRn.START bit becomes 1, it is not reflected in the pulse generation.	R/W

35.6.3.8 POTPERnL : Period Setting Register nL (n = 0 to 3)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x101C + 0x30 × n

Bit position:	31	30	29–0
Bit field:	—	—	
Value after reset:	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
29:0	n/a	Nanoseconds (30 bits) of pulse output period. Functions: ● Sets the nanosecond of the time-to-period pulse output ● Calculates the reference time of the start time to be compared next. Valid value: ● The register value to generate the pulse output uses the value when the POTCRn.START bit changes from 0 to 1 ● Even if the register value is changed after the POTCRn.START bit becomes 1, it is not reflected in the pulse generation ● When reading, the written value can be read. Restrictions: ● Set the value between 0 and 0x3B9AC9FF. 0x3B9ACA00 is 1 second, therefore set using the Second register.	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

35.6.3.9 POTPWRn : Pulse Width Setting Register n (n = 0 to 3)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x1020 + 0x30 × n

Bit position:	31–16	15–0
Bit field:	— — — — — — — — — — — — — — — —	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	n/a	High Pulse Width Count Number for the Pulse Output. Functions: ● Sets the number of counts for the high pulse width ● Counting begins from the start of pulse output ● The high pulse width time varies depending on the clock period. Valid value: ● The register value to generate the pulse output uses the value when the POTCRn.START bit changes from 0 to 1 ● Even if the register value is changed after the POTCRn.START bit becomes 1, it is not reflected in the pulse generation.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

35.6.3.10 POTCPRnU : Time Capture Register nU (n = 0 to 3)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x1028 + 0x30 × n

Bit position: 31 15 0

Bit field: — — — — — — — — — — — — — — — —

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	n/a	The upper 16-bit value of seconds for the pulse output start time.	R
31:16	—	These bits are read as 0.	R

35.6.3.11 POTCPRnM : Time Capture Register nM (n = 0 to 3)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x102C + 0x30 × n

Bit position: 31 0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	The lower 32-bit value of seconds for the pulse output start time.	R

35.6.3.12 POTCPRnL : Time Capture Register nL (n = 0 to 3)

Base address: GPTP = 0x403E_0000
GPTP_NS = 0x503E_0000

Offset address: 0x1030 + 0x30 × n

Bit position: 31 29 0

Bit field: — —

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
29:0	n/a	The nanosecond value for the pulse output start time.	R
31:30	—	These bits are read as 0.	R

35.6.4 Operation of plsotim

35.6.4.1 Pulse Output Timer Operation Settings

Figure 35.14 shows the operation setting flow of the pulse output timer n.

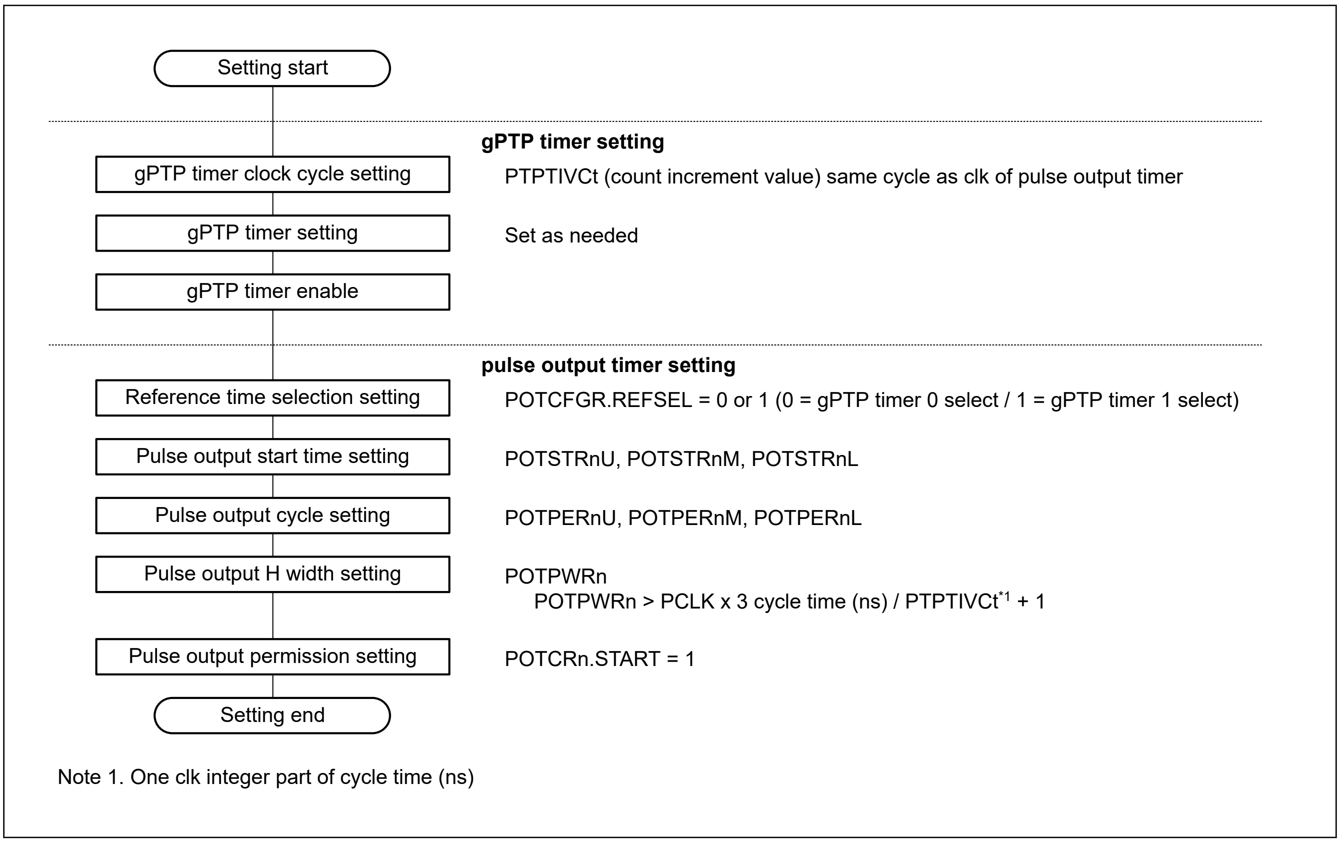


Figure 35.14 Operation setting flow of the pulse output timer

35.6.4.2 Pulse Output Timer n (n = 0 to 3)

This module uses the reference time of the GPTP timer to generate a pulse output of any period and any pulse width.

35.6.4.2.1 GPTP Timer Select

Select the GPTP timer of the reference time to be used by setting the POTCFGR register.

POTCFGR.REFSEL bit setting:

0 : Generates a pulse output using the reference time of GPTP timer 0.

1 : Generates a pulse output using the reference time of GPTP timer 1.

Functions:

- The reference time used for pulse generation can be selected by setting the POTCFGR register
- When POTCFGR.REFSEL = 0: Generates a pulse using the time input signal of GPTP timer 0
- When POTCFGR.RFGSEL = 1: Generate a pulse using the time input signal of GPTP timer 1
- When changing the POTCFGR register, stop the pulse output (POTCRn.START = 0) before modification.

35.6.4.2.2 Pulse Output Generation Register

By setting the following registers, it is possible to generate a pulse with an arbitrary start time, an arbitrary period, and an arbitrary H pulse width.

- Pulse output start time register: POTSTRnU, POTSTRnM, POTSTRnL
- Pulse output cycle register: POTPERnU, POTPERnM, POTPERnL
- Pulse output H width register: POTPWRn.

When the H width register of the pulse output is 0, the pulse is not generated, so the L output is fixed.

If the pulse output cycle is set to be less than or equal to the H width time of the pulse output, the H output is fixed.

35.6.4.2.3 Pulse Output Generation

By changing the POTCRn.START bit from 0 to 1, the value of the pulse output generation register is captured and held.

Pulse is generated according to the set value of the held pulse output generation.

The output pulse is as follows.

Rise of the first pulse:

Pulse output H width register ≠ 0: It becomes H at the set start time.

Pulse output H width register = 0: L output remains (no pulse output).

Pulse fall:

Pulse output cycle register value > Pulse output H width register value × clk cycle.

When the clk number is counted by the amount of the H width register value after the output becomes H, it becomes L.

Pulse output cycle register value ≤ H width register value of pulse output × clk cycle.

The output remains H (H output is fixed until the POTCRn.START bit is set to 0).

Rise from the second time onward:

When H width register = other than 0 and pulse output cycle > H width register value × clk cycle.

It becomes H at the time of the previous start time + pulse output cycle register value.

35.6.4.2.4 Pulse Output Stop

Pulse output can be stopped by setting the POTCRn.START bit to 0.

When stopped, the output is fixed at L.

When the POTCRn.START bit is set to 1 again, the register setting value is taken in and held, and a new pulse is generated and output.

Note: Even if the pulse output is stopped in the middle, it is not restarted. A new pulse is generated and output.

35.6.4.2.5 Timing of Pulse Output Permission

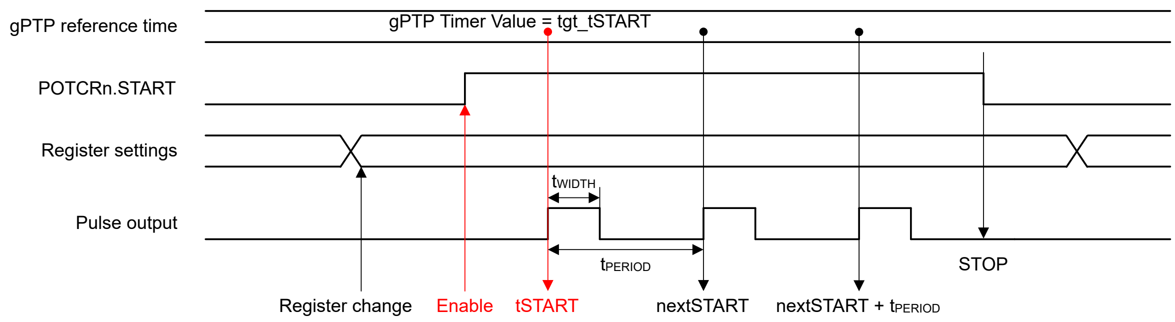
If the POTCRn.START bit is changed from 0 to 1 after the set pulse output start time, no pulse output is performed.

The pulse is output from the time when the reference time goes around once and the start time is reached again.

35.6.4.3 Pulse Output Timing Diagrams

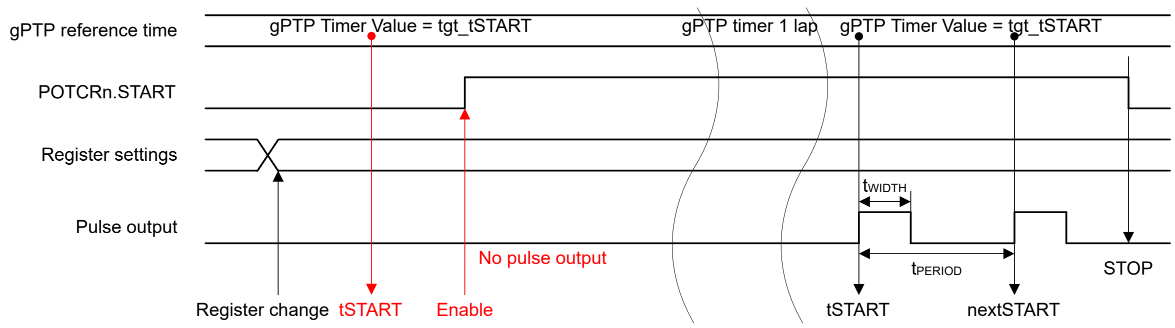
Six examples of pulse outputs are shown in the timing diagrams as following:

1. Figure 35.15 When changing the POTCRn.START bit from 0 to 1 after setting the register and before the start time
2. Figure 35.16 When changing the POTCRn.START bit from 0 to 1 after setting the register and after the start time
3. Figure 35.17 Pulse output H width register ≥ when the POTCRn.START bit is changed from 0 to 1 by setting the pulse output cycle
4. Figure 35.18 When changing the POTCRn.START bit from 0 to 1 with pulse output H width register = 0
5. Figure 35.19 When the pulse output is stopped with the POTCRn.START bit set to 0
6. Figure 35.20 When the register setting is changed with the POTCRn.START bit set to 1.



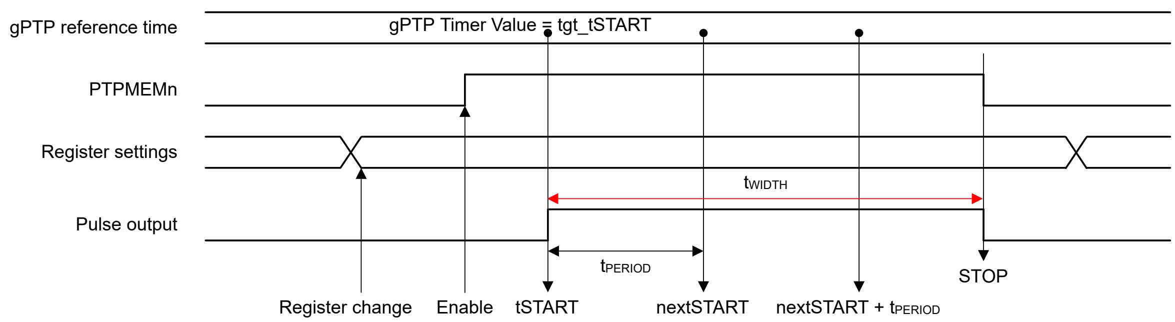
Note: tgt_tSTART = First time : {POTSTRnU, POTSTRnM, POTSTRnL}, From the time onwards (from nextSTART) : {tgt_tSTART + t_{PERIOD}}
t_{PERIOD} = {POTPERnU, POTPERnM, POTPERnL}
t_{WIDTH} = {POTPWRn}

Figure 35.15 When changing the POTCRn.START bit from 0 to 1 after setting the register and before the start time



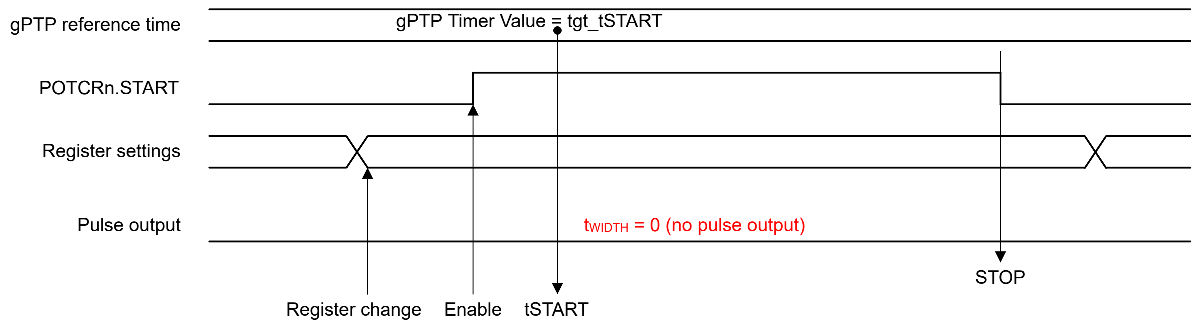
Note: tgt_tSTART = First time : {POTSTRnU, POTSTRnM, POTSTRnL}, From the time onwards (from nextSTART) : {tgt_tSTART + t_{PERIOD}}
t_{PERIOD} = {POTPERnU, POTPERnM, POTPERnL}
t_{WIDTH} = {POTPWRn}

Figure 35.16 When changing the POTCRn.START bit from 0 to 1 after setting the register and after the start time



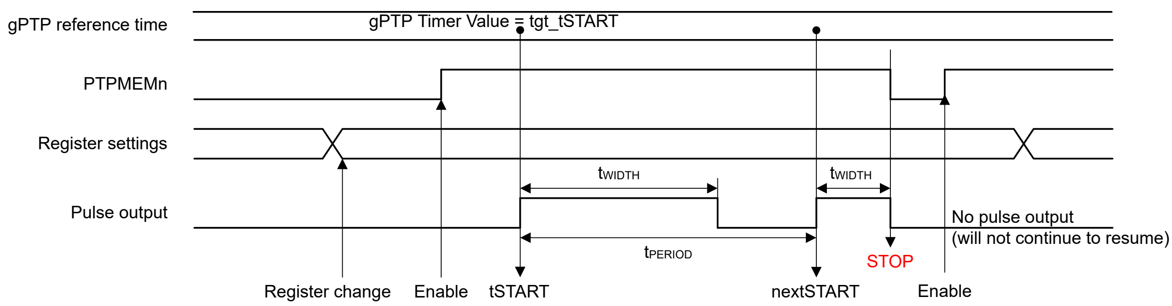
Note: tgt_tSTART = First time : {POTSTRnU, POTSTRnM, POTSTRnL}, From the time onwards (from nextSTART) : {tgt_tSTART + t_{PERIOD}}
t_{PERIOD} = {POTPERnU, POTPERnM, POTPERnL}
t_{WIDTH} = {POTPWRn}

Figure 35.17 Pulse output H width register ≥ when the POTCRn.START bit is changed from 0 to 1 by setting the pulse output cycle



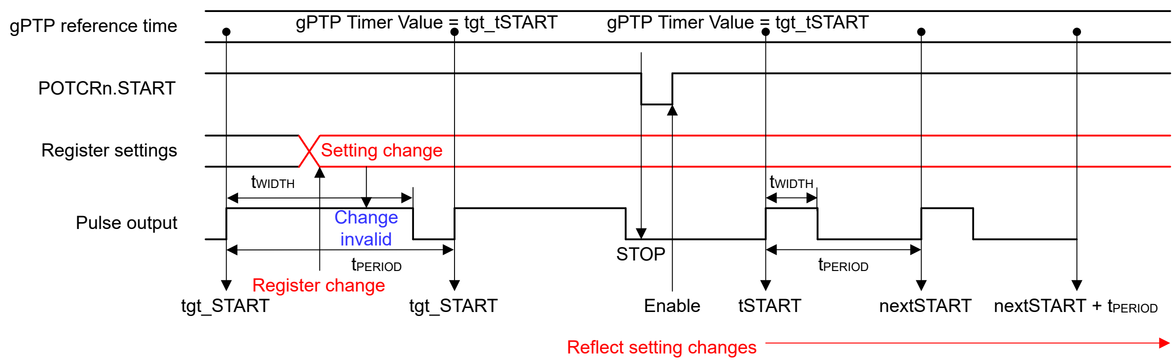
Note: tSTART = {POTSTRnU, POTSTRnM, POTSTRnL} = anything
t_{PERIOD} = {POTPERnU, POTPERnM, POTPERnL} = anything
t_{WIDTH} = {POTPWRn} = 0
nextSTART = Cannot calculate

Figure 35.18 When changing the POTCRn.START bit from 0 to 1 with pulse output H width register = 0



Note: tgt_tSTART = First time : {POTSTRnU, POTSTRnM, POTSTRnL}, From the time onwards (from nextSTART) : {tgt_tSTART + t_{PERIOD}}
t_{PERIOD} = {POTPERnU, POTPERnM, POTPERnL}
t_{WIDTH} = {POTPWRn}

Figure 35.19 When the pulse output is stopped with the POTCRn.START bit set to 0



Note: tgt_tSTART = First time : {POTSTRnU, POTSTRnM, POTSTRnL}, From the time onwards (from nextSTART) : {tgt_tSTART + t_{PERIOD}}
t_{PERIOD} = {POTPERnU, POTPERnM, POTPERnL}
t_{WIDTH} = {POTPWRn}

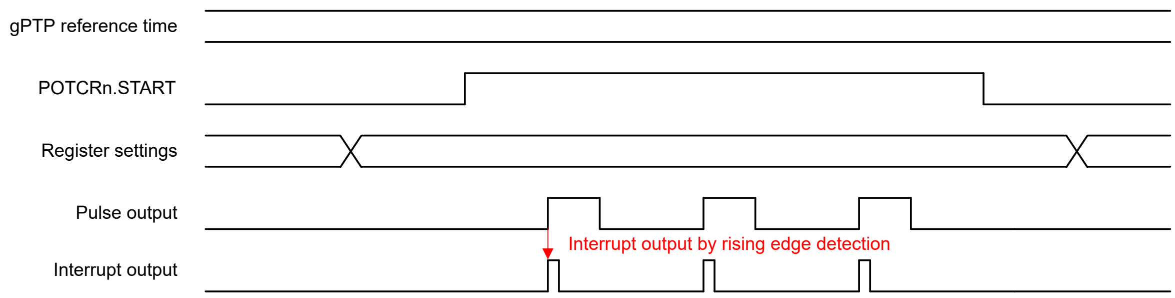
Figure 35.20 When the register setting is changed with the POTCRn.START bit set to 1

35.6.5 Interrupt Output

Interrupt output detects the rising edge of the pulse output and outputs an interrupt signal synchronized with PCLK.

There are four channels of interrupt output, and the interrupt signal corresponding to each pulse output channel is output. The interrupt signal outputs H with a width of 1 cycle of PCLK.

Figure 35.21 shows the timing diagram of interrupt output.



Note: There is no register to control the interrupt output, and one pulse is output in PCLK synchronization after the rising edge of the pulse output is detected.
The interrupt signal is output by generating one pulse after synchronizing the pulse output with PCLK.
It is necessary to set POTPWRn so that the H width of the pulse output is 3 cycles or more with the PCLK required for synchronization.

Figure 35.21 Detects the rising edge of the pulse output and outputs one pulse synchronized with PCLK

35.6.6 Precautions

35.6.6.1 Register Setting

The POTCFGR register switches the GPTP timer selection immediately when the setting is changed regardless of whether it is stopped or operating.

Be careful when changing setting during operation.

35.6.7 Restrictions (Including Known Problems)

35.6.7.1 Register Setting

The following registers should be set within the limit range of the set value:

- POTSTRnL, POTPERnL: Set the value in the range of 0 - 0x3B9AC9FF.
  0x3B9ACA00 is 1 second, so if it is 1 second or longer, reflect and set it in the register in seconds.
- POTPWRn: For settings other than 0, set for a time of 3 or more cycles of the PCLK cycle.
  The time required to generate the interrupt output signal.

36. USB 2.0 Full-Speed Module (USBFS)

36.1 Overview

The USB 2.0 Full-Speed module (USBFS) operates as a host or device controller compliant with the Universal Serial Bus (USB) specification revision 2.0. The host controller supports USB 2.0 full-speed and low-speed transfers, and the device controller supports USB 2.0 full-speed transfers. The USBFS has an internal USB transceiver and supports all of the transfer types defined in the USB 2.0 specification

The USBFS has FIFO buffer for data transfers, providing a maximum of 10 pipes. Any endpoint number can be assigned to pipes 1 to 9, based on the peripheral devices or the communication requirements for your system.

Table 36.1 lists the USBFS specifications, Figure 36.1 shows a block diagram, and Table 36.2 lists the I/O pins.

Table 36.1 USBFS specifications

Parameter	Specifications
Features	• USB Device Controller (UDC) and USB 2.0 transceiver supporting host controller, and device controller, and On-The-Go (OTG) functions • Host and device controller can be switched by software • Self-power or bus power mode selectable
	Host controller features: • Full-speed transfer (12 Mbps) and low-speed transfer (1.5 Mbps) • Automatic scheduling for SOF and packet transmissions • Programmable intervals for isochronous and interrupt transfers • Communications with multiple peripheral devices connected through a single hub
	Device controller features: • Full-speed transfer (12 Mbps)*1 • Control transfer stage control function • Device state control function • Auto response function for SET_ADDRESS request • SOF interpolation
Supported transfer types	• Control transfer • Bulk transfer • Interrupt transfer • Isochronous transfer
Pipe configuration	• FIFO buffer for USB communication • Up to 10 pipes selectable, including the Default Control Pipe (DCP) • Pipes 1 to 9 assignable to any endpoint number
	Transfer conditions specifiable for each pipe: • Pipe 0: Control transfer with 64-byte single buffer • Pipes 1 and 2: Selectable to bulk transfer with 64-byte double buffer or isochronous transfer with 256-byte double buffer • Pipes 3 to 5: Bulk transfer with 64-byte double buffer • Pipes 6 to 9: Interrupt transfer with 64-byte single buffer
Other features	• Reception end function using transaction count • Function that changes the BRDY interrupt event notification timing (BFRE) • Automatic clearing of the FIFO buffer after the data for the pipe specified in the DnFIFO port (n = 0, 1) is read (DCLRM) • NAK setting function for response PID generated on transfer end (SHTNAK) • On-chip pull-up and pull-down resistors for D+ and D–
Module-stop function	Module-stop state can be set to reduce power consumption
TrustZone Filter	Security and Privilege attribution can be set

Note 1. Low-speed transfer (1.5 Mbps) is not supported.

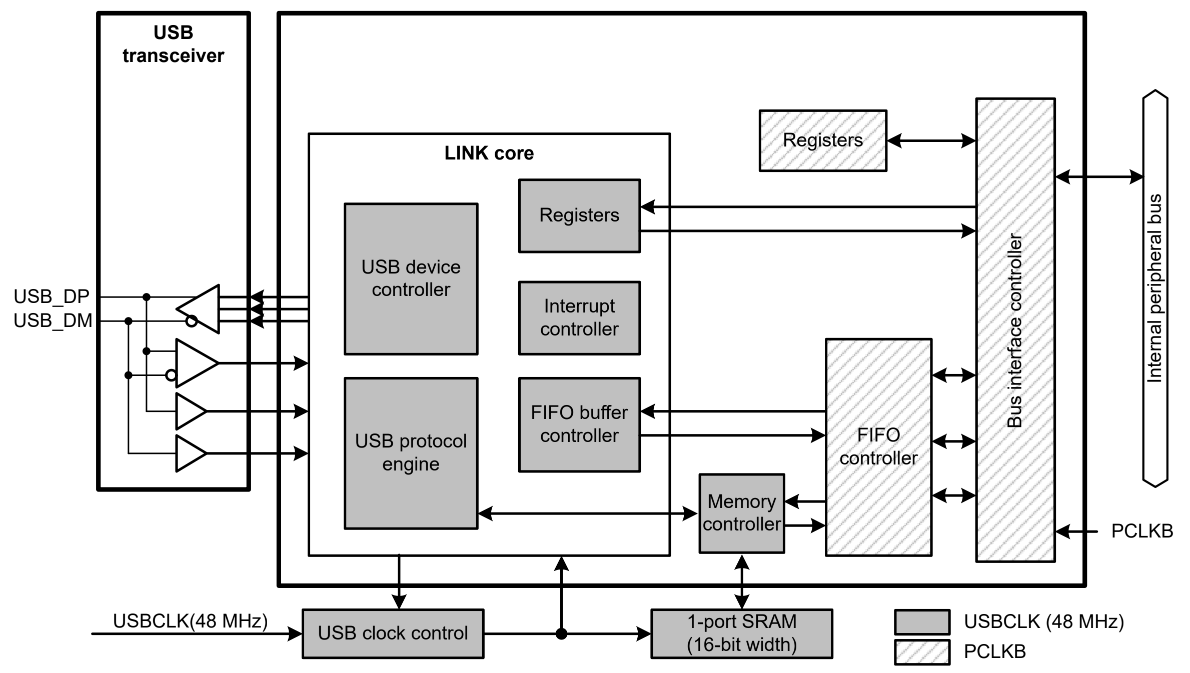


Figure 36.1 USBFS block diagram

Table 36.2 USBFS pin configuration

Function	Pin name	I/O	Description
USBFS	USB_DP	I/O	D+ I/O pin of the USB on-chip transceiver. This pin should be connected to the D+ pin of the USB bus.
	USB_DM	I/O	D– I/O pin of the USB on-chip transceiver. This pin should be connected to the D– pin of the USB bus.
	USB_VBUS	Input	USB cable connection monitor pin. This pin should be connected to VBUS of the USB bus. The VBUS pin status (connected or disconnected) can be detected when the USB module is operating as a function controller.
	USB_EXICEN	Output	Low-power control signal for external power supply (OTG) chip.
	USB_VBUSEN	Output	VBUS (5 V) supply enable signal for external power supply chip.
	USB_OVRCURA, USB_OVRCURB USB_OVRCURA-DS, USB_OVRCURB-DS	Input	External overcurrent detection signals should be connected to these pins. VBUS comparator signals should be connected to these pins when the OTG power supply chip is connected. USB_OVRCURA or USB_OVRCURB pins can be used in Software standby mode or in normal mode. USB_OVRCURA-DS, USB_OVRCURB-DS are dedicated pins that can generate interrupt in Software standby mode or in normal mode as well as for cancel Deep Software Standby mode 1.
	USB_ID	Input	MicroAB connector ID input signal should be connected to this pin during operation in OTG mode.
	VCC_USB	Input	Power supply pins.
	VSS_USB	Input	Ground pins.

36.2 Register Descriptions

36.2.1 SYSCFG : System Configuration Control Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x000

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SCKE	—	—	—	DCFM	DRPD	DPRPU	—	—	—	USBE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	USBE	USBFS Operation Enable 0: Disable 1: Enable	R/W
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	DPRPU	D+ Line Resistor Control 0: Disable line pull-up 1: Enable line pull-up	R/W
5	DRPD	D+/D– Line Resistor Control 0: Disable line pull-down 1: Enable line pull-down	R/W
6	DCFM	Controller Function Select 0: Select device controller 1: Select host controller	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
9:8	—	These bits are read as 0. The write value should be 0.	R/W
10	SCKE	USB Clock Enable 0: Stop clock supply to the USBFS 1: Enable clock supply to the USBFS	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: After writing 1 to the SCKE bit, read it to confirm that it is set to 1.

USBE bit (USBFS Operation Enable)

The USBE bit enables or disables operation of the USBFS.

Changing the USBE bit from 1 to 0 initializes the bits listed in Table 36.3. Only change this bit while the SCKE bit is 1. In host controller mode, this bit must be set to 1 after setting the DRPD bit to 1, eliminating SYSSTS0.LNST[1:0] flags chattering, and confirming that the USB bus state is stable.

Table 36.3 Registers initialized by writing 0 to the SYSCFG.USBE bit

Selected function	Register	Bit	Remarks
Device controller	SYSSTS0	LNST[1:0]	Value is saved in host controller mode
	DVSTCTR0	RHST[2:0]	—
	INTSTS0	DVSQ[2:0]	Value is saved in host controller mode
	USBADDR	USBADDR[6:0]	Value is saved in host controller mode
	USBREQ	BREQUEST[7:0], BMREQUESTTYPE[7:0]	Value is saved in host controller mode
	USBVAL	WVALUE[15:0]	Value is saved in host controller mode
	USBINDX	WINDEX[15:0]	Value is saved in host controller mode
	USBLENG	WLENTUH[15:0]	Value is saved in host controller mode
Host controller	DVSTCTR0	RHST[2:0]	—
	FRMNUM	FRNM[10:0]	Value is saved in device controller mode

DPRPU bit (D+ Line Resistor Control)

The DPRPU bit enables or disables pulling up the D+ line in device controller mode.

When the DPRPU bit is set to 1 in device controller mode, the USBFS pulls up the D+ line to notify the USB host that it attached. Changing the DPRPU bit from 1 to 0 releases the pull-up, thereby notifying the USB host that it detached.

Set this bit to 1 in device controller mode and to 0 in host controller mode.

DRPD bit (D+/D– Line Resistor Control)

TheDRPD bit enables or disables pulling down D+ and D– lines in host controller mode.

Set this bit to 1 in host controller mode and to 0 in device controller mode.

DCFM bit (Controller Function Select)

The DCFM bit selects the host or device function of the USBFS.

Only change this bit when the DPRPU and DRPD bits are both 0.

SCKE bit (USB Clock Enable)

The SCKE bit stops or enables the 48-MHz clock supply to the USBFS.

When this bit is 0, only SYSCFG is permitted to be read from and written to; the other registers related to the USB should not be read from or written to.

36.2.2 SYSSTS0 : System Configuration Status Register 0

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x004

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OVCMON[1:0]		—	—	—	—	—	—	—	HTACT	SOFEA	—	—	IDMON	LNST[1:0]	
Value after reset:	x	x	0	0	0	0	0	0	0	0	0	0	0	x	0	0

Bit	Symbol	Function	R/W
1:0	LNST[1:0]	USB Data Line Status Monitor Indicates the status of the USB data lines, see Table 36.4	R
2	IDMON	External ID0 Input Pin Monitor 0: USB_ID pin is low 1: USB_ID pin is high	R
4:3	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
5	SOFEA	Active Monitor When the Host Controller Is Selected 0: SOF output stopped 1: SOF output operating	R
6	HTACT	USB Host Sequencer Status Monitor 0: Host sequencer completely stopped 1: Host sequencer not completely stopped	R
13:7	—	These bits are read as 0.	R
15:14	OVCMON[1:0]	External USB_OVRCURA, USB_OVRCURA-DS, USB_OVRCURB or USB_OVRCURB-DS Input Pin Monitor OVCMON[1] indicates the USB_OVRCURA or USB_OVRCURA-DS pin status. OVCMON [0] indicates the USB_OVRCURB or USB_OVRCURB-DS pin status.	R

Note: S-TYPE-3, P-TYPE-3

Note: The values of the OVCMON[1:0] and IDMON bits depend on the status of the USB_OVRCURA, USB_OVRCURA-DS, USB_OVRCURB or USB_OVRCURB-DS and USB_ID pins.

LNST[1:0] bits (USB Data Line Status Monitor)

The LNST[1:0] bits indicate the state of the USB data lines, D+ and D–. For details, see Table 36.4.

In device controller mode, read the LNST[1:0] bits after connection processing (SYSCFG.DPRPU bit = 1). In host controller mode, read them after enabling pull-down of the lines (SYSCFG.DRPD bit = 1).

Table 36.4 Status of the USB data bus lines (D+ and D–)

LNST[1:0] bits	During full-speed operation	During low-speed operation
00b	SE0	SE0
01b	J-State	K-State
10b	K-State	J-State
11b	SE1	SE1

SOFEA bit (Active Monitor When the Host Controller Is Selected)

The SOFEA bit is used in host controller mode to check whether the output of the last SOF is complete when the USBFS is suspended because of a 0 setting to the DVSTCTR0.UACT bit.

In host controller mode, check that both the HTACT and SOFEA bits are 0 before setting the SYSCFG.USBE bit to 0 to stop the USBFS or setting the SYSCFG.SCKE bit to 0 to stop the clock signal supply during communication.

HTACT bit (USB Host Sequencer Status Monitor)

The HTACT bit is set to 0 when the host sequencer of the USBFS is completely stopped.

In host controller mode, check that the HTACT bit is 0 before setting the DVSTCTR0.UACT bit to 0 to place the USBFS in the suspended state or setting the SYSCFG.SCKE bit to 0 to stop the clock signal supply during communication.

OVCMON[1:0] bits (External USB_OVRCURA, USB_OVRCURA-DS, USB_OVRCURB or USB_OVRCURB-DS Input Pin Monitor)

The OVCMON[1:0] bits indicate the status of the overcurrent signals from an external power supply IC.

36.2.3 DVSTCTR0 : Device State Control Register 0

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x008

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	HNPBTOA	EXICEN	VBUSEN	WKUP	RWUPE	USBRST	RESUME	UACT	—	RHST[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	RHST[2:0]	USB Bus Reset Status 0 0 0: In host controller mode: Communication speed indeterminate (powered state or no connection) In device controller mode: Communication speed indeterminate 0 0 1: In host controller mode: Low-speed connection In device controller mode: USB bus reset in progress 0 1 0: In host controller mode: Full-speed connection In device controller mode: USB bus reset in progress or full-speed connection 0 1 1: Setting prohibited Others: In host controller mode: USB bus reset in progress In device controller mode: Setting prohibited	R
3	—	This bit is read as 0. The write value should be 0.	R/W
4	UACT	USB Bus Enable 0: Disable downstream port (disable SOF transmission) 1: Enable downstream port (enable SOF transmission)	R/W
5	RESUME	Resume Output 0: Do not output resume signal 1: Output resume signal	R/W
6	USBRST	USB Bus Reset Output 0: Do not output USB bus reset signal 1: Output USB bus reset signal	R/W
7	RWUPE	Wakeup Detection Enable 0: Disable downstream port remote wakeup 1: Enable downstream port remote wakeup	R/W
8	WKUP	Wakeup Output 0: Do not output remote wakeup signal 1: Output remote wakeup signal	R/W
9	VBUSEN	USB_VBUSEN Output Pin Control 0: Output low on external USB_VBUSEN pin 1: Output high on external USB_VBUSEN pin	R/W
10	EXICEN	USB_EXICEN Output Pin Control 0: Output low on external USB_EXICEN pin 1: Output high on external USB_EXICEN pin	R/W
11	HNPBTOA	Host Negotiation Protocol (HNP) Control Use this bit when switching from device B to device A in OTG mode. If the HNPBTOA bit is 1, the internal function control remains in the Suspend state until the HNP processing ends even if SYSCFG.DPRPU = 0 or SYSCFG.DCFM = 1.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The USBFS controller does not support low-speed connections in device controller mode. When this value is read, abnormal connection processing must be executed in higher level application software.

RHST[2:0] bits (USB Bus Reset Status)

RHST[2:0] bits indicate the status of the USB bus reset.

In host controller mode, writing 1 to the USBRST bit causes the RHST[2:0] bits to set to 100b. When 0 is written to the USBRST bit and the USBFS ends the SE0 state, the RHST[2:0]bits update to a new value.

In device controller mode, if the USBFS detects a USB bus reset, the RHST[2:0] bits indicate 010b if the DPRPU bit is 1, and a DVST interrupt is generated.

UACT bit (USB Bus Enable)

When set to 1 in host controller mode, the UACT bit enables USB bus operation by controlling SOF packet transmission to the USB bus in addition to data and reception. The USBFS starts SOF packet output within one frame period after the UACT bit is set to 1. When UACT is set to 0, the USBFS enters the idle state after the SOF packet output.

The USBFS sets the UACT bit to 0 on any of the following conditions:

- A DTCH interrupt is detected during communication (when UACT = 1)

- An EOFERR interrupt is detected during communication (when UACT = 1)

Always write 1 to the UACT bit at the end of the USB bus reset processing (writing 0 to the USBRST bit) or at the end of resume processing from the suspended state (writing 0 to the RESUME bit).

In device controller mode, always set this bit to 0.

RESUME bit (Resume Output)

The RESUME bit controls the resume signal output in host controller mode.

When this bit is set to 1, the USBFS drives the USB port to the K-state and outputs the resume signal. The USBFS sets the bit to 1 on detection of a remote wakeup signal while the RWUPE bit is 1 and in the USB Suspend state.

The USBFS continues outputting the K-state while the RESUME bit is 1, until the bit is cleared to 0 by software. The RESUME bit must be 1 (resume period) for the time defined in the USB 2.0 specification. Only set this bit to 1 while the interface is in the Suspend state. Write 1 to the UACT bit simultaneously with the end of the resume processing (writing 0 to the RESUME bit).

Always set this bit to 0 in device controller mode.

USBRST bit (USB Bus Reset Output)

The USBRST bit controls the output of the USB bus signal in host controller mode. When this bit set to 1, the USBFS drives the USB port to the SE0 state to reset the USB bus. The USBFS continues outputting SE0 while the USBRST bit is 1, until the bit is cleared to 0 by software. The USBRST bit must be 1 (USB bus reset period) for the time defined in the USB 2.0 specification. Writing 1 to the USBRST bit during communication (UACT = 1) or during resume processing (RESUME = 1) prevents the USBFS from starting USB bus reset processing until both the UACT and RESUME bits become 0. Write 1 to the UACT bit simultaneously with the end of the USB bus reset processing (writing 0 to the USBRST bit).

Always set this bit to 0 in device controller mode.

RWUPE bit (Wakeup Detection Enable)

The RWUPE bit enables or disables remote wakeup signals (resume signals) from downstream peripheral devices in host controller mode. When this bit is set to 1, the USBFS detects a remote wakeup signal (K-state for 2.5 µs) from a downstream peripheral device, and performs resume processing, driving the K-state. When the RWUPE bit is set to 0, the USBFS ignores remote wakeup signals (K-states) from peripheral devices connected to the USB port.

Do not stop the internal clock when the RWUPE bit is 1, even in the Suspend state (SYSCFG.SCKE bit must be set to 1).

Always set this bit to 0 in device controller mode.

WKUP bit (Wakeup Output)

The WKUP bit enables or disables remote wakeup signals (resume signals) to the USB bus in device controller mode.

The USBFS controls the output timing of the remote wakeup signals. When this bit is set to 1, the USBFS clears it to 0 after outputting the K-state for 10 ms. The USB 2.0 specification specifies that the USB bus idle state must be kept for 5 ms or longer before a remote wakeup signal is sent. If the USBFS writes 1 to the WKUP bit immediately after detecting the Suspend state, the K-state is output after 2 ms.

Only write 1 to the WKUP bit when the device is in the Suspend state (INTSTS0.DVSQ[2:0] = 1xxb) and the USB host enables the remote wakeup signal. Do not stop the internal clock while this bit is 1, even in the Suspend state (SYSCFG.SCKE bit must be set to 1).

Always set this bit to 0 in host controller mode.

HNPBTOA bit (Host Negotiation Protocol (HNP) Control)

TheHNPBTOA bit is used when switching from device B to device A while in OTG mode.

If the HNPBTOA bit is 1, the internal function control maintains the Suspend state until HNP processing ends, even if the SYSCFG.DPRPU bit is set to 0 or the SYSCFG.DCFM bit is set to 1. Resume interrupts (RESM) are not generated even if a falling edge of D+ is detected.

After this bit is set to 1, the HNP processing ends when a host attach event is detected, because of a pull-up by the initiating party, or the HNPBTOA bit is cleared to 0 by software because the HNP processing times out.

36.2.4 CFIFO/CFIFOL : CFIFO Port Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x014

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FIFOPORT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	FIFOPORT[15:0]*1	FIFO Port Read receive data from the FIFO buffer or write transmit data to the FIFO buffer by accessing these bits	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. The valid bits depend on the MBW settings (CFIFOSEL.MBW) and BIGEND settings (CFIFOSEL.BIGEND) in the associated port selection register. See Table 36.5 and Table 36.6.

Three FIFO ports are available:

- CFIFO
- D0FIFO
- D1FIFO

Each FIFO port is configured with:

- A port register (CFIFO, D0FIFO, or D1FIFO) that handles reading of data from the FIFO buffer and writing of data to the FIFO buffer
- A port selection register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) that selects the pipe assigned to the FIFO port
- A port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR)

Each FIFO port has the following constraints:

- Access to the FIFO buffer for DCP control transfers is through the CFIFO port
- Access to the FIFO buffer for DMA or DTC transfers is through the D0FIO or D1FIFO port
- The D0FIFO and D1FIFO ports can also be accessed by the CPU
- When using functions specific to the FIFO port, such as the DMA or DTC transfer function, you cannot change the pipe number selected in the CURPIPE[3:0] bits of the port selection register
- Registers configuring a FIFO port do not affect other FIFO ports
- The same pipe must not be assigned to two or more FIFO ports
- There are two FIFO buffer states, one giving access rights to the CPU and the other to the serial interface engine (SIE). When the SIE has access rights, the FIFO buffer cannot be accessed by the CPU

FIFOPORT[15:0] bits (FIFO Port)

When the FIFOPORT[15:0] bit is accessed, the USBFS reads the received data from the FIFO buffer or writes the transmit data to the FIFO buffer. The FIFO port register can be accessed only when the FRDY bit in the associated port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1.

The valid bits in the FIFO port register depend on the MBW and BIGEND settings in the port selection register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL). See Table 36.5 and Table 36.6.

Table 36.5 Endian operation in 16-bit access

CFIFOSEL.BIGEND bit	Bits [15:8]	Bits [7:0]
0	N + 1 data	N + 0 data
1	N + 0 data	N + 1 data

Table 36.6 Endian operation in 8-bit access

CFIFOSEL.BIGEND bit	Bits [15:8]	Bits [7:0]
0	Access prohibited[*1]	N + 0 data
1	Access prohibited[*1]	N + 0 data

Note 1. Writing to or reading from these areas is not allowed.

36.2.5 DnFIFO/DnFIFOL : DnFIFO Port Register (n = 0, 1)

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x018 + 0x4 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FIFOPORT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	FIFOPORT[15:0][*1]	FIFO Port Read receive data from the FIFO buffer or write transmit data to the FIFO buffer by accessing these bits	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The valid bits depend on the MBW settings (CFIFOSEL.MBW, D0FIFOSEL.MBW, and D1FIFOSEL.MBW) and BIGEND settings (CFIFOSEL.BIGEND, D0FIFOSEL.BIGEND, and D1FIFOSEL.BIGEND) in the associated port selection register. See Table 36.7 and Table 36.8.

Three FIFO ports are available:

- CFIFO
- D0FIFO
- D1FIFO

Each FIFO port is configured with:

- A port register (CFIFO, D0FIFO, or D1FIFO) that handles reading of data from the FIFO buffer and writing of data to the FIFO buffer
- A port selection register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) that selects the pipe assigned to the FIFO port
- A port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR)

Each FIFO port has the following constraints:

- Access to the FIFO buffer for DCP control transfers is through the CFIFO port
- Access to the FIFO buffer for DMA or DTC transfers is through the D0FIO or D1FIFO port
- The D0FIFO and D1FIFO ports can also be accessed by the CPU
- When using functions specific to the FIFO port, such as the DMA or DTC transfer function, you cannot change the pipe number selected in the CURPIPE[3:0] bits of the port selection register
- Registers configuring a FIFO port do not affect other FIFO ports
- The same pipe must not be assigned to two or more FIFO ports
- There are two FIFO buffer states, one giving access rights to the CPU and the other to the serial interface engine (SIE). When the SIE has access rights, the FIFO buffer cannot be accessed by the CPU

FIFOPORT[15:0] bits (FIFO Port)

When the FIFOPORT bit is accessed, the USBFS reads the received data from the FIFO buffer or writes the transmit data to the FIFO buffer. The FIFO port register can be accessed only when the FRDY bit in the associated port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1.

The valid bits in the FIFO port register depend on the MBW and BIGEND settings in the port selection register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL). See Table 36.7 and Table 36.8.

Table 36.7 Endian operation in 16-bit access

CFIFOSEL.BIGEND bit D0FIFOSEL.BIGEND bit D1FIFOSEL.BIGEND bit	Bits [15:8]	Bits [7:0]
0	N + 1 data	N + 0 data
1	N + 0 data	N + 1 data

Table 36.8 Endian operation in 8-bit access

CFIFOSEL.BIGEND bit D0FIFOSEL.BIGEND bit D1FIFOSEL.BIGEND bit	Bits [15:8]	Bits [7:0]
0	Access prohibited*1	N + 0 data
1	Access prohibited*1	N + 0 data

Note 1. Writing to or reading from these areas is not allowed.

36.2.6 CFIFOSEL : CFIFO Port Select Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x020

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RCNT	REW	—	—	—	MBW	—	BIGEND	—	—	ISEL	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CURPIPE[3:0]	CFIFO Port Access Pipe Specification 0x0: Default Control Pipe 0x1: Pipe 1 0x2: Pipe 2 0x3: Pipe 3 0x4: Pipe 4 0x5: Pipe 5 0x6: Pipe 6 0x7: Pipe 7 0x8: Pipe 8 0x9: Pipe 9 Others: Setting prohibited	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	ISEL	CFIFO Port Access Direction When DCP Is Selected 0: Select reading from the FIFO buffer 1: Select writing to the FIFO buffer	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	BIGEND	CFIFO Port Endian Control 0: Little endian 1: Big endian	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
10	MBW	CFIFO Port Access Bit Width 0: 8-bit width 1: 16-bit width	R/W
13:11	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
14	REW	Buffer Pointer Rewind 0: Do not rewind buffer pointer 1: Rewind buffer pointer	W*1
15	RCNT	Read Count Mode 0: The DTLN[8:0] bits (CFIFOCTR.DTLN[8:0], D0FIFOCTR.DTLN[8:0], D1FIFOCTR.DTLN[8:0]) are cleared when all receive data is read from the CFIFO. In double buffer mode, the DTLN[8:0] value is cleared when all data is read from only a single plane. 1: The DTLN[8:0] bits are decremented each time the receive data is read from the CFIFO.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Only 0 can be read.

Do not specify the same pipe number in the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected.

Do not change the pipe number while DMA or DTC transfer is enabled.

CURPIPE[3:0] bits (CFIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number to use for reading or writing data through the CFIFO port. After writing to these bits, read them to check that the written value agrees with the read value before proceeding to the next process. Do not set the same pipe number to the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

During FIFO buffer access, even when an attempt is made to change the CURPIPE[3:0] setting, the current access setting is retained until access is complete.

ISEL bit (CFIFO Port Access Direction When DCP Is Selected)

After writing a new value to the ISEL bit with the DCP as the selected pipe, read the ISEL bit to check that the written value agrees with the read value before proceeding to the next process. Set the ISEL and CURPIPE[3:0] bits simultaneously.

MBW bit (CFIFO Port Access Bit Width)

The MBW bit specifies the bit width for accessing the CFIFO port.

When the selected pipe is receiving, set the CURPIPE[3:0] and MBW bits simultaneously. After a write to these bits starts a data read from the FIFO buffer, do not change the MBW bit until all of the data is read.

When the selected pipe is transmitting, the bit width cannot be changed from 8-bit to 16-bit while data is being written to the FIFO buffer.

An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

REW bit (Buffer Pointer Rewind)

The REW bit specifies whether to rewind the buffer pointer.

When the selected pipe is receiving, setting this bit to 1 while the FIFO buffer is being read allows re-reading of the FIFO buffer from the first data. In double buffering, this setting enables re-reading of the currently-read FIFO buffer plane from the first entry.

Do not set this bit to 1 while simultaneously changing the CURPIPE[3:0] bits. Before setting the REW bit to 1, be sure to check that the FRDY bit is 1.

To rewrite to the FIFO buffer from the first data for the transmitting pipe, use the BCLR bit.

36.2.7 DnFIFOSEL : DnFIFO Port Select Register (n = 0, 1)

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x028 + 0x4 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RCNT	REW	DCLRM	DREQE	—	MBW	—	BIGEND	—	—	—	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CURPIPE[3:0]	FIFO Port Access Pipe Specification 0x0: Default Control Pipe 0x1: Pipe 1 0x2: Pipe 2 0x3: Pipe 3 0x4: Pipe 4 0x5: Pipe 5 0x6: Pipe 6 0x7: Pipe 7 0x8: Pipe 8 0x9: Pipe 9 Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	BIGEND	FIFO Port Endian Control 0: Little endian 1: Big endian	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
10	MBW	FIFO Port Access Bit Width 0: 8-bit width 1: 16-bit width	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	DREQE	DMA/DTC Transfer Request Enable 0: Disable DMA/DTC transfer request 1: Enable DMA/DTC transfer request	R/W
13	DCLRM	Auto Buffer Memory Clear Mode Accessed after Specified Pipe Data is Read 0: Disable auto buffer clear mode 1: Enable auto buffer clear mode	R/W
14	REW	Buffer Pointer Rewind 0: Do not rewind buffer pointer 1: Rewind buffer pointer	W
15	RCNT	Read Count Mode 0: Clear DTLN[8:0] bits in (CFIFOCTR.DTLN[8:0], D0FIFOCTR.DTLN[8:0], D1FIFOCTR.DTLN[8:0]) when all receive data is read from DnFIFO (after read of a single plane in double buffer mode) 1: Decrement DTLN[8:0] bits each time receive data is read from DnFIFO	R/W

Note: S-TYPE-3, P-TYPE-3

The same pipe must not be specified in the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected. The pipe number must not be changed while DMA or DTC transfer is enabled.

CURPIPE[3:0] bits (FIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number to use for reading or writing data through the DnFIFO port. After writing to these bits, read them to check that the written value agrees with the read value before proceeding to the next process. Do not set the same pipe number to the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

During FIFO buffer access, even when an attempt is made to change the CURPIPE[3:0] setting, the current access setting is retained until access is complete.

MBW bit (FIFO Port Access Bit Width)

The MBW bit specifies the bit width for accessing the DnFIFO port.

When the selected pipe is receiving, after a write to these bits starts a data read from the FIFO buffer, do not change the MBW bit until all of the data is read. Set the CURPIPE[3:0] and MBW bits simultaneously.

When the selected pipe is transmitting, the bit width cannot be changed from 8-bit to 16-bit while data is being written to the FIFO buffer.

An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

DREQE bit (DMA/DTC Transfer Request Enable)

The DREQE bit enables or disables issuing of DMA or DTC transfer requests. To enable DMA or DTC transfer requests, set this bit to 1 after setting the CURPIPE[3:0] bits. To change the CURPIPE[3:0] setting, first set this bit to 0.

DCLRM bit (Auto Buffer Memory Clear Mode Accessed after Specified Pipe Data is Read)

The DCLRM bit enables or disables automatic FIFO buffer clearing after data in the selected pipe is read.

When this bit is set to 1, on receiving a zero-length packet while the FIFO buffer assigned to the selected pipe is empty, or when reading of a received short packet is complete while the PIPECFG.BFRE bit is 1, the USBFS sets the BCLR bit in the FIFO port control register to 1.

When using the USBFS with the SOFCFG.BRDYM bit set to 1, set this bit to 0.

REW bit (Buffer Pointer Rewind)

The REW bit specifies whether to rewind the buffer pointer.

When the selected pipe is receiving, setting this bit to 1 while the FIFO buffer is being read allows re-reading of the FIFO buffer from the first data. In double buffering, this setting enables re-reading of the currently-read FIFO buffer plane from the first entry.

Do not set this bit to 1 while simultaneously changing the CURPIPE[3:0] bits. Before setting the bit to 1, be sure to check that the FRDY bit is 1.

To rewrite to the FIFO buffer from the first data for the transmitting pipe, use the BCLR bit.

RCNT bit (Read Count Mode)

The RCNT bit specifies the read mode for the value in the D0FIFOCTL.DTLN bit and D1FIFOCTL.DTLN bit. When accessing DnFIFO with the PIPECFG.BFRE bit set to 1, set the RCNT bit to 0.

36.2.8 CFIFOCTR : CFIFO Port Control Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x022

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BVAL	BCLR	FRDY	—	—	—	—	DTLN[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	DTLN[8:0]	Receive Data Length Indicates the receive data length. The meaning of the values differs depending on the RCNT bit setting in the port select register. For details, see the description of the DTLN[8:0] bits.	R
12:9	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
13	FRDY	FIFO Port Ready 0: FIFO port access disabled 1: FIFO port access enabled	R
14	BCLR	CPU Buffer Clear 0: No operation 1: Clear FIFO buffer on the CPU side	W
15	BVAL	Buffer Memory Valid Flag 0: Invalid (writing 0 has no effect) 1: Writing ended	R/W

Note: S-TYPE-3, P-TYPE-3

The CFIFOCTR, D0FIFOCTR, and D1FIFOCTR registers correspond to the CFIFO, D0FIFO, and D1FIFO buffers.

DTLN[8:0] bits (Receive Data Length)

The DTLN[8:0] bits indicate the length of the receive data.

While the FIFO buffer is being read, the DTLN[8:0] bits indicate different values depending on the DnFIFOSEL.RCNT bit (n = 0, 1), as follows:

- RCNT = 0

  The USBFS sets the DTLN[8:0] bits to indicate the length of the receive data until the CPU or DMA/DTC has read all of the received data from a single FIFO buffer plane.

  While the PIPECFG.BFRE bit = 1, the USBFS retains the length of the receive data until the BCLR bit is set to 1, even after all the data is read.

- RCNT = 1

  The USBFS decrements the value indicated in the DTLN[8:0] bits each time data is read from the FIFO buffer. The value is decremented by 1 when MBW = 0, and by 2 when MBW = 1.

  The USBFS sets these bits to 0 when all the data is read from one FIFO buffer plane. In double buffer mode, if data is received in one FIFO buffer plane before all of the data is read from the other plane, the USBFS sets these bits to indicate the length of the receive data in the former plane when all of the data is read from the latter plane.

FRDY bit (FIFO Port Ready)

The FRDY bit indicates whether the FIFO port can be accessed by the CPU or DMA/DTC.

In the following cases, the USBFS sets the FRDY bit to 1 but data cannot be read through the FIFO port because there is no data to be read:

- A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty
- A short packet is received and the data is completely read while the PIPECFG.BFRE bit = 1

In these cases, set the BCLR bit to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.

BCLR bit (CPU Buffer Clear)

Set the BCLR bit to 1 to clear the FIFO buffer on the CPU side for the selected pipe.

When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USBFS clears only one plane of the FIFO buffer even when both planes are read-enabled.

When the DCP is the selected pipe, setting the BCLR bit to 1 allows the USBFS to clear the FIFO buffer regardless of whether the CPU or SIE has access rights. To clear the buffer when the SIE has access rights, set the DCPCTR.PID[1:0] bits to 00b (NAK response) before setting the BCLR bit to 1.

When the selected pipe is transmitting, if 1 is written to the BVAL flag and the BCLR bit simultaneously, the USBFS clears the data that is already written, enabling transmission of a zero-length packet.

When the selected pipe is not the DCP, only write 1 to the BCLR bit while the FRDY bit in the FIFO port control register is 1 (set by the USBFS).

BVAL flag (Buffer Memory Valid Flag)

Set the BVAL flag to 1 when data is completely written to the FIFO buffer on the CPU side for the pipe selected in CURPIPE[3:0].

When the selected pipe is transmitting, set this flag to 1 in the following cases:

- To transmit a short packet, set this flag to 1 after data is written
- To transmit a zero-length packet, set this flag to 1 before data is written to the FIFO buffer

The USBFS then switches the FIFO buffer from the CPU side to the SIE side, enabling transmission.

When data of the maximum packet size is written for the pipe in continuous transfer mode, the USBFS sets the BVAL flag to 1 and switches the FIFO buffer from the CPU side to the SIE side, enabling transmission.

Only write 1 to the BVAL flag while the FRDY bit is 1 (set by the USBFS). When the selected pipe is receiving, do not set the BVAL flag to 1.

36.2.9 DnFIFOCTR : DnFIFO Port Control Register (n = 0, 1)

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x02A + 0x4 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BVAL	BCLR	FRDY	—	—	—	—	DTLN[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	DTLN[8:0]	Receive Data Length Indicates the receive data length. The meaning of the values differs depending on the RCNT bit setting in the port select register. For details, see the description of the DTLN[8:0] bits.	R
12:9	—	These bits are read as 0. The write value should be 0.	R/W
13	FRDY	FIFO Port Ready 0: FIFO port access disabled 1: FIFO port access enabled	R
14	BCLR	CPU Buffer Clear 0: No operation 1: Clear FIFO buffer on the CPU side	R/W*1
15	BVAL	Buffer Memory Valid Flag 0: Invalid (writing 0 has no effect) 1: Writing ended	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Only 0 can be read.

The CFIFOCTR, D0FIFOCTR, and D1FIFOCTR registers correspond to the CFIFO, D0FIFO, and D1FIFO buffers.

DTLN[8:0] bits (Receive Data Length)

The DTLN[8:0] bits indicate the length of the receive data.

While the FIFO buffer is being read, the DTLN[8:0] bits indicate different values depending on the DnFIFOSEL.RCNT bit (n = 0, 1), as follows:

- RCNT = 0

  The USBFS sets the DTLN[8:0] bits to indicate the length of the receive data until the CPU or DMA/DTC has read all of the received data from a single FIFO buffer plane.

  While the PIPECFG.BFRE bit = 1, the USBFS retains the length of the receive data until the BCLR bit is set to 1, even after all the data is read.

- RCNT = 1

The USBFS decrements the value indicated in the DTLN[8:0] bits each time data is read from the FIFO buffer. The value is decremented by 1 when MBW = 0, and by 2 when MBW = 1.

The USBFS sets these bits to 0 when all the data is read from one FIFO buffer plane. In double buffer mode, if data is received in one FIFO buffer plane before all of the data is read from the other plane, the USBFS sets these bits to indicate the length of the receive data in the former plane when all of the data is read from the latter plane.

FRDY bit (FIFO Port Ready)

The FRDY bit indicates whether the FIFO port can be accessed by the CPU or DMA/DTC.

In the following cases, the USBFS sets the FRDY bit to 1 but data cannot be read through the FIFO port because there is no data to be read:

- A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty
- A short packet is received and the data is completely read while the PIPECFG.BFRE bit = 1

In these cases, set the BCLR bit to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.

BCLR bit (CPU Buffer Clear)

Set the BCLR bit to 1 to clear the FIFO buffer on the CPU side for the selected pipe.

When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USBFS clears only one plane of the FIFO buffer even when both planes are read-enabled.

When the DCP is the selected pipe, setting the BCLR bit to 1 allows the USBFS to clear the FIFO buffer regardless of whether the CPU or SIE has access rights. To clear the buffer when the SIE has access rights, set the DCPCTR.PID[1:0] bits to 00b (NAK response) before setting the BCLR bit to 1.

When the selected pipe is transmitting, if 1 is written to the BVAL flag and the BCLR bit simultaneously, the USBFS clears the data that is already written, enabling transmission of a zero-length packet.

When the selected pipe is not the DCP, only write 1 to the BCLR bit while the FRDY bit in the FIFO port control register is 1 (set by the USBFS).

BVAL flag (Buffer Memory Valid Flag)

Set the BVAL flag to 1 when data is completely written to the FIFO buffer on the CPU side for the pipe selected in CURPIPE[3:0].

When the selected pipe is transmitting, set this flag to 1 in the following cases:

- To transmit a short packet, set this flag to 1 after data is written
- To transmit a zero-length packet, set this flag to 1 before data is written to the FIFO buffer

The USBFS then switches the FIFO buffer from the CPU side to the SIE side, enabling transmission.

When data of the maximum packet size is written for the pipe in continuous transfer mode, the USBFS sets the BVAL flag to 1 and switches the FIFO buffer from the CPU side to the SIE side, enabling transmission.

Only write 1 to the BVAL flag while the FRDY bit is 1 (set by the USBFS). When the selected pipe is receiving, do not set the BVAL flag to 1.

36.2.10 INTENB0 : Interrupt Enable Register 0

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x030

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	BRDYE	Buffer Ready Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
9	NRDYE	Buffer Not Ready Response Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
10	BEMPE	Buffer Empty Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
11	CTRE	Control Transfer Stage Transition Interrupt Enable *1 0: Disable interrupt request 1: Enable interrupt request	R/W
12	DVSE	Device State Transition Interrupt Enable *1 0: Disable interrupt request 1: Enable interrupt request	R/W
13	SOFE	Frame Number Update Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
14	RSME	Resume Interrupt Enable*1 0: Disable interrupt request 1: Enable interrupt request	R/W
15	VBSE	VBUS Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. The RSME, DVSE, and CTRE bits can only be set to 1 in device controller mode. Do not set these bits to 1 in host controller mode.

When a status flag in the INTSTS0 register sets to 1 and the associated interrupt request enable bit setting in theINTENB0 register is 1, the USBFS issues a USBFS interrupt request.

Regardless of the INTENB0 register setting, the status flag in the INTSTS0 register sets to 1 in response to a state change that satisfies the associated condition.

When an interrupt request enable bit in the INTENB0 register is switched from 0 to 1 while the associated status flag in the INTSTS0 register is set to 1, a USBFS interrupt is requested.

36.2.11 INTENB1 : Interrupt Enable Register 1

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x032

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OVRCRE	BCHGE	—	DTCHE	ATTCHE	—	—	—	—	EOFERRE	SIGNE	SACKE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	SACKE	Setup Transaction Normal Response Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
5	SIGNE	Setup Transaction Error Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W

Bit	Symbol	Function	R/W
6	EOFERRE	EOF Error Detection Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
10:7	—	These bits are read as 0. The write value should be 0.	R/W
11	ATTCHE	Connection Detection Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
12	DTCHE	Disconnection Detection Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
13	—	This bit is read as 0. The write value should be 0.	R/W
14	BCHGE	USB Bus Change Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
15	OVRCRE	Overcurrent Input Change Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W

Note: S-TYPE-3, P-TYPE-3
Note: The bits in INTENB1 can only be set to 1 in host controller mode. Do not set these bits to 1 in device controller mode.

INTENB1 specifies the interrupt masks in host controller mode and for the setup transaction.

When a status flag in the INTSTS1 register sets to 1 and the associated interrupt request enable bit setting in theINTENB1 register is 1, the USBFS issues a USBFS interrupt request.

Regardless of the INTENB1 register setting, the status flag in the INTSTS1 register sets to 1 in response to a state change that satisfies the associated condition.

When an interrupt request enable bit in the INTENB1 register is switched from 0 to 1 while the associated status flag in the INTSTS1 register is set to 1, a USBFS interrupt is requested.

Do not enable interrupts in device controller mode.

36.2.12 BRDYENB : BRDY Interrupt Enable Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x036

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPE9 BRDY E	PIPE8 BRDY E	PIPE7 BRDY E	PIPE6 BRDY E	PIPE5 BRDY E	PIPE4 BRDY E	PIPE3 BRDY E	PIPE2 BRDY E	PIPE1 BRDY E	PIPE0 BRDY E
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIPE0BRDYE	BRDY Interrupt Enable for Pipe 0 0: Disable interrupt request 1: Enable interrupt request	R/W
1	PIPE1BRDYE	BRDY Interrupt Enable for Pipe 1 0: Disable interrupt request 1: Enable interrupt request	R/W
2	PIPE2BRDYE	BRDY Interrupt Enable for Pipe 2 0: Disable interrupt request 1: Enable interrupt request	R/W
3	PIPE3BRDYE	BRDY Interrupt Enable for Pipe 3 0: Disable interrupt request 1: Enable interrupt request	R/W

Bit	Symbol	Function	R/W
4	PIPE4BRDYE	BRDY Interrupt Enable for Pipe 4 0: Disable interrupt request 1: Enable interrupt request	R/W
5	PIPE5BRDYE	BRDY Interrupt Enable for Pipe 5 0: Disable interrupt request 1: Enable interrupt request	R/W
6	PIPE6BRDYE	BRDY Interrupt Enable for Pipe 6 0: Disable interrupt request 1: Enable interrupt request	R/W
7	PIPE7BRDYE	BRDY Interrupt Enable for Pipe 7 0: Disable interrupt request 1: Enable interrupt request	R/W
8	PIPE8BRDYE	BRDY Interrupt Enable for Pipe 8 0: Disable interrupt request 1: Enable interrupt request	R/W
9	PIPE9BRDYE	BRDY Interrupt Enable for Pipe 9 0: Disable interrupt request 1: Enable interrupt request	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The BRDYENB register enables or disables the INTSTS0.BRDY bit to be set to 1 when a BRDY interrupt is detected for each pipe.

When a status flag in the BRDYSTS register sets to 1 and the associated PIPEnBRDYE bit (n = 0 to 9) setting in theBRDYENB register is 1, the INTSTS0.BRDY flag sets to 1. In this case, if the BRDYE bit in INTENB0 is 1, the USBFS generates a BRDY interrupt request. While at least one PIPEnBRDY bit indicates 1, the USB generates the BRDY interrupt request when the associated interrupt request enable bit in the BRDYENB register is changed from 0 to 1 by software.

36.2.13 NRDYENB : NRDY Interrupt Enable Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x038

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPE9NRDYE	PIPE8NRDYE	PIPE7NRDYE	PIPE6NRDYE	PIPE5NRDYE	PIPE4NRDYE	PIPE3NRDYE	PIPE2NRDYE	PIPE1NRDYE	PIPE0NRDYE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIPE0NRDYE	NRDY Interrupt Enable for Pipe 0 0: Disable interrupt request 1: Enable interrupt request	R/W
1	PIPE1NRDYE	NRDY Interrupt Enable for Pipe 1 0: Disable interrupt request 1: Enable interrupt request	R/W
2	PIPE2NRDYE	NRDY Interrupt Enable for Pipe 2 0: Disable interrupt request 1: Enable interrupt request	R/W
3	PIPE3NRDYE	NRDY Interrupt Enable for Pipe 3 0: Disable interrupt request 1: Enable interrupt request	R/W
4	PIPE4NRDYE	NRDY Interrupt Enable for Pipe 4 0: Disable interrupt request 1: Enable interrupt request	R/W

Bit	Symbol	Function	R/W
5	PIPE5NRDYE	NRDY Interrupt Enable for Pipe 5 0: Disable interrupt request 1: Enable interrupt request	R/W
6	PIPE6NRDYE	NRDY Interrupt Enable for Pipe 6 0: Disable interrupt request 1: Enable interrupt request	R/W
7	PIPE7NRDYE	NRDY Interrupt Enable for Pipe 7 0: Disable interrupt request 1: Enable interrupt request	R/W
8	PIPE8NRDYE	NRDY Interrupt Enable for Pipe 8 0: Disable interrupt request 1: Enable interrupt request	R/W
9	PIPE9NRDYE	NRDY Interrupt Enable for Pipe 9 0: Disable interrupt request 1: Enable interrupt request	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The NRDYENB register enables or disables the INTSTS0.NRDY bit to be set to 1 when a NRDY interrupt is detected for each pipe.

When a status flag in the NRDYSTS register sets to 1 and the associated PIPEnNRDYE (n = 0 to 9) bit setting in theNRDYENB register is 1, the INTSTS0.NRDY flag sets to 1. In this case, if the NRDYE bit in INTENB0 is 1, the USBFS generates a NRDY interrupt request. While at least one PIPEnNRDY bit indicates 1, the USBFS generates the NRDY interrupt request when the associated interrupt request enable bit in the NRDYENB register is changed from 0 to 1 by software.

36.2.14 BEMPENB : BEMP Interrupt Enable Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x03A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPE9BEMPE	PIPE8BEMPE	PIPE7BEMPE	PIPE6BEMPE	PIPE5BEMPE	PIPE4BEMPE	PIPE3BEMPE	PIPE2BEMPE	PIPE1BEMPE	PIPE0BEMPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIPE0BEMPE	BEMP Interrupt Enable for Pipe 0 0: Disable interrupt request 1: Enable interrupt request	R/W
1	PIPE1BEMPE	BEMP Interrupt Enable for Pipe 1 0: Disable interrupt request 1: Enable interrupt request	R/W
2	PIPE2BEMPE	BEMP Interrupt Enable for Pipe 2 0: Disable interrupt request 1: Enable interrupt request	R/W
3	PIPE3BEMPE	BEMP Interrupt Enable for Pipe 3 0: Disable interrupt request 1: Enable interrupt request	R/W
4	PIPE4BEMPE	BEMP Interrupt Enable for Pipe 4 0: Disable interrupt request 1: Enable interrupt request	R/W

Bit	Symbol	Function	R/W
5	PIPE5BEMPE	BEMP Interrupt Enable for Pipe 5 0: Disable interrupt request 1: Enable interrupt request	R/W
6	PIPE6BEMPE	BEMP Interrupt Enable for Pipe 6 0: Disable interrupt request 1: Enable interrupt request	R/W
7	PIPE7BEMPE	BEMP Interrupt Enable for Pipe 7 0: Disable interrupt request 1: Enable interrupt request	R/W
8	PIPE8BEMPE	BEMP Interrupt Enable for Pipe 8 0: Disable interrupt request 1: Enable interrupt request	R/W
9	PIPE9BEMPE	BEMP Interrupt Enable for Pipe 9 0: Disable interrupt request 1: Enable interrupt request	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The BEMPENB register enables or disables the INTSTS0.BEMP bit to be set to 1 when a BEMP interrupt is detected for each pipe.

When a status flag in the BEMPSTS register sets to 1 and the associated PIPEnBEMPE (n = 0 to 9) bit setting in the BEMPENB register is 1, the INTSTS0.BEMP flag sets to 1. In this case, if the BEMPE bit in INTENB0 is 1, the USBFS generates a BEMP interrupt request. While at least one PIPEnBEMP bit indicates 1, the USBFS generates the BEMP interrupt request when the associated interrupt request enable bit in the BEMPENB register is changed from 0 to 1 by software.

36.2.15 SOFCFG : SOF Output Configuration Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x03C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	TRNENSEL	—	BRDYM	—	EDGESTS	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	EDGESTS	Edge Interrupt Output Status Monitor*1 Indicates 1 during the edge processing of an edge interrupt output signal.	R
5	—	This bit is read as 0. The write value should be 0.	R/W
6	BRDYM	BRDY Interrupt Status Clear Timing 0: Clear BRDY flag by software 1: Clear BRDY flag by the USBFS through a data read from the FIFO buffer or data write to the FIFO buffer	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	TRNENSEL	Transaction-Enabled Time Select*1 0: Not low-speed communication 1: Low-speed communication	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Confirm that the EDGESTS flag is 0 before stopping the clock supply to the USBFS.

EDGESTS bit (Edge Interrupt Output Status Monitor)

The EDGESTS bit indicates 1 during the edge processing of an edge interrupt output signal. Confirm that this bit is 0 before stopping the clock supply to the USBFS.

BRDYM bit (BRDY Interrupt Status Clear Timing)

The BRDYM bit specifies how the BRDY interrupt status flags for the pipes are cleared.

TRNENSEL bit (Transaction-Enabled Time Select)

When the USB port is in use for full- or low-speed communications, the TRNENSEL bit specifies the timing with which the USBFS issues tokens in a frame (transaction-enabled time).

Set this bit to 1 when a low-speed device is connected. The bit is only valid in host controller mode. Set this bit to 0 in device controller mode.

36.2.16 INTSTS0 : Interrupt Status Register 0

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x040

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ[2:0]			VALID	CTSQ[2:0]		
Value after reset:	0	0	0	x	0	0	0	0	x	0	0	x	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CTSQ[2:0]	Control Transfer Stage 0 0 0: Idle or setup stage 0 0 1: Control read data stage 0 1 0: Control read status stage 0 1 1: Control write data stage 1 0 0: Control write status stage 1 0 1: Control write (no data) status stage 1 1 0: Control transfer sequence error	R
3	VALID	USB Request Reception 0: Setup packet not received 1: Setup packet received	R/W
6:4	DVSQ[2:0]	Device State Indicates the device state. 0 0 0: Powered state 0 0 1: Default state 0 1 0: Address state 0 1 1: Configured state Others: Suspend state	R
7	VBSTS	VBUS Input Status 0: USB_VBUS pin is low 1: USB_VBUS pin is high	R
8	BRDY	Buffer Ready Interrupt Status 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R
9	NRDY	Buffer Not Ready Interrupt Status 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R
10	BEMP	Buffer Empty Interrupt Status 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R
11	CTRT	Control Transfer Stage Transition Interrupt Status*2 0: No control transfer stage transition interrupt occurred 1: Control transfer stage transition interrupt occurred	R/W*1

Bit	Symbol	Function	R/W
12	DVST	Device State Transition Interrupt Status*2 0: No device state transition interrupt occurred 1: Device state transition interrupt occurred	R/W*1
13	SOFR	Frame Number Refresh Interrupt Status 0: No SOF interrupt occurred 1: SOF interrupt occurred	R/W*1
14	RESM	Resume Interrupt Status*2 *3 0: No resume interrupt occurred 1: Resume interrupt occurred	R/W*1
15	VBINT	VBUS Interrupt Status*3 0: No VBUS interrupt occurred 1: VBUS interrupt occurred	R/W*1

Note: S-TYPE-3, P-TYPE-3

Note: The value of the DVST bit is 0 when the MCU is reset and 1 after a USB bus reset.

Note: The value of the VBSTS bit is 1 when the USB_VBUS pin is high and 0 when the USB_VBUS pin is low.

Note: The value of the DVSQ[2:0] bits is 000b when the MCU is reset and 001b after a USB bus reset.

Note 1. To clear the VBINT, RESM, SOFR, DVST, CTRT, or VALID bits, write 0 only to the bits to be cleared. Write 1 to the other bits. Do not write 0 to the status bits indicating 0.

Note 2. The status of the RESM, DVST, and CTRT bits are changed only in device controller mode. Set the associated interrupt enable bits to 0 (disabled) in host controller mode.

Note 3. The USBFS detects a change in the status indicated in the VBINT andRESM bits even while the clock supply is stopped (SYSCFG.SCKE bit = 0), and it requests the interrupt when the associated interrupt request bit is 1. Enable the clock supply before clearing the status by software.

CTSQ[2:0] bits (Control Transfer Stage)

In host controller mode, the read value of the CTSQ[2:0] bits is invalid.

VALID bit (USB Request Reception)

In host controller mode, the read value of the VALID bit is invalid.

DVSQ[2:0] bits (Device State)

The DVSQ[2:0] bits are initialized by a USB bus reset. In host controller mode, the read value is invalid.

BRDY flag (Buffer Ready Interrupt Status)

The BRDY flag indicates the BRDY interrupt status.

The USBFS sets the BRDY bit to 1 when it detects a BRDY interrupt status (PIPEnBRDY = 1, n = 0 to 9) on at least one pipe for which BRDY interrupts are enabled (BRDYENB.PIPEnBRDYE = 1).

For the conditions that cause the PIPEnBRDY status to be asserted, see section 36.3.3.1. BRDY Interrupt.

The USBFS sets the BRDY bit to 0 when the software writes 0 to all of the PIPEnBRDY bits associated with the PIPEnBRDYE bits that are set to 1. Writing 0 to the BRDY flag in the software does not clear the flag.

NRDY flag (Buffer Not Ready Interrupt Status)

The NRDY flag indicates the NRDY interrupt status.

The USBFS sets the NRDY bit to 1 when it detects a NRDY interrupt status (PIPEnNRDY = 1, n = 0 to 9) on at least one pipe for which NRDY interrupts are enabled (NRDYENB.PIPEnNRDYE = 1).

For the conditions that cause the PIPEnNRDY status to be asserted, see section 36.3.3.2. NRDY Interrupt.

The USBFS sets the NRDY bit to 0 when the software writes 0 to all of the PIPEnNRDY bits associated with the PIPEnNRDYE bits that are set to 1. Writing 0 to the NRDY flag in the software does not clear the flag.

BEMP flag (Buffer Empty Interrupt Status)

The BEMP flag indicates the BEMP interrupt status.

The USBFS sets the BEMP bit to 1 when it detects a BEMP interrupt status (PIPEnBEMP = 1, n = 0 to 9) on at least one pipe for which BEMP interrupts are enabled (BEMPENB.PIPEnBEMPE = 1).

For the conditions that cause the PIPEnBEMP status to be asserted, see section 36.3.3.3. BEMP Interrupt.

The USBFS sets the BEMP bit to 0 when the software writes 0 to all of the PIPEnBEMP bits associated with the PIPEnBEMPE bits that are set to 1. Writing 0 to the BEMP flag in the software does not clear the flag.

CTRT flag (Control Transfer Stage Transition Interrupt Status)

In device controller mode, the USBFS updates the value of the CTSQ[2:0] bits and sets the CTRT flag to 1 on detecting a transition in the control transfer stage. When a control transfer stage transition interrupt occurs, clear the CTRT flag before the USBFS detects the next control transfer stage transition.

Values read from the CTRT flag in host controller mode are invalid.

DVST flag (Device State Transition Interrupt Status)

In device controller mode, the USBFS updates the value of the DVSQ[2:0] bits and sets the DVST flag to 1 on detecting a change in the device state. When a device state transition interrupt occurs, clear theDVST flag before the USBFS detects the next device state transition.

Values read from the DVST flag in host controller mode are invalid.

SOFR flag (Frame Number Refresh Interrupt Status)

In host controller mode, the USBFS sets the SOFR flag to 1 on updating the frame number when the DVSTCTR0.UACT bit is set to 1 by software. A SOFR interrupt is detected every 1 ms.

In device controller mode, the USBFS sets the SOFR flag to 1 on updating the frame number. A frame number refresh interrupt is detected every 1 ms.

The USBFS can detect an SOFR interrupt through the internal interpolation function even when a corrupted SOF packet is received from the USB host.

RESM flag (Resume Interrupt Status)

In device controller mode, the USBFS sets the RESM flag to 1 on detecting the falling edge of the signal on the USB_DP pin in the Suspend state (DVSQ[2:0] = 1xxb). Values read from the RESM flag in host controller mode are invalid.

VBINT flag (VBUS Interrupt Status)

The USBFS sets the VBINT flag to 1 on detecting a level change (high to low or low to high) in the USB_VBUS pin input value. The USBFS sets the VBSTS flag to indicate the USB_VBUS pin input value. When a VBUS interrupt occurs, eliminate transient elements by reading the VBSTS flag at least three times through software processing and check that the values read are the same.

36.2.17 INTSTS1 : Interrupt Status Register 1

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x042

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OVRCR	BCHG	—	DTCH	ATTCH	—	—	—	—	EOFERR	SIGN	SACK	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	SACK	Setup Transaction Normal Response Interrupt Status 0: No SACK interrupt occurred 1: SACK interrupt occurred	R/W*1
5	SIGN	Setup Transaction Error Interrupt Status 0: No SIGN interrupt occurred 1: SIGN interrupt occurred	R/W*1
6	EOFERR	EOF Error Detection Interrupt Status 0: No EOFERR interrupt occurred 1: EOFERR interrupt occurred	R/W*1

Bit	Symbol	Function	R/W
10:7	—	These bits are read as 0. The write value should be 0.	R/W
11	ATTCH	ATTCH Interrupt Status 0: No ATTCH interrupt occurred 1: ATTCH interrupt occurred	R/W*1
12	DTCH	USB Disconnection Detection Interrupt Status 0: No DTCH interrupt occurred 1: DTCH interrupt occurred	R/W*1
13	—	This bit is read as 0. The write value should be 0.	R/W
14	BCHG	USB Bus Change Interrupt Status*2 0: No BCHG interrupt occurred 1: BCHG interrupt occurred	R/W*1
15	OVRCR	Overcurrent Input Change Interrupt Status*2 0: No OVRCR interrupt occurred 1: OVRCR interrupt occurred	R/W*1

Note: S-TYPE-3, P-TYPE-3

Note 1. To clear the bits in INTSTS1, write 0 only to the bits to be cleared. Write 1 to the other bits.

Note 2. The USBFS detects a change in the status in the OVRCR or BCHG bit even when the clock supply is stopped (SYSCFG.SCKE = 0), and it requests the interrupt when the associated interrupt request bit is 1. Enable the clock supply (SYSCFG.SCKE = 1) before clearing the status through the software. No other interrupts can be detected while the clock supply is stopped (SYSCFG.SCKE bit = 0).

INTSTS1 is used to confirm the status of each interrupt in host controller mode. Only enable the status change interrupts indicated in the bits in INTSTS1 in host controller mode.

SACK flag (Setup Transaction Normal Response Interrupt Status)

The SACK flag indicates the status of the setup transaction normal response interrupt in host controller mode.

The USBFS detects the SACK interrupt and sets this flag to 1 when an ACK response is returned from a peripheral device during the setup transactions issued by the USBFS. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

Values read from the SACK flag in device controller mode are invalid.

SIGN flag (Setup Transaction Error Interrupt Status)

The SIGN flag indicates the status of setup transaction error interrupts in host controller mode.

The USBFS detects the SIGN interrupt and sets this flag to 1 when an ACK response is not returned from a peripheral device three consecutive times during the setup transactions issued by the USBFS. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

The USBFS detects the SIGN interrupt when any of the following response conditions occur for three consecutive setup transactions:

- Timeout is detected by the USBFS when the peripheral device has returned no response
- A corrupted ACK packet is received
- A handshake other than ACK (NAK, NYET, or STALL) is received

Values read from the SIGN flag in device controller mode are invalid.

EOFERR flag (EOF Error Detection Interrupt Status)

The EOFERR flag indicates the status of EOF error detection interrupts in host controller mode.

The USBFS detects the EOFERR interrupt and sets this flag to 1 on detecting that communication did not complete at the EOF2 timing defined in the USB 2.0 specification. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

After detecting the EOFERR interrupt, the USBFS controls the hardware as follows, regardless of the associated interrupt enable bit setting:

- Sets the DVSTCTR0.UACT bit for the port in which the EOFERR interrupt was detected to 0
- Puts the port in which the EOFERR interrupt occurred into the idle state

The software must terminate all pipes in which communications are being carried out and re-enumerate the USB port.

Values read from the EOFERR flag in device controller mode are invalid.

ATTCH flag (ATTCH Interrupt Status)

The ATTCH flag indicates the status of USB attach detection interrupts in host controller mode.

The USBFS detects the ATTCH interrupt and sets this flag to 1 on detecting a J- or K-state on the full- or low-speed signal level for 2.5 µs. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

The USBFS detects the ATTCH interrupt on any of the following conditions.

- K-state, SE0, or SE1 changes to J-state, and J-state continues for 2.5 µs
- J-state, SE0, or SE1 changes to K-state, and K-state continues for 2.5 µs

Values read from the ATTCH flag in device controller mode are invalid.

DTCH flag (USB Disconnection Detection Interrupt Status)

The DTCH flag indicates the status of USB disconnection detection interrupts in host controller mode.

The USBFS detects the DTCH interrupt and sets this flag to 1 on detecting a USB bus detach event. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

The USBFS detects bus detach events based on the USB 2.0 specification.

After detecting the DTCH interrupt, the USBFS controls hardware as follows, regardless of the associated interrupt enable bit setting:

- Sets the DVSTCTR0.UACT bit for the port in which the DTCH interrupt was detected to 0
- Puts the port in which the DTCH interrupt occurred into the idle state

The software must terminate all pipes in which communications are being carried out and transition to a wait state for connecting to the USB port (waiting for ATTCH interrupt generation).

Values read from the DTCH flag in device controller mode are invalid.

BCHG flag (USB Bus Change Interrupt Status)

The BCHG flag indicates the status of USB bus change interrupts in host controller mode.

The USBFS detects the BCHG interrupt and sets this flag to 1 when a change in the full- or low-speed signal level occurs on the USB port. This includes any change from J-state, K-state, or SE0 to J-state, K-state, or SE0. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

The USBFS sets the LNST[1:0] flags to indicate the input state of the USB port. When a BCHG interrupt occurs, eliminate transient elements by repeat reading the LNST[1:0] flags by software until the same value is read at least three times.

Change in the USB bus state can be detected while the internal clock is stopped.

Values read from the BCHG flag in device controller mode are invalid.

OVRCR flag (Overcurrent Input Change Interrupt Status)

The OVRCR flag indicates the status of USB_OVRCURA, USB_OVRCURA-DS, USB_OVRCURB or USB_OVRCURB-DS input pin change interrupts.

The USBFS detects the OVRCR interrupt and sets this flag to 1 when a change (high to low or low to high) occurs in at least one of the input values to the USB_OVRCURA, USB_OVRCURA-DS, USB_OVRCURB or USB_OVRCURB-DS pins. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

36.2.18 BRDYSTS : BRDY Interrupt Status Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x046

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPE9 BRDY	PIPE8 BRDY	PIPE7 BRDY	PIPE6 BRDY	PIPE5 BRDY	PIPE4 BRDY	PIPE3 BRDY	PIPE2 BRDY	PIPE1 BRDY	PIPE0 BRDY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIPE0BRDY	BRDY Interrupt Status for Pipe 0 *2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
1	PIPE1BRDY	BRDY Interrupt Status for Pipe 1 *2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
2	PIPE2BRDY	BRDY Interrupt Status for Pipe 2*2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
3	PIPE3BRDY	BRDY Interrupt Status for Pipe 3*2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
4	PIPE4BRDY	BRDY Interrupt Status for Pipe 4 *2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
5	PIPE5BRDY	BRDY Interrupt Status for Pipe 5 *2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
6	PIPE6BRDY	BRDY Interrupt Status for Pipe 6*2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
7	PIPE7BRDY	BRDY Interrupt Status for Pipe 7 *2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
8	PIPE8BRDY	BRDY Interrupt Status for Pipe 8 *2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
9	PIPE9BRDY	BRDY Interrupt Status for Pipe 9 *2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. When the SOFCFG.BRDYM bit is set to 0, to clear the status indicated in the bits in BRDYSTS, write 0 only to the bits to be cleared. Write 1 to the other bits.

Note 2. When the SOFCFG.BRDYM bit is set to 0, clear BRDY interrupts before accessing the FIFO.

36.2.19 NRDYSTS : NRDY Interrupt Status Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x048

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPE9 NRDY	PIPE8 NRDY	PIPE7 NRDY	PIPE6 NRDY	PIPE5 NRDY	PIPE4 NRDY	PIPE3 NRDY	PIPE2 NRDY	PIPE1 NRDY	PIPE0 NRDY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIPE0NRDY	NRDY Interrupt Status for Pipe 0 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W*1
1	PIPE1NRDY	NRDY Interrupt Status for Pipe 1 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W*1
2	PIPE2NRDY	NRDY Interrupt Status for Pipe 2 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W*1
3	PIPE3NRDY	NRDY Interrupt Status for Pipe 3 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W*1
4	PIPE4NRDY	NRDY Interrupt Status for Pipe 4 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W*1
5	PIPE5NRDY	NRDY Interrupt Status for Pipe 5 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W*1
6	PIPE6NRDY	NRDY Interrupt Status for Pipe 6 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W*1
7	PIPE7NRDY	NRDY Interrupt Status for Pipe 7 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W*1
8	PIPE8NRDY	NRDY Interrupt Status for Pipe 8 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W*1
9	PIPE9NRDY	NRDY Interrupt Status for Pipe 9 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W*1
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. To clear the status indicated in the bits in NRDYSTS, write 0 only to the bits to be cleared. Write 1 to the other bits.

36.2.20 BEMPSTS : BEMP Interrupt Status Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x04A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPE9 BEMP	PIPE8 BEMP	PIPE7 BEMP	PIPE6 BEMP	PIPE5 BEMP	PIPE4 BEMP	PIPE3 BEMP	PIPE2 BEMP	PIPE1 BEMP	PIPE0 BEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIPE0BEMP	BEMP Interrupt Status for Pipe 0 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W*1
1	PIPE1BEMP	BEMP Interrupt Status for Pipe 1 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W*1
2	PIPE2BEMP	BEMP Interrupt Status for Pipe 2 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W*1
3	PIPE3BEMP	BEMP Interrupt Status for Pipe 3 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W*1
4	PIPE4BEMP	BEMP Interrupt Status for Pipe 4 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W*1
5	PIPE5BEMP	BEMP Interrupt Status for Pipe 5 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W*1
6	PIPE6BEMP	BEMP Interrupt Status for Pipe 6 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W*1
7	PIPE7BEMP	BEMP Interrupt Status for Pipe 7 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W*1
8	PIPE8BEMP	BEMP Interrupt Status for Pipe 8 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W*1
9	PIPE9BEMP	BEMP Interrupt Status for Pipe 9 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W*1
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. To clear the status indicated in the bits in BEMPSTS, write 0 only to the bits to be cleared. Write 1 to the other bits.

36.2.21 FRMNUM : Frame Number Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x04C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OVRN	CRCE	—	—	—	FRNM[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	FRNM[10:0]	Frame Number Latest frame number.	R
13:11	—	These bits are read as 0. The write value should be 0.	R/W
14	CRCE	Receive Data Error 0: No error occurred 1: Error occurred	R/W*1
15	OVRN	Overrun/Underrun Detection Status 0: No error occurred 1: Error occurred	R/W*1

Note: S-TYPE-3, P-TYPE-3
Note 1. To clear the status, write 0 only to the bits to be cleared. Write 1 to the other bits.

FRNM[10:0] flags (Frame Number)

The USBFS sets the FRNM[10:0] flags to indicate the latest frame number, which is updated every 1 ms, when an SOF packet is issued or received.

CRCE flag (Receive Data Error)

The CRCE flag sets to 1 when a CRC error or bit stuffing error occurs during isochronous transfer. On detecting a CRC error in host controller mode, the USBFS generates an internal NRDY interrupt.

To clear the CRCE flag, write 0 to it while writing 1 to the other bits in the FRMNUM register.

OVRN flag (Overrun/Underrun Detection Status)

The OVRN flag sets to 1 when an overrun or underrun error occurs during isochronous transfer. To clear the flag, write 0 to it while writing 1 to the other bits in the FRMNUM register.

In host controller mode, the OVRN flag sets to 1 on any of the following conditions:

- For a transmitting isochronous pipe, the time to issue an OUT token comes before all of the transmit data is written to the FIFO buffer
- For a receiving isochronous pipe, the time to issue an IN token comes when no FIFO buffer planes are empty

In device controller mode, the OVRN flag sets to 1 on any of the following conditions:

- For a transmitting isochronous pipe, the IN token is received before all of the transmit data is written to the FIFO buffer
- For a receiving isochronous pipe, the OUT token is received when no FIFO buffer planes are empty

36.2.22 DVCHGR : Device State Change Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x04E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DVCHG	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
14:0	—	These bits are read as 0. The write value should be 0.	R/W
15	DVCHG	Device State Change 0: Disable writes to the USBADDR.STSRECOV[3:0] and USBADDR.USBADDR[6:0] bits 1: Enable writes to the USBADDR.STSRECOV[3:0] and USBADDR.USBADDR[6:0] bits	R/W

Note: S-TYPE-3, P-TYPE-3

For details, see section 36.3.1.5. Release from Deep Software Standby Mode 1 Because of USB Suspend/Resume Interrupts.

36.2.23 USBADDR : USB Address Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x050

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	STSRECOV[3:0]				—	USBADDR[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	USBADDR[6:0]	USB Address In device controller mode, these bits indicate the USB address assigned by the host when the USBFS processed the SET_ADDRESS request successfully.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
11:8	STSRECOV[3:0]	Status Recovery 0x4: Recovery in device controller mode: Setting prohibited Recovery in host controller mode: Return to the low-speed state (bits DVSTCTR0.RHST[2:0] = 001b) 0x8: Recovery in device controller mode: Setting prohibited Recovery in host controller mode: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b) 0x9: Recovery in device controller mode: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b), bits INTSTS0.DVSQ[2:0] = 001b (default state) Recovery in host controller mode: Setting prohibited 0xA: Recovery in device controller mode: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b), bits INTSTS0.DVSQ[2:0] = 010b (address state) Recovery in host controller mode: Setting prohibited 0xB: Recovery in device controller mode: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b), bits INTSTS0.DVSQ[2:0] = 011b (configured state) Recovery in host controller mode: Setting prohibited Others: Setting prohibited	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

USBADDR[6:0] bits (USB Address)

In device controller mode, the USBADDR[6:0] flags indicate the USB address received when the USBFS processed a SetAddress request successfully. The USBFS sets the USBADDR[6:0] bits to 0x00 on detecting a USB bus reset.

Writing to these bits is enabled while the DVCHGR.DVCHG bit is set to 1. On recovering from a USB power shut-off, the operation can resume from the USB address set before the software shut-off.

In host controller mode, the USBADDR[6:0] bits are invalid.

STSRECOV[3:0] bits (Status Recovery)

Use the STSRECOV[3:0] bits to resume the state of the internal sequencer on recovering from USB power shut-off. For details, see section 36.3.1.5. Release from Deep Software Standby Mode 1 Because of USB Suspend/Resume Interrupts.

Writing to these bits is enabled while the DVCHGR.DVCHG bit is set to 1.

36.2.24 USBREQ : USB Request Type Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x054

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BREQUEST[7:0]								BMREQUESTTYPE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	BMREQUESTTYPE [7:0]	Request Type USB request bmRequestType value	R/W*1
15:8	BREQUEST[7:0]	Request USB request bRequest value	R/W*1

Note: S-TYPE-3, P-TYPE-3

Note 1. In device controller mode, these bits can be read, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

USBREQ stores setup requests for control transfers.

In device controller mode, the USBREQ stores the received bRequest and bmRequestType values. In host controller mode, it sets to the bRequest and bmRequestType values to be transmitted.

USBREQ is initialized by a USB bus reset.

BMREQUESTTYPE[7:0] bits (Request Type)

The BMREQUESTTYPE[7:0] bits hold the bmRequestType value of USB requests.

- In host controller mode:

  Set these bits to the value of the USB request data in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.

- In device controller mode:

  These bits indicate the value of the USB request data in reception setup transactions. Writing to the bits has no effect.

BREQUEST[7:0] bits (Request)

The BREQUEST[7:0] bits store bRequest value of the USB request.

- In host controller mode:

  Set these bits to the value of the USB request data in setup transmission transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.

- In device controller mode:

  These bits indicate the value of the USB request data in reception setup transactions. Writing to the bits has no effect.

36.2.25 USBVAL : USB Request Value Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x056

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WVALUE[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	WVALUE[15:0]	Value USB request wValue value	R/W*1

Note: S-TYPE-3, P-TYPE-3

Note 1. In device controller mode, these bits can be read, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

In device controller mode, USBVAL stores the received wValue value. In host controller mode, it sets to the wValue value to be transmitted is set.

USBVAL is initialized by a USB bus reset.

WVALUE[15:0] bits (Value)

The WVALUE[15:0] bits store wValue value of the USB request.

- In host controller mode:

  Set these bits to the value of the wValue field in USB requests of transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.

- In device controller mode:

These bits indicate the wValue value of USB requests in reception setup transactions. Writing to the bits has no effect.

36.2.26 USBINDX : USB Request Index Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x058

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WINDEX[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	WINDEX[15:0]	Index USB request wIndex value	R/W*1

Note: S-TYPE-3, P-TYPE-3

Note 1. In device controller mode, these bits can be read, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

USBINDX stores setup requests for control transfers.

In device controller mode, it stores the received wIndex value. In host controller mode, it sets to the wIndex value to be transmitted.

USBINDX is initialized by a USB bus reset.

WINDEX[15:0] bits (Index)

The WINDEX[15:0] bits hold the wIndex value of a USB request.

- In host controller mode:

  Set these bits to the wIndex value in USB requests in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.

- In device controller mode:

  These bits indicate the wIndex value in USB requests received in reception setup transactions. Writing to the bits has no effect.

36.2.27 USBLENG : USB Request Length Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x05A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WLENTUH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	WLENTUH[15:0]	Length USB request wLength value	R/W*1

Note: S-TYPE-3, P-TYPE-3

Note 1. In device controller mode, these bits can be read, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

USBLENG stores setup requests for control transfers.

In device controller mode, the value of wLength that is received is stored. In host controller mode, the value of wLength to be transmitted is set.

USBLENG is initialized by a USB bus reset.

WLENTUH[15:0] bits (Length)

The WLENTUH[15:0]bits hold the wLength value of a USB request.

- In host controller mode:

  Set these bits to the wLength value in USB requests in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.

- In device controller mode:

  These bits indicate the wLength value in USB requests received in reception setup transactions. Writing to the bits has no effect.

36.2.28 DCPCFG : DCP Configuration Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x05C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	SHTNAK	—	—	DIR	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	DIR	Transfer Direction*1 0: Data receiving direction 1: Data transmitting direction	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	SHTNAK	Pipe Disabled at End of Transfer*1 0: Keep pipe open after transfer ends 1: Disable pipe after transfer ends	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Only set this bit while the PID is NAK. Before setting this bit, check that the DCPCTR.PBUSY bit is 0, and then change the DCPCTR.PID[1:0] bits for the DCP from BUF to NAK. If the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through the software is not necessary.

DIR bit (Transfer Direction)

In host controller mode, the DIR bit sets the transfer direction of the data stage and status stage for control transfers. In device controller mode, set the DIR bit to 0.

SHTNAK bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to change PID to NAK on transfer end when the selected pipe is receiving. It is only valid when the selected pipe is receiving.

When the SHTNAK bit is 1, the USBFS changes the DCPCTR.PID[1:0] bits for the DCP to NAK on determining that a transfer has ended. The USBFS determines transfer end on the following condition:

- A short packet, including a zero-length packet, is successfully received.

36.2.29 DCPMAXP : DCP Maximum Packet Size Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x05E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DEVSEL[3:0]				—	—	—	—	—	MXPS[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	MXPS[6:0]	Maximum Packet Size*1 Maximum data payload specification (maximum packet size) for the DCP	R/W
11:7	—	These bits are read as 0. The write value should be 0.	R/W
15:12	DEVSEL[3:0]	Device Select*2 0x0: Address 0000b 0x1: Address 0001b 0x2: Address 0010b 0x3: Address 0011b 0x4: Address 0100b 0x5: Address 0101b Others: Setting prohibited	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Only set the MXPS[6:0] bits while PID is NAK. Before setting these bits, check that the DCPCTR.PBUSY bit is 0, and then change the DCPCTR.PID[1:0] bits for the DCP from BUF to NAK. If the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through the software is not necessary. After the MXPS[6:0] bits are set and the DCP is set to the CURPIPE[3:0] bits in a port select register, clear the buffer by setting the BCLR bit in the port control register to 1.

Note 2. Only set the DEVSEL[3:0] bits while PID is NAK and the DCPCTR.SUREQ bit is 0. Before setting these bits, check that the DCPCTR.PBUSY bit is 0, and then change the DCPCTR.PID[1:0] bits for the DCP from BUF to NAK. If the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through the software is not necessary.

MXPS[6:0] bits (Maximum Packet Size)

The MXPS[6:0] bits specify the maximum data payload (maximum packet size) for the DCP. The initial value is 0x40 (64 bytes). Set the bits to a USB 2.0-compliant value. Do not write to the FIFO buffer or set PID = BUF whileMXPS[6:0] is set to 0.

DEVSEL[3:0] bits (Device Select)

In host controller mode, the DEVSEL[3:0] bits specify the address of the target peripheral device for a control transfer. Set up the associated DEVADDn (n = 0 to 5) register first, and then set these bits to the corresponding value. To set the DEVSEL[3:0] bits to 0010b, for example, first set the address in the DEVADD2 register.

In device controller mode, set these bits to 0000b.

36.2.30 DCPCTR : DCP Control Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x060

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSTS	SUREQ	—	—	SUREQCLR	—	—	SQCLR	SQSET	SQMON	PBUSY	—	—	CCPL	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PID[1:0]	Response PID 0 0: NAK response 0 1: BUF response (depends on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
2	CCPL	Control Transfer End Enable 0: Disable control transfer completion 1: Enable control transfer completion	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	PBUSY	Pipe Busy 0: DCP not used for the USB bus 1: DCP in use for the USB bus	R
6	SQMON	Sequence Toggle Bit Monitor 0: DATA0 1: DATA1	R
7	SQSET	Sequence Toggle Bit Set*2 Sets the sequence toggle bit in DCP transfers. 0: Invalid (writing 0 has no effect) 1: Set the expected value for the next transaction to DATA1	R/W*1
8	SQCLR	Sequence Toggle Bit Clear*2 Clears the sequence toggle bit in DCP transfers. 0: Invalid (writing 0 has no effect) 1: Clear the expected value for the next transaction to DATA0	R/W*1
10:9	—	These bits are read as 0. The write value should be 0.	R/W
11	SUREQCLR	SUREQ Bit Clear Clears the SUREQ bit in host controller mode. 0: Invalid (writing 0 has no effect) 1: Clear SUREQ to 0	R/W*1
13:12	—	These bits are read as 0. The write value should be 0.	R/W
14	SUREQ	Setup Token Transmission Sets up token transmission in host controller mode. 0: Invalid (writing 0 has no effect) 1: Transmit setup packet	R/W
15	BSTS	Buffer Status 0: Buffer access disabled 1: Buffer access enabled	R

Note: S-TYPE-3, P-TYPE-3
Note 1. This bit is read as 0.
Note 2. Only set the SQSET and SQCLR bits while PID is NAK. Before setting these bits, check that the PBUSY bit is 0, and then change the PID[1:0] bits for the DCP from BUF to NAK. If the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through the software is not necessary.

PID[1:0] bits (Response PID)

The PID[1:0] bits control the USB response type during control transfers.

In host controller mode, to change the PID[1:0] setting from NAK to BUF:

- When the transmitting direction is set:
  a. Write all of the transmit data to the FIFO buffer while the DVSTCTR0.UACT bit is 1 and PID is NAK.
  b. Set PID[1:0] bits to 01b (BUF).
     The USBFS then executes the OUT transaction.
- When the receiving direction is set:
  a. Check that the FIFO buffer is empty (or empty the buffer) while the DVSTCTR0.UACT bit is 1 and PID is NAK.
  b. Set PID[1:0] bits to 01b (BUF).
     The USBFS then executes the IN transaction.

The USBFS changes the PID[1:0] setting as follows:

- When the PID[1:0] bits are set to BUF (01b) by software and the USBFS has received data exceeding MaxPacketSize, the USBFS sets the PID[1:0] to STALL (11b)
- When a reception error, such as a CRC error, is detected three times consecutively, the USBFS sets thePID[1:0] bits to NAK (00b)
- On receiving the STALL handshake, the USBFS sets PID[1:0] to STALL (11b)

In device controller mode, the USBFS changes the PID[1:0] setting as follows:

- On receiving a setup packet, the USBFS sets PID[1:0] to NAK (00b). The USBFS then sets the INTSTS0.VALID flag to 1, and the PID[1:0] setting cannot be changed until the software clears the VALID flag to 0.
- When the PID[1:0] bits are set to BUF (01b) by software and the USBFS has received data exceeding MaxPacketSize, the USBFS sets PID[1:0] to STALL (11b)
- On detecting a control transfer sequence error, the USBFS sets PID[1:0] to STALL (1xb)
- On detecting a USB bus reset, the USBFS sets PID[1:0] to NAK

The USBFS does not check the PID[1:0] setting while processing a SET_ADDRESS request.

The PID[1:0] bits are initialized by a USB bus reset.

CCPL bit (Control Transfer End Enable)

In device controller mode, setting the CCPL bit to 1 enables the status stage of the control transfer to be completed. When the bit is set to 1 by software while the associated PID[1:0] bits are set to BUF, the USBFS completes the control transfer status stage.

During control read transfers, the USBFS transmits the ACK handshake in response to the OUT transaction from the USB host. During control write or no-data control transfers, it transmits the zero-length packet in response to the IN transaction from the USB host. On detecting a SET_ADDRESS request, the USBFS operates in auto response mode from the setup stage up to status stage completion regardless of the CCPL bit setting.

The USBFS changes the CCPL bit from 1 to 0 on receiving a new setup packet. The software cannot write 1 to the bit while the INTSTS0.VALID bit is 1. The bit is initialized by a USB bus reset.

In host controller mode, always write 0 to the CCPL bit.

PBUSY bit (Pipe Busy)

The PBUSY bit indicates whether DCP is used for the transaction when USBFS changes the PID[1:0] bits from BUF to NAK. The USBFS changes the PBUSY bit from 0 to 1 on start of a USB transaction for the selected pipe. It changes the PBUSY bit from 1 to 0 on completion of one transaction.

After PID is set to NAK by software, the value in the PBUSY bit indicates whether changes to pipe settings can proceed.

For details, see section 36.3.4.1. Pipe Control Register Switching Procedures.

SQMON bit (Sequence Toggle Bit Monitor)

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction during a DCP transfer.

The USBFS toggles the bit on normal completion of the transaction. It does not toggle the bit, however, when a DATAPID mismatch occurs during a transfer in the receiving direction.

In device controller mode, the USBFS sets the SQMON bit to 1 (specifies DATA1 as the expected value) on successful reception of the setup packet.

In device controller mode, the USBFS does not reference this bit during IN or OUT transactions at the status stage, and it does not toggle the bit on normal completion.

SQSET bit (Sequence Toggle Bit Set)

The SQSET bit specifies DATA1 as the expected value of the sequence toggle bit for the next transaction during a DCP transfer.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

SQCLR bit (Sequence Toggle Bit Clear)

The SQCLR bit specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during a DCP transfer. It is read as 0.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

SUREQCLR bit (SUREQ Bit Clear)

In host controller mode, setting the SUREQCLR bit to 1 clears the SUREQ bit to 0. The bit is read as 0.

If transfer stops while the SUREQ bit is set to 1 in a setup transaction, set the SUREQCLR bit to 1 by software. This is not necessary at the end of a normal setup transaction, because the USBFS automatically clears the SUREQ bit to 0.

Only control the SUREQ bit through the SUREQCLR bit while the DVSTCTR0.UACT bit is 0. When UACT is 0, communication is halted or no transfer is occurring because a bus disconnection was detected.

In device controller mode, always write 0 to this bit.

SUREQ bit (Setup Token Transmission)

In host controller mode, setting the SUREQ bit to 1 triggers the USBFS to transmit the setup packet. After completing the setup transaction process, the USBFS generates either the SACK or SIGN interrupt and clears the SUREQ bit to 0. The USBFS also clears the SUREQ bit to 0 when the software sets the SUREQCLR bit to 1.

Before setting the SUREQ bit to 1, set the DCPMAXP.DEVSEL[3:0] bits, USBREQ, USBVAL, USBINDX, and USBLENG appropriately to transmit the target USB request in the setup transaction. Also check that the PID[1:0] bits for the DCP are set to NAK. After setting the SUREQ bit to 1, do not change the DCPMAXP.DEVSEL[3:0] bits, USBREQ, USBVAL, USBINDX, or USBLENG until the setup transaction is complete (SUREQ bit = 1). Write 1 to the SUREQ bit only when transmitting the setup token. Otherwise, write 0.

In device controller mode, always write 0 to this bit.

BSTS flag (Buffer Status)

The BSTS flag indicates the status of access to the DCP FIFO buffer. The meaning of this flag varies as follows depending on the CFIFOSEL.ISEL setting:

- When ISEL = 0, the bit indicates whether receive data can be read from the buffer
- When ISEL = 1, the bit indicates whether transmit data can be written to the buffer

36.2.31 PIPESEL : Pipe Window Select Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x064

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	PIPESEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	PIPESEL[3:0]	Pipe Window Select 0x0: No pipe selected 0x1: Pipe 1 0x2: Pipe 2 0x3: Pipe 3 0x4: Pipe 4 0x5: Pipe 5 0x6: Pipe 6 0x7: Pipe 7 0x8: Pipe 8 0x9: Pipe 9 Others: Setting prohibited	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Set pipes 1 to 9 using the PIPESEL, PIPECFG, PIPEMAXP, PIPEPERI, PIPEnCTR, PIPEnTRE, and PIPEnTRN registers (n = 1 to 9).

After selecting the pipe in the PIPESEL register, pipe functions must be set in the associated PIPECFG, PIPEMAXP, and PIPEPERI registers. PIPEnCTR, PIPEnTRE, and PIPEnTRN can be set independently of the pipe selection in this register.

PIPESEL[3:0] bits (Pipe Window Select)

The PIPESEL[3:0] bits select the pipe number associated with the PIPECFG, PIPEMAXP, and PIPEPERI registers used for data writing and reading. Selecting a pipe number in the PIPESEL[3:0] bits allows writing to and reading from PIPECFG, PIPEMAXP, and PIPEPERI associated with the selected pipe number.

When PIPESEL[3:0] = 0000b, 0 is read from all of the bits in PIPECFG, PIPEMAXP, and PIPEPERI. Writing to these bits has no effect.

36.2.32 PIPECFG : Pipe Configuration Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x068

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TYPE[1:0]		—	—	—	BFRE	DBLB	—	SHTNAK	—	—	DIR	EPNUM[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	EPNUM[3:0]	Endpoint Number*1 Specifies the endpoint number for the selected pipe. Setting 0000b indicates that the pipe is not used.	R/W
4	DIR	Transfer Direction*2 *3 0: Receiving direction 1: Transmitting direction	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	SHTNAK	Pipe Disabled at End of Transfer*1 0: Continue pipe operation after transfer ends 1: Disable pipe after transfer ends	R/W
8	—	This bit is read as 0. The write value should be 0.	R/W
9	DBLB	Double Buffer Mode*2 *3 0: Single buffer 1: Double buffer	R/W
10	BFRE	BRDY Interrupt Operation Specification*2 *3 0: Generate BRDY interrupt on transmitting or receiving data 1: Generate BRDY interrupt on completion of reading data	R/W
13:11	—	These bits are read as 0. The write value should be 0.	R/W
15:14	TYPE[1:0]	Transfer Type*1 0 0: Pipe not used 0 1: Pipes 1 and 2: Bulk transfer Pipes 3 to 5: Bulk transfer Pipes 6 to 9: Setting prohibited 1 0: Pipes 1 and 2: Setting prohibited Pipes 3 to 5: Setting prohibited Pipes 6 to 9: Interrupt transfer 1 1: Pipes 1 and 2: Isochronous transfer Pipes 3 to 5: Setting prohibited Pipes 6 to 9: Setting prohibited	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Only set the TYPE[1:0], SHTNAK, and EPNUM[3:0] bits while PID is NAK. Before setting these bits, check that the PIPEnCTR.PBUSY bit is 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

Note 2. Only set the BFRE, DBLB, and DIR bits while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting these bits, check that the PIPEnCTR.PBUSY bit is 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

Note 3. To change the BFRE, DBLB, or DIR bits after completing USB communication on the selected pipe, in addition to the constraints described in Note 2, write 1 and 0 to the PIPEnCTR.ACLRM bit continuously through the software and clear the FIFO buffer assigned to the pipe.

PIPECFG specifies the transfer type, FIFO buffer access direction, and endpoint numbers for pipes 1 to 9. It also selects single or double buffer mode, and whether to continue or disable pipe operation at the end of transfer.

EPNUM[3:0] bits (Endpoint Number)

The EPNUM[3:0] bits specify the endpoint number for the selected pipe. Setting 0000b indicates the pipe not used.

Set these bits so that the combination of the DIR and EPNUM[3:0] settings is different from those for other pipes. The EPNUM[3:0] bits can be set to 0000b for all pipes.

DIR bit (Transfer Direction)

The DIR bit specifies the transfer direction for the selected pipe.

When the software sets this bit to 0, the USBFS uses the selected pipe for receiving. When the software sets this bit to 1, the USBFS uses the selected pipe for transmitting.

SHTNAK bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to change the PIPEnCTR.PID[1:0] bits to 00b (NAK) at the end of transfer when the selected pipe is set in the receiving direction. The bit is valid for pipes 1 to 5 in the receiving direction.

When the software sets this bit to 1 for a receiving pipe, the USBFS changes the associated PIPEnCTR.PID[1:0] bits to 00b (NAK) on determining the transfer end. The USBFS determines that the transfer has ended on the following conditions:

- A short packet data (including a zero-length packet) was successfully received
- The transaction counter is used and the number of packets specified for the transaction counter are successfully received

DBLB bit (Double Buffer Mode)

The DBLB bit selects either single or double buffer mode for the FIFO buffer used by the selected pipe. The bit is valid for pipes 1 to 5.

BFRE bit (BRDY Interrupt Operation Specification)

The BFRE bit specifies the BRDY interrupt generation timing from the USBFS to the CPU for the selected pipe.

When the software sets the BFRE bit to 1 and the selected pipe is in the receiving direction, the USBFS detects the transfer completion and generates the BRDY interrupt on reading the packet.

When a BRDY interrupt is generated with this setting, the software must write 1 to the BCLR bit in the port control register. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to the BCLR bit.

When the BFRE bit is set to 1 by software and the selected pipe is in the transmitting direction, the USBFS does not generate the BRDY interrupt. For details, see section 36.3.3.1. BRDY Interrupt.

TYPE[1:0] bits (Transfer Type)

The TYPE[1:0] bits specify the transfer type for the pipe selected in the PIPESEL.PIPESEL[3:0] bits. Before setting PID to BUF and starting USB communication on the selected pipe, set the TYPE[1:0] bits to a value other than 00b.

36.2.33 PIPEMAXP : Pipe Maximum Packet Size Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x06C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DEVSEL[3:0]				—	—	—	MXPS[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	MXPS[8:0]	Maximum Packet Size*1 ● Pipes 1 and 2 1 byte (0x001) to 256 bytes (0x100) (Bit [9] not supported.) ● Pipes 3 to 5 8 bytes (0x008), 16 bytes (0x010), 32 bytes (0x020), 64 bytes (0x040) (Bits [9:7] and [2:0] not supported.) ● Pipes 6 to 9 1 byte (0x001) to 64 bytes (0x040) (Bits [9:7] not supported.)	R/W
11:9	—	These bits are read as 0. The write value should be 0.	R/W
15:12	DEVSEL[3:0]	Device Select*2 0x0: Address 0000b 0x1: Address 0001b 0x2: Address 0010b 0x3: Address 0011b 0x4: Address 0100b 0x5: Address 0101b Others: Setting prohibited	R/W

Note: S-TYPE-3, P-TYPE-3

Note: The value of the MXPS[8:0] bits is 0x000 when no pipe is selected in the PIPESEL.PIPESEL[3:0] bits and 0x040 when a pipe is selected.

Note 1. Only set the MXPS[8:0] bits while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting these bits, check that the PIPEnCTR.PBUSY bit is 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

Note 2. Only set the DEVSEL[3:0] bits while PID is NAK. Before setting these bits, check that the PIPEnCTR.PBUSY bit is 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00b (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

PIPEMAXP specifies the maximum packet size for pipes 1 to 9.

MXPS[8:0] bits (Maximum Packet Size)

The MXPS[8:0] bits specify the maximum data payload (maximum packet size) for the selected pipe.

Set these bits to the appropriate value for each transfer type based on the USB 2.0 specification. When MXPS[8:0] = 0, do not write to the FIFO buffer or set PID to BUF. These writes have no effect.

DEVSEL[3:0] bits (Device Select)

In host controller mode, the DEVSEL[3:0] bits specify the address of the target device for USB communication. Set up the device address in the associated DEVADDn (n = 0 to 5) register first, and then set these bits to the corresponding value. To set the DEVSEL[3:0] bits to 0x2, for example, first set the address in the DEVADD2 register.

In device controller mode, set these bits to 0x0.

36.2.34 PIPEPERI : Pipe Cycle Control Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x06E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	IFIS	—	—	—	—	—	—	—	—	—	IITV[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	IITV[2:0]*1	Interval Error Detection Interval Specifies the interval error detection timing for the selected pipe as the n-th power of 2 of the frame timing	R/W
11:3	—	These bits are read as 0. The write value should be 0.	R/W
12	IFIS	Isochronous IN Buffer Flush 0: Do not flush buffer 1: Flush buffer	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Only set the IITV[2:0] bits while PID is NAK. Before setting these bits, check that the PBUSY bit is 0, and then change the PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

PIPEPERI selects whether the buffer is flushed or not when an interval error occurred during isochronous IN transfers, and sets the interval error detection interval for pipes 1 to 9.

IITV[2:0] bits (Interval Error Detection Interval)

To change the IITV[2:0] bits to another value after they are set and USB communication is performed, set the PIPEnCTR.PID[1:0] bits to 00b (NAK) and then set the PIPEnCTR.ACLRM bit to 1 to initialize the interval timer.

The IITV[2:0] bits are not provided for pipes 3 to 5. Write 000b to bit positions of the IITV[2:0] bits associated with pipes 3 to 5.

IFIS bit (Isochronous IN Buffer Flush)

The IFIS bit specifies whether to flush the buffer when the pipe selected in the PIPESEL.PIPESEL[3:0] bits is used for isochronous IN transfers.

In device controller mode when the selected pipe is for isochronous IN transfers, the USBFS automatically clears the FIFO buffer if the USBFS fails to receive the IN token from the USB host within the interval set in theIITV[2:0] bits in terms of frames.

When double buffering is specified (PIPECFG.DBLB = 1), the USBFS only clears the data in the previously used plane.

The USBFS clears the FIFO buffer on receiving the SOF packet immediately after the frame in which the USBFS expected to receive the IN token. Even if the SOF packet is corrupted, the FIFO buffer is cleared at the time the SOF packet is expected to be received by using the internal interpolation function.

When the host controller function is selected, set this bit to 0.

Set this bit to 0 when the selected pipe is not for isochronous transfer.

36.2.35 PIPEnCTR : PIPEn Control Registers (n = 1 to 5)

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x070 + 0x2 × (n - 1)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PID[1:0]	Response PID 0 0: NAK response 0 1: BUF response (depends buffer state) 1 0: STALL response 1 1: STALL response	R/W
4:2	—	These bits are read as 0. The write value should be 0.	R/W
5	PBUSY	Pipe Busy 0: Pipe n not in use for the transaction 1: Pipe n in use for the transaction	R
6	SQMON	Sequence Toggle Bit Confirmation 0: DATA0 1: DATA1	R
7	SQSET	Sequence Toggle Bit Set*2 Sets the sequence toggle bit for pipe n. 0: Invalid (writing 0 has no effect) 1: Set the expected value for the next transaction to DATA1	R/W*1
8	SQCLR	Sequence Toggle Bit Clear*2 Clears the sequence toggle bit for pipe n. 0: Invalid (writing 0 has no effect) 1: Clear the expected value for the next transaction to DATA0	R/W*1
9	ACLRM	Auto Buffer Clear Mode*3 0: Disable 1: Enable (initialize all buffers)	R/W
10	ATREPM	Auto Response Mode*2 0: Disable auto response mode 1: Enable auto response mode	R/W
13:11	—	These bits are read as 0. The write value should be 0.	R/W
14	INBUFM	Transmit Buffer Monitor 0: No data to be transmitted is in the FIFO buffer 1: Data to be transmitted is in the FIFO buffer	R
15	BSTS	Buffer Status 0: Buffer access by the CPU disabled 1: Buffer access by the CPU enabled	R

Note: S-TYPE-3, P-TYPE-3
Note 1. This bit is read as 0.
Note 2. Only set the ATREPM bit or write 1 to the SQCLR or SQSETbit while PID is NAK. Before setting these bits, check that the PBUSY bit is 0, and then change the PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSYbit through the software is not necessary.
Note 3. Only set the ACLRM bit while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting this bit, check that the PBUSY bit is 0, and then change the PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

PIPEnCTR can be set for any pipe selection in the PIPESEL register.

PID[1:0] bits (Response PID)

The PID[1:0]bits specify the response type for the next transaction on the selected pipe.

The default PID[1:0] setting is NAK. Change the PID[1:0] setting to BUF to use the associated pipe for USB transfer. Table 36.9 and Table 36.10 show the basic operations of the USBFS (when there are no errors in the communication packets) based on the PID[1:0] bit setting.

After changing the PID[1:0] setting from BUF to NAK through the software during USB communication on the selected pipe, check that the PBUSY bit is 1 to see if USB transfer on the pipe has actually entered the NAK state. If the USBFS changes the PID[1:0] bits to NAK, checking the PBUSY bit through the software is not necessary.

The USBFS changes the PIPEnCTR.PID[1:0] setting in the following cases:

- The USBFS sets PID to NAK on recognizing completion of the transfer when the selected pipe is in the receiving direction and the PIPECFG.SHTNAK bit for the selected pipe is set to 1 by software
- The USBFS sets PID to STALL (11b) on receiving a data packet with a payload exceeding the maximum packet size of the selected pipe
- The USBFS sets PID to NAK on detecting a USB bus reset in device controller mode
- The USBFS sets PID to NAK on detecting a reception error, such as a CRC error, three consecutive times in host controller mode
- The USBFS sets PID to STALL (11b) on receiving the STALL handshake in host controller mode

To specify the response type, set the PID[1:0] bits as follows:

- To transition from NAK (00b) to STALL, set 10b
- To transition from BUF (01b) to STALL, set 11b
- To transition from STALL (11b) to NAK, set 10b and then 00b
- To transition from STALL to BUF, set 00b (NAK) and then 01b (BUF)

Table 36.9 Operation of the USBFS based on the PID[1:0] setting in host controller mode

PID[1:0] value	Transfer type	Transfer direction (DIR bit)	USBFS operation
00b (NAK)	Does not depend on the setting	Does not depend on the setting	Does not issue tokens
01b (BUF)	Bulk or interrupt	Does not depend on the setting	Issues tokens when the DVSTCTR0.UACT bit is 1 and the FIFO buffer associated with the selected pipe is ready for transmission and reception. Does not issue tokens when the DVSTCTR0.UACT bit is 0 or the FIFO buffer associated with the selected pipe is not ready for transmission or reception.
	Isochronous	Does not depend on the setting	Issues tokens regardless of the status of the FIFO buffer associated with the selected pipe.
10b (STALL) or 11b (STALL)	Does not depend on the setting	Does not depend on the setting	Does not issue tokens.

Table 36.10 Operation of the USBFS based on the PID[1:0] setting in device controller mode (1 of 2)

PID[1:0] value	Transfer type	Transfer direction (DIR bit)	USBFS operation
00b (NAK)	Bulk or interrupt	Does not depend on the setting	Returns NAK in response to the token from the USB host
	Isochronous	Does not depend on the setting	Returns nothing in response to the token from the USB host

Table 36.10 Operation of the USBFS based on the PID[1:0] setting in device controller mode (2 of 2)

PID[1:0] value	Transfer type	Transfer direction (DIR bit)	USBFS operation
01b (BUF)	Bulk	Receiving direction (DIR = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception
	Interrupt	Receiving direction (DIR = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception
	Bulk or interrupt	Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the FIFO buffer associated with the selected pipe is ready for transmission. Otherwise, returns NAK.
	Isochronous	Receiving direction (DIR = 0)	Receives data in response to the OUT token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception. Otherwise, discards the data.
	Isochronous	Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the associated FIFO buffer is ready for transmission. Otherwise, transmits a zero-length packet.
10b (STALL) or 11b (STALL)	Bulk or interrupt	Does not depend on the setting	Returns STALL in response to the token from the USB host
	Isochronous	Does not depend on the setting	Returns nothing in response to the token from the USB host

PBUSY bit (Pipe Busy)

The PBUSY bit indicates whether the selected pipe is being used for the current transaction.

The USBFS changes the PBUSY bit from 0 to 1 on start of the USB transaction for the selected pipe, and changes the PBUSY bit from 1 to 0 on completion of one transaction.

Reading the PBUSY bit by software after PID is set to NAK allows you to check whether changing the pipe setting is possible. For details, see section 36.3.4.1. Pipe Control Register Switching Procedures.

SQMON bit (Sequence Toggle Bit Confirmation)

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction of the selected pipe.

When the selected pipe is not the isochronous transfer type, the USBFS toggles the SQMON flag on successful completion of the transaction. However, the USBFS does not toggle the SQMON flag when a DATA-PID mismatch occurs during transfer in the receiving direction.

SQSET bit (Sequence Toggle Bit Set)

Setting the SQSET bit to 1 through the software causes the USBFS to set DATA1 as the expected value of the sequence toggle bit for the next transaction on the selected pipe. The USBFS clears the SQSET bit to 0.

SQCLR bit (Sequence Toggle Bit Clear)

Setting the SQCLR bit to 1 through the software causes the USBFS to clear the expected value of the sequence toggle bit for the next transaction on the selected pipe to DATA0. The USBFS clears the SQCLR bit to 0.

ACLRM bit (Auto Buffer Clear Mode)

The ACLRM bit enables or disables auto buffer clear mode for the selected pipe. To completely clear the data in the FIFO buffer allocated to the selected pipe, write 1 and then 0 to the ACLRM bit continuously.

Table 36.11 shows the data cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which this processing is required.

Table 36.11 Data cleared by the USBFS when ACLRM = 1 (1 of 2)

Number	Data cleared by setting the ACLRM bit	Situations requiring data clear
1	All data in the FIFO buffer allocated to the selected pipe (two FIFO buffers in double buffer mode)	When initializing the selected pipe

Table 36.11 Data cleared by the USBFS when ACLRM = 1 (2 of 2)

Number	Data cleared by setting the ACLRM bit	Situations requiring data clear
2	Interval count value when the selected pipe is the isochronous transfer type	When resetting the interval count value
3	Internal flags related to the PIPECFG.BFRE bit	When changing the PIPECFG.BFRE setting
4	FIFO buffer toggle control	When changing the PIPECFG.DBLB setting
5	Internal flags related to the transaction count	When forcing the transaction count function to terminate

ATREPM bit (Auto Response Mode)

The ATREPM bit enables or disables auto response mode for the selected pipe.

This bit can be set to 1 in device controller mode when the selected pipe is the bulk transfer type. When the bit is set to 1, the USBFS responds to the token from the USB host as follows:

- When the selected pipe is set for bulk IN transfers (PIPECFG.TYPE[1:0] = 01b and PIPECFG.DIR = 1):
  a. When the ATREPM bit = 1 and PID = BUF, the USBFS transmits a zero-length packet in response to the IN token.
  b. The USBFS updates the sequence toggle bit (DATA-PID) each time the USBFS receives ACK from the USB host. In a single transaction, the IN token is received, a zero-length packet is transmitted, and then ACK is received. The USBFS does not generate the BRDY or BEMP interrupt.

- When the selected pipe is set for bulk OUT transfers (PIPECFG.TYPE[1:0] = 01b and PIPECFG.DIR = 0):

  When the ATREPM bit = 1 and PID = BUF, the USBFS returns NAK in response to the OUT token and generates an NRDY interrupt.

For USB communication in auto response mode, set the ATREPM bit to 1 while the FIFO buffer is empty. Do not write to the FIFO buffer during USB communication in auto response mode. When the selected pipe uses isochronous transfer, always set this bit to 0.

In host controller mode, always set the ATREPM bit to 0.

INBUFM bit (Transmit Buffer Monitor)

The INBUFM bit indicates the FIFO buffer status for the selected pipe in the transmitting direction.

When the selected pipe is set in the transmitting direction (PIPECFG.DIR = 1), the USBFS sets this bit to 1 when the CPU or DMA/DTC completes writing data to at least one FIFO buffer plane.

The USBFS sets this bit to 0 when the USBFS completes transmission of the data from the FIFO buffer plane to which all the data is written. In double buffer mode (PIPECFG.DBLB = 1), the USBFS sets the INBUFM bit to 0 when the USBFS completes transmission of the data from the two FIFO buffer planes before the CPU or DMA/DTC completes writing data to one FIFO buffer plane.

The INBUFM bit indicates the same value as the BSTS bit when the selected pipe is in the receiving direction (PIPECFG.DIR = 0).

BSTS bit (Buffer Status)

The BSTS bit indicates the FIFO buffer status for the selected pipe.

The meaning of the BSTS bit depends on the PIPECFG.DIR, PIPECFG.BFRE, and DnFIFOSEL.DCLRM settings, as shown in Table 36.12.

Table 36.12 BSTS bit operation

DIR value	BFRE value	DCLRM value	BSTS bit function
0	0	0	Sets to 1 when receive data can be read from the FIFO buffer, and clears to 0 on completion of data read
		1	Setting prohibited
	1	0	Sets to 1 when receive data can be read from the FIFO buffer, and clears to 0 when the software sets the BCLR bit in the port control register to 1 after the data read is complete
		1	Sets to 1 when receive data can be read from the FIFO buffer, and clears to 0 on completion of data read
1	0	0	Sets to 1 when transmit data can be written to the FIFO buffer, and clears to 0 on completion of data write
		1	Setting prohibited
	1	0	Setting prohibited
		1	Setting prohibited

36.2.36 PIPEnCTR : PIPEn Control Registers (n = 6 to 9)

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x07A + 0x2 × (n - 6)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSTS	—	—	—	—	—	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PID[1:0]	Response PID 0 0: NAK response 0 1: BUF response (depends buffer state) 1 0: STALL response 1 1: STALL response	R/W
4:2	—	These bits are read as 0. The write value should be 0.	R/W
5	PBUSY	Pipe Busy 0: Pipe n not in use for the transaction 1: Pipe n in use for the transaction	R
6	SQMON	Sequence Toggle Bit Confirmation 0: DATA0 1: DATA1	R
7	SQSET	Sequence Toggle Bit Set*1 Sets the sequence toggle bit for pipe n. 0: Invalid (writing 0 has no effect) 1: Set the expected value for the next transaction to DATA0	W
8	SQCLR	Sequence Toggle Bit Clear*1 Clears the sequence toggle bit for pipe n. 0: Invalid (writing 0 has no effect) 1: Clear the expected value for the next transaction to DATA0	W
9	ACLRM	Auto Buffer Clear Mode*2 0: Disable 1: Enable (all buffers initialized)	R/W
14:10	—	These bits are read as 0. The write value should be 0.	R/W
15	BSTS	Buffer Status 0: Buffer access disabled 1: Buffer access enabled	R

Note: S-TYPE-3, P-TYPE-3

Note 1. Only write 1 to the SQCLR or SQSET bit while PID is NAK. Before setting these bits, check that the PBUSY bit is 0, and then change the PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00b (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

Note 2. Only set the ACLRM bit while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting this bits, check that the PIPEnCTR.PBUSY bit is 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00b (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

PID[1:0] bits (Response PID)

The PID[1:0]bits specify the response type for the next transaction of the selected pipe.

The default PID[1:0] setting is NAK. Change the PID[1:0] setting to BUF to use the associated pipe for USB transfer. Table 36.9 and Table 36.10 show the basic operation (when there are no errors in the transmitted and received packets) of the USBFS depending on the PID[1:0] setting.

After changing the PID[1:0] setting from BUF to NAK through the software during USB communication on the selected pipe, check that the PBUSY bit is 1 to see if USB transfer on the selected pipe has actually entered the NAK state. If the USBFS changes the PID[1:0] bits to NAK, checking the PBUSY bit through the software is not necessary.

The USBFS changes the PIPEnCTR.PID[1:0] setting in the following cases:

- The USBFS sets PID to STALL (11b) on receiving a data packet with a payload exceeding the maximum packet size of the selected pipe
- The USBFS sets PID to NAK on detecting a USB bus reset in device controller mode
- The USBFS sets PID to NAK on detecting a reception error, such as a CRC error, three consecutive times in host controller mode
- The USBFS sets PID to STALL (11b) on receiving the STALL handshake in host controller mode

To specify each response type, set the PID[1:0] bits as follows:

- To transition from NAK (00b) to STALL, set 10b
- To transition from BUF (01b) to STALL, set 11b
- To transition from STALL (11b) to NAK, set 10b and then 00b
- To transition from STALL to BUF, set 00b (NAK) and then 01b (BUF)

PBUSY bit (Pipe Busy)

The PBUSY bit indicates whether the selected pipe is being used for the current transaction.

The USBFS changes the PBUSY bit from 0 to 1 on start of the USB transaction for the selected pipe, and changes the PBUSY bit from 1 to 0 on completion of one transaction.

Reading the PBUSY bit by software after PID is set to NAK allows you to check whether changing the pipe setting is possible.

SQMON bit (Sequence Toggle Bit Confirmation)

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction of the selected pipe.

The USBFS toggles the SQMON bit on successful completion of the transaction. However, the USBFS does not toggle the SQMON bit when a DATA-PID mismatch occurs during transfer in the receiving direction.

SQSET bit (Sequence Toggle Bit Set)

Setting the SQSET bit to 1 through the software causes the USBFS to set DATA1 as the expected value of the sequence toggle bit for the next transaction on the selected pipe. The USBFS sets the SQSET bit to 0.

SQCLR bit (Sequence Toggle Bit Clear)

Setting the SQCLR bit to 1 through the software causes the USBFS to clear the expected value of the sequence toggle bit for the next transaction on the selected pipe to DATA0. The USBFS sets the SQCLR bit to 0.

ACLRM bit (Auto Buffer Clear Mode)

The ACLRM bit enables or disables auto buffer clear mode for the selected pipe. To completely clear the data in the FIFO buffer allocated to the selected pipe, write 1 and then 0 to the ACLRM bit continuously.

Table 36.13 shows the data cleared by writing 1 and 0 continuously to the ACLRM bit and the cases in which this processing is required.

Table 36.13 Data cleared by the USBFS when ACLRM = 1

Number	Data cleared by setting the ACLRM bit	Situations requiring data clear
1	All data in the FIFO buffer allocated to the selected pipe	When initializing the selected pipe
2	Interval count value when the selected pipe is the isochronous transfer type	When resetting the interval count value
3	Internal flags related to the PIPECFG.BFRE bit	When changing the PIPECFG.BFRE setting
4	Internal flags related to the transaction count	When forcing the transaction count function to terminate

BSTS bit (Buffer Status)

The BSTS bit indicates the FIFO buffer status for the selected pipe.

The meaning of the BSTS bit depends on the PIPECFG.DIR, PIPECFG.BFRE, and DnFIFOSEL.DCLRM settings, as shown in Table 36.12.

36.2.37 PIPEnTRE : PIPEn Transaction Counter Enable Register (n = 1 to 5)

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x090 + 0x4 × (n - 1)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TRENB	TRCLR	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	TRCLR	Transaction Counter Clear 0: Invalid (writing 0 has no effect) 1: Clear counter value	R/W
9	TRENB	Transaction Counter Enable 0: Disable transaction counter 1: Enable transaction counter	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note: Set each bit in PIPEnTRE while PID is NAK. Before setting these bits after changing the PIPEnCTR.PID[1:0] bits for the selected pipe from BUF to NAK, check that the PIPEnCTR.PBUSY bit is 0. However, if the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through the software is not necessary.

TRCLR bit (Transaction Counter Clear)

When the TRCLR bit sets to 1, the USBFS clears the value of the transaction counter associated with the selected pipe and then sets the TRCLR bit to 0.

TRENB bit (Transaction Counter Enable)

The TRENB bit enables or disables the transaction counter.

For receiving pipes, setting the TRENB bit to 1 after setting the total number of the packets to be received in the PIPEnTRN.TRNCNT[15:0] bits through the software allows the USBFS to control hardware on having received the number of packets equal to the TRNCNT[15:0] setting, as follows:

- When the PIPECFG.SHTNAK bit is 1, the USBFS changes the PID bits to NAK for the associated pipe on having received the number of packets equal to the TRNCNT[15:0] setting
- When the PIPECFG.BFRE bit is 1, the USBFS asserts the BRDY interrupt on having received the number of packets equal to the TRNCNT[15:0] setting and then reading the last received data

For transmitting pipes, set the TRENB bit to 0.

When the transaction counter is not used, set this bit to 0. When the transaction counter is used, set the TRNCNT[15:0] bits before setting this bit to 1. Set this bit to 1 before receiving the first packet to be counted by the transaction counter.

36.2.38 PIPEnTRN : PIPEn Transaction Counter Register (n = 1 to 5)

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x092 + 0x4 × (n - 1)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TRNCNT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TRNCNT[15:0]	Transaction Counter When written to, this bit specifies the total packets (number of transactions) to be received by the selected pipe. When read from, when PIPEnTRE.TRENB is 0, this bit indicates the specified number of transactions. When PIPEnTRE.TRENB is 1, this bit indicates the current transaction count.	R/W

Note: S-TYPE-3, P-TYPE-3

The PIPEnTRN registers retain their settings during a USB bus reset.

TRNCNT[15:0] bits (Transaction Counter)

The USBFS increments the value of the TRNCNT[15:0] bits by 1 when all of the following conditions are satisfied on receiving the packet:

- The PIPEnTRE.TRENB bit = 1
- (TRNCNT[15:0] set value ≠ current counter value + 1) on receiving the packet
- The payload of the received packet agrees with the PIPEMAXP.MXPS[9:0] setting

The USBFS clears the value of the TRNCNT[15:0] bits to 0 when any of the following conditions are satisfied:

All of the following conditions are satisfied:

- The PIPEnTRE.TRENB bit = 1
- (TRNCNT[15:0] set value = current counter value + 1) on receiving the packet
- The payload of the received packet agrees with the PIPEMAXP.MXPS[9:0] setting

Both of the following conditions are satisfied:

- The PIPEnTRE.TRENB bit = 1
- The USBFS received a short packet

Both of the following conditions are satisfied:

- The PIPEnTRE.TRENB bit = 1
- The PIPEnTRE.TRCLR bit was set to 1 by software

For transmitting pipes, set the TRNCNT[15:0] bits to 0. When the transaction counter is not used, set the TRNCNT[15:0] bits to 0.

Setting the number of transactions to be transferred to the TRNCNT[15:0] bits is only enabled when the PIPEnTRE.TRENB bit is 0. To set the number of transactions to be transferred, set the TRCLR bit to 1 to clear the current counter value before setting the PIPEnTRE.TRENB bit to 1.

36.2.39 DEVADDn : Device Address n Configuration Register (n = 0 to 5)

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x0D0 + 0x2 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	USBSPD[1:0]		—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
7:6	USBSPD[1:0]	Transfer Speed of Communication Target Device 0 0: Do not use DEVADDn 0 1: Low-speed 1 0: Full-speed 1 1: Setting prohibited	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The DEVADDn register specifies the transfer speed of the peripheral device that is the communication target for pipes 0 to 9.

In host controller mode, set all DEVADDn bits before starting communication to any pipes. Only change the bits in DEVADDn when no valid pipes are using the bit settings. A valid pipe is defined as one that satisfies both of the following conditions:

- The target device of the DEVADDn register is selected in the DEVSEL[3:0] bits
- The PID[1:0] bits are set to BUF for the selected pipe, or the selected pipe is the DCP with the DCPCTR.SUREQ bit set to 1

In device controller mode, set all bits in this register to 0.

USBSPD[1:0] bits (Transfer Speed of Communication Target Device)

TheUSBSPD[1:0] bits specify the USB transfer speed of the target peripheral device. Set these bits to 10b when a full-speed device is connected through the hub. In host controller mode, the USBFS generates packets based on the USBSPD[1:0] setting. In device controller mode, set these bits to 00b.

36.2.40 DPUSR0R : Deep Software Standby USB Transceiver Control/Pin Monitor Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x400

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	DVBSTS0	—	DOVCB0	DOVCA0	—	—	DM0	DP0
Value after reset:	0	0	0	0	0	0	0	0	x	0	x	x	0	0	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	FIXPHY0	DRPD0	—	RPUE0	SRPC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SRPC0*1	USB Single-ended Receiver Control 0: Disable input through DP and DM inputs 1: Enable input through DP and DM inputs	R/W
1	RPUE0*1	DP Pull-Up Resistor Control 0: Disable DP pull-up resistor 1: Enable DP pull-up resistor	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	DRPD0*1	D+/D- Pull-Down Resistor Control 0: Disable DP/DM pull-down resistor 1: Enable DP/DM pull-down resistor	R/W
4	FIXPHY0	USB Transceiver Output Fix 0: Fix outputs in Normal mode and on return from Deep Software Standby mode 1 1: Fix outputs on transition to Deep Software Standby mode 1	R/W
15:5	—	These bits are read as 0. The write value should be 0.	R/W
16	DP0	USB D+ Input Indicates D+ input signal on the USBFS side	R
17	DM0	USB D- Input Indicates D– input signal on the USBFS side	R
19:18	—	These bits are read as 0. The write value should be 0.	R/W
20	DOVCA0	USB OVRCURA-DS Input*2 Indicates OVRCURA-DS input signal on the USBFS side	R
21	DOVCB0	USB OVRCURB-DS Input*2 Indicates OVRCURB-DS input signal on the USBFS side	R
22	—	The read value is undefined. The write value should be 0.	R/W
23	DVBSTS0	USB VBUS Input Indicates VBUS input signal on the USBFS side	R
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note: This register is accessed by the clock of PCLKB/64 to reduce current consumption in Deep Software Standby mode1.

Note 1. Use this bit during operation in Deep Software Standby mode 1. For details, see section 36.3.1.5. Release from Deep Software Standby Mode 1 Because of USB Suspend/Resume Interrupts.

Note 2. OVRCURA or OVRCURB can not be used. Only OVRCURA-DS or OVRCURB-DS can be used in Deep software standby mode1.

SRPC0 bit (USB Single-ended Receiver Control)

The SRPC0 bit controls the D+ and D- inputs of the USB transceiver. In host controller mode, set this bit to 1. In device controller mode, set this bit to 0 when disconnected, set to 1 when suspended. This bit is only valid when the FIXPHY0 bit is 1.

FIXPHY0 bit (USB Transceiver Output Fix)

The FIXPHY0 bit keeps the outputs of the USB transceiver disabled.

36.2.41 DPUSR1R : Deep Software Standby USB Suspend/Resume Interrupt Register

Base address: USBFS = 0x4025_0000
USBFS_NS = 0x5025_0000

Offset address: 0x404

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	DVBINT0	—	DOVRCRB0	DOVRCRA0	—	—	DMINT0	DPINT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	DVBSE0	—	DOVRCRBE0	DOVRCRAE0	—	—	DMINTE0	DPINTE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DPINTE0	USB DP Interrupt Enable/Clear 0: Disable recovery from Deep Software Standby mode 1 by DP input 1: Enable recovery from Deep Software Standby mode 1 by DP input	R/W
1	DMINTE0	USB DM Interrupt Enable/Clear 0: Disable recovery from Deep Software Standby mode 1 by DM input 1: Enable recovery from Deep Software Standby mode by DM input	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	DOVRCRAE0	USB OVRCURA-DS Interrupt Enable/Clear*1 0: Disable recovery from Deep Software Standby mode 1 by OVRCURA-DS input 1: Enable recovery from Deep Software Standby mode 1 by OVRCURA-DS input	R/W
5	DOVRCRBE0	USB OVRCURB-DS Interrupt Enable/Clear*1 0: Disable recovery from Deep Software Standby mode 1 by OVRCURB-DS input 1: Enable recovery from Deep Software Standby mode 1 by OVRCURB-DS input	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	DVBSE0	USB VBUS Interrupt Enable/Clear 0: Disable recovery from Deep Software Standby mode 1 by VBUS input 1: Enable recovery from Deep Software Standby mode 1 by VBUS input	R/W
15:8	—	These bits are read as 0. The write value should be 0..	R/W
16	DPINT0	USB DP Interrupt Source Recovery 0: System has not recovered from Deep Software Standby mode 1 1: System recovered from Deep Software Standby mode 1 because of DP	R
17	DMINT0	USB DM Interrupt Source Recovery 0: System has not recovered from Deep Software Standby mode 1 1: System recovered from Deep Software Standby mode 1 because of DM input	R
19:18	—	These bits are read as 0. The write value should be 0.	R/W
20	DOVRCRA0	USB OVRCURA-DS Interrupt Source Recovery*1 0: System has not recovered from Deep Software Standby mode 1 1: System recovered from Deep Software Standby mode 1 because of OVRCURA-DS input	R
21	DOVRCRB0	USB OVRCURB-DS Interrupt Source Recovery*1 0: System has not recovered from Deep Software Standby mode 1 1: System recovered from Deep Software Standby mode 1 because of OVRCURB-DS input	R
22	—	This bit is read as 0. The write value should be 0.	R/W
23	DVBINT0	USB VBUS Interrupt Source Recovery 0: System has not recovered from Deep Software Standby mode 1 1: System recovered from Deep Software Standby mode 1 because of VBUS input	R
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This register is accessed by the clock of PCLKB/64 to reduce current consumption in Deep Software Standby mode1.
Note 1. OVRCURA or OVRCURB can not be used. Only OVRCURA-DS or OVRCURB-DS can be used in Deep Software Standby mode1.

DPINTE0 bit (USB DP Interrupt Enable/Clear)

The DPINTE0 bit enables or disables triggering of recovery from Deep Software Standby mode 1 by the DP input of the USBFS. Writing 0 to this bit while the DPINT0 bit is 1 sets the DPINT0 bit to 0.

DMINTE0 bit (USB DM Interrupt Enable/Clear)

The DMINTE0 bit enables or disables triggering of recovery from Deep Software Standby mode 1 by the DM input of the USBFS. Writing 0 to this bit while the DMINT0 bit is 1 clears the DMINT0 bit to 0.

DOVRCRAE0 bit (USB OVRCURA-DS Interrupt Enable/Clear)

The DOVRCRAE0 bit enables or disables triggering of recovery from Deep Software Standby mode 1 by the OVRCURA-DS input of the USBFS. Writing 0 to this bit while the DOVRCRA0 bit is 1 clears the DOVRCRA0 bit to 0.

DOVRCRBE0 bit (USB OVRCURB-DS Interrupt Enable/Clear)

The DOVRCRBE0 bit enables or disables triggering of recovery from Deep Software Standby mode 1 by the OVRCURB-DS input of the USBFS. Writing 0 to this bit while the DOVRCRB0 bit is 1 clears the DOVRCRB0 bit to 0.

DVBSE0 bit (USB VBUS Interrupt Enable/Clear)

The DVBSE0 bit enables or disables triggering of recovery from Deep Software Standby mode 1 by the VBUS input of the USBFS. Writing 0 to this bit while the DVBINT0 bit is 1 clears the DVBINT0 bit to 0.

DPINT0 bit (USB DP Interrupt Source Recovery)

The DPINT0 bit indicates that the system has returned from Deep Software Standby mode 1 because of the DP input of the USBFS. This recovery is only enabled when the DPINTE0 bit is 1. Writing 0 to the DPINTE0 bit while this bit is 1 clears this bit to 0.

DMINT0 bit (USB DM Interrupt Source Recovery)

The DMINT0 bit indicates that the system has returned from Deep Software Standby mode 1 because of the DM input of the USBFS. This recovery is only enabled when the DMINTE0 bit is 1. Writing 0 to the DMINTE0 bit while this bit is 1 clears this bit to 0.

DOVRCRA0 bit (USB OVRCURA-DS Interrupt Source Recovery)

The DOVRCRA0 bit indicates that the system has returned from Deep Software Standby mode 1 because of the OVRCURA-DS input of the USBFS. This recovery is only enabled when the DOVRCRAE0 bit is 1. Writing 0 to the DOVRCRAE0 bit while this bit is 1 clears this bit to 0.

DOVRCRB0 bit (USB OVRCURB-DS Interrupt Source Recovery)

The DOVRCRB0 bit indicates that the system has returned from Deep Software Standby mode 1 because of the OVRCURB-DS input of the USBFS. This recovery is only enabled when the DOVRCRBE0 bit is 1. Writing 0 to the DOVRCRBE0 bit while this bit is 1 clears this bit to 0.

DVBINT0 bit (USB VBUS Interrupt Source Recovery)

The DVBINT0 bit indicates that the system has returned from Deep Software Standby mode 1 because of the VBUS input of the USBFS. This recovery is only enabled when the DVBSE0 bit is 1. Writing 0 to the DVBSE0 bit while this bit is 1 clears this bit to 0.

36.3 Operation

36.3.1 System Control

This section describes register settings required for initializing the USBFS and controlling power consumption.

36.3.1.1 Setting Data to the USBFS Registers

Setting the SYSCFG.USBE bit to 1 after starting the clock supply (SYSCFG.SCKE bit = 1) enables and starts USBFS operation.

36.3.1.2 Selecting the Controller Function

The USBFS can operate as either a host or device controller.

Use the SYSCFG.DCFM bit to select one of these USBFS functions. The DCFM bit must be changed in the initial settings immediately after a reset or in the D+ pull-up-disabled state (SYSCFG.DPRPU bit = 0) and D+ and D– pull-down-disabled state (SYSCFG.DRPD bit = 0).

36.3.1.3 Controlling the USB Data Bus Using Resistors

The USBFS provides pull-up and pull-down resistors for the D+ and D– lines. Pull these lines up or down by setting the SYSCFG.DPRPU and DRPD bits.

In device controller mode, confirm that connection to the USB host is made, and then set the SYSCFG.DPRPU bit to 1 and pull up the D+ line (in full-speed communication).

When the SYSCFG.DPRPU bit is set to 0 during communication with a PC, the USBFS disables the pull-up resistor of the USB data line, thereby notifying the USB host of disconnection.

In host controller mode, set the SYSCFG.DRPD bit to 1 to pull down the D+ and D– lines.

Table 36.14 USB data bus resistor control

SYSCFG register settings		USB data bus control		
DRPD bit	DPRPU bit	D–	D+	Function
0	0	Open	Open	When resistors not used
0	1	Open	Pull-up	When operating as a device controller at full-speed
1	0	Pull-down	Pull-down	When operating as a host controller
1	1	—	—	Setting prohibited

36.3.1.4 Example External Connection Circuits

Figure 36.2 shows an example OTG connection in the self-powered system. The USBFS controls the pull-up resistor of the D+ line and the pull-down resistor of D+ and D- lines. Select pull-up and pull-down for the lines in the SYSCFG.DPRPU and SYSCFG.DRPD bits. In device controller mode, the pull-up resistor of USB data line is disabled if SYSCFG.DPRPU bit is set to 0 while communicating with the USB host. The USBFS can use this to notify the USB host of a device disconnect.

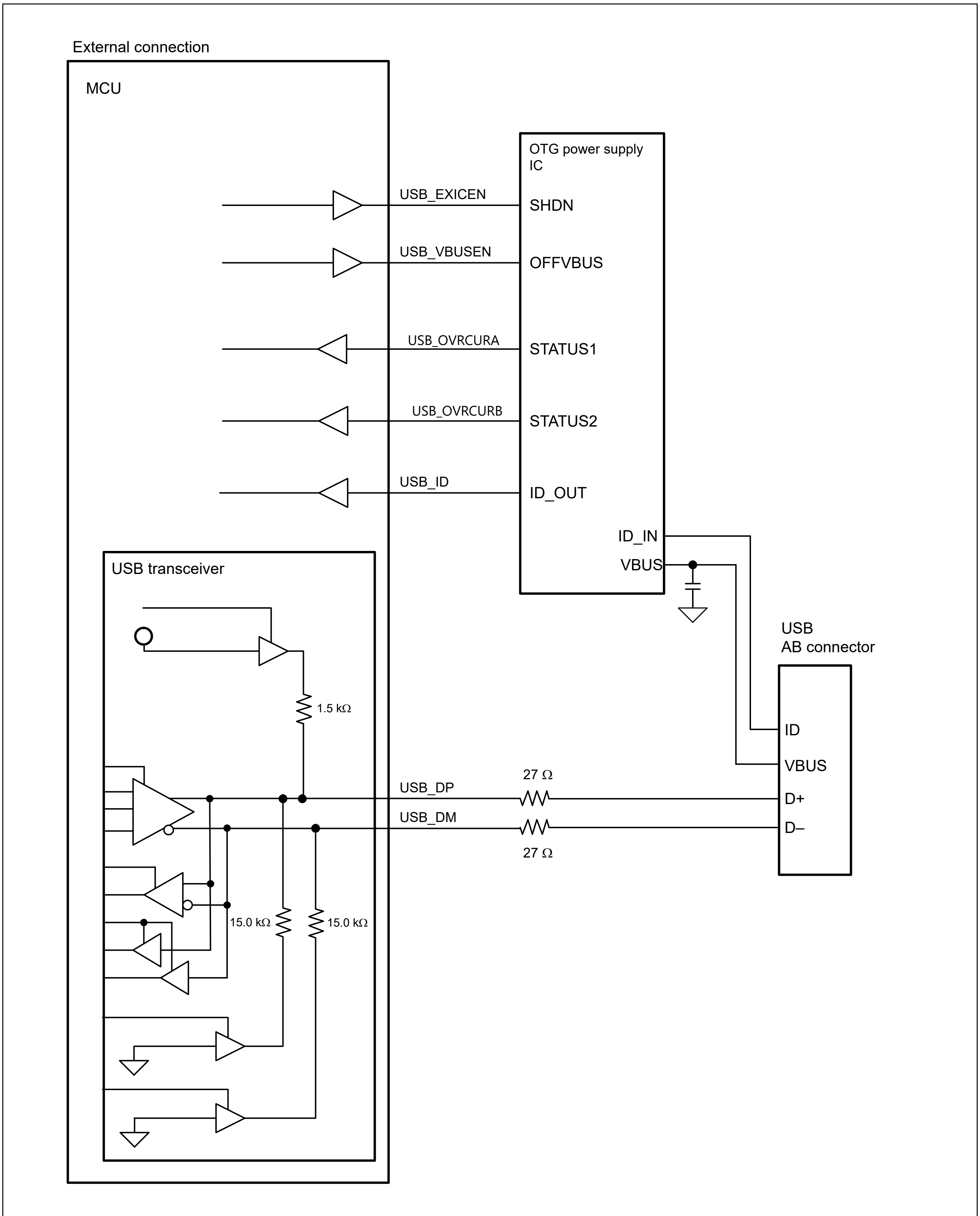


Figure 36.2 Example OTG connection in a self-powered system

Figure 36.3 shows an example device connection in a self-powered system.

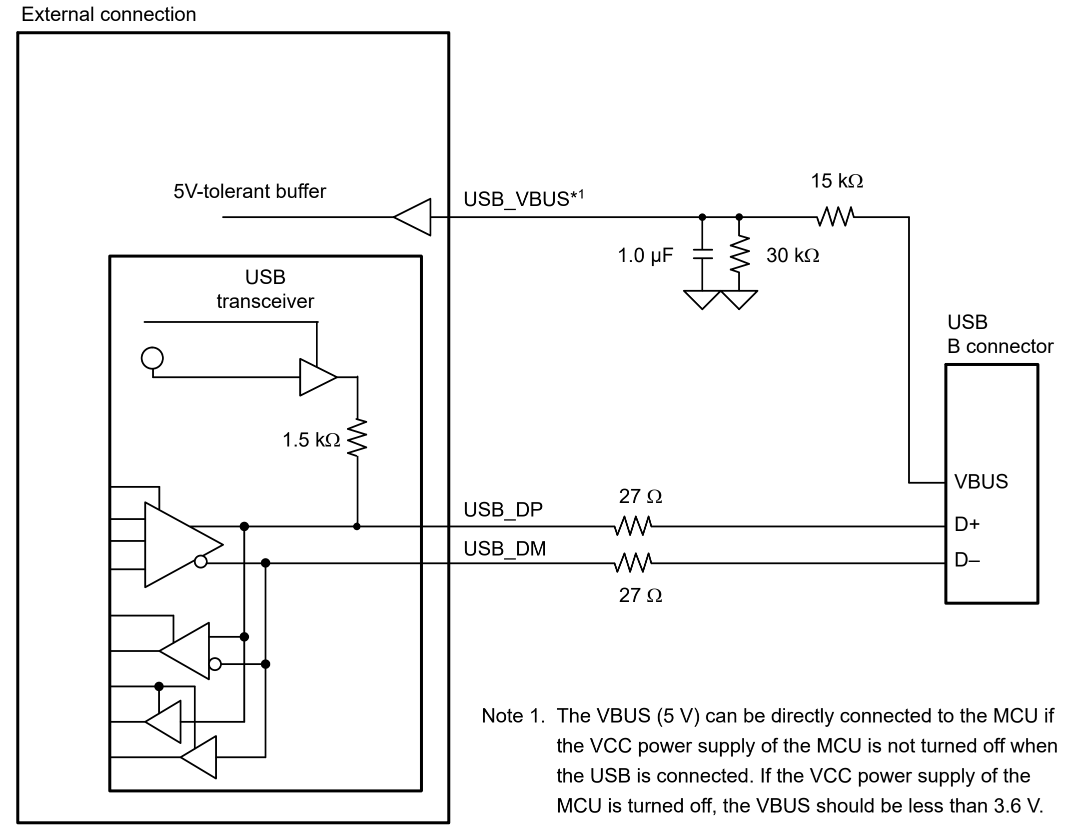


Figure 36.3 Example device connection in a self-powered system

Figure 36.4 shows an example host connection.

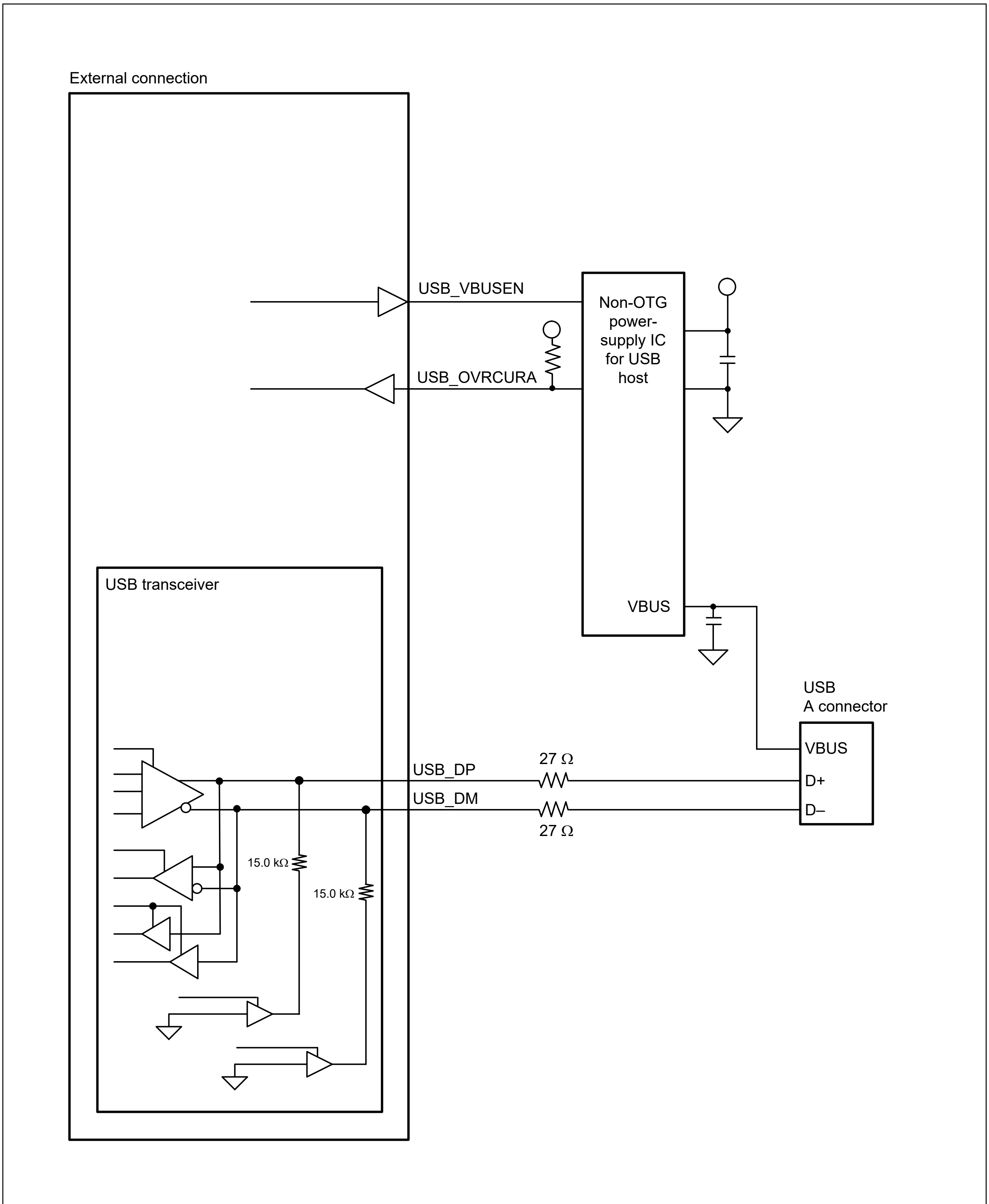


Figure 36.4 Example host connection

Figure 36.5 shows an example device connection in a bus-powered system.

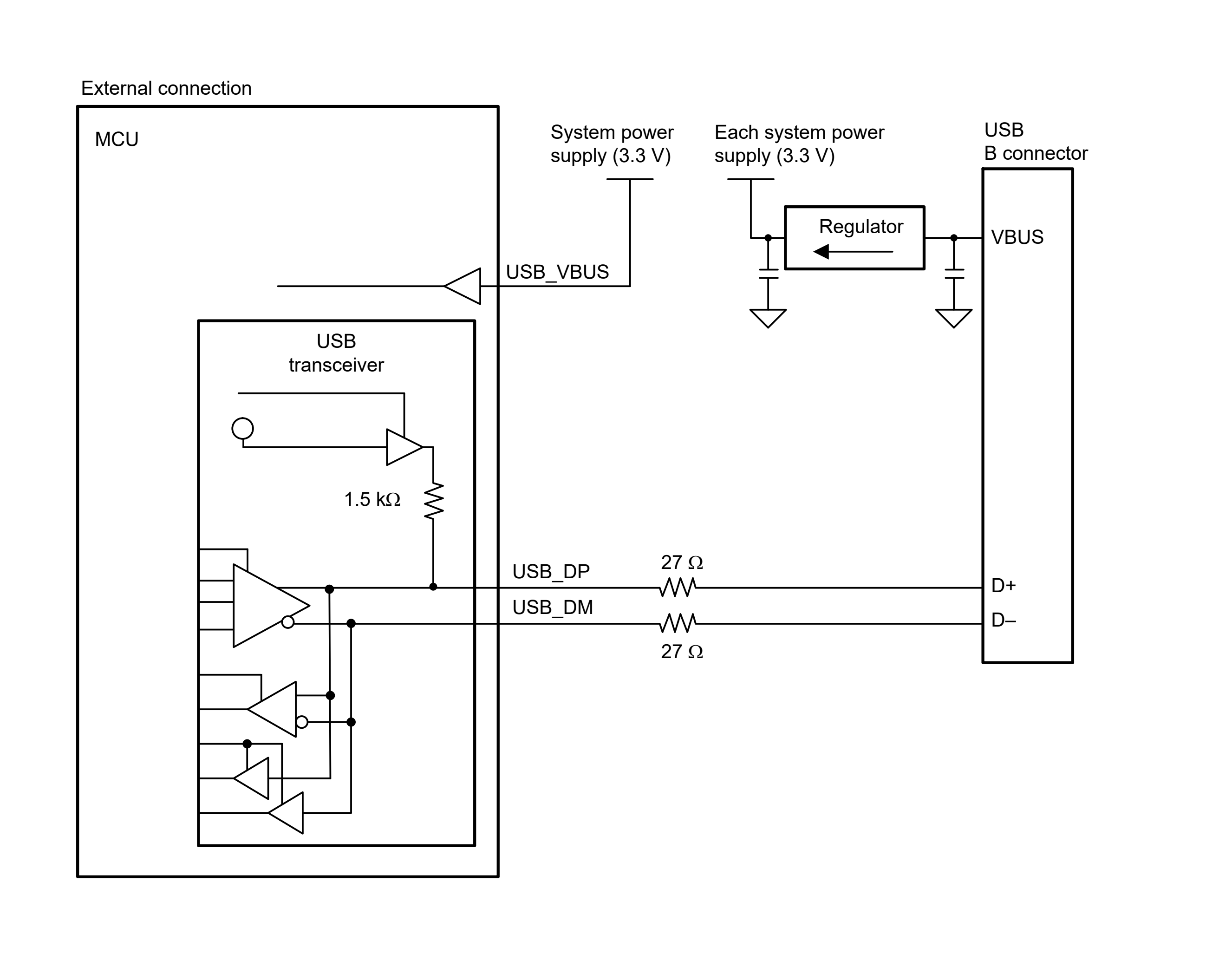


Figure 36.5 Example device connection in a bus-powered state

The examples of external circuits given in this section are simplified circuits, and their operation in every system is not guaranteed.

36.3.1.5 Release from Deep Software Standby Mode 1 Because of USB Suspend/ Resume Interrupts

Deep Software Standby mode1 can be canceled by a USB suspend/resume interrupt. USB suspend/resume interrupts are detected by the USB resume detecting unit, which controls and monitors the USB I/O pins to detect the interrupts.

Figure 36.6 shows a schematic diagram of the connection between the USB resume detecting unit and the USB I/O pins.

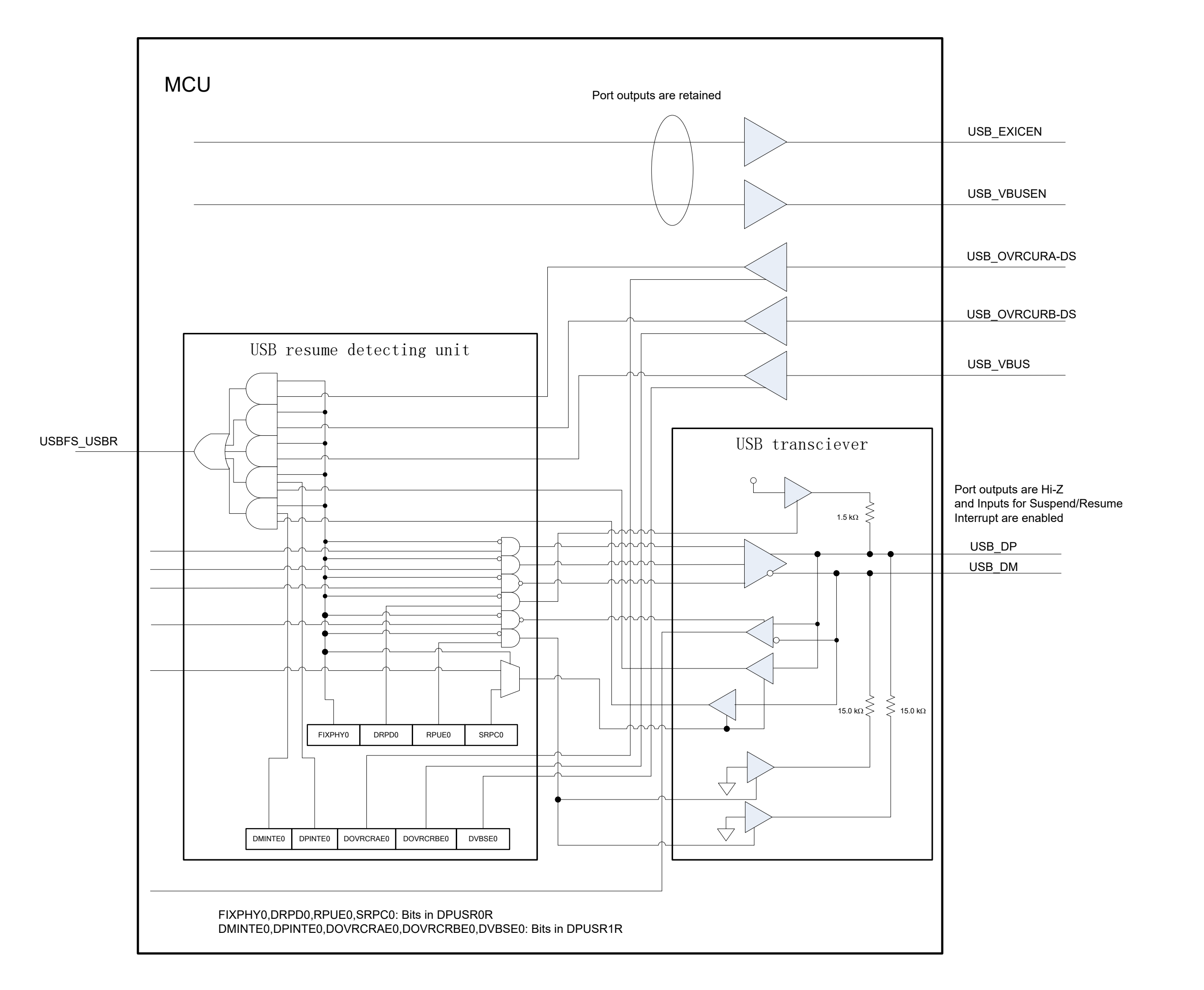


Figure 36.6 Connection between the USB resume detecting unit and the USB I/O pins

Table 36.15 shows the USB suspend and resume interrupt sources and their associated I/O pins.

Table 36.15 USB suspend and resume interrupt sources and their associated I/O pins

USB operating mode	Source	Pin name
Device, OTG	Resume	USB_DP
Host, OTG	Attach or detach	USB_DP, USB_DM
Device	Attach or detach	USB_VBUS
Host	Overcurrent detection	USB_OVRCURA-DS
OTG	Overcurrent detection	USB_OVRCURA-DS, USB_OVRCURB-DS

Figure 36.7 shows the flow for setting the USBFS when entering Deep Software Standby mode 1 from either host or device controller mode. Figure 36.8 shows the flow for setting the USBFS when canceling Deep Software Standby mode 1 from host controller mode. Figure 36.9 shows the flow for setting the USBFS when canceling Deep Software Standby mode 1 from device controller mode.

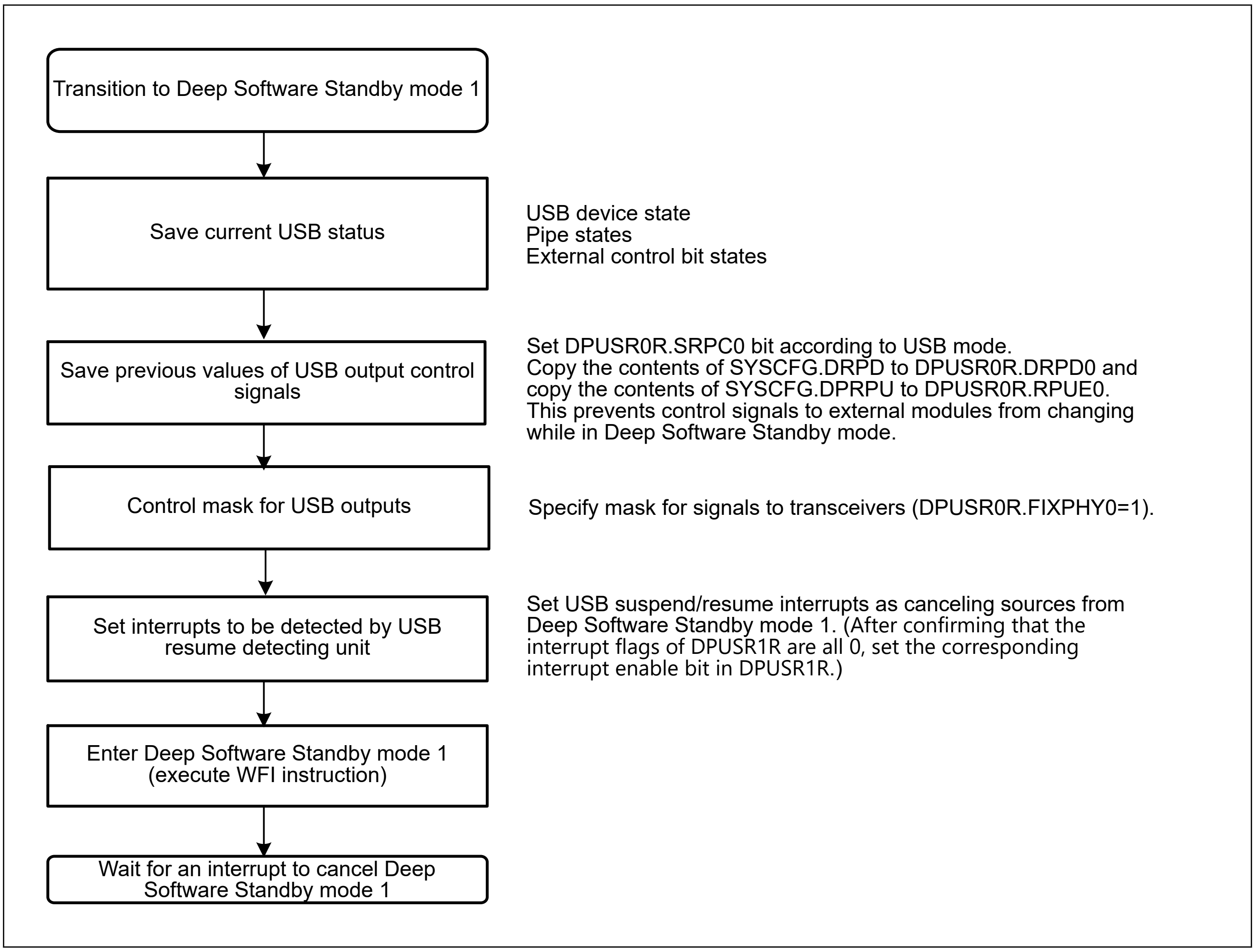


Figure 36.7 USBFS setup flow for transition to Deep Software Standby mode 1 as host or device controller

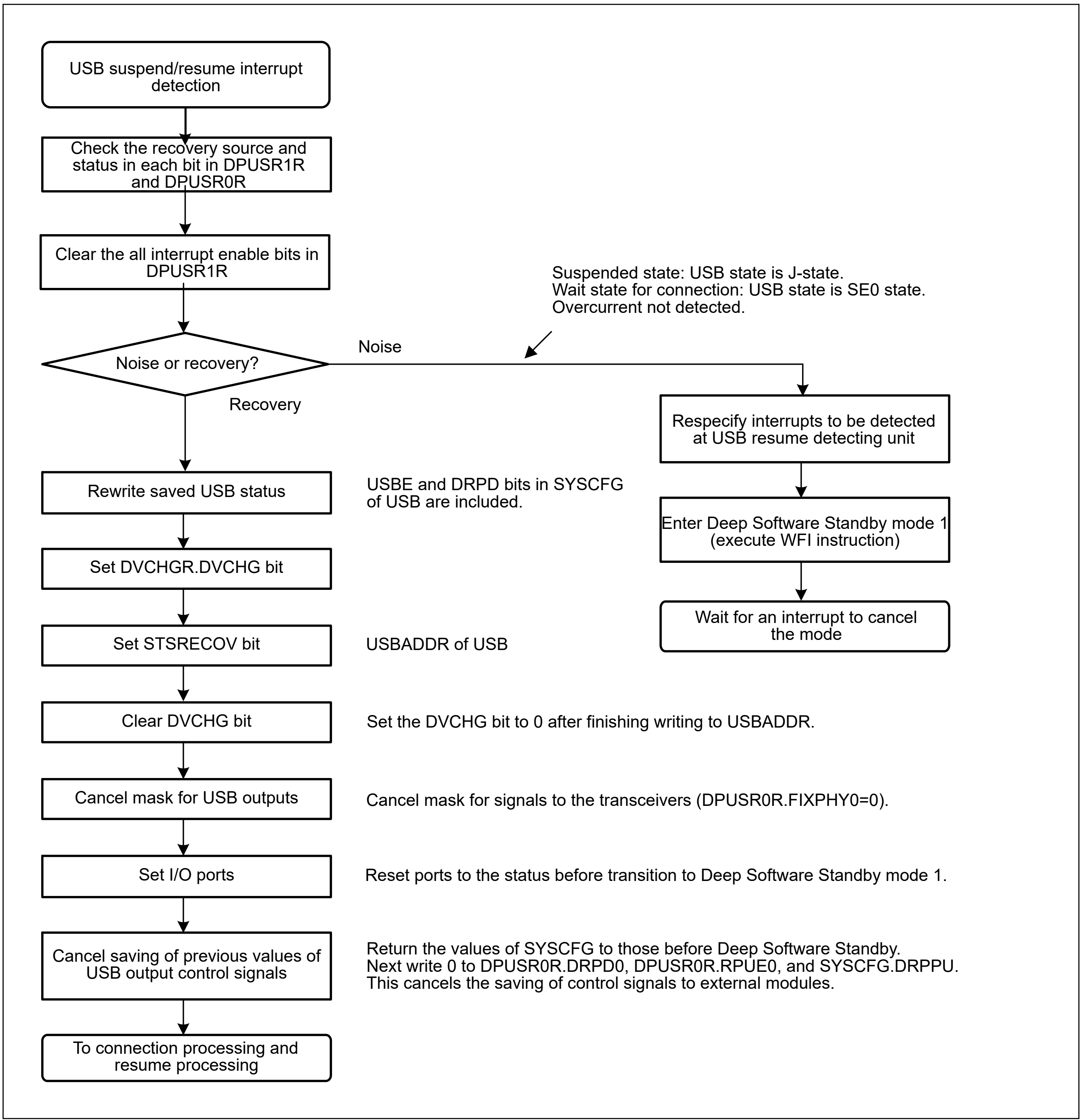


Figure 36.8 USBFS setup flow for canceling Deep Software Standby mode 1 as host controller

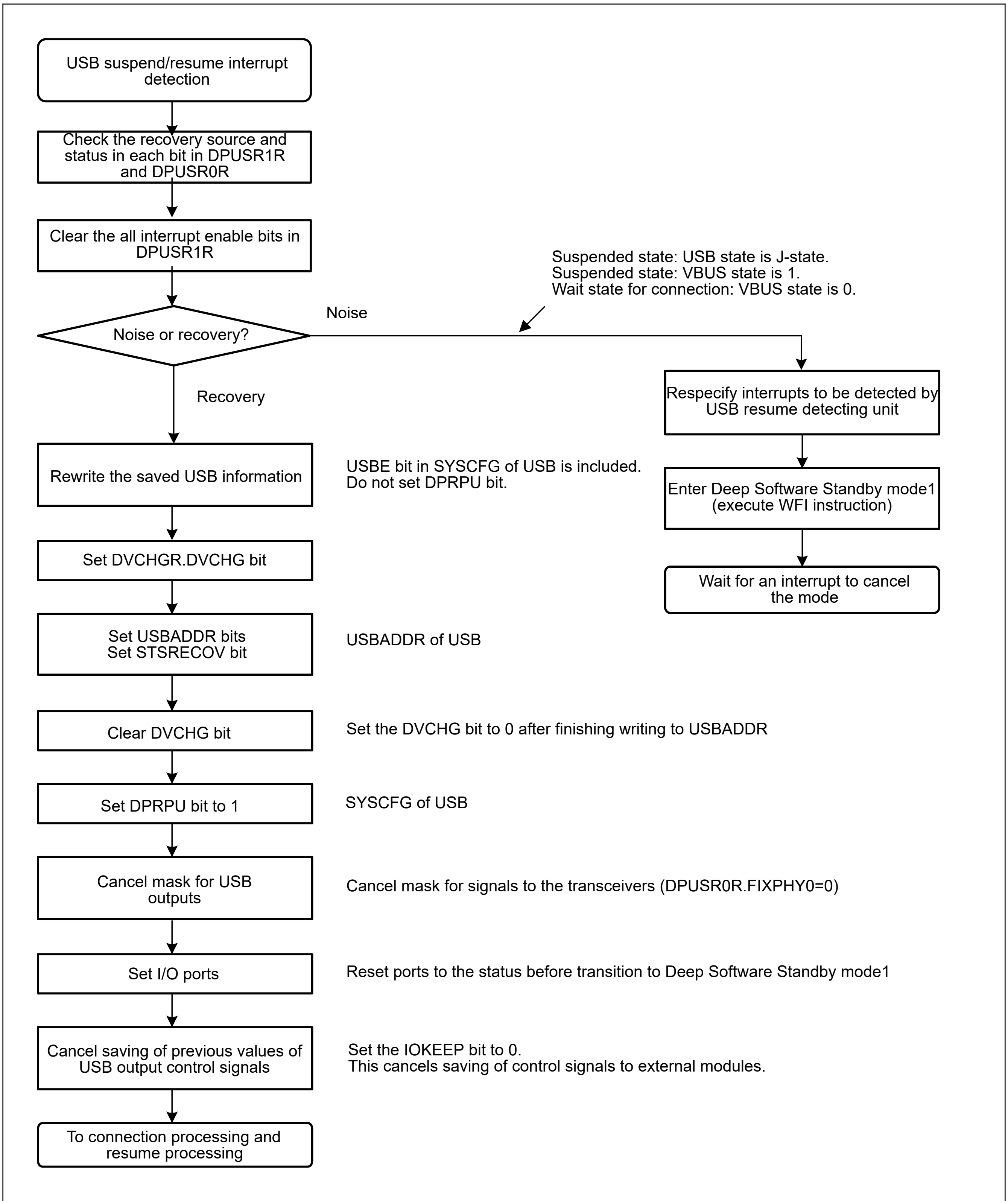


Figure 36.9 USBFS setup flow for canceling Deep Software Standby mode 1 as device controller

36.3.2 Interrupts

Table 36.16 lists the interrupt sources in the USBFS. When an interrupt generation condition is satisfied and the interrupt output is enabled using the associated interrupt enable register, a USBFS interrupt request is issued to the Interrupt Controller Unit (ICU) and an USBFS interrupt is generated.

Table 36.16 Interrupt sources (1 of 2)

Bit to be set to 1	Name	Interrupt source	Applicable controller function	Status flag
VBINT	VBUS interrupt	• A change in the state of the USB_VBUS input pin was detected (low to high or high to low)	Host or device*1	INTSTS0.VBSTS
RESM	Resume interrupt	• A change in the state of the USB bus was detected in the Suspend state (J-state to K-state or J-state to SE0)	Device	—
SOFR	Frame number update interrupt	In host controller mode: • An SOF packet with a different frame number was transmitted In device controller mode: • An SOF packet with a different frame number was received	Host or device	—
DVST	Device state transition interrupt	• One of the following device state transitions was detected: – USB bus reset was detected – Suspend state was detected – SET_ADDRESS request was received – SET_CONFIGURATION request was received	Device	INTSTS0.DVSQ[2:0]
CTRT	Control transfer stage transition interrupt	• A control transfer stage transition was detected because of one of the following: – Setup stage completed – Control write transfer status stage transition occurred – Control read transfer status stage transition occurred – Control transfer completed – Control transfer sequence error occurred	Device	INTSTS0.CTSQ[2:0]
BEMP	Buffer empty interrupt	• The buffer is empty after all FIFO buffer data was transmitted • A packet larger than the maximum packet size was received	Host or device	BEMPSTS.PIPEnBEMP
NRDY	Buffer not ready interrupt	In host controller mode • A STALL response was received from the peripheral device in response to the issued token • The response from the peripheral device in response to the issued token was not received successfully (no response three times consecutively or packet reception error three times consecutively) • An overrun or underrun error occurred during isochronous transfer In device controller mode • NAK was returned for an IN or OUT token while the PID [1:0] bits were set to 01b (BUF) • A CRC error or bit stuffing error occurred during data reception in isochronous transfer • An overrun or underrun occurred during data reception in isochronous transfer	Host or device	NRDYSTS.PIPEnNRDY
BRDY	Buffer ready interrupt	• The buffer is ready (readable or writable state)	Host or device	BRDYSTS.PIPEnBRDY
OVRCR	Overcurrent input change interrupt	• USB_OVRCURA, USB_OVRCURA-DS, USB_OVRCURB or USB_OVRCURB-DS input pin state change was detected (low to high or high to low)	Host	INTSTS1.OVRCR
BCHG	Bus change interrupt	• USB bus state change was detected	Host or device	SYSSTS0.LNST[1:0]
DTCH	Disconnect detection during full-speed operation	Peripheral device disconnect was detected in full-speed operation	Host	DVSTCTR0.RHST[2:0]
ATTCH	Device connect detection interrupt	• J-state or K-state was detected on the USB bus for 2.5 μs continuously This interrupt can be used to check whether peripheral devices are connected	Host	—

Table 36.16 Interrupt sources (2 of 2)

Bit to be set to 1	Name	Interrupt source	Applicable controller function	Status flag
EOFERR	EOF error detection interrupt	● An EOF error was detected for a peripheral device	Host	—
SACK	Setup normal interrupt	● A setup transaction normal response (ACK) was received	Host	—
SIGN	Setup error interrupt	● A setup transaction error (no response or ACK packet corruption) was detected three consecutive times	Host	—

Note 1. Although this interrupt can be generated in host controller mode, it is not usually used in this mode.

Figure 36.10 shows the circuits related to the USBFS interrupts.

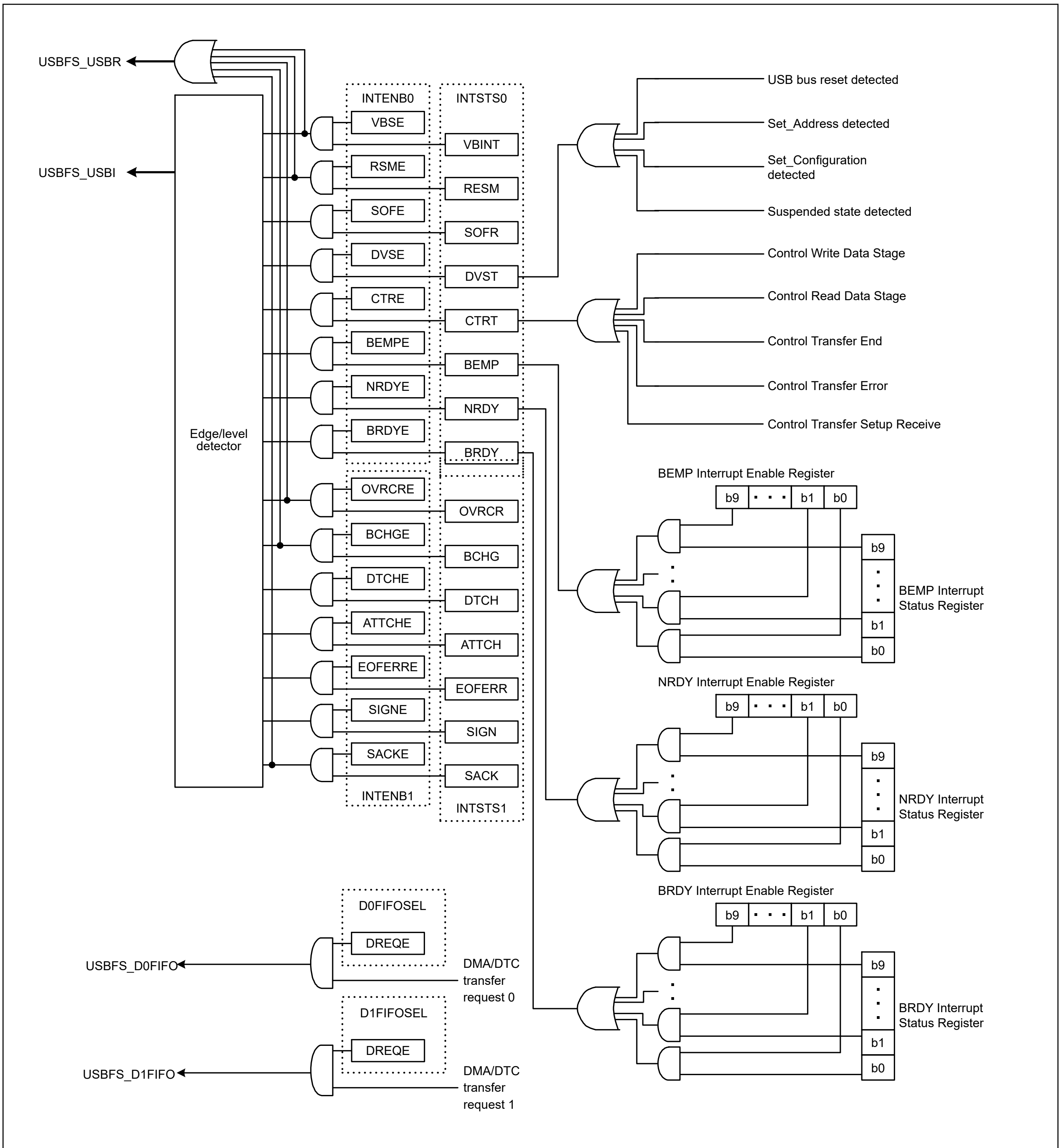


Figure 36.10 USBFS interrupt-related circuits

Table 36.17 shows the interrupts generated by the USBFS.

Table 36.17 USBFS interrupts

Interrupt name	Interrupt status flag	DTC activation	DMAC activation	Priority
USBFS_D0FIFO	DMA transfer request 0	Possible	Possible	High ↑ Low
USBFS_D1FIFO	DMA transfer request 1	Possible	Possible	
USBFS_USBI	VBUS interrupt, resume interrupt, frame number update interrupt, device state transition interrupt, control transfer stage transition interrupt, buffer empty interrupt, buffer not ready interrupt, buffer ready interrupt, overcurrent input change interrupt, bus change interrupt, disconnect detection interrupt during full-speed operation, device connect detection interrupt, EOF error detection interrupt, normal setup operation interrupt, and setup error interrupt	Not possible	Not possible	
USBFS_USBR*1	VBUS interrupt, resume interrupt, overcurrent input change interrupt and setup error interrupt	Not possible	Not possible	—

Note 1. Software Standby mode can be canceled. And Deep Software Standby mode 1 also can be canceled, but only dedicated pin (OVRCURA-DS, OVRCURB-DS) can be used for overcurrent pins.

36.3.3 Interrupt Descriptions

36.3.3.1 BRDY Interrupt

The BRDY interrupt is generated in both host and device controller modes. This section describes the conditions in which the USBFS sets the associated bit in BRDYSTS to 1. Under these conditions, the USBFS generates a BRDY interrupt if the software has set the bit in BRDYENB associated with the given pipe to 1 and the INTENB0.BRDYE bit to 1.

The conditions for generating and clearing the BRDY interrupt depend on the SOFCFG.BRDYM and PIPECFG.BFRE settings for each pipe as follows:

(1) When SOFCFG.BRDYM = 0 and PIPECFG.BFRE = 0

With these settings, the BRDY interrupt indicates that the FIFO port is accessible.

On any of the following conditions, the USBFS generates an internal BRDY interrupt request trigger and sets the BRDYSTS.PIPEnBRDY bit associated with the selected pipe to 1.

For transmitting pipes

- When the DIR bit is changed from 0 to 1 by software
- When packet transmission is complete for a pipe while write-access from the CPU to the FIFO buffer for the pipe is disabled (when the BSTS bit is read as 0)
- When one FIFO buffer is empty on completion of writing data to the other FIFO buffer in double buffer mode
- No request trigger is generated until completion of writing data to the currently-written FIFO buffer even if transmission to the other FIFO buffer is complete
- When the hardware flushes the buffer of the pipe for isochronous transfers
- When 1 is written to the PIPEnCTR.ACLRM bit, which causes the FIFO buffer to transition from the write-disabled to write-enabled state

No request trigger is generated for the DCP, that is, during data transmission for control transfers.

For receiving pipes

- When packet reception is successfully complete, enabling the FIFO buffer to be read while read-access from the CPU to the FIFO buffer for the given pipe is disabled (when the BSTS bit is read as 0). No request trigger is generated for transactions in which a DATA-PID mismatch has occurred.
- When one FIFO buffer is read-enabled on completion of reading data from the other FIFO buffer in double buffer mode. No request trigger is generated until completion of reading data from the currently-read FIFO buffer, even if reception by the other FIFO buffer is complete.

In device controller mode, the BRDY interrupt is not generated in the status stage of control transfers. The PIPEBRDY interrupt status of the selected pipe can be set to 0 by writing 0 to the associated PIPEnBRDY bit through software. In this case, the other PIPEBRDY bit should be set to 1.

Clear the BRDY status before accessing the FIFO buffer.

(2) When SOFCFG.BRDYM = 0 and PIPECFG.BFRE = 1

With these settings, the USBFS generates a BRDY interrupt on completion of reading all data for a single transfer using the receiving pipe, and sets the bit in BRDYSTS associated with the pipe to 1.

On any of the following conditions, the USBFS determines that the last data for a single transfer was received.

- When a short packet including a zero-length packet is received
- When the PIPEn transaction counter register (PIPEnTRN) is used and the number of packets specified in the PIPEnTRN.TRNCNT[15:0] bits are completely received

When the data is completely read after any of these conditions is satisfied, the USBFS determines that all data for a single transfer is completely read.

When a zero-length packet is received while the FIFO buffer is empty, the USBFS determines that all data for a single transfer is completely read when the FRDY bit in the FIFO port control register is 1 and the DTLN[8:0] bits are 0. In this case, to start the next transfer, write 1 to the BCLR bit in the associated port control register through the software. With these settings, the USBFS does not detect a BRDY interrupt for the transmitting pipe.

The PIPEBRDY interrupt status of a pipe can be set to 0 by writing 0 to the associated BRDYSTS.PIPEnBRDY bit through the software. In this case, 1 must be written to the PIPEBRDY bits for the other pipes.

In this mode, do not change the PIPECFG.BFRE bit setting until all data for a single transfer is processed. When it is necessary to change the PIPECFG.BFRE bit before completion of processing, all FIFO buffers for the pipe must be cleared using the PIPEnCTR.ACLRM bit.

(3) When SOFCFG.BRDYM = 1 and PIPECFG.BFRE = 0

With these settings, the BRDYSTS.PIPEnBRDY values are linked to the BSTS bit setting for each pipe. In other words, the BRDY interrupt status bits (PIPEBRDY) are set to 1 or 0 by the USB depending on the FIFO buffer status.

For transmitting pipes

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for write access, and are set to 0 when it is not ready. The BRDY interrupt is not generated for the DCP in the transmitting direction even when it is ready for write access.

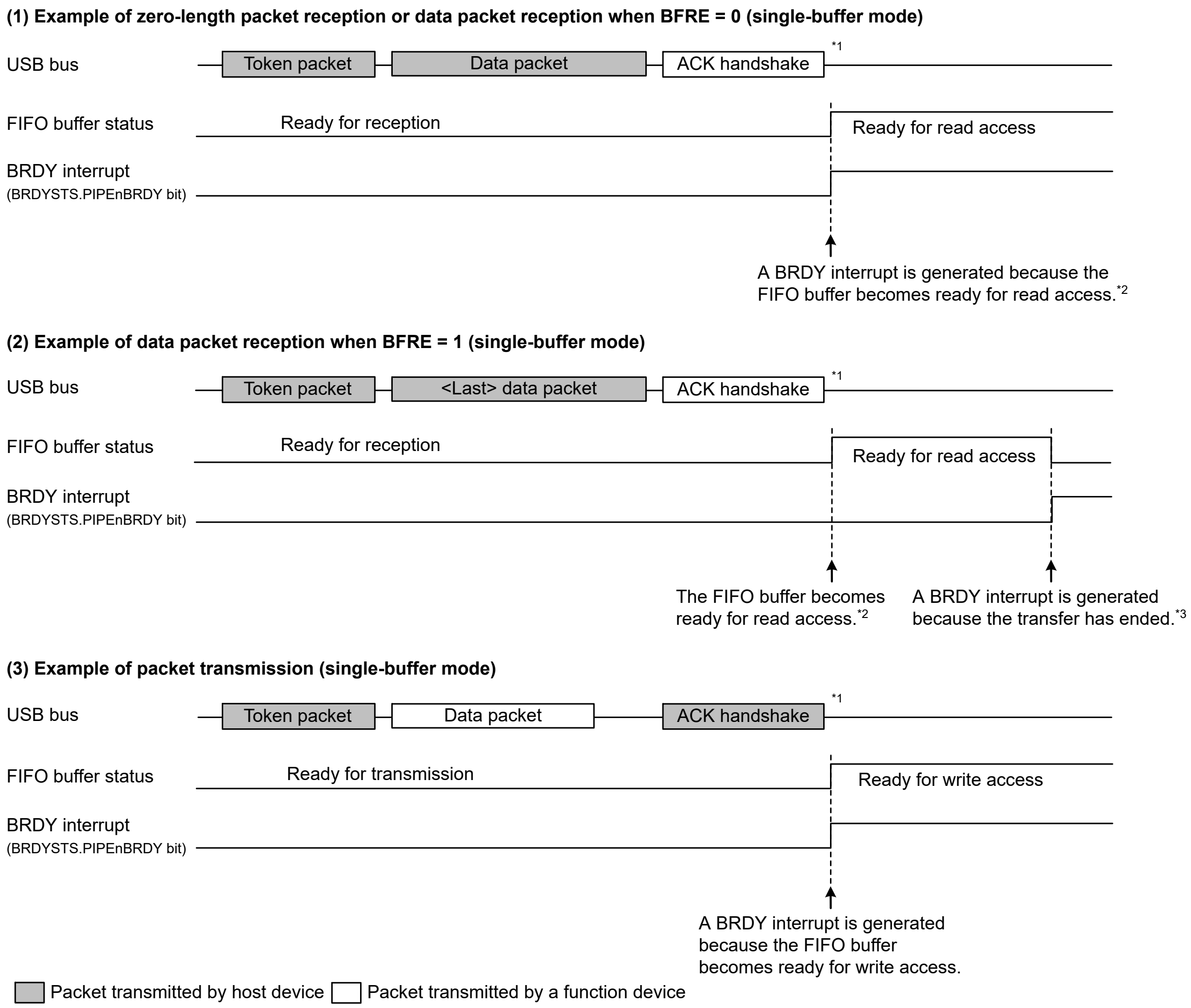
For receiving pipes

The BRDY interrupt status bits set to 1 when the FIFO buffer is ready for read access, and set to 0 when all data is read (not ready for read access).

When a zero-length packet is received while the FIFO buffer is empty, the associated bit is set to 1 and the BRDY interrupt is continuously generated until the software writes 1 to BCLR. With this setting, the PIPEnBRDY bit cannot be set to 0 by software.

When the SOFCFG.BRDYM bit is set to 1, set the PIPECFG.BFRE bit for all pipes to 0.

Figure 36.11 shows the timing of BRDY interrupt generation.



Note 1. The ACK handshake is not used in isochronous transfers.
Note 2. The FIFO buffer becomes ready for read access under the following condition:
When a packet is received while no data remains unread in the FIFO buffer in the CPU.
Note 3. A transfer ends under either of the following conditions:
1. When a short packet including a zero-length packet is received.
2. When the number of packets specified in the transaction counter are received.

Figure 36.11 Timing of BRDY interrupt generation

The condition for clearing the INTSTS0.BRDY bit depends on the SOFCFG.BRDYM bit setting, as shown in Table 36.18.

Table 36.18 Conditions for clearing the BRDY bit

BRDYM bit	Condition for clearing BRDY bit
0	The USBFS clears the BRDY bit to 0 when all bits in BRDYSTS are set to 0 by software.
1	The USBFS clears the BRDY bit to 0 when the BSTS bits for all pipes have cleared to 0.

36.3.3.2 NRDY Interrupt

On generating an internal NRDY interrupt request for the pipe whose PID bits are set to BUF by software, the USBFS sets the associated PIPEnNRDY bit in NRDYSTS to 1. If the associated bit in NRDYENB is set to 1 by software, the USBFS sets the INTSTS0.NRDY bit to 1 and generates a USBFS interrupt.

This section describes the conditions in which the USBFS generates the internal NRDY interrupt request for a given pipe.

The internal NRDY interrupt request is not generated during setup transaction execution in host controller mode. During setup transactions in host controller mode, the SACK or SIGN interrupt is detected.

The internal NRDY interrupt request is not generated during status stage execution of the control transfer in device controller mode.

(1) In host controller mode

For transmitting pipes

On any of the following conditions, the USBFS detects an NRDY interrupt:

- For isochronous transfer pipes, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer. In this case, the USBFS transmits a zero-length packet following the OUT token and sets the associated NRDYSTS.PIPEnNRDY bit and the FRMNUM.OVRN bit to 1.
- During communications other than setup transactions on pipes not used for isochronous transfers, when any combination of the following two cases occur three consecutive times:
  - No response is returned from the peripheral device (when timeout is detected before detection of the handshake packet from the peripheral device
  - An error is detected in the packet from the peripheral device. In this case, the USBFS sets the associated PIPEnNRDY bit to 1 and changes the associated PID[1:0] setting for the pipe to NAK
- During communications other than setup transactions, when the STALL handshake is received from the peripheral device. In this case, the USBFS sets the associated PIPEnNRDY bit to 1 and changes the PID[1:0] setting for the associated pipe to STALL (11b).

For receiving pipes

- For isochronous transfer pipes, when the time to issue an IN token comes but there is no space available in the FIFO buffer. In this case, the USBFS discards the received data for the IN token and sets the PIPEnNRDY bit associated with the pipe and the OVRN bit to 1. When a packet error is detected in the received data for the IN token, the USBFS also sets the FRMNUM.CRCE bit to 1.
- For non-isochronous transfer pipes, when any combination of the following two cases occur three consecutive times:
  - No response is returned from the peripheral device for the IN token issued by the USBFS (when timeout is detected before detection of the DATA packet from the peripheral device)
  - An error is detected in the packet from the peripheral device. In this case, the USBFS sets the associated PIPEnNRDY bit to 1 and changes the associated PID[1:0] setting for the pipe to NAK
- For isochronous transfer pipes, when no response is returned from the peripheral device for the IN token (when timeout is detected before detection of the DATA packet from the peripheral device) or an error is detected in the packet from the peripheral device. In this case, the USBFS sets the PIPEnNRDY bit associated with the pipe to 1. The PID[1:0] setting for the pipe is not changed.
- For isochronous transfer pipes, when a CRC error or a bit stuffing error is detected in the received data packet. In this case, the USBFS sets the PIPEnNRDY bit associated with the pipe and the CRCE bit to 1.
- When the STALL handshake is received. In this case, the USBFS sets the PIPEnNRDY bit associated with the pipe to 1 and changes the PID[1:0] setting for the associated pipe to STALL.

(2) In device controller mode

For transmitting pipes

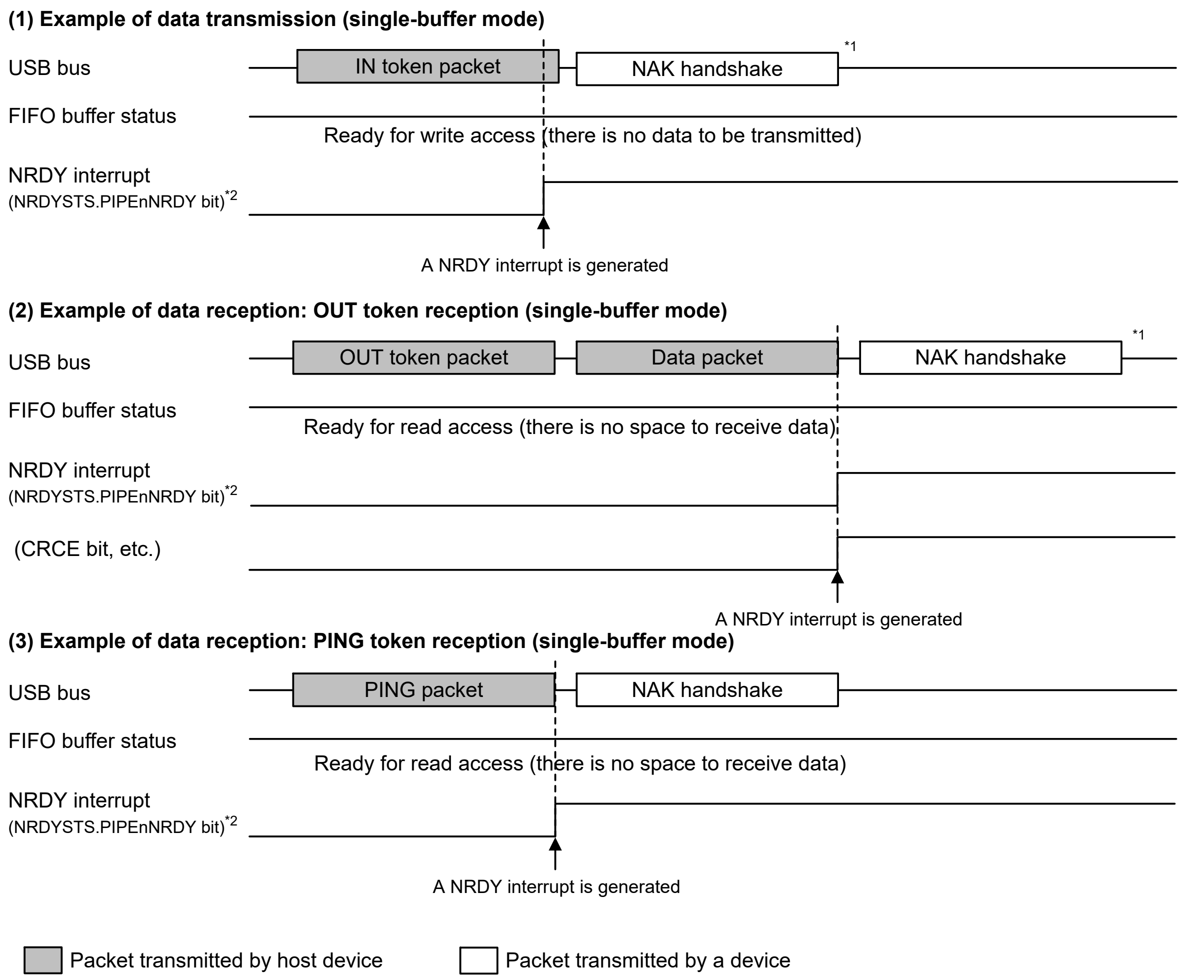
- When an IN token is received while there is no data to be transmitted in the FIFO buffer. In this case, the USBFS generates a NRDY interrupt request on reception of the IN token and sets the NRDYSTS.PIPEnNRDY bit to 1. For an isochronous transfer pipe in which an interrupt is generated, the USBFS transmits a zero-length packet and sets the FRMNUM.OVRN bit to 1.

For receiving pipes

- When an OUT token is received but there is no space available in the FIFO buffer. For an isochronous transfer pipe in which an interrupt is generated, the USBFS generates a NRDY interrupt request on reception of the OUT token and sets the PIPEnNRDY bit to 1 and OVRN bit to 1. For a non-isochronous transfer pipe in which an interrupt is generated, the USBFS generates a NRDY interrupt request when a NAK handshake is transferred after the data following the OUT token is received, and sets the PIPEnNRDY bit to 1. The NRDY interrupt request is not generated during retransmission because of a DATA-PID mismatch. In addition, the NRDY interrupt request is not generated if an error occurs in the DATA packet.

- For isochronous transfer pipes, when a token is not received successfully within an interval frame. In this case, the USBFS generates an NRDY interrupt request when the SOF is received, and sets the PIPEnNRDY bit to 1.

Figure 36.12 shows the timing of NRDY interrupt generation in device controller mode.



Note 1. The handshake is not used in isochronous transfers.
Note 2. The CRCE and OVRN bits change only while the target pipe is set to isochronous transfers.

Figure 36.12 Timing of NRDY interrupt generation in device controller mode

36.3.3.3 BEMP Interrupt

On detecting a BEMP interrupt for the pipe whose PID bits are set to BUF by software, the USBFS sets the associated BEMPSTS.PIPEnBEMP bit to 1. If the associated bit in BEMPENB is set to 1 by software, the USBFS sets the INTSTS0.BEMP bit to 1 and generates a USBFS interrupt. This section describes the conditions in which the USBFS generates an internal BEMP interrupt request.

(1) For transmitting pipes

When the FIFO buffer of the associated pipe is empty on completion of transmission, including zero-length packet transmission, and in single buffer mode, an internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for a non-DCP pipe. The internal BEMP interrupt request is not generated in any of the following conditions:

- When the CPU or DMA/DTC has already started writing data to the FIFO buffer of the CPU on completion of transmitting data from one FIFO buffer in double buffer mode

- When the buffer is cleared (emptied) by setting the PIPEnCTR.ACLRM or the BCLR bit to 1 in the port control register
- When an IN transfer (zero-length packet transmission) is performed during the control transfer status stage in device controller mode

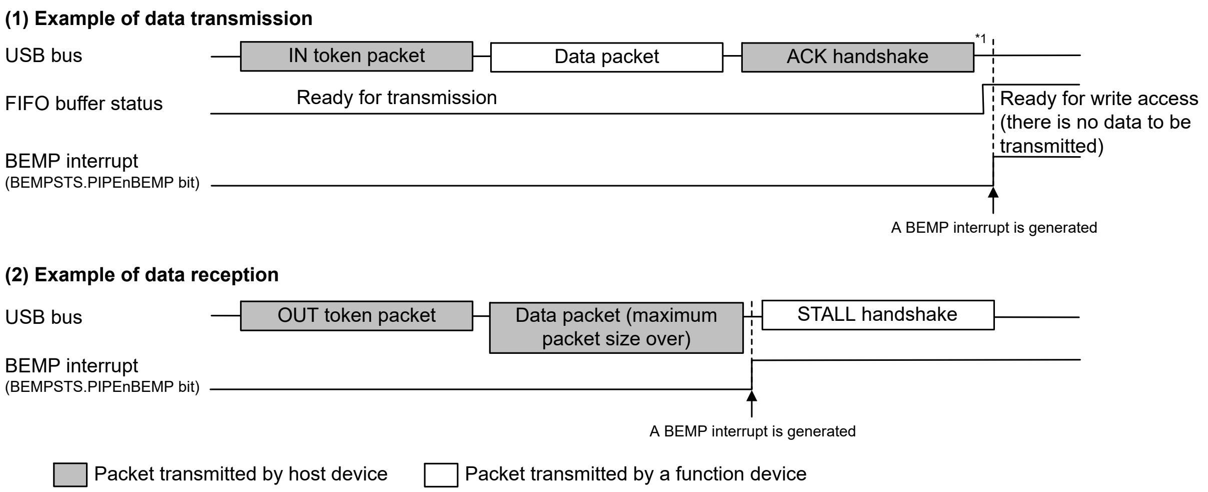
(2) For receiving pipes

When a successfully-received data packet size exceeds the specified maximum packet size. In this case, the USBFS generates a BEMP interrupt request, sets the associated BEMPSTS.PIPEnBEMP bit to 1, discards the received data, and changes the associated PID[1:0] setting for the pipe to STALL (11b). The USBFS returns no response in host controller mode, and returns STALL response in device controller mode.

The internal BEMP interrupt request is not generated in any of the following conditions:

- When a CRC error or a bit stuffing error is detected in the received data
- When a setup transaction is being performed:
  - Writing 0 to the BEMPSTS.PIPEnBEMP bit clears the status
  - Writing 1 to the BEMPSTS.PIPEnBEMP bit has no effect

Figure 36.13 shows the timing of BEMP interrupt generation in device controller mode.



Note 1. The handshake is not used in isochronous transfers.

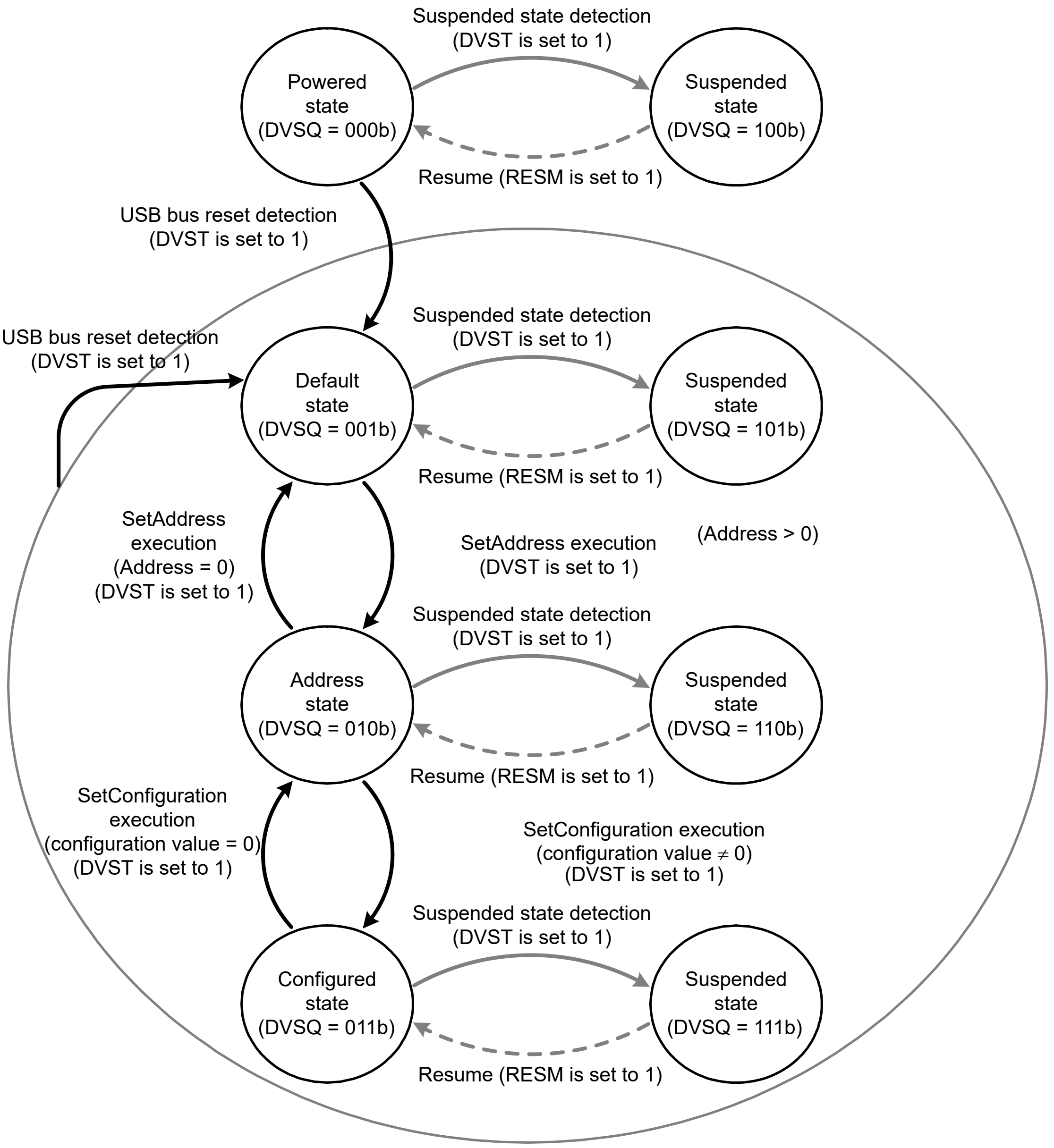
Figure 36.13 Timing of BEMP interrupt generation in device controller mode

36.3.3.4 Device State Transition Interrupt (Device Controller Mode)

Figure 36.14 shows a diagram of the USBFS device state transitions. The USBFS controls device states and generates device state transition interrupts. However, recovery from the Suspend state (resume signal detection) is detected by means of the resume interrupt. Device state transition interrupts can be enabled or disabled independently in INTENB0. Devices whose states have changed can be checked in the INTSTS0.DVSQ[2:0] bits.

When a transition is made to the default state, a device state transition interrupt is generated after a USB bus reset is detected.

The USBFS controls device states, and device state transition interrupts can be generated, only in device controller mode.



Note: For a transition indicated by a solid line, the DVST bit is set to 1.
For a resume indicated by a dashed line, the RESM bit is set to 1.

Figure 36.14 Device state transitions

36.3.3.5 Control Transfer Stage Transition Interrupt (Device Controller Mode)

Figure 36.15 shows a diagram of the control transfer stage transitions of the USBFS. The USBFS controls the control transfer sequence and generates control transfer stage transition interrupts. Control transfer stage transition interrupts can be enabled or disabled independently in INTENB0. Transfer stages that have transitioned can be checked in the INTSTS0.CTSQ[2:0] bits.

Control transfer stage transition interrupts are generated only in device controller mode. This section describes control transfer sequence errors. If an error occurs, the DCPCTR.PID[1:0] bits are set to 1xb (STALL response).

(1) Control read transfer errors

- An OUT token is received but no data is transferred in response to the IN token at the data stage
- An IN token is received at the status stage

- A data packet with DATAPID = DATA0 is received at the status stage

(2) Control write transfer errors

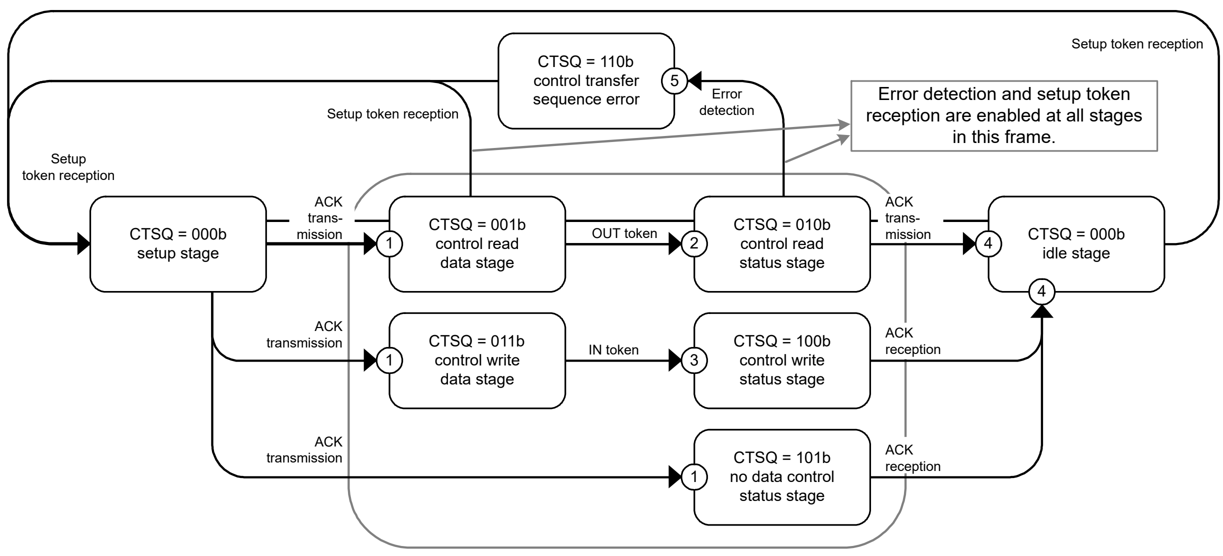
- An IN token is received but no ACK is returned in response to the OUT token at the data stage
- A data packet with DATAPID = DATA0 is received as the first data packet at the data stage
- An OUT token is received at the status stage

(3) Control write no data transfer errors

- An OUT token is received at the status stage

At the control write transfer data stage, if the receive data length exceeds the wLength value of the USB request, it is not recognized as a control transfer sequence error. At the control read transfer status stage, packets other than zero-length packets are received by an ACK response and the transfer ends normally.

When a CTRT interrupt occurs in response to a sequence error (INTSTS0.CTRT = 1), the CTSQ[2:0] = 110b value is saved until the CTRT bit is set to 0, clearing the interrupt status. While CTSQ[2:0] = 110b is being saved, no CTRT interrupt for ending the setup stage is generated, even if a new USB request is received. The USBFS saves the setup stage completion status, and it generates a CTRT interrupt after the interrupt status is cleared by software.



Note: CTRT interrupts
1. Setup stage end
2. Control read transfer status stage transition
3. Control write transfer status stage transition
4. Control transfer end
5. Control transfer sequence error

Figure 36.15 Control transfer stage transitions

36.3.3.6 Frame Update Interrupt

In host controller mode, an interrupt is generated when the frame number is updated.

In device controller mode, an SOFR interrupt is generated when the frame number is updated. The USBFS updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation.

36.3.3.7 VBUS Interrupt

When the USB_VBUS pin level changes, a VBUS interrupt is generated. The level of the USB_VBUS pin can be checked with the INTSTS0.VBSTS bit. Whether the host controller is connected or disconnected can be confirmed using the VBUS interrupt. If the system is activated with the host controller connected, the first VBUS interrupt is not generated, because there is no change in the USB_VBUS pin level.

36.3.3.8 Resume Interrupt

In device controller mode, a resume interrupt is generated when the device state is the Suspend state and the USB bus state has changed (from J-state to K-state, or from J-state to SE0). Recovery from the Suspend state is detected by means of the resume interrupt.

In host controller mode, no resume interrupt is generated. Use the BCHG interrupt to detect a change in the USB bus state.

36.3.3.9 OVRCR Interrupt

An OVRCR interrupt is generated when the USB_OVRCURA, USB_OVRCURA-DS, USB_OVRCURB or USB_OVRCURB-DS pin level has changed. The levels of the USB_OVRCURA, USB_OVRCURA-DS, USB_OVRCURB and USB_OVRCURB-DS pins can be checked in the SYSSTS0.OVCMON[1:0] flags. The external power supply IC can check whether overcurrent is detected using the OVRCR interrupt.

For OTG connections, the OVRCR interrupt allows you to check whether a change is detected in the VBUS comparator.

36.3.3.10 BCHG Interrupt

A BCHG interrupt is generated when the USB bus state has changed. The BCHG interrupt can be used to detect whether a peripheral device is connected and can also be used to detect a remote wakeup in host controller mode. The BCHG interrupt is generated in both host and device controller modes.

36.3.3.11 DTCH Interrupt

A DTCH interrupt occurs when a USB bus disconnect is detected in host controller mode. The USBFS detects bus disconnects in compliance with the USB 2.0 specification.

On interrupt detection, all pipes in which communications are being carried out for the relevant port must be terminated by software. The pipes enter the wait state for a bus connection to the port, waiting for an ATTCH interrupt to occur. Regardless of the value set in the associated interrupt enable bit, the USBFS hardware:

- Sets the DVSTCTR0.UACT bit for the port in which the DTCH interrupt is detected to 0
- Puts the port in which the DTCH interrupt occurred into the idle state

36.3.3.12 SACK Interrupt

A SACK interrupt is generated when an ACK response for the transmitted setup packet is received from the peripheral device in host controller mode. The SACK interrupt can be used to confirm that the setup transaction is successfully complete.

36.3.3.13 SIGN Interrupt

A SIGN interrupt is generated when an ACK response for the transmitted setup packet is not correctly received from the peripheral device three consecutive times in host controller mode. The SIGN interrupt can be used to detect no ACK response transmitted from the peripheral device or corruption of an ACK packet.

36.3.3.14 ATTCH Interrupt

An ATTCH interrupt is generated when J-state or K-state of the full-speed signal level is detected on the USB port for 2.5 µs in host controller mode. To be more specific, an ATTCH interrupt is detected on any of the following conditions:

- When K-state, SE0, or SE1 changes to J-state, and J-state continues 2.5 µs
- When J-state, SE0, or SE1 changes to K-state, and K-state continues 2.5 µs

36.3.3.15 EOFERR Interrupt

An EOFERR interrupt occurs when the USBFS detects that communication is not complete at the EOF2 timing defined in the USB 2.0 specification.

On interrupt detection, all pipes in which communications are being carried out for the relevant port must be terminated by software, and the port must be re-enumerated. Regardless of the value set in the associated interrupt enable bit, the USBFS hardware:

- Sets the DVSTCTR0.UACT bit for the port in which the EOFERR interrupt is detected to 0
- Puts the port in which the EOFERR interrupt is generated into the idle state

36.3.4 Pipe Control

Table 36.19 lists the pipe settings for the USBFS. USB data transfer is performed through logical pipes that the software associates with endpoints. The USBFS provides 10 pipes that are used for data transfer. Set up the pipes based on your system specifications.

Table 36.19 Pipe settings

Register name	Bit name	Setting	Notes
DCPCFG PIPECFG	TYPE	Transfer type	Pipes 1 to 9: Settable
	BFRE	BRDY interrupt mode	Pipes 1 to 5: Settable
	DBLB	Double buffer select	Pipes 1 to 5: Settable
	DIR	Transfer direction select	IN or OUT settable
	EPNUM	Endpoint number	Pipes 1 to 9: Settable A value other than 0000b must be set when the pipe is used.
	SHTNAK	Selects disabled state for pipe when transfer ends	Pipes 1 and 2: Settable only for bulk transfers Pipes 3 to 5: Settable
DCPMAXP PIPEMAXP	DEVSEL	Device select	Referenced only in host controller mode.
	MXPS	Maximum packet size	Compliant with the USB 2.0 specification.
PIPEPERI	IFIS	Buffer flush	Pipes 1 and 2: Settable only for isochronous transfers Pipes 3 to 9: Setting disabled
	IITV	Interval counter	Pipes 1 and 2: Settable only for isochronous transfers Pipes 3 to 5: Setting disabled Pipes 6 to 9: Settable only in host controller mode
DCPCTR PIPEnCTR	BSTS	Buffer status	For the DCP, receive buffer status and transmit buffer status are switched with the ISEL bit.
	INBUFM	IN buffer monitor	Available only for pipes 1 to 5.
	SUREQ	Setup request	Settable only for the DCP and controlled in host controller mode
	SUREQCLR	SUREQ clear	Settable only for the DCP and controlled in host controller mode
	ATREPM	Auto response mode	Pipes 1 to 5: Settable only in device controller mode
	ACLRM	Auto buffer clear	Pipes 1 to 9: Settable
	SQCLR	Sequence clear	Clears the data toggle bit
	SQSET	Sequence set	Sets the data toggle bit
	SQMON	Sequence monitor	Monitors the data toggle bit
	PBUSY	Pipe busy status	—
	PID	Response PID	See section 36.3.4.6. Response PID.
PIPEnTRE	TRENB	Transaction counter enable	Pipes 1 to 5: Settable
	TRCLR	Current transaction counter clear	Pipes 1 to 5: Settable
PIPEnTRN	TRNCNT	Transaction counter	Pipes 1 to 5: Settable

36.3.4.1 Pipe Control Register Switching Procedures

The following bits in the pipe control registers can be changed only when USB communication is prohibited (PID = NAK).

Do not change the following registers and bits when USB communication is enabled (PID = BUF):

- Bits in DCPCFG and DCPMAXP

- SQCLR and SQSET bits in DCPCTR
- Bits in PIPECFG, PIPEMAXP, and PIPEPERI
- ATREPM, ACLRM, SQCLR, and SQSET bits in PIPEnCTR
- Bits in PIPEnTRE and PIPEnTRN

To set these bits when USB communication is enabled (PID = BUF):

1. A request to change the bits in the pipe control register occurs.
2. Set the PID[1:0] bits associated with the pipe to NAK.
3. Wait until the associated PBUSY bit clears to 0.
4. Set the bits in the pipe control register.

The following bits in the pipe control registers can be changed only when the selected pipe information is not set in the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

Do not set the following registers when the CURPIPE[3:0] bits are set:

- Bits in DCPCFG and DCPMAXP
- Bits in PIPECFG, PIPEMAXP and PIPEPERI

To change pipe information, you must set the CURPIPE[3:0] bits in the port select registers to a pipe other than the one to be changed. For the DCP, the buffer must be cleared using the BCLR bit in the Port Control Register after the pipe information is changed.

36.3.4.2 Transfer Types

The PIPECFG.TYPE[1:0] bits specify the following transfer types for each pipe:

- DCP: No setting is necessary (fixed at control transfer)
- Pipes 1 and 2: Set to bulk or isochronous transfer
- Pipes 3 to 5: Set to bulk transfer
- Pipes 6 to 9: Set to interrupt transfer

36.3.4.3 Endpoint Number

The PIPECFG.EPNUM[3:0] bits are used to set the endpoint number for each pipe. The DCP is fixed at endpoint 0. The other pipes can be set from endpoint 1 to 15.

- DCP: No setting is necessary (fixed at endpoint 0)
- Pipes 1 to 9: Select and set the endpoint numbers from 1 to 15 so that the combination of the PIPECFG.DIR and EPNUM[3:0] bits is unique

36.3.4.4 Maximum Packet Size Setting

Specify the maximum packet size for each pipe in the DCPMAXP.MXPS[6:0] and PIPEMAXP.MXPS[9:0] bits. The DCP and pipes 1 to 5 can be set to any of the maximum pipe sizes defined in the USB 2.0 specification. For pipes 6 to 9, the maximum packet size is 64 bytes. Set the maximum packet size as follows before starting a transfer (PID = BUF):

- DCP: Set to 8, 16, 32, or 64
- Pipes 1 to 5: Set to 8, 16, 32, or 64 for bulk transfers
- Pipes 1 and 2: Set between 1 and 256 for isochronous transfers
- Pipes 6 to 9: Set between 1 and 64

36.3.4.5 Transaction Counter for Pipes 1 to 5 in the Receiving Direction

When the specified number of transactions is complete in the data packet receiving direction, the USBFS recognizes that the transfer ended. Two transaction counters are provided: one is the PIPEnTRN register, which specifies the number of transactions to be executed, and the other is the current counter, which internally counts the number of executed transactions. If the PIPECFG.SHTNAK bit is set to 1, when the current counter value matches the specified number of transactions, the associated PIPEnCTR.PID[1:0] bits are set to NAK and the subsequent transfer is disabled. The transactions can be counted again from the beginning by initializing the current counter of the transaction counter function through the PIPEnTRE.TRCLR bit. The data read from PIPEnTRN differs depending on the PIPEnTRE.TRENB setting as follows:

- The TRENB bit = 0: Specified transaction counter value can be read
- The TRENB bit = 1: Current counter value indicating the internally counted number of executed transactions can be read

The following constraints apply when working with the TRCLR bit:

- If the transactions are being counted and PID = BUF, the current counter cannot be cleared
- If there is any data left in the buffer, the current counter cannot be cleared

36.3.4.6 Response PID

Specify the response PID for each pipe in the PID[1:0] bits in DCPCTR and PIPEnCTR. This section describes the USBFS operation with different response PID settings.

(1) Software response PID settings in host controller mode

Select the response PID to specify the execution of transactions as follows:

- NAK setting: Using pipes is disabled and no transactions are executed
- BUF setting: Transactions are executed based on the FIFO buffer state:
  - OUT direction: An OUT token is issued if the FIFO buffer contains transmit data.
  - IN direction: An IN token is issued if the FIFO buffer is not full and can receive data.
- STALL setting: Using pipes is disabled and no transactions are executed

Note: Use the DCPCTR.SUREQ bit to execute setup transactions for the DCP.

(2) Software response PID settings in device controller mode

Select the response PID to respond as follows to transactions from the host:

- NAK setting: A NAK response is returned to all generated transactions
- BUF setting: A response is returned to transactions based on the FIFO buffer
- STALL setting: A STALL response is returned to all generated transactions

Note: For setup transactions, an ACK response is always returned, regardless of the PID[1:0] bits setting, and the USB request is stored in the register.

Sections (3) and (4) describe situations in which the USBFS writes to the PID[1:0] bits because of specific transaction results.

(3) Hardware response PID settings in host controller mode

- NAK setting: PID = NAK is set in the following cases, and issuing of tokens is automatically stopped:
  - When a non-isochronous transfer is performed and an NRDY interrupt is generated

    (For details, see section 36.3.3.2. NRDY Interrupt.)
  - If a short packet is received when the PIPECFG.SHTNAK bit is set to 1 for bulk transfers
  - If transaction counting ends when the SHTNAK bit is set to 1 for bulk transfers
- BUF setting: The USBFS does not write this setting.
- STALL setting: PID = STALL is set in the following cases, and issuing of tokens is automatically stopped:
  - When STALL is received in response to a transmitted token
  - When a received data packet exceeds the maximum packet size

(4) Hardware response PID settings in device controller mode

- NAK setting: PID = NAK is set in the following cases, and a NAK response is returned to transactions:
  - When the setup token is received normally (DCP only)
  - If transaction counting ends or a short packet is received when the PIPECFG.SHTNAK bit is set to 1 for bulk transfers
- BUF setting: There is no BUF writing by the USBFS.
- STALL setting: PID = STALL is set in the following cases, and a STALL response is returned to transactions:
  - When a received data packet exceeds the maximum packet size
  - When a control transfer sequence error is detected (DCP only)

36.3.4.7 Data PID Sequence Bit

The USBFS automatically toggles the sequence bit in the data PID when data is transferred successfully in the control transfer data stage, bulk transfer, and interrupt transfer. The sequence bit of the next data PID to be transmitted can be confirmed with the SQMON bit in DCPCTR and PIPEnCTR. When data is transmitted, the sequence bit toggles on ACK handshake reception. When data is received, the sequence bit toggles on ACK handshake transmission. The SQCLR bit in DCPCTR and the SQSET bit in PIPEnCTR can be used to change the data PID sequence bit.

In device controller mode when control transfers are used, the USBFS automatically sets the sequence bit for stage transitions. DATA1 is returned when the setup stage ends. The sequence bit is not referenced and PID = DATA1 is returned in the status stage. Therefore, no software settings are required. However, in host controller mode when control transfers are used, the sequence bit must be set by software for the stage transitions.

For ClearFeature requests for transmission or reception, the data PID sequence bit must be set by software in both host and device controller modes.

36.3.4.8 Response PID = NAK Function

The USBFS provides a function for disabling pipe operation (PID response = NAK) when the final data packet of a transaction is received. The USBFS automatically distinguishes this based on reception of a short packet or the transaction counter. Enable this function by setting the PIPECFG.SHTNAK bit to 1.

When the double buffer mode is being used for the FIFO buffer, using this function enables reception of data packets in transfer units. If pipe operation is disabled, the software must enable the pipe again (PID response = BUF).

The response PID = NAK function can be used only for bulk transfers.

36.3.4.9 Auto Response Mode

For bulk transfer pipes (1 to 5), when the PIPEnCTR.ATREPM bit is set to 1, a transition is made to auto response mode. During an OUT transfer (PIPECFG.DIR = 0), OUT-NAK mode is invoked, and during an IN transfer (DIR = 1), null auto response mode is invoked.

36.3.4.10 OUT-NAK Mode

For bulk OUT transfer pipes, NAK is returned in response to an OUT token, and an NRDY interrupt is output when the PIPEnCTR.ATREPM bit is set to 1. To transition from normal mode to OUT-NAK mode, specify OUT-NAK mode while pipe operation is disabled (PID[1:0] = 00b for NAK response). Next enable pipe operation (PID[1:0] = 01b for BUF response), on which OUT-NAK mode becomes valid. If an OUT token is received immediately before pipe operation is disabled, the token data is normally received, and an ACK is returned to the host.

To transition from OUT-NAK mode to normal mode, cancel OUT-NAK mode while pipe operation is disabled (NAK). Next enable pipe operation (BUF). In normal mode, reception of OUT data is enabled.

36.3.4.11 Null Auto Response Mode

For bulk IN transfer pipes, zero-length packets are continuously transmitted when the PIPEnCTR.ATREPM bit is set to 1.

To transition from normal mode to null auto response mode, specify null auto response mode while pipe operation is disabled (response PID = NAK). Next enable pipe operation (response PID = BUF) on which null auto response mode becomes valid. Before setting null auto response mode, check that PIPEnCTR.INBUFM = 0, because the mode can be set only when the buffer is empty. If the INBUFM bit is 1, empty the buffer using the PIPEnCTR.ACLRM bit. Do not write data from the FIFO port while a transition to null auto response mode is being made.

To transition from null auto response mode to normal mode, keep pipe operation disabled (response PID = NAK) for the period of the zero-length packet transmission (about 10 µs) before canceling the null auto response mode. In normal mode, data can be written from the FIFO port, so packet transmission to the host is enabled by enabling pipe operation (response PID = BUF).

36.3.5 FIFO Buffer

The USBFS provides a FIFO buffer for data transfers, and it manages the memory area used for each pipe. The FIFO buffer has two states depending on whether the access right is assigned to the system (CPU side) or the USBFS (SIE side).

(1) Buffer status

Table 36.20 and Table 36.21 show the buffer status in the USBFS. The FIFO buffer status can be confirmed using the DCPCTR.BSTS and PIPEnCTR.INBUFM bits. The transfer direction for the FIFO buffer can be specified in either the PIPECFG.DIR or CFIFOSEL.ISEL bit (when DCP is selected).

The INBUFM bit is valid for pipes 1 to 5 in the transmitting direction.

When a transmitting pipe uses double buffering, the software can read the BSTS bit to monitor the FIFO buffer status on the CPU side and the INBUFM bit to monitor the FIFO buffer status on the SIE side. When write access to the FIFO port by the CPU or DMA/DTC is slow and the buffer empty status cannot be determined using the BEMP interrupt, the software can use the INBUFM bit to confirm the end of transmission.

Table 36.20 Buffer status indicated in the BSTS bit

ISEL or DIR	BSTS	FIFO buffer status
0 (receiving direction)	0	There is no received data, or data is being received. Reading from the FIFO port is disabled.
0 (receiving direction)	1	There is received data, or a zero-length packet is received. Reading from the FIFO port is allowed. When a zero-length packet is received, reading is not possible and the buffer must be cleared.
1 (transmitting direction)	0	Transmission has not completed. Writing to the FIFO port is disabled.
1 (transmitting direction)	1	Transmission is complete. CPU write is allowed.

Table 36.21 Buffer status indicated in the INBUFM bit

DIR	INBUFM	FIFO buffer status
0 (receiving direction)	Invalid	Invalid
1 (transmitting direction)	0	Transmission is complete. There is no data waiting to be transmitted.
1 (transmitting direction)	1	The FIFO port has written data to the buffer. There is data to be transmitted.

36.3.6 FIFO Buffer Clearing

Table 36.22 shows the methods for clearing the FIFO buffer. The FIFO buffer can be cleared using BCLR bit in the port control register, DnFIFOSEL.DCLRM, or the PIPEnCTR.ACLRM bit.

Single or double buffering can be selected for pipes 1 to 5 in the PIPECFG.DBLB bit.

Table 36.22 Buffer clearing methods (1 of 2)

FIFO buffer clearing mode	Clearing FIFO buffer on the CPU side	Mode for automatically clearing the FIFO buffer after reading the specified pipe data	Auto buffer clear mode for discarding all received packets
Register used	CFIFOCTR DnFIFOCTR	DnFIFOSEL	PIPEnCTR

Table 36.22 Buffer clearing methods (2 of 2)

FIFO buffer clearing mode	Clearing FIFO buffer on the CPU side	Mode for automatically clearing the FIFO buffer after reading the specified pipe data	Auto buffer clear mode for discarding all received packets
Bit used	BCLR	DCLRM	ACLRM
Clearing condition	Cleared by writing 1	1: Mode valid 0: Mode invalid	1: Mode valid 0: Mode invalid

(1) Auto buffer clear mode function

The USBFS discards all received data packets if the PIPEnCTR.ACLRM bit is set to 1. If a correct data packet is received, the ACK response is returned to the host controller. The auto buffer clear mode function can only be set in the FIFO buffer reading direction.

Setting the ACLRM bit to 1 and then to 0 clears the FIFO buffer of the selected pipe regardless of the access direction. An access cycle of at least 100 ns is required for the internal hardware sequence processing between ACLRM = 1 and ACLRM = 0.

36.3.7 FIFO Port Functions

Table 36.23 shows the settings for the FIFO port functions. In write access, writing data until the maximum packet size is reached automatically enables transmission of the data. To enable transmission before the maximum packet size is reached, set the BVAL flag in the port control register to end writing. To send a zero-length packet, use the BCLR bit to clear the buffer, and then set the BVAL flag to end writing.

In reading, reception of new packets is automatically enabled when all data is read. Data cannot be read when a zero-length packet is received (DTLN[8:0] = 0), so the buffer must be cleared with the BCLR bit. The length of the receive data can be confirmed in the DTLN[8:0] bits in the port control register.

Table 36.23 FIFO port function settings

Register name	Bit name	Description
CFIFOSEL, DnFIFOSEL (n = 0, 1)	RCNT	Selects DTLN[11:0] read mode
	REW	FIFO buffer rewind (re-read, rewrite)
	DCLRM	Automatically clears receive data for a specified pipe after the data is read (only for DnFIFO)
	DREQE	Enables DMA/DTC transfers (only for DnFIFO)
	MBW	FIFO port access bit width
	BIGEND	Selects FIFO port endian
	ISEL	FIFO port access direction (only for DCP)
	CURPIPE	Selects the current pipe
CFIFOCTR, DnFIFOCTR (n = 0, 1)	BVAL	Ends writing to the FIFO buffer
	BCLR	Clears the FIFO buffer on the CPU side
	DTLN	Checks the length of receive data

(1) FIFO port selection

Table 36.24 shows the pipes that can be selected with the different FIFO ports. The pipe to be accessed must be selected in the CURPIPE[3:0] bits in the port select register. After the pipe is selected, the software must check whether the written value can be read correctly from the CURPIPE[3:0] bits. (If the previous pipe number is read, it indicates that the USBFS is modifying the pipe.) Next, the software checks that the FRDY bit in the port control register is 1.

In addition, the software must specify the bus width to be accessed in the MBW bit in the port select register. The FIFO buffer access direction conforms to the PIPECFG.DIR setting. For the DCP only, the ISEL bit in the port select register determines the direction.

Table 36.24 FIFO port access by pipe

Pipe	Access method	Ports that can be used
DCP	CPU access	CFIFO port register
Pipes 1 to 9	CPU access	• CFIFO port register • D0FIFO/D1FIFO port register
	DMA/DTC access	D0FIFO/D1FIFO port register

(2) REW bit

It is possible to temporarily stop access to a pipe currently being accessed, access a different pipe, and then continue processing for the first pipe again. The REW bit in the port select register is used for this processing.

If a pipe is selected in the CURPIPE[3:0] bits in the port select register with the REW bit set to 1, the pointer used for reading from and writing to the FIFO buffer is reset, and reading or writing can be carried out from the first byte. If a pipe is selected with 0 set for the REW bit, data can be read and written in continuation from the previous selection, without the pointer being reset.

To access the FIFO port, the software must check that the FRDY bit in the port control register is 1 after selecting a pipe.

36.3.8 DMA Transfers (D0FIFO and D1FIFO Ports)

(1) Overview of DMA transfers

For pipes 1 to 9, the FIFO port can be accessed using the DMAC. When buffer access for a pipe targeted for DMA transfer is enabled, a DMA transfer request is issued.

Select the unit of transfer to the FIFO port in the DnFIFOSEL.MBW bit, and select the pipe targeted for the DMA transfer in the DnFIFOSEL.CURPIPE[3:0] bits. Do not change the selected pipe during the DMA transfer.

(2) DnFIFO auto clear mode (D0FIFO and D1FIFO port reading direction)

If 1 is set in the DnFIFOSEL.DCLRM bit, the USBFS automatically clears the FIFO buffer of the selected pipe when reading of data from the FIFO buffer is complete.

Table 36.25 shows the packet reception and FIFO buffer clearing processing by software for each of the settings.
As shown in the table, the buffer clearing conditions depend on the value set in the PIPECFG.BFRE bit. Using the DnFIFOSEL.DCLRM bit eliminates the need for the buffer to be cleared by software in any situation that requires buffer clearing. This enables DMA transfers without involving software.

The DnFIFO auto clear mode can only be set in the FIFO buffer reading direction.

Table 36.25 Packet reception and FIFO buffer clearing processing by software

Buffer status when packet is received	Register setting			
	DCLRM = 0		DCLRM = 1	
	BFRE = 0	BFRE = 1	BFRE = 0	BFRE = 1
Buffer full	No clearing required	No clearing required	No clearing required	No clearing required
Zero-length packet reception	Clearing required	Clearing required	No clearing required	No clearing required
Normal short packet reception	No clearing required	Clearing required	No clearing required	No clearing required
Transaction count end	No clearing required	Clearing required	No clearing required	No clearing required

36.3.9 Control Transfers Using the DCP

The DCP is used for data transfers in the control transfer data stage. The FIFO buffer of the DCP is a 64-byte single buffer with a fixed area for both control reads and control writes. The FIFO buffer can be accessed only through the CFIFO port.

36.3.9.1 Control Transfers in Host Controller Mode

(1) Setup stage

The USQREQ, USBVAL, USBINDX, and USBLENG registers are used to transmit USB requests for setup transactions. Writing the setup packet data to the registers and then writing 1 to the DCPCTR.SUREQ bit transmits the specified data for the setup transaction. On completion of the transaction, the SUREQ bit clears to 0. Do not change these USB request registers while SUREQ = 1.

When an attached function device is detected, the software must issue the first setup transaction for the device using this sequence with the DCPMAXP.DEVSEL[3:0] bits cleared to 0 and the DEVADD0.USBSPD[1:0] bits set appropriately.

When an attached function device is shifted to the Address state, the software must issue setup transactions using this sequence with the assigned USB address set in the DEVSEL[3:0] bits and the bits in DEVADDn corresponding to the specified USB address set appropriately. For example, when PIPEMAXP.DEVSEL[3:0] = 0010b, make appropriate settings in DEVADD2. When PIPEMAXP.DEVSEL[3:0] = 0101b, make appropriate settings in DEVADD5.

When the setup transaction data is sent, an interrupt request is generated based on the response from the peripheral device (SIGN or SACK bit in INTSTS1). This interrupt request allows the software to check the setup transaction result.

A DATA0 data packet (USB request) for the setup transaction is always transmitted regardless of the status of the DCPCTR.SQMON bit.

(2) Data stage

The data stage is used to transfer data using the DCP FIFO buffer.

Before accessing the DCP FIFO buffer, specify the access direction in the CFIFOSEL.ISEL bit. Specify the transfer direction in the DCPCFG.DIR bit.

For the first data packet of the data stage, the data PID must be transferred as DATA1. Set data PID = DATA1 in the DCPCTR.SQSET bit and set the PID bits = BUF. Completion of data transfer is detected using the BRDY or BEMP interrupt.

For control write transfers, when the number of data bytes to be sent is an integer multiple of the maximum packet size, the software must send a zero-length packet at the end.

(3) Status stage

The status stage is used for zero-length packet data transfers in the reverse direction of the data stage. As in the data stage, data is transfered using the DCP FIFO buffer. Transactions are executed using the same procedure as the data stage.

Data packets in the status stage must be transmitted and received with the data PID set to DATA1 using the DCPCTR.SQSET bit.

When a zero-length packet is received, check the receive-data length in the CFIFOCTR.DTLN[8:0] bits after a BRDY interrupt is generated, and then clear the FIFO buffer using the BCLR bit.

36.3.9.2 Control Transfers in Device Controller Mode

(1) Setup stage

The USBFS sends an ACK response to a normal setup packet for the USBFS. The USBFS operates in the setup stage as follows:

On receiving a new setup packet, the USBFS sets the following bits:

- Sets the INTSTS0.VALID bit to 1
- Sets the DCPCTR.PID[1:0] bits to NAK
- Sets the DCPCTR.CCPL bit to 0

When the USBFS receives a data packet following a setup packet, it stores the USB request parameters in USBREQ, USBVAL, USBINDX, and USBLENG.

Before performing the response processing for a control transfer, set the VALID flag to 0. When the VALID bit = 1, PID = BUF cannot be set, and the data stage cannot be terminated.

Using the VALID bit function, the USBFS can suspend a request being processed when it receives a new USB request during a control transfer and return a response to the latest request.

In addition, the USBFS automatically detects the direction bit (bmRequestType bit [8]) and the request data length (wLength) in the received USB request. It distinguishes between control read transfers, control write transfers, and no-data control transfers, and it controls stage transitions. For an incorrect sequence, a sequence error occurs in the control transfer stage transition interrupt, and the interrupt is reported to the software. For a diagram of the stage control by the USBFS, see Figure 36.15.

(2) Data stage

The DCP must be used to execute data transfers for received USB requests. Before accessing the DCP FIFO buffer, specify the access direction in the CFIFOSEL.ISEL bit.

If the transfer data is larger than the size of the DCP FIFO buffer, execute the data transfer using the BRDY interrupt for control write transfers and the BEMP interrupt for control read transfers.

(3) Status stage

Control transfers are terminated by setting the DCPCTR.CCPL bit to 1 while the DCPCTR.PID[1:0] bits are set to BUF.

After this setting is made, the USBFS automatically executes the status stage based on the data transfer direction determined at the setup stage. The procedure is as follows:

- For control read transfers

  The USBFS receives a zero-length packet from the USB host and transmits an ACK response.
- For control write transfers and no-data control transfers

  The USBFS transmits a zero-length packet and receives an ACK response from the USB host.

(4) Control transfer auto response function

The USBFS automatically responds to a correct SET_ADDRESS request. If any of the following errors occurs in the SET_ADDRESS request, a response from the software is necessary.

- bmRequestType is not 0x00: Any transfer other than a control write transfer
- wIndex is not 0x00: Request error
- wLength is not 0x00: Any transfer other than a no-data control transfer
- wValue is larger than 0x7F: Request error
- INTSTS0.DVSQ[2:0] are 011b (Configured state): Control transfer of a device state error

For all requests other than the SET_ADDRESS request, a response is required from the corresponding software.

36.3.10 Bulk Transfers (Pipes 1 to 5)

The FIFO buffer usage (single/double buffer setting) is configurable for bulk transfers. The USBFS provides the following functions for bulk transfers:

- BRDY interrupt function (PIPECFG.BFRE bit), see section 36.3.3.1. BRDY Interrupt
- Transaction count function (PIPEnTRE.TRENB, TRCLR, and PIPEnTRN.TRNCNT[15:0] bits), see section 36.3.4.5. Transaction Counter for Pipes 1 to 5 in the Receiving Direction
- Response PID = NAK function (PIPECFG.SHTNAK bit), see section 36.3.4.8. Response PID = NAK Function
- Auto response mode (PIPEnCTR.ATREPM bit), see section 36.3.4.9. Auto Response Mode

36.3.11 Interrupt Transfers (Pipes 6 to 9)

In device controller mode, the USBFS performs interrupt transfers based on the timing dictated by the host controller.

In host controller mode, the software can set the timing for issuing tokens using the interval counter.

36.3.11.1 Interval Counter for Interrupt Transfers in Host Controller Mode

Specify the transaction interval for interrupt transfers in the PIPEPERI.IITV[2:0] bits. The USBFS issues interrupt transfer tokens based on this interval.

(1) Counter initialization

The USBFS initializes the interval counter under the following conditions:

- Power-on reset

  This initializes the IITV[2:0] bits.

- FIFO buffer initialization using the PIPEnCTR.ACLRM bit:

  This does not initialize the IITV[2:0] bits, but does initialize the count value. Setting the PIPEnCTR.ACLRM bit to 0 starts counting from the value set in IITV[2:0].

The interval counter is not initialized in the following case:

- USB bus reset or USB suspended

  The IITV[2:0] bits are not initialized. Setting 1 to the DVSTCTR0.UACT bit starts counting from the value saved before entering the USB bus reset state or USB suspend state.

(2) Operation when tokens cannot be transmitted or received even on token generation

No token is generated in the following cases even at token generation time. In these cases, the USBFS tries to execute the transaction in the next interval.

- When the PID is set to NAK or STALL
- When the FIFO buffer is full at token transmit time in the receiving (IN) direction
- When there is no data to be transmitted in the FIFO buffer at token transmit time in the transmitting (OUT) direction

36.3.12 Isochronous Transfers (Pipes 1 and 2)

The USBFS provides the following functions for isochronous transfers:

- Notification of isochronous transfer error
- Interval counter specified in the PIPEPERI.IITV[2:0] bits
- Isochronous IN transfer data setup control (IDLY function)
- Isochronous IN transfer buffer flush function specified in the PIPEPERI.IFIS bit

36.3.12.1 Error Detection in Isochronous Transfers

The USBFS provides a function for detecting the errors described in this section, so that when errors occur in isochronous transfers, they can be controlled by software. Table 36.26 and Table 36.27 show the priority order for errors detected by the USBFS and the associated interrupts.

PID errors

- The PID value of the received packet is invalid.

CRC errors and bit stuffing errors

- A CRC error is found in a received packet or the bit stuffing is illegal.

Maximum packet size exceeded

- The data size of the received packet exceeds the specified maximum packet size.

Overrun and underrun errors

In host controller mode:

- The FIFO buffer is full at token transmit time in the IN (receiving) direction
- There is no data to be sent in the FIFO buffer at token transmit time in the OUT (transmitting) direction

In device controller mode:

- There is no data to be sent in the FIFO buffer at token receive time in the IN (transmitting) direction
- The FIFO buffer is full at token receive time in the OUT (receiving) direction

Interval errors

In device controller mode, the following cases are treated as an interval error:

- Failure to receive an IN token in the interval frame during an isochronous IN transfer
- Failure to receive an OUT token in the interval frame during an isochronous OUT transfer

Table 36.26 Error detection for token transmission and reception

Detection priority	Error	Generated interrupt and status
1	PID error	No interrupts are generated in either host or device controller mode (ignored as a corrupted packet)
2	CRC or bit stuffing error	No interrupts are generated in either host or device controller mode (ignored as a corrupted packet)
3	Overrun or underrun error	An NRDY interrupt is generated to set the FRMNUM.OVRN bit to 1 in both host and device controller modes. In device controller mode, a zero-length packet is transmitted in response to an IN token. No data packets are received in response to OUT token.
4	Interval error	An NRDY interrupt is generated in device controller mode. No interrupt is generated in host controller mode.

Table 36.27 Error detection for data packet reception

Detection priority	Error	Generated interrupt and status
1	PID error	No interrupts are generated (ignored as a corrupted packet)
2	CRC or bit stuffing error	An NRDY interrupt is generated and the FRMNUM.CRCE bit sets to 1 in both host and device controller modes
3	Maximum packet size exceeded error	A BEMP interrupt is generated and the PID[1:0] bits set to STALL in both host and device controller modes

36.3.12.2 DATA-PID

In device controller mode, the USBFS responds to a received PID as follows:

(1) IN direction

- DATA0: Transmitted as data packet PID
- DATA1: Not transmitted
- DATA2: Not transmitted
- mDATA: Not transmitted

(2) OUT direction

- DATA0: Received normally as data packet PID
- DATA1: Received normally as data packet PID
- DATA2: Packets ignored
- mDATA: Packets ignored

36.3.12.3 Interval Counter

The isochronous transfer interval can be set in the PIPEPERI.IITV[2:0] bits. In device controller mode, the interval counter enables the functions as shown in Table 36.28. In host controller mode, the USBFS generates the token issuance timing, and the interval counter operation is the same as that for interrupt transfers.

Table 36.28 Interval counter functions in device controller mode

Transfer direction	Function	Conditions for detection
IN	Transmit buffer flush	Failure to receive an IN token successfully in the interval frame during an isochronous IN transfer.
OUT	Notification of no reception of token	Failure to receive an OUT token successfully in the interval frame during an isochronous OUT transfer.

The interval count is performed when an SOF is received or for interpolated SOFs, so the isochronism can be maintained even if an SOF is corrupt. The frame interval can be set to 2^{IITV} frames.

(1) Counter initialization in device controller mode

The USBFS initializes the interval counter under the following conditions:

- Power-on reset

  This initializes the PIPEPERI.IITV[2:0] bits.
- FIFO buffer initialization using the ACLRM bit

  This does not initialize the IITV[2:0] bits, but does initialize the count value.

After the interval counter is initialized, the interval count starts under one of the following conditions when a packet is transferred successfully:

- An SOF is received after data is transmitted in response to an IN token when PID = BUF
- An SOF is received after data is received in response to an OUT token when PID = BUF

The interval counter is not initialized in the following conditions:

- When the PID[1:0] bits are set to NAK or STALL

  This does not stop the interval timer. The USBFS attempts the transaction in the next interval.
- When the USB bus is reset or USBFS is suspended

  This does not initialize the IITV[2:0] bits. When an SOF is received, the interval counter starts counting from the value set before SOF was received.

(2) Interval counting and transfer control in host controller mode

The USBFS controls the interval between token issuance operations based on the PIPEPERI.IITV[2:0] bits settings. Specifically, the USBFS issues a token for a selected pipe once every 2^{IITV} frames.

The USBFS starts counting the token issuance interval at the frame following the frame in which the PID[1:0] bits are set to BUF by software.

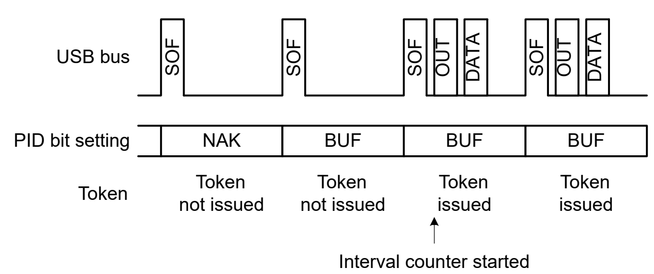


Figure 36.16 Token issuance when IITV = 0

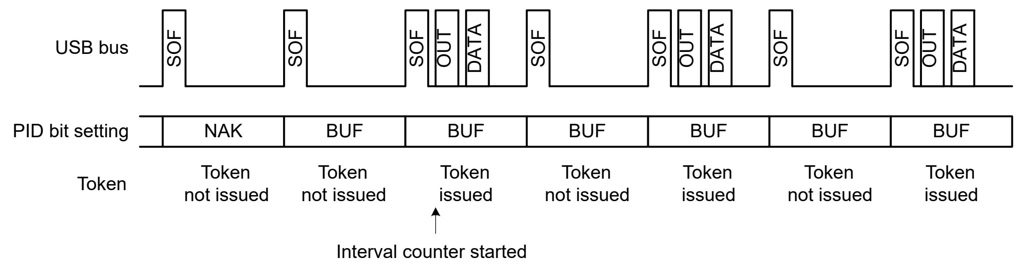


Figure 36.17 Token issuance when IITV = 1

When the selected pipe is set for isochronous transfers, the USBFS performs the following operation in addition to controlling the token issuance interval. The USBFS issues a token even when the NRDY interrupt generation condition is satisfied.

When the selected pipe is for isochronous IN transfers

The USBFS generates an NRDY interrupt when the USBFS issues an IN token but does not successfully receive a packet from a peripheral device (no response or packet error).

The USBFS sets the FRMNUM.OVRN bit to 1, generating an NRDY interrupt, when the time to issue an IN token occurs while the USBFS cannot receive data because the FIFO buffer is full, because the CPU or DMAC/DTC is too slow in reading data from the FIFO buffer.

When the selected pipe is for isochronous OUT transfers

The USBFS sets the OVRN bit to 1, generating an NRDY interrupt and transmitting a zero-length packet, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer, or because the CPU or DMAC/DTC is too slow in writing data to the FIFO buffer.

The token issuance interval is reset on any of the following conditions:

- When the USBFS is reset through a reset pin

  This initializes the IITV[2:0] bits.
- When the PIPEnCTR.ACLRM bit is set to 1 by software

(3) Interval counting and transfer control in device controller mode

When the selected pipe is for isochronous OUT transfers

The USBFS generates an NRDY interrupt when it fails to receive a data packet within the interval set in the PIPEPERI.IITV[2:0] bits.

The USBFS also generates an NRDY interrupt when it fails to receive data because of a CRC error or other errors contained in the data packet or because the FIFO buffer is full.

The NRDY interrupt is generated on SOF packet reception. Even if the SOF packet is corrupted, internal interpolation allows the interrupt to be generated when the SOF packet is received. However, when the IITV bits are set to a value other than 0, the USBFS generates an NRDY interrupt on receiving an SOF packet for every interval after interval counting starts.

When the PID[1:0] bits are set to NAK by software after starting the interval timer, the USBFS does not generate an NRDY interrupt on receiving an SOF packet.

The timing for starting interval counting depend on the IITV[2:0] setting as follows:

- When the IITV[2:0] bits = 0:

  Interval counting starts at the next frame after the software changes the PID[1:0] bits of the selected pipe to BUF.
- When the IITV[2:0] bits ≠ 0:

  Interval counting starts on completion of successful reception of the first data packet after the PID[1:0] bits for the selected pipe are changed to BUF.

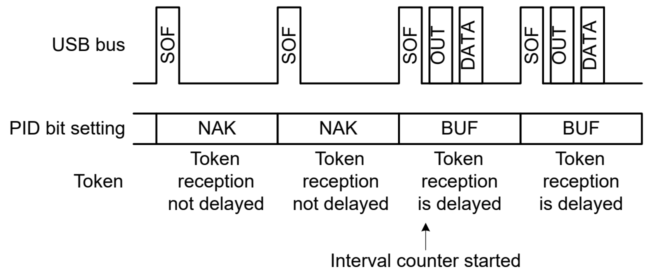


Figure 36.18 Relationship between frames and expected token reception when IITV[2:0] = 0

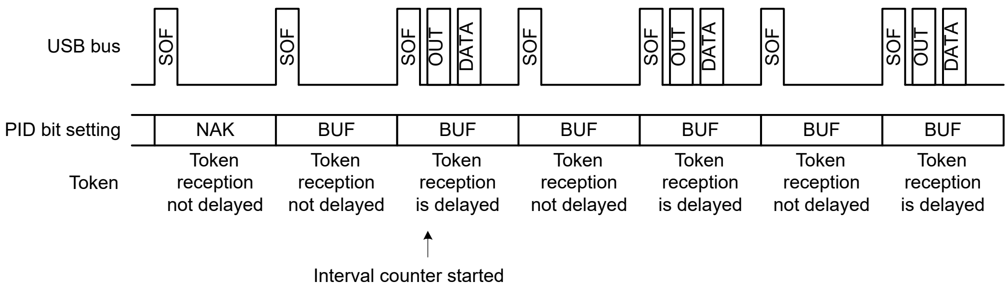


Figure 36.19 Relationship between frames and expected token reception when IITV[2:0] ≠ 0

When the selected pipe is for isochronous IN transfers

The PIPEPERI.IFIS bit must be 1 for this use case. When IFIS = 0, the USBFS transmits a data packet in response to a received IN token regardless of the PIPEPERI.IITV[2:0] setting.

When IFIS is 1 and there is data to be transmitted in the FIFO buffer, the USBFS clears the FIFO buffer when it fails to receive an IN token in the frame at the interval set in the IITV[2:0] bits.

The USBFS also clears the FIFO buffer when it fails to receive an IN token successfully because of a bus error, such as a CRC error, contained in the IN token.

The FIFO buffer is cleared on SOF packet reception. Even if the SOF packet is corrupted, the internal interpolation allows the FIFO buffer to be cleared when the SOF packet is received.

The timing to start interval counting depends on the IITV[2:0] setting, as with OUT transfers.

The interval is counted on any of the following conditions in device controller mode:

- When a hardware reset is applied to the USBFS (which also sets the IITV[2:0] bits to 000b)
- When the PIPEnCTR.ACLRM bit is set to 1 by software
- When the USBFS detects a USB bus reset

(4) Transmit data setup for isochronous transfers in device controller mode

With isochronous data transmission using the USBFS in device controller mode, after data is written to the FIFO buffer, a data packet can be transmitted in the first frame after the SOF packet is detected. This isochronous transfer transmit data setup function can identify the frame that started transmission.

When the double buffering is used, transmission is only enabled for the buffer where data writing was completed first, even after the data write to both buffers is complete. Accordingly, even if multiple IN tokens are received, only the one packet of FIFO buffer data is transmitted.

When the FIFO buffer is ready to transmit data when an IN token is received, the data is transferred and a normal response is returned. However, if the FIFO buffer cannot transmit data, a zero-length packet is transmitted and an underrun error occurs.

Figure 36.20 shows an example transmission using the isochronous transfer transmission data setup function when IITV = 0 (every frame) is set.

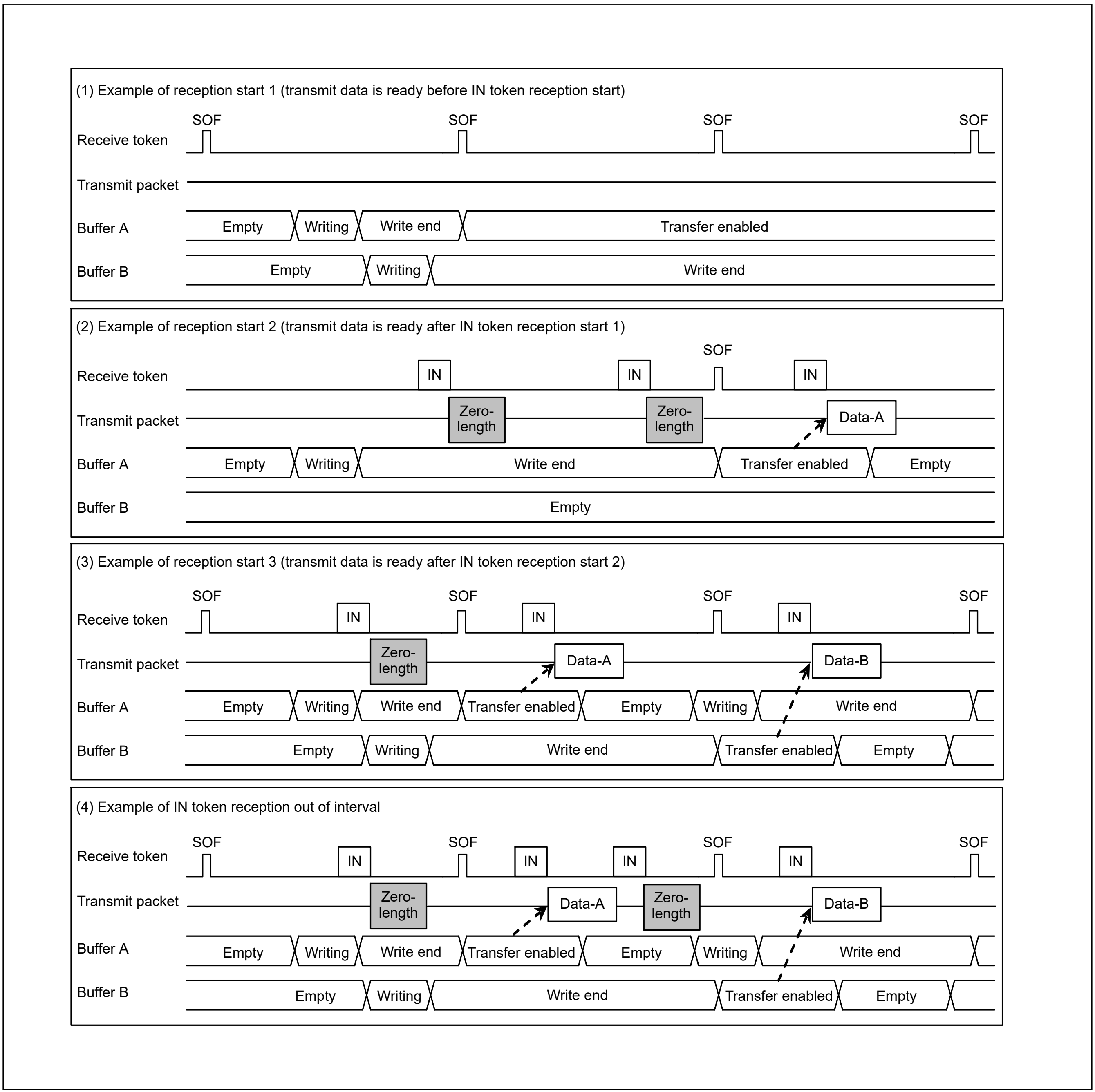


Figure 36.20 Example data setup operation

(5) Transmit buffer flush for isochronous transfers in device controller mode

In device controller mode during isochronous data transmission, if the USBFS receives an SOF packet for the next frame without receiving an IN token in the interval frame, it operates as if the IN token is corrupt and clears the buffer that is enabled for transmission, putting that buffer in the writing enabled state.

When double buffering is used and writing to both buffers is complete, the cleared FIFO buffer is assumed to be the one where the data was transmitted in the interval frame, and transmission is enabled for the FIFO buffer that was not cleared on SOF packet reception.

The timing of the buffer flush function depends on the PIPEPERI.IITV[2:0] setting as follows:

- When IITV = 0:

  The buffer flush operation starts from the first frame after the pipe is enabled.

- When IITV ≠ 0:

  The buffer flush operation starts after the first normal transaction.

Figure 36.21 shows an example buffer flush. When an unanticipated token is received before the interval frame, the USBFS sends the write data or a zero-length packet as an underrun error, depending on the data setup status.

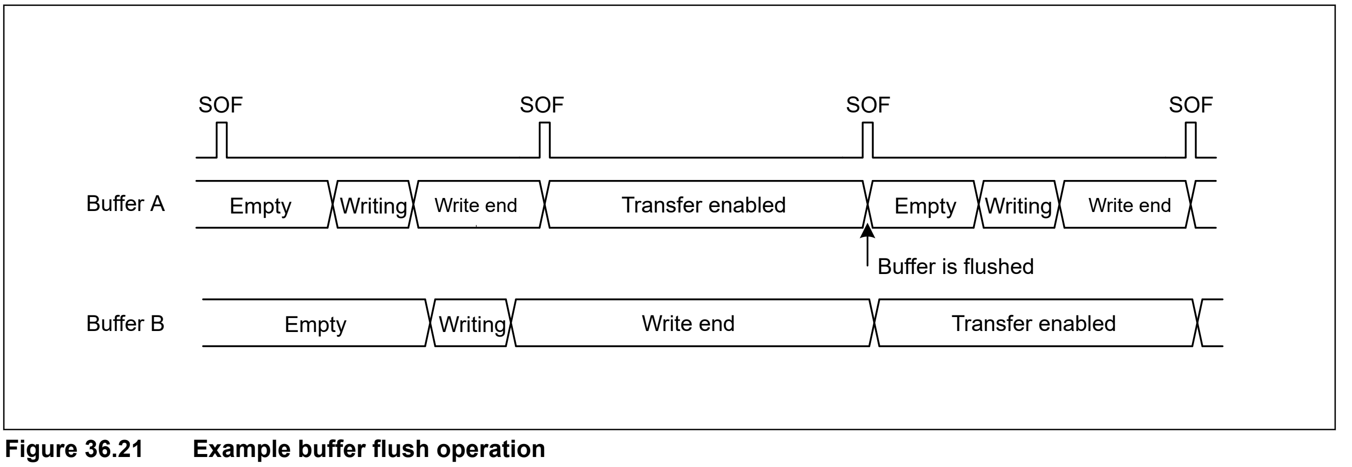


Figure 36.21 Example buffer flush operation

Figure 36.22 shows an example interval error occurrence. There are five types of interval errors, as shown in the figure. An interval error occurs at timing (A), and the buffer flush function is activated.

If an interval error occurs during an IN transfer, the buffer flush function is activated. If it occurs during an OUT transfer, an NRDY interrupt is generated. Use the FRMNUM.OVRN bit to distinguish between this and NRDY interrupts triggered by received packet errors and overrun errors.

For tokens that are shaded in the figure, responses are returned based on the FIFO buffer status.

- IN direction:
  - If the buffer is ready to transfer data, the data is transferred and a normal response is returned
  - If the buffer is not ready to transfer data, a zero-length packet is transmitted and an underrun error occurs
- OUT direction:
  - If the buffer is ready to receive data, the data is received and a normal response is returned
  - If the buffer is not ready to receive data, the received data is discarded and an overrun error occurs

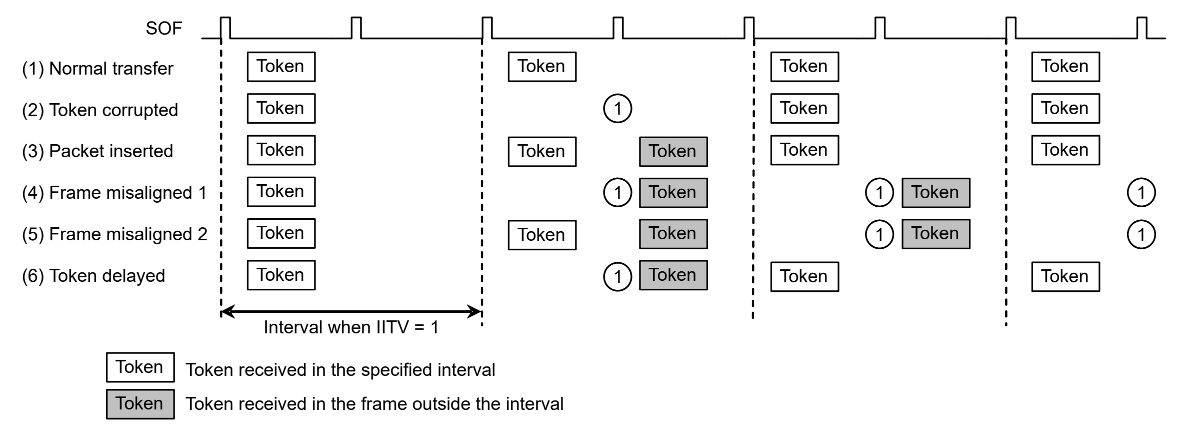


Figure 36.22 Example interval error occurrence when IITV = 1

36.3.13 SOF Interpolation Function

In device controller mode, if packet reception is disabled at intervals of 1 ms because the SOF packet is corrupted or missing, the USBFS interpolates the SOF. SOF interpolation begins when the USBE and SCKE bits in SYSCFG are set to 1 and an SOF packet is received.

The interpolation function is initialized under the following conditions:

- MCU reset
- USB bus reset
- Suspend state detection

The SOF interpolation operates as follows:

- The interpolation function is not activated until an SOF packet is received
- When the first SOF packet is received, interpolation is performed by counting 1 ms on the 48-MHz internal clock
- When the second and subsequent SOF packets are received, interpolation is performed at the previous reception interval
- Interpolation is not performed in the Suspend state or on reception of a USB bus reset

The USBFS supports the following functions controlled by SOF packet reception. These functions operate normally with SOF interpolation if the SOF packet is missing:

- Updating of the frame number
- SOFR interrupt timing
- Isochronous transfer interval count

If an SOF packet is missing during full-speed operation, the FRMNUM.FRNM[10:0] bits are not updated.

36.3.14 Pipe Schedule

36.3.14.1 Conditions for Generating Transactions

In host controller mode and when the DVSTCTR0.UACT bit is set to 1, the USBFS generates transactions under the conditions shown in Table 36.29.

Table 36.29 Conditions for generating transactions

Transaction	Conditions for generation				
	DIR	PID	IITV0	Buffer state	SUREQ
Setup	—*1	—*1	—*1	—*1	1 setting
Control transfer data stage, status stage, bulk transfer	IN	BUF	Invalid	Receive area exists	—*1
	OUT	BUF	Invalid	Transmit data exists	—*1
Interrupt transfer	IN	BUF	Valid	Receive area exists	—*1
	OUT	BUF	Valid	Transmit data exists	—*1
Isochronous transfer	IN	BUF	Valid	*2	—*1
	OUT	BUF	Valid	*3	—*1

Note 1. An em dash (—) in the table indicates that the condition is unrelated to the generating of tokens. "Valid" indicates that, for interrupt transfers and isochronous transfers, a transaction is generated only in transfer frames that are based on the interval counter. "Invalid" indicates that a transaction is generated regardless of the interval counter.
Note 2. This indicates that a transaction is generated regardless of whether there is a receive area. If there is no receive area, however, the received data is discarded.
Note 3. This indicates that a transaction is generated regardless of whether there is any data to be transmitted. If there is no data to be transmitted, however, a zero-length packet is transmitted.

36.3.14.2 Transfer Schedule

This section describes the transfer scheduling within a frame of the USBFS. After the USBFS sends an SOF, the transfer is carried out in the following sequence:

1. Execution of periodic transfers:

   A pipe is searched for in the order of pipe 1 → pipe 2 → pipe 6 → pipe 7 → pipe 8 → pipe 9, and then if there is a pipe for which an isochronous or interrupt transfer transaction can be generated, the transaction is generated.

2. Setup transactions for control transfers:

   The DCP is checked, and if a setup transaction is possible, it is sent.

3. Execution of bulk transfers, control transfer data stages, and control transfer status stages:

   A pipe is searched for in the order of DCP → pipe 1 → pipe 2 → pipe 3 → pipe 4 → pipe 5, and then if there is a pipe for which a transaction for a bulk transfer, a control transfer data stage, or a control transfer status stage can be generated, the transaction is generated.

   When a transaction is generated, processing moves to the next pipe transaction regardless of whether the response from the peripheral device is ACK or NAK. If there is time for transfer within the frame, step 3 is repeated.

36.3.14.3 Enabling USB Communication

Setting the DVSTCTR0.UACT bit to 1 initiates an SOF transmission, and transaction generation is enabled. Setting the UACT bit to 0 stops SOF transmission and the Suspend state is invoked. If the UACT setting is changed from 1 to 0, processing stops after the next SOF is sent.

36.4 Usage Notes

36.4.1 Settings for the Module-Stop State

USBFS operation can be disabled or enabled using Module Stop Control Register B (MSTPCRB). The USBFS is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Mode.

36.4.2 Clearing the Interrupt Status Register on Canceling Software Standby Mode

Because the input buffer is always enabled in Software Standby mode, an unexpected interrupt might occur under the following conditions:

- When the interrupt is enabled in Normal mode
- When the interrupt is disabled in Software Standby mode
- When the input level of the pin that cancels software standby mode is changed in Software Standby mode

These conditions might cause the associated interrupt flag in the Interrupt Status Register to set unexpectedly. After the MCU cancels the Software Standby mode, the unexpected interrupt might be sent to the interrupt controller. To avoid this, always clear the INTSTS0 and INTSTS1 registers in the canceling sequence.

36.4.3 Clearing the Interrupt Status Register after Setting Up the Port Function

The input buffer is disabled before the PmnPFS.PSEL and PmnPFS.PMR port is set up, so the internal signal is fixed high or low. The input buffer is enabled after the port is set so that the external pin state is propagated to the MCU. An unexpected interrupt might occur at this time, causing the VBINT and OVRCR bits in INTSTS0 and INTSTS1, or other interrupt status flags to set to 1. To avoid a malfunction, always clear the INTSTS0 and INTSTS1 registers after setting up the port.

36.4.4 Setting Up the Port Function Before Setting USB Function

USB_DP, USB_DM pins are compatible with P814, P815 I/O ports. So before setting USBFS registers to enable USBFS function, set register bit P814PFS.PMR and P815PFS.PMR to 1.

37. USB 2.0 High-Speed Module (USBHS)

37.1 Overview

The MCU provides a USB 2.0 High-Speed Module (USBHS) that operates as a host or a device controller compliant with the Universal Serial Bus (USB) Specification revision 2.0. The host controller supports USB 2.0 high-speed, full-speed, and low-speed transfers, and the device controller supports USB 2.0 high-speed and full-speed transfers. The USBHS has an internal USB transceiver and supports all of transfer types defined in the USB 2.0 specification.

The USBHS has FIFO buffer for data transfers, providing a maximum of 10 pipes. Any endpoint number can be assigned to pipes 1 to 9, based on the peripheral devices or the communication requirements for your system.

Table 37.1 lists the USBHS specifications, Figure 37.1 shows a block diagram, and Table 37.2 lists the I/O pins.

Table 37.1 USBHS specifications

Parameter	Specifications
Features	● USB Device Controller (UDC) and USB 2.0 transceiver supporting host controller, device controller, and On-The-Go (OTG) functions ● Software can switch between host and device controller modes
	Host controller features: ● High-speed transfer (480 Mbps), full-speed transfer (12 Mbps), and low-speed transfer (1.5 Mbps) ● Automatic scheduling for SOF and packet transmissions ● Programmable intervals for isochronous and interrupt transfers ● Communications with multiple peripheral devices connected through a single hub
	Device controller features: ● High-speed transfer (480 Mbps) and full-speed transfer (12 Mbps) ● Control transfer stage control function ● Device state control function ● Auto response function for SET_ADDRESS request ● SOF complementation
Supported transfer types	● Control transfer ● Bulk transfer ● Interrupt transfer ● Isochronous transfer
Pipe configuration	● FIFO buffer of up to 8.5 KB for USB communications ● Up to 10 pipes selectable, including the default control pipe ● Programmable pipe configurations ● Pipes 1 to 9 assignable to any endpoint number
	Transfer conditions specifiable for each pipe: ● Pipe 0: Control transfer with 64-byte single buffer ● Pipes 1 and 2: Bulk isochronous transfer continuous transfer mode with programmable buffer size up to 2 KB and optional double buffer ● Pipes 3 to 5: Bulk transfer continuous transfer mode with programmable buffer size up to 2 KB and optional double buffer ● Pipes 6 to 9: Interrupt transfer with 64-byte single buffer
Other features	● Force-end transfer function using transaction count ● Function that changes the BRDY interrupt event notification timing ● Automatic clearing of the FIFO buffer after data for the pipe specified in the DnFIFO port (n = 0, 1) is read ● NAK setting function for response PID generated on transfer end ● On-chip pull-up and pull-down resistors for D+ and D- ● Support for Link Power Management (LPM) ECN, including a new Sleep state (the L1 state) ● Compliance with Battery Charging Class Specification Revision 1.2 ● For power reduction, selectable classic-only mode (CL-only mode) in which operation is only USB 1.1-compliant
Module-stop function	Module-stop state can be set to reduce power consumption
TrustZone Filter	Security and Privilege attribution can be set

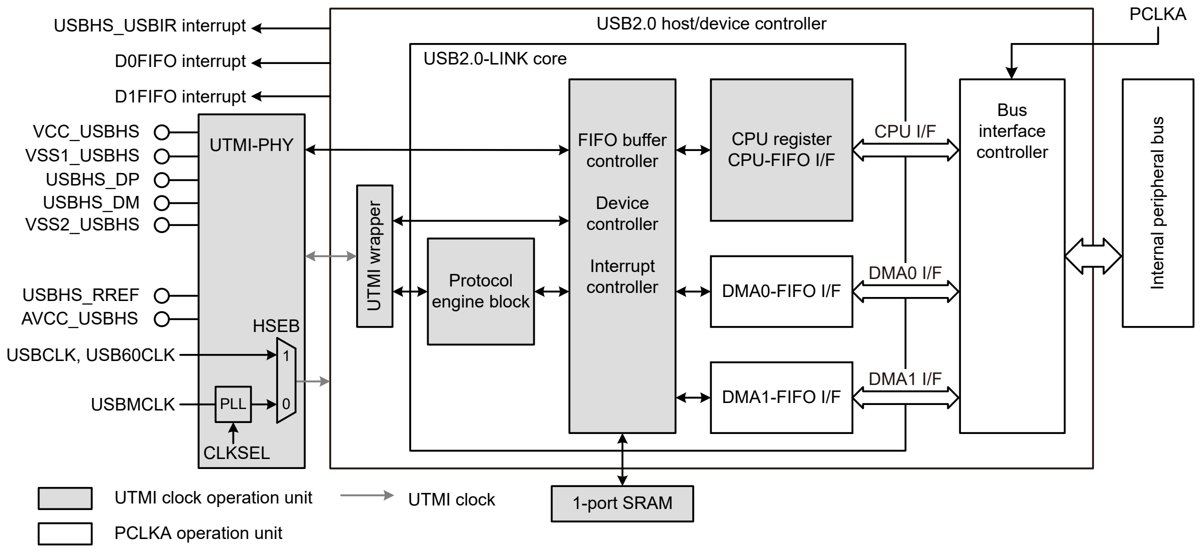


Figure 37.1 USBHS block diagram

Table 37.2 USBHS I/O pins

Pin name	I/O	Function
VCC_USBHS	Input	Power supply pin for the USBHS
VSS1_USBHS VSS2_USBHS	Input	Ground pin for the USBHS
AVCC_USBHS	Input	Analog power supply pin for the USBHS
USBHS_RREF	I/O	Reference current source pin for the USBHS Must be connected to the VSS2_USBHS pin through a 2.2-kΩ (±1%) resistor.
USBHS_DP	I/O	Input/output pin for the D+ data line of the USB bus
USBHS_DM	I/O	Input/output pin for the D- data line of the USB bus
USBHS_EXICEN	Output	Must be connected to the OTG power supply IC
USBHS_ID	Input	Must be connected to the OTG power supply IC
USBHS_VBUSEN	Output	VBUS power supply enable pin for the USBHS
USBHS_OVRCURA, USBHS_OVRCURB, USBHS_OVRCURA-DS, USBHS_OVRCURB-DS	Input	Overcurrent pin for the USBHS USBHS_OVRCURA or USBHS_OVRCURB pins can be used in Software standby mode or in normal mode. USBHS_OVRCURA-DS, USBHS_OVRCURB-DS are dedicated pins that can generate interrupt in Software standby mode or in normal mode as well as for cancel Deep Software Standby mode 1.
USBHS_VBUS	Input	USB cable connection monitor input pin

37.2 Register Descriptions

37.2.1 SYSCFG : System Configuration Control Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x000

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CNEN	HSE	DCFM	DRPD	DPRPU	—	—	—	USBE
Value after reset:	x	x	x	x	x	x	x	0	0	0	1	0	x	x	x	0

Bit	Symbol	Function	R/W
0	USBE	USBHS Operation Enable 0: Disable 1: Enable	R/W
3:1	—	The read values are undefined. The write value should be 0.	R/W
4	DPRPU	D+ Line Resistor Control 0: Disable line pull-up 1: Enable line pull-up	R/W
5	DRPD	D+/D- Line Resistor Control 0: Disable line pull-down 1: Enable line pull-down	R/W
6	DCFM	Controller Operation Select 0: Select device controller mode 1: Select host controller mode	R/W
7	HSE	High-Speed Operation Enable 0: Disable Device controller mode: full-speed Host controller mode: full-speed or low-speed 1: Enable The controller detects the communication speed	R/W
8	CNEN	Single-ended Receiver Enable 0: Disable 1: Enable	R/W
15:9	—	The read values are undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Writing to the SYSCFG register can proceed while the PHY clock is stopped. However, written values are only reflected in the SYSCFG register after the PHY clock is oscillating again.

USBE bit (USBHS Operation Enable)

The USBE bit enables or disables operation of USBHS.

Changing the USBE bit from 1 to 0 initializes the bits listed in Table 37.3. Only change this bit after specifying the input clock in the PHYSET.CLKSEL[1:0] bits and confirming that the PLLSTA.PLLLOCK flag is 1. In CL-only mode, change the USBE bit after setting the PHYSET.HSEB bit to 1. At that time, the USBCLK must be set to 48 MHz and USB60CLK must be set to 60 MHz. For the clock settings, see section 37.3.3. Supplying the Clock.

In host controller mode, always set this bit to 1 after setting the DRPD bit to 1, eliminating SYSSTS0.LNST[1:0] bit chattering, and confirming that the USB bus state is stable.

Table 37.3 Bits initialized by writing SYSCFG.USBE = 0

Selected function	Register	Bit	Remarks
Device controller (DCFM = 0)	SYSSTS0	LNST[1:0]	Value is saved in host controller mode
	DVSTCTR0	RHST[2:0]	—
	PL1CTRL1	DVSQ[3:0]	Value is saved in host controller mode
	USBADDR	USBADDR[6:0]	Value is saved in host controller mode
	USBREQ	• BREQUEST[7:0] • BMREQUESTTYPE [7:0]	Value is saved in host controller mode
	USBVAL	WVALUE[15:0]	Value is saved in host controller mode
	USBINDX	WINDEX[15:0]	Value is saved in host controller mode
	USBLENG	WLENTUH[15:0]	Value is saved in host controller mode
Host controller (DCFM = 1)	DVSTCTR0	RHST[2:0]	—
	FRMNUM	FRNM[10:0]	Value is saved in device controller mode
	UFRMNUM	UFRNM[2:0]	Value is saved in device controller mode

DPRPU bit (D+ Line Resistor Control)

The DPRPU bit enables or disables pulling up the D+ line in device controller mode.

When the DPRPU bit is set to 1 in device controller mode, the USBHS pulls up the D+ line to notify the USB host that it attached. Changing the DPRPU bit from 1 to 0 releases the pull-up, thereby notifying the USB host that it detached.

Set this bit to 1 in device controller mode and to 0 in host controller mode.

DRPD bit (D+/D- Line Resistor Control)

The DRPD bit enables or disables pulling down D+ and D- lines in host controller mode.

Set this bit to 1 in host controller mode. Set it to 0 when OTG is not used in device controller mode.

DCFM bit (Controller Operation Select)

The DCFM bit selects the host or device function of the USBHS.

Only change this bit when the DPRPU and DRPD bits are both 0.

HSE bit (High-Speed Operation Enable)

The HSE bit enables or disables high-speed operation.

When this bit is 1, the USBHS operates in high-speed or full-speed based on the results of the reset handshake.

In host controller mode, setting this bit to 0 allows the USBHS to operate in full-speed or low-speed. If the DVSTCTR0.RHST[2:0] flags indicate that a low-speed device has attached, set the HSE bit to 0.

In host controller mode, setting this bit to 1 allows the USBHS to operate in high-speed or full-speed based on the results of the reset handshake. Change the HSE bit after detection of an attach event (ATTCH interrupt) and before the USB bus reset (when DVSTCTR0.USBRST = 1), or after detection of a detach event.

In device controller mode, setting this bit to 0 allows the USBHS to operate in full-speed. Setting the bit to 1 allows the USBHS to perform the reset handshake and then operate in high-speed or full-speed, based on the results.

In device controller mode, only change this bit when the DPRPU bit is 0.

CNEN bit (Single-ended Receiver Enable)

Setting the CNEN bit to 1 enables single-ended receiver operation and selects monitoring of the D+ and D- line states in the SYSSTS0.LNST[1:0] flags. Use this bit to prevent through-current damage that might otherwise be caused during single-ended receiver operation, where the terminals are floating while the USBHS is detached.

In host controller mode, set this bit to 1 after confirming that the PHY clock is being supplied. In device controller mode, set this bit to 1 when the VBUS is detected because of a VBUS interrupt, and set it to 0 when the VBUS line is removed.

37.2.2 BUSWAIT : CPU Bus Wait Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x002

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	BWAIT[3:0]			
Value after reset:	x	x	0	0	0	0	0	0	x	x	0	0	1	1	1	1

Bit	Symbol	Function	R/W
3:0	BWAIT[3:0]	CPU Bus Access Wait Specification 0x0: 0 waits (2 access cycles) : 0x2: 2 waits (4 access cycles) : 0x4: 4 waits (6 access cycles) : 0xF: 15 waits (17 access cycles) (initial value)	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
7:6	—	The read values are undefined. The write value should be 0.	R/W
13:8	—	These bits are read as 0. The write value should be 0.	R/W
15:14	—	The read values are undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

BWAIT[3:0] bits (CPU Bus Access Wait Specification)

The BWAIT[3:0] bits specify the wait time for access to the USBHS registers.

When accessing the registers at offset addresses in range 0x004 to 0x15F for writing and range 0x004 to 0x167 for reading, set the cycle time for consecutive access to at least 41 ns. The initial value is 0xF (17 cycles), but Renesas recommends that you satisfy this condition by setting the best wait time for the frequency of PCLKA in your application.

This setting is the same as the wait time for accesses to the FIFO port register. The maximum speed of access to the FIFO port is as follows:

- MBW[1:0] = 10b (32-bit width): Maximum 60 MB/s
- MBW[1:0] = 01b (16-bit width): Maximum 30 MB/s
- MBW[1:0] = 00b (8-bit width): Maximum 15 MB/s

37.2.3 SYSSTS0 : System Configuration Status Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x004

Bit position:	15		13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OVCMON[1:0]		—	—	—	—	—	—	—	HTACT	SOFEA	—	—	IDMON	LNST[1:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	0	0	x	x	x	x	x

Bit	Symbol	Function	R/W
1:0	LNST[1:0]	USB Data Line Status Monitor Flag Indicates the status of the USB data lines. See Table 37.4.	R
2	IDMON	USBHS_ID Pin Monitor Flag 0: USBHS_ID pin is low 1: USBHS_ID pin is high	R

Bit	Symbol	Function	R/W
4:3	—	The read values are undefined.	R
5	SOFEA	SOF Active Monitor Flag While Host Controller Operation Is Selected 0: SOF output stopped 1: SOF output operating	R
6	HTACT	Host Sequencer Status Monitor Flag 0: Host sequencer stopped 1: Host sequencer operating	R
13:7	—	The read values are undefined.	R
15:14	OVCMON[1:0]	External USBHS_OVRCURA or USBHS_OVRCURA-DS, and USBHS_OVRCURB or USBHS_OVRCURBB-DS Input Pin Monitor Flag OVCMON[1] indicates the USBHS_OVRCURA or USBHS_OVRCURA-DS pin status. OVCMON[0] indicates the USBHS_OVRCURB or USBHS_OVRCURB-DS pin status.	R

Note: S-TYPE-3, P-TYPE-3

LNST[1:0] flags (USB Data Line Status Monitor Flag)

The LNST[1:0] flags indicate the state of the USB data lines, D+ and D-. For details, see Table 37.4.

In device controller mode, read the LNST[1:0] flags after setting the SYSCFG.CNEN and SYSCFG.USBE bits to 1. In host controller mode, read them after setting the SYSCFG.DRPD bit to 1.

When you are checking hardware contacts for the battery charging function in device controller mode, read the LNST[1:0] flags after setting the SYSCFG.DRPD, SYSCFG.CNEN, and BCCTRL.IDPSRCE bits to 1. For details, see section 37.3.15. Battery Charging Detection Processing.

Table 37.4 Status of USB data bus lines (D+ and D-)

LNST[1]	LNST[0]	Low-speed operation (host controller mode only)	Full-speed operation	High-speed operation	Chirp operation
0	0	SE0	SE0	Squelch	Squelch
0	1	K-State	J-State	Unsquelch	Chirp J
1	0	J-State	K-State	Invalid	Chirp K
1	1	SE1	SE1	Invalid	Invalid

Chirp: The reset handshake protocol is being executed when high-speed operation is enabled (HSE bit is 1).

Squelch: SE0 or idle state

Unsquelch: High-speed J-state or high-speed K-state

Chirp J: Chirp J-State

Chirp K: Chirp K-State

SOFEA flag (SOF Active Monitor Flag While Host Controller Operation Is Selected)

The SOFEA flag is used in host controller mode to check whether the output of the last SOF is complete when the USBHS is suspended because of a 0 setting to the DVSTCTR0.UACT bit.

In host controller mode, check that both the HTACT and SOFEA flags are 0 before setting the SYSCFG.USBE bit to 0 to stop the USBHS or setting the LPSTS.SUSPENDM bit to 0 to stop the clock signal supply during communication.

HTACT flag (Host Sequencer Status Monitor Flag)

The HTACT flag clears to 0 when the host sequencer of the USBHS is completely stopped.

In host controller mode, check that the HTACT flag is 0 before setting the DVSTCTR0.UACT bit to 0 to place the USBHS in the Suspend state or setting the LPSTS.SUSPENDM bit to 0 to stop the clock signal supply during communication.

OVCMON[1:0] flags (External USBHS_OVRCURA or USBHS_OVRCURA-DS, and USBHS_OVRCURB or USBHS_OVRCURBB-DS Input Pin Monitor Flag)

The OVCMON[1:0] flags indicate the status of the overcurrent signals from an external power supply IC.

37.2.4 PLLSTA : PLL Status Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x006

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLLLO CK
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	PLLLOCK	PLL Lock Flag 0: PLL not locked 1: PLL locked	R
15:1	—	The read values are undefined.	R

Note: S-TYPE-3, P-TYPE-3

PLLLOCK flag (PLL Lock Flag)

The PLLLOCK flag indicates whether the USB-PHY internal PLL is locked. When not using CL-only mode, make sure that the PLL is locked before starting USB communication.

37.2.5 DVSTCTR0 : Device State Control Register 0

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x008

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2		0
Bit field:	—	—	—	—	HNPB TOA	EXICE N	VBUS EN	WKUP	RWUP E	USBR ST	RESU ME	UACT	—	RHST[2:0]		
Value after reset:	x	x	x	x	0	0	0	0	0	0	0	0	x	0	0	0

Bit	Symbol	Function	R/W
2:0	RHST[2:0]	USB Bus Reset Status Flag 0 0 0: Communication speed indeterminate (powered state or no connection) 0 0 1: Host controller mode Low-speed connection 0 1 0: Host controller mode Full-speed connection Device controller mode USB bus reset in progress or full-speed connection 0 1 1: Host controller mode High-speed connection Device controller mode USB bus reset in progress or high-speed connection Others: Host controller mode USB bus reset in progress Device controller mode Setting prohibited	R
3	—	The read value is undefined. The write value should be 0.	R/W
4	UACT	USB Bus Operation Enable for the Host Controller Operation 0: Disable downstream port (disable SOF or micro-SOF transmission) 1: Enable downstream port (enable SOF or micro-SOF transmission)	R/W
5	RESUME	Resume Signal Output for the Host Controller Operation 0: Do not output resume signal 1: Output resume signal	R/W

Bit	Symbol	Function	R/W
6	USBRST	USB Bus Reset Output for the Host Controller Operation 0: Do not output USB bus reset signal 1: Output USB bus reset signal	R/W
7	RWUPE	Remote Wakeup Detection Enable for the Host Controller Operation 0: Disable downstream port remote wakeup 1: Enable downstream port remote wakeup	R/W
8	WKUP	Remote Wakeup Output for the Device Controller Operation 0: Do not output remote wakeup signal 1: Output remote wakeup signal	R/W
9	VBUSEN	USBHS_VBUSEN Output Pin Control 0: Output low on external USBHS_VBUSEN pin 1: Output high on external USBHS_VBUSEN pin	R/W
10	EXICEN	USBHS_EXICEN Output Pin Control 0: Output low on external USBHS_EXICEN pin 1: Output high on external USBHS_EXICEN pin	R/W
11	HNPBTOA	Host Negotiation Protocol (HNP) Control Use this bit when switching from device B to device A in OTG mode. If the HNPBTOA bit is 1, the internal function control remains in the Suspend state until the HNP processing ends even if SYSCFG.DPRPU = 0 or SYSCFG.DCFM = 1 is set.	R/W
15:12	—	The read values are undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

RHST[2:0] flags (USB Bus Reset Status Flag)

The RHST[2:0] flags indicate the USB bus reset status.

In host controller mode, writing 1 to the USBRST bit causes the RHST[2:0] flags to set to 100b. When 0 is written to the USBRST bit and the USBHS ends the SE0 state, the RHST[2:0] flags update to a new value.

In device controller mode, if the USBHS detects a USB bus reset, the RHST[2:0] flags set to 010b if an attach event occurs while the DPRPU bit is 1, and a DVST interrupt is generated.

UACT bit (USB Bus Operation Enable for the Host Controller Operation)

When set to 1 in host controller mode, the UACT bit enables USB bus operation by controlling SOF packet transmission to the USB bus in addition to data and reception. The USBHS starts SOF packet output within one frame period after this bit is set to 1. If UACT is set to 0, the USBHS enters the idle state after the SOF packet output.

The USBHS sets the bit to 0 on any of the following conditions:

- A DTCH interrupt is detected during communication (while UACT = 1)
- An EOFERR interrupt is detected during communication (while UACT = 1)

Always write 1 to the UACT bit at the end of the USB bus reset processing (on a 0 write to the USBRST bit) or at the end of resume processing from the Suspend state (on a 0 write to the RESUME bit).

The USBHS clears the UACT bit to 0 if it receives an ACK response to an LPM token while the HL1CTRL1.L1REQ bit is set to 1. The USBHS sets the UACT bit to 1 when it finishes resume processing from the L1 state.

In device controller mode, always set this bit to 0.

RESUME bit (Resume Signal Output for the Host Controller Operation)

The RESUME bit controls the resume signal output in host controller mode. When this bit is set to 1, the USBHS drives the USB port to the K-state and outputs the resume signal. The USBHS sets the bit to 1 on detection of a remote wakeup signal while the RWUPE bit is 1 and in the USB suspend state. The USBHS continues outputting the K-state while the RESUME bit is 1, until the bit is cleared to 0 by software. The RESUME bit must be 1 (= resume period) for the time defined in the USB 2.0 specification. Only set this bit to 1 while the interface is in the Suspend state. Write 1 to the UACT bit simultaneously with the end of the resume processing (0 write to the RESUME bit).

Setting the RESUME bit to 1 during transition to the L1 state allows the USBHS to drive the USB port to the K-state and output the resume signal. The USBHS clears the RESUME bit to 0 at the end of the resume period, the value set in the HL1CTRL2.HIRD[3:0] bits.

Always set this bit to 0 in device controller mode.

USBRST bit (USB Bus Reset Output for the Host Controller Operation)

The USBRST bit controls the output of the USB bus signal in host controller mode. When this bit set to 1, the USBHS drives the USB port to the SE0 state to reset the USB bus. The USBHS continues outputting SE0 while the USBRST bit is 1, until the bit is cleared to 0 by software. The USBRST bit must be 1 (= USB bus reset period) for the time defined in the USB 2.0 specification. Writing 1 to the USBRST bit during communication (UACT bit = 1) or during resume processing (RESUME bit = 1) prevents the USBHS from starting USB bus reset processing until both the UACT and RESUME bits clear to 0. Write 1 to the UACT bit simultaneously with the end of the USB bus reset processing (0 write to the USBRST bit).

Always set this bit to 0 in device controller mode.

RWUPE bit (Remote Wakeup Detection Enable for the Host Controller Operation)

The RWUPE bit enables or disables remote wakeup signals (resume signals) from downstream peripheral devices in host controller mode. When this bit is set to 1, the USBHS detects a remote wakeup signal (K-state for 2.5 µs) from a downstream peripheral device, and it performs resume processing, driving the K-state. When the RWUPE bit is set to 0, the USBHS ignores remote wakeup signals (K-states) from peripheral devices connected to the USB port.

Do not stop the PHY clock while the RWUPE bit is 1, even in the Suspend state (the LPSTS.SUSPENDM bit must be set to 1). Also, do not reset the USB bus (setting USBRST to 1) from the Suspend state. This is prohibited in the USB 2.0 specification.

The RWUPE bit is also used to enable or disable detection of a remote wakeup signal during transition to the L1 state.

Always set this bit to 0 in device controller mode.

WKUP bit (Remote Wakeup Output for the Device Controller Operation)

The WKUP bit enables or disables remote wakeup signals (resume signals) to the USB bus in device controller mode.

The USBHS controls the output timing of the remote wakeup signals. When this bit is set to 1, the USBHS clears it to 0 after outputting the K-state for 10 ms. The USB 2.0 specification dictates that the USB bus idle state must be maintained for 5 ms or longer before a remote wakeup signal is sent. If the USBHS writes 1 to the WKUP bit immediately after detecting the Suspend state, the K-state is output after 2 ms.

Only write 1 to the WKUP bit when the device is in the Suspend state (the PL1CTRL1.DVSQ[3:0] flags are 01xxb) and the USB host enables the remote wakeup signal (RWUPE = 1). Do not stop the PHY clock while this bit is 1, even in the Suspend state (the LPSTS.SUSPENDM bit must be set to 1).

If the WKUP bit is set to 1 during transition to the L1 state, the USBHS outputs the K-state for 50 µs and then clears the bit to 0. Before writing 1 to the bit during the L1 state, check that the PL1CTRL1.DVSQ[3:0] flags are 10xxb.

Always set this bit to 0 in host controller mode.

HNPBTOA bit (Host Negotiation Protocol (HNP) Control)

The HNPBTOA bit is used when switching from device B to device A while in OTG mode.

If the HNPBTOA bit is 1, the internal function control maintains the Suspend state until HNP processing ends, even if the SYSCFG.DPRPU bit is set to 0 or the SYSCFG.DCFM bit is set to 1. Resume interrupts (RESM) are not generated even if a falling edge of D+ is detected.

The HNP processing ends when a host attach event is detected, because of a pull-up by the initiating party, or the HNPBTOA bit is cleared to 0 by software because the HNP processing times out.

37.2.6 TESTMODE : USB Test Mode Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x00C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	UTST[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	UTST[3:0]	Test Mode These bits output the USB test signals. See Table 37.5	R/W
15:4	—	The read values are undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

UTST[3:0] bits (Test Mode)

Writing values to the UTST[3:0] bits allows the USBHS to output USB test signals in high-speed operation mode. Table 37.5 shows the test mode operation settings.

Table 37.5 Test mode operation settings

Test mode	UTST[3:0] bit setting	
	In device controller mode	In host controller mode
Normal operation	0x0	0x0
Test_J	0x1	0x9
Test_K	0x2	0xA
Test_SE0_NAK	0x3	0xB
Test_Packet	0x4	0xC
Test_Force_Enable	—	0xD
Reserved	0x5 to 0x7	0xE to 0xF

Host controller mode

In host controller mode, these bits can be set after setting the SYSCFG.DRPD bit to 1. After the UTST[3:0] bits are set, the USBHS outputs waveforms to the USB port by setting the DVSTCTR0.UACT bit to 1. The USBHS also performs high-speed termination for the USB port by setting these bits in host controller mode.

To set the UTST[3:0] bits in host controller mode:

1. Reset the hardware.
2. Start supplying the PHY clock, and then set the LPSTS.SUSPENDM bit to 1.
3. Set the SYSCFG.DCFM and SYSCFG.DRPD bits to 1. (Setting the SYSCFG.HSE bit to 1 is not required.)
4. Set the SYSCFG.USBE bit to 1.
5. Set the UTST[3:0] bits based on the test requirements.
6. Set the DVSTCTR0.UACT bit to 1.

Assuming the initial steps (1) to (6) are already complete, to change the UTST[3:0] bits in host controller mode:

1. Set the DVSTCTR0.UACT and SYSCFG.USBE bits to 0.
2. Set the SYSCFG.USBE bit to 1.
3. Set the UTST[3:0] bits based on the test requirements.
4. Set the DVSTCTR0.UACT bit to 1.

When the UTST[3:0] bits are set to 1011b (Test_SE0_NAK), the USBHS does not output SOF packets to ports for which the DVSTCTR0.UACT bit is set to 1.

When the UTST[3:0] bits are set to 1101b (Test_Force_Enable), the USBHS outputs SOF packets to ports for which the DVSTCTR0.UACT bit is set to 1. In this test mode, the USBHS does not control the hardware related to attach detection, even if it detects a high-speed detach event (DTCH interrupt).

Before setting the UTST[3:0] bits, set the PID[1:0] bits of all pipe control registers to 00b (NAK response). To return to normal USB communication after setting a test mode, issue a hardware reset.

Device controller mode

In device controller mode, set these bits using a SetFeature request from the USB host during high-speed communication. The USBHS does not enter the Suspend state while these bits are 0001b to 0100b. To return to normal USB communication after setting a test mode, issue a hardware reset.

37.2.7 CFIFO, DnFIFO : FIFO Port Register (n = 0, 1)

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x014 (CFIFO/CFIFOL/CFIFOLL)
0x016 (CFIFOH)
0x017 (CFIFOHH)
0x018 + 0x04 × n (DnFIFO/DnFIFOL/DnFIFOLL)
0x01A + 0x04 × n (DnFIFOH)
0x01B + 0x04 × n (DnFIFOHH)

Bit position:	31 … 0
Bit field:	FIFOPORT[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	FIFOPORT[31:0]*1	Read receive data from the FIFO buffer or write transmit data to the FIFO buffer by accessing these bits.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The valid bits depend on the MBW[1:0] and BIGEND settings in the associated port selection register.

Three FIFO ports are provided:

- CFIFO
- D0FIFO
- D1FIFO

Each FIFO port is configured with:

- A port register (CFIFO, D0FIFO, or D1FIFO) that handles reading of data from the FIFO buffer and writing of data to the FIFO buffer
- A port selection register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) that selects the pipe assigned to the FIFO port
- A port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR)

Each FIFO port has the following constraints:

- Access to the FIFO buffer for DCP control transfers is through the CFIFO port
- Access to the FIFO buffer for DMAC or DTC transfers is through the D0FIFO or D1FIFO port
- The D0FIFO and D1FIFO ports can also be accessed by the CPU
- When using functions specific to the FIFO port, such as the DMAC or DTC transfer function, you cannot change the pipe number selected in the CURPIPE[3:0] bits of the Port Selection Register
- Registers configuring one FIFO port do not affect other FIFO ports
- The same pipe must not be assigned to two or more FIFO ports

- There are two FIFO buffer states, one giving access rights to the CPU and the other to the serial interface engine (SIE). When the SIE has access rights, the FIFO buffer cannot be accessed by the CPU

FIFOPORT bits (FIFOPORT[31:0])

When the FIFOPORT bits are accessed, the USBHS reads the received data from the FIFO buffer or writes the transmission data to the FIFO buffer. The FIFO port register can be accessed only when the FRDY flag in the associated port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1.

The valid bits in the FIFO port register depend on the MBW[1:0] and BIGEND settings in the port selection register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL). See Table 37.6 to Table 37.8.

Table 37.6 Endian operation in 32-bit access (MBW[1:0] = 10b)

BIGEND	CFIFO, D0FIFO, D1FIFO b31 to b24	CFIFO, D0FIFO, D1FIFO b23 to b16	CFIFO, D0FIFO, D1FIFO b15 to b8	CFIFO, D0FIFO, D1FIFO b7 to b0	Remarks
0	Located at N+3	Located at N+2	Located at N+1	Located at N+0	Transmit data is sent from the address N+0. Receive data is stored from the address N+0.
1	Located at N+0	Located at N+1	Located at N+2	Located at N+3	Transmission data is sent from the address N+3. Receive data is stored from the address N+3.

Table 37.7 Endian operation in 16-bit access (MBW[1:0] = 01b)

BIGEND	CFIFOL, D0FIFOL, D1FIFOL b15 to b8	CFIFOL, D0FIFOL, D1FIFOL b7 to b0	CFIFOH, D0FIFOH, D1FIFOH b15 to b8	CFIFOH, D0FIFOH, D1FIFOH b7 to b0	Remarks
0	Access prohibited*1		Located at N+1	Located at N+0	Transmit data is sent from the address N+0. Receive data is stored from the address N+0.
1	Located at N+0	Located at N+1	Access prohibited*1		Transmit data is sent from the address N+1. Receive data is stored from the address N+1.

Note 1. Writing to or reading from these areas is prohibited.

Table 37.8 Endian operation in 8-bit access (MBW[1:0] = 00b)

BIGEND	CFIFOLL, D1FIFOLL, D0FIFOLL	CFIFOHH, D1FIFOHH, D0FIFOHH
0	Access prohibited*1	Located at N+0
1	Located at N+0	Access prohibited*1

Note 1. Writing to or reading from these locations is prohibited.

37.2.8 CFIFOSEL : CFIFO Port Selection Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x020

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RCNT	REW	—	—	MBW[1:0]		—	BIGEND	—	—	ISEL	—	CURPIPE[3:0]			
Value after reset:	0	0	x	x	0	0	x	0	x	x	0	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CURPIPE[3:0]	FIFO Port Access Pipe Specification 0x0: DCP (default control pipe) 0x1: Pipe 1 0x2: Pipe 2 0x3: Pipe 3 0x4: Pipe 4 0x5: Pipe 5 0x6: Pipe 6 0x7: Pipe 7 0x8: Pipe 8 0x9: Pipe 9 Others: Setting prohibited	R/W
4	—	The read value is undefined. The write value should be 0.	R/W
5	ISEL	FIFO Port Access Direction when DCP Is Selected 0: Select reading from the FIFO buffer 1: Select writing to the FIFO buffer	R/W
7:6	—	The read values are undefined. The write value should be 0.	R/W
8	BIGEND	FIFO Port Endian Control 0: Little endian 1: Big endian	R/W
9	—	The read value is undefined. The write value should be 0.	R/W
11:10	MBW[1:0]	CFIFO Port Access Bit Width 0 0: 8-bit width 0 1: 16-bit width 1 0: 32-bit width 1 1: Setting prohibited	R/W
13:12	—	The read values are undefined. The write value should be 0.	R/W
14	REW	Buffer Pointer Rewind 0: Do not rewind buffer pointer (Writing 0 has no effect.) 1: Rewind buffer pointer	W
15	RCNT	Read Count Mode 0: Clear DTLN[11:0] flags in the FIFO port control register to 0x000 when all receive data is read from CFIFO 1: Decrement DTLN[11:0] flags each time receive data is read from CFIFO	R/W

Note: S-TYPE-3, P-TYPE-3

Do not specify the same pipe number in the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers.

Do not change the pipe number while DMAC or DTC transfer is enabled.

CURPIPE[3:0] bits (FIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number used to read or write data through the CFIFO port. After writing to these bits, read them to check that the written value agrees with the read value before proceeding to the next process. Do not set the same pipe number to the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

During FIFO buffer access, the pipe specification is maintained until the access is complete, even if the software attempts to change the CURPIPE[3:0] setting. Access continues after the current value is written back to the CURPIPE[3:0] bits.

ISEL bit (FIFO Port Access Direction when DCP Is Selected)

After writing a new value to the ISEL bit while the DCP is the selected pipe, read this bit to check that the written value agrees with the read value before proceeding to the next process. Set the ISEL and CURPIPE[3:0] bits simultaneously.

BIGEND bit (FIFO Port Endian Control)

Use the BIGEND bit to set the byte endian order of the CFIFO port.

MBW[1:0] bits (CFIFO Port Access Bit Width)

The MBW[1:0] bits specify the bit width for accessing the CFIFO port.

When the selected pipe is receiving, after a write to these bits starts a data read from the FIFO buffer, do not change the bits until all of the data is read. When reading the FIFO buffer, read with the access size set in MBW.

When the selected pipe is transmitting, set the CURPIPE[3:0] and MBW[1:0] bits simultaneously. The bit width cannot be changed from 8-bit to 16- or 32-bit, or from 16-bit to 32-bit while data is being written to the FIFO buffer.

An odd number of bytes can also be written through byte-access control even when 16- or 32-bit width is selected.

REW bit (Buffer Pointer Rewind)

The REW bit specifies whether or not to rewind the buffer pointer.

When the selected pipe is receiving, setting this bit to 1 while the FIFO buffer is being read allows re-reading of the FIFO buffer from the first data. In double-buffering when reading is already in progress, this setting enables reading either FIFO buffer from the first entry.

Do not set this bit to 1 while simultaneously changing the CURPIPE[3:0] bits. Before setting the bit to 1, always check that the FRDY flag is 1.

To rewrite to the FIFO buffer from the first data for the transmitting pipe, use the BCLR bit.

RCNT bit (Read Count Mode)

When the RCNT bit set to 0, the USBHS clears the CFIFOCTR.DTLN[11:0] flags to 0 on finishing reading all of the received data in the FIFO buffer assigned to the pipe specified in the CURPIPE[3:0] bits, or after reading a single plane in double buffer mode.

With this bit set to 1, the USBHS decrements the value in the CFIFOCTR.DTLN[11:0] flags each time it reads data received from the FIFO buffer assigned to the pipe specified in the CURPIPE[3:0] bits.

37.2.9 DnFIFOSEL : DnFIFO Port Selection Register (n = 0, 1)

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x028 + 0x4 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RCNT	REW	DCLRM	DREQE	MBW[1:0]		—	BIGEND	—	—	—	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	x	0	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CURPIPE[3:0]	FIFO Port Access Pipe Specification 0x0: No pipe specification 0x1: Pipe 1 0x2: Pipe 2 0x3: Pipe 3 0x4: Pipe 4 0x5: Pipe 5 0x6: Pipe 6 0x7: Pipe 7 0x8: Pipe 8 0x9: Pipe 9 Others: Setting prohibited	R/W

Bit	Symbol	Function	R/W
7:4	—	The read values are undefined. The write value should be 0.	R/W
8	BIGEND	FIFO Port Endian Control 0: Little endian 1: Big endian	R/W
9	—	The read value is undefined. The write value should be 0.	R/W
11:10	MBW[1:0]	FIFO Port Access Bit Width 0 0: 8-bit width 0 1: 16-bit width 1 0: 32-bit width 1 1: Setting prohibited	R/W
12	DREQE	DMAC/DTC Transfer Request Enable 0: Disable DMAC/DTC transfer request 1: Enable DMAC/DTC transfer request	R/W
13	DCLRM	Auto FIFO Buffer Clear Mode after Specified Pipe is Read 0: Disable auto buffer clear mode 1: Enable auto buffer clear mode	R/W
14	REW	Buffer Pointer Rewind 0: Do not rewind buffer pointer (writing 0 has no effect) 1: Rewind buffer pointer	W
15	RCNT	Read Count Mode 0: Clear DTLN[11:0] flags in the FIFO port control register to 0x000 when all receive data is read from DnFIFO (after read of a single plane in double buffer mode) 1: Decrement DTLN[11:0] flags each time receive data is read from DnFIFO	R/W

Note: S-TYPE-3, P-TYPE-3

Do not specify the same pipe number in the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected.

Do not change the pipe number while DMAC or DTC transfer is enabled.

CURPIPE[3:0] bits (FIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number used to read or write data through the DnFIFO port. After writing to these bits, read them to check that the written value agrees with the read value before proceeding to the next process. Do not set the same pipe number to the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

During FIFO buffer access, the pipe specification is maintained until the access is complete, even if the software attempts to change the CURPIPE[3:0] setting. Access continues after the current value is written back to the CURPIPE[3:0] bits.

BIGEND bit (FIFO Port Endian Control)

Use the BIGEND bit to set the byte endian order of the D0FIFO or D1FIFO port.

MBW[1:0] bits (FIFO Port Access Bit Width)

The MBW[1:0] bits specify the bit width for accessing the DnFIFO port.

When the selected pipe is receiving, after a write to these bits starts a data read from the FIFO buffer, do not change the bits until all of the data is read. When reading the FIFO buffer, read with the access size set in MBW.

When the selected pipe is transmitting, set the CURPIPE[3:0] and MBW[1:0] bits simultaneously. The bit width cannot be changed from 8-bit to 16- or 32-bit, or from 16-bit to 32-bit while data is being written to the FIFO buffer.

An odd number of bytes can also be written through byte-access control even when 16- or 32-bit width is selected.

DREQE bit (DMAC/DTC Transfer Request Enable)

The DREQE bit enables or disables issuing of DMAC or DTC transfer requests. Only change the settings of DREQE bit when the CURPIPE[3:0] bits are 0x0.

To enable DMAC or DTC transfer requests, set this bit to 1 after setting the CURPIPE[3:0] bits to 0x0, and then set the CURPIPE[3:0] bits to the PIPE number for the transfer.

DCLRM bit (Auto FIFO Buffer Clear Mode after Specified Pipe is Read)

The DCLRM bit enables or disables automatic FIFO buffer clearing after data in the selected pipe is read.

When this bit is set to 1, on receiving a zero-length packet while the FIFO buffer assigned to the selected pipe is empty, or when reading of a received short packet is complete while the PIPECFG.BFRE bit is 1, the USBHS sets the BCLR bit in the FIFO port control register to 1.

When using the USBHS with the SOFCFG.BRDYM bit set to 1, set this bit to 0.

REW bit (Buffer Pointer Rewind)

The REW bit specifies whether or not to rewind the buffer pointer.

When the selected pipe is receiving, setting this bit to 1 while the FIFO buffer is being read allows re-reading of the FIFO buffer from the first data. In double-buffering when reading is already in progress, this setting enables reading either FIFO buffer from the first entry.

Do not set this bit to 1 while simultaneously changing the CURPIPE[3:0] bits. Before setting the bit to 1, always check that the FRDY flag is 1.

To rewrite to the FIFO buffer from the first data for the transmitting pipe, use the BCLR bit.

RCNT bit (Read Count Mode)

When the RCNT bit set to 0, the USBHS clears the DnFIFOCTR.DTLN[11:0] flags (n = 0, 1) to 0 on finishing reading all of the received data in the FIFO buffer assigned to the pipe specified in the CURPIPE[3:0] bits, or after reading a single plane in double buffer mode.

With this bit set to 1, the USBHS decrements the value in the CFIFOCTR.DTLN[11:0] flags each time it reads data received from the FIFO buffer assigned to the pipe specified in the CURPIPE[3:0] bits. When accessing DnFIFO with the PIPECFG.BFRE bit set to 1, set the RCNT bit to 0.

37.2.10 CFIFOCTR, DnFIFOCTR : FIFO Port Control Register (n = 0, 1)

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x022 (CFIFOCTR)
0x02A + 0x4 × n (DnFIFOCTR)

Bit position:	15	14	13	12	11											0
Bit field:	BVAL	BCLR	FRDY	—	DTLN[11:0]											
Value after reset:	0	0	0	x	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	DTLN[11:0]	Receive Data Length Flag The meaning of the values differs depending on the RCNT bit setting in the port selection register. For details, see the description of the DTLN[11:0] bits.	R
12	—	The read value is undefined. The write value should be 0.	R/W
13	FRDY	FIFO Port Ready Flag 0: FIFO port access disabled 1: FIFO port access enabled	R
14	BCLR	CPU Buffer Clear 0: No operation (writing 0 has no effect) 1: Clear FIFO buffer on the CPU side	W
15	BVAL	FIFO Buffer Valid Flag Set this bit to 1 when data is completely written to the FIFO buffer on the CPU side for the selected pipe (CURPIPE[3:0] setting). 0: Invalid (writing 0 has no effect) 1: Writing ended	R/W

Note: S-TYPE-3, P-TYPE-3

The CFIFOCTR, D0FIFOCTR, and D1FIFOCTR registers correspond to the CFIFO, D0FIFO, and D1FIFO buffers.

DTLN[11:0] flags (Receive Data Length Flag)

The DTLN[11:0] flags indicate the length of the receive data.

While the FIFO buffer is being read, the DTLN[11:0] bits indicate different values depending on the DnFIFOSEL.RCNT bit (n = 0, 1), as follows:

- RCNT = 0:
  The USBHS sets the DTLN[11:0] flags to indicate the length of the receive data until the CPU or DMAC/DTC has read all of the received data in the FIFO buffer (or until it has read a single plane in double buffer mode).
  While the PIPECFG.BFRE bit is 1, the USBHS retains the length of the receive data until the BCLR bit is set to 1, even after all the data is read.
- RCNT = 1:
  The USBHS decrements the value indicated in the DTLN[11:0] flags each time the CPU or DMAC/DTC reads the receive data from the FIFO buffer. (The value is decremented by 1 when MBW[1:0] = 00b, by 2 when MBW[1:0] = 01b, and by 4 when MBW[1:0] = 10b.)
  The USBHS sets these flags to 0 when all the data is read from the FIFO buffer. In double buffer mode, if data is received in one FIFO buffer plane before all of the data is read from the other plane, the USBHS sets these bits to indicate the length of the receive data in the latter plane when all of the data is read from the former plane.
  When the RCNT bit is 1, reading the DTLN[11:0] flags while the FIFO buffer is being read returns the latest value within 150 ns after the FIFO port read cycle.

FRDY flag (FIFO Port Ready Flag)

The FRDY flag indicates whether the FIFO port can be accessed by the CPU or DMAC/DTC.

In the following cases, the USBHS sets the FRDY flag to 1 but data cannot be read through the FIFO port because there is no data to be read:

- A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty
- A short packet is received and the data is completely read while the PIPECFG.BFRE bit is 1

In these cases, set the BCLR bit to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.

BCLR bit (CPU Buffer Clear)

Set the BCLR bit to 1 to clear the FIFO buffer on the CPU for the selected pipe.

When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USBHS clears only one plane of the FIFO buffer even when both planes are read-enabled.

When the DCP is the selected pipe, setting the BCLR bit to 1 allows the USBHS to clear both sets of FIFO buffers regardless of whether the CPU or SIE has access rights. To clear the buffer when the SIE has access rights, set the DCPCTR.PID[1:0] bits to 00b (NAK response) before setting the BCLR bit to 1.

When the selected pipe is not the DCP, only write 1 to the BCLR bit while the FRDY flag in the FIFO port control register is 1 (set by the USBHS).

BVAL bit (FIFO Buffer Valid Flag)

Set the BVAL bit to 1 when data is completely written to the FIFO buffer on the CPU for the pipe selected in CURPIPE[3:0].

When the selected pipe is transmitting, set this bit to 1 in the following cases:

- To transmit a short packet, set this bit to 1 after data is written
- To transmit a zero-length packet, set this bit to 1 before data is written to the FIFO buffer
- Set this bit to 1 after the specified number of data bytes is written for the pipe in continuous transfer mode, where the number is a natural integer multiple of the maximum packet size and less than the buffer size

The USBHS then switches the FIFO buffer from the CPU side to the SIE side, enabling transmission.

When the selected pipe is in use for transmission, simultaneously setting the BVAL flag and the BCLR bit to 1 causes the USBHS to clear the data that is already written and enables transmission of a zero-length packet. When data of the maximum packet size is written for the pipe in non-continuous transfer mode, the USBHS sets this bit to 1 and switches the FIFO buffer from the CPU side to the SIE side, enabling transmission.

Only write 1 to the BVAL flag while the FRDY bit is 1 (set by the USBHS). When the selected pipe is receiving, do not set the BVAL flag to 1.

37.2.11 INTENB0 : Interrupt Enable Register 0

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x030

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
7:0	—	The read values are undefined. The write value should be 0.	R/W
8	BRDYE	Buffer Ready Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
9	NRDYE	Buffer Not Ready Response Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
10	BEMPE	Buffer Empty Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
11	CTRE	Control Transfer Stage Transition Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
12	DVSE	Device State Transition Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
13	SOFE	Frame Number Update Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
14	RSME	Resume Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
15	VBSE	VBUS Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W

Note: S-TYPE-3, P-TYPE-3
Note: The RSME, DVSE, and CTRE bits can only be set to 1 in device controller mode. Do not set these bits to 1 in host controller mode.

When a status flag in the INTSTS0 register sets to 1 and the associated interrupt request enable bit setting in the INTENB0 register is 1, the USBHS issues a USBHS interrupt request.

Regardless of the INTENB0 register setting, the status flag in the INTSTS0 register sets to 1 in response to a state change that satisfies the associated condition.

When an interrupt request enable bit in the INTENB0 register is switched from 0 to 1 while the associated status flag in the INTSTS0 register is set to 1, a USBHS interrupt is requested.

37.2.12 INTENB1 : Interrupt Enable Register 1

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x032

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OVRCRE	BCHGE	—	DTCHE	ATTCHE	—	L1RSMENDE	LPMENDE	—	EOFERRE	SIGNE	SACKE	—	—	—	PDDETINTE
Value after reset:	0	0	x	0	0	x	0	0	x	0	0	0	x	x	x	0

Bit	Symbol	Function	R/W
0	PDDETINTE	PDDETINT Detection Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
3:1	—	The read values are undefined. The write value should be 0.	R/W
4	SACKE	Setup Transaction Normal Response Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
5	SIGNE	Setup Transaction Error Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
6	EOFERRE	EOF Error Detection Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
7	—	The read values are undefined. The write value should be 0.	R/W
8	LPMENDE	LPM Transaction End Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
9	L1RSMENDE	L1 Resume End Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
10	—	The read values are undefined. The write value should be 0.	R/W
11	ATTCHE	Connection Detection Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
12	DTCHE	Disconnection Detection Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
13	—	The read values are undefined. The write value should be 0.	R/W
14	BCHGE	USB Bus Change Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
15	OVRCRE	OVRCRE Interrupt Request Enable 0: Disable interrupt request 1: Enable interrupt request	R/W

Note: S-TYPE-3, P-TYPE-3

When a status flag in the INTSTS1 register sets to 1 and the associated interrupt request enable bit setting in the INTENB1 register is 1, the USBHS issues a USBHS interrupt request.

Regardless of the INTENB1 register setting, the status flag in the INTSTS1 register sets to 1 in response to a state change that satisfies the associated condition.

When an interrupt request enable bit in the INTENB1 register is switched from 0 to 1 while the associated status flag in the INTSTS1 register is set to 1, a USBHS interrupt is requested.

37.2.13 BRDYENB : BRDY Interrupt Enable Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x036

Bit position:	15	14	13	12	11	10	9									0
Bit field:	—	—	—	—	—	—	PIPEBRDYE[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PIPEBRDYE[9:0]	BRDY Interrupt Request Enable for Pipes [9:0]*1 0: Disable interrupt request 1: Enable interrupt request	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Each bit number corresponds to the same pipe number.

The BRDYENB register enables or disables the INTSTS0.BRDY bit to be set to 1 when a BRDY interrupt is detected for each pipe.

When a status flag in the BRDYSTS register sets to 1 and the associated PIPEBRDYEn (n = 9 to 0) bit setting in the BRDYENB register is 1, the INTSTS0.BRDY flag sets to 1. In this case, if the BRDYE bit in INTENB0 is 1, the USBHS generates a BRDY interrupt request. While at least one PIPEBRDYEn flag indicates 1, the INTSTS0.BRDY flag sets to 1 when the associated interrupt request enable bit in the BRDYENB register is changed from 0 to 1 by software.

37.2.14 NRDYENB : NRDY Interrupt Enable Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x038

Bit position:	15	14	13	12	11	10	9									0
Bit field:	—	—	—	—	—	—	PIPENRDYE[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PIPENRDYE[9:0]	NRDY Interrupt Enable for Pipes [9:0]*1 0: Disable interrupt request 1: Enable interrupt request	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Each bit number corresponds to the same pipe number.

The NRDYENB register enables or disables the INTSTS0.NRDY bit to be set to 1 when a NRDY interrupt is detected for each pipe.

When a status flag in the NRDYSTS register sets to 1 and the associated PIPENRDYEn (n = 0 to 9) bit setting in the NRDYENB register is 1, the INTSTS0.NRDY flag sets to 1. In this case, if the NRDYE bit in INTENB0 is 1, the USBHS generates a NRDY interrupt request. While at least one PIPEBRDYEn flag indicates 1, the INTSTS0.NRDY flag sets to 1 when the associated interrupt request enable bit in the NRDYENB register is changed from 0 to 1 by software.

37.2.15 BEMPENB : BEMP Interrupt Enable Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x03A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPEBEMPE[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PIPEBEMPE[9:0]	BEMP Interrupt Enable for Pipes [9:0]*1 0: Disable interrupt request 1: Enable interrupt request	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Each bit number corresponds to the same pipe number.

The BEMPENB register enables or disables the INTSTS0.BEMP bit to be set to 1 when a BEMP interrupt is detected for each pipe.

When a status flag in the BEMPSTS register sets to 1 and the associated PIPEBEMPEn (n = 0 to 9) bit setting in the BEMPENB register is 1, the INTSTS0.BEMP flag sets to 1. In this case, if the BEMPE bit in INTENB0 is 1, the USBHS generates a BEMP interrupt request. While at least one PIPEBEMPEn flag indicates 1, the INTSTS0.BEMP flag sets to 1 when the associated interrupt request enable bit in the BEMPENB register is changed from 0 to 1 by software.

37.2.16 SOFCFG : SOF Output Configuration Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x03C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	TRNENSEL	—	BRDYM	INTL	EDGESTS	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	0	x	0	0	0	x	x	x	x

Bit	Symbol	Function	R/W
3:0	—	The read values are undefined. The write value should be 0.	R/W
4	EDGESTS	Interrupt Edge Processing Status Flag*1 Indicates 1 during the edge processing of an edge interrupt output signal.	R
5	INTL	Interrupt Output Sense Select*2 0: Edge detection 1: Level detection	R/W
6	BRDYM	PIPEBRDY Interrupt Status Clear Timing*3 0: Clear BRDY flag through software 1: Clear BRDY flag by the USBHS through a data read from the FIFO buffer or data write to the FIFO buffer	R/W
7	—	The read values are undefined. The write value should be 0.	R/W
8	TRNENSEL	Transaction-Enabled Time Select*4 0: Not low-speed communication 1: Low-speed communication	R/W
15:9	—	The read values are undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Confirm that the EDGESTS flag is 0 before stopping the clock supply to the USBHS.

Note 2. When the INTL bit is set to 0, to stop the PHY clock (LPSTS.SUSPENDM = 0) after clearing the interrupt status, write 0 to the LPSTS.SUSPENDM bit after confirming that the EDGESTS flag is cleared to 0.

Note 3. When setting the BRDYM bit to 1, set the INTL bit to 1.

Note 4. The setting in the TRNENSEL bit is only valid in host controller mode. Even in host controller mode, the setting of this bit has no effect on the transaction-enabled time during high-speed communication.

EDGESTS flag (Interrupt Edge Processing Status Flag)

The EDGESTS flag indicates 1 during the edge processing of an edge interrupt output signal. Confirm that this flag is 0 before stopping the PHY clock.

BRDYM bit (PIPEBRDY Interrupt Status Clear Timing)

The BRDYM bit specifies how the BRDY interrupt status flags for the pipes are cleared.

TRNENSEL bit (Transaction-Enabled Time Select)

When the USB port is in use for full-speed or low-speed communications, the TRNENSEL bit specifies the timing with which the USBHS issues tokens in a frame (transaction-enabled time).

Set this bit to 1 when a low-speed device is connected through a hub. The bit is only valid in host controller mode. Set this bit to 0 when the interface is in use as a device controller.

37.2.17 PHYSET : PHY Setting Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x03E

Bit position:	15	14	13	12	11	10	9		7	6	5		3	2	1	0
Bit field:	HSEB	—	—	—	REPSTART	—	REPSEL[1:0]		—	—	CLKSEL[1:0]		CDPEN	—	PLLRESET	DIRPD
Value after reset:	x	x	x	x	0	x	0	0	x	x	1	1	0	x	1	1

Bit	Symbol	Function	R/W
0	DIRPD	Power-Down Control 0: Do not enter low power mode 1: Enter low power mode	R/W
1	PLLRESET	PLL Reset Control*1 0: Disable PLL reset control for UTMI_PHY 1: Enable PLL reset control for UTMI_PHY	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	CDPEN	Charging Downstream Port Enable 0: Disable downstream port charging 1: Enable downstream port charging	R/W
5:4	CLKSEL[1:0]	Input Clock Frequency Select 0 0: 12 MHz 0 1: 48 MHz 1 0: 20 MHz 1 1: 24 MHz	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
9:8	REPSEL[1:0]	Terminating Resistance Adjustment Cycle 0 0: No cycle is set 0 1: Adjust terminating resistance at 16-second intervals 1 0: Adjust terminating resistance at 64-second intervals 1 1: Adjust terminating resistance at 128-second intervals	R/W
10	—	This bit is read as 0. The write value should be 0.	R/W
11	REPSTART	Forcibly Start Terminating Resistance Adjustment 0: Force terminating resistance adjustment to start 1: Do not force terminating resistance adjustment to start	R/W
14:12	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
15	HSEB	CL-only mode 0: Disable CL-only mode 1: Enable CL-only mode	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Because the value of the PLLRESET bit is 1 after a reset, changing the setting after release from the reset state is not required. Do not set the PLLRESET bit to 1 after setting the PLLRESET bit to 0. Operation is not guaranteed.

CLKSEL[1:0] bits (Input Clock Frequency Select)

The CLKSEL[1:0] bits select the transfer clock source for the USBHS.

For the transfer clock generated in the USB-PHY internal PLL, these bits set the input clock frequency. To input the clock source from the EXTAL pin, the USB 2.0 clock specification must be strictly followed.

Writing to the CKSEL[1:0] bits is invalid in CL-only mode because the internal PLL is stopped (see the description for HSEB bit (CL-only mode)). For the clock settings, see section 37.3.3. Supplying the Clock.

HSEB bit (CL-only mode)

The HSEB bit selects whether the USBHS operates in CL-only mode. High-speed transfer by the USBHS requires the use of internal high-speed analog circuits including the PLL, clock, and data recovery (CDR) circuit in the USB-PHY block.

CL-only mode limits the transfer to the USB 1.1 specification (full-speed and low-speed transfer only). Power consumption can be reduced by stopping the internal PLL of the PHY module and other high-speed analog circuits.

In CL-only mode, the USBHS requires supply clocks of 48 MHz and 60 MHz, generated in the Clock Generation Circuit. For the clock supply method, see section 9, Clock Generation Circuit.

37.2.18 INTSTS0 : Interrupt Status Register 0

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x040

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ[2:0]			VALID	CTSQ[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CTSQ[2:0]	Control Transfer Stage Flag[*1] 0 0 0: Idle or setup stage 0 0 1: Control read data stage 0 1 0: Control read status stage 0 1 1: Control write data stage 1 0 0: Control write status stage 1 0 1: Control write (no data) status stage 1 1 0: Control transfer sequence error	R
3	VALID	USB Request Reception Flag[*1] 0: Setup packet not received 1: Setup packet received	R/W[*3]
6:4	DVSQ[2:0]	Device State[*1] Indicates the device state. 0 0 0: Powered state 0 0 1: Default state 0 1 0: Address state 0 1 1: Configured state Others: Suspend state	R
7	VBSTS	VBUS Input Status Flag 0: USBHS_VBUS pin is low 1: USBHS_VBUS pin is high	R

Bit	Symbol	Function	R/W
8	BRDY	BRDY Interrupt Status Flag 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R
9	NRDY	NRDY Interrupt Status Flag 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R
10	BEMP	BEMP Interrupt Status Flag 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R
11	CTRT	Control Transfer Stage Transition Interrupt Status Flag*2 0: No control transfer stage transition interrupt occurred 1: Control transfer stage transition interrupt occurred	R/W*3
12	DVST	Device State Transition Interrupt Status Flag*2 0: No device state transition interrupt occurred 1: Device state transition interrupt occurred	R/W*3
13	SOFR	Frame Number Refresh Interrupt Status Flag 0: No SOF interrupt occurred 1: SOF interrupt occurred	R/W*3
14	RESM	Resume Interrupt Status Flag*2 *4 0: No resume interrupt occurred 1: Resume interrupt occurred	R/W*3
15	VBINT	VBUS Interrupt Status Flag*4 0: No VBUS interrupt occurred on detecting a change in the USBHS_VBUS pin 1: VBUS interrupt occurred on detecting a change in the USBHS_VBUS pin	R/W*3

Note: S-TYPE-3, P-TYPE-3

Note 1. The CTSQ[2:0], VALID, and DVSQ[2:0] flags are only valid in device controller mode.

Note 2. The status of the CTRT, DVST, and RESM flags are changed only in device controller mode. Set the associated interrupt enable bits to 0 (disabled) in host controller mode.

Note 3. To clear the CTRT, DVST, SOFR, RESM, or VBINT flags, write 0 only to the flags to be cleared. Write 1 to the other flags. Do not write 0 to the status flags indicating 0.

Note 4. The USBHS detects a change in the status in the RESM or VBINT flag even while the clock supply is stopped (LPSTS.SUSPENDM = 0), and it requests the interrupt when the associated interrupt request bit is 1. Enable the clock supply before clearing the status by software.

BRDY flag (BRDY Interrupt Status Flag)

The BRDY flag indicates the BRDY interrupt state. For the conditions that cause the flag to be set, see section 37.2.13. BRDYENB : BRDY Interrupt Enable Register.

The USBHS clears the BRDY flag to 0 when 0 is written to the BRDYSTS.PIPEBRDYn (n = 0 to 9) flags for all pipes for which the BRDY interrupt is enabled (BRDYENB.PIPEBRDYEn bits). Writing 0 to the BRDY flag in the software does not clear the flag.

NRDY flag (NRDY Interrupt Status Flag)

The NRDY flag indicates the NRDY interrupt state. For the conditions that cause the flag to be set, see section 37.2.14. NRDYENB : NRDY Interrupt Enable Register.

The USBHS clears the NRDY flag to 0 when 0 is written to the NRDYSTS.PIPENRDYn (n = 0 to 9) flags for all pipes for which the NRDY interrupt is enabled (NRDYENB.PIPENRDYEn bits). Writing 0 to the NRDY flag in the software does not clear the flag.

BEMP flag (BEMP Interrupt Status Flag)

The BEMP indicates the BEMP interrupt state. For the conditions that cause the flag to be set, see section 37.2.15. BEMPENB : BEMP Interrupt Enable Register.

The USBHS clears the BEMP flag to 0 when 0 is written to the BEMPSTS.PIPEBEMPn (n = 0 to 9) flags for all pipes for which the BEMP interrupt is enabled (BEMPENB.PIPEBEMPEn bits). Writing 0 to the BEMP flag in the software does not clear the flag.

CTRT flag (Control Transfer Stage Transition Interrupt Status Flag)

In device controller mode, the USBHS updates the value of the CTSQ[2:0] bits and sets the CTRT flag to 1 on detecting a transition in the control transfer stage. When a control transfer stage transition interrupt occurs, clear the CTRT flag before the USBHS detects the next control transfer stage transition.

Values read from the CTRT flag in host controller mode are invalid.

DVST flag (Device State Transition Interrupt Status Flag)

In device controller mode, the USBHS updates the value of the PL1CTRL1.DVSQ[3:0] bits and sets the DVST flag to 1 on detecting a change in the device state. When a device state transition interrupt occurs, clear the DVST flag before the USBHS detects the next device state transition.

Values read from the DVST flag in host controller mode are invalid.

SOFR flag (Frame Number Refresh Interrupt Status Flag)

In host controller mode, the USBHS sets the SOFR flag to 1 on updating the frame number when the DVSTCTR0.UACT bit is set to 1 by software. An SOFR interrupt is detected every 1 ms.

In device controller mode, the USBHS sets the SOFR flag to 1 on updating the frame number. An SOFR interrupt is detected every 1 ms. The USBHS can detect an SOFR interrupt through the SOF complementation function even when a corrupted SOF packet is received from the USB host. See section 37.3.13. SOF Complementation Function.

RESM flag (Resume Interrupt Status Flag)

In device controller mode, the USBHS sets the RESM flag to 1 on detecting the falling edge of the signal on the USBHS_DP pin in the Suspend state (PL1CTRL1.DVSQ[3:0] = 01xxb).

Values read from the RESM flag in host controller mode are invalid.

VBINT flag (VBUS Interrupt Status Flag)

The USBHS sets the VBINT flag to 1 on detecting a level change (high to low or low to high) in the USBHS_VBUS pin input value. The USBHS sets the VBSTS flag to indicate the USBHS_VBUS pin input value. When a VBINT interrupt occurs, eliminate transient elements by reading the VBSTS flag at least three times through software processing and check that the values read are the same.

37.2.19 INTSTS1 : Interrupt Status Register 1

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x042

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OVRCR	BCHG	—	DTCH	ATTCH	—	L1RSMEND	LPMEND	—	EOFERR	SIGN	SACK	—	—	—	PDDETINT
Value after reset:	0	0	x	0	0	x	0	0	x	0	0	0	x	x	x	0

Bit	Symbol	Function	R/W
0	PDDETINT	PDDET Detection Interrupt Status Flag*1 0: No PDDET interrupt occurred 1: PDDET interrupt occurred	R/W*2
3:1	—	The read values are undefined. The write value should be 0.	R/W
4	SACK	Setup Transaction Normal Response Interrupt Status Flag 0: No SACK interrupt occurred 1: SACK interrupt occurred	R/W*2
5	SIGN	Setup Transaction Error Interrupt Status Flag 0: No SIGN interrupt occurred 1: SIGN interrupt occurred	R/W*2
6	EOFERR	EOF Error Detection Interrupt Status Flag 0: No EOFERR interrupt occurred 1: EOFERR interrupt occurred	R/W*2

Bit	Symbol	Function	R/W
7	—	The read value is undefined. The write value should be 0.	R/W
8	LPMEND	LPM Transaction End Interrupt Status Flag 0: No LPMEND interrupt occurred 1: LPMEND interrupt occurred	R/W*2
9	L1RSMEND	L1 Resume End Interrupt Status Flag 0: No L1RSMEND interrupt occurred 1: L1RSMEND interrupt occurred	R/W*2
10	—	The read value is undefined. The write value should be 0.	R/W
11	ATTCH	USB Connection Detection Interrupt Status Flag 0: No ATTCH interrupt occurred 1: ATTCH interrupt occurred	R/W*2
12	DTCH	USB Disconnection Detection Interrupt Status Flag 0: No DTCH interrupt occurred 1: DTCH interrupt occurred	R/W*2
13	—	The read value is undefined. The write value should be 0.	R/W
14	BCHG	USB Bus Change Interrupt Status Flag*1 0: No BCHG interrupt occurred 1: BCHG interrupt occurred	R/W*2
15	OVRCR	OVRCR Interrupt Status Flag*1 0: No OVRCR interrupt occurred 1: OVRCR interrupt occurred	R/W*2

Note: S-TYPE-3, P-TYPE-3

Note: Only enable the status change interrupts indicated in the flags in INTSTS1 in host controller mode, except for the PDDET detection interrupt.

Note 1. The USBHS detects a change in the status in the PDDETINT, BCHG, or OVRCR flag even while the clock supply is stopped (LPSTS.SUSPENDM = 0), and it requests the interrupt when the associated interrupt request bit is 1. Enable the clock supply before clearing the status by software. No other interrupts can be detected while the clock supply is stopped (LPSTS.SUSPENDM = 0).

Note 2. To clear the flags in INTSTS1, write 0 only to the flags to be cleared. Write 1 to the other bits.

PDDETINT flag (PDDET Detection Interrupt Status Flag)

The USBHS sets the PDDETINT flag to 1 on detecting a level change (high to low or low to high) in the PDDET pin input value. When the PDDETINT interrupt is generated, perform debouncing by reading the PDDETSTS flag at least three times through software processing and checking that the values read are the same.

SACK flag (Setup Transaction Normal Response Interrupt Status Flag)

The SACK flag indicates the status of the setup transaction normal response interrupt in host controller mode.

The USBHS detects the SACK interrupt and sets this bit to 1 when an ACK response is returned from a peripheral device during the setup transactions issued by the USBHS. If the associated interrupt enable bit is set to 1 by software, the USBHS generates the interrupt.

Values read from the SACK flag in device controller mode are invalid.

SIGN flag (Setup Transaction Error Interrupt Status Flag)

The SIGN flag indicates the status of setup transaction error interrupts in host controller mode.

The USBHS detects the SIGN interrupt and sets this bit to 1 when an ACK response is not returned from a peripheral device three consecutive times during the setup transactions issued by the USBHS. If the associated interrupt enable bit is set to 1 by software, the USBHS generates the interrupt.

The USBHS detects the SIGN interrupt when any of the following response conditions occur for three consecutive setup transactions:

- Timeout is detected by the USBHS when the peripheral device has returned no response
- A corrupted ACK packet is received
- A handshake other than ACK (NAK, NYET, or STALL) is received

Values read from the SIGN flag in device controller mode are invalid.

EOFERR flag (EOF Error Detection Interrupt Status Flag)

The EOFERR flag indicates the status of EOF error detection interrupts in host controller mode.

The USBHS detects the EOFERR interrupt and sets this bit to 1 on detecting that communication did not complete at the EOF2 timing defined in the USB 2.0 specification. If the associated interrupt enable bit is set to 1 by software, the USBHS generates the interrupt.

After detecting the EOFERR interrupt, the USBHS controls the hardware as follows, regardless of the associated interrupt enable bit setting:

- Sets the DVSTCTR0.UACT bit for the port in which the EOFERR interrupt was detected to 0
- Puts the port in which the EOFERR interrupt occurred into the idle state

The software must terminate all pipes in which communications are being carried out and re-enumerate the USB port.

Values read from the EOFERR flag in device controller mode are invalid.

LPMEND flag (LPM Transaction End Interrupt Status Flag)

The PLPMEND flag indicates the status of LPM transaction end interrupts in host controller mode.

When the HL1CTRL1.L1REQ bit sets to 1, the USBHS sends an LPM token. When the LPM transaction is ended because a response from the function device or a timeout is detected, the USBHS sets this flag to 1.

Values read from the LPMEND flag in device controller mode are invalid.

L1RSMEND flag (L1 Resume End Interrupt Status Flag)

The L1RSMEND flag indicates the status of L1 resume end interrupts in host controller mode.

When performing resume processing after transitioning to the L1 state because an ACK was received in response to an LPM token, the USBHS sets this flag to 1.

Values read from the L1RSMEND flag in device controller mode are invalid.

ATTCH flag (USB Connection Detection Interrupt Status Flag)

The ATTCH flag indicates the status of USB attach detection interrupts in host controller mode.

The USBHS detects the ATTCH interrupt and sets this bit to 1 on detecting a J-state or K-state on the full-speed or low-speed signal level for 2.5 µs. If the associated interrupt enable bit is set to 1 by software, the USBHS generates the interrupt.

The USBHS detects the ATTCH interrupt on any of the following conditions:

- K-state, SE0, or SE1 changes to J-state, and J-state continues for 2.5 µs
- J-state, SE0, or SE1 changes to K-state, and K-state continues for 2.5 µs

Values read from the ATTCH flag in device controller mode are invalid.

DTCH flag (USB Disconnection Detection Interrupt Status Flag)

The DTCH flag indicates the status of USB detach detection interrupts in host controller mode.

The USBHS detects the DTCH interrupt and sets this bit to 1 on detecting a USB bus detach event. If the associated interrupt enable bit is set to 1 by software, the USBHS generates the interrupt.

The USBHS detects bus detach events based on the USB 2.0 specification.

After detecting the DTCH interrupt, the USBHS controls hardware as follows, regardless of the associated interrupt enable bit setting:

- Sets the DVSTCTR0.UACT bit for the port in which the DTCH interrupt was detected to 0
- Puts the port in which the DTCH interrupt occurred into the idle state

The software must terminate all pipes in which communications are being carried out and invoke the wait state for attaching to the USB port (waiting for ATTCH interrupt generation).

Values read from the DTCH flag in device controller mode are invalid.

BCHG flag (USB Bus Change Interrupt Status Flag)

The BCHG flag indicates the status of USB bus change interrupts in host controller mode.

The USBHS detects the BCHG interrupt and sets this bit to 1 when a change in the full-speed signal level occurs on the USB port. This includes any change from J-state, K-state, or SE0 to J-state, K-state, or SE0. If the associated interrupt enable bit is set to 1 by software, the USBHS generates the interrupt.

The USBHS sets the SYSSTS0.LNST[1:0] flags to indicate the input state of the USB port. When a BCHG interrupt occurs, eliminate transient elements by repeat reading the LNST[1:0] bits by software until the same value is read at least three times.

Changes in the USB bus state can be detected while the PHY clock is stopped.

Values read from the BCHG flag in device controller mode are invalid.

OVRCR flag (OVRCR Interrupt Status Flag)

The OVRCR flag indicates the input status on the USBHS_OVCUR0A pin or changes on the USBHS_OVCUR0B pin. If the INTENB1.OVRCRE bit sets to 1, the USBHS requests the interrupt.

The USBHS sets the SYSSTS0.OVCMON[1:0] flags to indicate the input state of the USBHS_OVCUR0A and USBHS_OVCUR0B pins.

These pins allow overcurrent detection by software in host controller mode. To implement this function, connect the overcurrent signal from the external power supply IC that supplies VBUS to connected USB devices to the OVCUR0A or OVCUR0B pin. On detection of an OVRCR interrupt, eliminate transients by repeatedly reading the OVCMON[1:0] flags through the software until the same value is read at least three times.

37.2.20 BRDYSTS : BRDY Interrupt Status Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x046

Bit position:	15	14	13	12	11	10	9									0
Bit field:	—	—	—	—	—	—	PIPEBRDY[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PIPEBRDY[9:0]	BRDY Interrupt Status Flag for Pipe[9:0]*1 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*2
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Each bit number corresponds to the same pipe number.
Note 2. When the SOFCFG.BRDYM bit is set to 0, to clear the status indicated in the PIPEBRDY[9:0] flags, write 0 only to the bits to be cleared. Write 1 to the other bits.
When the SOFCFG.BRDYM bit is set to 0, clear BRDY interrupts before accessing the FIFO.

PIPEBRDY[9:0] flags (BRDY Interrupt Status Flag for Pipe[9:0])

When the BRDY interrupt is detected, the USBHS sets the associated bit in the PIPEBRDY[9:0] flags to 1. For details on BRDY interrupts, see section 37.3.6.1. BRDY Interrupt.

37.2.21 NRDYSTS : NRDY Interrupt Status Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x048

Bit position:	15	14	13	12	11	10	9									0
Bit field:	—	—	—	—	—	—	PIPENRDY[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PIPENRDY[9:0]	NRDY Interrupt Status Flag for Pipe[9:0]*1 0: No NRDY interrupt occurred 1: NRDY interrupt occurred.	R/W*2
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Each bit number corresponds to the same pipe number.
Note 2. To clear the status indicated in the PIPENRDY[9:0] flags, write 0 only to the bits to be cleared. Write 1 to the other bits.

PIPENRDY[9:0] flags (NRDY Interrupt Status Flag for Pipe[9:0])

If an internal NRDY interrupt is detected while the PID[1:0] bits in a pipe control register are 01b (BUF response), the USBHS sets the associated bit in the PIPENRDY[9:0] flags to 1. For details on NRDY interrupts, see section 37.3.6.2. NRDY Interrupt.

37.2.22 BEMPSTS : BEMP Interrupt Status Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x04A

Bit position:	15	14	13	12	11	10	9									0
Bit field:	—	—	—	—	—	—	PIPEBEMP[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PIPEBEMP[9:0]	BEMP Interrupt Status Flag for Pipe[9:0]*1 0: No BEMP interrupt occurred 1: BEMP interrupt occurred.	R/W*2
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Each bit number corresponds to the same pipe number.
Note 2. To clear the status indicated in the PIPEBEMP[9:0] flags, write 0 only to the bits to be cleared. Write 1 to the other bits.

PIPEBEMP[9:0] flags (BEMP Interrupt Status Flag for Pipe[9:0])

If an BEMP interrupt is detected while the PID[1:0] bits in a pipe control register are 01b (BUF response), the USBHS sets the associated bit in the PIPEBEMP[9:0] flags to 1. For details on BEMP interrupts, see section 37.3.6.3. BEMP Interrupt.

37.2.23 FRMNUM : Frame Number Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x04C

Bit position:	15	14	13	12	11	10										0
Bit field:	OVRN	CRCE	—	—	—	FRNM[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	FRNM[10:0]*1	Frame Number Flag	R
13:11	—	These bits are read as 0. The write value should be 0.	R/W
14	CRCE	CRC Error Detection Status Flag 0: No error occurred 1: Error occurred	R/W
15	OVRN	Overrun/Underrun Detection Status Flag 0: No error occurred 1: Error occurred.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The OVRN flag is for debugging. Design the timing so that no overrun or underrun occurs in the system.

FRNM[10:0] flags (Frame Number Flag)

The USBHS sets the FRNM[10:0] flags to indicate the latest frame number, which is updated every 1 ms, when an SOF packet is issued or received.

CRCE flag (CRC Error Detection Status Flag)

The CRCE flag sets to 1 when a CRC error or bit stuffing error occurs during isochronous transfer. On detecting a CRC error, the USBHS generates an internal NRDY interrupt.

To clear the CRCE flag, write 0 to it while writing 1 to the other bits in the FRMNUM register.

OVRN flag (Overrun/Underrun Detection Status Flag)

The OVRN flag sets to 1 when an overrun or underrun error occurs during isochronous transfer. To clear the flag, write 0 to it while writing 1 to the other bits in the FRMNUM register.

In host controller mode, the OVRN flag sets to 1 on any of the following conditions:

- For a transmitting isochronous pipe, the time to issue an OUT token comes before all of the transmit data is written to the FIFO buffer
- For a receiving isochronous pipe, the time to issue an IN token comes when no FIFO buffer planes are empty

In device controller mode, the OVRN flag sets to 1 on any of the following conditions:

- For a transmitting isochronous pipe, the IN token is received before all of the transmit data is written to the FIFO buffer
- For a receiving isochronous pipe, the OUT token is received when no FIFO buffer planes are empty

37.2.24 UFRMNUM : μFrame Number Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x04E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2		0
Bit field:	DVCHG	—	—	—	—	—	—	—	—	—	—	—	—	UFRNM[2:0]		
Value after reset:	0	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
2:0	UFRNM[2:0]	Microframe number	R
14:3	—	These bits are read as 0. The write value should be 0.	R/W
15	DVCHG	Device State Change 0: Disable writes to the USBADDR.STSRECOV0[2:0] and USBADDR.USBADDR [6:0] bits 1: Enable writes to the USBADDR.STSRECOV0[2:0] and USBADDR.USBADDR[6:0] bits	R/W

Note: S-TYPE-3, P-TYPE-3

UFRNM[2:0] flags (Microframe number)

The USBHS sets the UFRNM[2:0] flags to indicate the microframe number during high-speed operation. When not in high-speed operation, the USBHS sets these bits to 00b.

Read these bits repeatedly until the same value is read twice.

37.2.25 USBADDR : USB Address Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x050

Bit position:	15	14	13	12	11	10			7	6						0
Bit field:	—	—	—	—	—	STSRECOV0[2:0]			—	USBADDR[6:0]						
Value after reset:	x	x	x	x	x	0	0	0	x	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	USBADDR[6:0]	USB Address Flag In device controller mode, these flags indicate the USB address assigned by the host when the USBHS processed the SET_ADDRESS request successfully.	R
7	—	The read value is undefined. The write value should be 0.	R/W
10:8	STSRECOV0[2:0]	Status Recovery [D]: In deice controller mode [H]: In host controller mode (settings other than 010b, 100b, or 110b are prohibited) 0 0 0: Reserved 0 0 1: [D] Return to the full-speed connection and Default state 0 1 0: [D] Return to the full-speed connection and Address state [H] Return to the low-speed state (bits DVSTCTR0.RHST[2:0] = 001b) 0 1 1: [D] Return to the full-speed connection and Configured state 1 0 0: [D] Return to the suspend connection and Suspend state [H] Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b) 1 0 1: [D] Return to the high-speed connection and Default state 1 1 0: [D] Return to the high-speed connection and Address state [H] Return to the high-speed state (bits DVSTCTR0.RHST[2:0] = 011b) 1 1 1: [D] Return to the high-speed connection and Configured state	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

USBADDR[6:0] flags (USB Address Flag)

In device controller mode, the USBADDR[6:0] flags indicate the USB address received when the USBHS processed a SetAddress request successfully. The USBHS sets the USBADDR[6:0] bits to 00b on detecting a USB bus reset.

In host controller mode, the USBADDR[6:0] bits are invalid.

STSRECOV0[2:0] bits (Status Recovery)

Use the STSRECOV[3:0] bits to resume the state of the internal sequencer on recovering from USB power shut-off. For details, see section 37.3.17. Deep Software Standby Mode 1 Because of USB Suspend/Resume Interrupts.

Writing to these bits is enabled while the UFRMNUM.DVCHG bit is set to 1.

37.2.26 USBREQ : USB Request Type Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x054

Bit position:	15								7							0
Bit field:	BREQUEST[7:0]								BMREQUESTTYPE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	BMREQUESTTYPE [7:0]	USB request bmRequestType value	R/W*1
15:8	BREQUEST[7:0]	USB request bRequest value	R/W*1

Note: S-TYPE-3, P-TYPE-3
Note 1. In device controller mode, these bits can be read, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

BMREQUESTTYPE[7:0] bits (USB request bmRequestType value)

The BMREQUESTTYPE[7:0] bits hold the bmRequestType value of USB requests.

- In host controller mode:
  Set these bits to the value of the USB request data in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:
  These bits indicate the value of the USB request data in reception setup transactions. Writing to the bits has no effect.

BREQUEST[7:0] bits (USB request bRequest value)

The BREQUEST[7:0] bits hold the bRequest value of USB requests.

- In host controller mode:
  Set these bits to the value of the USB request data in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:
  These bits indicate the value of the USB request data in reception setup transactions. Writing to the bits has no effect.

37.2.27 USBVAL : USB Request Value Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x056

Bit position:	15															0
Bit field:	WVALUE[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	WVALUE[15:0]	USB request wValue value	R/W*1

Note: S-TYPE-3, P-TYPE-3
Note 1. In device controller mode, these bits are readable, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

WVALUE[15:0] bits (USB request wValue value)

The WVALUE[15:0] bits hold the wValue value of USB requests.

- In host controller mode:

Set these bits to the wValue value for USB requests in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.

- In device controller mode:
  These bits indicate the wValue value of USB requests in reception setup transactions. Writing to the bits has no effect.

37.2.28 USBINDX : USB Request Index Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x058

Bit position:	15															0
Bit field:	WINDEX[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	WINDEX[15:0]	USB request wIndex value	R/W*1

Note: S-TYPE-3, P-TYPE-3
Note 1. In device controller mode, these bits are readable, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

WINDEX[15:0] bits (USB request wIndex value)

- In host controller mode:
  Set these bits to the wIndex value of USB requests in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:
  These bits indicate the wIndex value of USB requests received in reception setup transactions. Writing to the bits has no effect.

37.2.29 USBLENG : USB Request Length Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x05A

Bit position:	15															0
Bit field:	WLENTUH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	WLENTUH[15:0]	USB request wLength value	R/W*1

Note: S-TYPE-3, P-TYPE-3
Note 1. In device controller mode, these bits are readable, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

WLENTUH[15:0] bits (USB request wLength value)

The WLENTUH[15:0] bits hold the wLength value of USB requests.

- In host controller mode:
  Set the wLength value of USB requests in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:
  These bits indicate the wLength value of USB requests in reception setup transactions. Writing to the bits has no effect.

37.2.30 DCPCFG : DCP Configuration Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x05C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CNTMD	SHTNAK	—	—	DIR	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	DIR	Transfer Direction 0: Data receiving direction 1: Data transmitting direction	R/W
6:5	—	The read values are undefined. The write value should be 0.	R/W
7	SHTNAK	Pipe Blocking on End of Transfer 0: Keep pipe open after transfer ends 1: Disable pipe after transfer ends	R/W
8	CNTMD	Continuous Transfer Mode 0: Non-continuous transfer mode 1: Continuous transfer mode	R/W
15:9	—	The read values are undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note: Only set the bits in the DCPCFG register while the PID is NAK. Before setting the bits, check that the DCPCTR.PBUSY bit is 0, and then change the DCPCTR.PID[1:0] bits for the DCP from BUF to NAK. If the PID[1:0] bits are changed to NAK by the USBHS, checking the PBUSY bit through software is not necessary.

DIR bit (Transfer Direction)

In host controller mode, the DIR bit sets the transfer direction of the data stage and status stage for control transfers. In device controller mode, set the DIR bit to 0.

SHTNAK bit (Pipe Blocking on End of Transfer)

The SHTNAK bit specifies whether to change PID to NAK on transfer end when the selected pipe is receiving. It is only valid when the selected pipe is receiving.

When the SHTNAK bit is 1, the USBHS changes the DCPCTR.PID[1:0] bits for the DCP to NAK on determining that a transfer has ended. The USBHS determines transfer end on the following condition:

- A short packet, including a zero-length packet, is successfully received

CNTMD bit (Continuous Transfer Mode)

The CNTMD bit indicates whether transfer through the default control pipe is in continuous transfer mode.

37.2.31 DCPMAXP : DCP Maximum Packet Size Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x05E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DEVSEL[3:0]				—	—	—	—	—	MXPS[6:0]						
Value after reset:	0	0	0	0	x	x	x	x	x	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	MXPS[6:0]	Maximum Packet Size*1 Maximum data payload specification (maximum packet size) for the DCP	R/W
11:7	—	The read values are undefined. The write value should be 0.	R/W
15:12	DEVSEL[3:0]	Device Select*2 0x0: Address 0x0 0x1: Address 0x1 0x2: Address 0x2 0x3: Address 0x3 0x4: Address 0x4 0x5: Address 0x5	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Only set the MXPS[6:0] bits while PID is NAK. Before setting this bit, check that the CSSTS and PBUSY bits are 0, and then change the DCPCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK), and the CFIFOSEL.CURPIPE[3:0] bits to 0000b. If the DCPCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the CSSTS and PBUSY bits through software is not necessary. After the MXPS[6:0] bits are set and the DCP is set to the CURPIPE[3:0] bits in a port select register, clear the buffer by setting the BCLR bit the port control register to 1.

Note 2. Only set the DEVSEL[3:0] bits while PID is NAK and the DCPCTR.SUREQ bits are 0. Before setting these bits, check that the CSSTS and PBUSY flags are 0, and then change the DCPCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK), and the DCPCTR.SUREQ[3:0] bits to 0. If the DCPCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the CSSTS and PBUSY bits through software is not necessary.

MXPS[6:0] bits (Maximum Packet Size)

The MXPS[6:0] bits specify the maximum data payload (maximum packet size) for the DCP. The initial value is 0x40 (64 bytes). Set the bits to a USB 2.0-compliant value. Do not write to the FIFO buffer or set PID = BUF while MXPS[6:0] is set to 0.

DEVSEL[3:0] bits (Device Select)

In host controller mode, the DEVSEL[3:0] bits specify the address of the target peripheral device for a control transfer. Set up the device address in the associated DEVADDn (n = 0 to A) register first, and then set these bits to the corresponding value. To set the DEVSEL[3:0] bits to 0010b, for example, first set the address in the DEVADD2 register. In device controller mode, set these bits to 0000b.

37.2.32 DCPCTR : DCP Control Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x060

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSTS	SUREQ	CSCLR	CSSTS	SUREQCLR	—	—	SQCLR	SQSET	SQMON	PBUSY	PINGE	—	CCPL	PID[1:0]	
Value after reset:	0	0	0	0	x	x	x	0	0	1	0	0	x	0	0	0

Bit	Symbol	Function	R/W
1:0	PID[1:0]	Response PID 0 0: NAK response 0 1: BUF response (depends on buffer state) 1 x: STALL response	R/W
2	CCPL	Control Transfer End Enable 0: Disable control transfer completion 1: Enable control transfer completion	R/W
3	—	The read value is undefined. The write value should be 0.	R/W
4	PINGE	PING Token Issue Enable*1 0: Disable PING token 1: Enable normal PING operation	R/W

Bit	Symbol	Function	R/W
5	PBUSY	Pipe Busy Flag 0: DCP not used for the USB bus 1: DCP in use for the USB bus	R
6	SQMON	Sequence Toggle Bit Monitor Flag 0: DATA0 1: DATA1	R
7	SQSET	Sequence Toggle Bit Set*1 0: Invalid (writing 0 has no effect) 1: Set the expected value for the next transaction to DATA1	W
8	SQCLR	Sequence Toggle Bit Clear*1 0: Invalid (writing 0 has no effect) 1: Clear the expected value for the next transaction to DATA0	W
10:9	—	The read values are undefined. The write value should be 0.	R/W
11	SUREQCLR	SUREQ Bit Clear 0: Invalid (writing 0 has no effect) 1: Clear SUREQ to 0	W
12	CSSTS	CSSTS Status Flag 0: Start-split (SSPLIT) transaction, or processing for devices that are not using split transactions, in progress 1: Complete-split (CSPLIT) transaction in progress	R
13	CSCLR	CSSTS Status Flag Clear 0: (writing 0 has no effect) 1: Clear CSSTS to 0	W
14	SUREQ	SETUP Token Transmission 0: Invalid (writing 0 has no effect) 1: Transmit setup packet	R/W
15	BSTS	Buffer Status Flag 0: Buffer access disabled 1: Buffer access enabled	R

Note: S-TYPE-3, P-TYPE-3

Note 1. Only set the SQSET, SQCLR, and PINGE bits while PID is NAK. Before setting these bits, check that the CSSTS and PBUSY bits are 0, and then change the DCPCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the DCPCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the CSSTS and PBUSY bits through the software is not necessary.

PID[1:0] bits (Response PID)

The PID[1:0] bits control the USB response type during control transfers.

In host controller mode, to change the PID[1:0] setting from NAK to BUF:

- When the transmitting direction is set:
  a. Write all of the transmit data to the FIFO buffer while the DVSTCTR0.UACT bit is 1 and PID is NAK.
  b. Set PID[1:0] bits to 01b (BUF).
  The USBHS then executes the OUT transaction (or PING transaction)
- When the receiving direction is set:
  a. Check that the FIFO buffer is empty (or empty the buffer) while the DVSTCTR0.UACT bit is 1 and PID is NAK.
  b. Set PID[1:0] bits to 01b (BUF).
  The USBHS then executes the IN transaction.

The USBHS changes the PID[1:0] setting as follows:

- When the PID[1:0] bits are set to BUF (01b) by software and the USBHS has received data exceeding MaxPacketSize, the USBHS sets PID[1:0] to STALL (11b)
- When a reception error, such as a CRC error, is detected three times consecutively, the USBHS sets PID[1:0] to NAK (00b)
- On receiving the STALL handshake, the USBHS sets PID[1:0] to STALL (11b)

In device controller mode, the USBHS changes the PID[1:0] setting as follows:

- On receiving a setup packet, the USBHS sets PID[1:0] to NAK (00b). The USBHS then sets the INTSTS0.VALID flag to 1, and the PID[1:0] setting cannot be changed until the software clears the VALID flag to 0.
- When the PID[1:0] bits are set to BUF (01b) by software and the USBHS has received data exceeding MaxPacketSize, the USBHS sets PID[1:0] to STALL (11b)
- On detecting a control transfer sequence error, the USBHS sets PID[1:0] to STALL (1xb)
- On detecting a USB bus reset, the USBHS sets PID[1:0] to NAK

The USBHS does not check the PID[1:0] setting while processing a SET_ADDRESS request.

CCPL bit (Control Transfer End Enable)

In device controller mode, setting the CCPL bit to 1 enables the status stage of the control transfer to be completed. When the bit is set to 1 by software while the associated PID[1:0] bits are set to BUF, the USBHS completes the control transfer status stage.

During control read transfers, the USBHS transmits the ACK handshake in response to the OUT transaction from the USB host. During control write or no-data control transfers, it transmits the zero-length packet in response to the IN transaction from the USB host. On detecting a SET_ADDRESS request, the USBHS operates in auto response mode from the setup stage up to status stage completion regardless of the CCPL bit setting.

The USBHS changes the CCPL bit from 1 to 0 on receiving a new setup packet. The software cannot write 1 to the bit while the INTSTS0.VALID bit is 1. The bit is initialized by a USB bus reset.

In host controller mode, always write 0 to the CCPL bit.

PINGE bit (PING Token Issue Enable)

In host controller mode, when the software sets the PINGE bit to 1, the USBHS issues a PING token for transfer in the transmitting direction, which triggers the transfer to start. If an ACK handshake is detected in the PING transaction, the OUT transaction is executed in the next transaction. If a NAK or NYET handshake is detected in the OUT transaction, the PING transaction is executed in the next transaction.

If the software sets this bit to 0, the USBHS issues no PING token for transfer in the transmitting direction. All transfers in the transmitting direction are executed in the OUT transaction.

PBUSY flag (Pipe Busy Flag)

The PBUSY bit indicates whether DCP is used for the transaction when USBHS changes the PID[1:0] bits from BUF to NAK. The USBHS changes the PBUSY flag from 0 to 1 on start of a USB transaction for the selected pipe. It changes the PBUSY flag from 1 to 0 on completion of one transaction.

After PID is set to NAK by software, the value in the PBUSY flag indicates whether changes to pipe settings can proceed.

For details, see section 37.3.7.1. Pipe Control Register Switching Procedures.

SQMON flag (Sequence Toggle Bit Monitor Flag)

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction during a DCP transfer.

The USBHS toggles the bit on normal completion of the transaction. It does not toggle the bit, however, when a DATAPID mismatch occurs during a transfer in the receiving direction.

In device controller mode, the USBHS sets the SQMON bit to 1 (specifies DATA1 as the expected value) on successful reception of the setup packet.

In device controller mode, the USBHS does not reference this bit during IN or OUT transactions at the status stage, and it does not toggle the bit on normal completion.

SQSET bit (Sequence Toggle Bit Set)

The SQSET bit specifies DATA1 as the expected value of the sequence toggle bit for the next transaction during a DCP transfer.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

SQCLR bit (Sequence Toggle Bit Clear)

The SQCLR bit specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during a DCP transfer. It is read as 0.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

SUREQCLR bit (SUREQ Bit Clear)

In host controller mode, setting the SUREQCLR bit to 1 clears the SUREQ bit to 0. The bit is read as 0.

If transfer stops while the SUREQ bit is set to 1 in a setup transaction, set the SUREQCLR bit to 1 through software. This is not necessary at the end of a normal setup transaction, because the USBHS automatically clears the SUREQ bit to 0.

Only control the SUREQ bit through the SUREQCLR bit while the DVSTCTR0.UACT bit is 0. When UACT is 0, communication is halted or no transfer is occurring because a bus disconnection was detected.

In device controller mode, always write 0 to the SUREQCLR bit.

CSSTS flag (CSSTS Status Flag)

In host controller mode, the CSSTS flag indicates the complete-split state in split transactions for pipes that are not isochronous. The USBHS sets the CSSTS flag to 1 at the beginning of a complete-split transaction and sets the flag back to 0 when it detects transaction completion.

Values read from the CSSTS flag in device controller mode are invalid.

CSCLR bit (CSSTS Status Flag Clear)

In host controller mode, setting the CSCLR bit to 1 clears the CSSTS bit to 0.

Set this bit to 1 through software when forcing the next transfer to restart from start-split in transfers using split transactions. This is not necessary at the end of a successful complete-split transaction in a normal split transaction, because the USBHS automatically clears the CSSTS flag to 0.

Only control the CSSTS flag through the CSCLR bit while the DVSTCTR0.UACT bit is 0. When UACT is 0, communication is halted or no transfer is occurring because a port disconnection was detected. Writing 1 to this bit while the CSSTS flag is 0 has no effect; the flag remains 0.

In device controller mode, always write 0 to this bit.

SUREQ bit (SETUP Token Transmission)

In host controller mode, setting the SUREQ bit to 1 triggers the USBHS to transmit the setup packet. After completing the setup transaction process, the USBHS generates either the SACK or SIGN interrupt and clears the SUREQ bit to 0. The USBHS also clears the SUREQ bit to 0 when the software sets the SUREQCLR bit to 1.

Before setting the SUREQ bit to 1, set the DCPMAXP.DEVSEL[3:0] bits, USBREQ, USBVAL, USBINDX, and USBLENG appropriately to transmit the wanted USB request in the setup transaction. Also check that the PID[1:0] bits for the DCP are set to NAK. After setting the SUREQ bit to 1, do not change the DCPMAXP.DEVSEL[3:0] bits, USBREQ, USBVAL, USBINDX, or USBLENG until the setup transaction is complete (SUREQ bit = 1). Write 1 to the SUREQ bit only when transmitting the setup token. Otherwise, write 0.

In device controller mode, always write 0 to this bit.

BSTS flag (Buffer Status Flag)

The BSTS flag indicates the status of access to the DCP FIFO buffer. The meaning of this flag varies as follows depending on the CFIFOSEL.ISEL setting:

- When ISEL = 0, the bit indicates whether receive data can be read from the buffer
- When ISEL = 1, the bit indicates whether transmit data can be written to the buffer

37.2.33 PIPESEL : Pipe Window Select Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x064

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	PIPESEL[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	PIPESEL[3:0]	Pipe Window Select 0x0: No pipe selected 0x1: Pipe 1 0x2: Pipe 2 0x3: Pipe 3 0x4: Pipe 4 0x5: Pipe 5 0x6 Pipe 6 0x7: Pipe 7 0x8: Pipe 8 0x9: Pipe 9 Others: Setting prohibited	R/W
15:4	—	The read values are undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Set pipes 1 to 9 using the PIPESEL, PIPECFG, PIPEMAXP, PIPEPERI, PIPEnCTR, PIPEnTRE, and PIPEnTRN registers (n = 0 to 9).

After selecting the pipe in the PIPESEL register, pipe functions must be set in the associated PIPECFG, PIPEMAXP, and PIPEPERI registers. PIPEnCTR, PIPEnTRE, and PIPEnTRN can be set independently of the pipe selection in this register.

PIPESEL[3:0] bits (Pipe Window Select)

The PIPESEL[3:0] bits select the pipe number associated with the PIPECFG, PIPEMAXP, and PIPEPERI registers used for data writing and reading. Selecting a pipe number in the PIPESEL[3:0] bits allows writing to and reading from PIPECFG, PIPEMAXP, and PIPEPERI associated with the selected pipe number.

When PIPESEL[3:0] = 0x0, 0 is read from all of the bits in PIPECFG, PIPEMAXP, and PIPEPERI. Writing to these bits has no effect.

37.2.34 PIPECFG : Pipe Configuration Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x068

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3			0
Bit field:	TYPE[1:0]		—	—	—	BFRE	DBLB	CNTMD	SHTNAK	—	—	DIR	EPNUM[3:0]			
Value after reset:	0	0	x	x	x	0	0	0	0	x	x	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	EPNUM[3:0]	Endpoint Number*1 Specifies the endpoint number for the selected pipe. Setting 0x0 indicates the pipe is not used.	R/W
4	DIR	Transfer Direction*2 *3 0: Receiving direction 1: Transmitting direction	R/W

Bit	Symbol	Function	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	SHTNAK	Pipe Disabled at End of Transfer*1 0: Continue pipe operation after transfer ends 1: Disable pipe after transfer ends	R/W
8	CNTMD	Continuous Transfer Mode*2 *3 0: Discontinuous transfer mode 1: Continuous transfer mode	R/W
9	DBLB	Double Buffer Mode*2 *3 0: Single buffer 1: Double buffer	R/W
10	BFRE	BRDY Interrupt Operation Specification*2 *3 0: Generate BRDY interrupt on transmitting or receiving data 1: Generate BRDY interrupt on completion of reading data	R/W
13:11	—	These bits are read as 0. The write value should be 0.	R/W
15:14	TYPE[1:0]	Transfer Type*1 0 0: Pipe not used 0 1: (Pipe 1 to 5) Bulk transfer (Pipe 6 to 9) Setting prohibited 1 0: (Pipe 1 to 5) Setting prohibited (Pipe 6 to 9) Interrupt transfer 1 1: (Pipe 1 to 2) Isochronous transfer (Pipe 3 to 9) Setting prohibited	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Only set the TYPE[1:0], SHTNAK, and EPNUM[3:0] bits while PID is NAK. Before setting these bits, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the CSSTS and PBUSY flags through the software is not necessary.

Note 2. Only set the BFRE, DBLB, CNTMD, and DIR bits while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting these bits, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the PBUSY flag through the software is not necessary.

Note 3. To change the BFRE, DBLB, CNTMD, or DIR bit after completing USB communication on the selected pipe, in addition to the constraints described in note 2, write 1 and 0 to the PIPEnCTR.ACLRM bit continuously through software and clear the FIFO buffer assigned to the pipe.

EPNUM[3:0] bits (Endpoint Number)

The EPNUM[3:0] bits specify the endpoint number for the selected pipe. Setting 0000b indicates the pipe not used.

Set these bits so that the combination of the DIR and EPNUM[3:0] settings is different from those for other pipes. (The EPNUM[3:0] bits can be set to 0000b for all pipes.)

DIR bit (Transfer Direction)

The DIR bit specifies the transfer direction for the selected pipe.

When the software sets this bit to 0, the USBHS uses the selected pipe for receiving. When the software sets this bit to 1, the USBHS uses the selected pipe for transmitting.

SHTNAK bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to change the PIPEnCTR.PID[1:0] bits to 00b (NAK) at the end of transfer when the selected pipe is set in the receiving direction. The bit is valid for pipes 1 to 5 in the receiving direction.

When the software sets this bit to 1 for a receiving pipe, the USBHS changes the associated PIPEnCTR.PID[1:0] bits to 00b (NAK) on determining the transfer end. The USBHS determines that the transfer has ended on the following conditions:

- Short packet data (including a zero-length packet) was successfully received
- The transaction counter is used and the number of packets specified for the transaction counter were successfully received

CNTMD bit (Continuous Transfer Mode)

The CNTMD bit specifies whether to operate the selected pipe in continuous transfer mode. The bit is valid for pipes 1 to 5 of the bulk transfer type.

Based on this bit setting, the USBHS determines the completion of transmission or reception for the FIFO buffer allocated to the selected pipe as shown in Table 37.9.

Table 37.9 Relationship between the CNTMD setting and methods for determining completion of FIFO buffer transmission or reception

CNTMD bit setting	Methods for determining readable state and transmittable state
0	Condition for FIFO buffer readable state in receiving direction (DIR = 0): ● The USBHS received one packet
	Conditions for FIFO buffer transmittable state in transmitting direction (DIR = 1): When one of the following is satisfied: 1. CPU or DMAC/DTC wrote data of the maximum packet size to the FIFO buffer 2. CPU or DMAC/DTC wrote data of the short packet size (including 0 bytes) to the FIFO buffer and set the BVAL flag in the port control register to 1
1	Condition for FIFO buffer readable state in receiving direction (DIR = 0): 1. The byte count of data received in the FIFO buffer allocated to the selected pipe is equal to the allocated byte count ((BUFSIZE + 1) × 64). 2. The USBHS received a short packet, other than a zero-length packet. 3. The USBHS received a zero-length packet when data was already contained in the FIFO buffer allocated to the selected pipe. 4. Software received the number of packets specified for the transaction counter set for the selected pipe.
	Conditions for FIFO buffer transmittable state in transmitting direction (DIR = 1): When one of the following is satisfied. 1. The amount of data written by CPU or DMAC/DTC is equal to the size of the FIFO buffer allocated to the selected pipe. 2. CPU or DMAC/DTC wrote data of smaller size than that of the FIFO buffer allocated to the selected pipe (including 0 bytes) and set the BVAL flag in the port control register to 1. 3. CPU or DMAC/DTC wrote data of smaller size than that of one FIFO buffer allocated to the selected pipe (including 0 bytes) and asserted the DENDx_N signal on the last write.

DBLB bit (Double Buffer Mode)

The DBLB bit selects either single or double buffer mode for the FIFO buffer used by the selected pipe. The bit is valid for pipes 1 to 5.

When the software sets this bit to 1, the USBHS allocates twice the FIFO buffer size specified in the PIPEBUF.BUFSIZE[5:0] bits for the selected pipe. The FIFO buffer size that the USBHS allocates to the selected pipe is as follows:

$$(\text{BUFSIZE} + 1) \times 64 \times (\text{DBLB} + 1) \text{ [bytes]}$$

BFRE bit (BRDY Interrupt Operation Specification)

The BFRE bit specifies the BRDY interrupt generation timing from the USBHS to the CPU for the selected pipe.

When the software sets the BFRE bit to 1 and the selected pipe is in the receiving direction, the USBHS detects the transfer completion and generates the BRDY interrupt on reading the packet.

When a BRDY interrupt is generated with this setting, the software must write 1 to the BCLR bit in the port control register. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to the BCLR bit.

When the BFRE bit is set to 1 by software and the selected pipe is in the transmitting direction, the USBHS does not generate the BRDY interrupt. For details, see section 37.3.6.1. BRDY Interrupt.

TYPE[1:0] bits (Transfer Type)

The TYPE[1:0] bits specify the transfer type for the pipe selected in the PIPESEL.PIPESEL[3:0] bits. Before setting PID to BUF and starting USB communication on the selected pipe, set the TYPE[1:0] bits to a value other than 00b.

37.2.35 PIPEBUF : Pipe Buffer Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x06A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	BUFSIZE[4:0]					—	—	BUFNMB[7:0]							
Value after reset:	x	0	0	0	0	0	x	x	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	BUFNMB[7:0]	Buffer Number Specifies the FIFO buffer number of the selected pipe (0x04 to 0x87).	R/W
9:8	—	The read values are undefined. The write value should be 0.	R/W
14:10	BUFSIZE[4:0]	Buffer Size 0x00: 64 bytes 0x01: 128 bytes : 0x1F: 2 KB	R/W
15	—	The read value is undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note: Only set the bits in the PIPEBUF register while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting these bits, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the CSSTS and PBUSY flags through the software is not necessary.

BUFNMB[7:0] bits (Buffer Number)

The BUFNMB[7:0] bits specify the first block number of the FIFO buffer to be allocated to the selected pipe.

The USBHS allocates the FIFO buffer blocks to the selected pipe as follows:

Block number: BUFNMB to block number: BUFNMB + (BUFSIZE + 1) × (DBLB + 1) - 1

Set a value within the memory size range for these bits (0 [0x00] to 8640 [0x87] for 8.5 KB), while observing the following conditions:

- 0x00 is for DCP only
- 0x04 is for pipe 6 only, but is available for other pipes when pipe 6 is not used. When pipe 6 is selected, writes to these bits are disabled. The USBHS automatically allocates 0x04 to the BUFNMB bits for pipe 6.
- 0x05 is for pipe 7 only, but is available for other pipes when pipe 7 is not used. When pipe 7 is selected, writes to these bits are disabled. The USBHS automatically allocates 0x05 to the BUFNMB bits for pipe 7.
- 0x06 is for pipe 8 only, but is available for other pipes when pipe 8 is not used. When pipe 8 is selected, writes to these bits are disabled. The USBHS automatically allocates 0x06 to the BUFNMB bits for pipe 8.
- 0x07 is for pipe 9 only, but is available for other pipes when pipe 9 is not used. When pipe 9 is selected, writes to these bits are disabled. The USBHS automatically allocates 0x07 to the BUFNMB bits for pipe 9.

BUFSIZE[4:0] bits (Buffer Size)

The BUFSIZE[4:0] bits specify the FIFO buffer size (number of blocks) to be allocated to the selected pipe. One block is 64 bytes.

When the software sets the DBLB bit to 1, the USBHS allocates twice the FIFO buffer size specified in these bits to the selected pipe. The DBLB = 1 setting is valid for pipes 1 to 5.

The USBHS allocates the FIFO buffer blocks to the selected pipe as follows:

(BUFSIZE + 1) × 64 × (DBLB + 1) [bytes]

Set the value within the following range:

- For pipes 1 to 5, set a value from 0x00 to 0x1F (up to 2 KB)

- For pipes 6 to 9, only set a value of 0x00 (64 bytes)

37.2.36 PIPEMAXP : Pipe Maximum Packet Size Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x06C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DEVSEL[3:0]				—	MXPS[10:0]										
Value after reset:	0	0	0	x	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	MXPS[10:0]*1 *2	Maximum Packet Size ● Pipes 1 and 2 1 byte (0x001) to 1024 bytes (0x400) ● Pipes 3 to 5 8 bytes (0x008), 16 bytes (0x010), 32 bytes (0x020), 64 bytes (0x040), 512 bytes (0x200) (Bits 2 to 0 not supported.) ● Pipes 6 to 9 1 byte (0x001) to 64 bytes (0x040) (Bits 10 to 7 not supported.)	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
15:12	DEVSEL[3:0]*3	Device Select 0x0: Address 0x0 0x1: Address 0x1 ⋮ 0x9: Address 0x9 0xA: Address 0xA Others: Reserved	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. The initial value of the MXPS[10:0] bits is 0x00 when no pipe is selected in the PIPESEL.PIPESEL[3:0] bits and 0x40 when a pipe is selected.

Note 2. Only set the MXPS[10:0] bits while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting these bits, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the CSSTS and PBUSY flags through the software is not necessary.

Note 3. Only set the DEVSEL[3:0] bits while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting these bits, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the PBUSY flag through the software is not necessary.

MXPS[10:0] bits (Maximum Packet Size)

The MXPS[10:0] bits specify the maximum data payload (maximum packet size) for the selected pipe.

Set these bits to the appropriate value for each transfer type based on the USB 2.0 specification. When MXPS[10:0] = 0, do not write to the FIFO buffer or set PID to BUF. These writes have no effect.

To communicate on an isochronous pipe using a split transaction, set the value in the MXPS[10:0] bits to 188 bytes or less.

DEVSEL[3:0] bits (Device Select)

In host controller mode, the DEVSEL[3:0] bits specify the address of the target device for USB communication. Set up the device address in the associated DEVADDn (n = 0 to A) register first, and then set these bits to the corresponding value. To set the DEVSEL[3:0] bits to 0x2, for example, first set the address in the DEVADD2 register.

In device controller mode, set these bits to 0x0.

37.2.37 PIPEPERI : Pipe Cycle Control Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x06E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	IFIS	—	—	—	—	—	—	—	—	—	IITV[2:0]		
Value after reset:	x	x	x	0	x	x	x	x	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
2:0	IITV[2:0]*1	Interval Error Detection Interval Specifies the interval error detection timing for the selected pipe as the n-th power of 2 of the frame timing.	R/W
11:3	—	These bits are read as 0. The write value should be 0.	R/W
12	IFIS	Isochronous IN Buffer Flush 0: Do not flush buffer 1: Flush buffer	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Only set the IITV[2:0] bits while PID is NAK. Before setting these bits, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the PBUSY flag through the software is not necessary.

PIPEPERI selects whether the buffer is flushed or not when an interval error occurred during isochronous IN transfers, and sets the interval error detection interval for pipes 1 to 9.

IITV[2:0] bits (Interval Error Detection Interval)

To change the IITV[2:0] bits to another value after they are set and USB communication is performed, set the PIPEnCTR.PID[1:0] bits to 00b (NAK) and then set the PIPEnCTR.ACLRM bit to 1 to initialize the interval timer.

The IITV[2:0] bits are not provided for pipes 3 to 5. Write 000b to bit positions of the IITV[2:0] bits associated with pipes 3 to 5.

IFIS bit (Isochronous IN Buffer Flush)

The IFIS bit specifies whether to flush the buffer when the pipe specified in the PIPESEL.PIPESEL[3:0] bits is used for isochronous IN transfers.

In device controller mode when the selected pipe is for isochronous IN transfers, the USBHS automatically clears the FIFO buffer if the USBHS fails to receive the IN token from the USB host within the interval set in the IITV[2:0] bits in terms of frames.

When double buffering is specified (PIPECFG.DBLB = 1), the USBHS only clears the data in the previously used plane.

The USBHS clears the FIFO buffer on receiving the SOF packet immediately after the frame in which the USBHS expected to receive the IN token. Even if the SOF packet is corrupted, the FIFO buffer is cleared at the time the SOF packet is expected to be received by using the internal complementation function.

In host controller mode, set the IITV[2:0] bits to 000b.

Set the IITV[2:0] bits to 000b when the selected pipe is not used for isochronous transfers.

37.2.38 PIPEnCTR : Pipe n Control Register (n = 1 to 9)

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x070 + 0x2 × (n - 1)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSTS	INBUFM	CSCLR	CSSTS	—	ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	x	0	0	0	0	0	0	x	x	x	0	0

Bit	Symbol	Function	R/W
1:0	PID[1:0]	Response PID 0 0: NAK response 0 1: BUF response (depends on buffer state) 1 0: STALL response 1 1: STALL response	R/W
4:2	—	The read values are undefined. The write value should be 0.	R/W
5	PBUSY	Pipe Busy Flag 0: Pipe n not in use for the transaction 1: Pipe n in use for the transaction	R
6	SQMON	Sequence Toggle Bit Monitor Flag 0: DATA0 1: DATA1	R
7	SQSET	Sequence Toggle Bit Set*1 0: Invalid (writing 0 has no effect) 1: Set the expected value for the next transaction to DATA1. This bit is read as 0.	R/W
8	SQCLR	Sequence Toggle Bit Clear*1 0: Invalid (writing 0 has no effect) 1: Clear the expected value for the next transaction to DATA0	W
9	ACLRM	Auto Buffer Clear Mode*2 0: Disable 1: Enable (initialize all buffers)	R/W
10	ATREPM	Auto Response Mode*1 *3 0: Disable auto response mode 1: Enable auto response mode	R/W
11	—	The read value is undefined. The write value should be 0.	R/W
12	CSSTS	CSSTS Status Flag 0: Start-split (SSPLIT) transaction, or processing for devices that are not using split transactions, in progress. 1: Complete-split (CSPLIT) transaction in progress.	R
13	CSCLR	CSPLIT Status Clear 0: Invalid (writing 0 has no effect) 1: Clear CSSTS to 0	W
14	INBUFM	Transmit Buffer Monitor Flag*3 0: No data to be transmitted is in the FIFO buffer 1: Data to be transmitted is in the FIFO buffer	R
15	BSTS	Buffer Status Flag 0: Buffer access disabled 1: Buffer access enabled	R

Note: S-TYPE-3, P-TYPE-3

Note 1. Only set the ATREPM, SQCLR, and SQSET bits while PID is NAK. Before setting this bit, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the PBUSY flag through the software is not necessary.

Note 2. Only set the ACLRM bit while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting this bit, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the PBUSY flag through the software is not necessary.

Note 3. The ATREPM bit and the INBUFM flag in the PIPE6CTR to PIPE9CTR registers are reserved. The read value is undefined. The write value must be 0.

PID[1:0] bits (Response PID)

The PID[1:0] bits specify the response type for the next transaction on the selected pipe.

The default PID[1:0] setting is NAK. Change the PID[1:0] setting to BUF to use the associated pipe for USB transfer. Table 37.10 and Table 37.11 show the basic operations of the USBHS (when there are no errors in the communication packets) based on the PID[1:0] bit setting.

After changing the PID[1:0] setting from BUF to NAK through the software during USB communication on the selected pipe, check that the PBUSY bit is 1 to see if USB transfer on the selected pipe has actually entered the NAK state. If the USBHS changes the PID[1:0] bits to NAK, checking the PBUSY bit through the software is not necessary.

The USBHS changes the PIPEnCTR.PID[1:0] setting in the following cases:

- The USBHS sets PID to NAK on recognizing completion of the transfer when the selected pipe is in the receiving direction and the PIPECFG.SHTNAK bit for the selected pipe is set to 1 by software
- The USBHS sets PID to STALL (11b) on receiving a data packet with a payload exceeding the maximum packet size of the selected pipe
- The USBHS sets PID to NAK on detecting a USB bus reset in device controller mode
- The USBHS sets PID to NAK on detecting a reception error, such as a CRC error, three consecutive times in host controller mode
- The USBHS sets PID to STALL (11b) on receiving the STALL handshake in host controller mode

To specify the response type, set the PID[1:0] bits as follows:

- To transition from NAK (00b) to STALL, set 10b
- To transition from BUF (01b) to STALL, set 11b
- To transition from STALL (11b) to NAK, set 10b and then 00b
- To transition from STALL to BUF, set 00b (NAK) and then 01b (BUF)

Table 37.10 Operation of the USBHS based on the PIPEnCTR.PID[1:0] setting in host controller mode

PID[1:0] value	Transfer type (TYPE[1:0] value)	Transfer direction (DIR value)	USBHS operation
00b (NAK)	Does not depend on the setting	Does not depend on the setting	Does not issue tokens
01b (BUF)	Bulk or Interrupt	Does not depend on the setting	Issues tokens when the DVSTCTR0.UACT bit is 1 and the FIFO buffer associated with the selected pipe is ready for transmission and reception. Does not issue tokens when the DVSTCTR0.UACT bit is 0 or the FIFO buffer associated with the selected pipe is not ready for transmission or reception.
	Isochronous	Does not depend on the setting	Issues tokens when the DVSTCTR0.UACT bit is 1, regardless of the state of the FIFO buffer associated with the selected pipe. Does not issue tokens when UACT = 0.
10b (STALL) or 11b (STALL)	Does not depend on the setting	Does not depend on the setting	Does not issue tokens.

Table 37.11 Operation of the USBHS based on the PIPEnCTR.PID[1:0] setting in device controller mode

PID[1:0] value	Transfer type (TYPE[1:0] value)	Transfer direction (DIR value)	USBHS operation
00b (NAK)	Bulk or Interrupt	Does not depend on the setting	Returns NAK in response to the token from the USB host
	Isochronous	Receiving direction (DIR = 0)	Returns nothing in response to the token from the USB host
		Transmitting direction (DIR = 1)	Transmits a zero-length packet in response to the token from the USB host
01b (BUF)	Bulk	Receiving direction (DIR = 0)	Receives data and returns ACK or NYET in response to the OUT token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception. Otherwise, returns NAK. Returns ACK in response to the PING token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception. Otherwise, returns NAK.
	Interrupt	Receiving direction (DIR = 0)	Receives data and returns ACK response in response to the OUT token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception. Otherwise, returns NAK.
	Bulk or Interrupt	Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the FIFO buffer associated with the selected pipe is ready for transmission. Otherwise, returns NAK.
	Isochronous	Receiving direction (DIR = 0)	Receives data in response to the OUT token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception. Otherwise, discards the data.
		Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the associated FIFO buffer is ready for transmission. Otherwise, transmits a zero-length packet.
10b (STALL) or 11b (STALL)	Bulk or Interrupt	Does not depend on the setting.	Returns STALL in response to the token from the USB host
	Isochronous	Does not depend on the setting.	Returns nothing in response to the token from the USB host

PBUSY flag (Pipe Busy Flag)

The PBUSY flag indicates whether the selected pipe is being used for the current transaction.

The USBHS changes the PBUSY bit from 0 to 1 on start of the USB transaction for the selected pipe, and changes the PBUSY bit from 1 to 0 on completion of one transaction.

Reading the PBUSY bit by software after PID is set to NAK allows you to check whether changing the pipe setting is possible. For details, see section 37.3.7.1. Pipe Control Register Switching Procedures.

SQMON flag (Sequence Toggle Bit Monitor Flag)

The SQMON flag indicates the expected value of the sequence toggle bit for the next transaction of the selected pipe.

When the selected pipe is not the isochronous transfer type, the USBHS toggles the SQMON flag on successful completion of the transaction. However, the USBHS does not toggle the SQMON flag when a DATA-PID mismatch occurs during transfer in the receiving direction.

SQSET bit (Sequence Toggle Bit Set)

Setting the SQSET bit to 1 through the software causes the USBHS to set DATA1 as the expected value of the sequence toggle bit for the next transaction on the selected pipe.

SQCLR bit (Sequence Toggle Bit Clear)

Setting the SQCLR bit to 1 through the software causes the USBHS to clear the expected value of the sequence toggle bit for the next transaction on the selected pipe to DATA0.

In host controller mode, when this bit is set to 1 for a bulk OUT transfer pipe, the USBHS starts the next transfer for the selected pipe from a PING token.

ACLRM bit (Auto Buffer Clear Mode)

The ACLRM bit enables or disables auto buffer clear mode for the selected pipe. To completely clear the data in the FIFO buffer allocated to the selected pipe, write 1 and then 0 to the ACLRM bit continuously.

Table 37.12 shows the data cleared by writing 1 and 0 continuously to the ACLRM bit and the cases in which this processing is required.

Table 37.12 Data cleared by the USBHS when ACLRM = 1

Number	Data cleared by setting the ACLRM bit	Situations requiring data clear
1	All data in the FIFO buffer allocated to the selected pipe (two FIFO buffers in double buffer mode)	When clearing all data in the FIFO buffer allocated to the selected pipe
2	Interval count value when the selected pipe is the isochronous transfer type	When resetting the interval count value

ATREPM bit (Auto Response Mode)

The ATREPM bit enables or disables auto response mode for the selected pipe.

This bit can be set to 1 in device controller mode when the selected pipe is the bulk transfer type. When the bit is set to 1, the USBHS responds to the token from the USB host as follows:

- When the selected pipe is set for bulk IN transfers (PIPECFG.TYPE[1:0] = 01b and PIPECFG.DIR = 1):
  - When the ATREPM bit = 1 and PID = BUF, the USBHS transmits a zero-length packet in response to the IN token.
  - The USBHS updates (allows toggling of) the sequence toggle bit (DATA-PID) each time the USBHS receives ACK from the USB host. In a single transaction, the IN token is received, a zero-length packet is transmitted, and then ACK is received. The USBHS does not generate the BRDY or BEMP interrupt.
- When the selected pipe is set for bulk OUT transfers (PIPECFG.TYPE[1:0] = 01b and PIPECFG.DIR = 0):
  When the ATREPM bit = 1 and PID = BUF, the USBHS returns NAK in response to the OUT token or PING token and generates an NRDY interrupt.

For USB communication in auto response mode, set the ATREPM bit to 1 while the FIFO buffer is empty. Do not write to the FIFO buffer during USB communication in auto response mode. When the selected pipe uses isochronous transfer, always set this bit to 0.

In host controller mode, always set the ATREPM bit to 0.

CSSTS flag (CSSTS Status Flag)

In host controller mode, the CSSTS flag indicates the complete-split status of a split transaction. It is valid for pipes that are not the isochronous transfer type.

The USBHS sets the CSSTS flag to 1 at the beginning of the complete-split transaction, and sets the CSSTS flag to 0 on detecting completion of the complete-split transaction. If a detach event is detected during the transaction, the CSSTS flag might stay set to 1. In this case, clear the CSSTS flag by setting the CSCLR bit to 1.

Values read from the CSSTS flag in device controller mode are invalid.

CSCLR bit (CSPLIT Status Clear)

In host controller mode, if the software sets the CSCLR bit to 1, the USBHS clears the CSSTS flag to 0. In split transactions, set the CSCLR bit to 1 by software to force the next transfer to restart from start-split. Because the USBHS automatically clears the CSSTS flag to 0 at the end of a successful complete-split transaction in a normal split transaction, clearing the flag through software is not required. Only clear the CSSTS flag using the CSCLR bit when the DVSTCTR0.UACT bit is set to 0 or when no transfer was made after a detach detect. If the CSCLR bit is set to 1 while the CSSTS flag is 0, the CSSTS flag remains 0.

In device controller mode, always write 0 to the CSCLR bit.

INBUFM flag (Transmit Buffer Monitor Flag)

The INBUMFM flag indicates the FIFO buffer status for the selected pipe in the transmitting direction.

When the selected pipe is set in the transmitting direction (PIPECFG.DIR = 1), the USBHS sets this bit to 1 when the CPU or DMAC/DTC completes writing data to at least one FIFO buffer plane.

The USBHS sets this bit to 0 when the USBHS completes transmission of data from the FIFO buffer plane to which all the data is written. In double buffer mode (PIPECFG.DBLB = 1), the USBHS sets the INBUFM flag to 0 when the USBHS completes transmission of data from the two FIFO buffer planes before the CPU or DMAC/DTC completes writing data to one FIFO buffer plane.

The INBUFM flag indicates the same value as the BSTS flag when the selected pipe is in the receiving direction (PIPECFG.DIR = 0).

BSTS flag (Buffer Status Flag)

The BSTS flag indicates the FIFO buffer status for the selected pipe. The meaning of the BSTS flag depends on the PIPECFG.DIR, PIPECFG.BFRE, and DnFIFOSEL.DCLRM settings, as shown in Table 37.13.

Table 37.13 BSTS flag operation

DIR value	BFRE value	DCLRM value	Meaning of BSTS flag
0	0	0	Sets to 1 when receive data can be read from the FIFO buffer, and clears to 0 on completion of data read
		1	Setting prohibited
	1	0	Sets to 1 when receive data can be read from the FIFO buffer, and clears to 0 when the software sets the BCLR bit in the port control register to 1 after the data read is complete
		1	Sets to 1 when receive data can be read from the FIFO buffer, and clears to 0 on completion of data read
1	0	0	Sets to 1 when transmit data can be written to the FIFO buffer, and clears to 0 on completion of data write
		1	Setting prohibited
	1	0	Setting prohibited
		1	Setting prohibited

37.2.39 PIPEnTRE : Pipe n Transaction Counter Enable Register (n = 1 to 5)

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x090 + 0x4 × (n - 1)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TRENB	TRCLR	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	0	0	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
7:0	—	The read values are undefined. The write value should be 0.	R/W
8	TRCLR	Transaction Counter Clear 0: Invalid (writing 0 has no effect) 1: Clear current counter value	R/W
9	TRENB	Transaction Counter Enable 0: Disable transaction counter 1: Enable transaction counter	R/W
15:10	—	The read values are undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note: Only change the PIPEnTRE register settings while the PIPEnCTR.CSSTS flag is 0 and the PIPEnCTR.PID[1:0] bits are 00b (NAK response). Only change the PIPEnCTR.PID[1:0] bits of the selected pipe from 01b (BUF response) to 00b (NAK response) after confirming that the value of the PIPEnCTR.PBUSY and PIPEnCTR.CSSTS flags is 0. However, software processing to check the PIPEnCTR.PBUSY flag is not required if the USBHS has changed the PID[1:0] bits to 00b (NAK response).

TRCLR bit (Transaction Counter Clear)

When the TRCLR bit sets to 1, the USBHS clears the count value of the transaction counter associated with the selected pipe and then clears the TRCLR bit to 0.

TRENB bit (Transaction Counter Enable)

The TRENB bit enables or disables the transaction counter.

For receiving pipes, setting the TRENB bit to 1 after setting the total number of the packets to be received in the PIPEnTRN.TRNCNT[15:0] bits through the software allows the USBHS to control hardware on having received the number of packets equal to the TRNCNT[15:0] setting as follows:

- When the PIPECFG.SHTNAK bit is 1, the USBHS changes the PID bits to NAK for the associated pipe on having received the number of packets equal to the TRNCNT[15:0] setting
- When the PIPECFG.BFRE bit is 1, the USBHS asserts the BRDY interrupt on having received the number of packets equal to the TRNCNT[15:0] setting and then reading the last received data.

For transmitting pipes, set the TRENB bit to 0.

When the transaction counter is not used, set this bit to 0. When the transaction counter is used, set the TRNCNT[15:0] bits before setting this bit to 1. Set this bit to 1 before receiving the first packet to be counted by the transaction counter.

37.2.40 PIPEnTRN : Pipe n Transaction Counter Register (n = 1 to 5)

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x092 + 0x4 × (n - 1)

Bit position:	15															0
Bit field:	TRNCNT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TRNCNT[15:0]	Transaction Counter*1 • When written to: Specifies the total packets (number of transactions) to be received by pipe n. • When read from: When PIPEnTRE.TRENB is 0, indicates the specified number of transactions. When PIPEnTRE.TRENB is 1, indicates the current transaction count.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Only set the TRNCNT[15:0] bits while PID is NAK and PIPEnTRE.TRENB is 0. Before setting these bits, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the PBUSY flag through the software is not necessary.

The PIPEnTRN registers retain their settings during a USB bus reset.

TRNCNT[15:0] bits (Transaction Counter)

The USBHS increments the value of the TRNCNT[15:0] bits by one when all of the following conditions are satisfied on receiving the packet:

- The PIPEnTRE.TRENB bit is 1
- (TRNCNT[15:0] set value ≠ current counter value + 1) on receiving the packet
- The payload of the received packet agrees with the PIPEMAXP.MXPS[8:0] setting

The USBHS clears the value of the TRNCNT[15:0] bits to 0 when any of the following conditions is satisfied.

All of the following conditions are satisfied:

- The PIPEnTRE.TRENB bit = 1
- (TRNCNT[15:0] set value = current counter value + 1) on receiving the packet
- The payload of the received packet agrees with the PIPEMAXP.MXPS[8:0] setting

Both the following conditions are satisfied:

- The PIPEnTRE.TRENB bit = 1
- The USBHS received a short packet

Both the following conditions are satisfied:

- The PIPEnTRE.TRENB bit = 1
- The PIPEnTRE.TRCLR bit was set to 1 by software

For transmitting pipes, set the TRNCNT[15:0] bits to 0. When the transaction counter is not used, set the TRNCNT[15:0] bits to 0.

Setting the number of transactions to be transferred to the TRNCNT[15:0] bits is enabled only when the PIPEnTRE.TRENB bit is 0. To set the number of transactions to be transferred, set the TRCLR bit to 1 to clear the current counter value before setting the PIPEnTRE.TRENB bit to 1.

37.2.41 DEVADDn : Device Address n Configuration Register (n = 0 to 9, A)

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x0D0 + 0x2 × n (n = 0 to 9)
0x0E4 (n = A)

Bit position:	15	14				10			7		5	4	3	2	1	0
Bit field:	—	UPPHUB[3:0]				HUBPORT[2:0]			USBSPD[1:0]		—	—	—	—	—	—
Value after reset:	x	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x

Bit	Symbol	Function	R/W
5:0	—	The read values are undefined. The write value should be 0.	R/W
7:6	USBSPD[1:0]	Transfer Speed of Communication Target Device 0 0: Do not use DEVADDn 0 1: Low speed 1 0: Full speed 1 1: High speed	R/W
10:8	HUBPORT[2:0]	Communication Target Connecting Hub Port 0 0 0: Connect directly to the USBHS port Others: Port number of the hub	R/W
14:11	UPPHUB[3:0]	Communication Target Connecting Hub Register 0x0: Connect directly to the USBHS port Others: USB address of the hub. The value as 0xB or more is reserved.	R/W
15	—	The read value is undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The DEVADDn register specifies the transfer speed of the peripheral device that is the communication target for pipes 0 to 9.

In host controller mode, set all DEVADDn bits before starting communication to any pipes. Only change the bits in DEVADDn when no valid pipes are using the bit settings. A valid pipe is defined as one that satisfies both of the following conditions:

- DEVADDn is selected in the DEVSEL[3:0] bits
- The PID[1:0] bits are set to BUF for the selected pipe, or the selected pipe is the DCP with the DCPCTR.SUREQ bit set to 1

In device controller mode, set all bits in this register to 0.

USBSPD[1:0] bits (Transfer Speed of Communication Target Device)

The USBSPD[1:0] bits specify the USB transfer speed of the target peripheral device. In host controller mode, the USBHS generates packets based on the USBSPD[1:0] setting. In device controller mode, set these bits to 00b.

HUBPORT[2:0] bits (Communication Target Connecting Hub Port)

In host controller mode, the USBHS generates packets based on the HUBPORT[2:0] setting when performing a split transaction.

UPPHUB[3:0] bits (Communication Target Connecting Hub Register)

In host controller mode, the USBHS generates packets based on the UPPHUB[3:0] setting when performing a split transaction.

37.2.42 LPCTRL : Low Power Control Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x100

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	HWUPM	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	0	0	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
6:0	—	The read values are undefined. The write value should be 0.	R/W
7	HWUPM	Resume Return Mode Setting 0: Hardware does not recover while CPU clock inactive 1: Hardware recovers while CPU clock inactive	R/W
8	—	This bit is read as 0. The write value should be 0.	R/W
15:9	—	The read values are undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

HWUPM bit (Resume Return Mode Setting)

The HWUPM bit specifies whether to enable hardware processing for return from low power mode even while the CPU clock is inactive.

In device controller mode, processing for return from low power mode on detecting Resume is enabled even while the CPU clock is inactive.

This bit specifies whether to detect Resume while the CPU clock is inactive. The PL1CTRL1.L1EXTMD bit controls whether to make a hardware return. To make a hardware return from the LPM L1 low power state while the CPU clock is inactive, set this bit and the PL1CTRL1.L1EXTMD bit to 1.

37.2.43 LPSTS : Low Power Status Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x102

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	SUSPENDM	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	0	x	0	x	x	x	0	x	x	x	x	0	x	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
2	—	The read value is undefined. The write value should be 0.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
7:4	—	The read values are undefined. The write value should be 0.	R/W
8	—	This bit is read as 0. The write value should be 0.	R/W
11:9	—	The read values are undefined. The write value should be 0.	R/W
12	—	This bit is read as 0. The write value should be 0.	R/W
13	—	The read value is undefined. The write value should be 0.	R/W
14	SUSPENDM	UTMI SuspendM Control 0: UTMI suspension mode 1: UTMI normal mode	R/W
15	—	The read value is undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

SUSPENDM bit (UTMI SuspendM Control)

The SUSPENDM bit controls the SuspendM signal to be sent to the PHY designed under the UTMI specification. The initial value is 0 with the UTMI is in suspension mode.

Set this bit to 1 to supply the PHY clock to operate the USB2.0 host or device controller.

In compliance with the UTMI specification, clock output is normally controlled by the SuspendM signal. When the SUSPENDM bit is 0, the clock to LINK is stopped. Because the PHY in this MCU follows the UTMI specification, setting the SUSPENDM bit to 1 is required to supply the PHY clock. For the clock settings, see section 37.3.3. Supplying the Clock.

When the SUSPENDM bit is 0, the USBHS cannot be written to but can be read from. The registers listed in Table 37.14 are writable even when the SUSPENDM bit is 0.

Table 37.14 Registers that can be written to by software when SUSPENDM = 0

Address	Register or bit name
0x4006_0000	SYSCFG register
0x4006_0002	BUSWAIT register
0x4006_0032	INTENB1.PDDETINTE bit
0x4006_0100	LPCTRL register
0x4006_0102	LPSTS register
0x4006_0140	BCCTRL register

The value written to the SYSCFG register while the PHY clock is inactive is updated only after the PHY clock begins oscillating. The PHY clock oscillates in the following cases described in this section.

When SUSPENDM bit is set to 1, the PLLSTA.PLLLOCK flag is set to 1 after the predetermined time has passed. The USB-PHY internal PLL is stopped when the SUSPENDM bit is set to 0.

For details on CL-only mode, see section 37.2.17. PHYSET : PHY Setting Register.

If the PL1CTRL1.L1EXTMD bit is 0, setting or clearing of this bit is controlled by software. If the PL1CTRL1.L1EXTMD bit is 1, transitions to the L1 or L2 state of this bit are controlled by software and recovery from the L1 or L2 state is controlled by hardware.

37.2.44 BCCTRL : Battery Charging Control Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x140

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PDDETSTS	CHGDETSTS	—	—	DCPMODE	VDMSRCE	IDPSINKE	VDPSRCE	IDMSINKE	IDPSRCE
Value after reset:	x	x	x	x	x	x	0	0	x	x	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IDPSRCE	IDPSRC Control*2 0: Disable IDP_SRC circuit 1: Enable IDP_SRC circuit	R/W
1	IDMSINKE	IDMSINK Control*2 0: Disable IDM_SINK circuit 1: Enable IDM_SINK circuit	R/W*1
2	VDPSRCE	VDPSRC Control*2 0: Disable VDP_SRC circuit 1: Enable VDP_SRC circuit	R/W
3	IDPSINKE	IDPSINK Control*2 0: Disable IDP_SINK circuit 1: Enable IDP_SINK circuit	R/W
4	VDMSRCE	VDMSRC Control*2 0: Disable VDM_SRC circuit 1: Enable VDM_SRC circuit	R/W
5	DCPMODE	DCP Mode Control 0: Disable RDCP_DAT resistor 1: Enable RDCP_DAT resistor	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	CHGDETSTS	CHGDET Status Flag 0: The CHGDET pin is at low level 1: The CHGDET pin is at high level	R
9	PDDETSTS	PDDET Status Flag 0: The PDDET pin is at low level 1: The PDDET pin is at high level	R
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. All bits in the BCCTRL register can be changed while the UTMI clock is inactive.
Note 2. In device controller mode, set the IDPSRCE, IDMSINKE, VDPSRCE, IDPSINKE, and VDMSRCE bits to 1 after setting the SYSCFG.DRPD bit to 0.

IDPSRCE bit (IDPSRC Control)

In device controller mode, set the IDPSRCE bit to 1 to perform data contact detection.

The Battery Charging Standard provides two ways to handle data contact detection, one through the software and one using hardware to contact the data line. The IDPSRE bit uses the hardware method.

When the IDPSRE bit is set to 1, the USBHS enables the IDP_SRC circuit and, at the same time, controls D- pull-down.

(D- pull-down is controlled with the VUH_DMPULLDOWN signal.)

IDMSINKE bit (IDMSINK Control)

In device controller mode, set the IDMSINKE bit to 1 to perform primary detection.

VDPSRCE bit (VDPSRC Control)

In device controller mode, set the VDPSRCE bit to 1 to perform primary detection.

IDPSINKE bit (IDPSINK Control)

In device controller mode, set the IDPSINKE bit to 1 to perform secondary detection. In host controller mode, set this bit to 1 to enable the portable device detection circuit.

VDMSRCE bit (VDMSRC Control)

In device controller mode, set the VDMSRCE bit to 1 to perform secondary detection. Setting this bit to 1 enables the DCP detection circuit. In host controller mode, set this bit to 1 when a portable device is detected. Setting this bit to 1 allows the device that is performing primary detection to determine the charger detection method.

DCPMODE bit (DCP Mode Control)

Set the DCPMODE bit to 1 to operate as a dedicated charging port (DCP). Setting this bit to 1 disables USB communication.

CHGDETSTS flag (CHGDET Status Flag)

The CHGDETSTS flag indicates the charger port detection state.

PDDETSTS flag (PDDET Status Flag)

The PDDETSTS flag indicates the following states based on the controller mode:

- In host controller mode: PD detection state
- In device controller mode: DCP detection state

37.2.45 PL1CTRL1 : Function L1 Control Register 1

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x144

Bit position:	15	14	13		11				7				3	2		0
Bit field:	—	L1EXTMD	—		HIRDTHR[3:0]				DVSQ[3:0]				L1NEGOMD	L1RESPMD[1:0]		L1RESPEN
Value after reset:	x	0	x	x	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	L1RESPEN	L1 Response Enable 0: Do not support LPM 1: Support LPM	R/W
2:1	L1RESPMD[1:0]	L1 Response Mode 0 0: NYET response 0 1: ACK response 1 0: STALL response 1 1: Response based on L1NEGOMD setting	R/W
3	L1NEGOMD	L1 Response Negotiation Control This bit is only valid when the L1RESPMD[1:0] value is 11b. 0: Return ACK when received HIRD is larger than HIRDTHR[3:0]. Otherwise (including when HIRD = HIRDTHR[3:0]), return NYET 1: Return ACK when received HIRD is smaller than HIRDTHR[3:0]. Otherwise (including when HIRD = HIRDTHR[3:0]), return NYET	R/W

Bit	Symbol	Function	R/W
7:4	DVSQ[3:0]	DVSQ Extension Flag 0 0 0 0: Powered state 0 0 0 1: Default state 0 0 1 0: Address state 0 0 1 1: Configured state 0 1 x x: Suspend state 1 0 x x: L1 state	R
11:8	HIRDTHR[3:0]	L1 Response Negotiation Threshold Value HIRD threshold value used when the L1RESPMD[1:0] bits are 11b. The format is the same as the HIRD field in HL1CTRL.	R/W
13:12	—	The read values are undefined. The write value should be 0.	R/W
14	L1EXTMD	PHY Control Mode at L1 Return 0: Do not set LPSTS.SUSPENDM bit through hardware when Host K is received 1: Set LPSTS.SUSPENDM bit through hardware when Host K is received	R/W
15	—	The read value is undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

L1RESPEN bit (L1 Response Enable)

If the USBHS receives an LPM token while the L1RESPEN bit is 0, it returns no response. If the USBHS receives an LPM token while this bit is 1, it returns a response based on the L1RESPMD[1:0] setting.

L1RESPMD[1:0] bits (L1 Response Mode)

When the L1RESPEN bit is set to 1, the USBHS returns a response to the LPM token based on the setting in the L1RESPMD[1:0] bits.

L1NEGOMD bit (L1 Response Negotiation Control)

The L1NEGOMD bit specifies the negotiation function for the HIRD value.

HIRDTHR[3:0] bits (L1 Response Negotiation Threshold Value)

The HIRDTHR[3:0] bits specify the HIRD threshold value used for L1NEGOMD. The format of the set value is the same as the HIRD field in HL1CTRL.

L1EXTMD bit (PHY Control Mode at L1 Return)

The L1EXTMD bit specifies the LPSTS.SUSPENDM bit control method when a host K signal is received in the L1 state while the LPSTS.SUSPENDM bit is 0 and the PHY is inactive.

Similar to the Suspend constraints, because the minimum host K period is 50 µs, the PHY might not recover within the host K period specified for software settings on return. The initial value is within software control, so set this bit to 1 during the initialization process when the L1 state is supported.

The LPSTS.SUSPENDM bit is controlled by software on transition to the L1 state regardless of the setting in this bit. It is not cleared by hardware.

When this bit is set to 1, the LPSTS.SUSPENDM bit is also set to 1 on return from L2.

37.2.46 PL1CTRL2 : Function L1 Control Register 2

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x146

Bit position:	15	14	13	12	11				7	6	5	4	3	2	1	0
Bit field:	—	—	—	RWEMON	HIRDMON[3:0]				—	—	—	—	—	—	—	—
Value after reset:	x	x	x	0	0	0	0	0	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
7:0	—	The read values are undefined. The write value should be 0.	R/W
11:8	HIRDMON[3:0]	HIRD Value Monitor When set, indicates that the HIRD field value reflects the last received LPM token.	R
12	RWEMON	RWE Value Monitor When set, indicates that the RWE bit value reflects the last received LPM token.	R
15:13	—	The read values are undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

HIRDMON[3:0] bits (HIRD Value Monitor)

Access the HIRDMON[3:0] bits when monitoring the HIRD field value of the received LPM token. The bits reflect the HIRD field value of the last received LPM token.

RWEMON bit (RWE Value Monitor)

Access the RWEMON bit when monitoring the RWE field value of the received LPM token. The bits reflect the RWE field value of the last received LPM token.

37.2.47 HL1CTRL1 : Host L1 Control Register 1

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x148

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	L1STATUS[1:0]		L1REQ
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
0	L1REQ	L1 Transition Request Set this bit to 1 when requesting a transition to the L1 state. This bit is cleared to 0 by the hardware when the LPM transaction is complete.	R/W
2:1	L1STATUS[1:0]	L1 Request Completion Status Indicates the result of the LPM transaction made by the L1REQ bit: 0 0: ACK received 0 1: NYET received 1 0: STALL received 1 1: Transaction error	R
15:3	—	The read value is undefined. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

L1REQ bit (L1 Transition Request)

Set the L1REQ bit to 1 to transition to the L1 state. When the USBHS detects that this bit is 1, it starts the LPM transaction. The USBHS clears this bit to 0 through hardware on completion of the transaction.

L1STATUS[1:0] bits (L1 Request Completion Status)

The L1STATUS[1:0] bits indicate the result of the LPM transaction initiated by the L1REQ bit.

37.2.48 HL1CTRL2 : Host L1 Control Register 2

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x14A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BESL	—	—	L1RWE	HIRD[3:0]				—	—	—	—	L1ADDR[3:0]			
Value after reset:	0	x	x	0	0	0	0	0	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
3:0	L1ADDR[3:0]	LPM Token Device Address Specify the value to be set in the ADDR field of the LPM token	R/W
7:4	—	The read values are undefined. The write value should be 0.	R/W
11:8	HIRD[3:0]	LPM Token HIRD Specify the value to be set in the HIRD field of the LPM token	R/W
12	L1RWE	LPM Token L1 Remote Wake Enable Specify the value to be set in the RWE field of the LPM token	R/W
14:13	—	The read values are undefined. The write value should be 0.	R/W
15	BESL	BESL & Alternate HIRD Selects the K-State drive period on L1 Resume	R/W

Note: S-TYPE-3, P-TYPE-3

L1ADDR[3:0] bits (LPM Token Device Address)

The L1ADDR[3:0] bits specify the value to be set in the ADDR field of the LPM token that the USBHS transmits when the HL1CTRL1.L1REQ bit is set to 1.

HIRD[3:0] bits (LPM Token HIRD)

The HIRD[3:0] bits specify the value to be set in the HIRD field of the LPM token that the USBHS transmits when the HL1CTRL1.L1REQ bit is set to 1. Table 37.15 shows the relationship between the HIRD settings and the HIRD field values.

Table 37.15 Relationship between the HIRD bit settings and the HIRD field values (1 of 2)

HIRD[3:0] setting	When BESL = 0	When BESL = 1
0x0	50 μs (setting prohibited)	75 μs
0x1	125 μs	100 μs
0x2	200 μs	150 μs
0x3	275 μs	250 μs
0x4	350 μs	350 μs
0x5	425 μs	450 μs
0x6	500 μs	950 μs
0x7	575 μs	1950 μs
0x8	650 μs	2950 μs
0x9	725 μs	3950 μs
0xA	800 μs	4950 μs
0xB	875 μs	5950 μs
0xC	950 μs	6950 μs
0xD	1025 μs (setting prohibited)	7950 μs
0xE	1100 μs (setting prohibited)	8950 μs

Table 37.15 Relationship between the HIRD bit settings and the HIRD field values (2 of 2)

HIRD[3:0] setting	When BESL = 0	When BESL = 1
0xF	1175 µs (setting prohibited)	9950 µs

Note: The set value of the HIRD bit is used for the host K drive period on host resume and for the host K period on remote wakeup.

L1RWE bit (LPM Token L1 Remote Wake Enable)

The L1RWE bit specifies the value to be set in the RWE field of the LPM token that the USBHS transmits when the HL1CTRL1.L1REQ bit is set to 1.

The USBHS does not control detection of the remote wakeup signal in the L1 state with this bit. The remote wakeup signal is controlled by the DVSTCTR0.RWUPE bit, as with Suspend.

BESL bit (BESL & Alternate HIRD)

The BESL bit selects the K-state drive period on L1 Resume. For details, see the description of the HIRD bits.

37.2.49 DPUSR0R : Deep Software Standby USB Transceiver Control/Pin Monitor Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x160

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	DVBSTSHM	—	DOVCBHM	DOVCAHM	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	x	0	x	x	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
19:0	—	These bits are read as 0. The write value should be 0.	R
20	DOVCAHM	OVRCURA Input Flag Indicates OVRCURA input signal on the USBHS side	R
21	DOVCBHM	OVRCURB Input Flag Indicates OVRCURB input signal on the USBHS side	R
22	—	This bit is read as 0. The write value should be 0.	R
23	DVBSTSHM	VBUS Input Flag Indicates VBUS input signal on the USBHS side	R
31:24	—	These bits are read as 0. The write value should be 0.	R

Note: S-TYPE-3, P-TYPE-3
Note: This register is accessed by the clock of PCLKA/64 to reduce current consumption in Deep Software Standby mode 1.

37.2.50 DPUSR1R : Deep Software Standby USB Suspend/Resume Interrupt Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x164

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	DVBSTSH	—	DOVCBH	DOVCAH	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	DVBSTSHE	—	DOVCBHE	DOVCAHE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	DOVCAHE	OVRCURA Interrupt Enable Clear 0: Disable recovery from Deep Software Standby mode 1 1: Enable recovery from Deep Software Standby mode 1	R/W
5	DOVCBHE	OVRCURB Interrupt Enable Clear 0: Disable recovery from Deep Software Standby mode 1 1: Enable recovery from Deep Software Standby mode 1	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	DVBSTSHE	VBUS Interrupt Enable/Clear 0: Disable recovery from Deep Software Standby mode 1 1: Enable recovery from Deep Software Standby mode 1	R/W
19:8	—	These bits are read as 0. The write value should be 0.	R/W
20	DOVCAH	OVRCURA Interrupt Source Return Status Flag 0: System has not recovered from Deep Software Standby mode 1 1: System recovered from Deep Software Standby mode 1	R
21	DOVCBH	OVRCURB Interrupt Source Return Status Flag 0: System has not recovered from Deep Software Standby mode 1 1: System recovered from Deep Software Standby mode 1	R
22	—	This bit is read as 0. The write value should be 0.	R/W
23	DVBSTSH	VBUS Interrupt Source Return Status Flag 0: System has not recovered from Deep Software Standby mode 1 1: System recovered from Deep Software Standby mode 1	R
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This register is accessed by the clock of PCLKA/64 to reduce current consumption in Deep Software Standby mode 1.

37.2.51 DPUSR2R : Deep Software Standby USB Suspend/Resume Interrupt Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x168

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	DPINTE	DMINTE	—	—	DMVAL	DPVAL	—	—	DPINT	DMINT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DMINT	Indication of Return from DM Interrupt Source 0: System has not recovered from Deep Software Standby mode 1 1: System recovered from Deep Software Standby mode 1	R
1	DPINT	Indication of Return from DP Interrupt Source 0: System has not recovered from Deep Software Standby mode 1 1: System recovered from Deep Software Standby mode 1	R
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	DPVAL	DP Input Indicates DP input signal on the USBHS side	R
5	DMVAL	DM Input Indicates DM input signal on the USBHS side	R
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	DMINTE	DM Interrupt Enable Clear 0: Disable recovery from Deep Software Standby mode 1 1: Enable recovery from Deep Software Standby mode 1	R/W
9	DPINTE	DP Interrupt Enable Clear 0: Disable recovery from Deep Software Standby mode 1 1: Enable recovery from Deep Software Standby mode 1	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This register is accessed by the clock of PCLKA/64 to reduce current consumption in Deep Software Standby mode 1.

37.2.52 DPUSRCR : Deep Software Standby USB Suspend/Resume Command Register

Base address: USBHS = 0x4035_1000
USBHS_NS = 0x5035_1000

Offset address: 0x16A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FIXPHYPD	FIXPHY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FIXPHY	USB Transceiver Control Fix 0: Normal mode 1: Invoke/recover from Deep Software Standby mode 1	R/W
1	FIXPHYPD	USB Transceiver Control Fix for PLL 0: Normal mode 1: Invoke/recover from Deep Software Standby mode 1	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: This register is accessed by the clock of PCLKA/64 to reduce current consumption in Deep Software Standby mode 1.

37.3 Operation

37.3.1 System Control

This section describes register settings required for initializing the USBHS and controlling power consumption.

37.3.1.1 Setting Data to the USBHS Registers

Setting the SYSCFG.USBE bit to 1 after starting the PHY clock supply enables and starts USBHS operation. For information on how to supply the PHY clock, see section 37.3.3. Supplying the Clock.

37.3.1.2 Selecting the Controller Function

The USBHS can operate as a host or device controller.

Use the SYSCFG.DCFM bit to select one of these USBHS functions. The DCFM bit must be changed in the initial settings immediately after a reset or in the D+ pull-up-disabled state (SYSCFG.DPRPU bit = 0) and D+ and D- pull-down- disabled state (SYSCFG.DRPD bit = 0).

37.3.2 Controlling the USB Data Bus Using Resisters

The USBHS provides pull-up and pull-down resistors for the D+ and D- lines. Pull these lines up or down by setting the SYSCFG.DPRPU and DRPD bits.

In device controller mode, confirm that connection to the USB host is made, and then set the SYSCFG.DPRPU bit to 1 and pull up the D+ line (in full-speed communication).

When the SYSCFG.DPRPU bit is set to 0 during communication with a PC, the USBHS disables the pull-up resistor for the USB data line, thereby notifying the USB host of disconnection.

Table 37.16 shows the settings for controlling the resistors for the USB data bus. Control the USB data bus appropriately for your system using the DRPD and DPRPU bit settings.

Table 37.16 Control settings for the USB data bus resistors (excluding OTG operation)

SYSCFG register settings		USB data bus control		
DRPD bit	**DPRPU bit**	**D-Line**	**D+Line**	**Function**
0	0	Open	Open	When resistors not used
0	1	Open	Pull-Up	When operating as a device controller at full-speed
1	0	Pull-Down	Pull-Down	When operating as a host controller
1	1	—	—	Setting prohibited except during OTG operation

37.3.3 Supplying the Clock

Table 37.17 shows the two input clocks required for the USBHS.

Table 37.17 Input clocks

Input clock name	Description
PCLKA	Peripheral module clock A input. There is no constraint on the frequency of the PCLKA input.
PHY clock	PHY clock generated from external input or internal supply ● External input: The clock is generated by the USB-PHY internal PLL based on a 12-MHz, 20-MHz, 24-MHz, or 48-MHz clock supplied to the EXTAL pin from outside the MCU. For the external clock specifications, especially the jitter characteristics, strictly follow the specifications of ±50 ppm. ● Internal supply: The clock is generated by supplying 48 MHz and 60 MHz to the USB-PHY module and selecting CL-only mode (PHYSET.HSEB). High-speed operation is not supported with this mode.

Figure 37.2 illustrates the PHY clock settings.

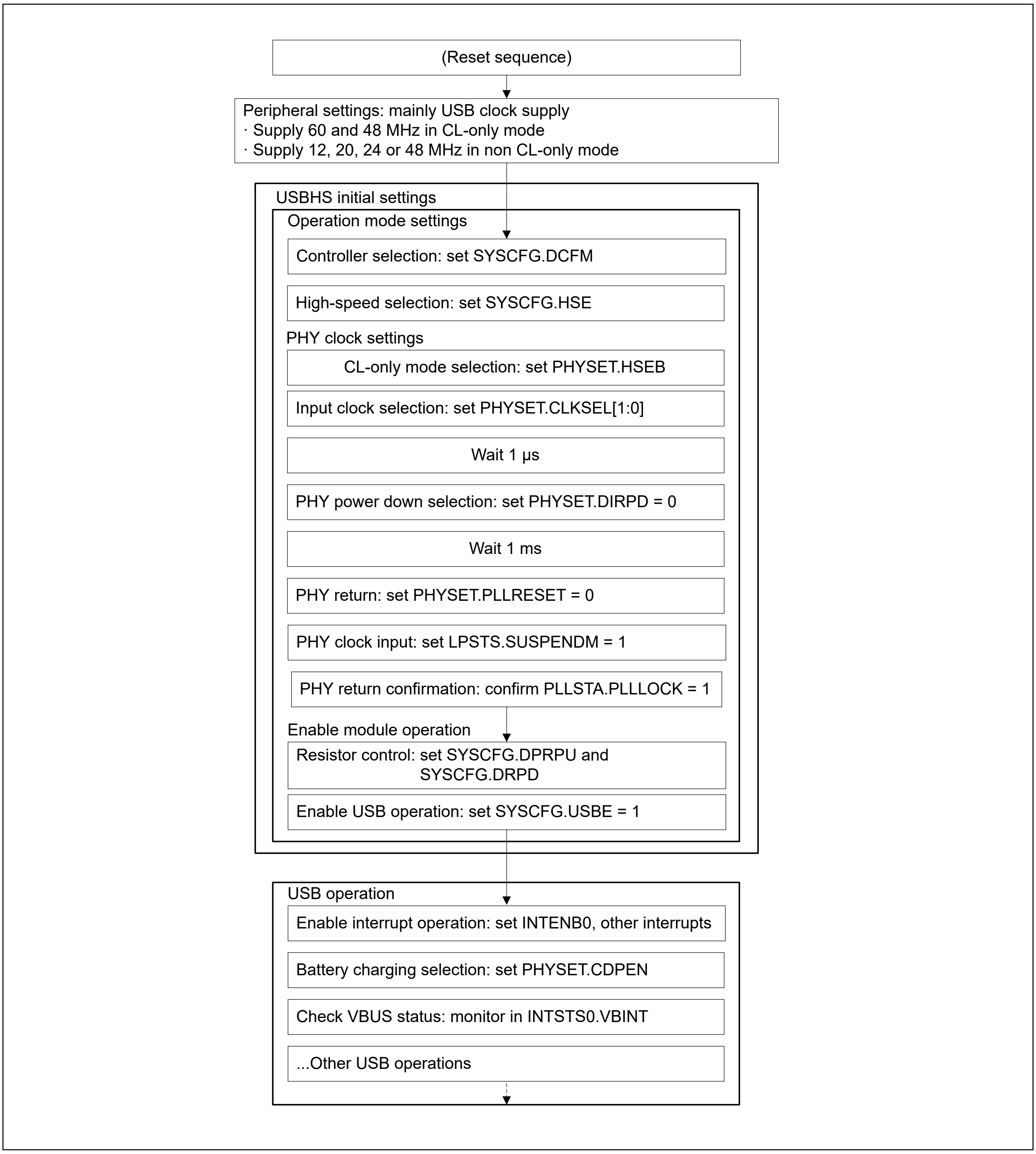


Figure 37.2 PHY clock settings

37.3.4 Constraints on Stopping the Clock

PCLKA and PHY clock can be stopped during disconnection or suspension. However, to stop any of these clocks while the USB is suspended in device controller mode, the stopped clock must be resupplied using the resume interrupt. The PHY clock must be resupplied within 5.5 ms after the resume interrupt is generated.

37.3.5 Interrupts

Table 37.18 lists the interrupt sources in the USBHS. When an interrupt generation condition is satisfied and the interrupt output is enabled using the associated interrupt enable register, the USBHS issues a USBHS interrupt request to the Interrupt Controller Unit (ICU) and a USBHS interrupt is generated.

Table 37.18 Interrupt sources (1 of 2)

Flag to be set to 1	Interrupt name	Interrupt source	Applicable controller function	Status flag
VBINT	VBUS interrupt	• A change in the state of the USB_VBUS input pin is detected (low to high or high to low)	Host or function*1	INTSTS0.VBSTS
RESM	Resume interrupt	• A change in the state of the USB bus is detected in the Suspend state (J-state to K-state or J-state to SE0)	Function	—
SOFR	Frame number update interrupt	In host controller mode: • An SOF packet with a different frame number is transmitted In device controller mode: • When SOFRM is 0: An SOF packet with a different frame number is received • When SOFRM is 1: Failed to receive an SOF packet with the μ frame number 0 because the packet is corrupted.	Host or function	—
DVST	Device state transition interrupt	• A device state transition is detected because of one of the following: – USB bus reset is detected – Suspend state is detected – SET_ADDRESS request is received – SET_CONFIGURATION request is received	Function	PL1CTRL.DVSQ[3:0]
CTRT	Control transfer stage transition interrupt	• A control transfer stage transition is detected because of one of the following: – Setup stage completed – Control write transfer status stage transition occurred – Control read transfer status stage transition occurred – Control transfer completed – Control transfer sequence error occurred	Function	INTSTS0.CTSQ[2:0]
BEMP	Buffer empty interrupt	• The buffer is empty after all FIFO buffer data is transmitted • A packet larger than the maximum packet size is received	Host or function	BEMPSTS.PIPEBEMP
NRDY	Buffer not ready interrupt	In host controller mode: • A STALL response is received from the peripheral device in response to the issued token • The response from the peripheral device in response to the issued token is not received successfully (no response three times consecutively or packet reception error three times consecutively) • An overrun or underrun error occurred during isochronous transfer In device controller mode: • NAK is returned for an IN or OUT token while the PID [1:0] bits were set to 01b (BUF) • A CRC error or bit stuffing error occurred during data reception in isochronous transfer • An interval error occurred during data reception in isochronous transfer	Host or function	NRDYSTS.PIPENRDY
BRDY	Buffer ready interrupt	• The buffer is ready (read or write state)	Host or function	BRDYSTS.PIPEBRDY
OVRCR	Overcurrent input change interrupt	• USBHS_OVRCURA, USBHS_OVRCURA-DS, USBHS_OVRCURB or USBHS_OVRCURAB-DS pin state change is detected (low to high or high to low)	Host or function	SYSSTS0.OVCMON [1:0]

Table 37.18 Interrupt sources (2 of 2)

Flag to be set to 1	Interrupt name	Interrupt source	Applicable controller function	Status flag
BCHG	Bus change interrupt	• USB bus state change is detected	Host	—
DTCH	Device disconnect detection interrupt	• Peripheral device disconnect is detected	Host	—
ATTCH	Device connect detection interrupt	• J-state or K-state is detected on the USB bus for 2.5 µs continuously This interrupt can be used to check whether peripheral devices are connected.	Host	—
EOFERR	EOF error detection interrupt	• An EOF error is detected for a peripheral device	Host	—
SACK	Setup normal interrupt	• A setup transaction normal response (ACK) is received	Host	—
SIGN	Setup error interrupt	• A setup transaction error (no response or ACK packet corruption) is detected three consecutive times	Host	—
PDDETINT	PDDETSTS change detect interrupt	• PDDET pin change is detected	Host or function	BCCTRL.PDDETSTS
LPMEND	LPM transaction end interrupt	• LPM transaction is complete	Host	PL1CTRL.DVSQ[3:0]
L1RSMEND	L1 resume end interrupt	• Resume (from L1 state) processing is complete	Host	PL1CTRL.DVSQ[3:0]

Note 1. Although this interrupt can be generated in host controller mode, it is not usually used in this mode.

37.3.5.1 Selecting the USBHS Interrupt Detection Method

Table 37.19 shows operations for an USBHS interrupt output from the USBHS. In case two or more interrupt sources are generated, the USBHS interrupt output method can be set in the SOFCFG.INTL bit. Set the USBHS interrupt output operation based on your system.

Table 37.19 USBHS interrupt operation

USBHS interrupt output (INTL setting)	When one interrupt source is generated	When two or more interrupt sources are generated
Edge detection (SOFCFG.INTL bit = 0)	Low level output until the source is cleared	When one source is cleared, the USBHS interrupt is negated for 32 clocks at 48 MHz (high pulse output)
Level detection (SOFCFG.INTL bit = 1)	Low level output until the source is cleared	Low level output until all sources are cleared

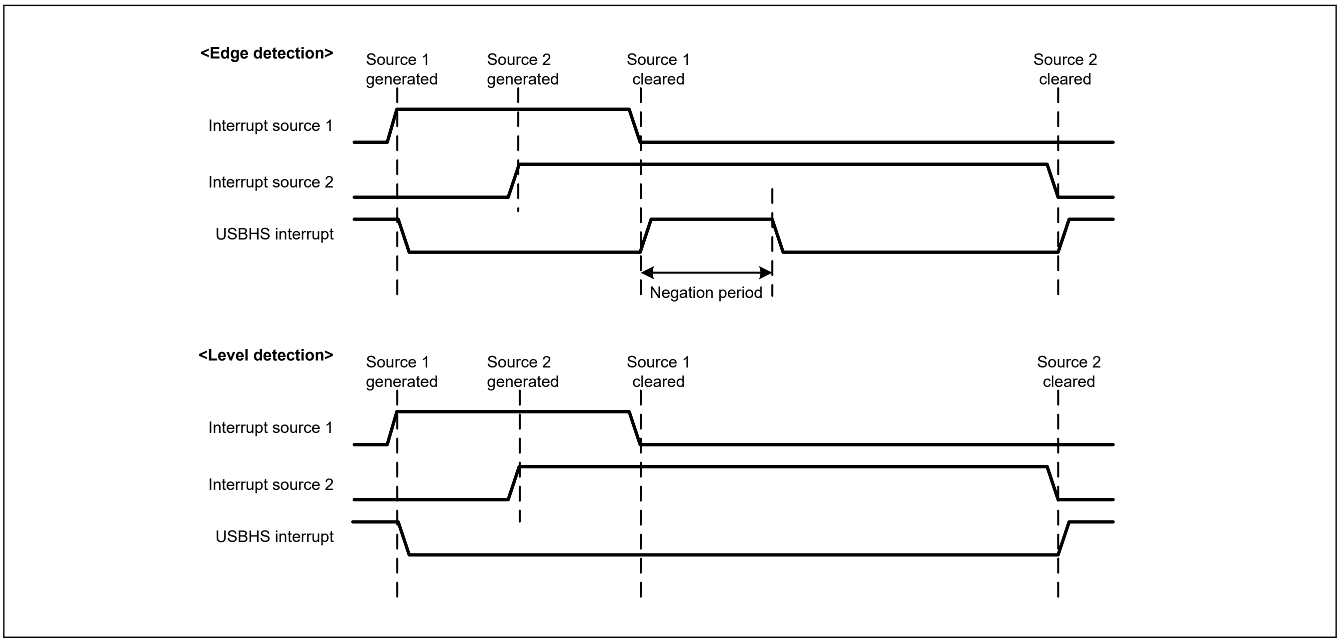


Figure 37.3 USBHS interrupt operation

Figure 37.4 shows an interrupt association chart of the USBHS.

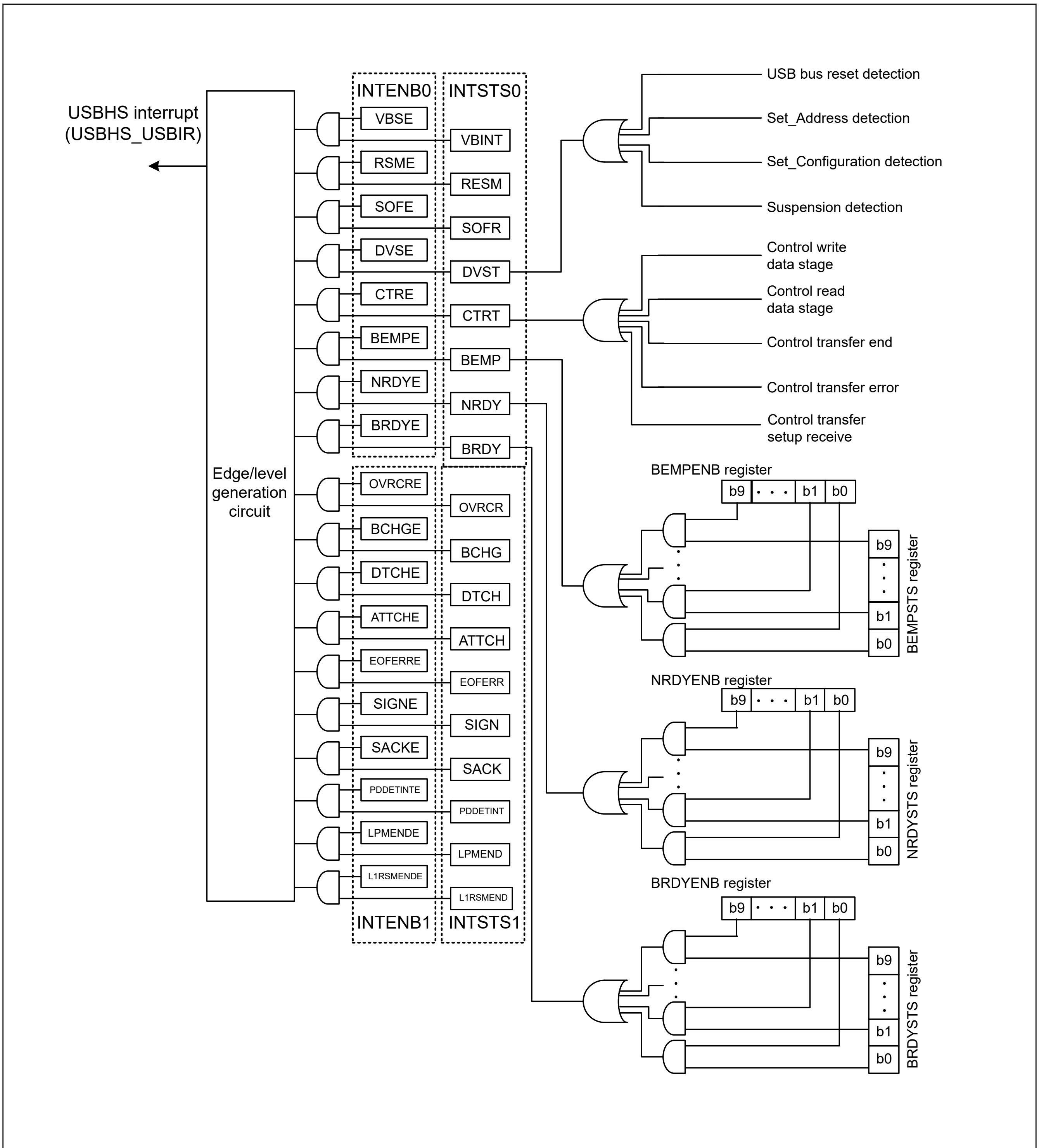


Figure 37.4 USBHS interrupt-related circuits

Table 37.20 shows the interrupts generated by the USBHS.

Table 37.20 USBHS interrupts (1 of 2)

Interrupt name	Interrupt status flag	DTC activation	DMAC activation
USBHS_D0FIFO	DMA transfer request 0	Possible	Possible
USBHS_D1FIFO	DMA transfer request 1	Possible	Possible

Table 37.20 USBHS interrupts (2 of 2)

Interrupt name	Interrupt status flag	DTC activation	DMAC activation
USBHS_USBIR	VBUS interrupt, resume interrupt, frame number update interrupt, device state transition interrupt, control transfer stage transition interrupt, buffer empty interrupt, buffer not ready interrupt, buffer ready interrupt, overcurrent interrupt, bus change interrupt, device disconnect detection interrupt, device connect detection interrupt, EOF error detection interrupt, normal setup operation interrupt, setup error interrupt, PDDETSTS change detection interrupt, LPM transaction end interrupt, and L1 resume end interrupt	Not possible	Not possible

37.3.6 Interrupt Descriptions

37.3.6.1 BRDY Interrupt

The BRDY interrupt is generated in both host and device controller modes. This section describes the conditions in which the USBHS sets the associated bit in BRDYSTS to 1. Under these conditions, the USBHS generates a BRDY interrupt if the software sets the bit in BRDYENB associated with the given pipe to 1 and INTENB0.BRDYE bit to 1.

The conditions for generating and clearing the BRDY interrupt depend on the SOFCFG.BRDYM and PIPECFG.BFRE settings for each pipe as follows:

(1) When SOFCFG.BRDYM = 0 and PIPECFG.BFRE = 0

With these settings, the BRDY interrupt indicates that the FIFO port is accessible.

On any of the following conditions, the USBHS generates an internal BRDY interrupt request trigger and sets the BRDYSTS.PIPEBRDY flag associated with the pipe to 1.

For transmitting pipes

- When the DIR bit is changed from 0 to 1 by software
- When writing by the CPU to the FIFO buffer is disabled for a pipe (when the BSTS flag is read as 0) and the USBHS has completed packet transmission. In continuous transfer, a BRDY interrupt is generated on completion of the transmission of data from one FIFO buffer.
- When one FIFO buffer is empty on completion of writing data to the other FIFO buffer in double buffer mode
- No request trigger is generated until completion of writing data to the currently-written FIFO buffer even if transmission to the other FIFO buffer is complete
- When the hardware flushes the buffer of the pipe for isochronous transfers
- When 1 is written to the PIPEnCTR.ACLRM bit, which causes the FIFO buffer to transition from the write-disabled to write-enabled state.

No request trigger is generated for the DCP, that is, during data transmission for control transfers.

For receiving pipes

- When packet reception is successfully complete, enabling the FIFO buffer to be read while read-access from the CPU to the FIFO buffer for the given pipe is disabled (when the BSTS flag is read as 0). No request trigger is generated for transactions in which DATA-PID mismatch has occurred. In continuous transmission or reception mode, the request trigger is not generated when the data is of the specified maximum packet size and the buffer has available space. When a short packet is received, the request trigger is generated even if the FIFO buffer has available space. When the transaction counter is used, the request trigger is generated on receiving the specified number of packets. In this case, the request trigger is generated even if the FIFO buffer has available space.
- When one FIFO buffer is read-enabled on completion of reading data from the other FIFO buffer in double buffer mode. No request trigger is generated until completion of reading data from the currently-read FIFO buffer, even if reception by the other FIFO buffer is complete.
  In device controller mode, the BRDY interrupt is not generated in the status stage of control transfers. The PIPEBRDY interrupt status of the selected pipe can be set to 0 by writing 0 to the associated PIPEBRDY flag through the software. In this case, 1s must be written to the associated bits for the other pipes. Clear the BRDY status before accessing the FIFO buffer.

(2) When SOFCFG.BRDYM = 0 and PIPECFG.BFRE = 1

With these settings, the USBHS generates a BRDY interrupt on completion of reading all data for a single transfer using the receiving pipe, and sets the bit in BRDYSTS associated with the pipe to 1.

On any of the following conditions, the USBHS determines that the last data for a single transfer was received:

- When a short packet including a zero-length packet is received
- When the PIPEnTRN register is used and the number of packets specified in the PIPEnTRN.TRNCNT[15:0] bits are completely received

When the data is completely read after any of these conditions is satisfied, the USBHS determines that all data for a single transfer is completely read.

When a zero-length packet is received while the FIFO buffer is empty, the USBHS determines that all data for a single transfer is completely read when the FRDY flag in the FIFO port control register is 1 and the DTLN[11:0] flags are 0. In this case, to start the next transfer, write 1 to the BCLR bit in the associated port control register through software. With these settings, the USBHS does not detect a BRDY interrupt for the transmitting pipe.

The PIPEBRDY interrupt status of a pipe can be set to 0 by writing 0 to the associated BRDYSTS.PIPEBRDY flag through the software. In this case, 1s must be written to the PIPEBRDY bits for the other pipes.

In this mode, do not change the PIPECFG.BFRE bit setting until all data for a single transfer is processed. When it is necessary to change the PIPECFG.BFRE bit before completion of processing, all FIFO buffers for the pipe must be cleared using the PIPEnCTR.ACLRM bit.

(3) When SOFCFG.BRDYM = 1 and PIPECFG.BFRE = 0

With these settings, the BRDYSTS.PIPEBRDY flag values are linked to the BSTS flag setting for each pipe. In other words, the BRDY interrupt status bits are set to 1 or 0 by the USBHS depending on the FIFO buffer status.

For transmitting pipes

The BRDY interrupt status bits are set to 1 when the FIFO port is ready for write access, and are set to 0 when it is not ready. The BRDY interrupt is not generated for the DCP in the transmitting direction even when it is ready for write access.

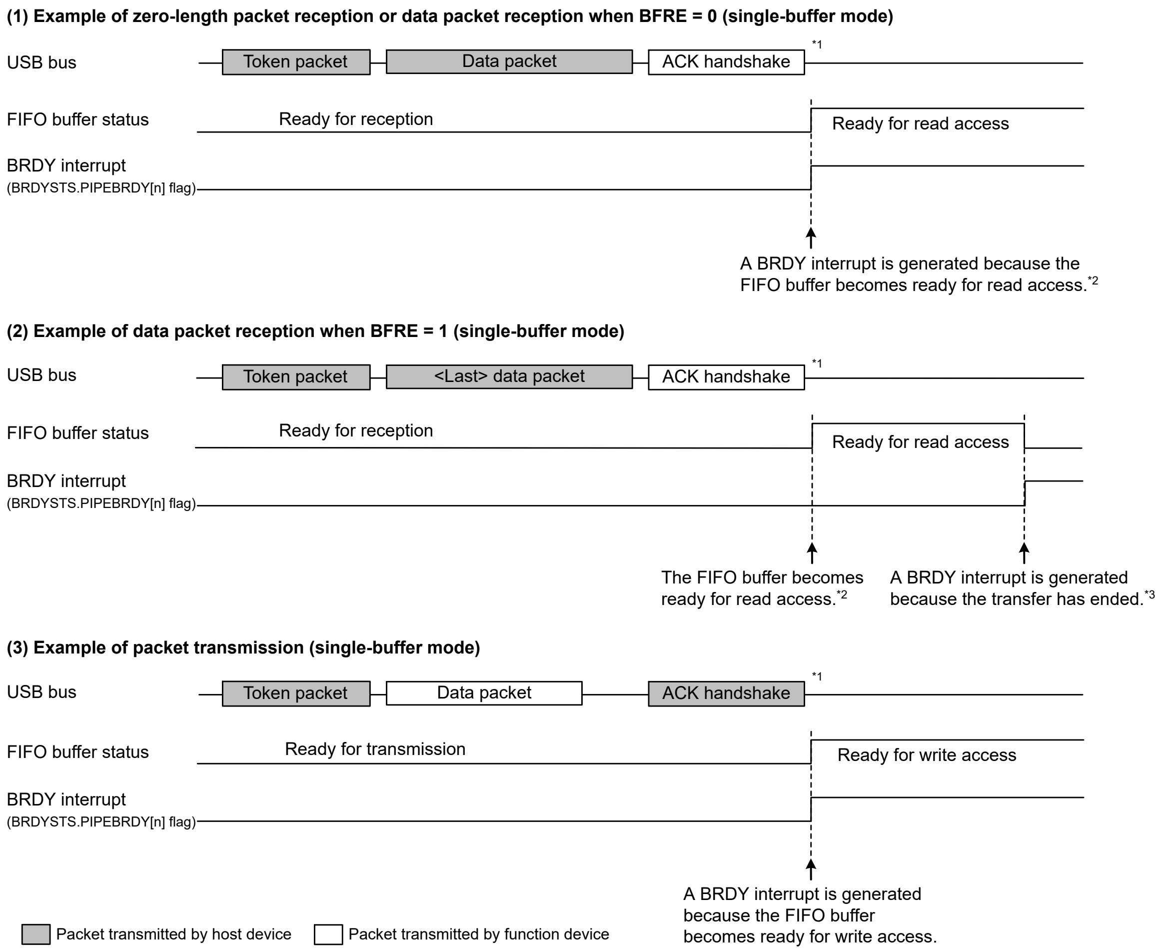
For receiving pipes

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for read access, and are set to 0 when all data is read (not ready for read access).

When a zero-length packet is received while the FIFO buffer is empty, the associated bit is set to 1 and the BRDY interrupt is continuously generated until the software writes 1 to BCLR. With this setting, the PIPEBRDY flag cannot be set to 0 by software.

When the SOFCFG.BRDYM bit is set to 1, set the PIPECFG.BFRE bit for all pipes to 0, and the SOFCFG.INTL bit to 1 for level detection.

Figure 37.5 shows the timing of BRDY interrupt generation.



Note 1. The ACK handshake is not used in isochronous transfers.
Note 2. The FIFO buffer becomes ready for read access under the following condition:
When a packet is received while no unread data remains in the FIFO buffer in the CPU.
Note 3. A transfer ends under either of the following conditions:
1. When a short packet including a zero-length packet is received
2. When the number of packets specified in the transaction counter are received

Figure 37.5 Timing of BRDY interrupt generation

The condition for clearing the INTSTS0.BRDY flag depends on the SOFCFG.BRDYM bit setting value, as shown in Table 37.21

Table 37.21 Conditions for clearing the BRDY flag

BRDYM bit	Condition for clearing BRDY flag
0	The USBHS clears the BRDY flag to 0 when all bits in BRDYSTS are set to 0 by software
1	The USBHS clears the BRDY flag to 0 when the BSTS flags for all pipes have cleared to 0

37.3.6.2 NRDY Interrupt

On generating an internal NRDY interrupt request for the pipe whose PID[1:0] bits are set to 01b (BUF response) by software, the USBHS sets the associated NRDYSTS.PIPENRDY flag to 1. If the associated bit in NRDYENB is set to 1 by software, the USBHS sets the INTSTS0.NRDY flag to 1 and generates a USBHS interrupt.

This section describes the conditions in which the USBHS generates the internal NRDY interrupt request for a given pipe.

The internal NRDY interrupt request is not generated during setup transaction execution in host controller mode. During setup transactions in host controller mode, the SACK or SIGN interrupt is detected.

The internal NRDY interrupt request is not generated during status stage execution of the control transfer in device controller mode

(1) In host controller mode when no split transactions occur in the connection

For transmitting pipes

On any of the following conditions, the USBHS detects an NRDY interrupt:

- For isochronous transfer pipes, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer. In this case, the USBHS transmits a zero-length packet following the OUT token and sets the associated NRDYSTS.PIPENRDY flag and the FRMNUM.OVRN flag to 1.
- During communications other than setup transactions on pipes not used for isochronous transfers, when any combination of the following two conditions occurs three consecutive times:
  - No response is returned from the peripheral device (when timeout is detected before detection of the handshake packet from the peripheral device
  - An error is detected in the packet from the peripheral device. In this case, the USBHS sets the associated PIPENRDY flag to 1 and changes the PID[1:0] setting for the associated pipe to 00b (NAK response)
- During communications other than setup transactions, when the STALL handshake is received from the peripheral device (includes STALL for both OUT and PING). In this case, the USBHS sets the associated PIPENRDY flag to 1 and changes the PID[1:0] setting for the associated pipe to 11b (STALL response).

For receiving pipes

- For isochronous transfer pipes, when the time to issue an IN token comes but there is no space available in the FIFO buffer. In this case, the USBHS discards the received data for the IN token and sets the associated PIPENRDY flag and the OVRN flag to 1. When a packet error is detected in the received data for the IN token, the USBHS also sets the FRMNUM.CRCE flag to 1.
- For non-isochronous transfer pipes, when any combination of the following two cases occur three consecutive times:
  - No response is returned from the peripheral device for the IN token issued by the USBHS (when timeout is detected before detection of the DATA packet from the peripheral device)
  - An error is detected in the packet from the peripheral device. In this case, the USBHS sets the associated PIPENRDY flag to 1 and changes the associated PID[1:0] setting for the pipe to 00b (NAK response).
- For isochronous transfer pipes, when no response is returned from the peripheral device for the IN token (when timeout is detected before detection of the DATA packet from the peripheral device) or an error is detected in the packet from the peripheral device. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for each pipe to 1. The PID[1:0] setting for the pipe is not changed.
- For isochronous transfer pipes, when a CRC error or a bit stuffing error is detected in the received data packet. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for each pipe and the CRCE flag to 1.
- When the STALL handshake is received. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for each pipe to 1 and changes the PID[1:0] setting for the associated pipe to STALL.

(2) In host controller mode when split transactions occur in the connection

For transmitting pipes

On any of the following conditions, the USBHS detects an NRDY interrupt:

- For isochronous transfer pipes, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer. In this case, the USBHS sets the associated RDYSTS.PIPENRDY flag for the given pipes to 1 on issuing a start-split transaction and sets the FRMNUM.OVRN flag to 1. The USBHS also transmits a zero-length packet following the OUT token.
- For non-isochronous transfer pipes, when any combination of the following two cases occurs three consecutive times:
  - No response is returned from the hub for start-split and complete-split transactions (when timeout is detected before detection of the handshake packet from the hub)

  - An error is detected in the packet from the hub. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for the pipe to 1 and changes the associated PID[1:0] setting for the pipe to 00b (NAK response). When an NRDY interrupt is detected on complete-split issuance, the USBHS clears the CSSTS flag to 0.
- When a STALL handshake is received for the complete-split transaction. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for the pipe to 1, changes the associated PID[1:0] setting for the pipe to 11b (STALL response), and clears the CSSTS flag to 0. An interrupt is not detected during setup transaction.

For receiving pipes

- For isochronous transfer pipes, when the time to issue an IN token comes but there is no space available in the FIFO buffer. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for the given pipe and the FRMNUM.OVRN flag to 1 on start-split issuance. The USBHS discards the received data for the IN token.
- During bulk-pipe transfers or transfers other than setup transactions with the DCP, when any combination of the following two cases occurs three consecutive times:
  - No response is returned from the hub for the IN token the USBHS issued on issuance of the start-split or complete-split transactions (when timeout is detected before detection of the data packet from the hub)
  - An error is detected in the packet from the hub. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for the pipe to 1 and changes the associated PID[1:0] setting for the pipe to 00b (NAK response). When this condition occurs during complete-split, the USBHS clears the CSSTS flag to 0.
- During a complete-split transaction for isochronous transfer or interrupt transfer pipes, when any combination of the following two cases occurs three consecutive times:
  - No response is returned from the hub for the IN token issued by the USBHS (when a timeout is detected before detection of the DATA packet from the hub)
  - An error is detected in the packet from the hub. On generating this condition for an interrupt transfer pipe, the USBHS sets the associated NRDYSTS.PIPENRDY flag to 1, changes the associated PID[1:0] setting for the pipe to 00b (NAK response), and clears the CSSTS flag to 0. On generating this condition for the pipe for isochronous transfers, the USBHS sets the associated NRDYSTS.PIPENRDY flag for the pipe to 1, CRCE flag to 1, and clears the CSSTS bit to 0. It does not change the PID[1:0] setting.
- During a complete-split transaction, when the STALL handshake is received for a non-isochronous transfer pipe. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for the pipe to 1, changes the associated PID[1:0] setting for the pipe to 11b (STALL response), and clears the CSSTS flag to 0.
- During a complete-split transaction, when the NYET handshake is received for an isochronous transfer or interrupt transfer pipe for the microframe number = 4. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for each pipe to 1 and the CRCE flag to 1, and clears the CSSTS flag to 0. It does not change the PID[1:0] setting.

(3) In device controller mode

For transmitting pipes

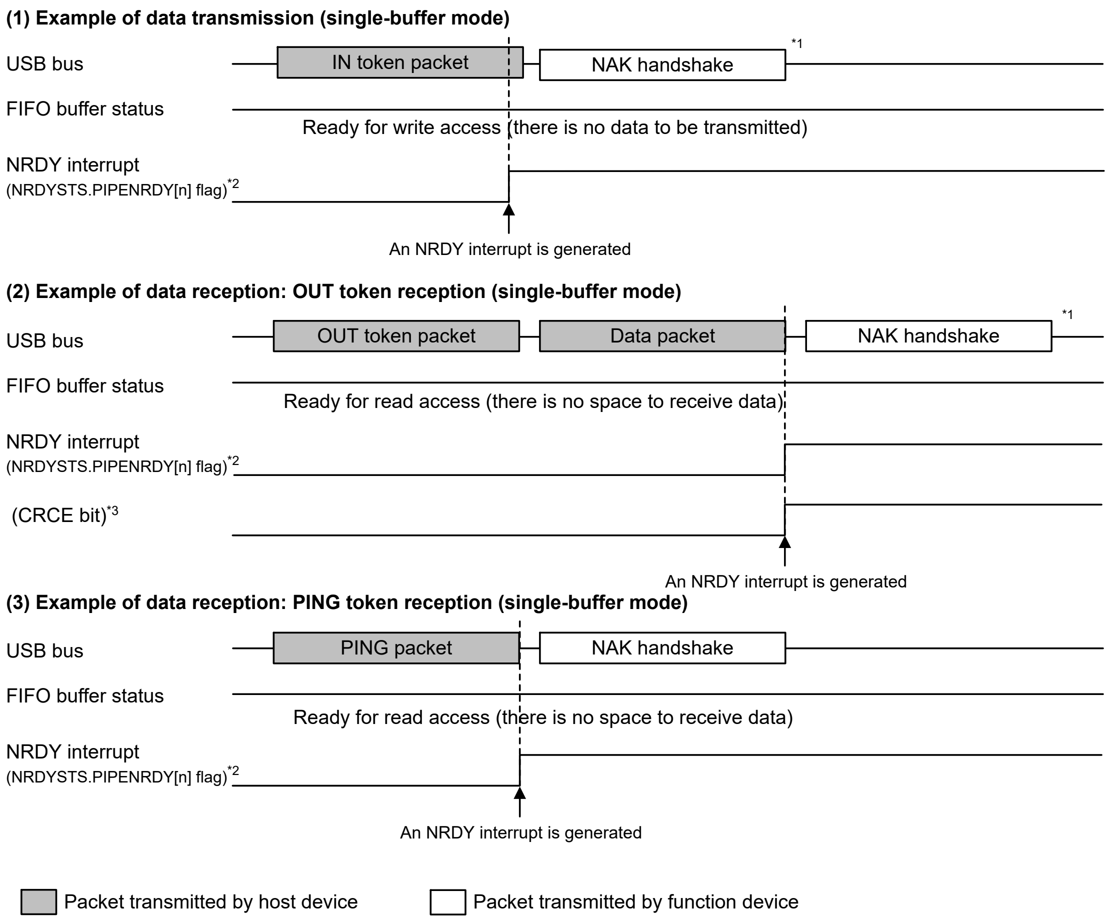
- When an IN token is received while there is no data to be transmitted in the FIFO buffer. In this case, the USBHS generates an NRDY interrupt request on reception of the IN token and sets the NRDYSTS.PIPENRDY flag to 1. For an isochronous transfer pipe in which an interrupt is generated, the USBHS transmits a zero-length packet and sets the FRMNUM.OVRN flag to 1.

For receiving pipes

- When an OUT token is received but there is no space available in the FIFO buffer. For an isochronous transfer pipe in which an interrupt is generated, the USBHS generates an NRDY interrupt request on reception of the OUT token and sets the NRDYSTS.PIPENRDY flag to 1 and the FRMNUM.OVRN flag to 1. For a non-isochronous transfer pipe in which an interrupt is generated, the USBHS generates an NRDY interrupt request when an NAK handshake is transferred after the data following the OUT token is received, and sets the NRDYSTS.PIPENRDY flag to 1. The NRDY interrupt request is not generated during retransmission because of a DATA-PID mismatch. In addition, the NRDY interrupt request is not generated if an error occurs in the DATA packet.
- On receiving a PING token when there is no space available in the FIFO buffer. The USBHS generates an NRDY interrupt request on reception of the PING token, setting the NRDYSTS.PIPENRDY flag to 1.

- For isochronous transfer pipes, when a token is not received successfully within an interval frame. In this case, the USBHS generates an NRDY interrupt request when the SOF is received, and sets the NRDYSTS.PIPENRDY flag to 1.

Figure 37.6 shows the timing of NRDY interrupt generation in device controller mode.



Note 1. The handshake is not used in isochronous transfers.
Note 2. The PIPENRDY[n] flag is changed to 1 only when PIPEnCTR.PID[1:0] bits are set to 01b (BUF response).
Note 3. The CRCE and OVRN flags change only while the target pipe is set to isochronous transfers.

Figure 37.6 Timing of NRDY interrupt generation in device controller mode

37.3.6.3 BEMP Interrupt

On detecting a BEMP interrupt for the pipe whose PID[1:0] bits in the pipe control register are set to 01b (BUF response) by software, the USBHS sets the associated BEMPSTS.PIPEBEMP flag to 1. If the associated BEMPENB bit is set to 1 by software, the USBHS sets the INTSTS0.BEMP flag to 1 and generates a USB interrupt. This section describes the conditions in which the USBHS generates an internal BEMP interrupt request.

(1) For transmitting pipes

When the FIFO buffer of the associated pipe is empty on completion of transmission, including zero-length packet transmission, and in single buffer mode, an internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for a non-DCP pipe. The internal BEMP interrupt request is not generated in any of the following conditions:

- When the CPU or DMAC/DTC has already started writing data to the FIFO buffer of the CPU on completion of transmitting data from one FIFO buffer in double buffer mode

- When the buffer is cleared (emptied) by setting 1 to the PIPEnCTR.ACLRM or the BCLR bit in the port control register
- When an IN transfer (zero-length packet transmission) is performed during the control transfer status stage in device controller mode

(2) For receiving pipes

When a successfully-received data packet size exceeds the specified maximum packet size. In this case, the USBHS generates a BEMP interrupt request, sets the associated BEMPSTS.PIPEBEMP flag to 1, discards the received data, and changes the associated PID[1:0] setting for the pipe to STALL (11b). The USBHS returns no response in host controller mode, and returns STALL response in device controller mode.

The internal BEMP interrupt request is not generated in any of the following conditions:

- When a CRC error or a bit stuffing error is detected in the received data
- When a setup transaction is being performed:
  - Writing 0 to the BEMPSTS.PIPEBEMP flag clears the status
  - Writing 1 to the BEMPSTS.PIPEBEMP flag has no effect

Figure 37.7 shows the timing of BEMP interrupt generation in device controller mode.

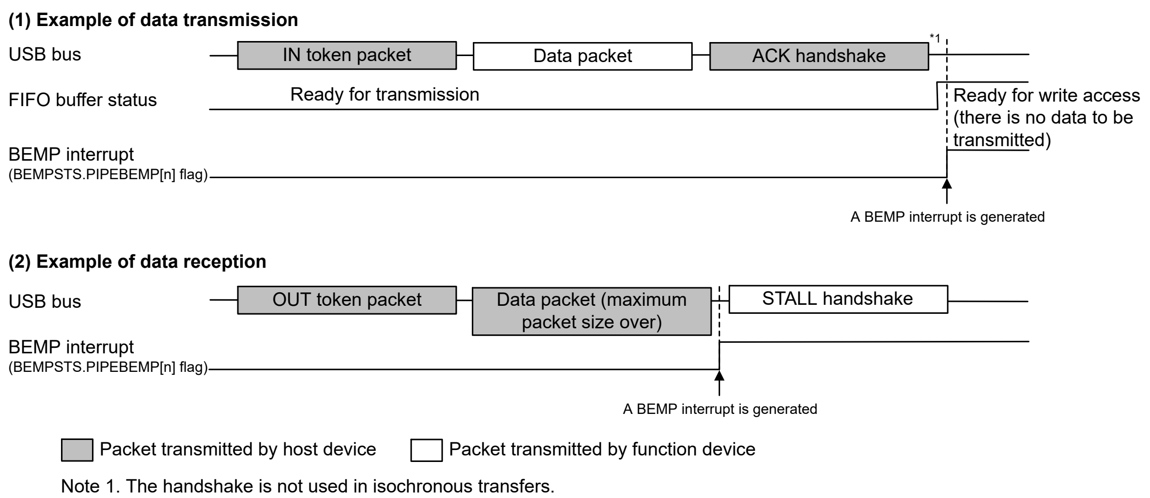


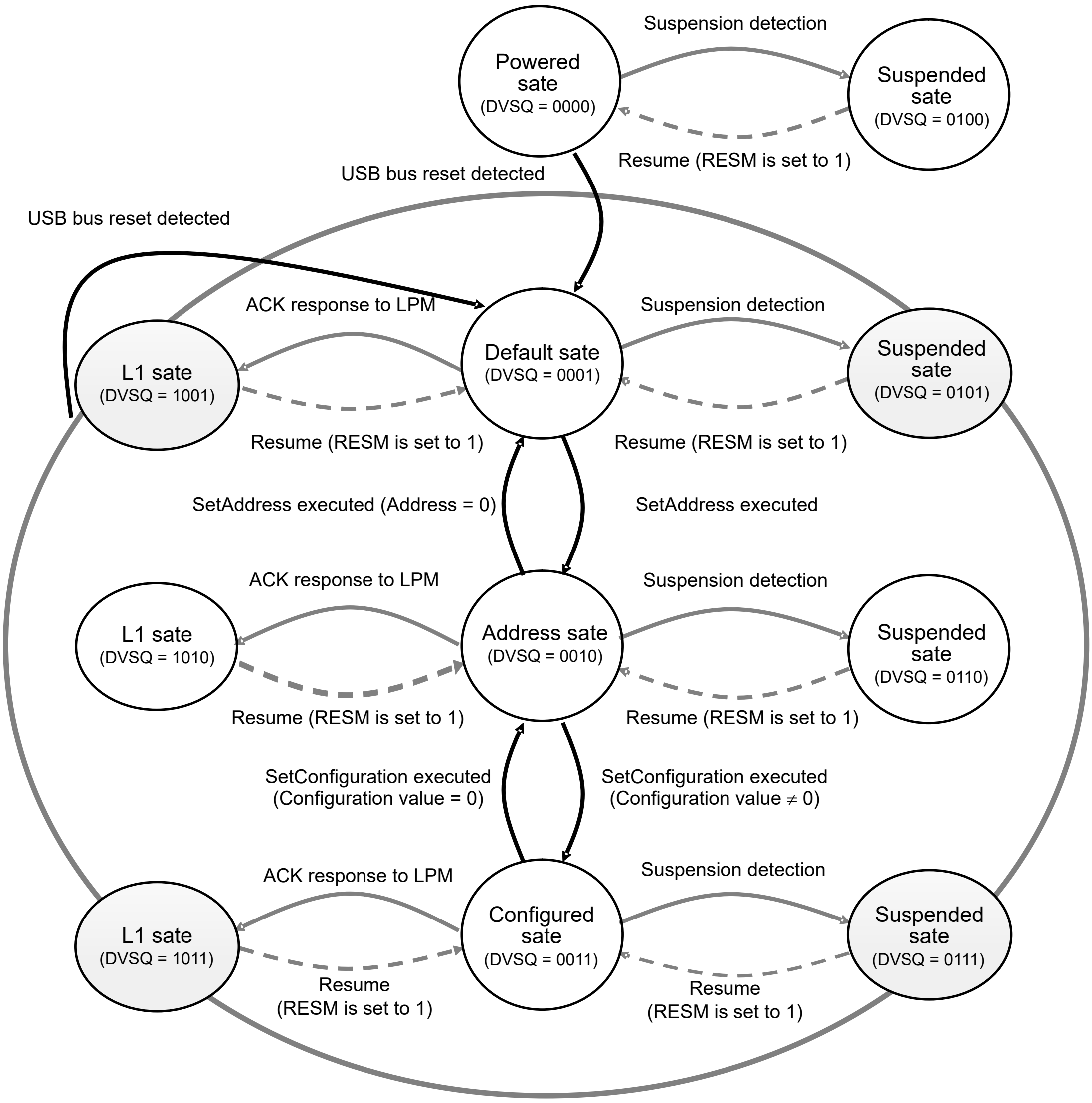
Figure 37.7 Timing of BEMP interrupt generation in device controller mode

37.3.6.4 Device State Transition Interrupt (Device Controller Mode)

Figure 37.8 shows a diagram of the USBHS device state transitions. The USBHS controls device states and generates device state transition interrupts. However, recovery from the Suspend state (resume signal detection) is detected by means of the resume interrupt. Device state transition interrupts can be enabled or disabled independently in INTENB0. Devices whose states have changed can be checked in the PL1CTRL.DVSQ[3:0] flags.

When a transition is made to the default state, a device state transition interrupt is generated after a USB bus reset is detected.

The USBHS controls device states, and device state transition interrupts can be generated, only in device controller mode.



Note 1. For a transition, indicated by a solid line, the DVST bit is set to 1. For a resume, indicated by a dashed line, the RESM bit is set to 1.

Figure 37.8 Device state transitions

37.3.6.5 Control Transfer Stage Transition Interrupt (Device Controller Mode)

Figure 37.9 shows a diagram of the control transfer stage transitions of the USBHS. The USBHS controls the control transfer sequence and generates control transfer stage transition interrupts. Control transfer stage transition interrupts can be enabled or disabled independently in INTENB0. Transfer stages that have transitioned can be checked in the INTSTS0.CTSQ[2:0] bits.

Control transfer stage transition interrupts are generated only in device controller mode. This section describes control transfer sequence errors. When an error occurs, the DCPCTR.PID[1:0] bits are set to 1xb (STALL response).

- Control read transfer errors
  - An OUT token or PING token is received, but no data is transferred in response to the IN token at the data stage
  - An IN token is received at the status stage

– A data packet with DATAPID = DATA0 is received at the status stage

- Control write transfer errors
  – An IN token is received, but no ACK returned in response to the OUT token in the data stage
  – A data packet with DATAPID = DATA0 is received as the first data packet at the data stage
  – An OUT token or PING token is received in the status stage
- Control write no data transfer errors
  – An OUT token or PING token is received at the status stage

At the control write transfer data stage, if the receive data length exceeds the wLength value of the USB request, it is not recognized as a control transfer sequence error. At the control read transfer status stage, packets other than zero-length packets are received by an ACK response and the transfer ends normally.

When a CTRT interrupt occurs in response to a sequence error (INTSTS0.CTRT flag = 1), the CTSQ[2:0] = 110b value is saved until CTRT flag clears to 0, clearing the interrupt status. While CTSQ[2:0] bits = 110b is being saved, no CTRT interrupt for ending the setup stage is generated, even if a new USB request is received. The USBHS saves the setup stage completion status, and it generates a CTRT interrupt after the interrupt status is cleared by software.

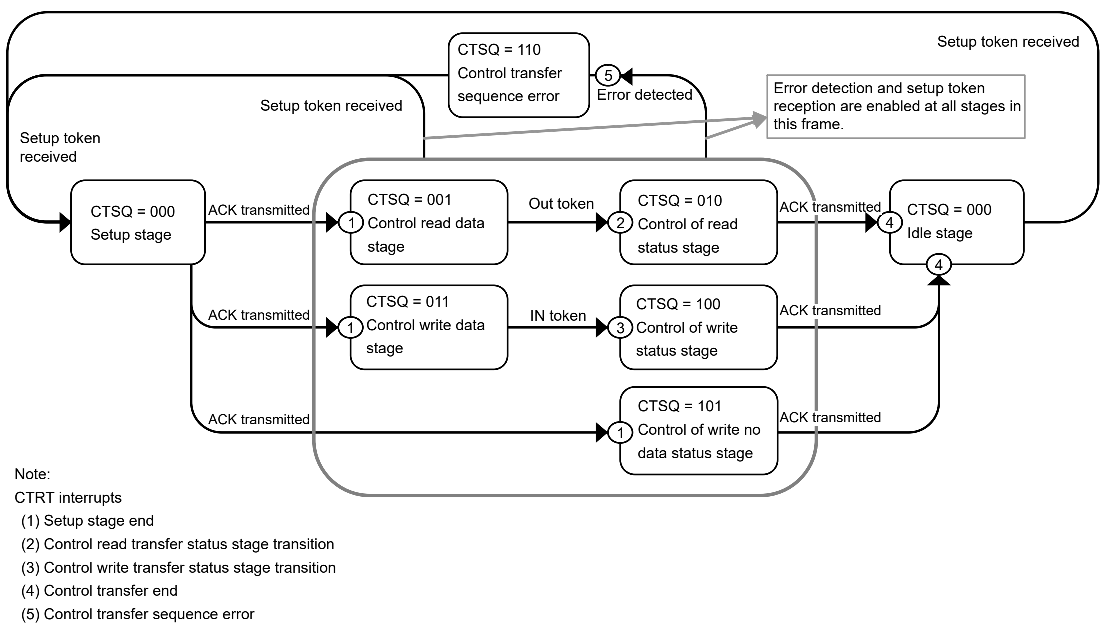


Figure 37.9 Control transfer stage transitions

37.3.6.6 Frame Update Interrupt

In host controller mode, an interrupt is generated when the frame number is updated.

In device controller mode, an SOFR interrupt is generated when the frame number is updated. The USBHS updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation.

37.3.6.7 VBUS Interrupt

When the USBHS_VBUS pin level changes, a VBUS interrupt is generated. The level of the USBHS_VBUS pin can be checked in the INTSTS0.VBSTS flag. Whether the host controller is connected or disconnected can be confirmed using the VBUS interrupt. If the system is activated with the host controller connected, the first VBUS interrupt is not generated, because there is no change in the USBHS_VBUS pin level.

37.3.6.8 Resume Interrupt

In device controller mode, a resume interrupt is generated when the device is in the Suspend state and the USB bus state has changed (from J-state to K-state, or from J-state to SE0). Recovery from the Suspend state is detected by means of the resume interrupt.

In host controller mode, no resume interrupt is generated. Use the BCHG interrupt to detect a change in the USB bus state.

37.3.6.9 OVRCR Interrupt

An OVRCR interrupt is generated when the USBHS_OVRCURA, USBHS_OVRCURA-DS, USBHS_OVRCURB or USBHS_OVRCURB-DS pin level has changed. The levels of the USBHS_OVRCURA, USBHS_OVRCURA-DS, USBHS_OVRCURB or USBHS_OVRCURB-DS pins can be checked in the SYSSTS0.OVCMON[1:0] flags. The external power supply IC can check whether overcurrent is detected using the OVRCR interrupt.

For OTG connections, the OVRCR interrupt allows you to check whether a change is detected in the VBUS comparator.

37.3.6.10 BCHG Interrupt

A BCHG interrupt is generated when the USB bus state has changed. The BCHG interrupt can be used to detect whether a peripheral device is connected. It can also be used to detect a remote wakeup in host controller mode. The BCHG interrupt is generated in both host and device controller modes.

37.3.6.11 DTCH Interrupt

A DTCH interrupt occurs when a USB bus disconnect is detected in host controller mode. The USBHS detects bus disconnects in compliance with the USB 2.0 specification.

On interrupt detection, all pipes in which communications are being carried out for the relevant port must be terminated by software. The pipes enter the wait state for a bus connection to the port, waiting for an ATTCH interrupt to occur. Regardless of the value set in the associated interrupt enable bit, the USBHS hardware:

- Sets the DVSTCTR0.UACT bit for the port in which the DTCH interrupt is detected to 0
- Puts the port in which the DTCH interrupt occurred into the idle state

37.3.6.12 SACK Interrupt

A SACK interrupt is generated when an ACK response for the transmitted setup packet is received from the peripheral device in host controller mode. The SACK interrupt can be used to confirm that the setup transaction is successfully complete.

37.3.6.13 SIGN Interrupt

A SIGN interrupt is generated when an ACK response for the transmitted setup packet is not correctly received from the peripheral device three consecutive times in host controller mode. The SIGN interrupt can be used to detect no ACK response transmitted from the peripheral device or corruption of an ACK packet.

37.3.6.14 ATTCH Interrupt

An ATTCH interrupt is generated when J-state or K-state of the full-speed signal level is detected on the USB port for 2.5 µs in host controller mode. To be more specific, an ATTCH interrupt is detected in any of the following conditions:

- When K-state, SE0, or SE1 changes to J-state, and J-state continues 2.5 µs
- When J-state, SE0, or SE1 changes to K-state, and K-state continues 2.5 µs

37.3.6.15 EOFERR Interrupt

An EOFERR interrupt occurs when the USBHS detects that communication is not complete at the EOF2 timing defined in the USB 2.0 specification.

On interrupt detection, all pipes in which communications are being carried out for the relevant port must be terminated by software, and the port must be re-enumerated. Regardless of the value set in the associated interrupt enable bit, the USBHS hardware:

- Sets the DVSTCTR0.UACT bit for the port in which the EOFERR interrupt is detected to 0
- Puts the port in which the EOFERR interrupt is generated into the idle state

37.3.6.16 PDDETINT Interrupt

The USBHS sets the INTSTS1.PDDETINT flag to 1 on detecting a level change (high to low or low to high) in the PDDET pin input value and generates the PDDETINT interrupt. When the PDDETINT interrupt is generated, use software to repeatedly read the BCCTRL.PDDETSTS flag until the same value is read three or more times, and perform debounce processing.

37.3.6.17 LPMEND Interrupt

When the LPM transaction ends because a response from the peripheral device or a timeout is detected, the INTSTS1.LPMEND flag sets to 1 and the LPMEND interrupt is generated.

37.3.6.18 L1RSMEND Interrupt

When performing resume processing when the USBHS has transitioned to the L1 state because an ACK is received in response to an LPM token, the USBHS sets the INTSTS1.L1RSMEND flag to 1 on completion of the resume processing.

37.3.7 Pipe Control

Table 37.22 lists the pipe settings for the USBHS. USB data transfer is performed through logical pipes that the software associates with endpoints. The USBHS provides 10 pipes for data transfer. Set up the pipes based on your system specifications.

Table 37.22 Pipe settings (1 of 2)

Register name	Bit name	Setting	Notes
DCPCFG PIPECFG	TYPE[1:0]	Transfer type	Pipes 1 to 9: Settable
	BFRE	BRDY interrupt mode	Pipes 1 to 5: Settable
	DBLB	Double buffer select	Pipes 1 to 5: Settable
	CNTMD	Selection of continuous transfer or discontinuous transfer	Pipes 1, 2: Settable only for bulk transfers Pipes 3 to 5: Settable
	DIR	Transfer direction select	IN or OUT settable
	EPNUM[3:0]	Endpoint number	Pipes 1 to 9: Settable Set this number to a value other than 0000 when one or more pipes are used.
	SHTNAK	Selects disabled state for pipe when transfer ends	Pipes 1, 2: Settable only for bulk transfers Pipes 3 to 5: Settable
PIPEBUF	BUFSIZE	FIFO buffer size	DCP: Setting disabled (fixed to 256 bytes) Pipes 1 to 5: Settable up to 2 KB Pipes 6 to 9: Setting disabled (fixed to 64 bytes)
	BUFNMB	FIFO buffer number	DCP: Setting disabled (fixed to 0x0-0x3 area) Pipes 1 to 5: Setting disabled (0x8-0x87 area specifiable) Pipes 6 to 9: Setting disabled (fixed to 0x4-0x7 area)
DCPMAXP PIPEMAXP	DEVSEL[3:0]	Device select	Viewable only in host controller mode
	MXPS	Maximum packet size	Setting compliant with USB specification

Table 37.22 Pipe settings (2 of 2)

Register name	Bit name	Setting	Notes
PIPEPERI	IFIS	Buffer flush	Pipes 1, 2: Settable only for isochronous transfers Pipes 3 to 5: Setting disabled Pipes 6 to 9: Setting disabled
	IITV[2:0]	Interval counter	Pipes 1, 2: Settable only for isochronous transfers Pipes 3 to 5: Setting disabled Pipes 6 to 9: Settable only in host controller mode
DCPCTR PIPEnCTR	BSTS	Buffer status	For the DCP, receive buffer status and transmit buffer status are switched with the ISEL bit
	INBUFM	IN buffer monitor	Available only for pipes 1 to 5
	SUREQ	Setup request	Settable only for the DCP and controlled in host controller mode
	SUREQCLR	SUREQ clear	Settable only for the DCP and controlled in host controller mode
	CSCLR	CSSTS clear	Controllable only in host controller mode
	CSSTS	Split status check	Viewable only in host controller mode
	ATREPM	Auto response mode	Pipes 1 to 5: Settable only in device controller mode
	ACLRM	Auto buffer clear	Pipes 1 to 9: Settable
	SQCLR	Sequence clear	Clears the data toggle bit
	SQSET	Sequence set	Sets the data toggle bit
	SQMON	Sequence check	Monitors the data toggle bit
	PBUSY	PIPE busy check	—
	PID[1:0]	Response PID	—
PIPEnTRE	TRENB	Transaction count enable	Pipes 1 to 5: Settable
	TRCLR	Current transaction counter clear	Pipes 1 to 5: Settable
PIPEnTRN	TRNCNT	Transaction counter	Pipes 1 to 5: Settable

37.3.7.1 Pipe Control Register Switching Procedures

The following bits in the pipe control registers can be changed only when USB communication is prohibited (PID[1:0] bits are 00b (NAK response)). Figure 37.10 shows pipe control register switching procedures when USB communication is enabled (PID[1:0] bits are 00b (BUF response)).

Do not change the following registers and bits when USB communication is enabled (PID[1:0] bits are 01b (BUF response)):

- Bits in DCPCFG and DCPMAXP
- SQCLR and SQSET bits in DCPCTR
- Bits in PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI
- ATREPM, ACLRM, SQCLR, and SQSET bits in PIPEnCTR
- Bits in PIPEnTRE and PIPEnTRN
- Bits in DEVADDn (n = 0 to A)

To set the CSCLR bits and bits in DEVADDn (n = 0 to A), follow the procedures described in section 37.2. Register Descriptions.

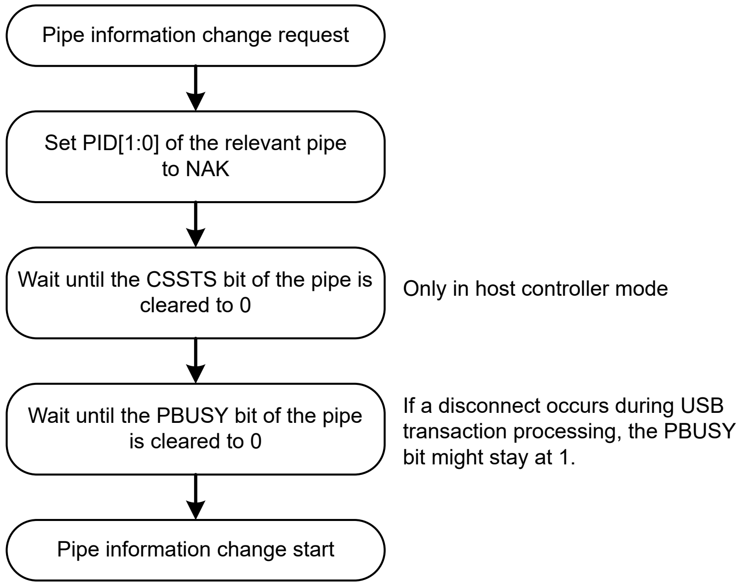


Figure 37.10 Procedure for changing pipe information when USB communication is enabled and PID[1:0] bits are 01b (BUF response)

The following bits in the pipe control registers can be changed only when the selected pipe information is not set in the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

Do not set the following registers while the CURPIPE[3:0] bits are set:

- Bits in DCPCFG and DCPMAXP
- Bits in PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI
- PIPEnCTR and ACLRM bits

To change pipe information, you must set the CURPIPE[3:0] bits to a pipe other than the one to be changed. For the DCP, the buffer must be cleared using the BCLR bit after the pipe information is changed.

37.3.7.2 Transfer Types

The PIPECFG.TYPE[1:0] bits specify the following transfer types for each pipe:

- DCP: No setting necessary (fixed at control transfer)
- Pipes 1 and 2: Set to bulk transfer or isochronous transfer
- Pipes 3 to 5: Set to bulk transfer
- Pipes 6 to 9: Set to interrupt transfer

37.3.7.3 Endpoint Number

The PIPECFG.EPNUM[3:0] bits are used to set the endpoint number for each pipe. The DCP is fixed at endpoint 0. The other pipes can be set from endpoint 1 to 15.

- DCP: No setting is necessary (fixed at endpoint 0)
- Pipes 1 to 9: Select and set the endpoint numbers from 1 to 15 so that the combination of the PIPECFG.DIR and EPNUM[3:0] bits is unique

37.3.7.4 Maximum Packet Size Setting

Specify the maximum packet size for each pipe in the MXPS bits in DCPMAXP and PIPEMAXP. The DCP and pipes 1 to 5 can be set to any of the maximum pipe sizes defined in the USB 2.0 specification. For pipes 6 to 9, the maximum packet size is 64 bytes. Set the maximum packet size as follows before starting a transfer (PID[1:0] bits are set to 01b (BUF response)):

- DCP: Set to 64 for high-speed operation
- DCP: Set to 8, 16, 32, or 64 for full-speed operation
- Pipes 1 to 5: Set to 512 for high-speed bulk transfers
- Pipes 1 to 5: Set to 8, 16, 32, or 64 for full-speed bulk transfers
- Pipes 1, 2: Set between 1 and 1024 for high-speed isochronous transfers
- Pipes 1, 2: Set between 1 and 1023 for full-speed isochronous transfers
- Pipes 6 to 9: Set between 1 and 64

High-bandwidth interrupt transfers and isochronous transfers are not supported.

37.3.7.5 Transaction Counter for Pipes 1 to 5 in the Receiving Direction

When the specified number of transactions is complete in the data packet receiving direction, the USBHS recognizes that the transfer has ended. Two transaction counters are provided. One is the PIPEnTRN register, which specifies the number of transactions to be executed, and the other is the current counter, which internally counts the number of executed transactions. If the PIPECFG.SHTNAK bit is set to 1, when the current counter value matches the specified number of transactions, the associated PIPEnCTR.PID[1:0] bits are set to 00b (NAK response) and the subsequent transfer is disabled. The transactions can be counted again from the beginning by initializing the current counter of the transaction counter function through the PIPEnTRE.TRCLR bit. The data read from PIPEnTRN differs depending on the PIPEnTRE.TRENB setting as follows:

- The TRENB bit = 0: Specified transaction counter value can be read
- The TRENB bit = 1: Current counter value indicating the internally counted number of executed transactions can be read.

The following constraints apply when working with the TRCLR bit:

- If the transactions are being counted and the PIPEnCTR.PID[1:0] bits are set to 01b (BUF response), the current counter cannot be cleared
- If there is any data left in the buffer, the current counter cannot be cleared

37.3.7.6 Response PID

Specify the response PID for each pipe in the PID[1:0] bits in DCPCTR and PIPEnCTR. This section describes the USBHS operation with different response PID settings.

Software response PID settings in host controller mode

Select the response PID to specify the execution of transactions as follows:

- NAK setting: Using pipes is disabled and no transactions are executed
- BUF setting: Transactions are executed based on the FIFO buffer state:
  OUT direction: An OUT token is issued if the FIFO buffer contains transmit data.
  IN direction: An IN token is issued if the FIFO buffer is not full and can receive data.
- STALL setting: Using pipes is disabled and no transactions are executed

Note: Use the SUREQ bit to execute setup transactions for the DCP.

Software response PID settings in device controller mode

Select the response PID to respond as follows to transactions from the host:

- NAK setting: A NAK response is returned to all generated transactions

- BUF setting: A response is returned to transactions based on the FIFO buffer
- STALL setting: A STALL response is returned to all generated transactions

Note: For setup transactions, an ACK response is always returned, regardless of the PID[1:0] bits setting, and the USB request is stored in the register.

Hardware response PID settings in host controller mode and Hardware response PID settings in device controller mode describe situations in which the USBHS writes to the PID[1:0] bits because of specific transaction results.

Hardware response PID settings in host controller mode

- NAK setting: The PID[1:0] bits are set to 00b (NAK response) in the following cases, and issuing of tokens is automatically stopped:
  - When a non-isochronous transfer is performed and an NRDY interrupt is generated. For details, see section 37.3.6.2. NRDY Interrupt
  - If a short packet is received when the PIPECFG.SHTNAK bit is set to 1 for bulk transfers
  - If transaction counting ends when the SHTNAK bit is set to 1 for bulk transfers
- BUF setting: The USBHS does not write this setting
- STALL setting: The PID[1:0] bits are set to STALL in the following cases, and issuing of tokens is automatically stopped:
  - When STALL is received in response to a transmitted token
  - When a received data packet exceeds the maximum packet size

Hardware response PID settings in device controller mode

- NAK setting: The PID[1:0] bits are set to 00b (NAK response) in the following cases, and a NAK response is returned to transactions:
  - When the setup token is received normally (DCP only)
  - If transaction counting ends or a short packet is received when the PIPECFG.SHTNAK bit is set to 1 for bulk transfers
- BUF setting: The USBHS does not write this setting
- STALL setting: The PID[1:0] bits are set to STALL in the following cases, and a STALL response is returned to transactions:
  - When a received data packet exceeds the maximum packet size
  - When a control transfer sequence error is detected

37.3.7.7 Data PID Sequence Bit

The USBHS automatically toggles the sequence bit in the data PID when data is transferred successfully in the control transfer data stage, bulk transfer, and interrupt transfer. The sequence bit of the next data PID to be transmitted can be confirmed with the SQMON bit in DCPCTR and PIPEnCTR. When data is transmitted, the sequence bit toggles on ACK handshake reception. When data is received, the sequence bit toggles on ACK handshake transmission. The SQCLR bit in DCPCTR and the SQSET bit in PIPEnCTR can be used to change the data PID sequence bit.

In device controller mode when control transfers are used, the USBHS automatically sets the sequence bit for stage transitions. DATA1 is returned when the setup stage ends. The sequence bit is not referenced and PID = DATA1 is returned in the status stage. Therefore, no software settings are required. However, in host controller mode when control transfers are used, the sequence bit must be set by software for the stage transitions.

For ClearFeature requests for transmission or reception, the data PID sequence bit must be set by software in both host and device controller modes.

37.3.7.8 Response PID = NAK Function

The USBHS provides a function for disabling pipe operation (PID[1:0] bits are set to 00b (NAK response)) when the final data packet of a transaction is received. The USBHS automatically distinguishes this based on reception of a short packet or the transaction counter. Enable this function by setting the PIPECFG.SHTNAK bit to 1.

When the double buffer mode is being used for the FIFO buffer, using this function enables reception of data packets in transfer units. If pipe operation is disabled, software must enable the pipe again (PID[1:0] bits are set to 01b (BUF response)).

The response PID = NAK function can be used only for bulk transfers.

37.3.7.9 Auto Response Mode

For bulk transfer pipes (1 to 5), when the PIPEnCTR.ATREPM bit is set to 1, a transition is made to auto response mode.

During an OUT transfer (PIPECFG.DIR = 0), OUT-NAK mode is invoked, and during an IN transfer (DIR = 1), null auto response mode is invoked.

37.3.7.10 OUT-NAK Mode

For bulk OUT transfer pipes, NAK is returned in response to an OUT token, and an NRDY interrupt is output when the PIPEnCTR.ATREPM bit is set to 1. To transition from normal mode to OUT-NAK mode, specify OUT-NAK mode while pipe operation is disabled (PID[1:0] = 00b for NAK response). Next, enable pipe operation (PID[1:0] = 01b for BUF response), on which OUT-NAK mode becomes valid. If an OUT token is received immediately before pipe operation is disabled, the token data is normally received, and an ACK is returned to the host.

To transition from OUT-NAK mode to normal mode, cancel OUT-NAK mode while pipe operation is disabled (NAK).

Next enable pipe operation (BUF). In normal mode, reception of OUT data is enabled.

37.3.7.11 Null Auto Response Mode

For bulk IN transfer pipes, zero-length packets are continuously transmitted when the PIPEnCTR.ATREPM bit is set to 1.

To transition from normal mode to null auto response mode, specify null auto response mode while pipe operation is disabled (PID[1:0] bits are set to 00b (NAK response)). Next, enable pipe operation (PID[1:0] bits are set to 01b (BUF response)), on which null auto response mode becomes valid. Before setting null auto response mode, check that PIPEnCTR.INBUFM = 0, because the mode can be set only when the buffer is empty. If the INBUFM bit is 1, empty the buffer using the PIPEnCTR.ACLRM bit. Do not write data from the FIFO port while a transition to null auto response mode is being made.

To transition from null auto response mode to normal mode, keep pipe operation disabled (PID[1:0] bits are set to 00b (NAK response)) for the period of the zero-length packet transmission (about 10 µs) before canceling the null auto response mode. In normal mode, data can be written from the FIFO port, so packet transmission to the host is enabled by enabling pipe operation (PID[1:0] bits are set to 01b (BUF response)).

37.3.8 FIFO Buffer

The USBHS provides a FIFO buffer for data transfers, and it manages the memory area used for each pipe. The FIFO buffer has two states depending on whether the access right is assigned to the system (CPU side) or the USBHS (SIE side).

37.3.8.1 Buffer Status

Table 37.23 and Table 37.24 show the buffer status in the USBHS. The FIFO buffer status can be confirmed using the DCPCTR.BSTS and PIPEnCTR.INBUFM bits. The transfer direction for the FIFO buffer can be specified in the PIPECFG.DIR or CFIFOSEL.ISEL bit (when DCP is selected).

The INBUFM bit is valid for pipes 1 to 5 in the transmitting direction.

When a transmitting pipe uses double buffering, the software can read the BSTS bit to monitor the FIFO buffer status on the CPU side and the INBUFM bit to monitor the FIFO buffer status on the SIE side. When write access to the FIFO port by the CPU or DMAC/DTC is slow and the buffer empty status cannot be determined using the BEMP interrupt, the software can use the INBUFM bit to confirm the end of transmission.

Table 37.23 Buffer status indicated in the BSTS flag

ISEL or DIR	BSTS	FIFO buffer status
0 (receiving direction)	0	There is no received data, or data is being received. Reading from the FIFO port is disabled.
0 (receiving direction)	1	There is received data, or a zero-length packet is received. Reading from the FIFO port is allowed. When a zero-length packet is received, reading is not possible and the buffer must be cleared.
1 (transmitting direction)	0	Transmission is not complete. Writing to the FIFO port is disabled.
1 (transmitting direction)	1	Transmission is complete. CPU write is allowed.

Table 37.24 Buffer status indicated in the INBUFM bit

DIR	INBUFM	FIFO buffer status
0 (receiving direction)	Invalid	Invalid
1 (transmitting direction)	0	Transmission is complete. There is no data waiting to be transmitted.
1 (transmitting direction)	1	The FIFO port has written data to the buffer. There is data to be transmitted.

37.3.8.2 FIFO Buffer Clearing

Table 37.25 shows the methods for clearing the FIFO buffer. The FIFO buffer can be cleared using the BCLR bit in the port control register, DnFIFOSEL.DCLRM, or the PIPEnCTR.ACLRM bit.

Single or double buffering can be selected for pipes 1 to 5 in the PIPECFG.DBLB bit.

Table 37.25 Buffer clearing methods

FIFO buffer clearing mode	Clearing the FIFO buffer on the CPU side	Mode for automatically clearing the FIFO buffer after reading the specified pipe data	Auto buffer clear mode for discarding all received packets
Register used	• CFIFOCTR • DnFIFOCTR	DnFIFOSEL	PIPEnCTR
Bit used	BCLR	DCLRM	ACLRM
Clearing condition	Cleared by writing 1	1: Mode valid 0: Mode invalid	1: Mode valid 0: Mode invalid

Auto buffer clear mode function

The USBHS discards all received data packets if the PIPEnCTR.ACLRM bit is set to 1. If a correct data packet is received, the ACK response is returned to the host controller. The auto buffer clear mode function can only be set in the FIFO buffer reading direction.

Setting the ACLRM bit to 1 and then to 0 clears the FIFO buffer of the selected pipe regardless of the access direction. An access cycle of at least 100 ns is required for the internal hardware sequence processing between ACLRM = 1 and ACLRM = 0.

37.3.8.3 FIFO Port Functions

Table 37.26 shows the settings for the FIFO port functions. In write access, writing data until the maximum packet size is reached automatically enables transmission of the data. To enable transmission before the maximum packet size is reached, set the BVAL flag in the port control register to end writing. To send a zero-length packet, use the BCLR bit to clear the buffer, and then set the BVAL flag to end writing.

In reading, reception of new packets is automatically enabled when all data is read. Data cannot be read when a zero-length packet is received (DTLN[11:0] = 0), so the buffer must be cleared with the BCLR bit. The length of the receive data can be confirmed in the DTLN[11:0] flags in the port control register.

Table 37.26 FIFO port function settings

Register name	Bit name	Description
CFIFOSEL, DnFIFOSEL (n = 0, 1)	RCNT	Selects DTLN[11:0] read mode
	REW	FIFO buffer rewind (re-read, rewrite)
	DCLRM	Automatically clears receive data for a specified pipe after the data is read (only for DnFIFO)
	DREQE	Enables DMAC/DTC transfers (only for DnFIFO)
	MBW[1:0]	FIFO port access bit width
	BIGEND	Selects FIFO port endian
	ISEL	FIFO port access direction (only for DCP)
	CURPIPE[3:0]	Selects the current pipe
CFIFOCTR, DnFIFOCTR (n = 0, 1)	BVAL	Ends writing to the FIFO buffer
	BCLR	Clears the FIFO buffer on the CPU side
	DTLN[11:0]	Checks the length of receive data

FIFO port selection

Table 37.27 shows the pipes that can be selected with the different FIFO ports. The pipe to be accessed must be selected in the CURPIPE[3:0] bits in the port selection register. After a pipe is selected, the software must check whether the written value can be correctly read from the CURPIPE[3:0] bits. (If the previous pipe number is read, it indicates that the USBHS is modifying the pipe.) Next, the software checks that the FRDY flag in the port control register is 1.

In addition, the software must specify the bus width to be accessed in the MBW[1:0] bits in the port selection register.

The FIFO buffer access direction conforms to the PIPECFG.DIR setting. For the DCP only, the ISEL bit in the port selection register determines the direction.

Table 37.27 FIFO port access by pipe

Pipe	Access Method	Port that can be used
DCP	CPU access	CFIFO port register
Pipes 1 to 9	CPU access	CFIFO port register D0FIFO/D1FIFO port register
	DMAC/DTC access	D0FIFO/D1FIFO port register

REW bit

It is possible to temporarily stop access to a pipe being accessed, access a different pipe, and then continue processing for the first pipe again. The REW bit in the port selection register is used for this processing.

If a pipe is selected in the CURPIPE[3:0] bits in the port selection register with the REW bit set to 1, the pointer used for reading from and writing to the FIFO buffer is reset, and reading or writing can be carried out from the first byte. If a pipe is selected with the REW bit set to 0, data can be read and written in continuation from the previous selection, without the pointer being reset.

To access the FIFO port, the software must check that the FRDY bit in the port control register is 1 after selecting a pipe.

37.3.8.4 DMAC/DTC Transfers (D0FIFO and D1FIFO Ports)

Overview of DMAC/DTC transfer

For pipes 1 to 9, the FIFO port can be accessed using the DMAC/DTC. When buffer access for the pipe targeted for DMAC/DTC transfer is enabled, a DMAC/DTC transfer request is issued.

Select the unit of transfer to the FIFO port in the DnFIFOSEL.MBW[1:0] bits, and select the pipe targeted for the DMAC/DTC transfer in the DnFIFOSEL.CURPIPE[3:0] bits. Do not change the selected pipe during the DMAC transfer.

DnFIFO auto clear mode (D0FIFO and D1FIFO port reading direction)

If 1 is set in the DnFIFOSEL.DCLRM bit, the USBHS automatically clears the FIFO buffer of the selected pipe when reading of data from the FIFO buffer is complete.

Table 37.28 shows the packet reception and FIFO buffer clearing processing by software for each of the different settings. As shown in the table, the buffer clearing conditions depend on the value set in the PIPECFG.BFRE bit. Using the DnFIFOSEL.DCLRM bit eliminates the need for the buffer to be cleared by software in any situation that requires buffer clearing. This enables DMAC/DTC transfers without involving software.

The DnFIFO auto clear mode can only be set in the FIFO buffer reading direction.

Table 37.28 Packet reception and FIFO buffer clearing processing by software

Buffer status when packet is received	Register setting			
	DCLRM = 0		DCLRM = 1	
	BFRE = 0	BFRE = 1	BFRE = 0	BFRE = 1
Buffer full	No clearing required	No clearing required	No clearing required	No clearing required
Zero-length packet reception	Clearing required	Clearing required	No clearing required	No clearing required
Normal short packet reception	No clearing required	Clearing required	No clearing required	No clearing required
Transaction count end	No clearing required	Clearing required	No clearing required	No clearing required

37.3.8.5 Allocating the FIFO Buffer

Figure 37.11 shows an example of a memory map of the FIFO buffer. The FIFO buffer is an area shared by the USBHS and the control CPU of the application. There are two situations for the FIFO buffer: (1) access rights are given to the application (CPU side), and (2) access rights are given to the USBHS (SIE side).

An independent area is set for the FIFO buffer for each pipe. A memory area is determined by the first block number and the number of blocks (specified in the BUFNMB[7:0] and BUFSIZE[4:0] bits in PIPEBUF), where 64 bytes is regarded as one block. When the continuous transfer mode is selected in the CNTMD bit in PIPECFG, set the BUFSIZE[4:0] bits to an integral multiple of the maximum packet size. When double buffering is selected in the DBLB bit in PIPECFG, twice the memory area specified in the BUFSIZE[4:0] bits in PIPEBUF is allocated to the same pipe.

Three FIFO ports are used to access (read data from and write data to) the FIFO buffer. Specify the number of the pipe to be allocated to the FIFO port in the CURPIPE[3:0] bits in C/DnFIFOSEL.

The FIFO buffer status of each pipe can be checked in the DCPCTR.BSTS, PIPEnCTR, and INBUFM bits. The FIFO port access rights can be checked in the FRDY flag in C/DnFIFOCTR.

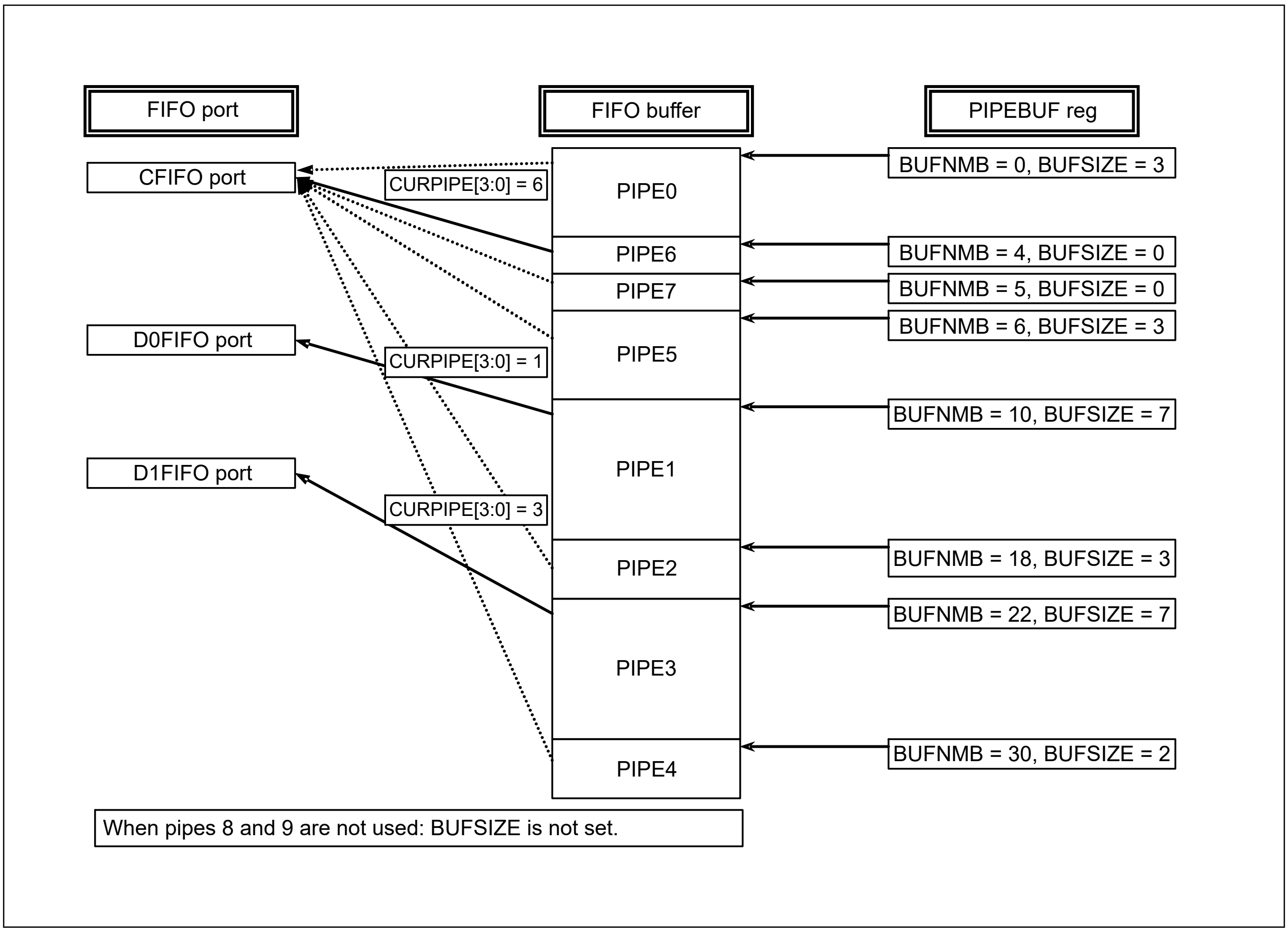


Figure 37.11 Example memory map of the FIFO buffer

37.3.9 Control Transfers Using the DCP

The Default Control Pipe (DCP) is used for data transfers in the control transfer data stage. The FIFO buffer of the DCP is a 64-byte single buffer with a fixed area for both control reads and control writes. The FIFO buffer can be accessed only through the CFIFO port.

37.3.9.1 Control Transfers in Host Controller Mode

Setup stage

The USBREQ, USBVAL, USBINDX, and USBLENG registers are used to transmit USB requests for setup transactions. Writing the setup packet data to the register and then writing 1 to the DCPCTR.SUREQ bit transmits the specified data for the setup transaction. On completion of the transaction, the SUREQ bit clears to 0. Do not change these USB request registers while SUREQ = 1.

When an attached function device is detected, the software must issue the first setup transaction for the device using this sequence with the DCPMAXP.DEVSEL[3:0] bits cleared to 0 and the DEVADD0.USBSPD[1:0] bits set appropriately.

When an attached function device is shifted to the Address state, the software must issue setup transactions using this sequence with the assigned USB Address set in the DEVSEL[3:0] bits and the bits in DEVADDn (n = 0 to A) corresponding to the specified USB Address set appropriately. For example, when PIPEMAXP.DEVSEL[3:0] = 0010b, make appropriate settings in DEVADD2. When PIPEMAXP.DEVSEL[3:0] = 0101b, make appropriate settings in DEVADD5.

When the setup transaction data is sent, an interrupt request is generated based on the response from the peripheral device (SIGN or SACK bit in INTSTS1). This interrupt request allows the software to check the setup transaction result.

The DATA0 data packet (USB request) for the setup transaction is always transmitted regardless of the status of the DCPCTR.SQMON flag.

Data stage

The data stage is used to transfer data using the DCP FIFO buffer.

Before accessing the DCP FIFO buffer, specify the access direction in the CFIFOSEL.ISEL bit. Specify the transfer direction in the DCPCFG.DIR bit.

For the first data packet of the data stage, the data PID must be transferred as DATA1. Set data PID to DATA1 in the DCPCTR.SQSET bit and set the PID[1:0] bits to 01b (BUF response). Completion of data transfer is detected using the BRDY or BEMP interrupt.

Data transfer of multiple packets is enabled in continuous transfer mode. However, when continuous transfer is specified in the receiving direction, the BRDY interrupt is not generated unless the buffer becomes full or a short packet is received (for 256 bytes or less, which is an integral multiple of the maximum packet size). If the transmit data size is an integral multiple of the maximum packet size, control the control write transfer through the software to transmit a zero-length packet last.

Status stage

The status stage is used for zero-length packet data transfers in the reverse direction of the data stage. As in the data stage, data is transferred using the DCP FIFO buffer. Transactions are executed using the same procedure as the data stage.

Data packets in the status stage must be transmitted and received with the data PID set to DATA1 using the DCPCTR.SQSET bit.

When a zero-length packet is received, check the receive-data length in the CFIFOCTR.DTLN[11:0] flags after a BRDY interrupt is generated, and then clear the FIFO buffer using the BCLR bit.

37.3.9.2 Control Transfers in Device Controller Mode

Setup stage

The USBHS returns an ACK response to a normal setup packet for the USBHS. The USBHS operates in the setup stage as follows:

On receiving a new setup packet, the USBHS sets the following bits:

- Sets the INTSTS0.VALID flag to 1
- Sets the DCPCTR.PID[1:0] bits to 00b (NAK response)
- Sets the DCPCTR.CCPL bit to 0

When the USBHS receives a data packet following a setup packet, it stores the USB request parameters in USBREQ, USBVAL, USBINDX, and USBLENG.

Before performing the response processing for a control transfer, set the VALID flag to 0. When the VALID flag = 1, the PID[1:0] bits cannot be set to 01b (BUF response), and the data stage cannot be terminated.

Using the VALID flag function, the USBHS can suspend a request being processed when it receives a new USB request during a control transfer and return a response to the latest request.

In addition, the USBHS automatically detects the direction bit (bmRequestType bit 8) and the request data length (wLength) in the received USB request. It distinguishes between control read transfers, control write transfers, and no data control transfers, and it controls stage transitions. For an incorrect sequence, a sequence error occurs in the control transfer stage transition interrupt, and the interrupt is reported to the software. For a diagram of the stage control by the USBHS, see Figure 37.9.

Data stage

The DCP must be used to execute data transfers for received USB requests. Before accessing the DCP FIFO buffer, specify the access direction in the CFIFOSEL.ISEL bit.

If the transfer data is larger than the size of the DCP FIFO buffer, execute the data transfer using the BRDY interrupt for control write transfers and the BEMP interrupt for control read transfers.

In high-speed control write transfers, a NYET handshake response is returned based on the FIFO buffer status.

Status stage

Control transfers are terminated by setting the DCPCTR.CCPL bit to 1 while the DCPCTR.PID[1:0] bits are set to 01b (BUF response). Control transfers are terminated by setting the DCPCTR.CCPL bit to 1 while the DCPCTR.PID[1:0] bits are set to 01b (BUF response).

After this setting is made, the USBHS automatically executes the status stage based on the data transfer direction determined at the setup stage. The status stage is executed as follows:

- For control read transfers:
  The USBHS receives a zero-length packet from the USB host and transmits an ACK response
- For control write transfer and no data control transfer:
  The USBHS transmits a zero-length packet and receives an ACK response from the USB host.

Control transfer auto response function

The USBHS automatically responds to a normal SET_ADDRESS request. If the SET_ADDRESS request contains any of the following errors, a response must be returned by software.

- When bmRequestType is not 0x00: except control write transfer
- When wIndex is not 0x00: request error
- When wLength is not 0x00: except no data control transfer
- When wValue is larger than 0x7F: request error
- When PL1CTRL.DVSQ[3:0] flags are 0011b (Configured): control transfer of device state error

A response by the corresponding software is required to all requests other than SET_ADDRESS.

37.3.10 Bulk Transfers (Pipes 1 to 5)

The FIFO buffer usage (setting of single buffer/double buffer or continuous/discontinuous transfer mode) is configurable for bulk transfers. The FIFO buffer size can be set up to 2 KB. The USBHS manages the FIFO buffer state and automatically responds to the PING packet and the NYET handshake.

37.3.10.1 PING Packet Control in Host Controller Mode

In the OUT direction, a PING packet is automatically transmitted by the USBHS. The USBHS starts communication in the transmitting direction beginning with the PING packet. When it receives an ACK handshake in response to the PING packet, the USBHS transmits an OUT packet. The USBHS returns to the PING transmission state on receiving a NAK or NYET response during an OUT transaction. The procedure is as follows:

Starting OUT data transmission

(1) Transmit PING packet
(2) Receive NAK handshake
(3) Transmit PING packet
(4) Receive ACK handshake
(5) Transmit OUT data packet
(6) Receive ACK handshake
(7) Transmit OUT data packet
⋮
(8) Return to (1) on receiving a NAK/NYET handshake

The USBHS returns to the PING packet transmission state when a hardware reset is issued, the NYET or NAK handshake is received, the sequence toggle bit is cleared (SQCLR), or the buffer clear bit (ACLRM) is set.

37.3.10.2 NYET Handshake Control in Device Controller Mode

Table 37.29 lists responses to received tokens during bulk and control transfers. The USBHS returns a NYET response when an available area for only one packet is left in the FIFO buffer when the USBHS has received an OUT token during a bulk or control transfer. When the USBHS receives a short packet, however, it returns an ACK response instead of NYET even when this condition occurs.

Table 37.29 Responses to received tokens

PID[1:0] bit setting	FIFO buffer state	Received token	Response	Note
NAK/STALL	—	SETUP	ACK	—
	—	IN/OUT/PING	NAK/STALL	—
BUF	—	SETUP	ACK	—
	RCV-BRDY	OUT/PING	ACK	When OUT token is received, data packet is received.*1
	RCV-BRDY	OUT	NYET	Data packet is received*2
	RCV-BRDY	OUT (Short)	ACK	Data packet is received*2
	RCV-BRDY	PING	ACK	*2
	RCV-NRDY	OUT/PING	NAK	—
	TRN-BRDY	IN	DATA0/1	Data packet is transmitted
	TRN-NRDY	IN	NAK	—

Note 1. RCV-BRDY: An available area for two packets is left in the FIFO buffer when an OUT token or a PING token is received.
Note 2. RCV-BRDY: An available area for only one packet is left in the FIFO buffer when an OUT token is received.
RCV-NRDY: No available area is left in the FIFO buffer when a PING token is received.
TRN-BRDY: The FIFO buffer contains transmit data when an IN token is received.
TRN-NRDY: The FIFO buffer contains no transmit data when an IN token is received.

37.3.11 Interrupt Transfers (Pipes 6 to 9)

In device controller mode, the USBHS performs interrupt transfers based on the timing dictated by the host controller. In the interrupt transfer, the USBHS ignores PING packets (no response) and does not transmit the NYET handshake, but returns an ACK, NAK, or STALL response.

In host controller mode, the software can set the timing for issuing tokens using the interval counter. The USBHS does not issue a PING token but issues an OUT token, including for transfers in the OUT direction.

The USBHS does not support high-bandwidth interrupt transfers.

37.3.11.1 Interval Counter for Interrupt Transfers in Host Controller Mode

Specify the transaction interval for interrupt transfers in the PIPEPERI.IITV[2:0] bits. The USBHS issues interrupt transfer tokens based on this interval.

(1) Initializing the counter

The USBHS initializes the interval counter under the following conditions:

- Power-on reset:
  This initializes the IITV[2:0] bits
- FIFO buffer initialization using the PIPEnCTR.ACLRM bit:
  This does not initialize the IITV[2:0] bits, but does initialize the count value. Setting the PIPEnCTR.ACLRM bit to 0 starts counting from the value set in IITV[2:0].

(2) Operation when tokens cannot be transmitted or received even on token generation

No token is generated in the following cases even at token generation time. In these cases, the USBHS tries to execute the transaction in the next interval.

- When the PID[1:0] bits are set to NAK or STALL
- When the FIFO buffer is full at token transmit time in the receiving (IN) direction
- When there is no data to be transmitted in the FIFO buffer at token transmit time in the transmitting (OUT) direction

37.3.12 Isochronous Transfers (Pipes 1 and 2)

The USBHS does not support high-bandwidth isochronous transfers but provides the following functions for isochronous transfers:

- Notification of isochronous transfer error

- Interval counter (specified in the PIPEPERI.IITV[2:0] bits)
- Isochronous IN transfer data setup control (IDLY function)
- Isochronous IN transfer buffer flush function (specified in the PIPEPERI.IFIS bit)
- SOF pulse output function

37.3.12.1 Error Detection in Isochronous Transfers

The USBHS provides a function for detecting the errors described in this section, so that when errors occur in isochronous transfers, they can be controlled by software. Table 37.30 and Table 37.31 show the priority order for errors detected by the USBHS and the associated interrupts.

PID errors

- The PID value of the received packet is invalid

CRC errors and bit stuffing errors

- A CRC error is found in a received packet or the bit stuffing is illegal

Maximum packet size exceeded

- The data size of the received packet exceeds the specified maximum packet size

Overrun and underrun errors

In host controller mode:

- The FIFO buffer is full at token transmit time in the IN (receiving) direction
- There is no data to be sent in the FIFO buffer at token transmit time in the OUT (transmitting) direction

In device controller mode:

- There is no data to be sent in the FIFO buffer at token receive time in the IN (transmitting) direction
- The FIFO buffer is full at token receive time in the OUT (receiving) direction

Interval error

In device controller mode, the following cases are treated as an interval error:

- Failure to receive an IN token in the interval frame during an isochronous IN transfer
- Failure to receive an OUT token in the interval frame during an isochronous OUT transfer

Table 37.30 Error detection for token transmission and reception

Detection priority	Error type	Interrupt generated at error detection and status
1	PID error	No interrupts are generated in either host or device controller mode. (Ignored as a corrupted packet.)
2	CRC or bit stuffing error	No interrupts are generated in either host or device controller mode. (Ignored as a corrupted packet.)
3	Overrun or underrun error	An NRDY interrupt is generated to set the OVRN flag to 1 in both host and device controller modes. In device controller mode, a zero-length packet is transmitted in response to an IN token. No data packets are received in response to the OUT token.
4	Interval error	An NRDY interrupt is generated in device controller mode. No interrupt is generated in host controller mode.

Table 37.31 Error detection for data packet reception (1 of 2)

Detection priority	Error type	Interrupt generated at error detection and status
1	PID error	No interrupt is generated. (Ignored as a corrupted packet.)
2	CRC or bit stuffing error	An NRDY interrupt is generated and the FRMNUM.CRCE bit sets to 1 in both host and device controller modes.

Table 37.31 Error detection for data packet reception (2 of 2)

Detection priority	Error type	Interrupt generated at error detection and status
3	Maximum packet size exceeded error	A BEMP interrupt is generated and the PID[1:0] bits set to STALL in both host and device controller modes.

37.3.12.2 DATA PID

The USBHS does not support high-bandwidth transfers. In device controller mode, the USBHS responds as follows to a received PID:

IN direction

- DATA0: Transmitted as data packet PID
- DATA1: Not transmitted
- DATA2: Not transmitted
- mDATA: Not transmitted

OUT direction (full-speed operation)

- DATA0: Received normally as data packet PID
- DATA1: Received normally as data packet PID
- DATA2: Packets ignored
- mDATA: Packets ignored

OUT direction (high-speed operation)

- DATA0: Received normally as data packet PID
- DATA1: Received normally as data packet PID
- DATA2: Received normally as data packet PID
- mDATA: Received normally as data packet PID

37.3.12.3 Interval Counter

The isochronous transfer interval can be set in the PIPEPERI.IITV[2:0] bits. In device controller mode, the interval counter enables functions as shown in Table 37.32. In host controller mode, the USBHS generates the token issuance timing, and the interval counter operation is the same as that for interrupt transfers.

Table 37.32 Interval counter functions in device controller mode

Transfer direction	Function	Conditions for detection
IN	Transmit buffer flush	Failure to receive an IN token successfully in the interval frame during an isochronous IN transfer
OUT	Notification of no reception of token	Failure to receive an OUT token successfully in the interval frame during an isochronous OUT transfer

The interval count is performed when an SOF is received or for complemented SOFs, so the isochronism can be maintained even if an SOF is corrupt. The frame interval can be set to 2IITV (μ) frames.

(1) Counter initialization in device controller mode

The USBHS initializes the interval counter under the following conditions:

- Power-on reset:
  This initializes the PIPEPERI.IITV[2:0] bits
- FIFO buffer initialization using the ACLRM bit:
  This does not initialize the IITV[2:0] bits, but does initialize the count value.

After the interval counter is initialized, the interval count starts under either of the following conditions when a packet is transferred successfully:

- An SOF is received after data is transmitted in response to an IN token, with the PID[1:0] bits set to 01b (BUF response)
- An SOF is received after data is received in response to an OUT token, with PID[1:0] bits set to 01b (BUF response)

The interval counter is not initialized under the following conditions:

- When the PID[1:0] bits are set to NAK or STALL
  This does not stop the interval timer. The USBHS attempts the transaction in the next interval.
- USB bus reset and USB suspension
  This does not initialize the IITV[2:0] bits. When an SOF is received, the interval counter starts counting from the value set before SOF was received.

(2) Interval counting and transfer control in host controller mode

The USBHS controls the interval between token issuance operations based on the PIPEPERI.IITV[2:0] bit settings. Specifically, the USBHS issues a token for a selected pipe once every 2IITV frames.

The USBHS starts counting the token issuance interval at the frame following the frame in which the PID[1:0] bits are set to 01b (BUF response) by software.

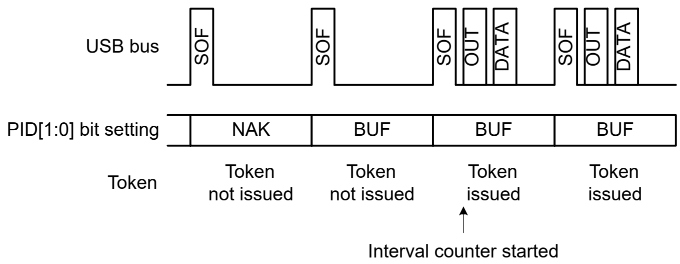


Figure 37.12 Token issuance when IITV[2:0] = 0

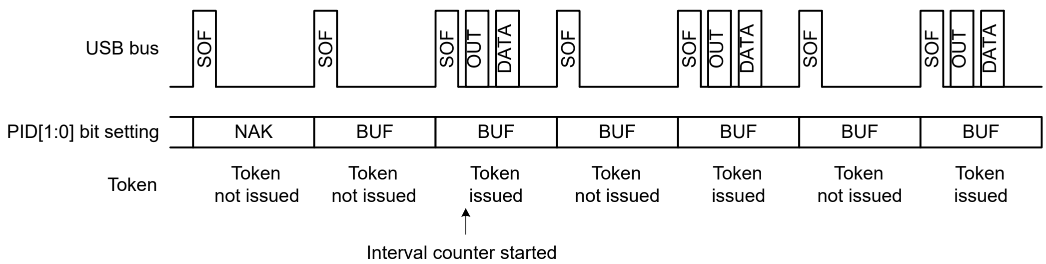


Figure 37.13 Token issuance when IITV[2:0] = 1

When the selected pipe is set for isochronous transfers, the USBHS carries out the following operation in addition to controlling the token issuance interval. The USBHS issues a token even when the NRDY interrupt generation condition is satisfied.

When the selected pipe is for isochronous IN transfers

The USBHS generates an NRDY interrupt when the USBHS issues an IN token but does not receive a packet successfully from a peripheral device (no response or packet error).

When the selected pipe is for isochronous OUT transfers

The USBHS sets the OVRN flag to 1, generating an NRDY interrupt and transmitting a zero-length packet, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer, because the CPU or DMAC/DTC is too slow in writing data to the FIFO buffer.

The token issuance interval is reset on any of the following conditions:

- When the MCU is reset
  This initializes the IITV[2:0] bits
- When the PIPEnCTR.ACLRM bit is set to 1 by software

(3) Interval counting and transfer control in device controller mode

When the selected pipe is for isochronous OUT transfers

The USBHS generates an NRDY interrupt when it fails to receive a data packet within the interval set in the PIPEPERI.IITV[2:0] bits.

The USBHS also generates an NRDY interrupt when it fails to receive data because of a CRC error or other errors contained in the data packet or because of the FIFO buffer is full.

The NRDY interrupt is generated on SOF packet reception. Even if the SOF packet is corrupted, internal complementation allows the interrupt to be generated when the SOF packet is received. However, when the IITV[2:0] bits are set to a value other than 0, the USBHS generates an NRDY interrupt on receiving an SOF packet for every interval after interval counting starts.

When the PID[1:0] bits are set to 00b (NAK response) by software after starting the interval timer, the USBHS does not generate an NRDY interrupt on receiving an SOF packet.

The timing for starting interval counting depend on the IITV[2:0] setting as follows:

- When the IITV[2:0] bits = 0:
  The interval counting starts when the PID[1:0] bits of the selected pipe are changed to BUF
- When the IITV[2:0] bits ≠ 0:
  The interval counting starts on completion of successful reception of the first data packet after the PID[1:0] bits for the selected pipe are changed to 01b (BUF response)

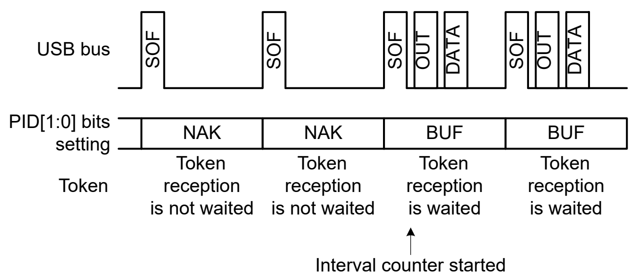


Figure 37.14 Relationship between frames and expected token reception when IITV[2:0] = 0

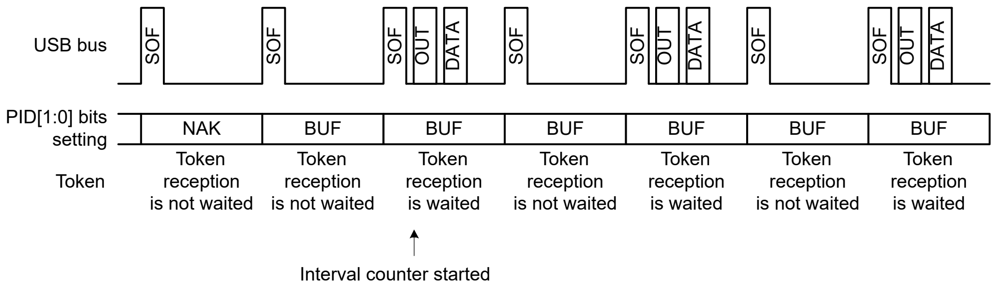


Figure 37.15 Relationship between frames and expected token reception when IITV[2:0] ≠ 0

When the selected pipe is for isochronous IN transfers

The PIPEPERI.IFIS bit must be 1 for this use case. When the IFIS bit is cleared to 0, the USBHS transmits a data packet in response to a received IN token, regardless of the PIPEPERI.IITV[2:0] setting.

When IFIS is 1 and there is data to be transmitted in the FIFO buffer, the USBHS clears the FIFO buffer when it fails to receive an IN token in the frame at the interval set in the IITV[2:0] bits.

The USBHS also clears the FIFO buffer when it fails to receive an IN token successfully because of a bus error, such as a CRC error, contained in the IN token.

The FIFO buffer is cleared on SOF packet reception. Even if the SOF packet is corrupted, the internal complementation allows the FIFO buffer to be cleared when the SOF packet is received.

The timing to start interval counting depends on the IITV[2:0] setting, as with OUT transfers.

The interval is counted on any of the following conditions in device controller mode:

- When a hardware reset is applied to the USBHS (which also sets the IITV[2:0] bits to 000b)
- When the PIPEnCTR.ACLRM bit is set to 1 by software
- When the USBHS detects a USB bus reset

(4) Transmit data setup for isochronous transfers in device controller mode

With isochronous data transmission using the USBHS in device controller mode, after data is written to the FIFO buffer, a data packet can be transmitted in the first frame after the SOF packet is detected. This isochronous transfer transmit data setup function can identify the frame that started transmission.

When the double buffering is used, transmission is only enabled for the buffer in which data writing was completed first, even after the data write to both buffers is complete. Accordingly, even if multiple IN tokens are received, only the one packet of FIFO buffer data is transmitted.

When the FIFO buffer is ready to transmit data when an IN token is received, the data is transferred and a normal response is returned. However, if the FIFO buffer cannot transmit data, a zero-length packet is transmitted and an underrun error occurs.

Figure 37.16 shows an example of transmission using the isochronous transfer transmit data setup function when the IITV[2:0] bits are set to 0 (every frame).

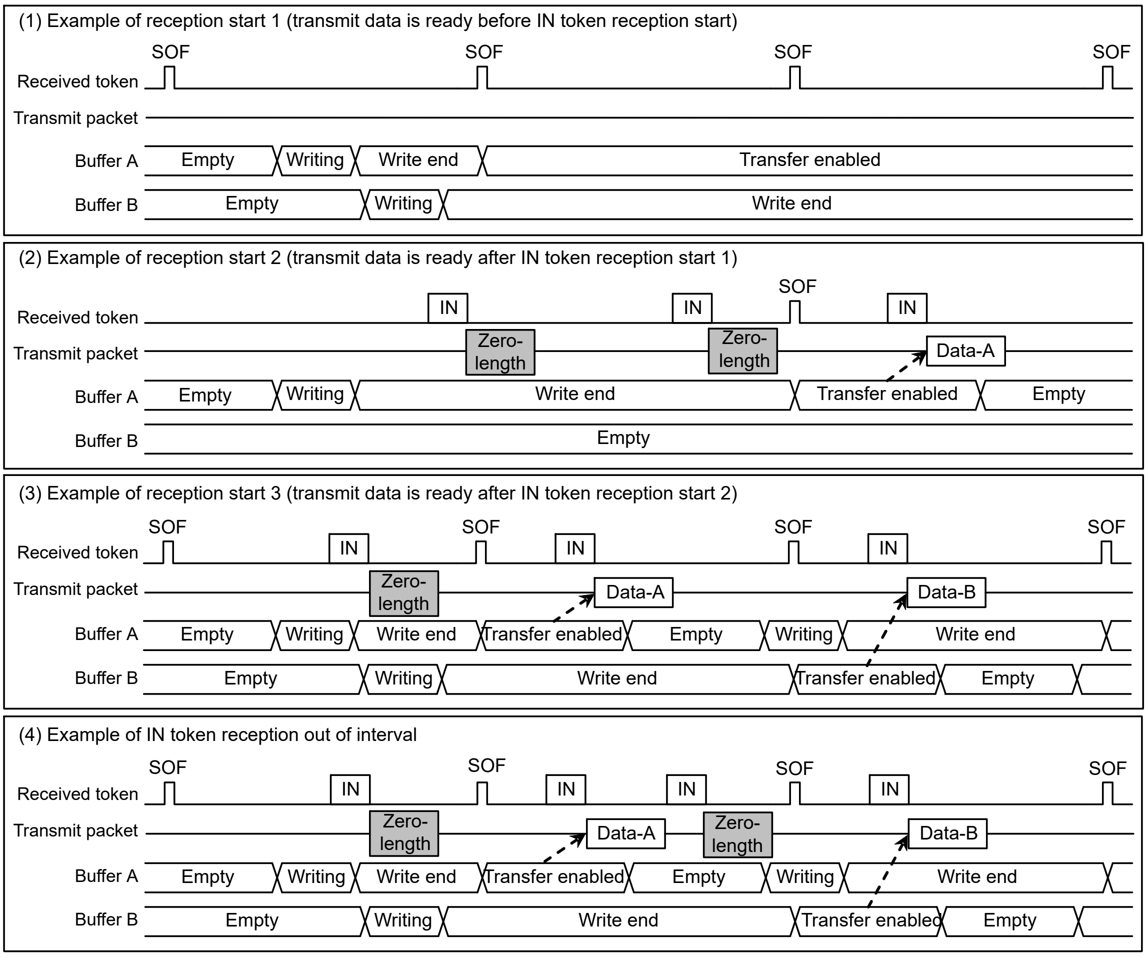


Figure 37.16 Example data setup operation

(5) Isochronous transfer transmit buffer flush in device controller mode

In device controller mode during isochronous data transmission, if the USBHS receives an SOF packet for the next frame without receiving an IN token in the interval frame, it operates as if the IN token is corrupt and clears the buffer that is enabled for transmission, putting that buffer in the writing enabled state.

When double buffering is used and writing to both buffers is complete, the cleared FIFO buffer is assumed to be the one where the data was transmitted in the interval frame, and transmission is enabled for the FIFO buffer that was not cleared on SOF packet reception.

The timing of the buffer flush function depends on the PIPEPERI.IITV[2:0] setting as follows:

- When IITV[2:0] = 0:
  The buffer flush operation starts from the first frame after the pipe is enabled
- When IITV[2:0] ≠ 0:
  The buffer flush operation starts after the first normal transaction

Figure 37.17 shows an example buffer flush. When an unanticipated token is received before the interval frame, the USBHS sends the write data or a zero-length packet as an underrun error, depending on the data setup status.

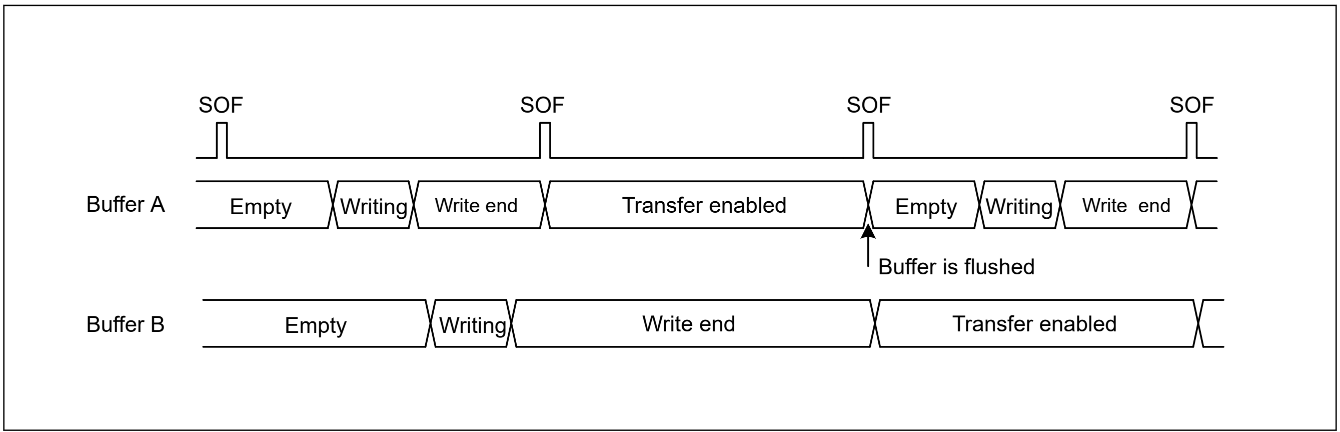


Figure 37.17 Example buffer flush operation

Figure 37.18 shows an example interval error occurrence. There are five types of interval errors, as shown in the figure. An interval error occurs at timing ①, and the buffer flush function is activated.

If an interval error occurs during an IN transfer, the buffer flush function is activated. If it occurs during an OUT transfer, an NRDY interrupt is generated. Use the FRMNUM.OVRN bit to distinguish between this and NRDY interrupts triggered by received packet errors and overrun errors.

For tokens that are shaded in the figure, responses are returned based on the FIFO buffer status.

- IN direction:
  - If the buffer is ready to transfer data, the data is transferred and a normal response is returned
  - If the buffer is not ready to transfer data, a zero-length packet is transmitted and an underrun error occurs
- OUT direction:
  - If the buffer is ready to receive data, the data is received and a normal response is returned
  - If the buffer is not ready to receive data, the received data is discarded and an overrun error occurs

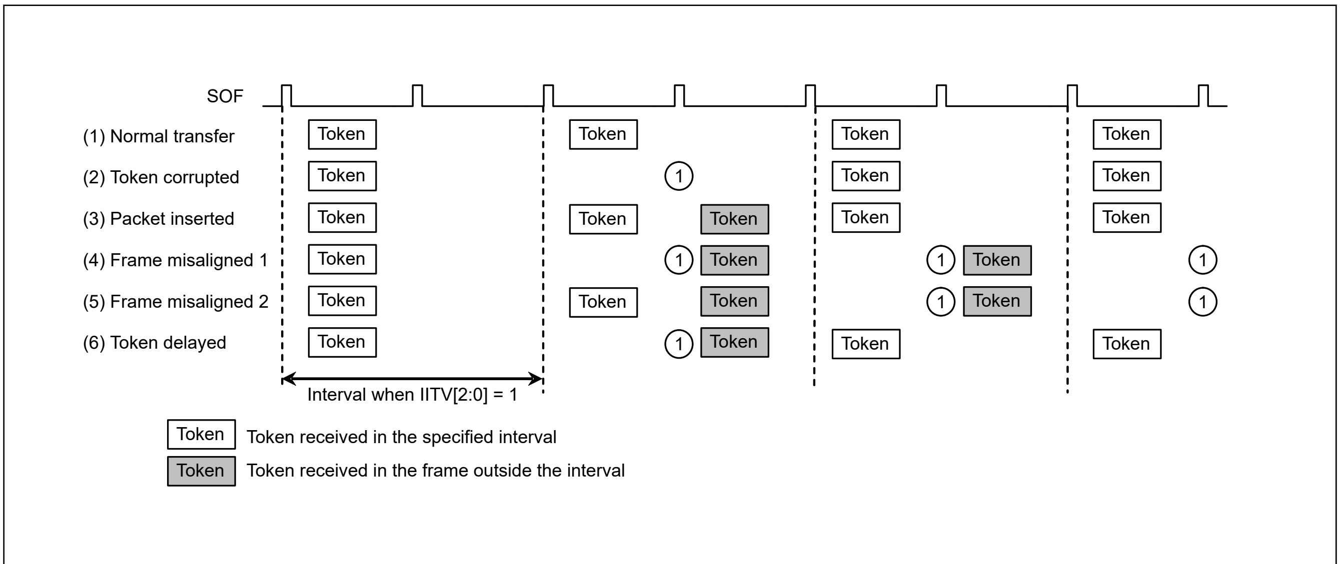


Figure 37.18 Example interval error occurrence when PIPEPERI.IITV[2:0] = 1

37.3.13 SOF Complementation Function

In device controller mode, if packet reception is disabled at intervals of 125 µs in high-speed mode or 1 ms in full-speed mode because the SOF packet is missing or corrupted, the USBHS complements the SOF. SOF complementation begins when the SYSCFG.USBE and LPSTS.SUSPENDM bits are set to 1 and an SOF packet is received. The complementation function is initialized under the following conditions:

- Power-on reset

- USB bus reset
- Suspend state detection

The SOF complementation function operates as follows:

- The frame interval (125 µs or 1 ms) is determined by the reset handshake protocol result
- The complementation function is not activated until an SOF packet is received
- When the first SOF packet is received, complementation is performed by counting 125 µs or 1 ms on the 48-MHz internal clock
- When the second or subsequent SOF packets are received, complementation is performed at the previous reception interval
- Complementation is not performed in the Suspend state or on reception of a USB bus reset. During high-speed operation, complementation continues for 3 ms from the last packet on transition to the Suspend state

The USBHS supports the following functions controlled by SOF packet reception. These functions operate normally with SOF complementation if the SOF packet is missing:

- Updating of the frame number and micro frame number
- SOFR interrupt and micro-SOF lock
- SOF pulse output
- Isochronous transfer interval count

If an SOF packet is missing during full-speed operation, the FFRMNUM.FRNM[10:0] flags are not updated. If a micro-SOF packet is missing during high-speed operation, the URMNUM.UFRNM[2:0] bits are updated.

However, if a micro-SOF packet is missing while the UFRNM[2:0] bits are set to 000b, the FRNM bits are not updated. In this case, even if a subsequent micro-SOF packet with a value other than UFRNM[2:0] bits = 000b is received successfully while UFRNM[2:0] bits are set to the value other than 000b, the FRNM bits are not updated.

37.3.14 Pipe Schedule

37.3.14.1 Conditions for Generating Transactions

In host controller mode and when the DVSTCTR0.UACT bit is set to 1, the USBHS generates transactions under the conditions as shown in Table 37.33.

Table 37.33 Conditions for generating transactions

Transaction	Conditions for generation				
	DIR	PID[1:0]	IITV0	Buffer state	SUREQ
Setup	—*1	—*1	—*1	—*1	1 setting
Control transfer data stage, status stage, bulk transfer	IN	BUF	—*1	Receive area exists	—*1
	OUT	BUF	—*1	Transmit data exists	—*1
Interrupt transfer	IN	BUF	Valid*2	Receive area exists	—*1
	OUT	BUF	Valid*2	Transmit data exists	—*1
Isochronous transfer	IN	BUF	Valid*2	*3	—*1
	OUT	BUF	Valid*2	*4	—*1

Note 1. An em dash (—) in the table indicates that the condition is unrelated to the generating of tokens.
Note 2. “Valid” indicates that, for interrupt transfers and isochronous transfers, a transaction is generated only in transfer frames that are based on the interval counter.
Note 3. This indicates that a transaction is generated regardless of whether there is a receive area. If there is no receive area, however, the received data is discarded.
Note 4. This indicates that a transaction is generated regardless of whether there is any data to be transmitted. If there is no data to be transmitted, however, a zero-length packet is transmitted.

37.3.14.2 Transfer Schedule

This section describes the transfer scheduling within a frame of the USBHS. After the USBHS sends an SOF, the transfer is carried out in the following sequence:

1. Execution of periodic transfers:
   A pipe is searched for in the order of pipe 1 → pipe 2 → pipe 6 → pipe 7 → pipe 8 → pipe 9, and then if there is a pipe for which an isochronous or interrupt transfer transaction can be generated, the transaction is generated.
2. Setup transactions for control transfers:
   The DCP is checked, and if a setup transaction is possible, it is sent.
3. Execution of bulk transfers, control transfer data stages, and control transfer status stages:
   A pipe is searched for in the order of DCP → pipe 1 → pipe 2 → pipe 3 → pipe 4 → pipe 5, and then if there is a pipe for which a transaction for a bulk transfer, a control transfer data stage, or a control transfer status stage can be generated, the transaction is generated.
   When a transaction is generated, processing moves to the next pipe transaction regardless of whether the response from the peripheral device is ACK or NAK. If there is time for transfer within the frame, step 3 is repeated.

37.3.14.3 Enabling USB Communication

Setting the DVSTCTR0.UACT bit to 1 initiates an SOF transmission, and transaction generation is enabled. Setting the UACT bit to 0 stops SOF transmission, and the Suspend state is invoked. If the UACT setting is changed from 1 to 0, processing stops after the next SOF is sent.

37.3.15 Battery Charging Detection Processing

The USBHS provides control over the data contact detection processing (D+ line contact checking), primary detection processing (charger detection processing), and secondary detection processing (charger determination processing) as defined in the Battery Charging Specification.

This section describes operations required in device and host controller modes.

37.3.15.1 Processing in Device Controller Mode

To operate a function device as a battery charging portable device:

1. Start primary detection processing after detecting contact with the D+ and D- lines. The Battery Charging Specification describes two processing methods for Data Contact Detection. The USBHS supports both methods as follows:
   - Software processing
     After a VBINT interrupt or polling of the VBSTS flag indicates a change in the state of the USBHS_VBUS input pin, software controls a wait from 300 to 900 ms. The BCCTRL.VDPSRCE and IDMSINKE bits are then both set to 1, enabling the VDP_SRC and IMP_SINK circuits, respectively, to start primary detection processing.
   - Hardware processing
     Apply 7 to 13 µA of current to the D+ line to hold the D+ line at the logical high level. This is done to detect the D+ and D- lines going to the logical low level because of pull-down resistors on the host device side when the D+ and D- lines come in contact with those of the host. Monitor the SYSSTS0.LNST[1:0] flags while the BCCTRL.IDPSRCE bit is set to 1, enabling the IDP_SRC circuit, to see when the level on the D+ line changes from high to low. After detecting a low level on the D+ line, clear the BCCTRL.IDPSRCE bit to 0, disabling the IDP_SRC circuit, and set both the BCCTRL.VDPSRCE and IDMSINKE bits to 1, enabling the VDP_SRC and IDM_SINK circuits, respectively, to start primary detection processing. The VDPSRCE and IDMSINKE bits must be set to 1 simultaneously.
2. After the start of primary detection processing followed by a software-controlled wait of 40 ms, check the BCCTRL.CHGDETSTS flag. A value of 1 indicates detection of a charger, and secondary detection processing starts.[*1]
3. To start secondary detection processing, clear both the BCCTRL.VDPSRCE and IDMSINKE bits to 0, disabling the VDP_SRC and IDM_SINK circuits, respectively. Next, set both the BCCTRL.VDMSRCE and IDPSINKE bits to 1, enabling the VDM_SRC and IDP_SINK circuits, respectively.
4. After the start of secondary detection processing followed by a software-controlled wait of 40 ms, check the BCCTRL.PDDETSTS flag. A value of 1 indicates that secondary detection processing is complete.

Note 1. In primary detection processing, detection of a voltage above the range from 0.25 to 0.4 V and below the range from 0.8 to 2.0 V on the D-Line indicates that the other device is a host device that supports battery charging (charging downstream port). The BCCTRL.CHGDETSTS flag in the PHY block only indicates whether the voltage on the D-line is higher than the range from 0.25 to 0.4 V, so add processing as required to read the SYSSTS0.LNST[1:0] flags and confirm that the voltage on the D- line is also below the range from 0.8 to 2.0 V.

Figure 37.19 illustrates this processing flow.

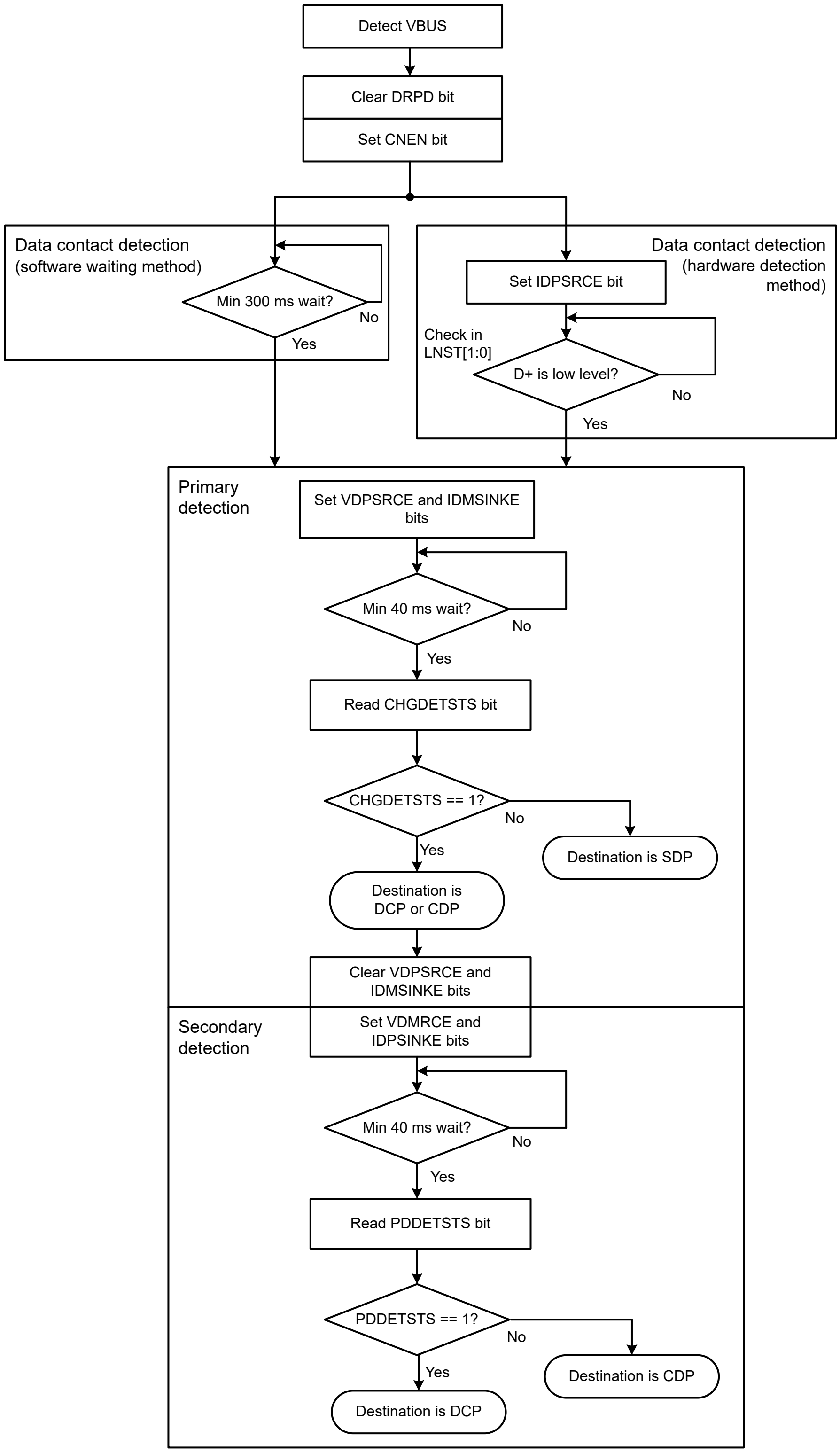


Figure 37.19 Processing flow as portable device

37.3.15.2 Processing in Host Controller Mode

In host controller mode, driving the D- line is required for a portable device to perform primary detection. The USBHS supports the following two primary detection methods:

- When the hardware has a portable device detection function
- When the hardware does not have the function or the function is present but not used

Figure 37.20 and Figure 37.21 show the processing flows for these methods.

(1) When the hardware has a portable device detection function

a. Start driving the USBHS_VBUS input pin.

b. Set the BCCTRL.IDMSINKE bit to 1 to enable the portable device detection circuit.

c. Monitor the portable device detection signal and start driving the D-line when the level of the portable device detection signal is high*1.

d. Stop driving the D-line when the portable device detection signal is at the low level*1.

Note 1. The PDDETINT interrupt indicates a change in the level of the portable device detection signal (EUH_CPDDET), and the current level can be obtained by reading the PDDETSTS flag.

(2) When the hardware does not have a portable device detection function or the function is not used

Software handles the timing of steps a. and b.

a. After a disconnect is detected, start driving the D-line within 200 ms.

b. After a connect is detected, stop driving the D-line within 10 ms.

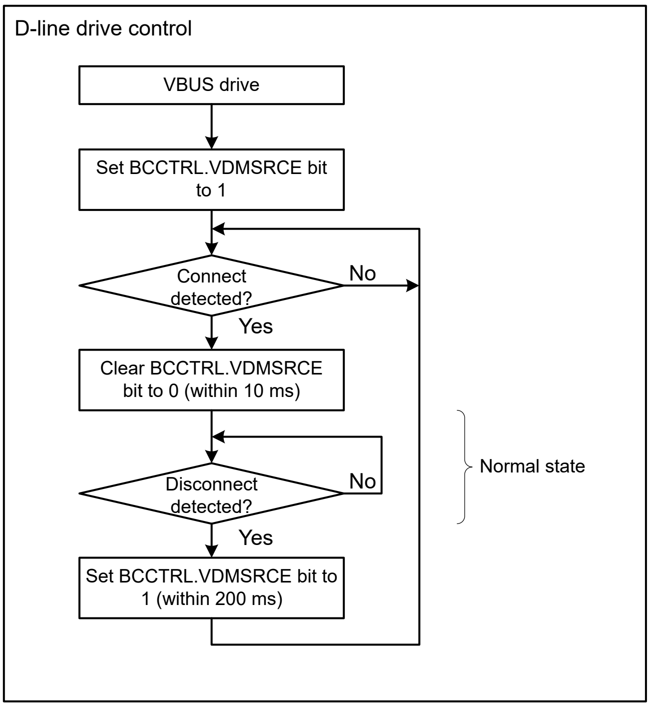


Figure 37.20 Processing flow as charging downstream port without hardware portable device detection function or when function is not used

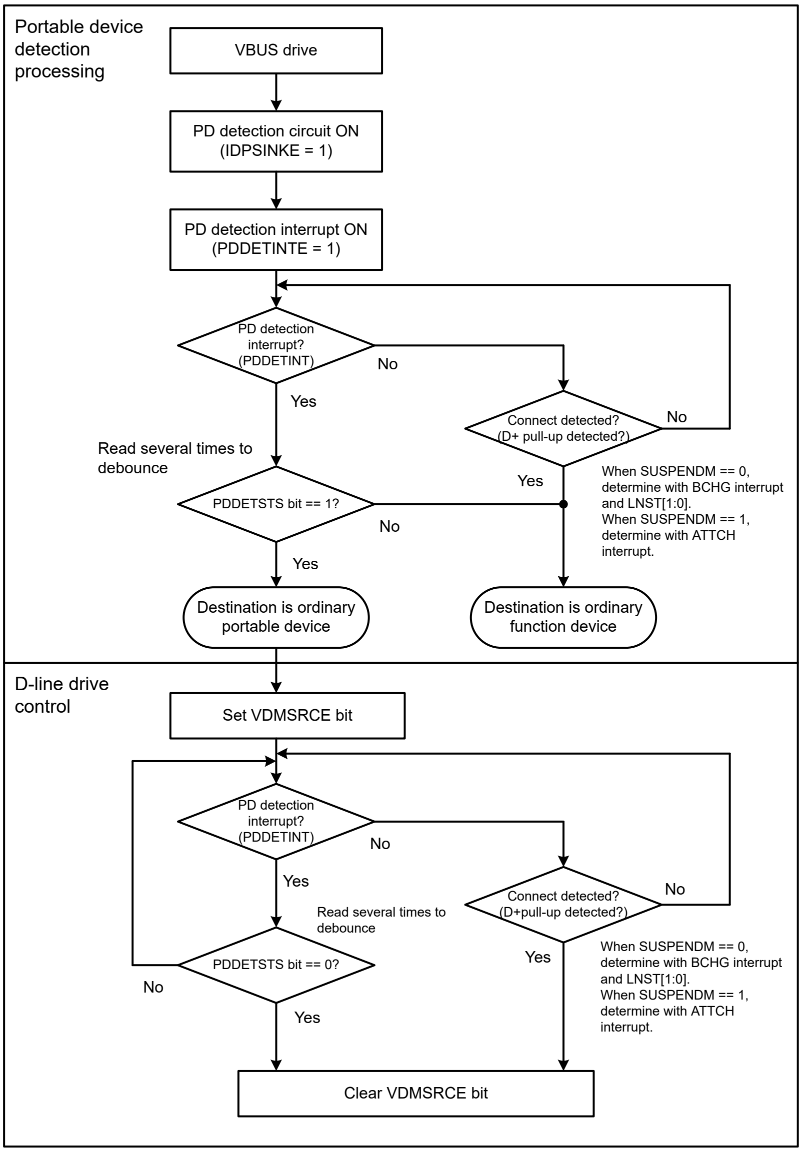


Figure 37.21 Processing flow as charging downstream port with hardware portable device detection function

37.3.16 Link Power Management Processing

The Link Power Management standard defines the existing Suspend state as the L2 state and also defines the L1 state as a state that allows transition and return with lower latency than the L2 state (Suspend). Table 37.34 provides a comparison between the L2 (Suspend) and L1 states.

Table 37.34 Comparison between L2 (Suspend) state and L1 state (1 of 2)

Parameter	L1 state	L2 (Suspend) state
Transition	LPM transaction	Idle for 3 ms

Table 37.34 Comparison between L2 (Suspend) state and L1 state (2 of 2)

Parameter	L1 state	L2 (Suspend) state
Return caused by host	Host: Minimum drive period (75 µs to 1.175 ms) can be specified by the host. Function: 10-µs K drive	Host: Minimum 20-ms K drive Function: 10-ms K drive
Return caused by function	Device: 50-µs K drive Function: 60- to 990-µs K drive Device: 10-µs K drive	Function: 1- to 15-ms K drive Host: Minimum 20-ms K drive Function: 10-ms K drive
Signaling	Full-speed and low-speed idle	Full-speed and low-speed idle

37.3.16.1 Processing in Device Controller Mode

(1) Descriptor contents

In device controller mode, the USBHS must return its descriptor on receiving the GetDescriptor command.

Change the content of the descriptor to be returned depending on whether the transition to and return from the L1 state corresponds to the processing for the LPM transaction. The following table shows the relationship between LPM correspondence and the descriptor.

Table 37.35 Relationship between LPM correspondence and descriptor

Correspondence with LPM	bcdUSB field	USB2.0 extension descriptor		Response to received LPM request	Notes
		Provided/ not provided	Value of LPM bit		
Does not correspond	0x0200	Not provided	—	No response	Normal operation when the LPM is not supported
	0x0201	Provided	0	STALL	Setting for clear non-correspondence to LPM. In this case, a STALL response must be returned.
Corresponds	0x0201	Provided	1	ACK or NYET	Normal operation when the LPM is supported

Declare whether to correspond to the transition to and return from L1 in the LPM bit in the USB 2.0 extension descriptor. To provide the USB2.0 extension descriptor, the bcdUSB field of the device descriptor must be set to a value of 0x0201 or larger.

When the LPM is not supported, the USB2.0 extension descriptor is not provided and the bcdUSB field value must be 0x0200. If an LPM token is received in this case, it must be ignored. It is also possible to set the bcdUSB field value to 0x0201 and the LPM bit in the USB2.0 extension descriptor to 0 (LPM tokens not supported). In this case, the LPM token cannot be ignored and a STALL response must be returned.

When the LPM token is supported, set the bcdUSB field value to 0x0201 and set the LPM bit in the USB 2.0 extension descriptor to 1 (LPM tokens supported). This allows acknowledgment when returning a NYET or ACK response to the LPM token.

(2) Processing during LPM token reception

Transition to and return from the L1 state in device controller mode is as follows:

a. When the USBHS receives an LPM token from the host, the L1RESPEN, L1RESPMD[1:0], and L1NEGOMD settings in PL1CTRL1 determine whether a response packet is sent or the token is ignored and, if a response is to be sent, whether it is an ACK, NYET, or STALL packet.

b. If an ACK response to the LPM token is sent and the host does not transmit another LPM token in 8 µs, the USBHS enters the L1 state. The USBHS handles detection of the newly transmitted packet and the transition to the L1 state. The DVST interrupt can be used to detect the transition.

c. Two types of processing can return the USBHS from the L1 state:

- When the host drives the D-line in the K-state:
  The function device detects the K-state and starts processing the return from the L1 state in response to an RESM interrupt request
- When the function device outputs a remote wakeup signal:
  If the software on the function device sets the DVSTCTR0.WKUP bit to 1, it sends a remote wakeup signal to the host

The software clears the DVSTCTR0.WKUP bit to 0 on returning from the L2 (Suspend) state, and the USBHS clears the DVSTCTR0.WKUP bit to 0 for return from the L1 state.

(3) HIRD field value negotiation function

The HIRD field value included in the LPM token indicates the host K-drive period on return from the L1 state. The HIRD field value can be adjusted according to the requirements of the target system. For example, a small HIRD field value is better for systems focusing on higher transfer efficiency, while a large HIRD field value is better for systems focusing on low power consumption.

Based on the L1NEGOMD and HIRDTHR[3:0] settings in PL1CTRL1, an ACK response is returned when the received HIRD field value is in the expected range, and otherwise a NYET response is returned, requesting the host to change the HIRD field value.

Note: This HIRD field value negotiation function at the host must also support negotiation processing.

37.3.16.2 Processing in Host Controller Mode

(1) Processing during LPM token transmission

Transition to and return from the L1 state in host controller mode is as follows:

a. When the HL1CTRL.L1REQ bit is set to 1, an LPM token is sent to the function device from the host device.

b. If an ACK response is received from the function device, a transition to the L1 state starts within 10 µs and is complete within 50 µs. If a transaction error is detected, another LPM token is transmitted within 8 µs. Retransmission can proceed up to two times. The USBHS handles all of this processing.

c. Two types of processing can return the USBHS from the L1 state:

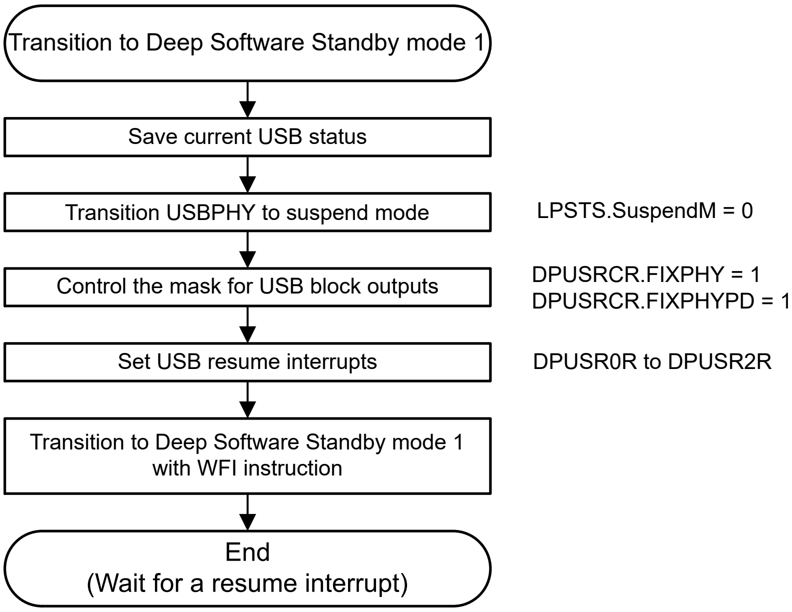
- When the host drives the D-line for the K state:
  When the DVSTCTR0.RESUME bit is set to 1, the host device starts driving the D-line for the K-state and starts processing the return
- When the function device generates a remote wakeup signal:
  When the host device detects a remote wakeup signal from the function device, it sets the DVSTCTR0.RESUME bit to 1 and starts driving the D-line for the K-state

Unlike when returning from the Suspend (L2) state, the USBHS clears the DVSTCTR0.RESUME bit to 0. After clearing the RESUME bit, it sets the DVSTCTR0.UACT bit to 1 and issues an L1RSMEND interrupt request.

37.3.17 Deep Software Standby Mode 1 Because of USB Suspend/Resume Interrupts

Deep Software Standby mode 1 can be canceled by a USB suspend/resume interrupt. USB suspend/resume interrupts are detected by the USB resume detecting unit, which controls and monitors the USB I/O pins to detect the interrupts.

Figure 37.22 shows the flow for setting the USBHS when entering Deep Software Standby mode 1 from either host or device controller mode. Figure 37.23 and Figure 37.24 show the flows for setting the USBHS when canceling Deep Software Standby mode 1 from host controller mode. Figure 37.25 shows the flow for setting the USBHS when canceling Deep Software Standby mode 1 from device controller mode.



Note: When connected to an external device, transition to Deep Software Standby mode 1 is enabled only on USB Suspend.

Note: Never invoke Deep Software Standby mode 1 in the following cases:

- In an L1 Suspend status with the USB-LPM protocol
- When remote wakeup is enabled in host controller mode.

Figure 37.22 USBHS setup flow for transition to Deep Software Standby mode 1 as a host or device controller

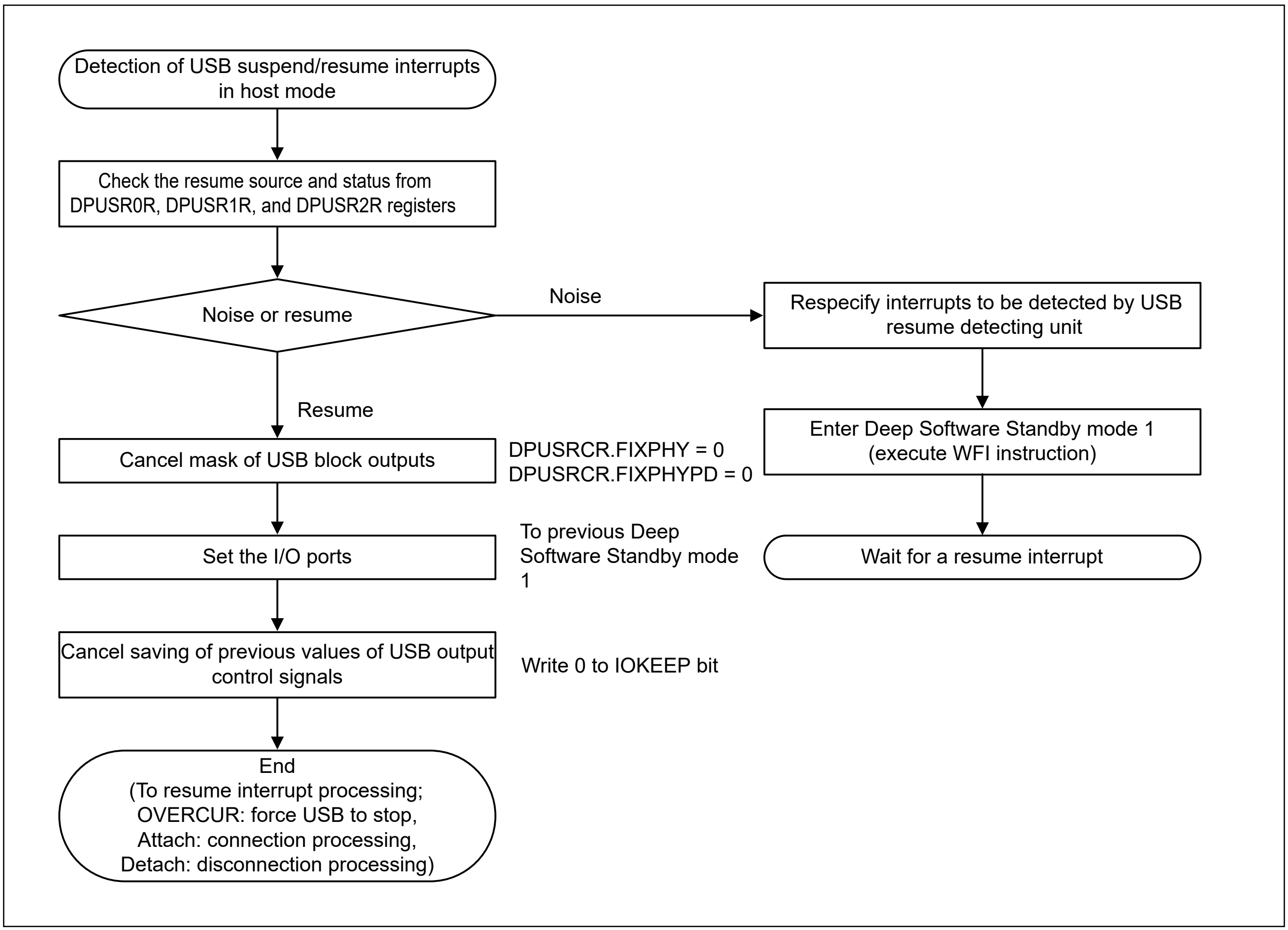


Figure 37.23 USBHS setup flow for canceling Deep Software Standby mode 1 as a host controller (1)

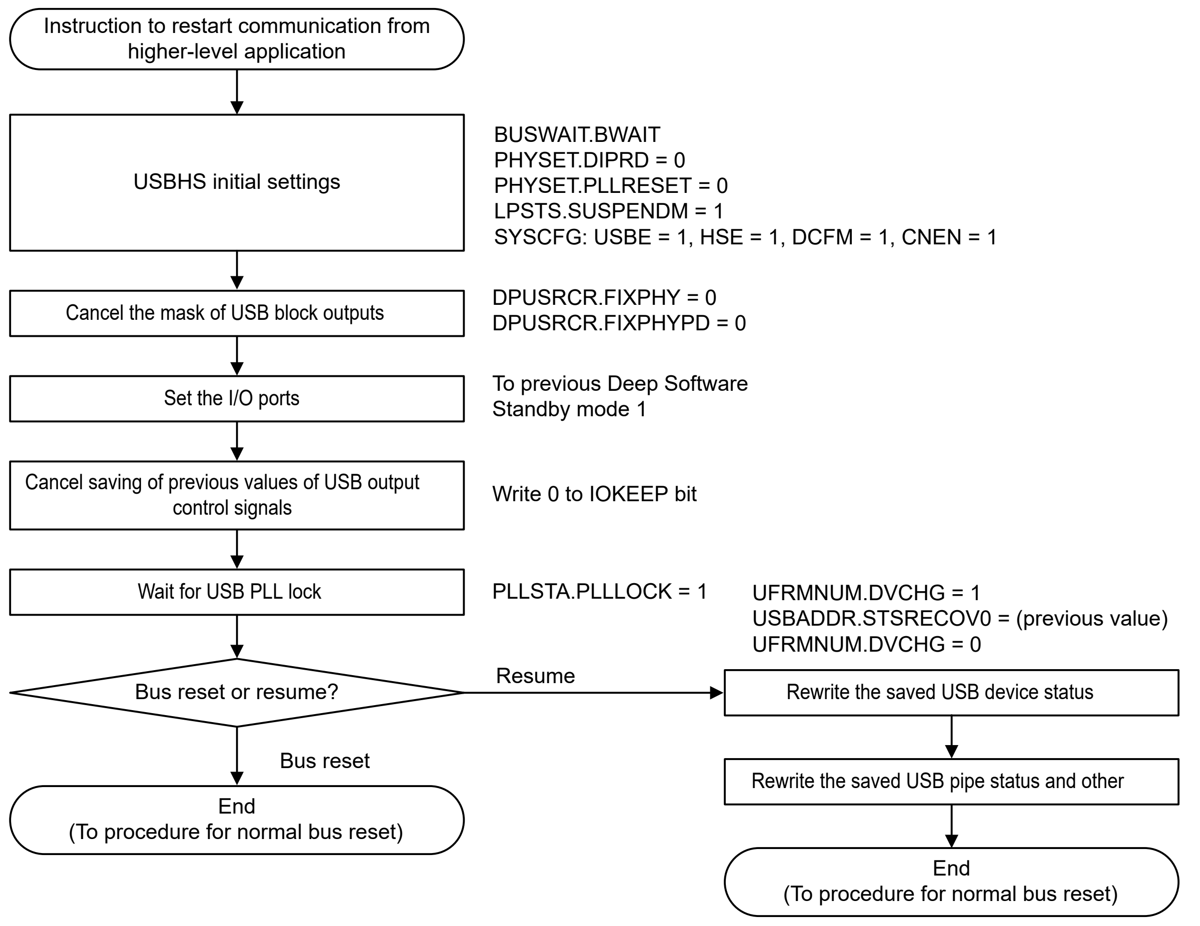
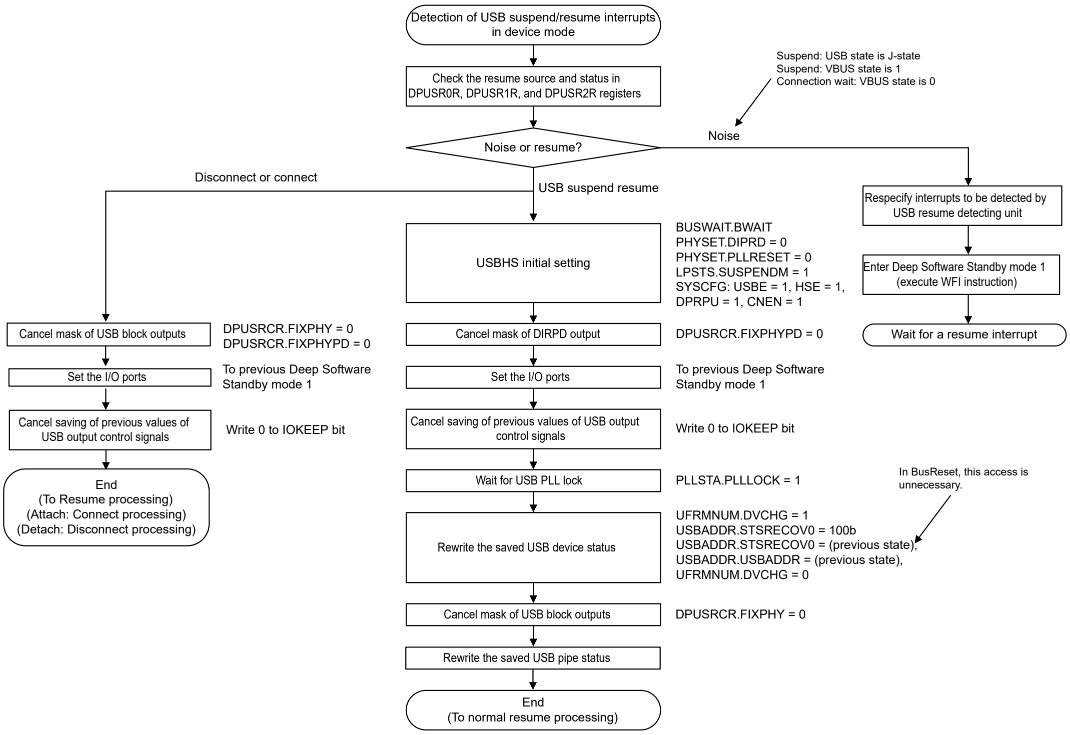


Figure 37.24 USBHS setup flow for canceling Deep Software Standby mode 1 as a host controller (2)



Note 1. In Bus Reset return, the write to the STSRECOV0 bit must be within 2.5 μs of USBHS-PHY PLL oscillation start.

Figure 37.25 USBHS setup flow for transition to Deep Software Standby mode 1 as a host or device controller

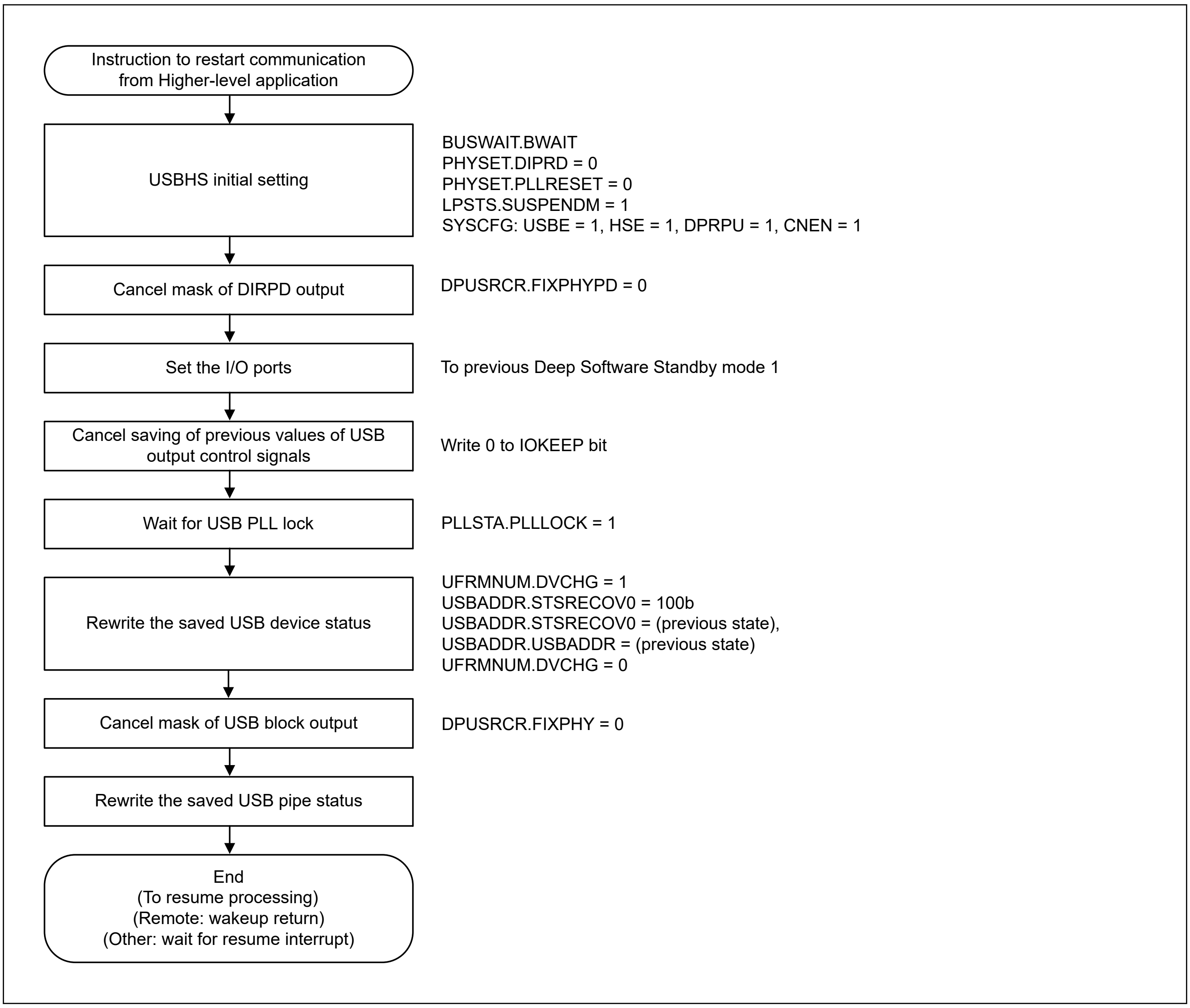


Figure 37.26 USBHS setup flow for canceling Deep Software Standby mode 1 as a device controller (2)

37.3.18 Example External Connection Circuits

Figure 37.27 shows an example OTG connection in a self-powered system. The USBHS controls the pull-up resistor of the D+ line and the pull-down resistor of D+ and D- lines. Select pull-up and pull-down for the lines in the SYSCFG.DPRPU and SYSCFG.DRPD bits. In device controller mode, the pull-up resistor of USB data line is disabled if SYSCFG.DPRPU bit is set to 0 while communicating with the USB host. The USBHS can use this to notify the USB host of a device disconnect.

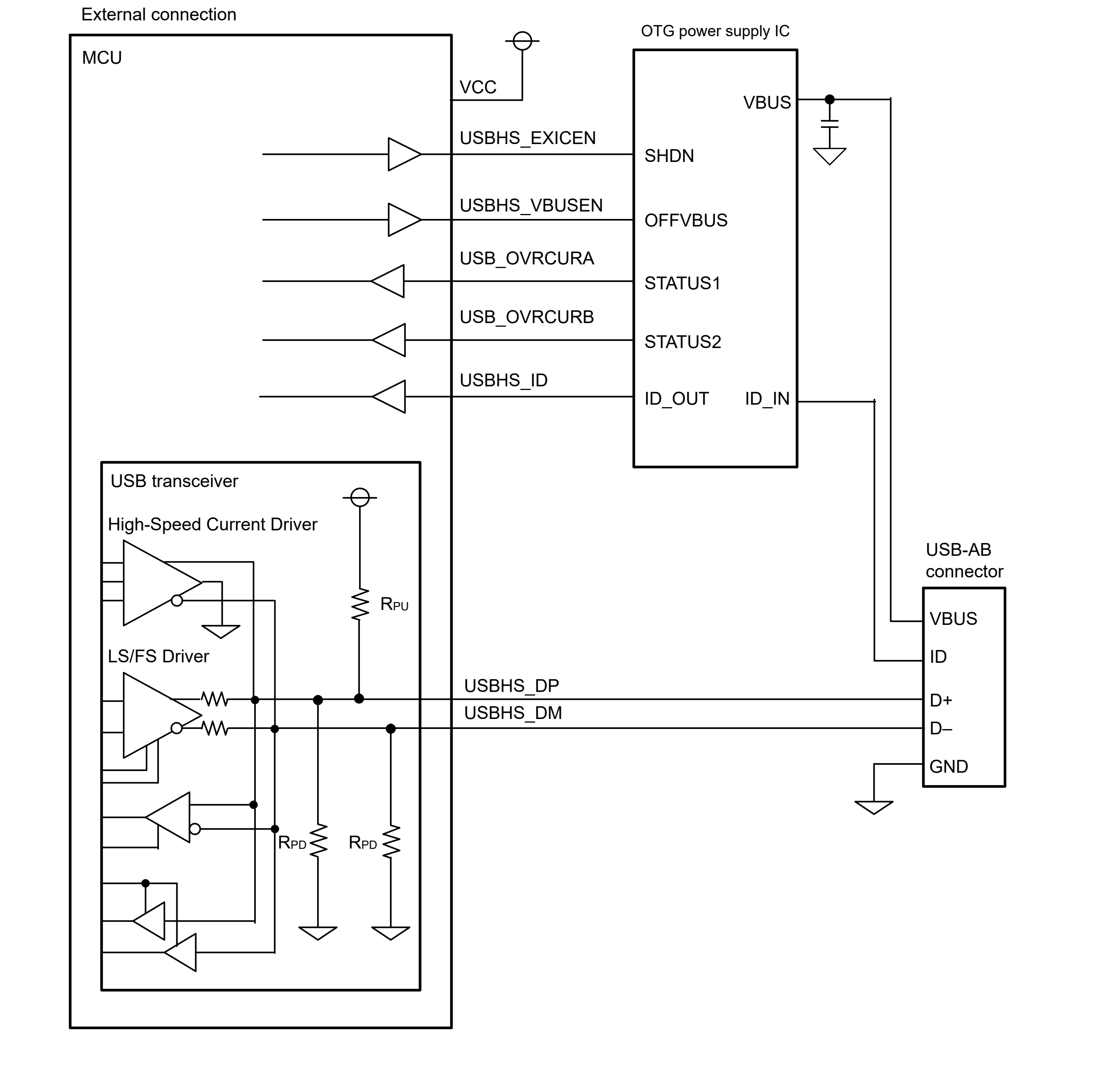
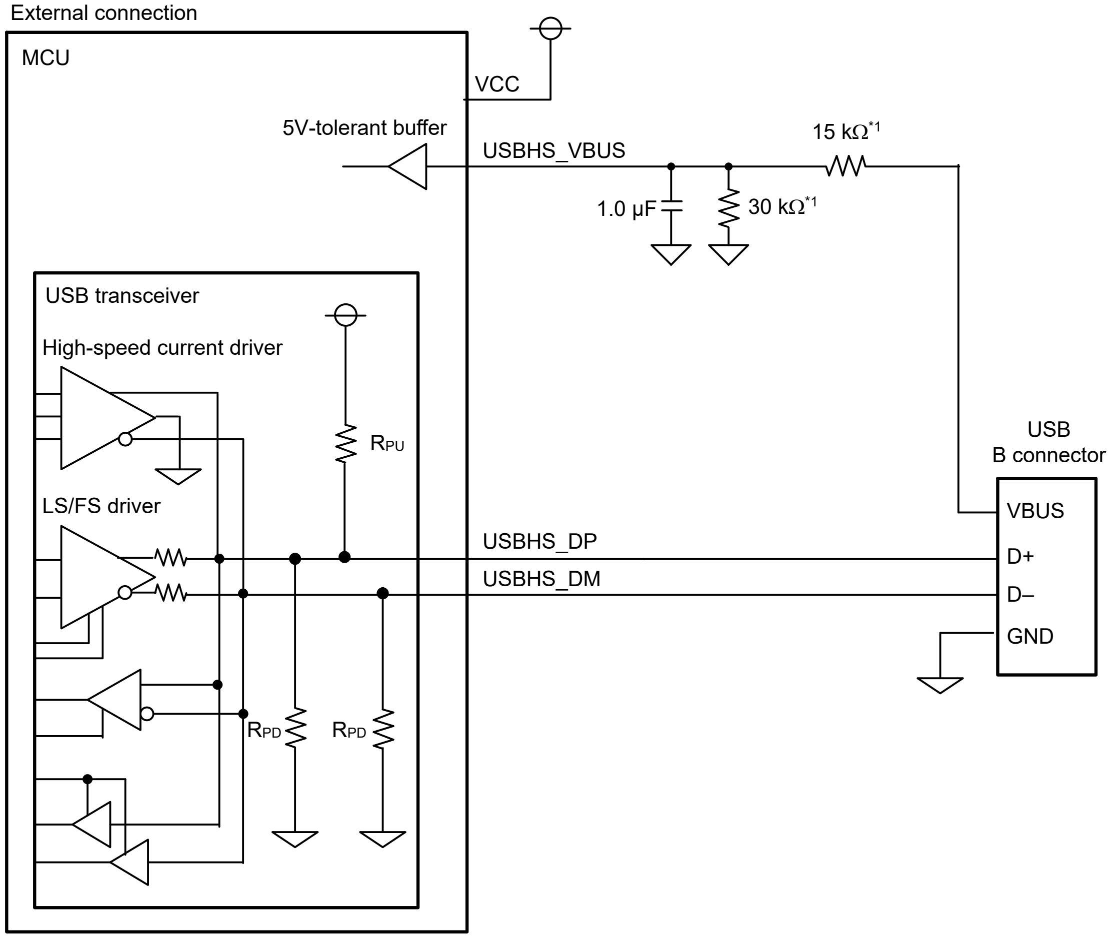


Figure 37.27 Example OTG connection in a self-powered system

Figure 37.28 shows an example USB device connection in a self-powered system.



Note 1. The VBUS (5 V) can be directly connected to the MCU if the VCC power supply of the MCU is not turned off when the USB is connected. If the VCC power supply of the MCU is turned off, the VBUS should be less than 3.6 V.

Figure 37.28 Example device connection in self-powered system

Figure 37.29 shows an example USB device connection in a bus-powered system.

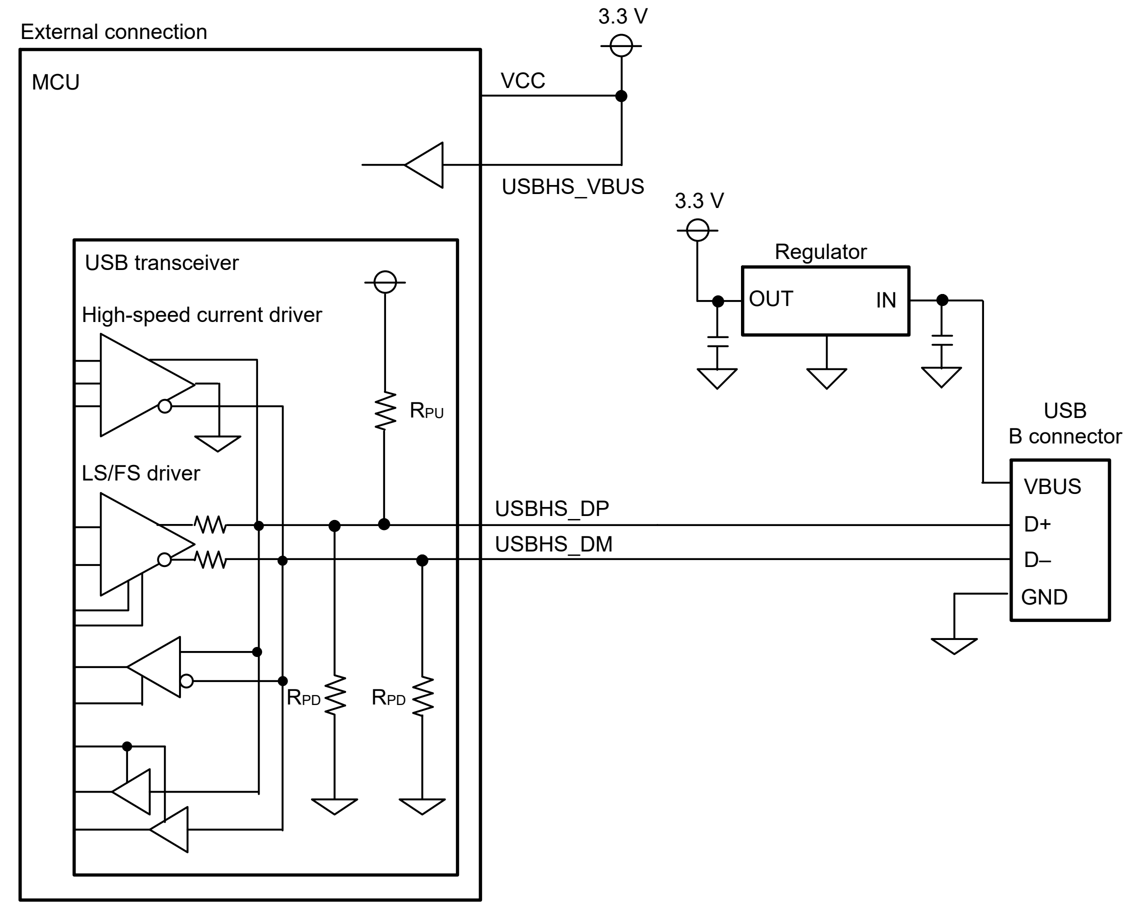


Figure 37.29 Example device connection in a bus-powered system

Figure 37.30 shows an example USB host connection.

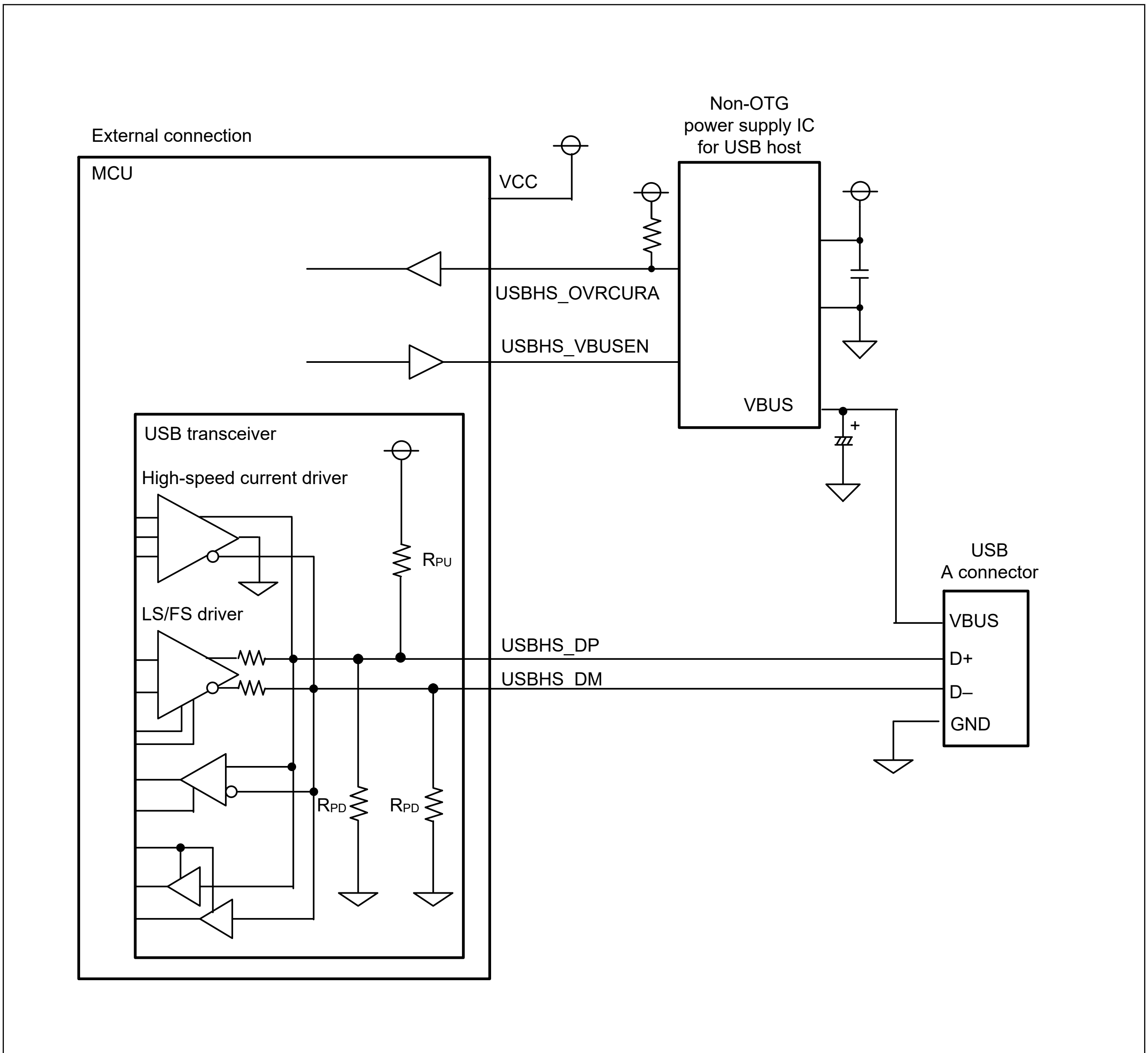


Figure 37.30 Example USB host connection

37.4 Usage Notes

37.4.1 Settings for the Module-Stop Function

USBHS operation can be disabled or enabled using Module Stop Control Register B (MSTPCRB). The USBHS is initially stopped after reset. Releasing the module-stop state enables access to the registers. After releasing module stop, make settings required to activate the PHY circuit, including the input clock frequency setting, and then clear the PHYSET.DIRPD bit to 0. For details, see section 11, Low Power Mode.

37.4.2 Setup for Transitioning to Deep Software Standby Mode 1

Before transitioning to Deep Software Standby mode 1, clear the DVSTCTR0.VBUSEN bit to 0.

37.4.3 Clearing the Interrupt Status Register on Exiting Software Standby Mode

Because the input buffer is always enabled in Software Standby mode, an unexpected interrupt might occur under the following conditions:

- When the interrupt is enabled in Normal mode

- When the interrupt is disabled in Software Standby mode
- When the input level of the pin that cancels Software Standby mode is changed in Software Standby mode

These conditions might cause the associated interrupt flag in the Interrupt Status Register to set unexpectedly. After the MCU exits Software Standby mode, the unexpected interrupt might be sent to the interrupt controller. To avoid this, always clear the INTSTS0 and INTSTS1 registers in the canceling sequence.

37.4.4 Clearing the Interrupt Status Register after Setting Up the Port Function

The input buffer is disabled before the PmnPFS.PSEL and PmnPFS.PMR ports are set up, so the internal signal is fixed high or low. The input buffer is enabled after the port is set so that the external pin state is propagated to the MCU. An unexpected interrupt might occur at this time, causing the VBINT and OVRCR bits in INTSTS0 and INTSTS1, or other interrupt status flags to set to 1. To avoid a malfunction, always clear the INTSTS0 and INTSTS1 registers after setting up the port.

37.4.5 Restriction of Register Access

When accessing the USBHS registers consecutively and writing to the Deep Software Standby-related registers (DPUSR0R, DPUSR1R, DPUSR2R, and DPUSRCR), there are constraints between the Deep Software Standby-related registers and the last accessed USBHS register. At least one of the following two constraints must be performed.

- Set an interval specified in the Table 37.36 between the last accessed USBHS register and write access to the Deep Software Standby-related registers.

Table 37.36 Required interval before write access to Deep Software Standby-related registers

Last accessed USBHS register	Write	Read
Other registers than following	BWAIT[3:0] + 3	BWAIT[3:0] + 3
DPUSR0R, DPUSR1R	130	BWAIT[3:0] + 3
DPUSR2R, DPUSRCR	130	7

Note: Unit: PCLKA cycles

[Examples]

1. Performing read access to other registers, then write access to Deep Software Standby-related registers.
   The required interval is BWAIT[3:0] + 3 cycles or longer.
2. Performing write access to the Deep Software Standby-related registers, then write access to the Deep Software Standby-related registers. The required interval is 130 cycles or longer.

- Before performing write access to the Deep Software Standby-related registers, when the read access to a USBHS register is performed, confirm the read value. When the write access to a USBHS register is performed, read it and confirm the written value.

38. Serial Communications Interface (SCI)

This is the SCI_B version of the SCI peripheral module.

SCI_B is referred to as SCI in this chapter.

38.1 Overview

The Serial Communications Interface (SCI) × 10 channels have asynchronous and synchronous serial interfaces:

- Asynchronous interfaces (UART and Asynchronous Communications Interface Adapter (ACIA))
- 8-bit clock synchronous interface
- Simple IIC (master-only)
- Simple SPI
- Simple LIN
- Smart card interface
- Manchester interface

The smart card interface complies with the ISO/IEC 7816-3 standard for electronic signals and transmission protocol. SCIn (n = 0 to 9) has FIFO buffers to enable continuous and full-duplex communication, and the data transfer speed can be configured independently using an on-chip baud rate generator.

In this section, PCLK refers to PCLKA.

Table 38.1 lists the SCI specifications, Table 38.2 shows the functions of each SCI channel, Figure 38.1 shows a block diagram of SCI, Figure 38.2 shows a clock source selector block diagram, and Table 38.3 lists the I/O pins.

Table 38.1 SCI specifications (1 of 4)

Parameter	Specifications
Number of modules	10 (SCIn (n = 0 to 9))
Serial communication modes	● Asynchronous ● Clock synchronous ● Simple IIC ● Simple SPI ● Simple LIN (SCIn) ● Smart card interface ● Manchester interface (SCIn)
Operation clock (TCLK)	Synchronized clock (PCLK) or independent clock (SCICLK) can be selected
Transfer speed	Bit rate specifiable with the on-chip baud rate generator
Full-duplex communications	● Transmitter: Continuous transmission possible using double-buffering ● Receiver: Continuous reception possible using double-buffering
Half-duplex communications	Half-duplex communication is possible by using only TXDn pins
Data transfer	Selectable as LSB-first or MSB-first transfer
Inverter for communication pins (RXDn, TXDn)	Selectable inverter for each pin (RXDn, TXDn)
Interrupt sources	Transmit end, transmit data empty, receive data full, receive error, receive data ready, address match. Break Field detection/output, Bus collision detection, Active edge detection (SCIn). Completion of generation of a start condition, restart condition, or stop condition (for simple IIC mode).
Loop Back function	Self-diagnosis of communication function by IP internal transmission/reception is possible
Synchronizer Bypass function	Ability to bypass synchronization circuit between bus clock (PCLK) and operation clock (TCLK)
Module-stop function	Module-stop state can be set for each channel
Reception in Software Standby mode	Only SCI0 is available

Table 38.1 SCI specifications (2 of 4)

Parameter		Specifications
Clock synchronous mode	Data length	8 bits
	Adjustment of receive sampling timing	Adjustable receive sampling timing after the default timing in master mode only when using internal clock
	Receive error detection	Overrun error
	Clock source	Selectable to internal clock (master mode) or external clock (slave mode)
	Double-speed mode	Baud rate generator double-speed mode is selectable
	Hardware flow control	Transmission and reception controllable with CTSn_RTSn pins
	Transmission and reception	Selectable to 1-stage register or 16-stage FIFO
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Adjustment of receive sampling timing	Adjustable receive sampling timing before/after the default timing
	Adjustment of transmit timing	Adjustable edge timing of transmit waveform controlled by the setting value of registers.
	Parity	Even parity, odd parity, or no parity
	Receive error detection	● Parity error ● Overrun error ● Framing error
	Hardware flow control	Transmission and reception controllable with CTSn_RTSn pin and CTSn pin
	Transmission and reception	Selectable to 1-stage register or 16-stage FIFO
	Address match	Interrupt request/event output can be issued upon detecting a match between received data and the value in the compare match register
	Start-bit detection	Selectable to low level or falling edge detection
	Break detection	Breaks from framing errors detectable by read from CSR register
	Clock source	Selectable to internal or external clock. Transfer rate clock input from the GPT can be used (SCIn (n = 1, 2)).
	Double-speed mode	Baud rate generator double-speed mode is selectable
	Multi-processor communications function	Serial communication enabled among multiple processors
	RS-485 driver control function	Output DEn signal to enable external transceiver transmit mode
	Noise cancellation	Digital noise filters included on signal paths from the RXDn pin inputs
Smart card interface mode	Error processing	Error signal can be automatically transmitted upon detecting a parity error during reception
		Data can be automatically retransmitted upon receiving an error signal during transmission
	Data type	Both direct and inverse convention supported

Table 38.1 SCI specifications (3 of 4)

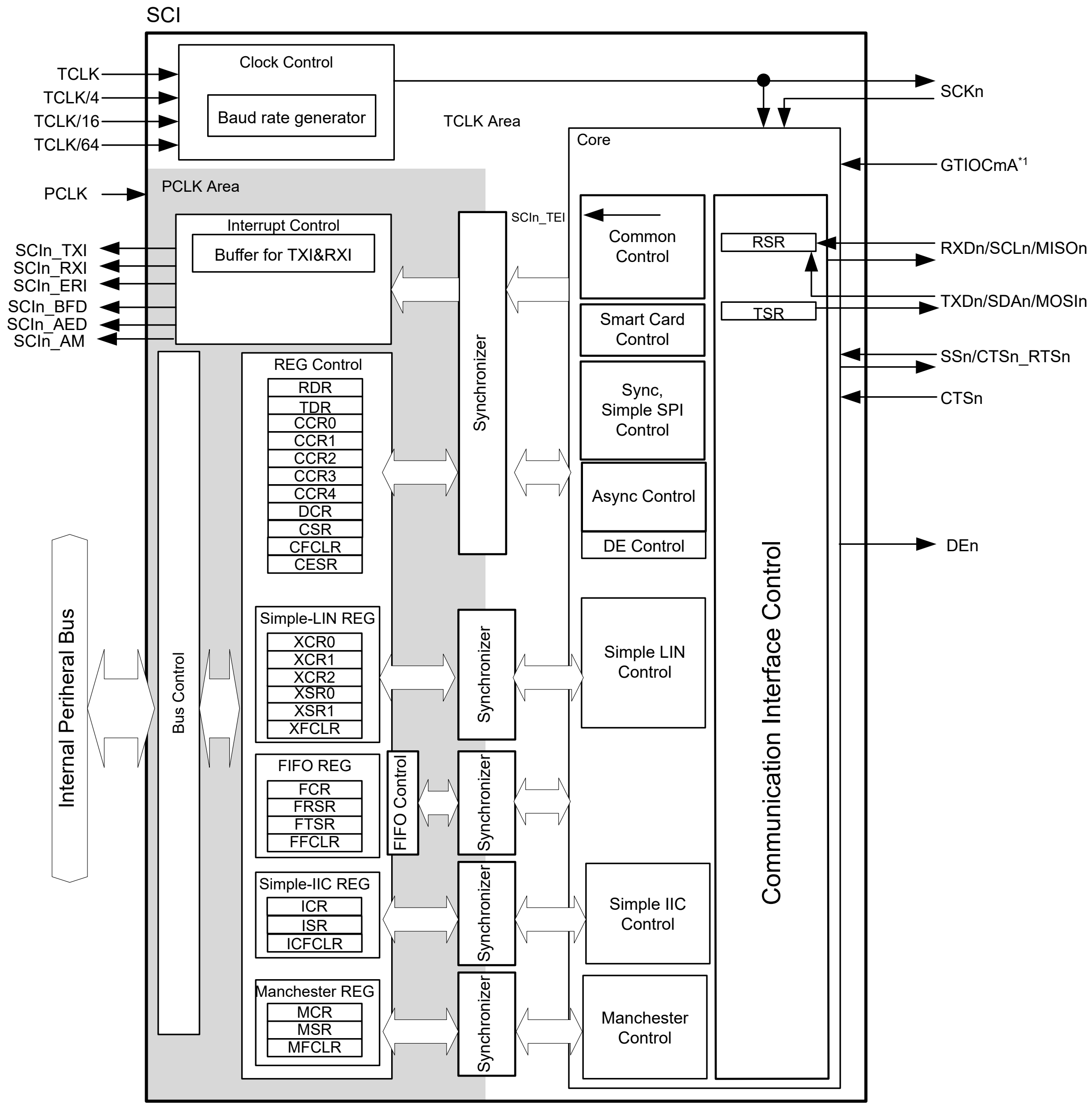
Parameter		Specifications
Manchester mode	Communication format	Manchester code with the preface and the Start Bit added
	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Parity function	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, framing, Manchester errors
	Hardware flow control	Transmission and reception controllable with CTSn_RTSn pin and CTSn pin
	Clock source	Only internal clock can be used.
	Double-speed mode	Baud rate generator double-speed mode is selectable
	Multi-processor communication function	Serial communication among multiple processors
	Manchester encoding/ decoding function	Function to perform Manchester encoding/decoding of transmission/reception data and communicate using Manchester code
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters
	Preface setting/detection function	The function outputs the configured the preface pattern and detects it.
	Start Bit setting/detection function	The function outputs the configured the Start Bit pattern and detects it.
	Reception retiming function	Timing correction is performed for each bit of the received signal
Simple IIC mode	Transfer format	I^2C bus format (MSB-first only)
	Operating mode	Master (single-master operation only)
	Transfer rate	Up to 400 kbps
	Noise cancellation	The signal paths from input on the SCLn and SDAn pins incorporate digital noise filters and provide an adjustable interval for noise cancellation
Simple SPI mode	Data length	8 bits
	Error detection	Overrun error
	Clock source	Selectable to internal clock (master mode) or external clock (slave mode)
	Double-speed mode	Baud rate generator double-speed mode is selectable
	Transmission/Reception	Selectable either 1 stage register or 16-stage FIFO
	Adjustment of receive sampling timing	Adjustable receive sampling timing after the default timing in master mode only when using internal clock
	SSn input pin function	High impedance state can be invoked on the output pins by driving the SSn pin high.
	Clock settings	Configurable among four clock phase and clock polarity settings

Table 38.1 SCI specifications (4 of 4)

Parameter		Specifications
Simple LIN mode	Start Frame Transmission	• Break Field output possible, Break Field output complete interrupt output possible • Bus collision detection possible, bus collision detection interrupt output possible
	Start Frame Reception	• Break Field detectable, Break Field detected interrupt output possible • Control Field 0/1 data comparison function • Control Field 1 can set two types of comparison data of primary and secondary • Priority interrupt bit can be set in Control Field 1 • Handling of Start Frames that do not include a Break Field • Handling of Start Frames that do not include a Control Field 0 • Bit rate measurement function
	Input/Output control function	• Selectable polarity for TXDn and RXDn signals • Selection of a digital filter for the RXDn signal • Half-duplex operation employing RXDn and TXDn signals multiplexed on the same pin • Selectable timing for the sampling of data received through RXDn
Bit rate modulation function		Error reduction through correction of outputs from the on-chip baud rate generator
Event link function		Error event output for receive error or error signal detection (SCIn_ERI) (n = 0 to 9)
		Receive data full event output (SCIn_RXI) (n = 0 to 9)
		Transmit data empty event output (SCIn_TXI) (n = 0 to 9)
		Address match event output (SCIn_AM) (n = 0 to 9)
		Active edge detection event output (SCIn_AED)
		Transmit end event output (SCIn_TEI) (n = 0 to 9)
TrustZone Filter		Security and Privilege attribution can be set for each channels

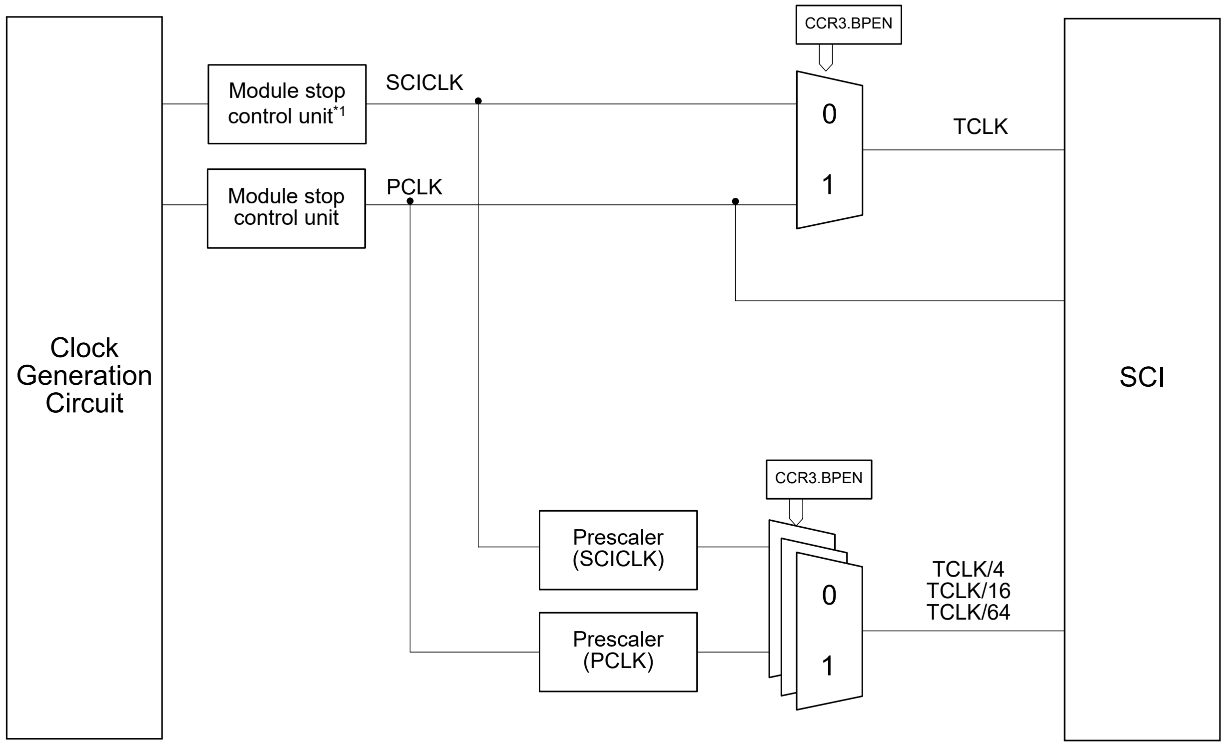
Table 38.2 Functions of SCI channel

Item	SCI0	SCI1	SCI2	SCI3 to SCI9
Asynchronous mode	Available			
Clock synchronous mode	Available			
Smart card interface mode	Available			
Simple I^2C mode	Available			
Simple SPI mode	Available			
FIFO mode	Available (16 stages)			
Address match	Available			
Manchester mode	Available	Available	Available	Available
Simple LIN mode	Available	Available	Available	Available
GPT clock input	Not Available	GTIOC3A GTIOC4A	GTIOC6A GTIOC7A	Not Available



Note: n = 0 to 9
Note: Only SCI0 and SCI1 support Simple LIN and only SCI0 supports Manchester
Note 1. m = 3, 4, 6, 7

Figure 38.1 SCI block diagram



Note 1. If the module-stop state of any SCI channel is canceled, SCICLK is supplied.

Figure 38.2 Clock source selector block diagram

Table 38.3 SCI I/O pins

Function	Pin name	Input/ Output	Description
SCIn (n = 0 to 9)	RXDn/SCLn/MISOn	Input/ Output	SCIn receive data input SCIn I^2C clock input/output SCIn slave transmit data input/output
	TXDn/SDAn/MOSIn	Input/ Output	SCIn transmit data output SCIn I^2C data input/output SCIn master transmit data input/output
	SSn/CTSn_RTSn	Input/ Output	SCIn chip select input, active-low SCIn transfer start control input/output, active-low
	CTSn	Input	SCIn transfer start control input, active-low
	DEn	Output	Driver Enable signal output
	SCKn	Input/ Output	SCIn clock input/output

38.2 Register Descriptions

38.2.1 RSR : Receive Shift Register

RSR is a shift register that receives serial data input from the RXDn pin and converts it into parallel data. When one frame of data is received, the data is automatically transferred to the RDR register. The RSR register cannot be directly accessed by the CPU.

38.2.2 RDR/RDR_BY : Receive Data Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	FER	PER	—	—	ORER	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	FFER	FPER	DR	MPB	RDAT[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	RDAT[8:0]	Serial receive data RDAT is a 9-bit register for storing received data. Received data is stored in [6:0] when 7-bit data is selected, in [7:0] when 8-bit data is selected, and in [8:0] when 9-bit data is selected. And 0 is stored in the unused bit. Use RDR_BY for byte access to RDR[7:0].	R
9	MPB	Multi-processor flag 0: Data transmission cycles 1: ID transmission cycles	R
10	DR	Receive data ready flag FRSR.DR can be read.	R
11	FPER	FIFO parity error flag Valid only in Asynchronous mode 0: There is no parity error in the data read from the receive-FIFO 1: There is parity error in the data read from the receive-FIFO	R
12	FFER	FIFO framing error flag Valid only in Asynchronous mode 0: There is no framing error in the data read from the receive-FIFO 1: There is framing error in the data read from the receive-FIFO	R
23:13	—	These bits are read as 0.	R
24	ORER	Overrun Error flag CSR.ORER can be read.	R
26:25	—	These bits are read as 0.	R
27	PER	Parity error flag CSR.PER can be read.	R
28	FER	Framing error flag CSR.FER can be read.	R
31:29	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

In FIFO mode (CCR3.FM = 1), the RDR is 16-stage FIFO buffer configuration.

When using FIFO mode, use RDR for 32-bit access.

RDAT[8:0] bits (Serial receive data)

After one frame of data is received, the received data is transferred from the RSR register to this register, thus allowing the RSR register to receive the next data.

The RSR and RDR registers have a double-buffered construction to enable continuous reception.

For Non-FIFO mode, read RDR only once when a receive data full interrupt (SCIn_RXI) request is issued. Without reading received data from RDR, if the next one frame is received, an overrun error occurs.

For FIFO mode, continuous reception is executed until 16 stages are stored. If data is read when there is no received data in the receive-FIFO(RDR), the value is undefined. When the receive-FIFO (RDR) are full of received data, subsequent serial receive data is lost.

The CPU cannot write to RDR.

0 is stored in the bit position which isn't received (RDR.bit8 or RDR.bit7) at the time of 7bit or 8bit communication of Asynchronous and Manchester mode.

MPB bit (Multi-processor flag)

In Asynchronous mode and Manchester mode, during multi-processor communication (CCR3.MP = 1), the value of the multi-processor bit corresponding to the received data (RDAT[8:0]) can be read.

FPER bit (FIFO parity error flag)

Indicates whether the data read from the receive-FIFO has a parity error.

0 is stored for non-FIFO mode.

FFER bit (FIFO framing error flag)

Indicates whether the data read from receive-FIFO has a framing error.

0 is stored for non-FIFO mode.

38.2.3 TDR/TDRLL/TDRLH : Transmit Data Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x04 (TDR/TDRLL)
0x05 (TDRLH)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TSYNC	—	—	MPBT	TDAT[8:0]								
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
8:0	TDAT[8:0]	Serial transmit data TDAT is a 9-bit register for setting transmit data. Transmit data is set in [6:0] when 7-bit data is selected, in [7:0] when 8-bit data is selected, and in [8:0] when 9-bit data is selected. When byte access, write TDR [15:8] and then write TDR [7:0]. Use TDRLL for byte access to TDR[7:0], TDRLH for byte access to TDR[15:8].	R/W
9	MPBT	Multi-processor transfer bit flag Value of the multi-processor bit in the transmission frame. This bit is use in Asynchronous and Manchester mode. When writing to this bit when not used, write the initial value. 0: Data transmission cycles 1: ID transmission cycles	R/W
11:10	—	These bits are read as 1. The write value should be 1.	R/W
12	TSYNC	Transmit SYNC data It is valid when MCR.SBSEL = 1 and MCR.SYNSEL = 1 in Manchester mode. When this bit is not used, write the initial value. 0: The Start Bit is transmitted as DATA SYNC. 1: The Start Bit is transmitted as COMMAND SYNC.	R/W
31:13	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-3, P-TYPE-3

In FIFO mode (CCR3.FM = 1), the TDR is 16-stage FIFO buffer configuration.

When using FIFO mode, use TDR for 32-bit access.

TDAT[8:0] bits (Serial transmit data)

The TDR is a 9-bit register for storing transmit data.

When empty space is detected in the TSR register, the transmit data stored in the TDR registers is transferred to TSR, and transmitting is started.

The TSR and TDR registers have a double-buffered construction to realize continuous transmission. When the next data to be transmitted is stored in TDR after one frame of data has been transmitted, the transmitting operation is continued by transfer to the TSR register.

When the SCI detects that the transmit shift register (TSR) is empty, it transmits data written in the transmit-FIFO (TDR) into TSR and starts serial transmission. Continuous serial transmission is executed until there is no transmit data left in the transmit-FIFO (TDR).

For non-FIFO mode, when a transmit data empty interrupt (SCIn_TXI) request is issued and CCR0.TE is 1, write transmit data to the TDR only once.

For FIFO mode, when transmit-FIFO is full of transmit data 16 frames, no more data can be written. If writing of new data is attempted, the data is ignored.

The TDR register can always be read / written from the CPU. And when byte access, write TDR[15:8] and then write TDR[7:0].

MPBT bit (Multi-processor transfer bit flag)

Selects the multi processor bit of transmit frame.

TSYNC bit (Transmit SYNC data)

When Manchester mode and MCR.SBSEL = 1 and MCR.SYNSEL = 1, the type of SYNC selected according to the TSYNC bit becomes the Start Bit of the transmission frame.

38.2.4 TSR : Transmit Shift Register

TSR is a shift register that transmits serial data. To perform serial data transmission, the SCI first automatically transfers transmit data from TDR to TSR, then sends the data to the TXDn pin. The CPU cannot directly access the TSR.

38.2.5 CCR0 : Common Control Register 0

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	SSE	—	—	TEIE	TIE	—	—	—	RIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	IDSEL	DCME	MPIE	—	—	—	TE	—	—	—	RE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RE	Receive Enable 0: Serial reception is disabled 1: Serial reception is enabled	R/W*1 *3
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	TE	Transmit Enable 0: Serial transmission is disabled 1: Serial transmission is enabled	R/W*1

Bit	Symbol	Function	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	MPIE	Multi-Processor Interrupt Enable Valid in Asynchronous mode and Manchester mode when CCR3.MP is 1. This bit should set 0 in Smart card interface mode. 0: Non-multi-processor reception 1: Multi-processor reception When the data with the multi-processor bit set to 0 is received, the data is not read, and setting the status flags to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and non-multi-processor reception is resumed. If you want to continue receiving operation using the multiprocessor function, set this bit to 1 sufficiently earlier than receiving the STOP bit of the next received frame (consider the synchronization delay time).	R/W*2
9	DCME	Data Compare Match Enable Valid only in Asynchronous mode 0: Address match function is disabled 1: Address match function is enabled	R/W*2
10	IDSEL	ID Frame Select Valid only in Asynchronous mode with multi-processor 0: Compare data irrespective of the value of the MPB bit 1: Compare data only when the MPB bit is 1 (ID frame)	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
16	RIE	Receive Interrupt Enable 0: SCIn_RXI and SCIn_ERI interrupt requests are disabled 1: SCIn_RXI and SCIn_ERI interrupt requests are enabled	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	TIE	Transmit Interrupt Enable 0: SCIn_TXI interrupt request is disabled 1: SCIn_TXI interrupt request is enabled	R/W
21	TEIE	Transmit End Interrupt Enable This bit should set 0 in Smart card interface mode. 0: SCIn_TEI interrupt request is disabled 1: SCIn_TEI interrupt request is enabled	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
24	SSE	SSn Pin Function Enable Valid in Simple SPI mode. In slave mode (CCR3.CKE[1:0] = 1x), set 1 to this bit. 0: SSn pin function is disabled 1: SSn pin function is enabled	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. In Clock synchronous mode (CCR3.MOD[2:0] = 010b), Simple SPI mode (CCR3.MOD[2:0] = 011b), and Simple IIC mode (CCR3.MOD[2:0] = 100b), 1 can be written only when TE = 0 and RE = 0. After setting TE or RE to 1, only 0 can be written in TE and RE. In other mode, writing is enabled under any condition.

Note 2. This bit is a bit that is cleared by hardware. Note that writing to a bit other than this bit with a bit manipulation instruction may cause this bit to be unintentionally set to 1 by a read-modify-write operation.

Note 3. In Clock synchronous mode or Simple SPI mode, and internal clock (master mode), receive-only setting is prohibited in the internal clock (master mode) (TE = 0 and RE = 1 setting prohibited).

RE bit (Receive Enable)

Enables or disables serial receive operation.

When the RE bit is set to 1, serial reception becomes possible after the synchronization delay time has elapsed in Asynchronous mode or the synchronous clock input in Clock synchronous mode or the neg-edge of RXDn in Manchester mode or start bit in Smart card interface mode.

Note that CCR3 should be set prior to setting the RE bit to 1 in order to designate the reception format.

Except Smart card interface mode, even if reception is halted by setting the RE bit to 0, the CSR. RDRF, FER, PER, ORER, MSR.MER, SBER, SYER, PFER, FRSR. DR flags are not affected, and the previous values is retained. In Smart card interface mode, even if reception is halted by setting the RE bit to 0, the CSR.FER, PER, ORER flags are not affected and the previous value is retained. Also, to stop the reception operation, synchronization delay time is required from when the RE bit is set to 0 until the reception operation is stopped.

TE bit (Transmit Enable)

Enables or disables serial transmission.

When the TE bit is set to 1, serial transmission becomes possible after the synchronization delay time has elapsed. After the synchronization delay time, transmission is started by writing transmit data to TDR. Note that CCR3 should be set prior to setting the TE bit to 1 in order to designate the transmission format. In addition, the synchronization delay time is required until the transmission control circuit is stopped after the TE bit is set to 0.

MPIE bit (Multi-Processor Interrupt Enable)

When the MPIE bit is set to 1 and data with the multi-processor bit set to 0 is received, the data is not read and setting the status flags (CSR.RDRF, ORER, FER, FRSR, DR, MSR.MER, SYER, PFER, SBER) are disabled.

When data with the multi-processor bit set to 1 is received, the MPIE is automatically cleared to 0, and normal reception is resumed. For details, see section 38.4. Multi-Processor Communication Function. If you want to continue receiving operation using the multi-processor function, set this bit to 1 sufficiently earlier than receiving the STOP bit of the next received frame.

When the receive data includes the MPB bit set to 0, the receive data is not transferred from the RSR to the RDR, a receive error is not detected, and setting the flags ORER, FER, MER, SYER, PFER, and SBER to 1 is disabled.

When the receive data includes the MPB bit set to 1, the MPIE bit is automatically cleared to 0, the SCIn_RXI and SCIn_ERI interrupt requests are enabled (if CCR0.RIE is set to 1), and setting the flags ORER, FER, MER, SYER, PFER, and SBER to 1 is enabled.

MPIE should be set to 0 if multi-processor communications function is not to be used.

DCME bit (Data Compare Match Enable)

The DCME bit selects whether the address match function (data compare match function) is enabled or not.

When DCME is 1, if SCI detects the match to the comparison data (CCR4.CMPD) with receive data, DCME is cleared automatically, and after that, SCI operation mode is in receive mode without data compare match function.

See section 38.3.6. Address Match (Receive Data Match Detection) Function.

The write value should be 0 other than Asynchronous mode.

IDSEL bit (ID Frame Select)

The IDSEL bit selects whether to compare data irrespective of the value of MPB bit or to compare data only when the MPB bit = 1 (ID frame) when the address match function is valid. Set at the same time as DCME.

RIE bit (Receive Interrupt Enable)

Enables or disables SCIn_RXI and SCIn_ERI interrupt requests.

SCIn_RXI and SCIn_ERI interrupt request is disabled by setting the RIE bit to 0.

An SCIn_ERI interrupt request can be canceled by reading 1 from the CSR.ORER, FER, or PER and then setting the flag to 0 or setting the RIE bit to 0.

In Manchester mode, the MER, SYER, PFER and SBER flags are also the cause of SCIn_ERI interrupt request, so the same processing is necessary. For details of these flags, see section 38.2.12. MCR : Manchester Control Register and section 38.2.21. MSR : Manchester Status Register.

TIE bit (Transmit Interrupt Enable)

Enables or disables SCIn_TXI interrupt request.

An SCIn_TXI interrupt request is disabled by setting the TIE bit to 0. At the beginning of transmission, set 1 to CCR0.TE and CCR0.TIE simultaneously. The SCIn_TXI interrupt request is then generated.

TEIE bit (Transmit End Interrupt Enable)

Enables or disables a SCIn_TEI interrupt request. A SCIn_TEI interrupt request is disabled by setting the TEIE bit to 0.

In simple IIC mode, the SCIn_TEI is allocated to the interrupt on completion of issuing a start, restart, or stop condition (STIn). In this case, the TEIE bit can be used to enable or disable the STIn.

SSE bit (SSn Pin Function Enable)

Set the SSE bit to 1 if the SSn pin is to be used in control of transmission and reception (in simple SPI mode).

Set the SSE bit to 0 in any other mode. Do not set both the SSE and CTSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

In the slave mode (CCR3.CKE[1:0] = 10 or 11), SSE should be set 1.

In the master mode (CCR3.CKE[1:0] = 00 or 01) and single-master, the SSn pin on the master side is not required to control reception and transmission, so SSE should be set 0.

38.2.6 CCR1 : Common Control Register 1

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	NFM	NFEN	—	NFCS[2:0]			—	—	—	SHARPS	—	—	—	SPLP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	RINV	TINV	—	—	PM	PE	—	—	SPB2IO	SPB2DT	—	—	CTSPEN	CTSE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
0	CTSE	CTS Enable 0: CTS function is disabled (RTS output function is enabled) 1: CTS function is enabled	R/W
1	CTSPEN	CTS External Pin Enable 0: Alternate setting to use CTS and RTS functions as either 1 pin 1: Dedicated setting for separately using CTS and RTS functions with 2 pins	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	SPB2DT	Serial Port Break Data Select The output level of TXDn pin is selected when CCR0.TE = 0 and SPB2IO = 1.*1 0: When TINV is 0, low level is output in TXDn pin. When TINV is 1, high level is output in TXDn pin. 1: When TINV is 0, high level is output in TXDn pin. When TINV is 1, low level is output in TXDn pin.	R/W
5	SPB2IO	Serial Port Break I/O This bit selects whether the value of SPB2DT is output to TXDn pin when CCR0.TE = 0.*1 0: The value of SPB2DT bit is not output in TXDn pin 1: The value of SPB2DT bit is output in TXDn pin	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	PE	Parity Enable Valid only in Asynchronous mode and Manchester mode. In Smart Card Interface mode, set 1 to this bit. 0: When transmitting: Do not add parity bit When receiving: Do not check parity bit 1: When transmitting: Add parity bit When receiving: Check parity bit	R/W
9	PM	Parity Mode Valid only when the PE bit is 1 0: Selects even parity 1: Selects odd parity	R/W

Bit	Symbol	Function	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	TINV	TXD Invert 0: Transmit data is not inverted and output to TXDn*2 1: Transmit data is inverted and output to TXDn	R/W
13	RINV	RXD Invert 0: Received data from RXDn is not inverted and input*2 1: Received data from RXDn is inverted and input	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	SPLP	Loopback Control This bit can be used for internal clock operation in Asynchronous mode, internal mode operation in Manchester mode, internal clock operation in Clock synchronous mode. 0: Normal mode 1: Loopback mode	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	SHARPS	Half-Duplex Communication Select In Simple IIC mode, Smart Card Interface mode, or Simple SPI mode, this bit should be set to 0. 0: TXDn pin, RXDn pin independent 1: TXDn/RXDn pin combination use (half-duplex communication using TXDn pin)	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
26:24	NFCS[2:0]	Noise Filter Clock Select Valid in Asynchronous mode and Manchester mode, Simple LIN mode, and Simple IIC mode. In Simple IIC mode, 000 setting is prohibited. The on-chip baud rate generator source clock means the clock selected by CCR2.CKS [1:0]. Select for the clock source of the noise filter. 0 0 0: The base clock signal divided by 1 0 0 1: The on-chip baud rate generator source clock divided by 1 0 1 0: The on-chip baud rate generator source clock divided by 2 0 1 1: The on-chip baud rate generator source clock divided by 4 1 0 0: The on-chip baud rate generator source clock divided by 8 others: Setting prohibited	R/W
27	—	These bits are read as 0. The write value should be 0.	R/W
28	NFEN	Digital Noise Filter Function Enable Valid in Asynchronous mode, Manchester mode, Simple LIN mode, and Simple IIC mode 0: In Asynchronous, Manchester, Simple LIN modes: Disable noise cancellation function for RXDn input signal In Simple IIC mode: Disable noise cancellation function for SCLn and SDAn input signals 1: In Asynchronous, Manchester, Simple LIN modes: Enable noise cancellation function for RXDn input signal In Simple IIC mode: Enable noise cancellation function for SCLn and SDAn input signals	R/W
29	NFM	Noise Filter Mode Valid in asynchronous mode and ABCSE2 = 1 0: 3-point matching mode 1: Majority vote mode	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Use this bit in Asynchronous mode and Manchester mode only. Operation in other mode is not guaranteed.
Note 2. RINV/TINV should be set to 0 in Smart card interface mode and Simple IIC mode.

CTSE bit (CTS Enable)

Set the CTSE bit to 1 if the SSn pin is to be used for inputting of the CTSn control signal to control of transmission and reception. The RTSn signal is output when this bit is set to 0. Set this bit to 0 in smart card interface mode, simple SPI mode, Simple LIN mode, and simple IIC mode. Do not set both the CTSE and SSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

CTSPEN bit (CTS External Pin Enable)

When CTSE is 1, select the pins usage method when using the CTS and RTS functions. Set this bit to 1 when assigning the CTS/RTS function to 2 pins and using them at the same time. Set it to 0 except in Asynchronous and Manchester modes.

Table 38.4 shows the relationship between the CTSE bit and CTSPEN bit settings and the functions of the CTSn_RTSn pin and CTSn pin.

Table 38.4 CTSE bit and CTSPEN bit settings and pin functions

CTSE bit	CTSPEN bit	CTSn_RTSn pin	CTSn pin
0	0	RTSn signal output	Not use
1	0	CTSn signal input	Not use
1	1	RTSn signal output	CTSn signal input

Note: Set CTSPEN bit = 0 when CTSE bit = 0.

SPB2DT bit (Serial Port Break Data Select), SPB2IO bit (Serial Port Break I/O)

The TXDn pin status determined by combination of CCR0.TE bit, CCR1.SPB2IO bit, and CCR1.SPB2DT bit is indicated in Table 38.5.

Table 38.5 TXDn pin status

Value of CCR0.TE	Value of CCR1.SPB2IO	Value of CCR1.SPB2DT	TXDn pin status (when TINV is 0)
0	0	—	Hi-Z (initial value)
0	1	0	Low level output
0	1	1	High level output
1	—	—	Serial transmission data is output

Note: —: Don't care

PE bit (Parity Enable)

When PE bit to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception.

In the multi-processor format, the parity bit is not added or checked regardless of the PE bit setting.

PM bit (Parity Mode)

Selects the parity mode for transmission and reception (even or odd). In Multi-processor mode, the PM bit is invalid.

For details on the usage of this bit in Smart card interface mode, see section 38.7.2. Data Format (Except in Block Transfer Mode).

TINV bit (TXD Invert), RINV bit (RXD Invert)

The data of RDR is controlled by RINV and CCR3.SINV. And the data from TXDn pin is controlled by TINV and CCR3.SINV. The control by RINV/TINV are done to communication pins (RXDn / TXDn), so they can control not only data-bits but also other bits (start bit, stop bit, parity bit). For details, see Figure 38.3.

During half-duplex communication and slave operation in Simple SPI mode, use the TXDn pin for reception, so set the inversion control of the received data with the TINV bit.

Note: Description and a timing chart of the IP operation are provided when TINV = 0 and RINV = 0 (when the value of TINV and the value of RINV are not specified).

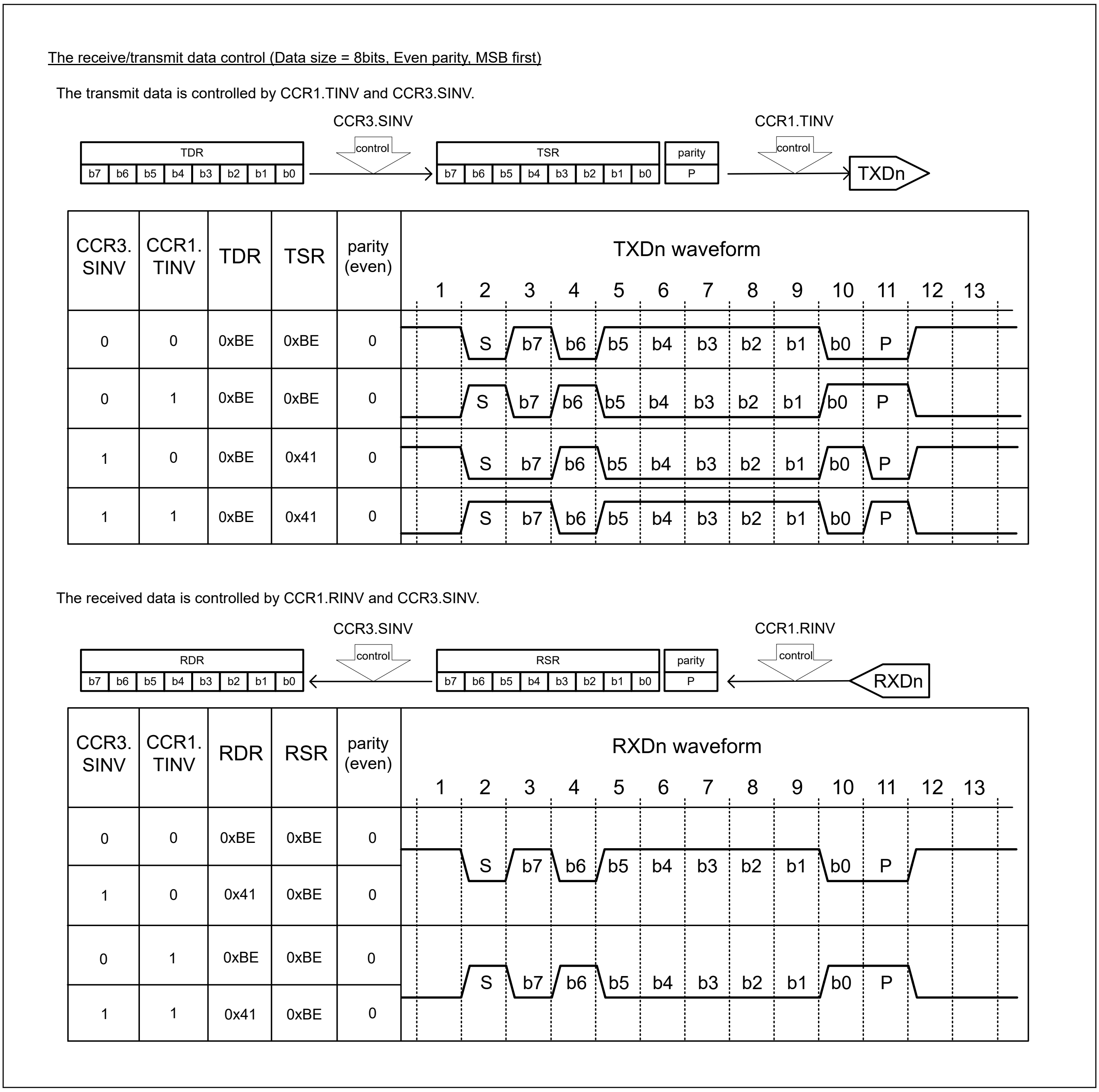


Figure 38.3 Example of receive or transmit data control

SPLP bit (Loopback Control)

When the SPLP bit is 1, SCI blocks the input path from RXDn and connects the output path to TXDn to the reception data register.

Transmit data can be inverted and received by combining it with TINV bit.

Set this bit to 0 for slave operation in Clock synchronous mode, when using an external clock in Asynchronous mode, and when in Simple LIN mode.

SHARPS bit (Half-Duplex Communication Select)

Setting the SHARPS bit to 1 enables half-duplex communication using the TXDn pin. However, the SHARPS bit cannot be used in Simple SPI mode, Simple IIC mode and Smart card interface mode.

If the SHARPS bit is set to 1 and CCR0.TE = 1, CCR0.RE = 0, the TXDn pin becomes the communication output. If this bit is set to 1 and CCR0.TE = 0, CCR0.RE = 1, the TXDn pin becomes the communication input. For details, see section 38.17. Half-Duplex Communication Function.

NFCS[2:0] bits (Noise Filter Clock Select)

The NFCS[2:0] bits select the sampling clock for the digital noise filter.

To use the noise filter in Asynchronous mode, Manchester mode, and Simple LIN mode set the NFCS[2:0] bits from 000b to 100b. In Simple IIC mode, set the bits to a value in the range from 001b to 100b. In particular, if the ABCSE or ABCSE2 bit is set to 1 in asynchronous mode, set it to 000b or 001b.

NFEN bit (Digital Noise Filter Function Enable)

The NFEN bit enables or disables the digital noise filter function. When the function is enabled, noise cancellation is applied to the RXDn input signal in Asynchronous mode, Manchester mode, Simple LIN mode, and noise cancellation is applied to the SDAn and SCLn input signals in Simple IIC mode. In any mode other than above, set the NFEN bit to 0 to disable the digital noise filter function. When the function is disabled, input signals are transferred as is, as internal signals.

NFM bit (Noise Filter Mode)

Selects the operation mode of the digital noise filter function. See xxx. Noise Cancellation Function for details.

38.2.7 CCR2 : Common Control Register 2

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MDDR[7:0]								—	—	CKS[1:0]		—	—	—	BRME
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BRR[7:0]								ABCSE2	ABCSE	ABCS	BGDM	—	BCP[2:0]		
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
2:0	BCP[2:0]	Base Clock Pulse Selects the number of base clock cycles in Smart card interface mode. 0 0 0: 93 clock cycles (S = 93)*1 0 0 1: 128 clock cycles (S = 128)*1 0 1 0: 186 clock cycles (S = 186)*1 0 1 1: 512 clock cycles (S = 512)*1 1 0 0: 32 clock cycles (S = 32)*1 (Initial value) 1 0 1: 64 clock cycles (S = 64)*1 1 1 0: 372 clock cycles (S = 372)*1 1 1 1: 256 clock cycles (S = 256)*1	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	BGDM	Baud Rate Generator Double-Speed Mode Select Valid in Asynchronous/Manchester/Clock-synchronous/Simple SPI mode and when CCR3.CKE[1] = 0. 0: Baud rate generator outputs the clock with single frequency 1: Baud rate generator outputs the clock with doubled frequency	R/W
5	ABCS	Asynchronous Mode Base Clock Select Valid only in Asynchronous mode, Manchester mode, and Simple LIN mode 0: Selects 16 base clock cycles for 1-bit period 1: Selects 8 base clock cycles for 1-bit period	R/W
6	ABCSE	Asynchronous Mode Extended Base Clock Select Valid only in Asynchronous mode and when CCR3.CKE[1] = 0 0: Clock cycles for 1-bit period is determined by combination between CCR2.BGDM and CCR2.ABCS. 1: Baud rate is 6 base clock cycles for 1-bit period and the clock of a double frequency is output from the baud rate generator.	R/W

Bit	Symbol	Function	R/W
7	ABCSE2	Asynchronous Mode Extended Base Clock Select 2 Valid only in asynchronous mode and CCR3.CKE[1] = 0 0: Clock cycles for 1-bit period is decided with combination between CCR2.BGDM and CCR2.ABCS. 1: Baud rate is 4 base clock cycles for 1-bit period and the clock of a double frequency is output from the baud rate generator.	R/W
15:8	BRR[7:0]	Bit Rate Setting BRR is an 8-bit register that adjusts the bit rate.	R/W
16	BRME	Bit Rate Modulation Enable 0: Bit rate modulation function is disabled 1: Bit rate modulation function is enabled	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
21:20	CKS[1:0]	Clock Select 0 0: TCLK clock (n = 0)*2 0 1: TCLK/4 clock (n = 1)*2 1 0: TCLK/16 clock (n = 2)*2 1 1: TCLK/64 clock (n = 3)*2	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
31:24	MDDR[7:0]	Modulation Duty Setting MDDR corrects the bit rate adjusted by the BRR[7:0] bits.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. S is the value of S in BRR[7:0] bits explanation.
Note 2. n is the decimal notation of the value of n in BRR[7:0] bits explanation.

BCP[2:0] bits (Base Clock Pulse)

The BCP[2:0] bits select the number of base clock cycles in a 1-bit data transfer time in smart card interface mode.

For details, see section 38.7.4. Receive Data Sampling Timing and Reception Margin.

BGDM bit (Baud Rate Generator Double-Speed Mode Select)

The BGDM bit is valid when the on-chip baud rate generator is selected as the clock source (CCR3.CKE[1] = 0) in Asynchronous mode, Manchester mode, Clock synchronous mode, Simple SPI mode. When external clock is selected (CCR3.CKE[1] = 1), set it to 0. For the clock output from the baud rate generator, either single or doubled frequency can be selected. The base clock is generated by the clock output from the baud rate generator. When the BGDM bit is set to 1, the base clock cycle is halved, and the bit rate is doubled.

Set this bit to 0 in modes other than Asynchronous mode, Manchester mode, Clock synchronous mode, or Simple SPI mode.

ABCS bit (Asynchronous Mode Base Clock Select)

Selects the clock cycles for 1-bit period.

Set the ABCS bit to 0 in modes other than Asynchronous mode, Manchester mode, or Simple LIN mode.

ABCSE bit (Asynchronous Mode Extended Base Clock Select)

The pulse number for a base clock at 1-bit period is 6 and the clock of a double frequency is output from baud rate generator. Only when the bit rate is set to 6 dividing frequency of the bus clock can this bit be used to set CCR2.CKS [1:0] = 00b and BRR[7:0] = 0x00.

Set it to 0 in modes other than Asynchronous mode. Even in Asynchronous mode, set it to 0 when using external clock.

ABCSE2 bit (Asynchronous Mode Extended Base Clock Select 2)

The pulse number for a base clock at 1-bit period is 4 and the clock of a double frequency is output from baud rate generator. Only when the bit rate is set to 4 dividing frequency of the bus clock, please use this bit and set CCR2.CKS [1:0] = 00 and BRR[7:0] = 0.

Set it to 0 in modes other than asynchronous mode. Even in asynchronous mode, set it to 0 when using external clock.

Table 38.6 Base clock cycle number per 1-bit

ABCSE2	ABCSE	ABCS	BGDM	The base clock cycles /1bit	The frequency of the baud rate generator
0	0	0	0	16	×1
0	0	0	1	16	×2
0	0	1	0	8	×1
0	0	1	1	8	×2
0	1	— (don't care)	— (don't care)	6	×2
1	0	— (don't care)	— (don't care)	4	×2

BRR[7:0] bit (Bit Rate Setting)

BRR is an 8-bit register that adjusts the bit rate.

SCI has independent baud rate generator control, different bit rates can be set for each. Table 38.7 shows the relation-ship between the setting (N) in the BRR and the bit rate (B) for Asynchronous mode, multiprocessor transfer, Manchester mode, clock synchronous mode, smart card interface mode, simple SPI mode, and simple IIC mode.

Table 38.7 Relationship between N Setting in BRR and Bit Rate B

Mode	CCR2 settings				BRR[7:0] setting	Error*4
	BGDM bit	ABCS bit	ABCSE bit	ABCSE 2 bit		
Asynchronous, multi-processor, Manchester*2, Simple-LIN*3	0	0	0	0	$N = \frac{TCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$\text{Error}(\%) = \left\{\frac{TCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1\right\} \times 100$
	1	0	0	0	$N = \frac{TCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$\text{Error}(\%) = \left\{\frac{TCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1\right\} \times 100$
	0	1	0	0		
	1	1	0	0	$N = \frac{TCLK \times 10^6}{16 \times 2^{2n-1} \times B} - 1$	$\text{Error}(\%) = \left\{\frac{TCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (N+1)} - 1\right\} \times 100$
	Don't care	Don't care	1	0	$N = \frac{TCLK \times 10^6}{12 \times 2^{2n-1} \times B} - 1$	$\text{Error}(\%) = \left\{\frac{TCLK \times 10^6}{B \times 12 \times 2^{2n-1} \times (N+1)} - 1\right\} \times 100$
	Don't care	Don't care	0	1	N = 0	—
Clock synchronous, simple SPI	0	0 (Initial value)	0 (Initial value)	0 (Initial value)	$N = \frac{TCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	—
	1	0 (Initial value)	0 (Initial value)	0 (Initial value)	$N = \frac{TCLK \times 10^6}{4 \times 2^{2n-1} \times B} - 1$	—
Smart card interface					$N = \frac{TCLK \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$\text{Error}(\%) = \left\{\frac{TCLK \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1\right\} \times 100$
Simple IIC*1					$N = \frac{TCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	—

Note: B: Bit rate (bps)
N: BRR setting for on-chip baud rate generator (0 ≤ N ≤ 255)
TCLK: Operating frequency (MHz)
n and S: Determined by the settings of the CCR2 registers as listed in the Table 38.9 and Table 38.10. Please be careful about $2^{(2n+1)}$ is used in the expression for Smart card interface, $2^{(2n-1)}$ is used in other mode.

Note 1. Adjust the bit rate so that the widths at high and low level of the SCL output in simple IIC mode satisfy the IIC standard.
Note 2. In Manchester mode, only ABCSE = 0 and ABCSE2 = 0 can be selected.
Note 3. In Simple-LIN mode, BGDM = 0, ABCSE = 0 and ABCSE2 = 0 can be selected.

Note 4. Error means that if the N value calculated by the above formula is not an integer for the baud rate value B to be set, the BRR is set to an approximate integer value in which case it represents the error between the baud rate value and B.

Table 38.8 Calculating widths at high and low Level for SCL

Mode	SCLn	Formula (result in seconds)
IIC	Width at high level (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 7 \times \frac{1}{TCLK \times 10^6}$
	Width at low level (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 8 \times \frac{1}{TCLK \times 10^6}$

Table 38.9 Clock source settings

CCR2 setting		
CKS[1:0] bit	Clock source	n
00	TCLK clock	0
01	TCLK/4 clock	1
10	TCLK/16 clock	2
11	TCLK/64 clock	3

Table 38.10 Base clock settings in Smart card interface mode

CCR2 setting		
BCP[2:0] setting	Base clock cycles for 1-bit period	S
0 0 0	93 clock cycles	93
0 0 1	128 clock cycles	128
0 1 0	186 clock cycles	186
0 1 1	512 clock cycles	512
1 0 0	32 clock cycles	32
1 0 1	64 clock cycles	64
1 1 0	372 clock cycles	372
1 1 1	256 clock cycles	256

Table 38.11 and Table 38.12 list examples of N settings in BRR in Asynchronous mode and Manchester mode. Table 38.13 lists the maximum bit rate settable for each operating frequency. Examples of BRR (N) settings in Clock synchronous mode and Simple SPI mode are listed in Table 38.16. Examples of BRR (N) settings in Smart card interface mode are listed in Table 38.18. Examples of BRR (N) settings in Simple IIC mode are listed in Table 38.20. In Smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, see section 38.7.4. Receive Data Sampling Timing and Reception Margin. Table 38.14 and Table 38.17 list the maximum bit rates with external clock input. Table 38.15 lists the maximum bit rates with GPT clock input.

When either the Asynchronous mode base clock select bit (ABCS) or the baud rate generator double-speed mode select bit (BGDM) is set to 1 in Asynchronous mode and Manchester mode, the bit rate becomes twice that listed in Table 38.11 and Table 38.12. When both of those registers are set to 1, the bit rate becomes four times the listed value.

Table 38.11 Examples of BRR Settings for various Bit Rates (Asynchronous Mode and Manchester Mode) (1) (1 of 3)

Bit rate (bps)	Operating Frequency TCLK (MHz)														
	8			9.8304			10			12			12.288		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	103	0.16	2	127	0	2	129	0.16	2	155	0.16	2	159	0
300	1	207	0.16	1	255	0	2	64	0.16	2	77	0.16	2	79	0
600	1	103	0.16	1	127	0	1	129	0.16	1	155	0.16	1	159	0

Table 38.11 Examples of BRR Settings for various Bit Rates (Asynchronous Mode and Manchester Mode) (1) (2 of 3)

Bit rate (bps)	Operating Frequency TCLK (MHz)														
	8			9.8304			10			12			12.288		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
1200	0	207	0.16	0	255	0	1	64	0.16	1	77	0.16	1	79	0
2400	0	103	0.16	0	127	0	0	129	0.16	0	155	0.16	0	159	0
4800	0	51	0.16	0	63	0	0	64	0.16	0	77	0.16	0	79	0
9600	0	25	0.16	0	31	0	0	32	–1.36	0	38	0.16	0	39	0
19200	0	12	0.16	0	15	0	0	15	1.73	0	19	–2.34	0	19	0
31250	0	7	0	0	9	–1.7	0	9	0	0	11	0	0	11	2.4
38400	—	—	—	0	7	0	0	7	1.73	0	9	–2.34	0	9	0

Table 38.11 Examples of BRR Settings for various Bit Rates (Asynchronous Mode and Manchester Mode) (1) (3 of 3)

Bit rate (bps)	Operating Frequency TCLK (MHz)														
	14			16			17.2032			18			19.6608		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	–0.17	3	70	0.03	3	75	0.48	3	79	–0.12	3	86	0.31
150	2	181	0.16	2	207	0.16	2	223	0	2	233	0.16	2	255	0
300	2	90	0.16	2	103	0.16	2	111	0	2	116	0.16	2	127	0
600	1	181	0.16	1	207	0.16	1	223	0	1	233	0.16	1	255	0
1200	1	90	0.16	1	103	0.16	1	111	0	1	116	0.16	1	127	0
2400	0	181	0.16	0	207	0.16	0	223	0	0	233	0.16	0	255	0
4800	0	90	0.16	0	103	0.16	0	111	0	0	116	0.16	0	127	0
9600	0	45	–0.93	0	51	0.16	0	55	0	0	58	–0.69	0	63	0
19200	0	22	–0.93	0	25	0.16	0	27	0	0	28	1.02	0	31	0
31250	0	13	0	0	15	0	0	16	1.2	0	17	0	0	19	–1.7
38400	—	—	—	0	12	0.16	0	13	0	0	14	–2.34	0	15	0

Note: This is an example when the CCR2.ABCS = 0, CCR2.BGDM = 0 and CCR2.ABCSE = 0.
When either ABCS bit or BGDM bit is set to 1, the bit rate doubles.
When both ABCS = 1 and BGDM = 1, the bit rate increases four times.

Table 38.12 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode and Manchester Mode) (2) (1 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)														
	20			25			30			33			40		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	88	–0.25	3	110	–0.02	3	132	0.13	3	145	0.33	3	177	–0.25
150	3	64	0.16	3	80	0.47	3	97	–0.35	3	106	0.39	3	129	0.16
300	2	129	0.16	2	162	–0.15	2	194	0.16	2	214	–0.07	3	64	0.16
600	2	64	0.16	2	80	0.47	2	97	–0.35	2	106	0.39	2	129	0.16
1200	1	129	0.16	1	162	–0.15	1	194	0.16	1	214	–0.07	2	64	0.16
2400	1	64	0.16	1	80	0.47	1	97	–0.35	1	106	0.39	1	129	0.16
4800	0	129	0.16	0	162	–0.15	0	194	0.16	0	214	–0.07	1	64	0.16
9600	0	64	0.16	0	80	0.47	0	97	–0.35	0	106	0.39	0	129	0.16
19200	0	32	–1.36	0	40	–0.76	0	48	–0.35	0	53	–0.54	0	64	0.16

Table 38.12 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode and Manchester Mode) (2) (2 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)														
	20			25			30			33			40		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
31250	0	19	0	0	24	0	0	29	0	0	32	0	0	39	0
38400	0	15	1.73	0	19	1.73	0	23	1.73	0	26	–0.54	0	32	–1.36

Table 38.12 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode and Manchester Mode) (2) (3 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)														
	50			60			100			120			125		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	221	–0.02	—	—	—	—	—	—	—	—	—	—	—	—
150	3	162	–0.15	3	194	0.16	—	—	—	—	—	—	—	—	—
300	3	80	0.47	3	97	–0.35	3	162	–0.15	3	194	0.16	3	202	0.22
600	2	162	–0.15	2	194	0.16	3	80	0.47	3	97	–0.35	3	101	-0.27
1200	2	80	0.47	2	97	–0.35	2	162	–0.15	2	194	0.16	2	202	0.22
2400	1	162	–0.15	1	194	0.16	2	80	0.47	2	97	–0.35	2	101	-0.27
4800	1	80	0.47	1	97	–0.35	1	162	–0.15	1	194	0.16	1	202	0.22
9600	0	162	–0.15	0	194	0.16	1	80	0.47	1	97	–0.35	1	101	-0.27
19200	0	80	0.47	0	97	–0.35	0	162	–0.15	0	194	0.16	0	202	0.22
31250	0	49	0	0	59	0	1	24	0	0	119	0	0	124	0.00
38400	0	40	–0.76	0	48	–0.35	0	80	0.47	0	97	–0.35	0	101	-0.27

Note: This is an example when the CCR2.ABCS = 0, CCR2.BGDM = 0, CCR2.ABCSE = 0 and CCR2.ABCSE2 = 0.
When either ABCS bit or BGDM bit is set to 1, the bit rate doubles.
When both ABCS = 1 and BGDM = 1, the bit rate increases four times.

Table 38.13 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode and Manchester mode) (1 of 3)

TCLK (MHz)	CCR2 settings				Maximum bit rate (bps)	TCLK (MHz)	CCR2 settings				Maximum bit rate (bps)
	BGDM	ABCS	ABCSE	ABCSE2			BGDM	ABCS	ABCSE	ABCSE2	
8	0	0	0	0	250000	16	0	0	0	0	500000
		1	0	0	500000			1	0	0	1000000
	1	0	0	0			1	0	0	0	
		1	0	0	1000000			1	0	0	2000000
	Don't care	Don't care	1	0	1333333		Don't care	Don't care	1	0	2666666
	Don't care	Don't care	0	1	2000000		Don't care	Don't care	0	1	4000000
9.8304	0	0	0	0	307200	17.2032	0	0	0	0	537600
		1	0	0	614400			1	0	0	1075200
	1	0	0	0			1	0	0	0	
		1	0	0	1228800			1	0	0	2150400
	Don't care	Don't care	1	0	1638400		Don't care	Don't care	1	0	2867200
	Don't care	Don't care	0	1	2457600		Don't care	Don't care	0	1	4300800

Table 38.13 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode and Manchester mode) (2 of 3)

TCLK (MHz)	CCR2 settings				Maximum bit rate (bps)	TCLK (MHz)	CCR2 settings				Maximum bit rate (bps)
	BGDM	ABCS	ABCSE	ABCSE2			BGDM	ABCS	ABCSE	ABCSE2	
10	0	0	0	0	312500	18	0	0	0	0	562500
		1	0	0	625000			1	0	0	1125000
	1	0	0	0			1	0	0	0	
		1	0	0	1250000			1	0	0	2250000
	Don't care	Don't care	1	0	1666666		Don't care	Don't care	1	0	3000000
	Don't care	Don't care	0	1	2500000		Don't care	Don't care	0	1	4500000
12	0	0	0	0	375000	19.6608	0	0	0	0	614400
		1	0	0	750000			1	0	0	1228800
	1	0	0	0			1	0	0	0	
		1	0	0	1500000			1	0	0	2457600
	Don't care	Don't care	1	0	2000000		Don't care	Don't care	1	0	3276800
	Don't care	Don't care	0	1	3000000		Don't care	Don't care	0	1	4915200
12.288	0	0	0	0	384000	20	0	0	0	0	625000
		1	0	0	768000			1	0	0	1250000
	1	0	0	0			1	0	0	0	
		1	0	0	1536000			1	0	0	2500000
	Don't care	Don't care	1	0	2048000		Don't care	Don't care	1	0	3333333
	Don't care	Don't care	0	1	3072000		Don't care	Don't care	0	1	5000000
14	0	0	0	0	437500	25	0	0	0	0	781250
		1	0	0	875000			1	0	0	1562500
	1	0	0	0			1	0	0	0	
		1	0	0	1750000			1	0	0	3125000
	Don't care	Don't care	1	0	2333333		Don't care	Don't care	1	0	4166666
	Don't care	Don't care	0	1	3500000		Don't care	Don't care	0	1	6250000
30	0	0	0	0	937500	50	0	0	0	0	1562500
		1	0	0	1875000			1	0	0	3125000
	1	0	0	0			1	0	0	0	
		1	0	0	3750000			1	0	0	6250000
	Don't care	Don't care	1	0	5000000		Don't care	Don't care	1	0	8333333
	Don't care	Don't care	0	1	7500000		Don't care	Don't care	0	1	12500000

Table 38.13 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode and Manchester mode) (3 of 3)

TCLK (MHz)	CCR2 settings				Maximum bit rate (bps)	TCLK (MHz)	CCR2 settings				Maximum bit rate (bps)
	BGDM	ABCS	ABCSE	ABCSE2			BGDM	ABCS	ABCSE	ABCSE2	
33	0	0	0	0	1031250	60	0	0	0	0	1875000
		1	0	0	2062500			1	0	0	3750000
	1	0	0	0			1	0	0	0	
		1	0	0	4125000			1	0	0	7500000
	Don't care	Don't care	1	0	5500000		Don't care	Don't care	1	0	10000000
	Don't care	Don't care	0	1	8250000		Don't care	Don't care	0	1	15000000
40	0	0	0	0	1250000	120	0	0	0	0	3750000
		1	0	0	2500000			1	0	0	7500000
	1	0	0	0			1	0	0	0	
		1	0	0	5000000			1	0	0	15000000
	Don't care	Don't care	1	0	6666666		Don't care	Don't care	1	0	20000000
	Don't care	Don't care	0	1	10000000		Don't care	Don't care	0	1	30000000
						125	0	0	0	0	3906250
								1	0	0	7812500
							1	0	0	0	
								1	0	0	15625000
							Don't care	Don't care	1	0	20833333
							Don't care	Don't care	0	1	31250000

Note: For maximum bit rate, the CKS and BRR register values should be n = 0 and N = 0, respectively. In Manchester mode, use ABCSE = ABCSE2 = 0.

Table 38.14 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

TCLK (MHz)	External clock (MHz)	Maximum bit rate (bps)		TCLK (MHz)	External clock (MHz)	Maximum bit rate (bps)	
		CCR2.ABCS bit = 0	CCR2.ABCS bit = 1			CCR2.ABCS bit = 0	CCR2.ABCS bit = 1
8	2	125000	250000	25	6.25	390625	781250
9.8304	2.4576	153600	307200	30	7.5	468750	937500
10	2.5	156250	312500	33	8.25	515625	1031250
12	3	187500	375000	40	10	625000	1250000
12.288	3.072	192000	384000	50	12.5	781250	1562500
14	3.5	218750	437500	60	15	937500	1875000
16	4	250000	500000	120	30	1875000	3750000
17.2032	4.3008	268800	537600	125	31.25	1953125	3906250
18	4.5	281250	562500				
19.6608	4.9152	307200	614400				
20	5	312500	625000				

Table 38.15 Maximum Bit Rate with GPT Clock Input (Asynchronous Mode)

TCLK (MHz)	GPT clock (MHz)	Maximum bit rate (bps)		TCLK (MHz)	GPT clock (MHz)	Maximum bit rate (bps)	
		CCR2.ABCS bit = 0	CCR2.ABCS bit = 1			CCR2.ABCS bit = 0	CCR2.ABCS bit = 1
8	4	250000	500000	25	12.5	781250	1562500
9.8304	4.9152	307200	614400	30	15	937500	1875000
10	5.0	312500	625000	33	16.5	1031250	2062500
12	6	375000	750000	40	20	1250000	2500000
12.288	6.144	384000	768000	50	25	1562500	3125000
14	7.0	437500	875000	60	30	1875000	3750000
16	8	500000	1000000	120	60	3750000	7500000
17.2032	8.6016	537600	1075200	125	62.5	3906250	781250
18	9.0	562500	1125000				
19.6608	9.8304	614400	1228800				
20	10	625000	1250000				

Table 38.16 BRR Settings for Various Bit Rates (Clock Synchronous Mode, Simple SPI Mode)

Bit rate (bps)	Operating frequency TCLK (MHz)																	
	8			10			30			60			120			125		
	BGDM	n	N	BGDM	n	N	BGDM	n	N	BGDM	n	N	BGDM	n	N	BGDM	n	N
250	0	3	124	0	3	177	—	—	—	—	—	—	—	—	—	—	—	—
500	0	2	249	0	3	77	0	3	233	—	—	—	—	—	—	—	—	—
1k	0	2	124	0	3	38	0	3	116	0	3	233	—	—	—	—	—	—
2.5k	0	2	49	0	1	249	0	3	46	0	3	93	0	3	187	(0)	(3)	(194)
5k	0	2	24	0	1	124	0	2	93	0	3	46	0	3	93	(1)	(3)	(194)
10k	0	1	49	0	0	249	0	2	46	0	2	93	0	3	46	(0)	(2)	(194)
25k	0	2	4	0	1	24	0	1	74	0	1	149	0	2	74	(0)	(2)	(77)
50k	0	1	9	0	0	49	0	0	149	0	1	74	0	1	149	(0)	(2)	(38)
100k	0	1	4	0	0	24	0	0	74	0	0	149	0	1	74	(0)	(1)	(77)
250k	0	1	1	0	0	9	0	0	29	0	1	14	0	1	29	0	0	124
500k	0	1	0	0	0	4	0	0	14	0	0	29	0	1	14	1	0	124
1M	0	0	1	1	0	4	1	0	14	0	0	14	0	0	29	(1)	(0)	(62)
2.5M	—	—	—	0	0	0	0	0	2	0	0	5	0	1	2	1	0	24
5M	—	—	—	1	0	0	1	0	2	0	0	2	0	0	5	(1)	(0)	(12)
7.5M	—	—	—	—	—	—	0	0	0	0	0	1	1	1	0	(0)	(1)	(0)
60M	—	—	—	—	—	—	—	—	—	—	—	—	1	0	0	(1)	(0)	(0)

Note: —: Can be set, but an error over 10% will occur.

Table 38.17 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode, Simple SPI Mode) (1 of 2)

TCLK (MHz)	External clock (MHz)	MAX Bit rate (Mbps)	TCLK (MHz)	External clock (MHz)	MAX Bit rate (Mbps)
8	4	4	25	12.5	12.5
10	5	5	30	15	15
12	6	6	33	16.5	16.5
14	7	7	40	20	20

Table 38.17 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode, Simple SPI Mode) (2 of 2)

TCLK (MHz)	External clock (MHz)	MAX Bit rate (Mbps)	TCLK (MHz)	External clock (MHz)	MAX Bit rate (Mbps)
16	8	8	50	25	25
18	9	9	60	30	30
20	10	10	120	60	60
			125	62.5	62.5

Table 38.18 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372) (1 of 4)

Bit rate (bps)	Operating frequency TCLK (MHz)											
	7.1424			10.00			10.7136			13.00		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	0	0.00	0	1	–30	0	1	–25	0	1	–8.99

Table 38.18 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372) (2 of 4)

Bit rate (bps)	Operating frequency TCLK (MHz)											
	14.2848			16.00			18.00			20.00		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	1	0.00	0	1	12.01	0	2	–15.99	0	2	–6.66

Table 38.18 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372) (3 of 4)

Bit rate (bps)	Operating frequency TCLK (MHz)											
	25.00			30.00			33.00			40.00		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	3	–12.49	0	3	5.01	0	4	–7.59	0	5	–6.66

Table 38.18 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372) (4 of 4)

Bit rate (bps)	Operating frequency TCLK (MHz)											
	50.00			60.00			120.00			125		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	6	0.01	0	7	5.01	0	16	–1.17	0	17	-2.71

Table 38.19 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 32)

TCLK (MHz)	MAX Bit rate (bps)	n	N	TCLK (MHz)	MAX Bit rate (bps)	n	N
10	156250	0	0	30	468750	0	0
10.7136	167400	0	0	33	515625	0	0
13	203125	0	0	40	625000	0	0
16	250000	0	0	50	781250	0	0
18	281250	0	0	60	937500	0	0
20	312500	0	0	120	1875000	0	0
25	390625	0	0	125	1953125	0	0

Table 38.20 BRR Settings for Various Bit Rates (Simple IIC Mode) (1 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)											
	8			10			16			20		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10k	0	24	0	0	31	–2.3	1	12	–3.8	1	15	–2.3
25k	0	9	0	0	12	–3.8	1	4	0	1	6	–10.7
50k	0	4	0	0	6	–10.7	1	2	–16.7	1	3	–21.9
100k	0	2	–16.7	0	3	–21.9	0	4	0	0	6	–10.7
250k	0	0	0	0	1	–37.5	0	1	0	0	2	–16.7
350k										0	1	–10.7
400k										0	1	–21.9

Table 38.20 BRR Settings for Various Bit Rates (Simple IIC Mode) (2 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)											
	25			30			33			40		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10k	1	19	–2.3	1	23	–2.3	1	25	–0.8	0	124	0
25k	1	7	–2.3	1	9	–6.3	1	10	–6.3	0	40	0
50k	1	3	–2.3	1	4	–6.3	1	5	–14.1	0	24	0
100k	1	1	–2.3	1	2	–21.9	1	2	–14.1	0	12	–3.85
250k	0	3	–21.9	0	3	–6.3	0	4	–17.5	0	4	0
350k	0	2	–25.6	0	2	–10.7	0	2	–1.8	0	3	–10.71
400k	0	1	–2.3	0	1	17.2	0	2	–14.1	0	2	4.17

Table 38.20 BRR Settings for Various Bit Rates (Simple IIC Mode) (3 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)											
	50			60			120			125		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10k	2	9	–2.3	1	46	–0.27	1	93	–0.27	1	97	-0.35
25k	2	3	–2.3	0	74	0	0	149	0	1	38	0.16
50k	2	1	–2.3	0	37	–1.32	0	74	0	0	77	0.16
100k	1	3	–2.3	0	18	–1.32	0	37	–1.31	0	38	0.16
250k	0	6	–10.7	0	7	–6.25	0	14	0	2	0	-2.34
350k	0	4	–10.7	0	4	7.14	0	10	–2.6	0	10	1.46
400k	0	3	–2.34	0	4	–6.25	0	8	4.17	0	10	-11.22

Table 38.21 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple IIC Mode) (1 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)											
	8			10			16			20		
	n	N	Minimum width of High/Low Level (μs)	n	N	Minimum width of High/Low Level (μs)	n	N	Minimum width of High/Low Level (μs)	n	N	Minimum width of High/Low Level (μs)
10k	0	24	43.75 / 50.00	0	31	44.80 / 51.20	1	12	45.50 / 52.00	1	15	44.80 / 51.20
25k	0	9	17.50 / 20.00	0	12	18.20 / 20.80	1	4	17.50 / 20.00	1	6	19.60 / 22.40
50k	0	4	8.75 / 10.00	0	6	9.80 / 11.20	1	2	10.50 / 12.00	1	3	11.20 / 12.80
100k	0	2	5.25 / 6.00	0	3	5.60 / 6.40	0	4	4.37 / 5.00	0	6	4.90 / 5.60
250k	0	0	1.75 / 2.00	0	1	2.80 / 3.20	0	1	1.75 / 2.00	0	2	2.10 / 2.40
350k										0	1	1.40 / 1.60
400k										0	1	1.40 / 1.60

Table 38.21 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple IIC Mode) (2 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)											
	25			30			33			40		
	n	N	Minimum width of High/Low Level (μs)	n	N	Minimum width of High/Low Level (μs)	n	N	Minimum width of High/Low Level (μs)	n	N	Minimum width of High/Low Level (μs)
10k	1	19	44.80 / 51.20	1	23	44.80 / 51.20	1	25	44.12 / 50.42	1	32	46.20 / 52.80
25k	1	7	17.92 / 20.48	1	9	18.66 / 21.33	1	10	18.66 / 21.33	1	12	18.20 / 20.80
50k	1	3	8.96 / 10.24	1	4	9.33 / 10.66	1	5	10.18 / 11.63	1	6	9.80 / 11.20
100k	1	1	4.48 / 5.12	1	2	5.60 / 6.40	1	2	5.09 / 5.81	0	13	4.90 / 5.60
250k	0	3	2.24 / 2.56	0	3	1.86 / 2.13	0	4	2.12 / 2.42	0	4	1.75 / 2.00
350k	0	2	1.68 / 1.92	0	2	1.40 / 1.60	0	2	1.27 / 1.45	0	3	1.40 / 1.60
400k	0	1	1.12 / 1.28	0	1	0.93 / 1.07	0	2	1.27 / 1.45	0	2	1.05 / 1.20

Table 38.21 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple IIC Mode) (3 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)											
	50			60			120			125		
	n	N	Minimum width of High/Low Level (μs)	n	N	Minimum width of High/Low Level (μs)	n	N	Minimum width of High/Low Level (μs)	n	N	Minimum width of High/Low Level (μs)
10k	2	9	44.80 / 51.20	1	47	44.80 / 51.20	1	93	43.87 / 50.13	1	97	43.90/50.18
25k	2	3	17.92 / 20.48	0	74	17.50 / 20.00	0	149	17.50 / 20.00	1	38	17.47/19.97
50k	2	1	8.96 / 10.24	0	37	8.87 / 10.13	0	74	8.75 / 10.00	0	77	8.74/9.98
100k	1	3	4.48 / 5.12	0	18	4.43 / 5.07	0	37	4.43 / 5.07	0	38	4.37/4.99
250k	0	6	1.96 / 2.24	0	7	1.87 / 2.13	0	15	1.87 / 2.13	2	0	1.79/2.05
350k	0	4	1.40 / 1.60	0	5	1.40 / 1.60	0	10	1.28 / 1.47	0	10	1.23/1.41
400k	0	3	1.12 / 1.28	0	4	1.17 / 1.33	0	9	1.17 / 1.33	0	10	1.23/1.41

BRME bit (Bit Rate Modulation Enable)

Enables and disables the bit rate modulation function. The bit rate generated by on-chip baud rate generator is evenly corrected when this function is enabled.

Set to 0 in Clock-synchronous mode, Simple SPI mode, Smart Card Interface mode, Manchester mode and Simple LIN mode.

Even in asynchronous mode, set this bit to 0 when ABCSE = 1 or ABCSE2 = 1.

CKS[1:0] bits (Clock Select)

The CKS[1:0] bits select the clock source for the on-chip baud rate generator.

For the relationship between the settings of the CKS[1:0] bits and the baud rate, see BRR[7:0] bits explanation.

MDDR[7:0] bits (Modulation Duty Setting)

When the BRME bit is set to 1, the bit rate generated by the on-chip baud rate generator is evenly corrected according to the settings of MDDR (M/256). The relationship between the MDDR setting (M) and the bit rate (B) is given in Table 38.22.

The initial value of MDDR is 0xFF. Bit 7 in the CCR2 is fixed to 1.

Table 38.22 Relationship between MDDR Setting (M) and Bit Rate (B) When Bit Rate Modulation Function is Used

Mode*1	CCR2 settings			BRR setting	Error
	BGDM bit	ABCS bit	ABCSE bit		
Asynchronous, Multiprocessor transfer	0	0	0	$N = \frac{TCLK \times 10^6}{64 \times 2^{2n-1} \times (256/M) \times B} - 1$	$\text{Error (\%)} = \left\{ \frac{TCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1 \right\} \times 100$
	1	0	0	$N = \frac{TCLK \times 10^6}{32 \times 2^{2n-1} \times (256/M) \times B} - 1$	$\text{Error (\%)} = \left\{ \frac{TCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1 \right\} \times 100$
	0	1	0		
	1	1	0	$N = \frac{TCLK \times 10^6}{16 \times 2^{2n-1} \times (256/M) \times B} - 1$	$\text{Error (\%)} = \left\{ \frac{TCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1 \right\} \times 100$
	x (Arbitrarily)	x (Arbitrarily)	1*2	$N = \frac{TCLK \times 10^6}{12 \times 2^{2n-1} \times (256/M) \times B} - 1$	$\text{Error (\%)} = \left\{ \frac{TCLK \times 10^6}{B \times 12 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1 \right\} \times 100$
Simple IIC*2				$N = \frac{TCLK \times 10^6}{64 \times 2^{2n-1} \times (256/M) \times B} - 1$	—

Note: B: Bit rate (bps)
M: MDDR setting ($128 \le M \le 255$)
N: BRR setting for bound rate generator ($0 \le N \le 255$)
TCLK: Operating frequency (MHz)
n: Determined by the settings of the CKS[1:0] as listed in Table 38.9.

Note 1. Do not use this function in Clock-synchronous mode, Simple SPI mode, Smart card Interface mode, Manchester mode and Simple LIN mode.

Note 2. Adjust the bit rate so that the widths at high and low level of the SCL output in simple IIC mode satisfy the I2C standard.

Table 38.23 and Table 38.24 list examples of N settings in BRR and M settings in MDDR in Asynchronous mode.

Table 38.23 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (1) (1 of 4)

Bit rate (bps)	Operating frequency TCLK (MHz)														
	8					9.8304					10				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	5	236	0	0.03	0	7	(256) *1	0	0	0	10	173	1	–0.01
57600	0	3	236	0	0.03	0	4	240	0	0	0	4	236	0	0.03

Table 38.23 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (1) (2 of 4)

Bit rate (bps)	Operating frequency TCLK (MHz)														
	8					9.8304					10				
	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)
115200	0	1	236	0	0.03	0	1	192	0	0	0	4	236	1	0.03
230400	0	0	236	0	0.03	0	0	192	0	0	0	1	189	1	0.14
460800	0	0	236	1	0.03	0	0	192	1	0	0	0	189	1	0.14

Table 38.23 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (1) (3 of 4)

Bit rate (bps)	Operating frequency TCLK (MHz)														
	12					12.288					14				
	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)
38400	0	8	236	0	0.03	0	9	(256) *1	0	0	0	16	191	1	0
57600	0	5	236	0	0.03	0	4	192	0	0	0	13	236	1	0.03
115200	0	2	236	0	0.03	0	4	192	1	0	0	6	236	1	0.03
230400	0	2	236	1	0.03	0	2	230	1	–0.17	0	2	202	1	–0.11
460800	0	0	157	1	–0.18	0	0	154	1	0.26	0	0	135	1	0.14

Table 38.23 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (1) (4 of 4)

Bit rate (bps)	Operating frequency TCLK (MHz)														
	16					17.2032					18				
	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)
38400	0	11	236	0	0.03	0	13	(256) *1	0	0	0	18	166	1	–0.01
57600	0	7	236	0	0.03	0	6	192	0	0	0	18	249	1	–0.01
115200	0	3	236	0	0.03	0	6	192	1	0	0	8	236	1	0.03
230400	0	1	236	0	0.03	0	3	219	1	–0.2	0	1	210	0	0.14
460800	0	1	236	1	0.03	0	1	219	1	–0.2	0	0	210	0	0.14

Note: This is an example when the CCR2.ABCS = 0 and CCR2.ABCSE = 0.
Note 1. It means that the bit rate modulation function is disable. (CCR2.BRME = 0, M = 256)

Table 38.24 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (2) (1 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)														
	19.6608					20					25				
	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)
38400	0	15	(256) *1	0	0	0	10	173	0	–0.01	0	11	151	0	0
57600	0	9	240	0	0	0	9	236	0	0.03	0	7	151	0	0
115200	0	4	240	0	0	0	4	236	0	0.03	0	3	151	0	0
230400	0	1	192	0	0	0	4	236	1	0.03	0	1	151	0	0
460800	0	0	192	0	0	0	0	189	0	0.14	0	0	151	0	0

Table 38.24 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (2) (2 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)														
	30					33					40				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	36	194	1	0.01	0	14	143	0	0.01	0	21	173	0	–0.01
57600	0	10	173	0	–0.01	0	9	143	0	0.01	0	38	230	1	–0.01
115200	0	10	173	1	–0.01	0	4	143	0	0.01	0	9	236	0	0.03
230400	0	6	220	1	–0.09	0	4	143	1	0.01	0	4	236	0	0.03
460800	0	3	252	1	0.14	0	1	229	0	0.1	0	4	236	1	0.03

Table 38.24 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (2) (3 of 3)

Bit rate (bps)	Operating frequency TCLK (MHz)																			
	50					60					120					125				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	23	151	0	0	0	36	194	0	0.01	0	73	194	0	0.01	0	90	229	0	0.00
57600	0	15	151	0	0	0	21	173	0	–0.01	0	58	232	0	0	0	132	251	1	-0.01
115200	0	7	151	0	0	0	10	173	0	–0.01	0	21	173	0	–0.01	0	19	151	0	0.00
230400	0	3	151	0	0	0	10	173	1	–0.01	0	10	173	0	–0.01	0	9	151	0	0.00
460800	0	1	151	0	0	0	6	220	1	–0.09	0	10	173	1	–0.01	0	4	151	0	0.00

Note: This is an example when the CCR2.ABCS = 0 and CCR2.ABCSE = 0.
Note 1. It means that the bit rate modulation function is disable. (CCR2.BRME = 0, M = 256)

38.2.8 CCR3 : Common Control Register 3

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	BLK	GM	—	ACS0	CKE[1:0]		—	—	DEN	FM	MP	MOD[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RXDESEL	STP	SINV	LSBF	—	—	CHR[1:0]		BPEN	—	—	—	—	—	CPOL	CPHA
Value after reset:	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	CPHA	Clock Phase Select Valid in Clock synchronous mode and Simple SPI mode. Set this bit only when CCR0.TE = 0 and RE = 0. 0: Data is sampled at an odd edge and changes at an even edge (clock is delayed) 1: Data changes at an odd edge and is sampled at an even edge (clock is not delayed)	R/W

Bit	Symbol	Function	R/W
1	CPOL	Clock Polarity Select Valid in Clock synchronous mode and Simple SPI mode. Set this bit only when CCR0.TE = 0 and RE = 0. 0: SCKn in idle state is 0 1: SCKn in idle state is 1	R/W
6:2	—	These bits are read as 0. The write value should be 0.	R/W
7	BPEN	Synchronizer Bypass Enable This bit controls whether to bypass the synchronizer circuit between the bus clock and operation clock. 0: Synchronizer circuit is not bypassed 1: Synchronizer circuit is bypassed	R/W
9:8	CHR[1:0]	Character Length Valid in Asynchronous mode and Manchester mode*1 Select the data length for transmission and reception. 0 0: Transmit/receive in 9-bit data length 0 1: Transmit/receive in 9-bit data length 1 0: Transmit/receive in 8-bit data length (initial value) 1 1: Transmit/receive in 7-bit data length*2	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	LSBF	LSB First select Set this bit to 0 in Simple IIC mode. Set this bit to 1 in Simple LIN mode. 0: MSB-first 1: LSB-first	R/W
13	SINV	Transmitted/Received Data Invert Set this bit to 0 in Simple IIC mode. The level of communication pins (RXDn/TXDn) are controlled by combination of this bit and CCR1.TINV/RINV. For details, see Figure 38.3. 0: TDR contents are transmitted to TSR as is. RSR contents are stored to RDR as is. 1: TDR contents are inverted before being transmitted to TSR. RSR contents are inverted and stored to RDR.	R/W
14	STP	Stop Bit Length Valid in Asynchronous mode, Manchester mode, and Simple LIN mode 0: 1 stop bit/break delimiter length is 1-bit 1: 2 stop bits/break delimiter length is 2-bit	R/W
15	RXDESEL	Asynchronous Start Bit Edge Detection Select Valid only in Asynchronous mode Set this bit to 1 in Simple LIN mode. 0: The low level on the RXDn pin is detected as the start bit 1: A falling edge on the RXDn pin is detected as the start bit	R/W
18:16	MOD[2:0]	Communication Mode Select Select the SCI communication mode. 0 0 0: Asynchronous mode (multi-processor mode) 0 0 1: Smart card interface mode 0 1 0: Clock synchronous mode 0 1 1: Simple SPI mode 1 0 0: Simple IIC mode 1 0 1: Manchester mode 1 1 0: Simple LIN mode 1 1 1: Setting prohibited	R/W
19	MP	Multi-Processor Mode Valid in Asynchronous mode and Manchester mode 0: Multi-processor communications function is disabled 1: Multi-processor communications function is enabled	R/W
20	FM	FIFO Mode Select Valid in Asynchronous mode (including Multi-processor mode), Clock synchronous mode, and Simple SPI mode 0: TDR register and RDR register are non-FIFO buffer configuration 1: TDR register and RDR register are FIFO buffer configuration	R/W

Bit	Symbol	Function	R/W
21	DEN	Driver Enable 0: RS-485 driver control function disabled 1: RS-485 driver control function enabled	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
25:24	CKE[1:0]	Clock Enable 0 0: In Asynchronous mode: On-chip baud rate generator The SCKn pin is available for use as an I/O port in accord with the I/O port settings. In Manchester mode and Simple LIN mode: On-chip baud rate generator The SCKn pin functions as I/O port. In Clock synchronous mode and Simple SPI mode: Internal clock (master operation) The SCKn pin functions as the clock output pin. In Smart card interface mode and when CCR3.GM = 0: Output disabled (the SCKn pin is available for use as an I/O port in accord with the I/O port settings). In Smart card interface mode and when CCR3.GM = 1: Output fixed low 0 1: In Asynchronous mode: On-chip baud rate generator The clock with the same frequency as the bit rate is output from the SCKn pin. In Manchester mode and Simple LIN mode: Prohibited In Clock synchronous mode and Simple SPI mode: Internal clock (master operation) The SCKn pin functions as the clock output pin. In Smart card interface mode and when CCR3.GM = 0: Clock output In Smart card interface mode and when CCR3.GM = 1: Clock output 1 0: In Asynchronous mode: External clock or GPT clock • When using the external clock 16 times the bit rate should be input from the SCKn pin when CCR2.ABCS bit is 0. Input a clock signal with a frequency 8 times the bit rate when the CCR2.ABCS bit is 1. • When using the GPT clock*3 The SCKn pin is available for use as an I/O port in accord with the I/O port settings when the GPT clock is used. In Manchester mode and Simple LIN mode: Prohibited In Clock synchronous mode and Simple SPI mode: External clock (slave operation) The SCKn pin functions as the clock input pin. In Smart card interface mode and when CCR3.GM = 0: Prohibited In Smart card interface mode and when CCR3.GM = 1: Output fixed high 1 1: In Asynchronous mode: External clock or GPT clock • When using the external clock 16 times the bit rate should be input from the SCKn pin when CCR2.ABCS bit is 0. Input a clock signal with a frequency 8 times the bit rate when the CCR2.ABCS bit is 1. • When using the GPT clock *3 The SCKn pin is available for use as an I/O port in accord with the I/O port settings when the GPT clock is used. In Manchester mode and Simple LIN mode: Prohibited In Clock synchronous mode and Simple SPI mode: External clock (slave operation) The SCKn pin functions as the clock input pin. In Smart card interface mode and when CCR3.GM = 0: Prohibited In Smart card interface mode and when CCR3.GM = 1: Clock output	R/W

Bit	Symbol	Function	R/W
26	ACS0	Asynchronous Mode Clock Source Select Valid only in Asynchronous mode. These bits for SCI channels other than SCIn (n = 1, 2) are reserved. 0: External clock input 1: Logical AND of compare matches output from the internal GPT	R/W
27	—	These bits are read as 0. The write value should be 0.	R/W
28	GM	GSM Mode Valid only in Smart card interface mode. 0: Non-GSM mode operation 1: GSM mode operation	R/W
29	BLK	Block Transfer Mode Valid only in Smart card interface mode. 0: Non-block transfer mode operation 1: Block transfer mode operation	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. In other than Asynchronous mode and Manchester mode, this bit setting is invalid and a fixed data length of 8 bits is used. In the Simple LIN mode, only the data length of 8 bits can be used, so set it to the initial value.

Note 2. LSB first is fixed and the MSB (bit 7) in TDR is not transmitted in transmission.

Note 3. GPT clock is selectable for SCI1 and SCI2.

CPHA bit (Clock Phase Select)

The CPHA bit selects the phase of the clock signal output through the SCKn pin. For details, see Figure 38.99.

Set the bit to 1 in other than Simple SPI mode and Clock synchronous mode.

CPOL bit (Clock Polarity Select)

The CPOL bit selects the polarity of the clock signal output through the SCKn pin. For details, see Figure 38.99.

Set the bit to 1 in other than Simple SPI mode and Clock synchronous mode.

BPEN bit (Synchronizer Bypass Enable)

The BPEN bit selects whether to enable or disable the synchronization bypass function. Set this bit to 1 when the bus clock (PCLK) is also used as operation clock (TCLK). For details, see Figure 38.2 and section 38.18. Synchronizer Bypass Function.

Note: For details of this bit setting, see section 38.19.17. Notes on CCR3.BPEN Bit Setting.

CHR[1:0] bits (Character Length)

Selects the data length for transmission and reception.

Except for Asynchronous mode and Manchester mode, a fixed data length of 8 bits is used.

LSBF bit (LSB First select)

Select whether to transmit/receive data in MSB-first or LSB-first.

SINV bit (Transmitted/Received Data Invert)

SINV can invert the transmit data bit from TDR to TSR, and can invert the received data from RSR to RDR. The SINV bit does not affect the logic level of the parity bit. To invert the parity bit, invert the CCR1.PM.

STP bit (Stop Bit Length)

Selects the stop bit length in transmission.

In reception, only the first stop bit is checked regardless of the STP bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.

In addition, it is used as the break delimiter length setting when sending start frame in Simple LIN mode.

RXDESEL bit (Asynchronous Start Bit Edge Detection Select)

Selects the detection method of the start bit for reception in Asynchronous mode. When a break occurs, data receiving operation depends on the settings of the RXDESEL bit. Set this bit to 1 when reception should be stopped while a break occurs or when reception should be started without retaining the RXDn pin input at high level for the period of one data frame or longer after completion of the break.

Set this bit to 1 in Simple LIN mode. Set this bit to 0 in modes except of Asynchronous mode and Simple LIN mode.

MOD[2:0] bits (Communication Mode Select)

Selects the SCI communication mode.

Table 38.25 Relationship between communication mode selection bits (MOD[2:0]) and other operation mode setting bits

<table>
<tr><th>Communication mode</th><th colspan="8">Asynchronous</th><th>Smart card interface</th><th colspan="2">Clock synchronous</th><th colspan="4">Simple SPI</th><th>Simple IIC</th><th colspan="2">Manchester</th><th>Simple LIN</th></tr>
<tr><td>CCR3.MOD[2:0]</td><td colspan="8">000b</td><td>001b</td><td colspan="2">010b</td><td colspan="4">011b</td><td>100b</td><td colspan="2">101b</td><td>110b</td></tr>
<tr><td>CCR3.MP</td><td colspan="4">0</td><td colspan="4">1</td><td>—</td><td colspan="2">—</td><td colspan="4">—</td><td>—</td><td>0</td><td>1</td><td>—</td></tr>
<tr><td>CCR3.FM</td><td colspan="2">0</td><td colspan="2">1</td><td colspan="2">0</td><td colspan="2">1</td><td>—</td><td>0</td><td>1</td><td colspan="2">0</td><td colspan="2">1</td><td>—</td><td colspan="2">—</td><td>—</td></tr>
<tr><td>CCR3.DEN</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>—</td><td colspan="2">—</td><td colspan="4">—</td><td>—</td><td colspan="2">—</td><td>—</td></tr>
<tr><td>CCR0.SSE</td><td colspan="8">—</td><td>—</td><td colspan="2"></td><td>0</td><td>1</td><td>0</td><td>1</td><td>—</td><td colspan="2">—</td><td>—</td></tr>
</table>

Note: — is prohibited setting.

MP bit (Multi-Processor Mode)

Disables/enables multi-processor communications function. The settings of the PE bit and PM bit are invalid in multi-processor mode.

FM bit (FIFO Mode Select)

When the FM bit is set to 1, the TDR register / RDR register switches to FIFO configuration, and transmit FIFO (TDR register) / receive FIFO (RDR register) can be used for serial transmission/reception.

DEN bit (Driver Enable)

Select RS-485 driver control function disabled or enabled.

CKE[1:0] bits (Clock Enable)

The CKE[1:0] bits select the clock source and SCKn pin function.

The combination of these bit settings and of the CCR3.ACS0 bit sets the internal GPT clock.

In Smart card interface mode, these bits control the clock output from the SCKn pin.

In GSM mode, clock output can be dynamically switched. For details, see section 38.7.8. Clock Output Control.

GM bit (GSM Mode)

Setting the GM bit to 1 allows GSM mode operation.

In GSM mode, the CSR.TEND flag set timing is put forward to 11.0 etu (elementary time unit = 1-bit transfer time) from the start and the clock output control function is appended. For details, see section 38.7.6. Serial Data Transmission (Except in Block Transfer Mode), section 38.7.8. Clock Output Control.

ACS0 bit (Asynchronous Mode Clock Source Select)

Selects the clock source in the Asynchronous mode.

The ACS0 bit is valid in Asynchronous mode (CCR3.MOD[2:0] = 000b) and when an external clock input is selected (CCR3.CKE[1:0] = 10b or 11b). This bit is used to select an external clock input or the logical AND of compare matches output from the internal GPT.

Set the ACS0 bit to 0 in other than Asynchronous mode.

For SCI1 and SCI2, the GTIOCmA output (m = 3, 4, 6, 7) of GPT can be set as the serial transfer base clock. See Table 38.26 for details.

These bits for SCI channels other than SCI1 and SCI2 are reserved. The write values to these bits for other than SCI1 and SCI2 should be 0.

Table 38.26 Association between SCI channels and compare match outputs

SCI	GPT	Compare match output
SCI1	GPT3	GTIOC3A
	GPT4	GTIOC4A
SCI2	GPT6	GTIOC6A
	GPT7	GTIOC7A

Figure 38.4, Figure 38.5 show a setting example of when GTIOC3A and GTIOC4A are selected for output. This figure shows an example when GPT clock is input to SCI1.

When generating 187.5 kbps of GPT average transfer rate for PCLKD = 32 MHz:

1. Generate a frequency of 4 MHz using GTIOC3A as the base clock.
2. Generate 3/4 clock enable of base clock to set an average transfer rate of 3 MHz/16 = 187.5 kbps using GTIOC4A.

Setting examples of GPT and SCI

- GPT3.GTSSR = 0x80000000, GPT4.GTSSR = 0x80000000 (enable software counter start)
- GPT3.GTPR = 0x00000007, GPT4.GTPR = 0x0000001F (maximum count value of GTCNT)
- GPT3.GTCCRA = 0x00000003, GPT4.GTCCRA = 0x00000007 (compare match value)
- GPT3.GTCR = 0x00000000, GPT4.GTCR = 0x00000000 (Saw-wave PWM mode, timer prescaler is PCLKD)
- GPT3.GTIOR = 0x00000306, GPT4.GTIOR = 0x00000306
  (initial output low, output high at GTCCRA compare match, output low at cycle end)
- CCR3.CKE[1:0] = 10b (external clock input or GPT clock input is selected)
- CCR3.ACS0 = 1, CCR2.ABCSE = 1, CCR2.ABCS=0 (GPT clock input is selected with 16 base clock cycles for 1-bit period)
- GPT3.GTSTR = 0x00000018 (software start GTCNT counter)

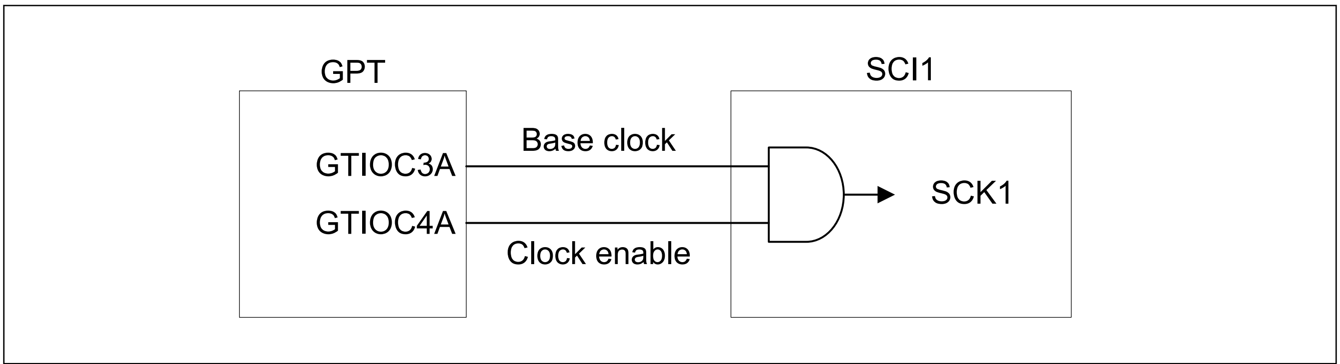


Figure 38.4 Example of GPT and SCI connection

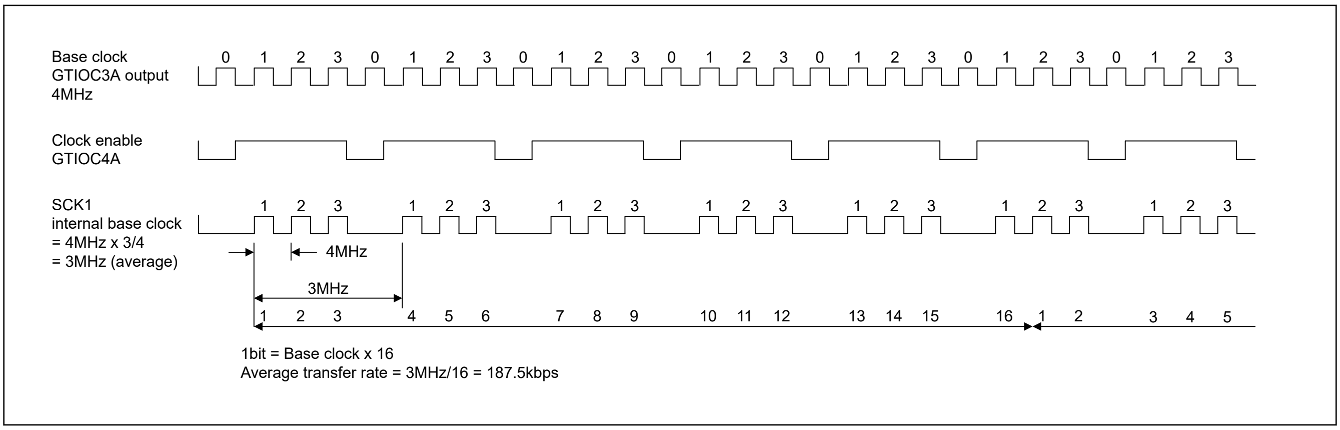


Figure 38.5 Example of average transfer rate setting when GPT clock is input

BLK bit (Block Transfer Mode)

Setting the BLK bit to 1 allows block transfer mode operation.

For details, see section 38.7.3. Block Transfer Mode.

38.2.9 CCR4 : Common Control Register 4

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x18

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	AET	ATT[2:0]			AJD	AST[2:0]			—	—	—	—	—	—	ATEN	ASEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CMPD[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	CMPD[8:0]	Compare Match Data Valid only in Asynchronous mode. Set the compare data pattern for address match wake-up function.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	ASEN	Adjust Receive Sampling Timing Enable Valid in Asynchronous mode using internal clock, Simple LIN mode using internal clock, Clock-synchronous mode operating as master, and Simple SPI mode operating as master. 0: Adjust sampling timing disabled 1: Adjust sampling timing enabled	R/W
17	ATEN	Adjust Transmit Timing Enable Valid only in Asynchronous mode using internal clock. 0: Adjust transmit timing disabled 1: Adjust transmit timing enabled	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
26:24	AST[2:0]	Adjustment Value for Receive Sampling Timing This bit enables only when ASEN is 1. In Asynchronous mode and Simple LIN mode using internal clock: The sampling timing of RXDn pin is adjusted from the middle of bit by the following formula. Adjustment sampling timing = base clock × the setting value of AST[2:0]. In Clock synchronous mode and Simple SPI mode using internal clock: The RXDn sampling timing can be adjusted by delaying 1 to 4 TCLK 0 0 0: 1-TCLK delay 0 0 1: 2-TCLK delay 0 1 0: 3-TCLK delay 0 1 1: 4-TCLK delay Others: Setting prohibited	R/W
27	AJD	Adjustment Direction for Receive Sampling Timing Valid in Asynchronous mode using internal clock and Simple LIN mode using internal clock. This bit enables only when ASEN is 1. Adjustment direction for RXDn receive sampling timing is determined by this bit. For details, see section 38.3.10. The Function of Adjust Receive Sampling Timing (Asynchronous Mode). 0: The sampling timing is adjusted backward to the middle of bit. 1: The sampling timing is adjusted forward to the middle of bit.	R/W
30:28	ATT[2:0]	Adjustment Value for Transmit Timing Valid in Asynchronous mode using internal clock and Simple LIN mode using internal clock. This bit enables only when ATEN is 1. The selected edge timing of TXDn is adjusted by the following formula. Adjustment edge timing = base clock × the setting value of ATT[2:0] This setting timing is limited by setting the base clock cycles. For details, see section 38.3.11. The Function of Adjust Transmit Timing (Asynchronous Mode).	R/W
31	AET	Adjustment Edge for Transmit Timing Valid in Asynchronous mode using internal clock and Simple LIN mode using internal clock. The adjustable edge is set by this bit. This bit enables only when ATEN is 1. For details, see section 38.3.11. The Function of Adjust Transmit Timing (Asynchronous Mode). 0: When CCR1.TINV is 0, adjust the rising edge timing. When CCR1.TINV is 1, adjust the falling edge timing. 1: When CCR1.TINV is 0, adjust the falling edge timing. When CCR1.TINV is 1, adjust the rising edge timing.	R/W

Note: S-TYPE-3, P-TYPE-3

CMPD[8:0] bits (Compare Match Data)

Set the comparison data for receive data when address match function is enabled (CCR0.DCME = 1). CCR4.CMPD[8:0] should be written while CCR0.DCME is 0.

For the comparison data, it can select length from three types, they are CMPD[6:0] with 7-bit length, CMPD[7:0] with 8-bit length, and CMPD[8:0] with 9-bit length.

Note: If the description in this document and the timing chart do not specify the ASEN / ATEN setting value, it means that the reception sampling adjustment function/transmission timing adjustment function are OFF (CCR4.ASEN = 0, CCR4.ATEN = 0).

ASEN bit (Adjust Receive Sampling Timing Enable)

When the ASEN bit is 1, the receive sampling timing adjustment function is enabled. Control is different in Asynchronous mode, Simple LIN mode, Clock-synchronous mode, and Simple SPI mode.

For details of using internal clock in Asynchronous mode, see section 38.6.7. Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with Internal Clock Used. The operation when the Simple LIN mode internal clock is selected is the same as when the Asynchronous mode internal clock is selected.

For details of Clock synchronous mode operating as master and Simple SPI mode operating as master, see section 38.6.6. Simultaneous Serial Data Transmission and Reception in Clock Synchronous Mode. Only the digital delay of the master mode receive sampling clock (MRCLK) can be controlled by this bit. MRCLK analog delay cannot be controlled.

ATEN bit (Adjust Transmit Timing Enable)

When the ATEN bit is 1, the transmission timing adjustment function is enabled. The transmission timing adjustment function can adjust the edge timing of the waveform output from the TXDn pin. For details, see section 38.3.11. The Function of Adjust Transmit Timing (Asynchronous Mode).

AST[2:0] bits (Adjustment Value for Receive Sampling Timing)

When the ASEN bit is 1, the receive sampling timing can be adjusted according to this bit setting value.

In Asynchronous mode and Simple LIN mode using internal clock.

The sampling timing of RXDn pin is adjusted from the middle of bit by the following formula. This setting value is limited by setting the base clock cycles. For details, see section 38.3.10. The Function of Adjust Receive Sampling Timing (Asynchronous Mode).

Adjustment sampling timing = base clock × the setting value of AST[2:0].

In Clock-synchronous mode and Simple SPI mode using internal clock

The sampling timing of RXDn pin can be adjusted by delaying 1 to 4 TCLK. For details, see section 38.6.7. Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with Internal Clock Used.

000: 1-TCLK delay

001: 2-TCLK delay

010: 3-TCLK delay

011: 4-TCLK delay

1xx: Setting prohibited

AJD bit (Adjustment Direction for Receive Sampling Timing)

Set the RXDn pin sampling timing adjustment direction from the bit center to the rear or front. For details, see section 38.3.10. The Function of Adjust Receive Sampling Timing (Asynchronous Mode).

ATT[2:0] bits (Adjustment Value for Transmit Timing)

The edge timing of the TXDn pin specified by the AET bit is adjusted by the base clock × ATT[2:0] setting value. The upper limit of the adjustment time that can be set is limited by the number of base clock cycles. For details, see section 38.3.11. The Function of Adjust Transmit Timing (Asynchronous Mode).

AET bit (Adjustment Edge for Transmit Timing)

Set the TXDn pin edge for timing adjustment. For details, see section 38.3.11. The Function of Adjust Transmit Timing (Asynchronous Mode).

38.2.10 ICR : Simple IIC Control Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	IICSCLS[1:0]		IICSDAS[1:0]		—	IICSTPREQ	IICRSTAREQ	IICSTAREQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	IICACKT	—	—	—	IICCSC	IICINTM	—	—	—	IICDL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	IICDL[4:0]	SDA Delay Output Select The following cycles are of the clock signal from the on-chip baud rate generator: 0x00: No output delay 0x01: 0 to 1 cycle 0x02: 1 to 2 cycles 0x03: 2 to 3 cycles 0x04: 3 to 4 cycles 0x05: 4 to 5 cycles ⋮ 0x1E: 29 to 30 cycles 0x1F: 30 to 31 cycles	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	IICINTM	IIC Interrupt Mode Select 0: Use ACK/NACK interrupts 1: Use reception and transmission interrupts	R/W
9	IICCSC	Clock Synchronization 0: No synchronization with the clock signal 1: Synchronization with the clock signal	R/W
12:10	—	These bits are read as 0. The write value should be 0.	R/W
13	IICACKT	ACK Transmission Data 0: ACK transmission 1: NACK transmission and reception of ACK/NACK	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	IICSTAREQ	Start Condition Generation 0: A start condition is not generated 1: A start condition is generated *1 *3 *4 *5	R/W
17	IICRSTAREQ	Restart Condition Generation 0: A restart condition is not generated 1: A restart condition is generated *2 *3 *4 *5	R/W
18	IICSTPREQ	Stop Condition Generation 0: A stop condition is not generated. 1: A stop condition is generated *2 *3 *4 *5	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
21:20	IICSDAS[1:0]	SDA Output Select 0 0: Serial data output 0 1: Generate a start, restart, or stop condition 1 0: Output the low level on the SDAn pin 1 1: Place the SDAn pin in the high-impedance state	R/W
23:22	IICSCLS[1:0]	SCL Output Select 0 0: Serial clock output 0 1: Generate a start, restart, or stop condition 1 0: Output the low level on the SCLn pin 1 1: Place the SCLn pin in the high-impedance state	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. In bus free state, perform the start condition generation.
Note 2. In bus busy state, perform restart or stop condition generation when the SCLn pin after acknowledgment described in Figure 38.86 and Figure 38.87 is low level.
Note 3. Do not set more than one from among the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits to 1 at a given time.
Note 4. Execute the generation of a condition after the value of the IICSTIF flag is 0.
Note 5. Do not write 0 to this bit while it is 1. Generation of a condition is suspended by writing 0 to this bit while it is 1.

IICDL[4:0] bits (SDA Delay Output Select)

The IICDL[4:0] bits are used to set a delay for output on the SDAn pin relative to the falling edge of the output on the SCLn pin. The available delay settings range from no delay to 31 cycles, with the clock signal from the on-chip baud rate generators the base. The signal obtained by frequency-dividing TCLK by the divisor set in CCR2.CKS[1:0] is supplied as the clock signal from the on-chip baud rate generator.

Set these bits to 00000b unless operation is in simple IIC mode. In simple IIC mode, set the bits to a value in the range from 00001b to 11111b.

IICINTM bit (IIC Interrupt Mode Select)

The IICINTM bit selects the sources of interrupt requests in Simple IIC mode.

IICCSC bit (Clock Synchronization)

Set the IICCSC bit to 1 if the internally generated SCLn clock signal is to be synchronized when the SCLn pin is placed at the low level and when a wait is inserted by other device, for example.

The SCLn clock signal is not synchronized if the IICCSC bit is 0. The SCLn clock signal is generated in accord with the rate selected in the BRR regardless of the level being input on the SCLn pin.

Set the IICCSC bit to 1 except during debugging.

IICACKT bit (ACK Transmission Data)

Transmitted data contains ACK bits. Set this bit to 1 when ACK and NACK bits are received.

IICSTAREQ bit (Start Condition Generation)

When a start condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTAREQ bit to 1.

To generate the start condition after generating the stop condition, start the generation of the start condition with a half cycle period of the bit rate from the stop condition generation interrupt (STI) request output.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the start condition

IICRSTAREQ bit (Restart Condition Generation)

When a restart condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICRSTAREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the restart condition

IICSTPREQ bit (Stop Condition Generation)

When a stop condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTPREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the stop condition

IICSDAS[1:0] bits (SDA Output Select)

The IICSDAS[1:0] bits control output from the SDAn pin.

IICSCLS[1:0] bits (SCL Output Select)

The IICSCLS[1:0] bits control output from the SCLn pin.

38.2.11 FCR : FIFO Control Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x24

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	RSTRG[4:0]					RFRST	—	—	RTRG[4:0]				
Value after reset:	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TFRST	—	—	TTRG[4:0]					—	—	—	—	—	—	—	DRES
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DRES	Receive Data Ready Error Select Bit Valid in Asynchronous mode. This bit selects the interrupt request for a reception data ready detection. 0: Reception data full interrupt (SCIn_RXI) 1: Receive error interrupt (SCIn_ERI)	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
12:8	TTRG[4:0]	Transmit FIFO Data Trigger Number Valid in Asynchronous mode (including Multiprocessor mode), Clock synchronous mode, and Simple SPI mode. Trigger number must be set to 15 or less. 0x00: Trigger number 0 ⋮ 0x1F: Trigger number 31	R/W
14:13	—	These bits are read as 0. The write value should be 0.	R/W
15	TFRST	Transmit FIFO Data Register Reset This bit enables only when CCR3.FM is 1. The read value is always 0. 0: When set, this bit is invalid and does not affect operation 1: The number of data stored in Transmit-FIFO (TDR register) is 0	W
20:16	RTRG[4:0]	Receive FIFO Data Trigger Number Valid in Asynchronous mode (including Multi processor mode), Clock synchronous mode, and Simple SPI mode. Trigger number must be set to 15 or less. 0x00: Trigger number 0 ⋮ 0x1F: Trigger number 31	R/W
22:21	—	These bits are read as 0. The write value should be 0.	R/W
23	RFRST	Receive FIFO Data Register Reset This bit enables only when CCR3.FM is 1. The read value is always 0. 0: When set, this bit is invalid and does not affect operation 1: The number of data stored in Receive-FIFO(RDR register) is 0	W
28:24	RSTRG[4:0]	RTS Output Active Trigger Number Select Valid in Asynchronous mode (including Multi processor mode) and Clock synchronous mode. This bit enables only when CCR3.FM = 1 and CCR1.CTSE = 0 and CCR0.SSE = 0. Trigger number must be set to 15 or less. 0x00: Trigger number 0 ⋮ 0x1F: Trigger number 31	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

DRES bit (Receive Data Ready Error Select Bit)

Select whether the detection of receive data ready (FRSR.DR flag = 1) is the cause of SCIn_RXI interrupt request or the cause of SCIn_ERI interrupt request.

TTRG[4:0] bits (Transmit FIFO Data Trigger Number)

The TDFE flag is set to 1 when the quantity of transmit data in the transmit-FIFO(TDR register) is equal to or less than the specified transmit triggering number. If CCR0.TIE = 1, SCIn_TXI interrupt request occurs.

Note: Trigger number must be set to 15. When the trigger number is set to 16 or more, unexpected SCIn_TXI interrupt occurs.

TFRST bit (Transmit FIFO Data Register Reset)

When the TFRST bit is set to 1, the number of the transmission data stored in Transmit-FIFO(TDR register) is 0.

RTRG[4:0] bits (Receive FIFO Data Trigger Number)

The CSR.RDRF flag is set to 1 when the quantity of receive data in the receive-FIFO(RDR register) is equal to or greater than the specified receive triggering number. If CCR0.RIE = 1, SCIn_RXI interrupt request occurs. When FCR.RTRG is set to 0, RDRF bit is set if the quantity of data in receive-FIFO is greater than or equal to 1.

Note: Trigger number must be set to 15. When the trigger number is set to 16 or more, unexpected SCIn_RXI interrupt occurs.

RFRST bit (Receive FIFO Data Register Reset)

When the RFRST bit is set to 1, the number of the reception data stored in Receive-FIFO(RDR register) is 0.

RSTRG[4:0] bits (RTS Output Active Trigger Number Select)

When the quantity of receive data stored in the receive-FIFO (RDR register) is equal to or greater than this number, the RTSn signal is in the High state. When FCR.RSTRG is set to 0, RTSn is in the high state if the quantity of data in receive FIFO is greater than or equal to 1.

Note: Trigger number must be set to 15. When the trigger number is set to 16 or more, RTSn goes to a high state at unexpected timing.

38.2.12 MCR : Manchester Control Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x2C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	SBEREN	SYEREN	PFEREN	—	—	RPPAT[1:0]		RPLEN[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	TPPAT[1:0]		TPLEN[3:0]				—	SBSEL	SYNSEL	SYNVAL	—	ERTEN	TMPOL	RMPOL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RMPOL	Polarity of Received Manchester Code Sets the polarity of the received Manchester code. 0: Logic 0 is coded as a zero-to-one transition in Manchester code Logic 1 is coded as a one-to-zero transition in Manchester code 1: Logic 0 is coded as a one-to-zero transition in Manchester code Logic 1 is coded as a zero-to-one transition in Manchester code	R/W

Bit	Symbol	Function	R/W
1	TMPOL	Polarity of Transmit Manchester Code Sets the polarity of the transmit Manchester code 0: Logic 0 is coded as a zero-to-one transition in Manchester code Logic 1 is coded as a one-to-zero transition in Manchester code 1: Logic 0 is coded as a one-to-zero transition in Manchester code Logic 1 is coded as a zero-to-one transition in Manchester code	R/W
2	ERTEN	Manchester Edge Retiming Enable Sets the receive retiming function. 0: Disables the receive retiming function 1: Enables the receive retiming function	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	SYNVAL	SYNC value Setting Sets the SYNC type of the start bit(s) in the Manchester code. When the start bit area consists of one bit (SBSEL = 0) ● When transmitting: 0: The start bit is added as a zero-to-one transition 1: The start bit is added as a one-to-zero transition ● When receiving: 0: Only when the start bit is a zero-to-one transition, the data is received. Other cases are determined as an error. 1: Only when the start bit is a one-to-zero transition, the data is received. Other cases are determined as an error. When the start bit area consists of three bits (SBSEL = 1) ● When transmitting: 0: The start bits are added as a zero-to-one transition (DATA SYNC) 1: The start bits are coded as a one-to-zero transition (COMMAND SYNC) ● When receiving: When the start bit area consists of three bits, data is received regardless of the value of this bit.	R/W
5	SYNSEL	SYNC Select 0: The start bit pattern is set with the SYNVAL bit 1: The start bit pattern is set with the TSYNC bit	R/W
6	SBSEL	Start Bit Select 0: The start bit area consists of one bit 1: The start bit area consists of three bits (COMMAND SYNC or DATA SYNC)	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
11:8	TPLEN[3:0]	Transmit Preface Length Set the preface length of the transmit data in Manchester mode. 0x0: Disables the transmit preface generation Others: Transmit preface length (bit length)	R/W
13:12	TPPAT[1:0]	Transmit Preface Pattern Set the preface pattern of the transmit data. 0 0: ALL ZERO 0 1: ZERO ONE 1 0: ONE ZERO 1 1: ALL ONE	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
19:16	RPLEN[3:0]	Receive Preface Length Set the preface length in received frames when Manchester mode is enabled. 0x0: Disables the receive preface generation Others: Receive preface length (bit length)	R/W
21:20	RPPAT[1:0]	Receive Preface Pattern Set the preface pattern of received frames 0 0: ALL ZERO 0 1: ZERO ONE 1 0: ONE ZERO 1 1: ALL ONE	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
24	PFEREN	Preface Error Enable Specifies whether to handle a preface error as an interrupt source. 0: Does not handle a preface error as an interrupt source 1: Handles a preface error as an interrupt source	R/W
25	SYEREN	Receive SYNC Error Enable Specifies whether to handle a receive SYNC error as an interrupt source. 0: Does not handle a receive SYNC error as an interrupt source 1: Handles a receive SYNC error as an interrupt source	R/W
26	SBEREN	Start Bit Error Enable Specifies whether to handle a start bit error as an interrupt source. 0: Does not handle a start bit error as an interrupt source 1: Handles a start bit error as an interrupt source	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

RMPOL bit (Polarity of Received Manchester Code)

The RMPOL bit sets the polarity of the received Manchester code. For details, see section 38.5.7. Serial Data Reception in Manchester Mode.

TMPOL bit (Polarity of Transmit Manchester Code)

The TMPOL bit sets the polarity of the transmit Manchester code. For details, see section 38.5.6. Serial Data Transmission in Manchester Mode.

ERTEN bit (Manchester Edge Retiming Enable)

The ERTEN bit sets the receive retiming function in Manchester mode.

For information on the receive retiming function, see section 38.5.9. Receive Retiming.

SYNVAL bit (SYNC value Setting)

The SYNVAL bit is valid when the SYNSEL bit of the MCR is set to 0.

The SYNC type can be set by combining this bit and the SBSEL bit.

For the start bit area determined by the combination of this bit and the SBSEL bit, see Figure 38.54 and Figure 38.55.

SYNSEL bit (SYNC Select)

The SYNSEL bit is valid when the SBSEL bit of the MCR is set to 1. This bit determines the destination to be referred to for setting the SYNC type of the start bit area added to Manchester frames.

When this bit is set to 0, the SYNVAL bit of the MCR is referred to.

When this bit is set to 1, the TSYNC bit in the TDR register is referred to.

SBSEL bit (Start Bit Select)

The SBSEL bit sets the start bit area in Manchester frames.

When this bit is set to 1, the start bit area added to each frame consists of three bits, and the SYNSEL and SYNVAL bits in this register are valid.

When this bit is set to 0, the start bit area added to each frame consists of one bit.

TPLEN[3:0] bits (Transmit Preface Length)

The TPLEN[3:0] bits set the preface bit length of the transmit data in Manchester mode.

The settable range is 0x0 to 0xF (0d to 15d). The setting of 0x0 disables the transmit preface, which is not added.

TPPAT[1:0] bits (Transmit Preface Pattern)

The TPPAT[1:0] bits set one of the four preface patterns in Manchester mode. For the transmit data when the TPPAT[1:0] bits are set, see Figure 38.53.

When these bits are set to 00b, the preface area is set to all 0s.

When these bits are set to 01b, the preface area is set to the zero-one-zero-one pattern.

When these bits are set to 10b, the preface area is set to the one-zero-one-zero pattern.

When these bits are set to 11b, the preface area is set to all 1s.

RPLEN[3:0] bits (Receive Preface Length)

The RPLEN[3:0] bits set the preface bit length of the received frames in Manchester mode.

The settable range is 0x0 to 0xF (0d to 15d). The setting of 0x0 disables the receive preface, which is not added. When 0x1 to 0xF is set, the set value is handled as the receive preface bit length.

RPPAT[1:0] bits (Receive Preface Pattern)

The RPPAT[1:0] bits set one of the four preface patterns in Manchester mode. For the transmit and receive data when the RPPAT[1:0] bits are set, see Figure 38.53.

When these bits are set to 00b, the preface area is handled as all 0s.

When these bits are set to 01b, the preface area is handled as the zero-one-zero-one pattern.

When these bits are set to 10b, the preface area is handled as the one-zero-one-zero pattern.

When these bits are set to 11b, the preface area is handled as all 1s.

PFEREN bit (Preface Error Enable)

The PFEREN bit specifies whether to handle a preface error as an interrupt source.

When it is set to 0, a preface error is not handled as an interrupt source. When it is set to 1, a preface error is handled as an interrupt source.

SYEREN bit (Receive SYNC Error Enable)

The SYEREN bit specifies whether to handle a receive SYNC error as an interrupt source.

When it is set to 0, a receive SYNC error is not handled as an interrupt source. When it is set to 1, a receive SYNC error is handled as an interrupt source.

SBEREN bit (Start Bit Error Enable)

The SBEREN bit specifies whether to handle a start bit error as an interrupt source.

When it is set to 0, a start bit error is not handled as an interrupt source. When it is set to 1, a start bit error is handled as an interrupt source.

38.2.13 DCR : Driver Control Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x30

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	DENGT[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DEAST[4:0]					—	—	—	—	—	—	—	DEPOL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DEPOL	Driver Effective Polarity Select Valid only in Asynchronous mode. 0: The DEn signal is active-high 1: The DEn signal is active-low	R/W

Bit	Symbol	Function	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
12:8	DEAST[4:0]	Driver Assertion Time Valid only in Asynchronous mode. Set the driver assertion time. When DEN = 1, the driver assertion time is inserted in addition to the normal transmission waiting time. Setting DEAST[4:0] = 0x00 is prohibited.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
20:16	DENGT[4:0]	Driver Negate Time Valid only in Asynchronous mode. Set the driver negation time. When DEN = 1, the driver negate time is inserted after STOP bit transmission end. Setting DENGT[4:0] = 0x00 is prohibited.	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

DEPOL bit (Driver Effective Polarity Select)

Select the active level of the DEn signal.

DEAST[4:0] bits (Driver Assertion Time)

Set the driver assertion time (= time from the activation of the DEn (Driver Enable) signal to the start of the start bit). It is expressed in base clock period.

Driver assertion time

= DEAST [4:0] set value × base clock period + transmission waiting time

DENGT[4:0] bits (Driver Negate Time)

Set the driver negation time (= time from the end of the last stop bit of the transmitted message until the DEn (Driver Enable) signal is disabled). It is expressed in base clock period.

Driver negate time

= DENGT[4:0] set value × base clock period

If the transmission data is written during the driver negate time, transmit starting operation is different depends on the writing timing. The DEn signal remains valid, and transmission of the start bit may start after the transmission wait time has elapsed. Also, the DEn signal may become invalid once, and start bit transmission may start after the Driver assertion time has elapsed.

38.2.14 XCR0 : Simple LIN Control Register 0

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x34

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	BCCS[1:0]		—	AEDIE	COFIE	BFDIE	—	—	BCDIE	BFOIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PIBS[2:0]			PIBE	CF1DS[1:0]		CF0RE	BFE	—	—	—	—	—	—	TCSS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	TCSS[1:0]	Timer Count Clock Source Selection Valid in Simple LIN mode. Select the clock source of the timer in Simple LIN module. 0 1: TCLK/4 1 0: TCLK/16 1 1: TCLK/64	R/W*1
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	BFE	Break Field Enable Set the presence or absence of Break Field of Start Frame. 0: No Break Field 1: With Break Field	R/W*3
9	CF0RE	Control Field 0 Enable Set the presence or absence of Control Field 0 of Start Frame 0: No Control Field 0 1: With Control Field 0	R/W*3
11:10	CF1DS[1:0]	Control Field1 Compare Data Select Select the compare data for Control Field 1. 0 0: Select XCR1.PCF1D[7:0] as the compare data 0 1: Select XCR1.SCF1D[7:0] as the compare data 1 0: Select both XCR1.PCF1D[7:0] and XCR1.SCF1D[7:0] as the compare data 1 1: Setting prohibited	R/W*3
12	PIBE	Priority Interrupt Bit Enable 0: Priority interrupt bit disabled 1: Priority interrupt bit enabled	R/W*3
15:13	PIBS[2:0]	Priority Interrupt Bit Select Specify one of bits 0 to 7 of Control Field 1 as the priority interrupt bit. 0 0 0: Bit 0 of Control Field 1 0 0 1: Bit 1 of Control Field 1 0 1 0: Bit 2 of Control Field 1 0 1 1: Bit 3 of Control Field 1 1 0 0: Bit 4 of Control Field 1 1 0 1: Bit 5 of Control Field 1 1 1 0: Bit 6 of Control Field 1 1 1 1: Bit 7 of Control Field 1	R/W*3
16	BFOIE	Break Field Output Completion Interrupt Enable Select whether to include Break Field output completion as a SCIn_TXI interrupt factor. 0: Break Field output completion is not included in SCIn_TXI interrupt factor 1: Break Field output completion is included in SCIn_TXI interrupt factor	R/W
17	BCDIE	Bus Conflict Detection Interrupt Enable Select whether to output an SCIn_ERI interrupt when a bus collision is detected. 0: Bus conflict detection is not included in SCIn_ERI interrupt factor 1: Bus conflict detection is included in SCIn_ERI interrupt factor	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
20	BFDIE	Break Field Detection Interrupt Enable Select whether to output a SCIn_BFD interrupt when a Break Field is detected. 0: Break Field detection interrupt disabled 1: Break Field detection interrupt enabled	R/W
21	COFIE	Counter Overflow Interrupt Enable Select whether to include counter overflow as an SCIn_ERI interrupt factor. 0: Counter overflow is not included in SCIn_ERI interrupt factor 1: Counter overflow is included in SCIn_ERI interrupt factor	R/W
22	AEDIE	Active Edge Detection Interrupt Enable Select whether to output an SCIn_AED interrupt when a valid edge is detected. 0: Active edge detection interrupt disabled 1: Active edge detection interrupt enabled	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
25:24	BCCS[1:0]	Bus Conflict Detection Clock Selection Select the sampling clock for the bus conflict detection circuit. When CCR2.ABCS = 1, setting BCCS[1] = 1 is prohibited. 0 0: Base clock*2 0 1: Base clock/2 1 0: Base clock/4 1 1: Setting prohibited	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. You can rewrite TCSS[1:0] only when the timer is stopped (TCST = 0 and SDST = 0 and BMEN = 0)
Note 2. Base clock: 1/16 period of 1 bit period when CCR2.ABCS = 0, 1/8 period of 1 bit period when CCR2.ABCS = 1.
Note 3. This bit is a setting bit required for the start frame reception operation. Rewrite this bit when the start frame reception or transmission is not in progress (XCR1.SDST = 0 and XCR1.TCST = 0).

TCSS[1:0] bits (Timer Count Clock Source Selection)

Select clock source of timer in Simple LIN module.

BFE bit (Break Field Enable)

Set the presence or absence of Break Field of Start Frame.

CF0RE bit (Control Field 0 Enable)

Set the presence or absence of Control Field 0 of Start Frame.

CF1DS[1:0] bits (Control Field1 Compare Data Select)

Select the compare data for Control Field 1.

PIBE bit (Priority Interrupt Bit Enable)

Select whether to enable priority interrupt bit comparison of Control Field 1. When the PIBE bit is 1, regardless of the XCR1.CF1CE [7:0] setting value, the bit specified in PIBS [2:0] is compared with the primary comparison data for Control Field 1 (XCR1.PCF1D [7:0]).

PIBS[2:0] bits (Priority Interrupt Bit Select)

Specify bit N (N = 0-7) of Control Field 1 as the priority interrupt bit.

BFOIE bit (Break Field Output Completion Interrupt Enable)

Select whether to include Break Field output completion as a SCIn_TXI interrupt factor. Set CCR0.TIE = 1 and CCR3.MOD [1: 0] = 110b, to output SCIn_TXI upon completion of Break Field output.

BCDIE bit (Bus Conflict Detection Interrupt Enable)

Select whether to output an SCIn_ERI interrupt when a bus collision is detected. In Simple LIN mode (CCR3.MOD [1:0] = 110b), SCIn_ERI output control is performed with the BCDIE bit. When CCR3.MOD [1:0] = 110b and BCDIE = 1, an SCIn_ERI interrupt is issued when a bus collision is detected even if CCR0.RIE = 0.

COFIE bit (Counter Overflow Interrupt Enable)

Select whether to include counter overflow as an SCIn_ERI interrupt factor. Set CCR0.RIE = 1 and CCR3.MOD [1:0] = 110b are required to output SCIn_ERI upon counter overflow.

AEDIE bit (Active Edge Detection Interrupt Enable)

Select whether to output an SCIn_AED interrupt when a valid edge is detected. To output SCIn_AED with valid edge detection, XCR1.BMEN = 1 and CCR3.MOD [1:0] = 110b must be set.

BCCS[1:0] bits (Bus Conflict Detection Clock Selection)

Select the sampling clock for the bus conflict detection circuit.

38.2.15 XCR1 : Simple LIN Control Register 1

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x38

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CF1CE[7:0]								SCF1D[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PCF1D[7:0]								—	—	BMEN	SDST	—	—	—	TCST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TCST	Break Field Output Timer Count Start Trigger 0: Break Field output timer count stopped 1: Break Field output timer count start	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	SDST	Start Frame Detection Enable 0: Start Frame/Break Field detection disabled 1: Start Frame/Break Field detection enabled Do not set this bit and TCST bit to 1 at the same time.	R/W
5	BMEN	Bit Rate Measurement Enable Set this bit to 1 simultaneously with the SDST bit. This bit can be set to 0 at any timing. 0: Bit rate measurement disabled 1: Bit rate measurement enabled	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PCF1D[7:0]	Priority Compare Data for Control Field 1 The priority compare data for Control Field 1.	R/W
23:16	SCF1D[7:0]	Secondary Compare Data for Control Field 1 The secondary compare data for Control Field 1.	R/W
31:24	CF1CE[7:0]	Control Field 1 Compare Bit Enable Select whether to compare bit N of Control Field 1. (N = 0 -7) 0: Control Field 1 bit N compare disabled 1: Control Field 1 bit N compare enabled	R/W

Note: S-TYPE-3, P-TYPE-3

TCST bit (Break Field Output Timer Count Start Trigger)

[Clearing condition]

- When 0 is written to TCST. Break Field output timer count is stopped and TXDn output becomes idle level.
- When Break Field output for the period set in XCR2.BFLW [15: 0] is completed.

[Setting condition]

- When 1 is written to TCST. Start Break Field output from TXDn. Holds 1 during Break Field output.

SDST bit (Start Frame Detection Enable)

When 1 is written to the SDST bit, Start Frame detection starts. When XCR0.BFE = 1 is set, Break Field can be detected during Start Frame is detected and after Start Frame is detected. When XCR0.BFE = 0 is set, Break Field is not detected.

When 0 is written to the SDST bit, Start Frame detection and Break Field detection are stopped. However, if XSR0.RXDSF = 0 at the time of stop, it is not possible to stop data reception of the SCI core with this bit. Write 0 to CCR0.RE to stop the reception operation or perform reception completion processing (CSR.RDRF clear or RDR read) after reception is completed.

BMEN bit (Bit Rate Measurement Enable)

Set this bit to 1 simultaneously with the SDST bit. When the BMEN bit is set to 1, the valid edge interval of Control Field 0 and Control Field 1 data is measured.

PCF1D[7:0] bits (Priority Compare Data for Control Field 1)

Set the priority compare data for Control Field 1.

SCF1D[7:0] bits (Secondary Compare Data for Control Field 1)

Set the secondary compare data for Control Field 1.

CF1CE[7:0] bits (Control Field 1 Compare Bit Enable)

Select whether to compare bit N of Control Field 1. (N = 0 - 7)

When all of the CF1CE[7:0] bits are set to 0 (CF1CE[7:0] = 0x00), it is always determined that Control Field 1 matches when reception is completed, and XSR0.CF1MF is set. This bit is a comparison enable with PCF1D[7:0] or SCF1D[7:0], and is not a priority interrupt bit comparison enable.

38.2.16 XCR2 : Simple LIN Control Register 2

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x3C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	BFLW[15:0]															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CF0CE[7:0]								CF0D[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CF0D[7:0]	Control Field 0 Compare Data The compare data for Control Field 0.	R/W
15:8	CF0CE[7:0]	Control Field 0 Compare Bit Enable Select whether to compare bit N of Control Field 0. (N = 0 - 7) 0: Control Field 0 bit N compare disabled 1: Control Field 0 bit N compare enabled	R/W
31:16	BFLW[15:0]	Break Field Length Setting This register sets the Break Field length. The Break Field length is (BFLW [15:0] setting value + 1) × clock of the timer. The upper limit for setting this register is 0xFFFE. Setting is prohibited for 0xFFFF.	R/W

Note: S-TYPE-3, P-TYPE-3

CF0D[7:0] bits (Control Field 0 Compare Data)

The compare data for Control Field 0.

CF0CE[7:0] bits (Control Field 0 Compare Bit Enable)

Select whether to compare bit N of Control Field 0. (N = 0 - 7)

When all of the CF0CE[7:0] bits are set to 0 (CF0CE[7:0] = 0x00), it is always determined that Control Field 0 matches when reception is completed, and XSR0.CF0MF is set.

BFLW[15:0] bits (Break Field Length Setting)

BFLW[15:0] are 16-bit Break Field length setting bits and the initial value is 0xFFFE.

Set the Break Field length to 1 frame or more. The LIN standard stipulates that the Break Field length is 13 bits or more.

When the Break Field is sending, writing 1 to TCST causes SCI to output the Break Field on TXDn. Up-counting is performed with the clock of the timer selected by XCR0.TCSS[1:0]. When the count value matches the value set in this register, up-counting is stopped and Break Field output from TXDn is also stopped.

When the Break Field is receiving, writing 1 to SDST enables Start Frame detection. SCI starts counting from the negative edge of RXDn. The clock of the timer is selected by XCR0.TCSS[1:0].

When the count value matches the value set in the XCR2, a Break Field has been detected. Up-counting continues until the next valid edge or counter overflow.

38.2.17 CSR : Common Status Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x48

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RDRF	TEND	TDRE	FER	PER	MFF	—	ORER	—	—	—	—	—	DFER	DPER	DCMF
Value after reset:	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RXDMON	—	—	—	—	—	—	—	—	—	—	ERS	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0.	R
4	ERS	Error Signal Status Flag Valid only in Smart card interface mode. 0: Error signal low not responded 1: Error signal low responded	R
14:5	—	These bits are read as 0.	R
15	RXDMON	Serial Input Data Monitor Bit The state of the RXDn pin without synchronizing by bus clock is shown. 0: When RINV is 0, RXDn pin is the low level. When RINV is 1, RXDn pin is the high level. 1: When RINV is 0, RXDn pin is the high level. When RINV is 1, RXDn pin is the low level.	R
16	DCMF	Data Compare Match Flag Valid only in Asynchronous mode. 0: No matched 1: Matched	R
17	DPER	Data Compare Match Parity Error Flag Valid only in Asynchronous mode. 0: No parity error occurred at address match detection 1: A parity error has occurred at address match detection	R
18	DFER	Data Compare Match Framing Error Flag Valid only in Asynchronous mode. 0: No framing error occurred at address match detection 1: A framing error has occurred at address match detection	R
23:19	—	These bits are read as 0.	R
24	ORER	Overrun Error Flag 0: No overrun error occurred 1: An overrun error has occurred	R
25	—	This bit is read as 0.	R

Bit	Symbol	Function	R/W
26	MFF	Mode Fault Error Flag Valid only in Simple SPI mode. 0: No mode fault error 1: Mode fault error	R
27	PER	Parity Error Flag 0: Non-FIFO selected (CCR3.FM = 0): No parity error occurred FIFO selected (CCR3.FM = 1): No parity error in all received data in receive-FIFO 1: Non-FIFO selected (CCR3.FM = 0): A parity error has occurred FIFO selected (CCR3.FM = 1): One or more parity errors occurred in received data in receive-FIFO	R
28	FER	Framing Error Flag 0: Non-FIFO selected (CCR3.FM = 0): No framing error occurred FIFO selected (CCR3.FM = 1): No framing error in all received data in receive-FIFO 1: Non-FIFO selected (CCR3.FM = 0): A framing error has occurred FIFO selected (CCR3.FM = 1): One or more framing errors occurred in received data in receive-FIFO	R
29	TDRE	Transmit Data Empty Flag 0: Non-FIFO selected (CCR3.FM = 0): Transmit data is in TDR register FIFO selected (CCR3.FM = 1): The quantity of transmit data written in transmit-FIFO exceeds the specified transmit triggering number. 1: Non-FIFO selected (CCR3.FM = 0): No transmit data is in TDR register FIFO selected (CCR3.FM = 1): The quantity of transmit data written in transmit-FIFO is equal to or less than the specified transmit triggering number.	R
30	TEND	Transmit End Flag 0: A character is being transmitted or standing by for transmission. 1: Character transfer has been completed, or sending Break Field.	R
31	RDRF	Receive Data Full Flag 0: Non-FIFO selected (CCR3.FM = 0): No received data is in RDR register FIFO selected (CCR3.FM = 1): The quantity of receive data written in receive-FIFO falls below the specified receive triggering number. 1: Non-FIFO selected (CCR3.FM = 0): Received data is in RDR register FIFO selected (CCR3.FM = 1): The quantity of receive data written in receive-FIFO is equal to or greater than the specified receive triggering number.	R

Note: S-TYPE-3, P-TYPE-3

ERS bit (Error Signal Status Flag)

[Setting condition]

- When an error signal low is sampled.

[Clearing condition]

- When write 1 to CFCLR.ERSC.

DCMF bit (Data Compare Match Flag)

The DCMF bit indicates that SCI detects the match to the comparison data (CCR4.CMPD) with receive data.

Clearing the CCR0.RE bit to 0 does not affect the DCMF flag, which retains its previous state.

[Setting condition]

- Matched to the comparison data (CCR4.CMPD) with receive data, while CCR0.DCME = 1.

[Clearing condition]

- When write 1 to CFCLR.DCMFC.

DPER bit (Data Compare Match Parity Error Flag)

The DPER bit indicates that a parity error occurred at address match detection (reception data match detection).

Clearing the CCR0.RE bit to 0 does not affect the DPER flag, which retains its previous state.

[Setting condition]

- When a parity error is detected by the frame in which an address match was detected.

[Clearing condition]

- When writing 1 to CFCLR.DPERC.

DFER bit (Data Compare Match Framing Error Flag)

The DFER bit indicates that a framing error occurred at address match detection (reception data match detection).

Clearing the CCR0.RE bit to 0 does not affect the DFER flag, which retains its previous state.

[Setting condition]

- When a stop bit of the frame in which an address match was detected is 0.
  When it is a 2-stop mode, the 1st bit of the stop bit only determines whether it is 1 and does not check the 2nd bit of the stop bit.

[Clearing condition]

- When writing 1 to CFCLR.DFERC.

ORER bit (Overrun Error Flag)

The ORER bit indicates that an overrun error has occurred during reception and the reception ends abnormally.

Clearing the CCR0.RE bit to 0 does not affect the ORER flag, which retains its previous state. In Simple IIC mode, the ORER bit is not used.

[Setting condition with non-FIFO mode (CCR3.FM = 0)]

- When the next data is received before reading out RDR with no-error reception data stored in RDR.
  In RDR, receive data prior to an overrun error occurrence is retained, but data received after the overrun error occurrence is lost. When the ORER flag is set to 1, reception data is not forwarded to RDR register.

  Note: In Clock synchronous mode and Simple SPI mode, serial reception is stopped.

[Setting condition with FIFO mode (CCR3.FM = 1)]

- When the next serial reception is completed while the receive FIFO is full of 16 receive data.

[Clearing condition]

- When write 1 to CFCLR.ORERC.

MFF bit (Mode Fault Error Flag)

The MFF bit indicates mode fault errors. In a multi-master configuration, determine the mode fault error occurrence by reading the MFF flag.

[Setting condition]

- Input on the SSn pin being at the low level during master operation (CCR3.CKE[1:0] = 00 or 01) in Simple SPI mode.

[Clearing condition]

- When writing 1 to CFCLR.MFFC.

PER bit (Parity Error Flag)

The PER bit indicates that a parity error has occurred during reception and the reception ends abnormally.

Clearing the CCR0.RE bit to 0 does not affect the PER flag, which retains its previous state.

In Clock synchronous mode, Simple SPI mode, and Simple IIC mode, the PER bit is not used.

[Setting condition]

- When a parity error is detected during reception. In FIFO select mode, when one or more parity error is detected in receive-FIFO data.
  In non-FIFO mode, although receive data when the parity error occurs is transferred to RDR, no SCIn_RXI interrupt request occurs. Note that when the PER flag is set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When writing 1 to CFCLR.PERC.

FER bit (Framing Error Flag)

The FER bit indicates that a framing error has occurred during reception and the reception ends abnormally.

Clearing the CCR0.RE bit to 0 does not affect the FER flag, which retains its previous state.

In Clock synchronous mode, Simple SPI mode, and Simple IIC mode, the FER bit is not used.

[Setting condition]

- When 0 is sampled as the stop bit during reception. In FIFO select mode, when one or more framing error is detected in receive-FIFO data. In Manchester mode, when both sampling results (1⁄4 and 3⁄4 sampling points) are not 1 for 1 stop bit. In Simple LIN mode, even if a condition that changes to 1 occurs when XCR1.SDST = 1, the FER set timing is delayed up to the Break Field evaluation timing at the longest, since it may be a Break Field. If an edge is detected on the RXDn signal before the Break Field evaluation timing, FER is detected. If no edge is detected in the RXDn signal before the Break Field evaluation timing, Break Field is detected.
  In 2-stop-bit mode, only the first stop bit is checked whether it is 1, but the second stop bit is not checked. In non-FIFO mode, although receive data when the framing error occurs is transferred to RDR, no SCIn_RXI interrupt request occurs. In addition, when the FER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When writing 1 to CFCLR.FERC.

TDRE bit (Transmit Data Empty Flag)

[Non-FIFO selected (CCR3.FM = 0)]

The TDRE bit indicates the presence of transmit data in the TDR register.

The condition of CCR0.TE = 0 has priority over the condition of 0.

If other conditions that become 1 and conditions that become 0 are satisfied at the same time, the TDRE flag is set to 0.

[Setting condition]

- When CCR0.TE is 0.
- When data is transmitted from the TDR register to the TSR register.

[Clearing condition]

- When writing 1 to CFCLR.TDREC.
- When the transmission data is written to the TDR register during CCR0.TE = 1.

[FIFO selected (CCR3.FM = 1)]

The TDRE bit indicates that data has been transferred from the transmit-FIFO (TDR) into the transmit shift register (TSR), the quantity of data in transmit-FIFO has fallen below the specified transmit triggering number.

When the condition which becomes 1 and the condition which becomes 0 are formed at the same time, TDRE flag is 0. After that, when the number of data stored in transmit-FIFO is evaluated, and if that is equal to or less than TTRG value, TDRE is set to 1 after 1 PCLK.

[Setting condition]

- When the quantity of transmit data written in transmit-FIFO is equal to or less than the specified transmit triggering number*1.

Note 1. The transmit-FIFO is FIFO register of 16 stages, the maximum number of data that can be written when the TDRE flag is 1 is indicated in 0x10 - FTSR.T[5:0]. Even if more data is written, data is discarded.

[Clearing condition]

- When writing 1 to CFCLR.TDREC.
- When the transmission data is written to transmit-FIFO by the DTC or DMAC (the last block transfer of block transfer).

TEND bit (Transmit End Flag)

[Non-FIFO selected (CCR3.FM = 0), and not Smart card interface mode (CCR3.MOD[2:0] ≠ 001)]

The TEND bit indicates completion of transmission.

[Setting condition]

- When CCR0.TE is 0.
- When the CCR0.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When the TDR register is not updated at the time of transmission of the tail-end bit of a character being transmitted.
- When the TDR register is not updated at the end of DE negate time with DE control function enable (CCR3.DEN = 1).
- When Break Field is sending.

[Clearing condition]

- After the synchronization delay time has elapsed since the transmission data was written to the TDR register during CCR0.TE = 1.
- When writing 1 to CFCLR.TDREC during CCR0.TE = 1.

[Non-FIFO selected (CCR3.FM = 0), and Smart card interface mode (CCR3.MOD [2:0] = 001)]

With no error signal from the receiving side, this bit is set to 1 when additional data for transfer is ready to be transferred to the TDR register.

[Setting condition]

- When CCR0.TE is 0.
- When the CCR0.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When a specified period has elapsed after the latest transmission of 1 byte, the ERS flag is 0, and the TDR register is not updated. The set timing is determined by register settings as listed below.1
  When GM = 0 and BLK = 0, 12.5etu after the start of transmission
  When GM = 0 and BLK = 1, 11.5etu after the start of transmission
  When GM = 1 and BLK = 0, 11.0etu after the start of transmission
  When GM = 1 and BLK = 1, 11.0etu after the start of transmission

[Clearing condition]

- After the synchronization delay time has elapsed since the transmission data was written to the TDR register during CCR0.TE = 1.
- When writing 1 to CFCLR.TDREC during CCR0.TE = 1.

[FIFO selected (CCR3.FM = 1)]

The TEND bit indicates that the transmit-FIFO does not contain valid data when transmitting the last bit of a serial character, so the transmission is halted.

[Setting condition]

- TEND is set to 1 when transmit-FIFO does not contain transmit data when the last bit of a one-byte serial character is transmitted.
- When the TDR register is not updated at the end of DE negate time with DE control function enable (CCR3.DEN = 1).

[Clearing condition]

- After the synchronization delay time has elapsed since the transmission data was written to the TDR register during CCR0.TE = 1.

RDRF bit (Receive Data Full Flag)

[Non-FIFO selected (CCR3.FM = 0)]

The RDRF bit indicates the presence of receive data in the RDR register.

[Setting condition]

- When the reception ends normally, and receive data is forwarded from the RSR register to the RDR register.

[Clearing condition]

- When writing 1 to CFCLR.RDRFC.
- When the read data is read from the RDR register.

[FIFO selected (CCR3.FM = 1)]

The RDRF bit indicates that receive data has been transferred to the receive FIFO data register (RDR), and the quantity of data in receive-FIFO is equal to or greater than the specified receive triggering number. When FCR.RTRG is set to 0, RDRF is set if the quantity of data in receive-FIFO is greater than or equal to 1.

[Setting condition]

- RDRF is set to 1 when the quantity of receive data in receive-FIFO is equal to or greater than the specified receive triggering number*1.

Note 1. Since the receive-FIFO is 16 stage FIFO register, the maximum quantity of data that can be read when RDRF is 1 is equivalent to the specified receive data count number (FDR.R[5:0]). If an attempt is made to read after all the data in receive-FIFO has been read, the data is undefined.

[Clearing condition]

- When writing 1 to CFCLR.RDRFC.
- When the reception data is read form receive-FIFO by the DTC or DMAC (the last block transfer of a block transfer).

When the condition which becomes 1 and the condition which becomes 0 are formed at the same time, RDRF flag is 0. After that, when the number of stored data in receive-FIFO is evaluated, and if that is equal to or greater than RTRG value, RDRF is set to 1 after 1 PCLK.

Note: When Non-FIFO is selected, except when interruption communication, RDRF and TDRE should not be cleared by CFCLR register.

38.2.18 ISR : Simple IIC Status Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x4C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	IICSTIF	—	—	IICACKR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IICACKR	ACK Reception Data Flag 0: ACK received 1: NACK received	R
1	—	This bit is read as 0.	R
2	—	The read value is undefined.	R
3	IICSTIF	Issuing of Start, Restart, or Stop Condition Completed Flag 0: There are no requests for generating conditions or a condition is being generated. 1: A start, restart, or stop condition is completely generated.	R
5:4	—	The read value is undefined.	R

Bit	Symbol	Function	R/W
31:6	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

IICACKR bit (ACK Reception Data Flag)

Received ACK and NACK bits can be read from this bit.

The IICACKR flag is updated at the rising of SCLn clock for the ACK/NACK receiving bit.

IICSTIF bit (Issuing of Start, Restart, or Stop Condition Completed Flag)

After generating a condition, this bit indicates that the generation is completed. When using the IICSTAREQ, IICRSTAREQ, or IICSTPREQ bit to cause generation of a condition, do so after setting the IICSTIF flag to 0.

When the IICSTIF flag is 1 while an interrupt request is enabled by setting the CCR0.TEIE bit, an STIn request is output.

[Setting condition]

- Completion of the generation of a start, restart, or stop condition (however, in cases where this conflicts with any of the conditions for the flag becoming 0 listed below, the clearing condition takes precedence)

[Clearing condition]

- Writing 1 to ICFCLR.IICSTIFC bit
- When operation is not in Simple IIC mode
- Writing 0 to CCR0.TE bit

38.2.19 FRSR : FIFO Receive Status Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x50

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	FNUM[5:0]						—	—	PNUM[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	R[5:0]						—	—	—	—	—	—	—	DR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0

Bit	Symbol	Function	R/W
0	DR	Receive Data Ready Flag 0: Receiving is in progress, or no received data remains in the receive-FIFO after a successful reception (receive-FIFO is empty) 1: The following receive data is not received for a fixed period after storing data under the threshold in the receive-FIFO	R
1	—	The read value is undefined.	R
7:2	—	These bits are read as 0.	R
13:8	R[5:0]	Receive-FIFO Data Count Valid in Asynchronous mode (including Multiprocessor mode), Clock synchronous mode, and Simple SPI mode, when CCR3.FM is 1. Indicate the quantity of receive data stored in receive-FIFO	R
15:14	—	These bits are read as 0.	R
21:16	PNUM[5:0]	Parity Error Count Valid only in Asynchronous mode Indicates the quantity of data with a parity error for the receive data stored in the Receive FIFO Data register.	R

Bit	Symbol	Function	R/W
23:22	—	These bits are read as 0.	R
29:24	FNUM[5:0]	Framing Error Count Valid only in Asynchronous mode. Indicates the quantity of data with a framing error for the receive data stored in the Receive FIFO Data register.	R
31:30	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

DR bit (Receive Data Ready Flag)

The DR bit indicates that the quantity of data stored in the Receive FIFO Data register (RDR) falls below the specified receive triggering number, and that no next data has been received yet after the elapse of 15 etu from the last stop bit in Asynchronous mode. This bit is valid only Asynchronous mode (including Multiprocessor mode) and FIFO selected. In other mode, this bit does not set to 1.

[Setting conditions]

- DR is set to 1 when the following conditions are met:
  - After receive FIFO Data register (RDR) receives less data than the specified receive triggering number, no next data has been received yet after the elapse of 15 etu*1 from the last stop bit
  - CSR.FER, PER flags are 0.

Note 1. This is equivalent to one and a half (1.5) frames in the 8-bit format with one stop bit (etu: elementary time unit).

[Clearing conditions]

- When all receive data in the receive FIFO Data (RDR register) is read and 1 is written to FFCLR.DRC.
- When CCR3.FM bit is 0.

R[5:0] bits (Receive-FIFO Data Count)

The R[5:0] bits indicate the quantity of receive data stored in the Receive FIFO Data register.

0x00 means no receive data. 0x10 means receive-FIFO is full.

PNUM[5:0] bits (Parity Error Count)

The value indicates the quantity of data stored in the Receive FIFO Data register with a parity error.

FNUM[5:0] bits (Framing Error Count)

The value indicates the quantity of data stored in the Receive FIFO Data registers with a framing error.

38.2.20 FTSR : FIFO Transmit Status Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x54

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	T[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	T[5:0]	Transmit-FIFO Data Count Valid in Asynchronous mode (including Multiprocessor mode), Clock synchronous mode, and Simple SPI mode, when CCR3.FM is 1. Indicate the quantity of non-transmit data stored in transmit-FIFO	R
31:6	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

T[5:0] bits (Transmit-FIFO Data Count)

The T[5:0] bits indicate the quantity of non-transmitted data stored in transmit-FIFO.

0x00 means no non-transmit data. 0x10 means transmit-FIFO is full.

38.2.21 MSR : Manchester Status Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x58

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	RSYNC	—	MER	—	SBER	SYER	PFER
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PFER	Preface Error Flag This bit is set when a preface error (pattern mismatch) is detected 0: No preface error detected 1: Preface error detected	R
1	SYER	SYNC Error Flag This bit is set when no edge is detected in the adjustable range during receive retiming 0: No receive SYNC error detected 1: Receive SYNC error detected	R
2	SBER	Start Bit Error Flag This bit is set when a pattern mismatch in the start bit area is detected 0: No start bit error detected 1: Start bit error detected	R
3	—	This bit is read as 0.	R
4	MER	Manchester Error Flag Valid for Manchester mode only 0: No Manchester error occurred 1: Manchester error has occurred	R
5	—	This bit is read as 0.	R
6	RSYNC	Receive SYNC Data Bit It is valid when MCR.SBSEL = 1 in Manchester mode, 0 is read otherwise. 0: The received start bit is DATA SYNC 1: The received start bit is COMMAND SYNC	R
31:7	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

PFER bit (Preface Error Flag)

The PFER bit indicates that a preface error was detected when receiving frames in Manchester mode.

Even when the RE bit in CCR0 is set to 0 (serial reception is disabled), the PFER flag is not affected and retains its previous value.

[Setting condition]

- When detecting a preface error when receiving frames in Manchester mode
  The following operations are performed when a preface error occurs.
  - When MCR.PFEREN = 1
    The received data is not transferred to the RDR register and no SCIn_RXI interrupt request occurs. Instead, an SCIn_ERI interrupt request occurs. Note that when the PFER flag is set to 1, the subsequently received data is not transferred to the RDR register.
  - When MCR.PFEREN = 0
    The received data is transferred to the RDR register and an SCIn_RXI interrupt request occurs. An SCIn_ERI interrupt request is not generated. The subsequent receive operations are not affected even with the PFER flag is set to 1.

[Clearing condition]

- Writing 1 to MFCLR.PFERC.

SYER bit (SYNC Error Flag)

The SYER bit indicates that a receive SYNC error was detected when receiving frames in Manchester mode with MCR.ERTEN = 1 (Manchester edge retiming enabled).

Even when the RE bit in CCR0 is set to 0 (serial reception is disabled), the SYER flag is not affected and retains its previous value.

[Setting condition]

- When detecting a receive SYNC error when receiving frames in Manchester mode
  The following operations are performed when a receive SYNC error occurs.
  - When MCR.SYEREN = 1
    Although the received data is transferred to the RDR register, no SCIn_RXI interrupt request occurs. Instead, an SCIn_ERI interrupt request occurs. Note that when the SYER flag is set to 1, the subsequently received data is not transferred to the RDR register.
  - When MCR.SYEREN = 0
    The received data is transferred to the RDR register and an SCIn_RXI interrupt request occurs. An SCIn_ERI interrupt request is not generated. The subsequent receive operations are not affected even with the SYER flag is set to 1.

[Clearing condition]

- Writing 1 to MFCLR.SYERC.

SBER bit (Start Bit Error Flag)

The SBER bit indicates that a start bit error was detected when receiving frames in Manchester mode.

Even when the RE bit in CCR0 is set to 0 (serial reception is disabled), the SBER flag is not affected and retains its previous value.

[Setting condition]

- When detecting a start bit error when receiving frames in Manchester mode
  The following operations are performed when a start bit error occurs.
  - When MCR.SBEREN = 1
    The received data is not transferred to the RDR register and no SCIn_RXI interrupt request occurs. Instead, an SCIn_ERI interrupt request occurs. Note that when the SBER flag is set to 1, the subsequently received data is not transferred to the RDR register.
  - When MCR.SBEREN = 0

The received data is transferred to the RDR register and an SCIn_RXI interrupt request occurs. An SCIn_ERI interrupt request is not generated. The subsequent receive operations are not affected even with the SBER flag is set to 1.

[Clearing condition]

- Writing 1 to MFCLR.SBERC.

MER bit (Manchester Error Flag)

When data is received in Manchester mode, Manchester error is detected and displayed. Even when the RE bit in CCR0 is set to 0 (serial reception is disabled), the MER flag is not affected and retains its previous value.

[Setting conditions]

- When receiving in Manchester mode and detecting Manchester code error in data area of received frame.
  Received data when an error occurs is transferred to the RDR register, but the SCIn_RXI interrupt request is not generated and the SCIn_ERI interrupt request is generated.
  When the Manchester error flag is set to 1, subsequent receive data is not transferred to the RDR register.
  For details on Manchester error, section 38.5.11. Errors in Manchester Mode.

[Clearing condition]

- Writing 1 to MFCLR.MERC.

RSYNC bit (Receive SYNC Data Bit)

When Manchester mode (CCR3.MOD[2:0] = 101b) and MCR.SBSEL = 1, this bit indicates the type of SYNC of the received start bit. For other settings, it is fixed to 0.

38.2.22 XSR0 : Simple LIN Status Register 0

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x5C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CF1RD[7:0]								CF0RD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AEDF	COF	PIBDF	CF1MF	CF0MF	BFDF	BCDF	BFOF	—	—	—	—	—	—	RXDSF	SFSF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SFSF	Start Frame Status Flag 0: Start Frame detection disabled or Start Frame detection complete 1: Before Start Frame detection or during detection	R*1
1	RXDSF	RXDn Input Status Flag 0: RXDn input to SCI is enabled 1: RXDn input to SCI is disabled	R*1
7:2	—	These bits are read as 0.	R
8	BFOF	Break Field Output Completion Flag 0: Break Field is being output or is not being output 1: Completed BF output	R
9	BCDF	Bus Conflict Detection Flag 0: No bus collision detected 1: Detected a bus collision	R

Bit	Symbol	Function	R/W
10	BFDF	Break Field Detection Flag 0: Break Field is not detected 1: Break Field is detected	R
11	CF0MF	Control Field 0 Compare Match Flag 0: Control Field 0 received data does not match the setting data 1: Control Field 0 received data matches the setting data	R
12	CF1MF	Control Field 1 Compare Match Flag 0: Control Field 1 received data does not match the setting data 1: Control Field 1 received data matches the setting data	R
13	PIBDF	Priority Interrupt Bit Detection Flag 0: Priority interrupt bit is not detected 1: Priority interrupt bit is detected	R
14	COF	Counter Overflow Flag 0: Break Field detection counter has not overflowed 1: Break Field detection counter overflowed	R
15	AEDF	Active Edge Detection Flag 0: No valid edge detected 1: Detected a valid edge	R
23:16	CF0RD[7:0]	Control Field 0 received data Control Field 0 received data.	R
31:24	CF1RD[7:0]	Control Field 1 received data Control Field 1 received data.	R

Note: S-TYPE-3, P-TYPE-3

Note 1. When PCLK is faster than TCLK, the flag set timing is delayed from the receive data full interrupt (SCIn_RXI) output. To see this flag under these conditions, wait at least 1 TCLK cycle after the receive data full interrupt (SCIn_RXI) before reading this register.

SFSF bit (Start Frame Status Flag)

Indicates whether detect Start Frame is being detected.

[Setting condition]

- When 1 is written to XCR1.SDST.
- When a Break Field is detected in the Control Field 0/Control Field 1/Information Field phase and the transition to the Control Field 0 or Control Field 1 reception state occurs.

[Clearing condition]

- When XCR1.SDST is 0.
- When Start Frame detection is completed.

RXDSF bit (RXDn Input Status Flag)

Indicates the RXDn input status to the SCI core. When the RXDSF bit is 1, RXDn input is received only by the simple LIN module and the Break Field is detected and is not input to the SCI core.

BFOF bit (Break Field Output Completion Flag)

Indicates the completion of Break Field output.

The BFOF bit can be cleared to 0 by writing 1 to XFCLR.BFOC.

BCDF bit (Bus Conflict Detection Flag)

Indicates the detection of bus conflict in Simple LIN transmit operation.

The BCDF bit can be cleared to 0 by writing 1 to XFCLR.BCDC.

BFDF bit (Break Field Detection Flag)

Indicates Break Field detection.

The BFDF bit can be cleared to 0 by writing 1 to XFCLR.BFDC.

CF0MF bit (Control Field 0 Compare Match Flag)

Indicates compare match of Control Field 0 and compare data.

The CF0MF bit can be cleared to 0 by writing 1 to XFCLR.CF0MC.

CF1MF bit (Control Field 1 Compare Match Flag)

Indicates compare match detection of Control Field 1 and compare data.

The CF1MF bit can be cleared to 0 by writing 1 to XFCLR.CF1MC.

PIBDF bit (Priority Interrupt Bit Detection Flag)

Indicates compare match detection of Control Field 1 and priority interrupt bit.

The PIBDF bit can be cleared to 0 by writing 1 to XFCLR.PIBDC.

COF bit (Counter Overflow Flag)

Indicates that the 16-bit counter in the simple LIN module has overflowed.

The COF bit can be cleared to 0 by writing 1 to XFCLR.COFC.

AEDF bit (Active Edge Detection Flag)

Indicates active edge detection.

The AEDF bit can be cleared to 0 by writing 1 to XFCLR.AEDC and when read out XSR1.TCNT[15:0].

CF0RD[7:0] bits (Control Field 0 received data)

Stores the received data with a Control Field 0 match detected.

CF1RD[7:0] bits (Control Field 1 received data)

Stores the received data with a Control Field 1 match detected.

38.2.23 XSR1 : Simple LIN Status Register 1

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x60

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TCNT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TCNT[15:0]	Timer Count Capture Value Stores the 16-bit counter capture value. The initial value is 0000.	R
31:16	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

TCNT[15:0] bits (Timer Count Capture Value)

Stores the capture value of the 16-bit counter of the Simple LIN module.

- When sending Start Frame
  The XSR1 holds the previous value.
- When receiving Start Frame with bit rate measurement disabled

If a Break Field is detected in the Break Field detection state (see Figure 38.81), the Break Field length is captured and held (counter value is captured at the rising edge of RXDn). If a Break Field is detected in a state other than the Break Field detection state, the previous value is retained.
If the counter overflows, it is not captured.

- When receiving Start Frame with bit rate measurement enabled
The count value is captured and held at the valid edge (both RXDn edges). However, in the Break Field detection state, the count value is not captured even if a valid edge occurs. Counter capture value retention is canceled by the XSR1 read. Even if a valid edge occurs before reading, the counter value is not captured.

38.2.24 CFCLR : Common Flag Clear Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x68

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RDRFC	—	TDREC	FERC	PERC	MFFC	—	ORERC	—	—	—	—	—	DFERC	DPERC	DCMFC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	ERSC	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	The write value should be 0.	W
4	ERSC	ERS Clear Bit Setting this bit to 1 clears the CSR.ERS bit. The read value is always 0.	W
15:5	—	The write value should be 0.	W
16	DCMFC	DCMF Clear Bit Setting this bit to 1 clears the CSR.DCMF bit. The read value is always 0.	W
17	DPERC	DPER Clear Bit Setting this bit to 1 clears the CSR.DPER bit. The read value is always 0.	W
18	DFERC	DFER Clear Bit Setting this bit to 1 clears the CSR.DFER bit. The read value is always 0.	W
23:19	—	The write value should be 0.	W
24	ORERC	ORER Clear Bit Setting this bit to 1 clears the CSR.ORER bit. The read value is always 0.	W
25	—	The write value should be 0.	W
26	MFFC	MFF Clear Bit Setting this bit to 1 clears the CSR.MFF bit. The read value is always 0.	W
27	PERC	PER Clear Bit Setting this bit to 1 clears the CSR.PER bit. The read value is always 0.	W
28	FERC	FER Clear Bit Setting this bit to 1 clears the CSR.FER bit. The read value is always	W
29	TDREC	TDRE Clear Bit Setting this bit to 1 clears the CSR.TDRE bit. The read value is always 0.	W
30	—	The write value should be 0.	W
31	RDRFC	RDRF Clear Bit Setting this bit to 1 clears the CSR.RDRF bit. The read value is always 0.	W

Note: S-TYPE-3, P-TYPE-3

38.2.25 ICFCLR : Simple IIC Flag Clear Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x6C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	IICSTIFC	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	The write value should be 0.	W
3	IICSTIFC	IICSTIF Clear Bit Setting this bit to 1 clears the ISR.IICSTIF bit. The read value is always 0.	W
31:4	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

38.2.26 FFCLR : FIFO Flag Clear Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x70

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DRC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DRC	DR Clear Bit Setting this bit to 1 clears the FRSR.DR bit. The read value is always 0.	W
31:1	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

38.2.27 MFCLR : Manchester Flag Clear Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x74

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	MERC	—	SBERC	SYERC	PFERC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PFERC	PFER Clear Bit Setting this bit to 1 clears the MSR.PFER bit. The read value is always 0.	W
1	SYERC	SYER Clear Bit Setting this bit to 1 clears the MSR.SYER bit. The read value is always 0.	W
2	SBERC	SBER Clear Bit Setting this bit to 1 clears the MSR.SBER bit. The read value is always 0.	W
3	—	The write value should be 0.	W
4	MERC	MER Clear Bit Setting this bit to 1 clears the MSR.MER bit. The read value is always 0.	W
31:5	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

38.2.28 XFCLR : Simple LIN Flag Clear Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x78

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AEDC	COFC	PIBDC	CF1MC	CF0MC	BFDC	BCDC	BFOC	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	The write value should be 0.	W
8	BFOC	BFOF Clear Bit Setting this bit to 1 clears the XSR0.BFOF bit. The read value is always 0.	W
9	BCDC	BCDF Clear Bit Setting this bit to 1 clears the XSR0.BCDF bit. The read value is always 0.	W
10	BFDC	BFDF Clear Bit Setting this bit to 1 clears the XSR0.BFDF bit. The read value is always 0.	W
11	CF0MC	CF0MF Clear Bit Setting this bit to 1 clears the XSR0.CF0MF bit. The read value is always 0.	W

Bit	Symbol	Function	R/W
12	CF1MC	CF1MF Clear Bit Setting this bit to 1 clears the XSR0.CF1MF bit. The read value is always 0.	W
13	PIBDC	PIBDF Clear Bit Setting this bit to 1 clears the XSR0.PIBDF bit. The read value is always 0.	W
14	COFC	COFF Clear Bit Setting this bit to 1 clears the XSR0.COF bit. The read value is always 0.	W
15	AEDC	AEDF Clear Bit Setting this bit to 1 clears the XSR0.AEDF bit and cancels holding the XSR1 register. The read value is always 0.	W
31:16	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

38.2.29 CESR : Communication Enable Status Register

Base address: SCIn_B = 0x4035_8000 + 0x0100 × n (n = 0 to 9)
SCIn_B_NS = 5035_8000 + 0x0100 × n (n = 0 to 9)

Offset address: 0x1C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TIST	—	—	—	RIST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RIST	RE Internal Status 0: RE signal internal state value 0 1: RE signal internal state value 1	R
3:1	—	These bits are read as 0.	R
4	TIST	TE Internal Status 0: TE signal internal state value 0 1: TE signal internal state value 1	R
7:5	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The operation clocks of the communication module and control register can be used asynchronously. Since some control register values are transmitted internally through the synchronization circuit so that they operate correctly even if they are asynchronous, it takes some time for the state to be reflected internally after rewriting the control register.

Communication enable CCR0.TE and CCR0.RE correspond to this register, and when these control bits change from 1 to 0 to rewrite the control bits for the next communication, the TE and RE bits signal it is necessary to rewrite the next control bit after the internal state becomes 0. If a very slow clock is used for the communication module clock, the TE and RE bit states are not reflected internally. At this time, you can check the internal status using this register.

38.3 Operation in Asynchronous Mode

Figure 38.6 shows the general format for asynchronous serial communications. One frame consists of a start bit (low level), transmit or receive data, a parity bit, and stop bits (high level). In asynchronous serial communications, the communications line is usually held in the mark state (high level).

The SCI monitors the communications line. When the SCI detects a start bit, it starts serial communication. The detection condition of the start bit changes according to the CCR3.RXDESEL setting. SCI regards space (Low level) as a start bit when CCR3.RXDESEL is 0. SCI regards a fall edge as a start bit when RXDESEL is 1.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications. Both the transmitter and receiver have a double-buffered structure in addition to FIFO mode, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

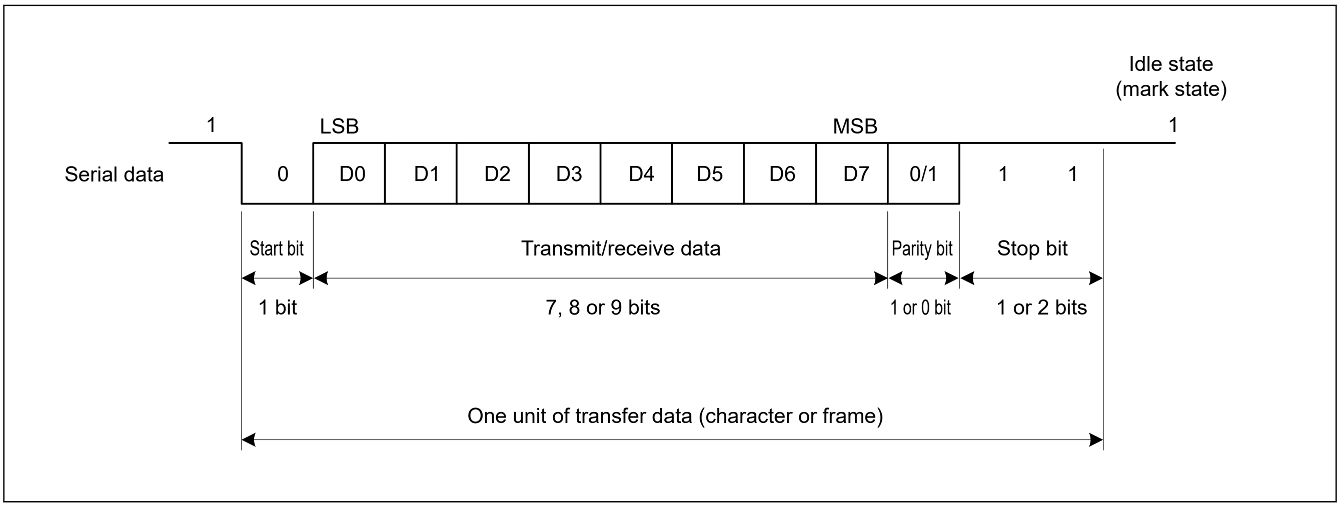


Figure 38.6 Data format in asynchronous serial communications with 8-bit data, parity bit, and 2 stop bits

38.3.1 Serial Data Transfer Format

Table 38.27 lists the serial data transfer formats that can be used in Asynchronous mode. Any of 18 transfer formats can be selected with the CCR1 and CCR3 settings. For details on the multi-processor function, see section 38.4. Multi-Processor Communication Function.

Table 38.27 Serial transfer formats in Asynchronous mode (1 of 2)

CCR3		CCR1		CCR3	Serial transfer format and frame length
CHR[1:0]		PE	MP	STP	1 2 3 4 5 6 7 8 9 10 11 12 13
0	0	0	0	0	ST / 9-bit data / SP
0	0	0	0	1	ST / 9-bit data / SP / SP
0	0	1	0	0	ST / 9-bit data / P / SP
0	0	1	0	1	ST / 9-bit data / P / SP / SP
1	0	0	0	0	ST / 8-bit data / SP
1	0	0	0	1	ST / 8-bit data / SP / SP
1	0	1	0	0	ST / 8-bit data / P / SP
1	0	1	0	1	ST / 8-bit data / P / SP / SP
1	1	0	0	0	ST / 7-bit data / SP
1	1	0	0	1	ST / 7-bit data / SP / SP
1	1	1	0	0	ST / 7-bit data / P / SP
1	1	1	0	1	ST / 7-bit data / P / SP / SP

Table 38.27 Serial transfer formats in Asynchronous mode (2 of 2)

CCR3		CCR1		CCR3	Serial transfer format and frame length
CHR[1:0]		PE	MP	STP	1 2 3 4 5 6 7 8 9 10 11 12 13
0	0	—	1	0	ST, 9-bit data, MPB, SP
0	0	—	1	1	ST, 9-bit data, MPB, SP, SP
1	0	—	1	0	ST, 8-bit data, MPB, SP
1	0	—	1	1	ST, 8-bit data, MPB, SP, SP
1	1	—	1	0	ST, 7-bit data, MPB, SP
1	1	—	1	1	ST, 7-bit data, MPB, SP, SP

ST: Start bit
SP: Stop bit
P: Parity bit
MPB: Multi-processor bit

38.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In Asynchronous mode, the SCI operates on a base clock with a frequency of 16 times*1 the bit rate.

In reception, the SCI samples the falling edge of the start bit using the base clock, and performs internal synchronization.*2

Because receive data is sampled on the rising edge of the 8th pulse*1 of the base clock, data is latched at the middle of each bit (when sampling timing does not adjust (CCR4.ASEN = 0 or CCR4.ASEN = 1 and CCR4.AST[2:0] = 000b)), as shown in Figure 38.7 The reception margin in Asynchronous mode is determined by the following formula (1):

$$M = \left|\left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F)\right| \times 100\ [\%] \quad \ldots \quad \text{Formula (1)}$$

Note: M: Reception margin

N: Ratio of bit rate to clock

See Table 38.6 for bit rate to basic clock ratios.

D: Duty cycle of clock (D = 0.5 to 1.0)

L: Frame length (L = 9 to 13)

F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determined using the following formula:
$M = \{0.5 - 1/(2 \times 16)\} \times 100\ (\%) = 46.875\ \%$

This represents the computed value. Renesas recommends a margin of 20% to 30% in system design.

Note 1. This is an example when CCR2.ABCS is 0, CCR2.ABCSE is 0 and CCR2.ABCSE2 is 0. When CCR2.ABCS is 1, CCR2.ABCSE is 0 and CCR2.ABCSE2 is 0, a frequency of 8 times the bit rate is used as a base clock and receive data is sampled at the rising edge of the 4th pulse of the base clock. When CCR2.ABCSE is 1 and CCR2.ABCSE2 is 0, a frequency of 6 times the bit rate is used as a base clock and receive data is sampled at the rising edge of the 3rd pulse of the base clock. When CCR2.ABCSE is 0 and CCR2.ABCSE2 is 1, a frequency of 4 times the bit rate is used as a base clock and receive data is sampled at the rising edge of the 2nd pulse of the base clock.

Note 2. The determination condition of the start bit is as follows.

The function of adjust sampling timing is OFF (ASEN = 0):

The determination condition of a start bit is that Low beyond half bit length continues. It is same as the sampling timing. In Figure 38.7, Low period should be kept over 8-cycles to detect a start bit. If Low period does not keep over 8-cycles, the IP judges this as a noise. So, the IP does not start reception and wait start bit.

The function of adjust sampling timing is ON (ASEN = 1):

The determination condition of a start bit is that Low keeps up until the sampling timing. Adjusting the sampling timing forward (AJD = 1) increases the possibility of erroneously determining a noise as the start bit.

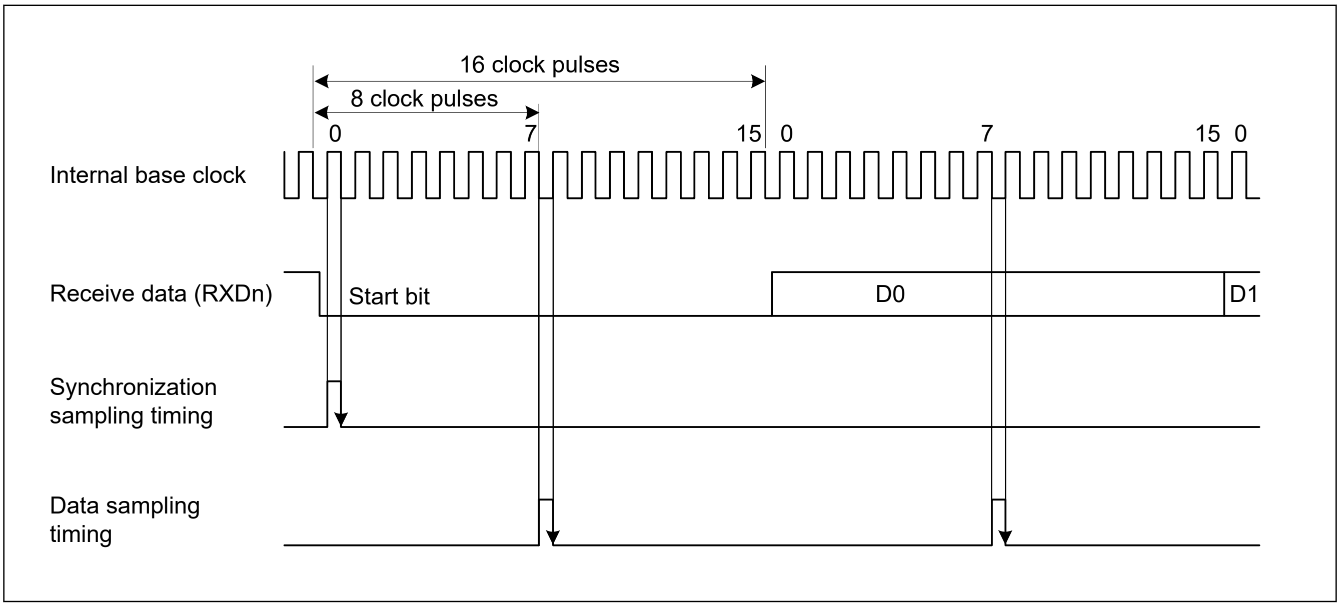


Figure 38.7 Receive data sampling timing in Asynchronous mode

38.3.3 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input to the SCKn pin can be selected as the transfer clock of the SCI, based on the CCR3.CKE[1:0] settings.

When an external clock is input to the SCKn pin, the clock frequency must be 16 times the bit rate (when CCR2.ABCS = 0) or 8 times the bit rate (when CCR2.ABCS = 1).

In addition, when an external clock is specified, the base clock of GPTn (n = 3, 4, 6, 7) can be selected by the CCR2.ACS0.

When the SCI uses its internal clock, the clock can be output from the SCKn pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is configured so that the rising edge of the clock is in the middle of the transmit data, as shown in Figure 38.8.

When the internal clock is selected, the SCKn pin is outputted after setting the CCR0.TE or CCR0.RE bit to 1.

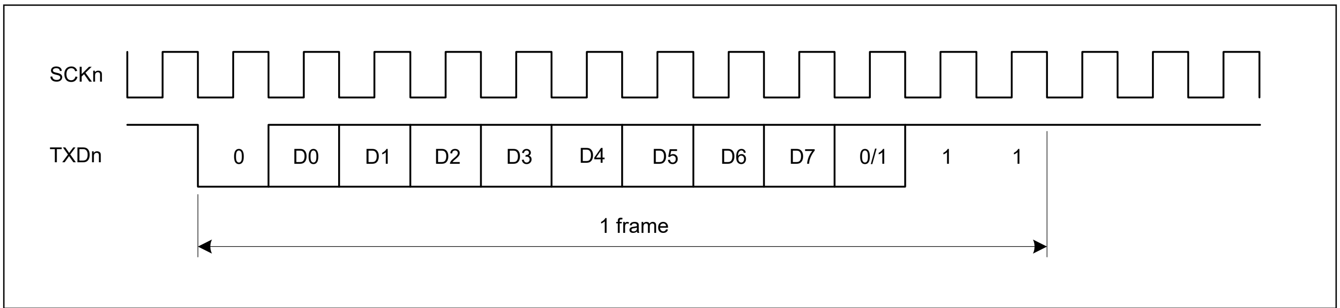


Figure 38.8 Phase relationship between output clock and transmit data in Asynchronous mode when CCR1.PE = 1, CCR3.CHR[1:0] = 10b, MP = 0, and STP = 1

38.3.4 Double-Speed Operation and Frequency of 6 or 4 Times the Bit Rate

When the CCR2.ABCS bit is set to 1, the SCI operates on the bit rate twice that of when ABCS is set to 0. When the CCR2.BGDM bit is set to 1, the cycle of the base clock is half and the bit rate is double that of when BGDM is set to 0. When the CCR3.CKE[1] bit is set to 0 and the on-chip baud rate generator is selected, setting the ABCS and BGDM bits to 1 allows the SCI to operate at a bit rate four times that when the ABCS and BGDM bits are set to 0.

When CCR2.ABCSE is set to 1 and CCR2.ABCSE2 is set to 0, the number of base clock pulses are 6 during a period of 1 bit, and the base clock frequency is half. And RSCI works 16/3 times of bit rate compared with a case of CCR2.ABCS = 0, CCR2.BGDM = 0, CCR2.ABCSE = 0 and CCR2.ABCSE2 = 0.

When CCR2.ABCSE is set to 1 and CCR2.ABCSE2 is set to 1, the number of base clock pulses are 4 during a period of 1 bit, and the base clock frequency is half. And RSCI works 8 times of bit rate compared with a case of CCR2.ABCS = 0, CCR2.BGDM = 0, CCR2.ABCSE = 0 and CCR2.ABCSE2 = 0.

As shown by Formula (1) in section 38.3.2. Receive Data Sampling Timing and Reception Margin in Asynchronous Mode, the reception margin decreases when CCR2.ABCS is set to 1, CCR2.ABCSE is set to 1 or CCR2.ABCSE2 is set to 1. Therefore, if the desired bit rate can be obtained, it is recommended to use the RSCI with CCR2.ABCS set to 1, CCR2.ABCSE set to 1 and CCR2.ABCSE2 set to 1.

38.3.5 CTS and RTS Functions

The CTS function controls transmission using the CTSn pin input. Setting the CCR1.CTSE bit to 1 enables the CTS function. For the functions of CTS and RTS, you can select the alternate setting to set CTSn_RTSn pin as a multiplexed pin that uses either function with one pin or the dedicated setting that uses each function independently with two pins at CTSn pin for CTSn signal and CTSn_RTSn pin for RTSn signal. This setting is done with the CCR1.CTSPEN bit.

When the CTS function is enabled, placing a low level on the CTSn_RTSn pin causes transmission to start.

If FIFO is used and CTSn_RTSn signal is held high before transmission, transmission will not start, so the number of TDR registers written, and the number of data stored are the same (unlike using clock synchronous FIFO).

Driving the CTSn_RTSn pin high while transmission is in progress does not affect transmission of the current frame.

In the RTS function, which uses output on the CTSn_RTSn pin, a low level is output when reception becomes possible. Conditions for output of the low and high levels are shown in this section.

[Conditions for low level output]

Satisfaction of all conditions are listed in this section.

Non-FIFO selected

- The value of the CCR0.RE bit is 1
- When the next reception is possible
  - There are no received data yet to be read and not receiving.
  - CSR.ORER, FER, PER flags are all 0

FIFO selected

- The value of the CCR0.RE bit is 1
- When the next reception is possible
  - When the quantity of receive data written in receive-FIFO(RDR) are less than the setting value of FCR.RSTRG[4:0]
  - CSR.ORER(RDR.ORER) is 0

[Condition for high level output]

- The conditions for low-level output are not satisfied

38.3.6 Address Match (Receive Data Match Detection) Function

The address match function can be used only in Asynchronous mode.

If the CCR0.DCME bit is set to 1*2, when one frame of data is received, the SCI compares that received data with the data set in CCR4.CMPD. If the SCI detects a match to the comparison data (CCR4.CMPD*1) with the received data, the SCI can issue the SCIn_RXI interrupt request.

If the CCR3.MP bit is set to 0, comparison occurs only for valid data in receive format. In multi-processor mode (CCR3.MP bit = 1), if the CCR0.IDSEL bit is set to 1, receive data where the MPB bit is 1 is subject to comparison for address match and receive data where the MPB bit is 0 is always treated as a non-match.

If the CCR0.IDSEL bit is set to 0, SCI performs address match detection regardless of the MPB bit value of the received data.

Until SCI detects a match to the comparison data (CCR4.CMPD*1) with receive data, received data is skipped (discarded), and the SCI cannot detect a parity error or framing error.

When SCI detects a match, the CCR0.DCME bit is automatically cleared, and the CSR.DCMF flag is set to 1. If the CCR0.IDSEL bit is set to 1, the CCR0.MPIE bit is automatically cleared. If CCR0.IDSEL is set to 0, the value of the CCR0.MPIE bit is retained. If the CCR0.RIE bit is set to 1, the SCI issues an SCIn_RXI interrupt request.

If the SCI detects a framing error in the receive data for which a match is detected, the CSR.DFER flag is set to 1, and if the SCI detects a parity error in that frame, the CSR.DPER flag is set to 1. The compared receive data and MPB bit is not stored in the RDR register, and CSR.RDRF remains 0.

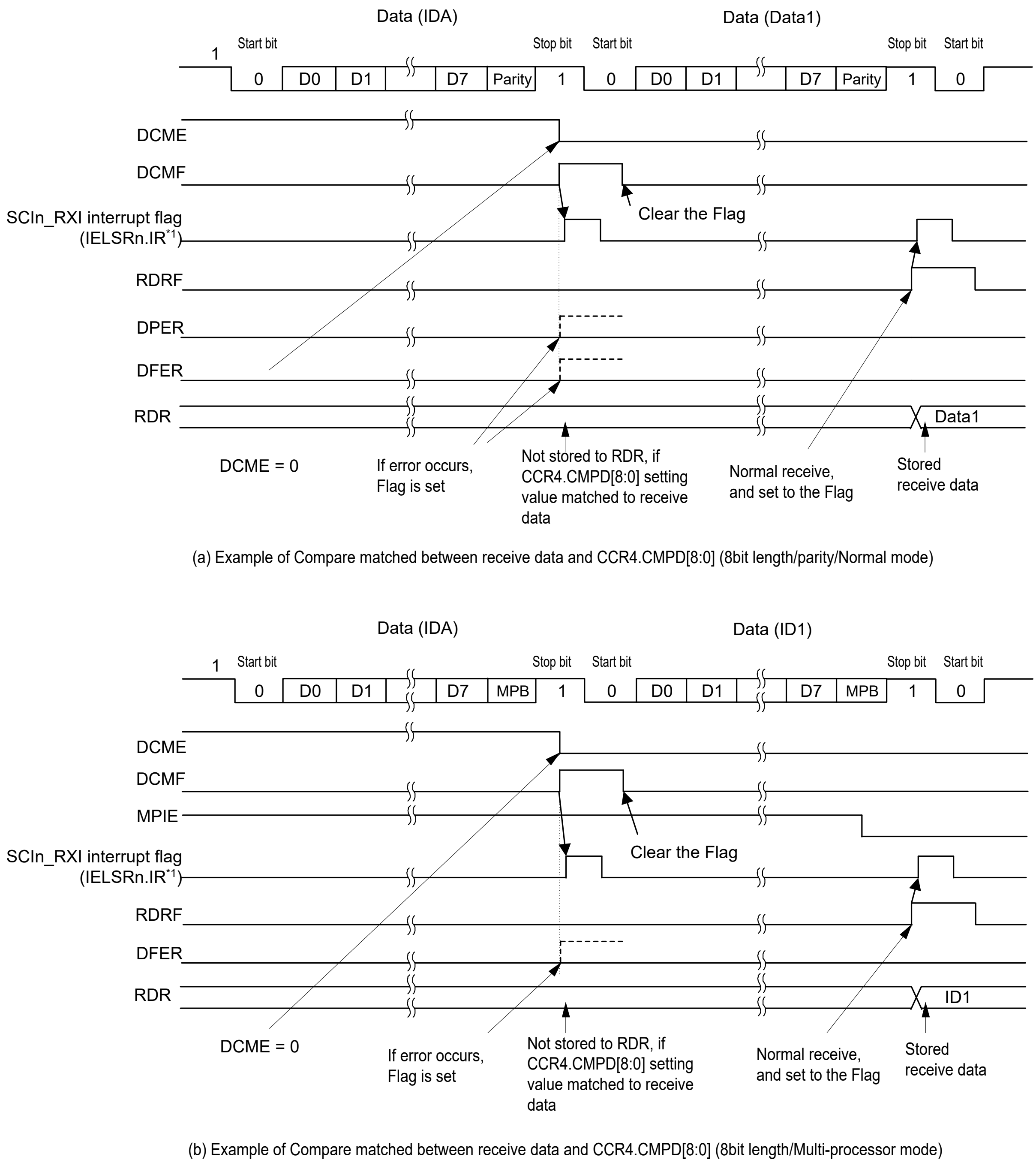
After the SCI detects a match, and CCR0.DCME is automatically cleared, the SCI receives the next data continuously based on the current register setting.

When the CSR.DFER or CSR.DPER flag is set, the address match is not performed. Before enabling the address match function, set the DCCR.DFER and DCCR.DPER flags to 0.

Examples of the address match function are shown in Figure 38.9 and Figure 38.10.

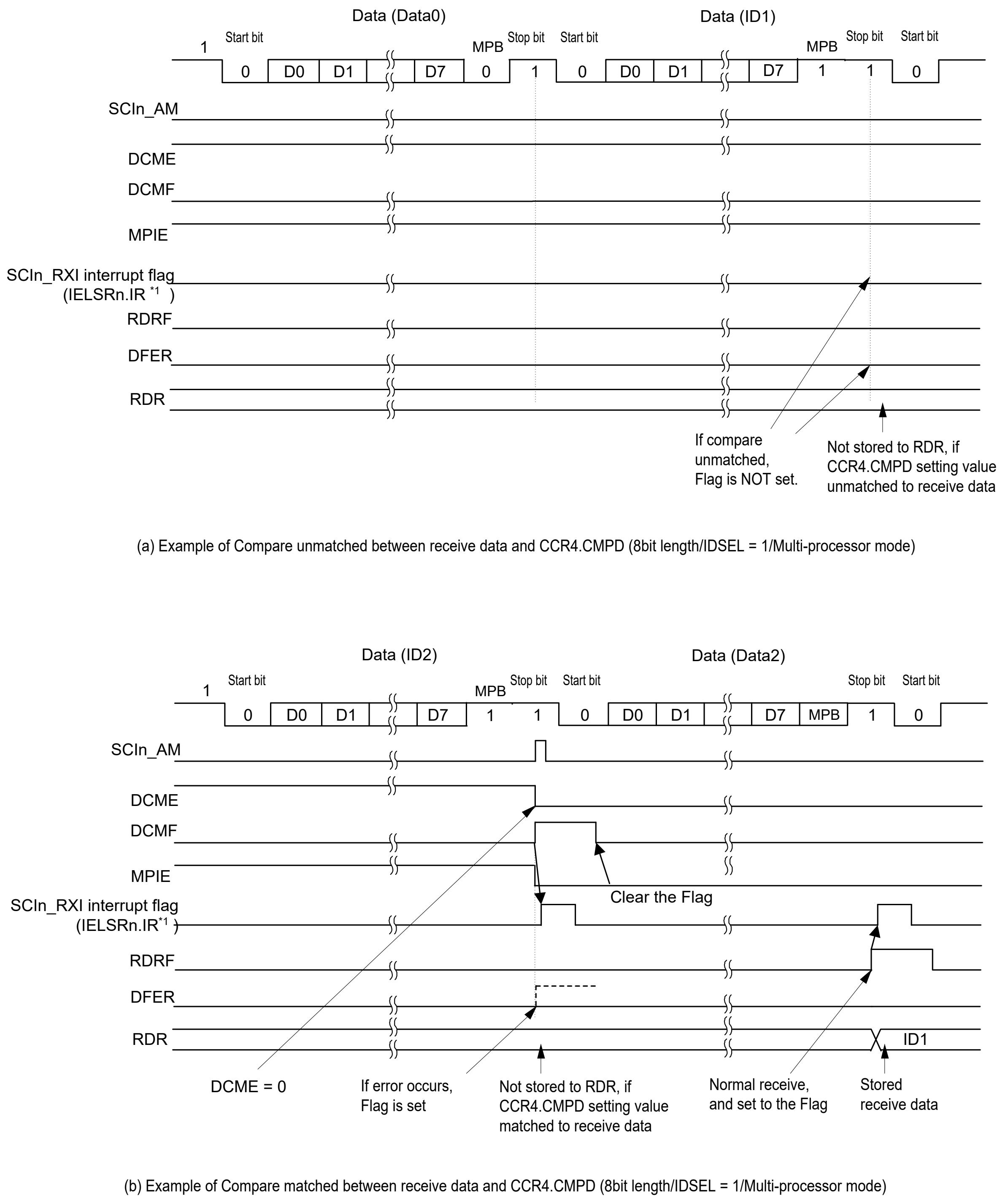
Note 1. This comparative target can select one length of 3 types: CMPD[6:0] with 7-bit length, CMPD[7:0] with 8-bit length, and CMPD[8:0] with 9-bit length.

Note 2. Set the CCR0.DCME bit to 1 before receiving the start bit of the received frame that performs address matching.



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 38.9 Example of address match (1) normal mode and multi-processor mode



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 38.10 Example of address match (2) multi-processor mode

38.3.7 SCI Initialization in Asynchronous Mode

Before transmitting and receiving data, start by writing the initial value 0 to the CCR0.TE and CCR0.RE (or writing the initial value to CCR0), then continue through the SCI initialization procedure (select non-FIFO or FIFO) shown in Table 38.28 and Table 38.29. Whenever the operating mode or transfer format is to be changed, the CCR0.TE and CCR0.RE bits must be initialized before the change is made.

When the external clock is used in Asynchronous mode, ensure that the clock signal is supplied during initialization.

Note: Setting the CCR0.RE bit to 0 initializes neither the ORER, FER, RDRF, RDAT, PER, and DR flags in CSR nor RDR. When FIFO selected, even if the TE bit is set as 0, the TEND flag for the selected FIFO buffer is not initialized.

Please be also careful at the time of change in the operation mode.

Note: Switching the value of the CCR0.TE bit from 0 to 1 while the CCR0.TIE bit is 1 leads to the generation of an SCIn_TXI interrupt request.

Table 38.28 Example flow of SCI initialization in Asynchronous mode with non-FIFO selected

No.	Step Name	Description
1	Start initialization	—
2	Set CCR0	Set CCR0.TEIE, TIE, RIE, TE, RE to 0. If you have not changed from the initial settings, you can skip this step.
3	Set CCR3	Set up following function and communication mode. Driver control function for RS-485, FIFO no-use, Multi-Processor mode, Communication mode (MOD[2:0] = 000b) Transmission / reception format Clock enable (Leave the initial value when outputting the clock) Leave unused bits at their initial values.
4	Set CCR2	Set up the bit-rate-modulation function*1 *2, select clock, set bit rate*2
5	Set CCR1	Set up the Noise-filter function, the loop-back function, communication pin status, the parity check function, and the CTSn_RTSn function.
6	Set CCR4	Set up the adjust sampling timing function and the adjust transmit timing function.
7	Set the I/O port functions	Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.
8	Set CCR3	Set clock enable bit (CKE[1:0]) at this step when outputting the clock. After this register setting, the clock pin will be in the output state immediately.
9	Set CFCLR	Write 1 to the following bits and clear the corresponding flag. CFCLR.RDRFC, FERC, PERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC If it's initialization flow after a reset, you can skip this step.
10	Set CCR0	Set the TE or RE bit to 1. To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time. Setting the TE and RE bits allows TXDn and RXDn to be used.
11	Initialization completed	—

Note 1. If you do not use the bit rate modulation function, you do not need to set it.
Note 2. If you use an external clock, you do not need to set it.

Table 38.29 Example flow of SCI initialization in Asynchronous mode with FIFO selected (1 of 2)

No.	Step Name	Description
1	Start initialization	—
2	Set CCR0	Set CCR0.TEIE, TIE, RIE, TE, RE to 0. If you have not changed from the initial settings, you can skip this step.
3	Set CCR3	Set up following function and communication mode. Driver control function for RS-485, FIFO use, Multi-Processor mode, Communication mode (MOD [2:0] = 000b) Transmission / reception format Clock enable (Leave the initial value when outputting the clock) Leave unused bits at their initial values.
4	Set CCR2	Set up the bit-rate-modulation function*1 *2, select clock, set bit rate*2
5	Set CCR1	Set up the Noise-filter function, the loop-back function, communication pin status, the parity check function, and the CTSn_RTSn function.
6	Set CCR4	Set up the adjust sampling timing function and the adjust transmit timing function.
7	Set FCR	Set the TFRST and RFRST to 1 to empty FIFO. Set the DRES, TTRG[4:0], RTRG[4:0], and RSTRG[4:0] bits

Table 38.29 Example flow of SCI initialization in Asynchronous mode with FIFO selected (2 of 2)

No.	Step Name	Description
8	Set the I/O port functions	Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.
9	Set CCR3	If you select a clock output in Asynchronous mode, set the CKE[1:0] bit here. After setting the register, the clock pin will be in the output state immediately. But the clock operates after setting TE or RE to 1.
10	Set CFCLR	Write 1 to the following bits and clear the corresponding flag. CFCLR.RDRFC, FERC, PERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC If it's initialization flow after a reset, you can skip this step.
11	Set FFCLR	Write 1 to The FFCLR.BRKC, DRC and clear the corresponding flag. If it's initialization flow after a reset, you can skip this step.
12	Set CCR0	Set the TE or RE bit to 1. To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time. Setting the TE and RE bits allows TXDn and RXDn to be used.
13	Initialization completed	—

Note 1. If you do not use the bit rate modulation function, you do not need to set it.
Note 2. If you use an external clock, you do not need to set it.

Figure 38.11 shows an example of the timing when data is transmitted after reset is released, and SCI is set to Asynchronous mode according to Table 38.28 or Table 38.29. As shown in the figure, when the pin function is set to the TXDn pin, the CCR0.TE bit is 0, so the pin is high impedance. When transmit data is written after setting the CCR0.TE bit to 1, data transmission starts. There is a transmission wait time from writing TDR to data transmission starts. In Asynchronous mode, TXDn is high during this period.

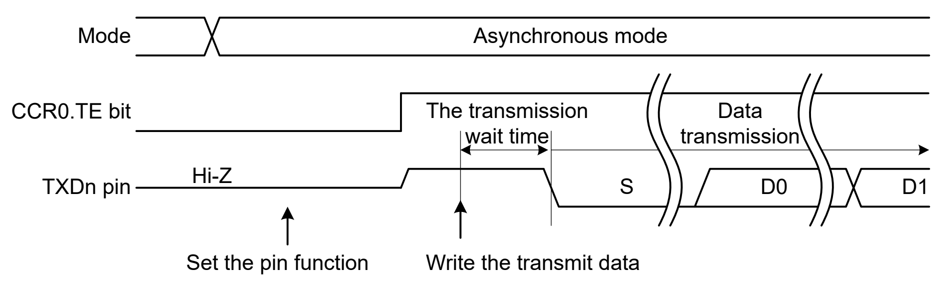


Figure 38.11 Data transmission timing example in Asynchronous mode

38.3.8 Serial Data Transmission in Asynchronous Mode

(1) Non-FIFO selected

Figure 38.12, Figure 38.13, Figure 38.14 and Figure 38.15 show examples of serial transmission in Asynchronous mode.

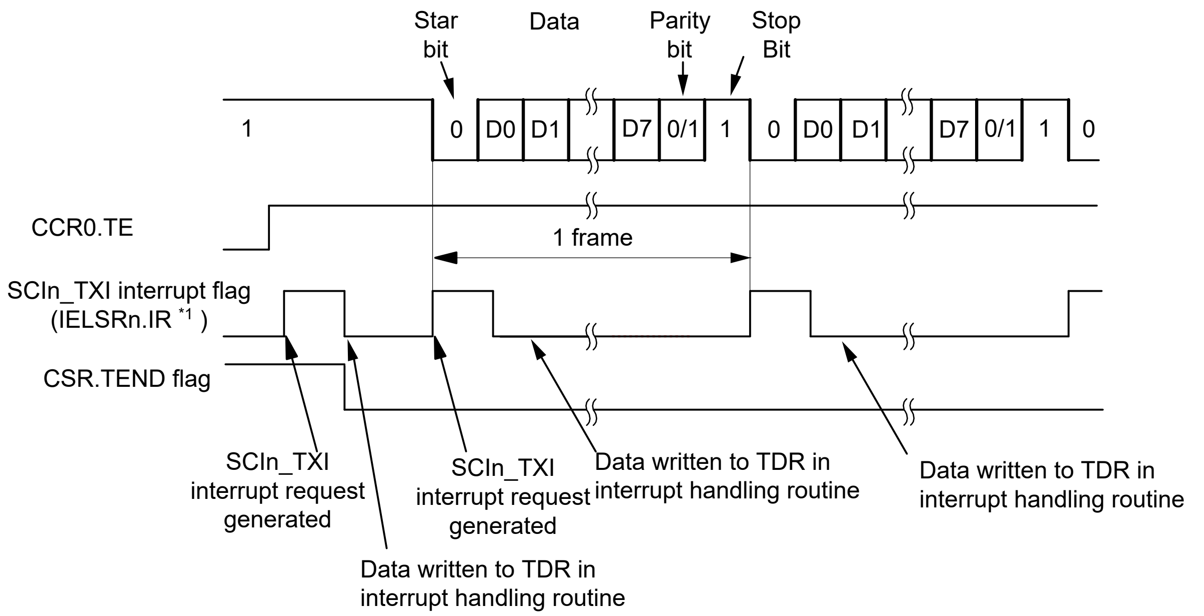
In serial transmission, the SCI operates as described in this section.

1. The SCI transfers data from the TDR register to the TSR register when data is written to TDR in the SCIn_TXI interrupt handling routine.
   The SCIn_TXI interrupt request at the beginning of transmission is generated when the CCR0.TE and CCR0.TIE bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the CCR1.CTSE bit is set to 0 (CTS function is disabled) or a low level on the CTSn_RTSn pin causes data transfer from the TDR register to the TSR register. If the CCR0.TIE bit is 1, an SCIn_TXI interrupt request is generated. Continuous transmission is possible by writing the next transmit data to the TDR register in the SCIn_TXI interrupt handling routine before transmission of the current transmit data is complete. When SCIn_TEI interrupt requests are in use, set the CCR0.TIE bit to 0 (SCIn_TXI interrupt requests are disabled) and the CCR0.TEIE bit to 1 (an SCIn_TEI interrupt request is enabled) after the last of the data to be transmitted is written to the TDR register from the handling routine for SCIn_TXI requests.
3. Data is sent from the TXDn pin in the following order:
   - Start bit

- Transmit data
- Parity bit or multi-processor bit (can be omitted depending on the format)
- Stop bit

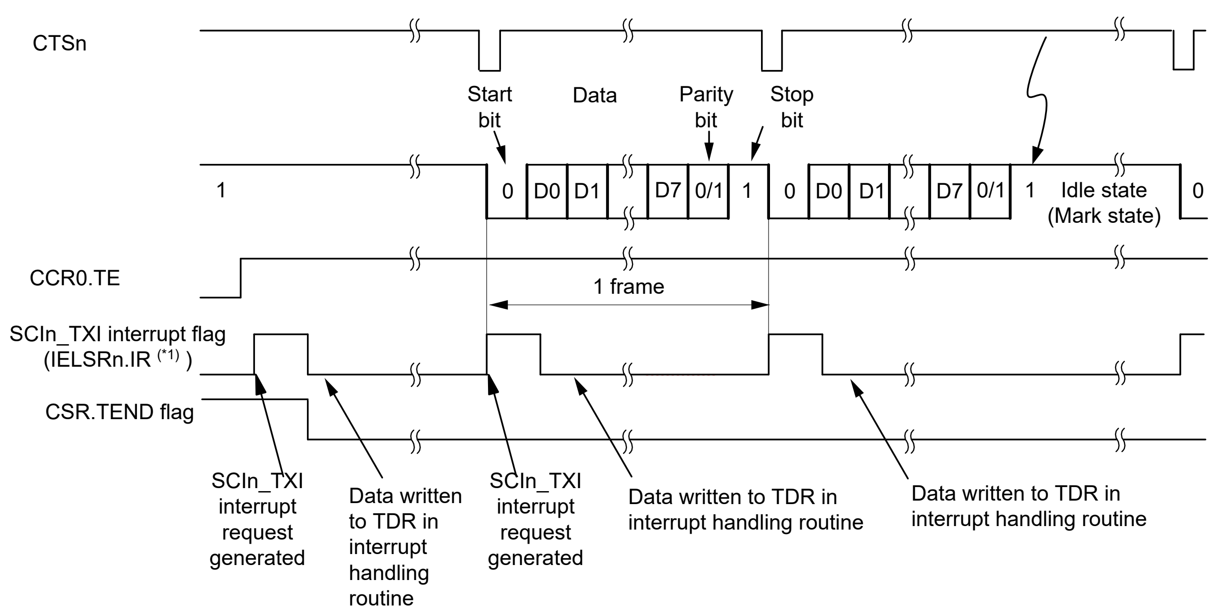
4. The SCI checks for update of the TDR register on output of the stop bit.
5. When the TDR register is updated, setting the CCR1.CTSE bit to 0 (CTS function is disabled) or a low level input on the CTSn_RTSn pin causes transfer of the next transmit data from the TDR register to the TSR register and transmission of the stop bit, after which serial transmission of the next frame starts.
6. If the TDR register is not updated, the CSR.TEND flag is set to 1, the stop bit is sent, and the mark state is entered, in which 1 is output. If the CCR0.TEIE bit is 1, the CSR.TEND flag is set to 1 and an SCIn_TEI interrupt request is generated.

Figure 38.12, Figure 38.13, Figure 38.14 and Figure 38.15 show examples of serial transmission in Asynchronous mode.



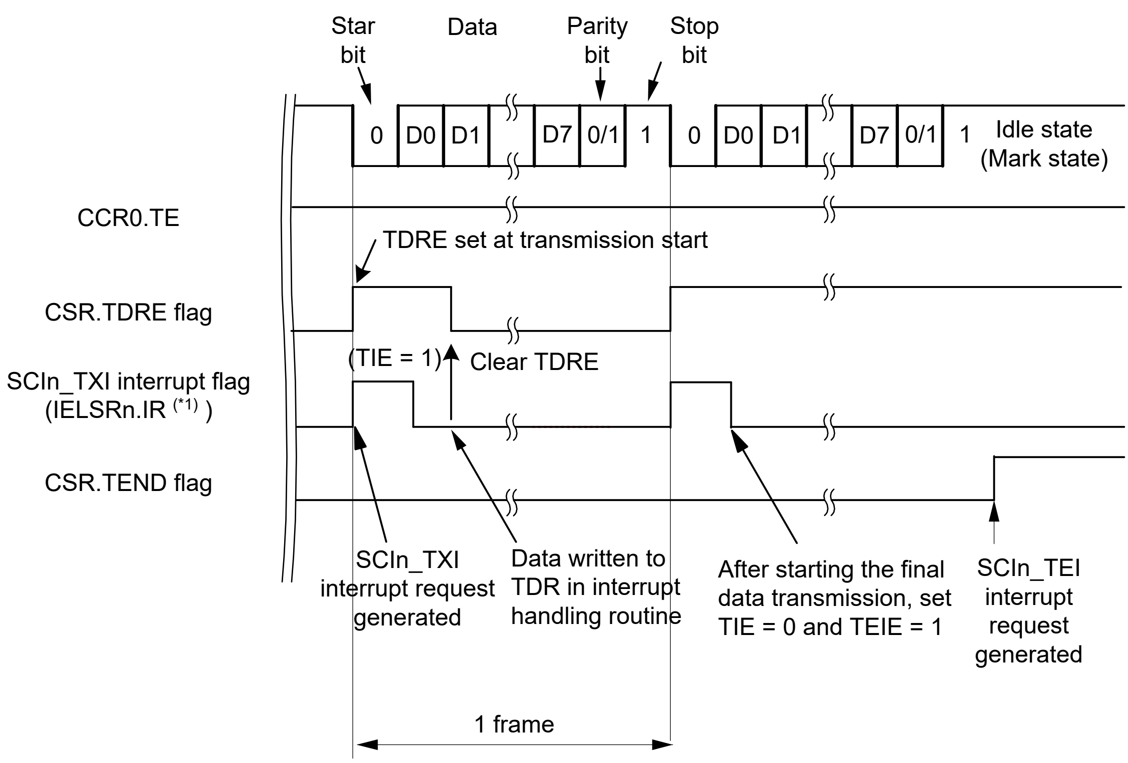
Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 38.12 Example operation for serial transmission in Asynchronous mode (1) with 8-bit data, parity bit, 1 stop bit, CTS function not used, and at the beginning of transmission



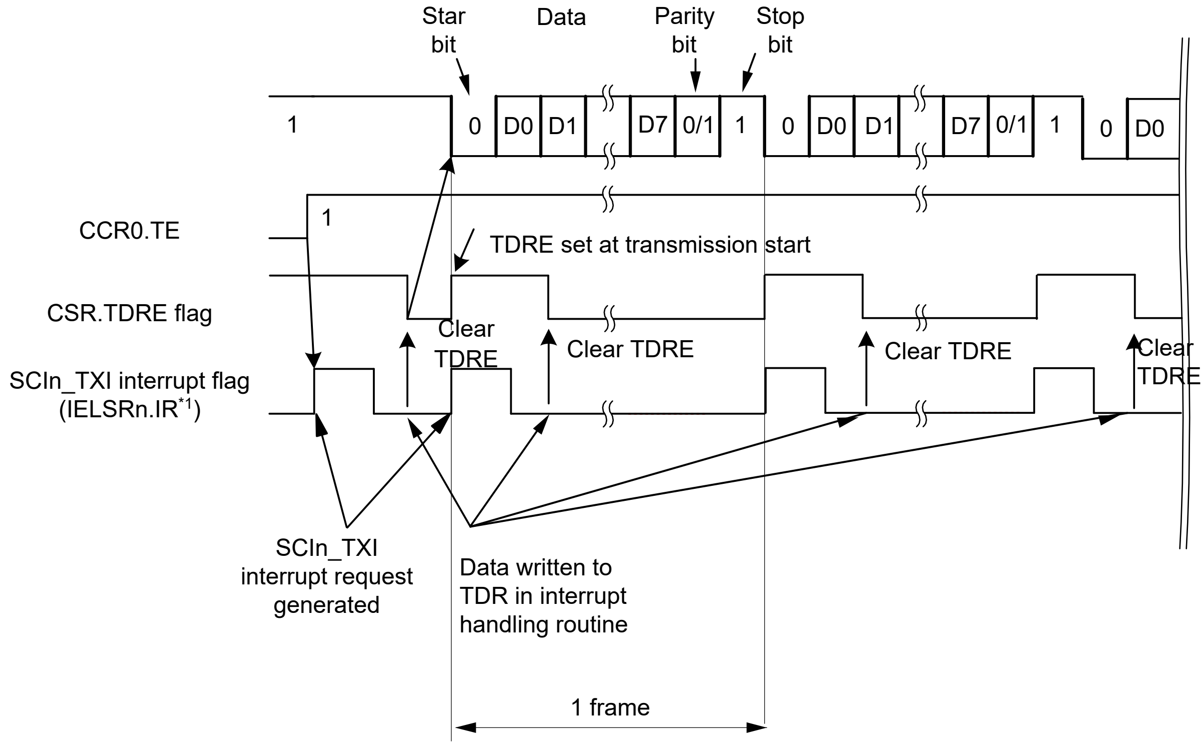
Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 38.13 Example operation for serial transmission in Asynchronous mode (2) with 8-bit data, parity bit, one stop bit, CTS function used, and at the beginning of transmission



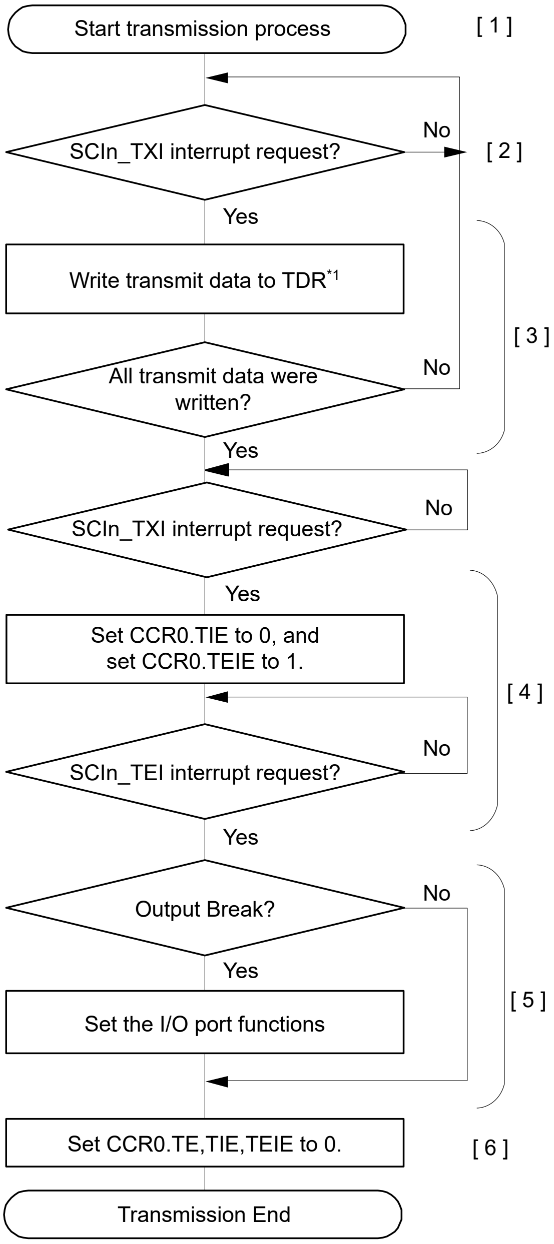
Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 38.14 Example operation for serial transmission in Asynchronous mode (3) with 8-bit data, parity bit, one stop bit, CTS function not used, and from the middle of transmission until transmission completion



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 38.15 Example of Operation for Serial Transmission in Asynchronous Mode (4)(with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, from the Middle of Transmission until Transmission Completion)



[1] Start transmission process

[2] SCIn_TXI interrupt request:
The transmit-data-empty-interrupt(SCIn_TXI) is generated at following conditions.
When set TE and TIE to 0 at same time.
When transmission-data forwards to TSR from TDR.

[3] Serial transmission continuation procedure:
To continue serial transmission write transmit data to TDR*1 once by using a SCIn_TXI.
Transmit data can also be written by activating the DMAC or DTC.

[4] If you use TEI, set TIE = 0 and TEIE = 1 after writing the last-transmit-data to TDR*1 and transmission is started.

[5] Break output at the end of serial transmission:
To output a break, it is necessary to set SPB2IO and SPB2DT, for output status setting when TE is 0.
(If you set these bits at initialization, you can skip this step.)

[6] Set CCR0.TE,TIE,TEIE to 0.

Note 1. In case of byte access with data length 9 bits, write transmit data in order of TDR[15:8] and TDR[7:0].

Figure 38.16 Example flow of serial transmission in Asynchronous mode with non-FIFO selected

(2) FIFO selected

Figure 38.17 shows an example of a data format that is written to TDR register in asynchronous mode with FIFO selected.

Data corresponding to the data length is set to TDR[8:0]. Write 0 for unused bits. Write in order from TDR[15:8] to TDR[7:0].

Data Length	Register setting		Transmit data in TDR[15:0]															
	CCR3.CHR[1:0]		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bit	1	1	–	–	–	–	–	–	MPBT	–	–	TDAT[6:0]						
8 bit	1	0	–	–	–	–	–	–	MPBT	–	TDAT[7:0]							
9 bit	0	Don't Care	–	–	–	–	–	–	MPBT	TDAT[8:0]								

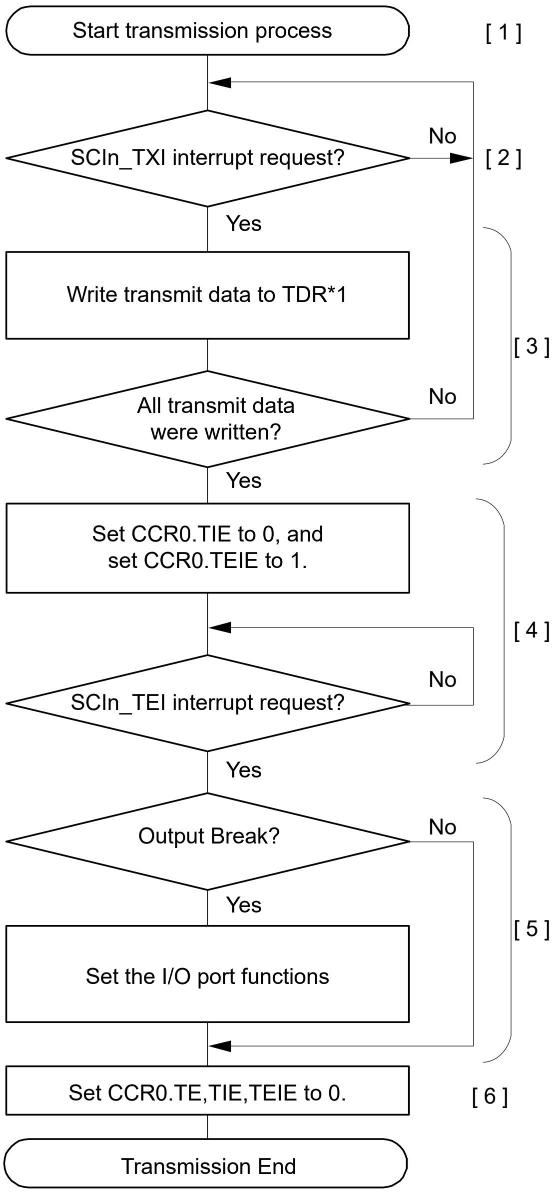
Note: —: Invalid. The write value should be 0.

Figure 38.17 Data format written to transmit-FIFO(TDR) with FIFO selected

In serial transmission, the SCI operates as described in this section.

1. The SCI transfers data from the TDR register to the TSR register when data is written to TDR in the SCIn_TXI interrupt handling routine. The amount of data that can be written to TDR is 16 minus FTSR.T[5:0] bytes. The SCIn_TXI interrupt request at the beginning of transmission is generated when the CCR0.TE and CCR0.TIE bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the CCR1.CTSE bit is set to 0 (CTS function is disabled) or a low level on the CTSn_RTSn pin causes data transfer from the TDR register to the TSR register. When the amount of transmit data written in TDR is equal to or less than the specified transmit triggering number, CSR.TDFE is set to 1. If the CCR0.TIE bit is 1, an SCIn_TXI interrupt request is generated. Continuous transmission is possible by writing the next transmit data to TDR in the SCIn_TXI interrupt handling routine before transmission of the current transmit data is complete. When SCIn_TEI interrupt requests are in use, set the CCR0.TIE bit to 0 (SCIn_TXI interrupt requests are disabled) and the CCR0.TEIE bit to 1 (an SCIn_TEI interrupt request is enabled) after the last of the data to be transmitted is written to the TDR register from the handling routine for SCIn_TXI requests.
3. Data is sent from the TXDn pin in the following order:
   - Start bit
   - Transmit data
   - Parity bit or multi-processor bit (can be omitted depending on the format)
   - Stop bit
4. On output of the stop bit, the SCI checks whether non-transmitted data remains in the TDR register.
5. When data is set to transmit-FIFO (TDR), setting the CCR1.CTSE bit to 0 (CTS function is disabled) or a low level input on the CTSn_RTSn pin causes transfer of the next transmit data from TDR to TSR and transmission of the stop bit, after which serial transmission of the next frame starts.
6. If data is not set in transmit-FIFO (TDR), the TEND flag in CSR is set to 1, the stop bit is sent, and the mark state is entered in which 1 is output. If the CCR0.TEIE bit is 1, an SCIn_TEI interrupt request is generated.

Figure 38.18 shows an example flow of serial transmission in Asynchronous mode with FIFO selected.



[1] Start transmission process

[2] SCIn_TXI interrupt request:
The transmit-data-empty-interrupt(SCIn_TXI) is generated when transmission-data forwards to TSR from the transmit-FIFO(TDR) and the number of transmit-data written in the transmit-FIFO is equal to or less than the transmit-FIFO threshold.

[3] Serial transmission continuation procedure:
To continue serial transmission, write all transmit data to the transmit-FIFO(TDR) by using a SCIn_TXI interrupt and clear TDRE flag to 0. (Transmit data can also be written to the transmit-FIFO(TDR) by activating the DMAC or DTC. In this case, the TDRE flag is cleared automatically, so please don't operate the TDRE flag.)
The number of transmission data it's possible to write in is 16 - (the number of stored transmit FIFO data).

[4] If you use SCIn_TEI, set TIE = 0 and TEIE = 1 after writing the last-transmit-data to TDR*1.

[5] Break output at the end of serial transmission:
To output a break, it is necessary to set SPB2IO and SPB2DT, for output status setting when TE is 0.
(If you set these bits at initialization, you can skip this step.)

[6] Set CCR0.TE,TIE,TEIE to 0.

Note 1. In case of byte access with data length 9 bits, write transmit data in order of TDR[15:8] and TDR[7:0].

Figure 38.18 Example flow of serial transmission in Asynchronous mode with FIFO selected

38.3.9 Serial Data Reception in Asynchronous Mode

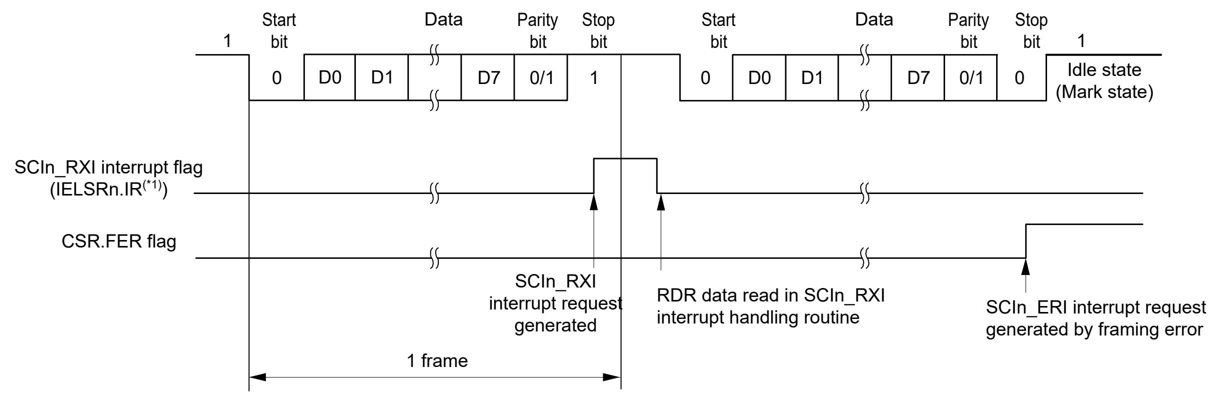
(1) Non-FIFO selected

Figure 38.19 and Figure 38.20 show an example of the operation for serial data reception in Asynchronous mode.

In serial data reception, the SCI operates as follows:

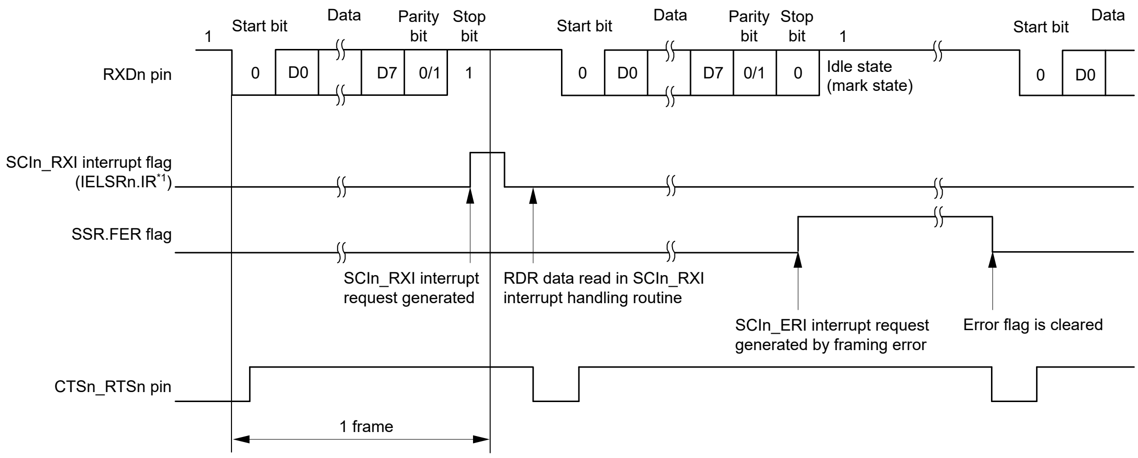
1. When the value of the CCR0.RE bit becomes 1, the output signal on the CTSn_RTSn pin goes low.
2. The SCI monitors the communications line and when it detects a start bit, the SCI performs internal synchronization, stores receive data in RSR, and checks the parity bit and stop bit.
3. If an overrun error occurs, the CSR.ORER flag is set to 1. If the CCR0.RIE bit is 1, an SCIn_ERI interrupt request is generated. Receive data is not transferred to the RDR register.
4. If a parity error is detected, the CSR.PER flag is set to 1 and receive data is transferred to the RDR register. If the SCR.RIE bit is 1, an SCIn_ERI interrupt request is generated.

5. If a framing error is detected, the CSR.FER flag is set to 1 and receive data is transferred to the RDR register. If the CCR0.RIE bit is 1, an SCIn_ERI interrupt request is generated.
6. When reception finishes successfully, receive data is transferred to the RDR register. If the CCR0.RIE bit is 1, an SCIn_RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the RDR register in the SCIn_RXI interrupt handling routine before reception of the next receive data is complete. Reading the received data that was transferred to the RDR register causes the CTSn_RTSn pin to output low.



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 38.19 Example of SCI operation for serial reception in Asynchronous mode (1) when the RTS function is not used, and with 8-bit data, parity bit, and 1 stop bit



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 38.20 Example of SCI operation for serial reception in Asynchronous mode (2) when RTS function is used, and with 8-bit data, parity bit, and 1 stop bit

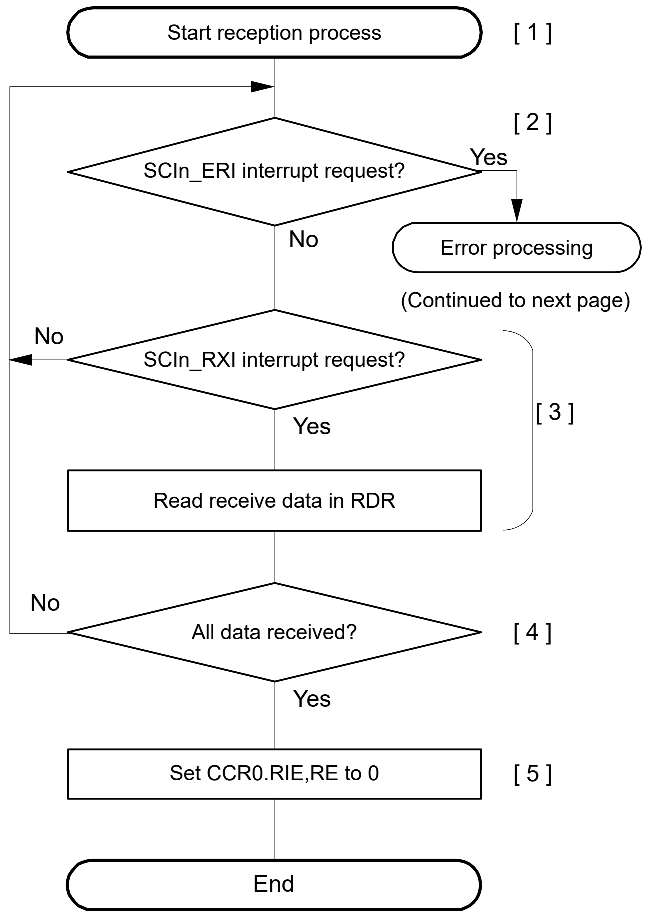
Table 38.30 lists the states of the flags in the CSR status register and receive data handling when a receive error is detected.

If a receive error is detected, an SCIn_ERI interrupt request is generated but an SCIn_RXI interrupt request is not generated. Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, FER, and PER bits to 0 before resuming reception. In addition, be sure to read the RDR register during overrun error processing. When a reception is forced to terminate by setting the CCR0.RE bit to 0 during operation, read the RDR register because received data that is not yet read might be left in the RDR.

Figure 38.21 and Figure 38.22 show example flows of serial data reception.

Table 38.30 Flags in CSR Status Register and receive data handling

Flags in the CSR Status Register				
ORER	**FER**	**PER**	**Receive data**	**Receive error type**
1	0	0	Lost	Overrun error
0	1	0	Transferred to RDR	Framing error
0	0	1	Transferred to RDR	Parity error
1	1	0	Lost	Overrun error + framing error
1	0	1	Lost	Overrun error + parity error
0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	Lost	Overrun error + framing error + parity error



[1] Start reception process

[2] Receive error processing and break detection: If a receive error occurs, an SCIn_ERI interrupt is generated. After performing the appropriate error processing, be sure to set the ORER, PER, and FER flags to 0. Reception cannot be resumed if any of these flags is set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RXDn pin.

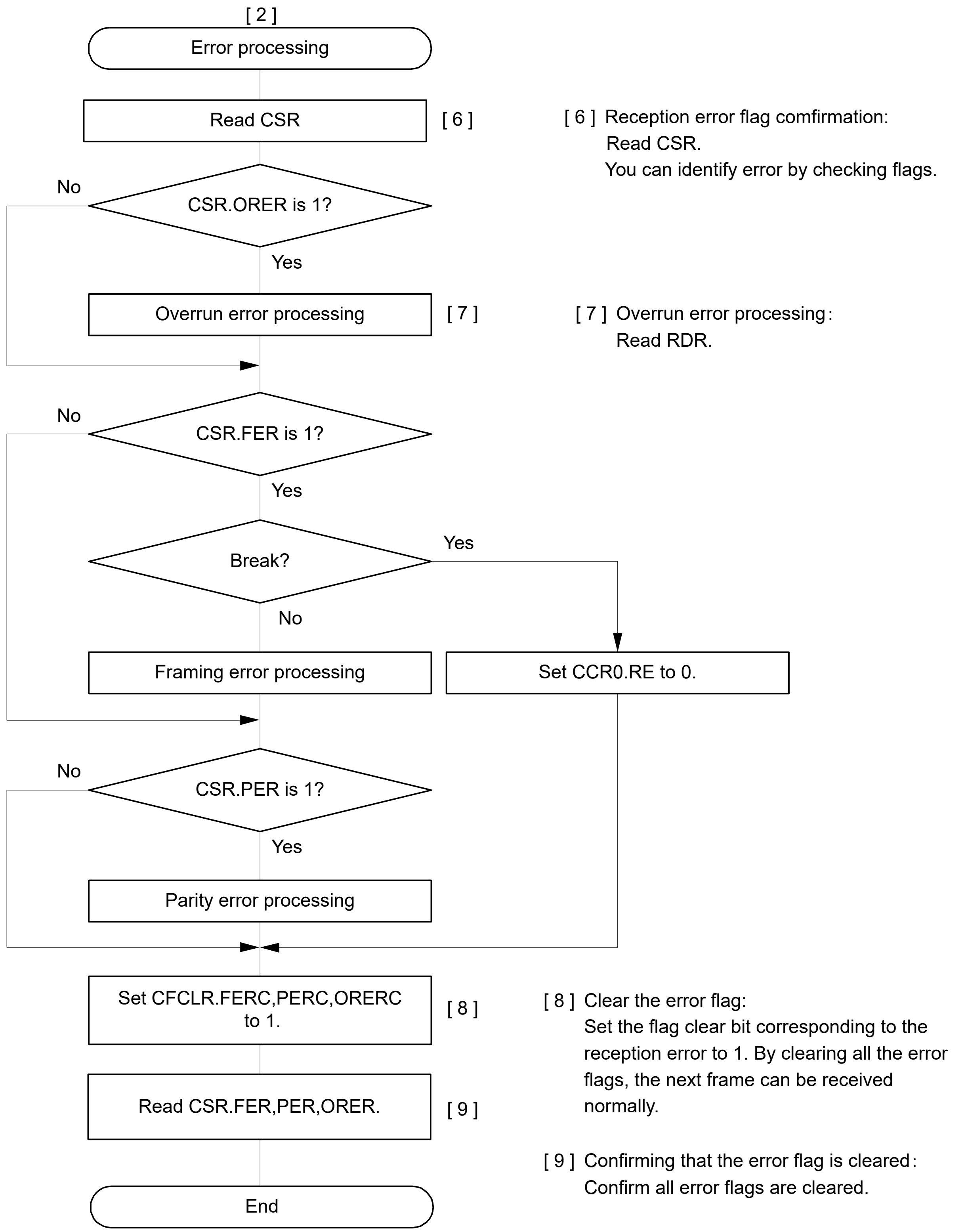
[3] Read the receive data in RDR*1 once in the SCIn_RXI interrupt handling routine.

[4] Serial reception continuation procedure: To continue serial reception, before the stop bit of the current frame is received, read data from RDR in the SCIn_RXI interrupt handling routine. The RDR data can also be read by activating the DMAC or DTC.

[5] Set CCR0.RIE,RE to 0.

Note 1. Do not set RE to 0 before reading RDR.

Figure 38.21 Example flow of serial reception in Asynchronous mode with non-FIFO selected and Address Matching Disabled (1)



Note: The RDR register becomes the RDRHL register when 9-bit data length is selected.

Figure 38.22 Example flow of serial reception in Asynchronous mode with non-FIFO selected and Address Matching Disabled (2)

(2) FIFO selected

Figure 38.23 shows an example of a data format that is written to Receive-FIFO(RDR) register in Asynchronous mode.

In Asynchronous mode, 0 is written to the MPB bit in the RDR register. Data that corresponds to the data length is written to RDR. Unused bits are written as 0. If software reads RDR, the SCI updates FER, PER, and receive data (RDAT[8:0]) in the RDR register with the next data. The flags RDF, ORER, and DR in the RDR register always reflect the associated flags in the CSR register.

Data Length	Register Setting		Receive flag in RDR[31:0], MPB, RDAT[8:0]															
	CCR3.CHR[1:0]		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7bit	1	1	-	-	-	FFER	FPER	DR	MPB	0	0	RDAT[6:0]						
8bit	1	0	-	-	-	FFER	FPER	DR	MPB	0	RDAT[7:0]							
9bit	0	Don't care	-	-	-	FFER	FPER	DR	MPB	RDAT[8:0]								
	CCR3.CHR[1:0]		b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
7bit	1	1	-	-	-	FER	PER	-	-	ORER	-	-	-	-	-	-	-	-
8bit	1	0	-	-	-	FER	PER	-	-	ORER	-	-	-	-	-	-	-	-
9bit	0	Don't care	-	-	-	FER	PER	-	-	ORER	-	-	-	-	-	-	-	-

Note: 0 is always read from the MPB flag (RDR [9] bit).
When a 7-bit data length is selected, 0 is read from the RDAT [8:7] bits.
When 8-bit data length is selected, 0 is read from the RDAT [8] bit.

Figure 38.23 Data format stored in receive-FIFO(RDR) with FIFO selected

Table 38.31 lists the states of the flags in CSR status register and receive data handling when a receive error is detected with FIFO selected. Figure 38.24 and Figure 38.25 show samples of flowcharts for serial data reception with FIFO selected.

In serial data reception, the SCI operates as follows:

1. When the value of the CCR0.RE bit becomes 1, the output signal on the CTSn_RTSn pin goes low.
2. The SCI monitors the communications line and, when it detects a start bit, the SCI performs internal synchronization, stores receive data in the RSR register.
3. If an overrun error occurs during normal communications, the CSR.ORER flag is set to 1. If the CCR0.RIE bit is 1, an SCIn_ERI interrupt request is generated. Receive data is not transferred to the RDR register.
4. If a parity error is detected, the PER flag and receive data are transferred to the RDR register. If the CCR0.RIE bit is set to 1, an SCIn_ERI interrupt request is generated.
5. If a framing error is detected, the FER flag and receive data are transferred to the RDR register. If the CCR0.RIE bit is set to 1, an SCIn_ERI interrupt request is generated.
6. After a framing error is detected and when SCI detects that the continuous receive data is zero for one frame, reception stops.
7. When the amount of data stored in the RDR register falls below the specified receive triggering number, and the next data is not received after 15 etu from the last stop bit in Asynchronous mode, the FRSR.DR flag is set to 1. When the CCR0.RIE bit is 1 and the FCR.DRES bit is 0, the SCI generates an SCIn_RXI interrupt request. When the FCR.DRES bit is 1, SCI generates an SCIn_ERI interrupt request.
8. When reception finishes successfully, receive data is transferred to the RDR register. The RDRF bit is set to 1 when the amount of receive data written to RDR is equal to or greater than the specified receive triggering number. If the CCR0.RIE bit is 1, an SCIn_RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the RDR register in the SCIn_RXI interrupt handling routine, before an overrun error occurs. If the received data that is transferred to RDR is less than the RTS trigger number, the CTSn_RTSn pin outputs low.

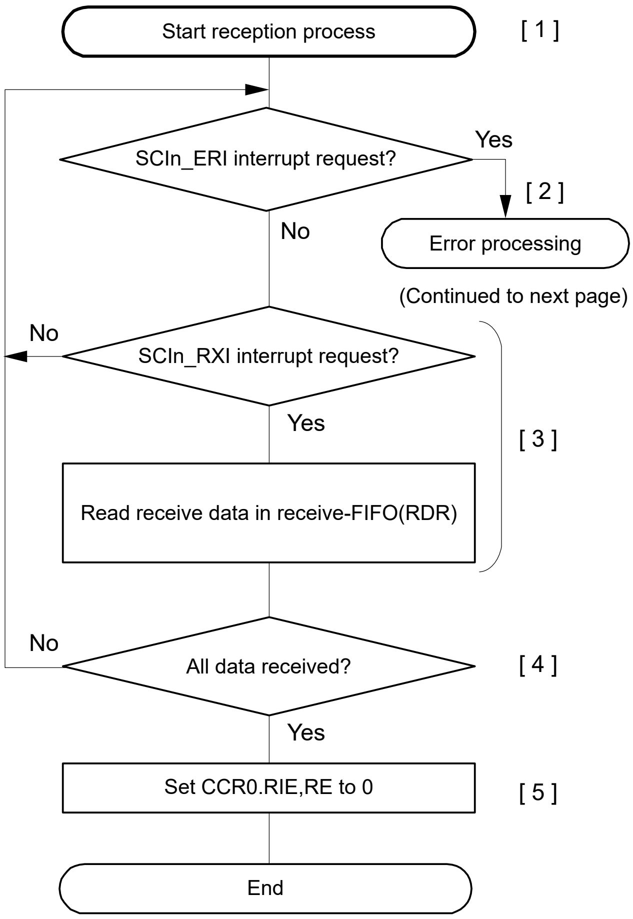
Table 38.31 Flags in the CSR Status Register and Receive Data Handling (FIFO selected) (1 of 2)

CSR value			Receive-FIFO (RDR)	Receive Error Type
ORER	FER*1	PER*1	RDAT[8:0]	
1	0	0	Lost	Overrun error
0	1	0	Transferred RDR	Framing error

Table 38.31 Flags in the CSR Status Register and Receive Data Handling (FIFO selected) (2 of 2)

CSR value			Receive-FIFO (RDR)	Receive Error Type
ORER	**FER*1**	**PER*1**	**RDAT[8:0]**	
0	0	1	Transferred RDR	Parity error
1	1	0	Lost	Overrun error + Framing error
1	0	1	Lost	Overrun error + Parity error
0	1	1	Transferred RDR	Framing error + Parity error
1	1	1	Lost	Overrun error + Framing error + Parity error

Note 1. This flag indicates whether there is an error in received data when reception is completed.



[1] Start reception process

[2] Receive error processing and break detection:
If a receive error occurs, an SCIn_ERI interrupt is generated. After performing the appropriate error processing, be sure to set the ORER flag*1 to 0. Reception cannot be resumed if ORER is set to 1. Reception can continue even if PER,FER,DR flags*1 are 1.

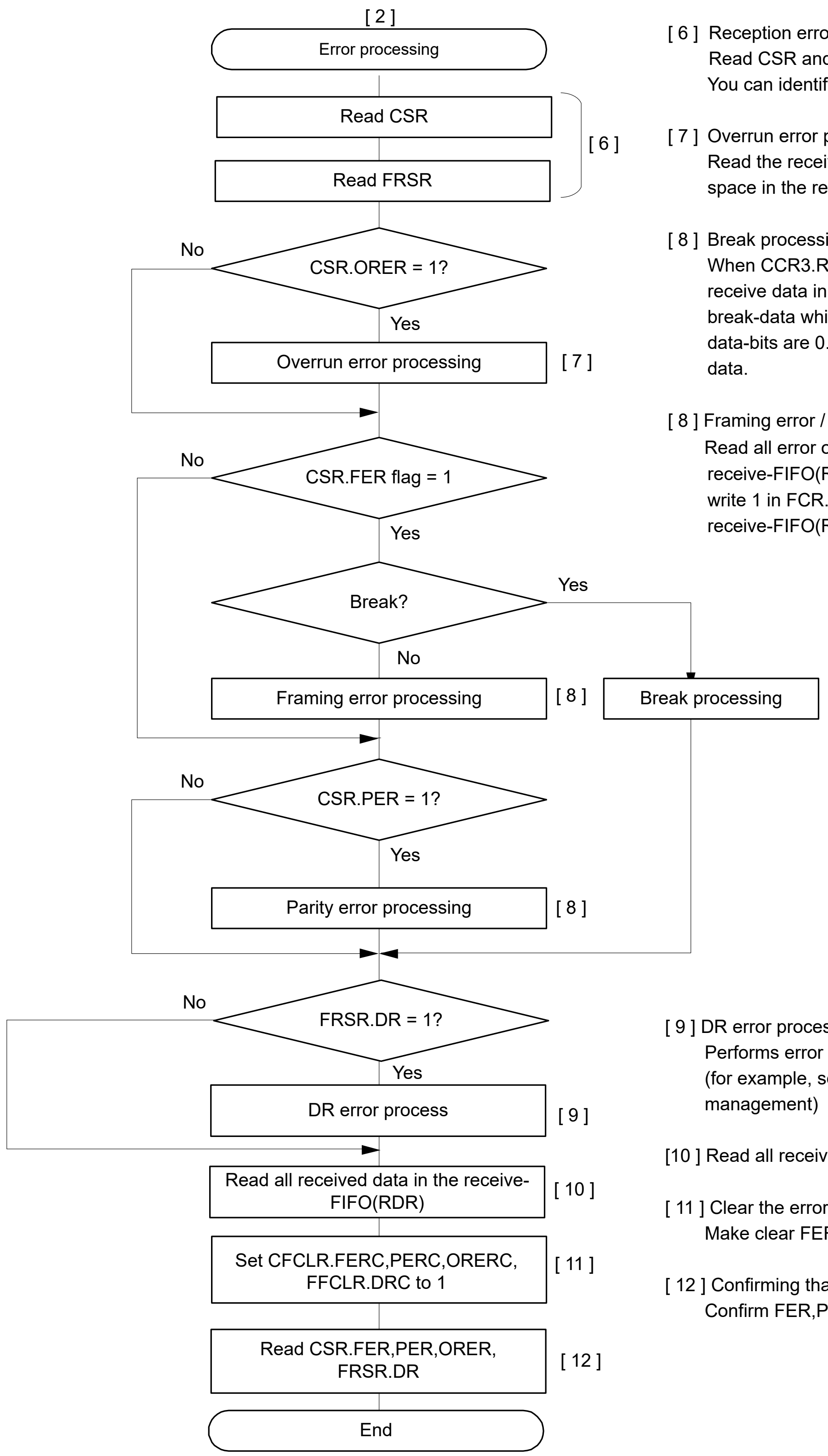
[3] Read the receive data in the receive-FIFO(RDR) in the SCIn_RXI interrupt handling routine. The reception data stored in the receive-FIFO(RDR) is read until the number of stored data will be below the FCR.RTRG value at least. You can confirm the number of the reception data in the FIFO by FRSR.R[5:0].

[4] Serial reception continuation procedure:
To continue serial reception, before an overrun error occurs, please read data from the receive-FIFO(RDR) in the SCIn_RXI interrupt handling routine and clear CSR.RDRF flag and FRSR.DR flag to 0. The receive-FIFO data can also be read by activating the DMA or DTC. The RDRF flag is cleared automatically in this case, so please don't operate the RDRF flag.

[5] Set CCR0.RIE,RE to 0.

Note 1. Can be read in the RDR.ORER, PER, FER and DR flags.

Figure 38.24 Example flow of serial reception in Asynchronous mode with FIFO selected and Address Matching Enabled (1)



[6] Reception error flag confirmation:
Read CSR and FRSR.
You can identify error by checking flags.

[7] Overrun error processing:
Read the receive-FIFO(RDR) and make space in the receive-FIFO(RDR).

[8] Break processing:
When CCR3.RXDESEL is 0, the latest receive data in the receive-FIFO(RDR) is the break-data which has framing error and all data-bits are 0. So, discard the latest receive data.

[8] Framing error / Parity error processing:
Read all error occurrence data stored in the receive-FIFO(RDR). (You can substitute to write 1 in FCR.RFRST bit and empty the receive-FIFO(RDR).)

[9] DR error process when FCR.DRES is 1.:
Performs error processing required for the system (for example, setting a counter for data number management)

[10] Read all received data in the receive-FIFO(RDR).

[11] Clear the error flag:
Make clear FER, PER, ORER, DR.

[12] Confirming that the error flag is cleared:
Confirm FER,PER,ORER,DR are cleared.

Figure 38.25 Example flow of serial reception in Asynchronous mode with FIFO selected Address Matching Disabled (2)

38.3.10 The Function of Adjust Receive Sampling Timing (Asynchronous Mode)

When there is the difference between the rising transfer time and the falling transfer time through a photo coupler, the receive sampling timing at middle of bit affects the reception margin. In this case, the receive sampling timing is able to adjust from the middle of bit to the optimum timing by using this function.

The receive sampling timing is adjusted from the middle of bit by following formula. The adjustable direction is set by CCR4.AJD. When adjusting backward (CCR4.AJD = 0), substitute AJD = +1 and substitute AJD = −1 when adjusting forward (CCR4.AJD = 1).

Adjusted sampling timing = the middle of bit + AJD × (base clock × the setting value of CCR4.AST[2:0])

The setting timing is limited by base clock cycles per 1 bit. For details, see Table 38.32.

An overview of reception operation of the communication through a photo coupler with this function is shown in Figure 38.26, Figure 38.27 and Figure 38.28, the explanation of operation with this function is shown in Figure 38.29.

Do not use this function when there is no difference between the rising transfer time and the falling transfer time, because there is a possibility of deteriorating the reception margin.

Table 38.32 The acceptable value of CCR4 register (Asynchronous mode using internal clock)

CCR2.ABCSE	CCR2.ABCS	The number of base clock cycles/1bit	The acceptable value of CCR4	
			CCR4.AJD	CCR4.AST
1	x	6	0	000b – 010b*1
			1	
0	1	8	0	000b – 011b*1
			1	
0	0	16	0	000b – 111b
			1	

Note: x: Don't care

Note 1. When the value of CCR4.AST exceeds the acceptable value, sampling is done at default timing. (Adjustment of sampling is not done.)

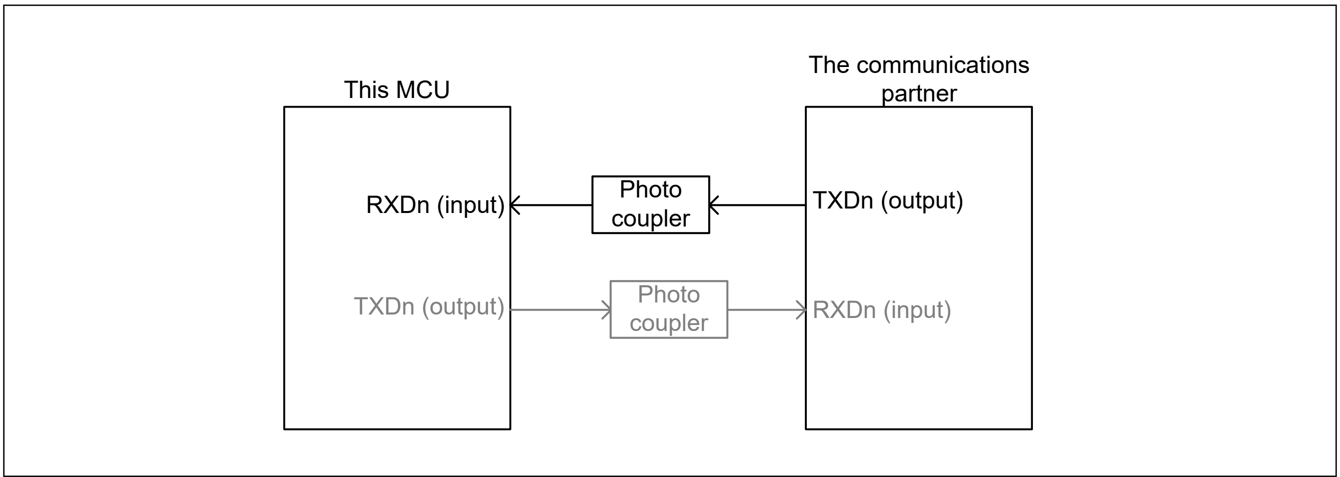
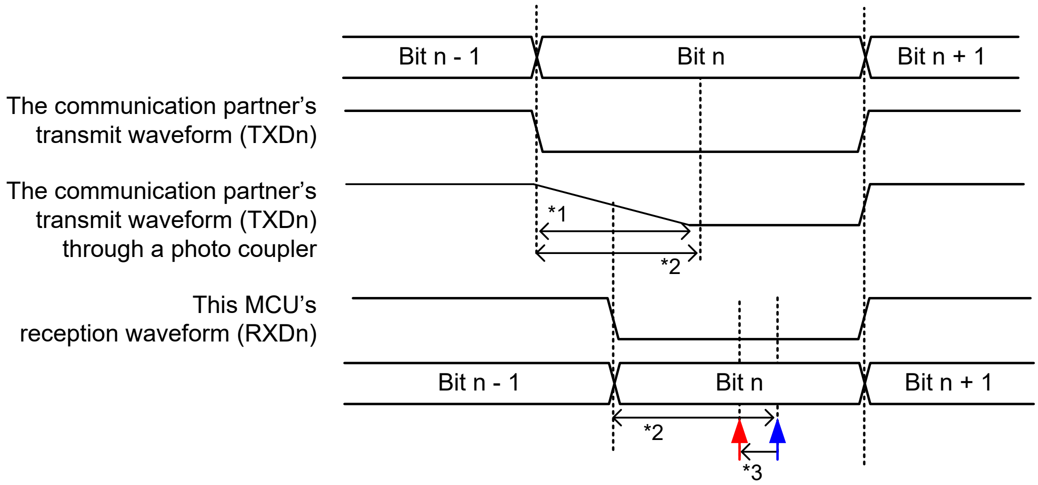


Figure 38.26 block diagram image of the reception through a photo couple

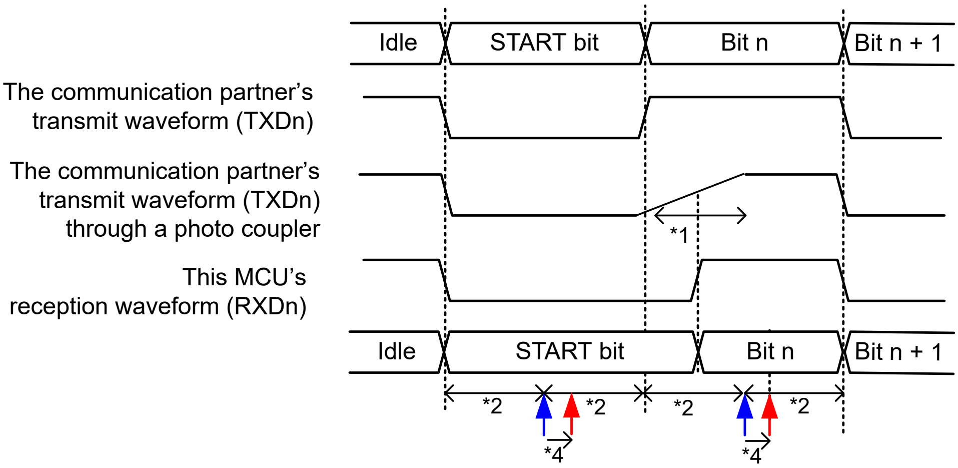
(a) In the case of the falling transfer time >> rising transfer time

The falling edge of reception waveform is made dull like following chart.
In this case, you can sampling at the middle of bit if you adjust the receive sampling timing to forward (AJD = 1).



(b) In the case of the falling transfer time << rising transfer time

The rising edge of reception waveform is made dull like following chart.
So, the reception margin of communications partner will be bad. In this case, you can improve the reception margin if you adjust the receive sampling timing to back.



The receive sampling timing when unadjusted (middle of bit)

The adjusted receive sampling timing

Note: This waveform shows the operation image of adjustment in reception sampling timing.
Note 1. The dull period by a photo coupler
Note 2. Bit center timing at set communication rate
Note 3. When CCR4.AJD is 1, the receive sampling timing is shifted to forward by the setting value of CCR4.AST[2:0].
Note 4. When CCR4.AJD is 0, the Receive sampling timing is shifted to backward by the setting value of CCR4.AST[2:0].

Figure 38.27 Overview of reception operation of the communication through a photo coupler

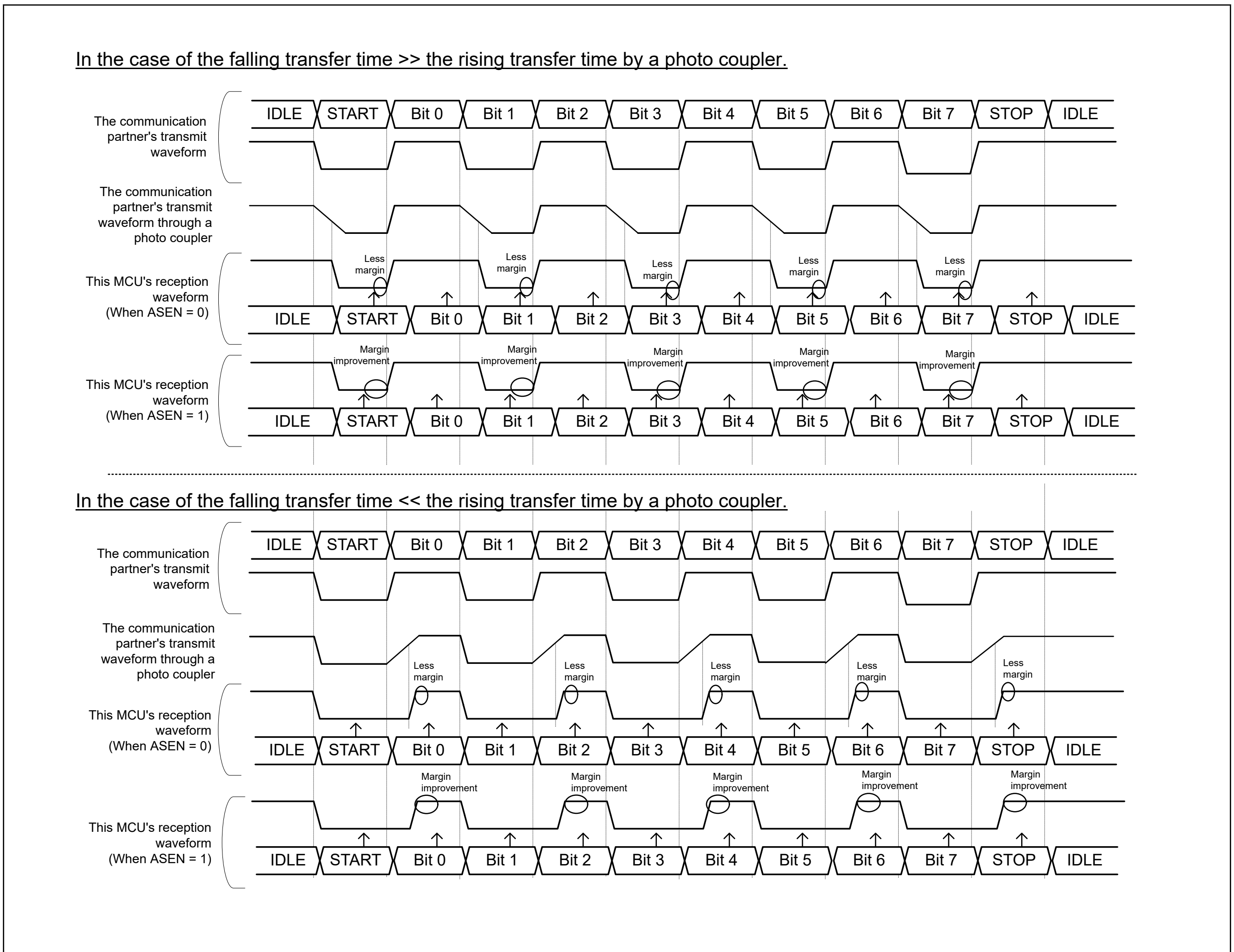


Figure 38.28 Example of improvement in reception margin by the reception sampling timing adjustment function

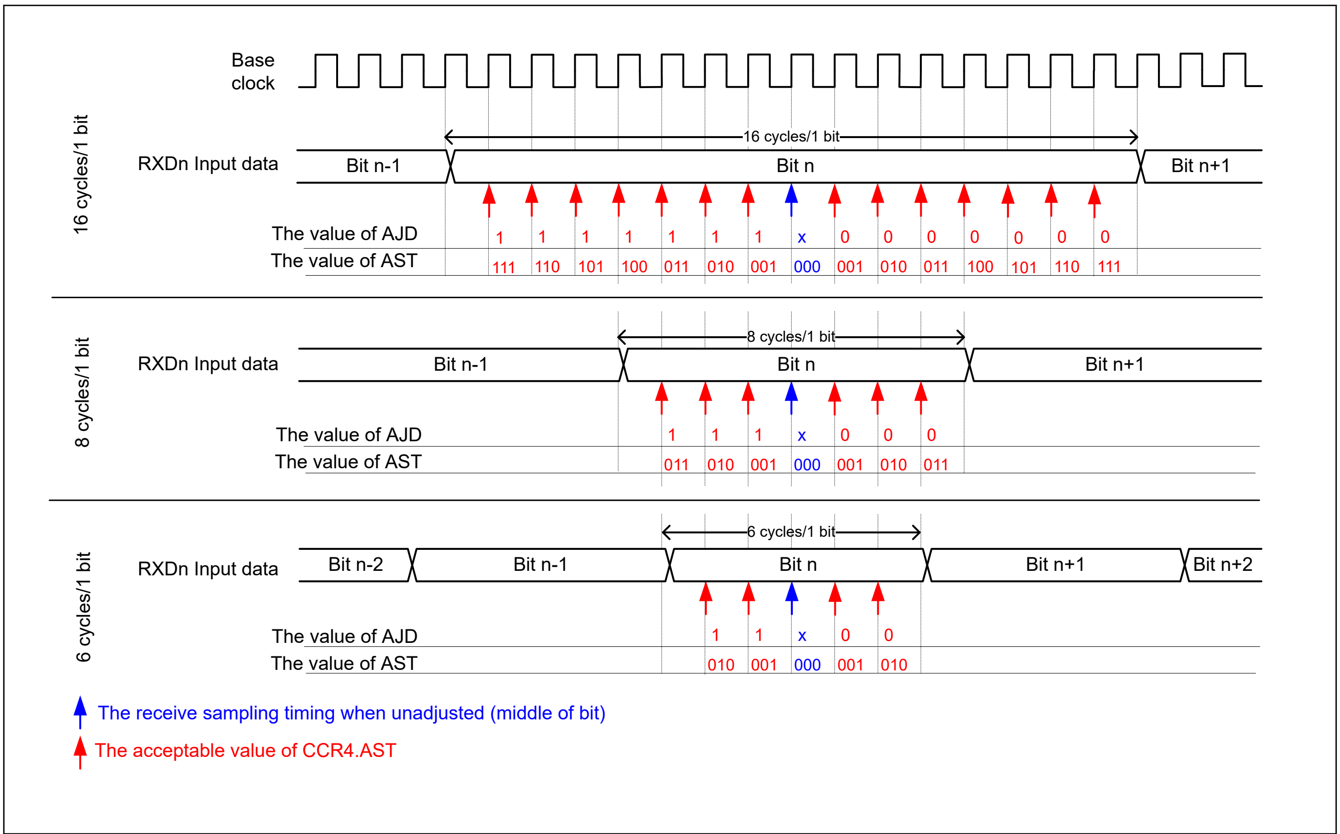


Figure 38.29 Overview of the adjustment operation for the reception sampling timing (Asynchronous mode using internal clock)

38.3.11 The Function of Adjust Transmit Timing (Asynchronous Mode)

In communication via a photo coupler or the like, when either the rising or falling transition time of the TXDn output signal is long, then a communication partner receive dulled waveform. In this case, the reception margin may be affected.

In these cases, make a communication partner to be sampling at middle of bit using the function of adjust transmit timing.

When CCR4.ATEN is 1, this function can adjust the edge timing at the timing calculated by the following formula for the edge set with CCR4.AET.

The adjustment edge timing = the base clock × CCR4.ATT[2:0]

In addition, the upper limit of the adjustment edge timing is limited by setting the base clock cycles. For details, see Table 38.33.

A transmission movement image figure of the communication through a photo coupler with this function is shown in Figure 38.30, Figure 38.31 and Figure 38.32, the overview of operation with this function is shown in Figure 38.33 and Figure 38.34.

Do not use this function when there is not the difference between the rising transfer time and the falling transfer time, there is a possibility of deteriorating the reception margin of a communication partner.

Table 38.33 The acceptable value of CCR4.AET and CCR4.ATT (Asynchronous mode using internal clock) (1 of 2)

ABCSE2	ABCSE	ABCS	The number of base clock cycles/1bit	The acceptable value	
				AET	ATT [2:0]
1	0	x	4	0	000 to 011
				1	
0	1	x	6	0	000 to 101
				1	

Table 38.33 The acceptable value of CCR4.AET and CCR4.ATT (Asynchronous mode using internal clock) (2 of 2)

<table>
<tr><th rowspan="2">ABCSE2</th><th rowspan="2">ABCSE</th><th rowspan="2">ABCS</th><th rowspan="2">The number of base clock cycles/1bit</th><th colspan="2">The acceptable value</th></tr>
<tr><th>AET</th><th>ATT [2:0]</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">1</td><td rowspan="2">8</td><td>0</td><td rowspan="2">000 to 111</td></tr>
<tr><td>1</td></tr>
<tr><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">0</td><td rowspan="2">16</td><td>0</td><td rowspan="2">000 to 111</td></tr>
<tr><td>1</td></tr>
</table>

Note: x: Don't care
Note: When the value of ACTR.AET/ATT is out of the acceptable value, this SCI module does not adjust transmit timing.

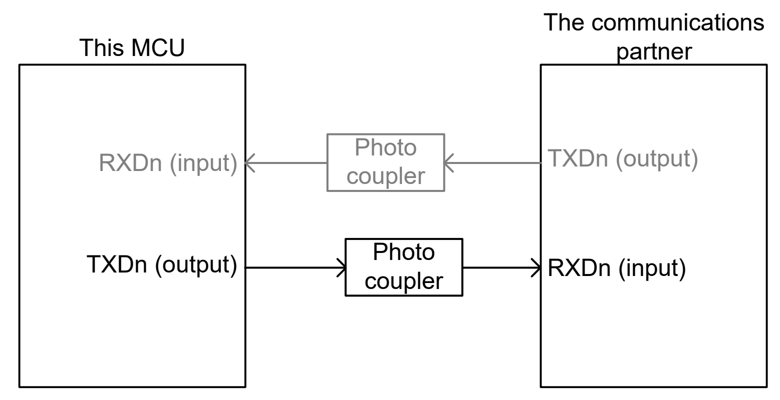
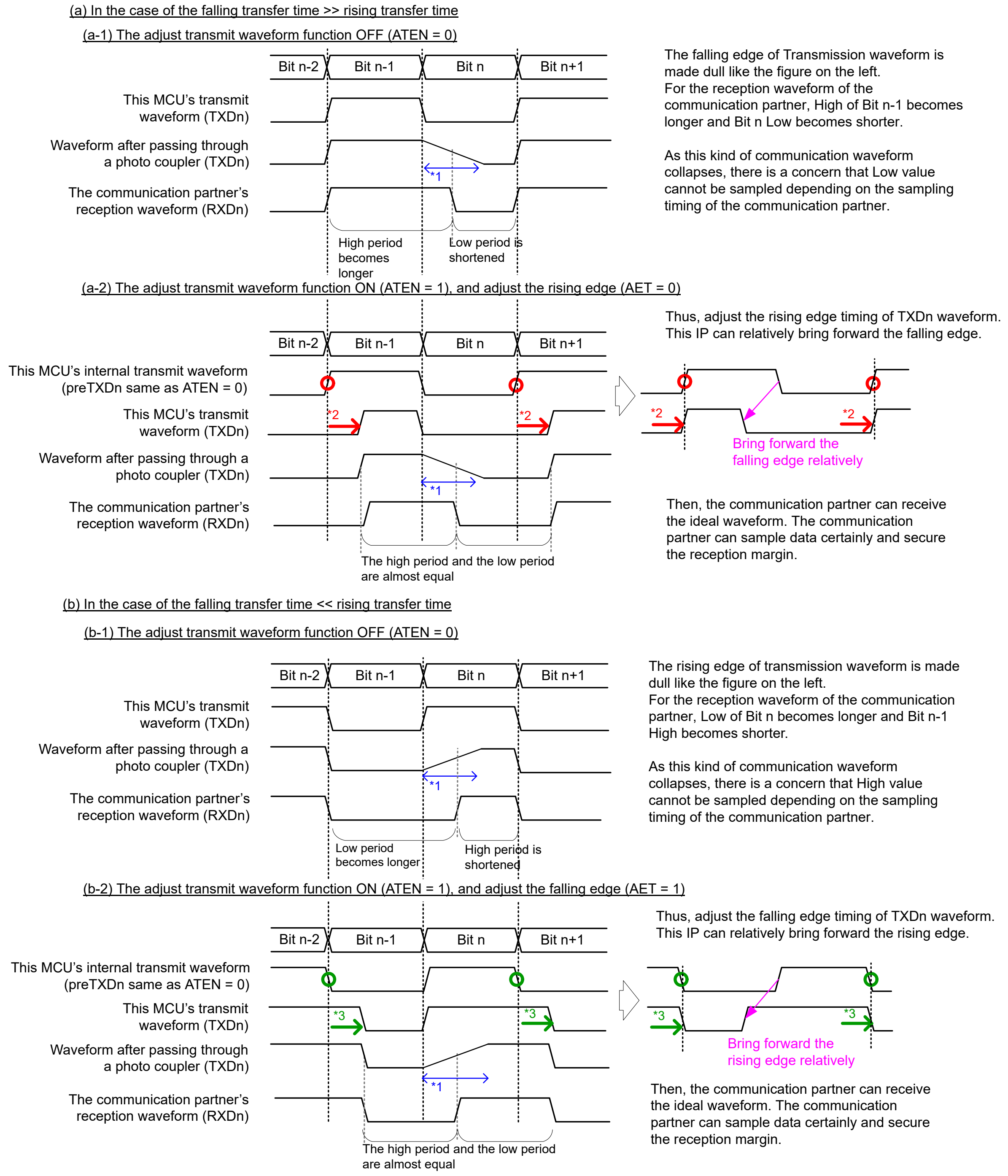


Figure 38.30 block diagram image of the transmission through a photo coupler



Note: This waveform shows the operation image of adjustment transmit waveform.

Note 1. The dull period by a photo coupler

Note 2. When CCR4.AET is 0, the rising edge of transmit waveform timing is shifted to back by the setting value of CCR4.ATT[2:0]. This MCU transmit waveform bringing forward the falling edge relatively.

Note 3. When CCR4.AET is 1, the falling edge of transmit waveform timing is shifted to back by the setting value of CCR4.ATT[2:0]. This MCU transmit waveform bringing forward the rising edge relatively.

Figure 38.31 The overview of transmission operation in the communication through a photo coupler

The explanation of transmit waveforms of the communication through a photo coupler using adjust transmit timing function

When using the transmission timing adjustment function, adjust the edge timing of the transmission waveform and correct the reception waveform of the communication partner

The following example is 8 bit long data.

(a) In the case of the falling edge transfer time >> the rising transfer time

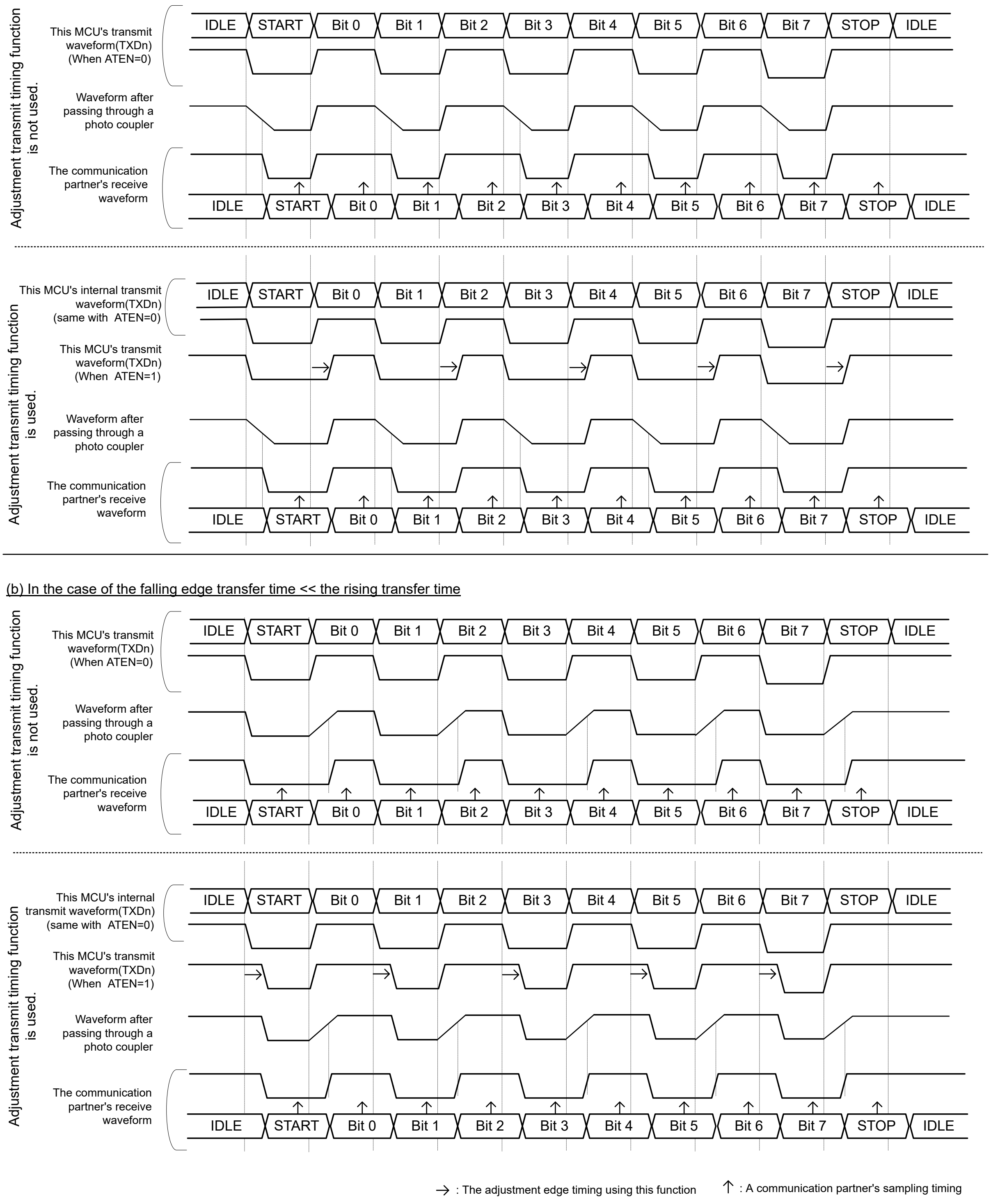
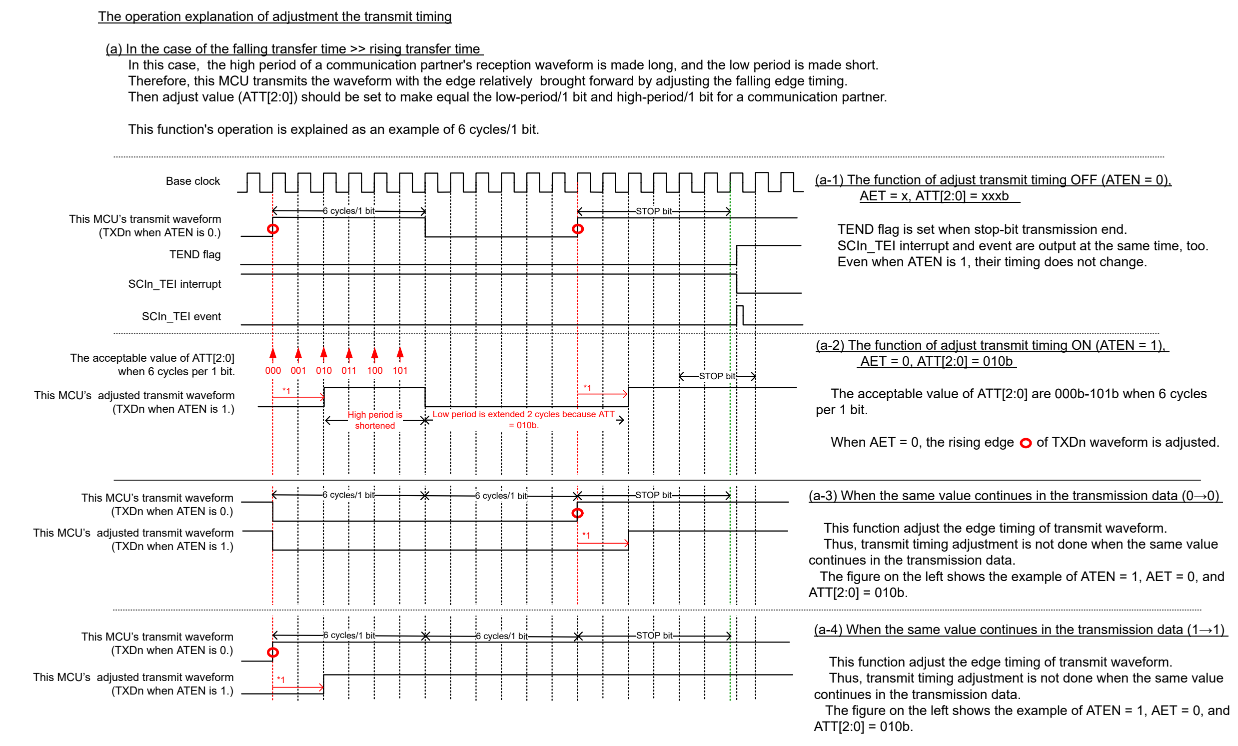
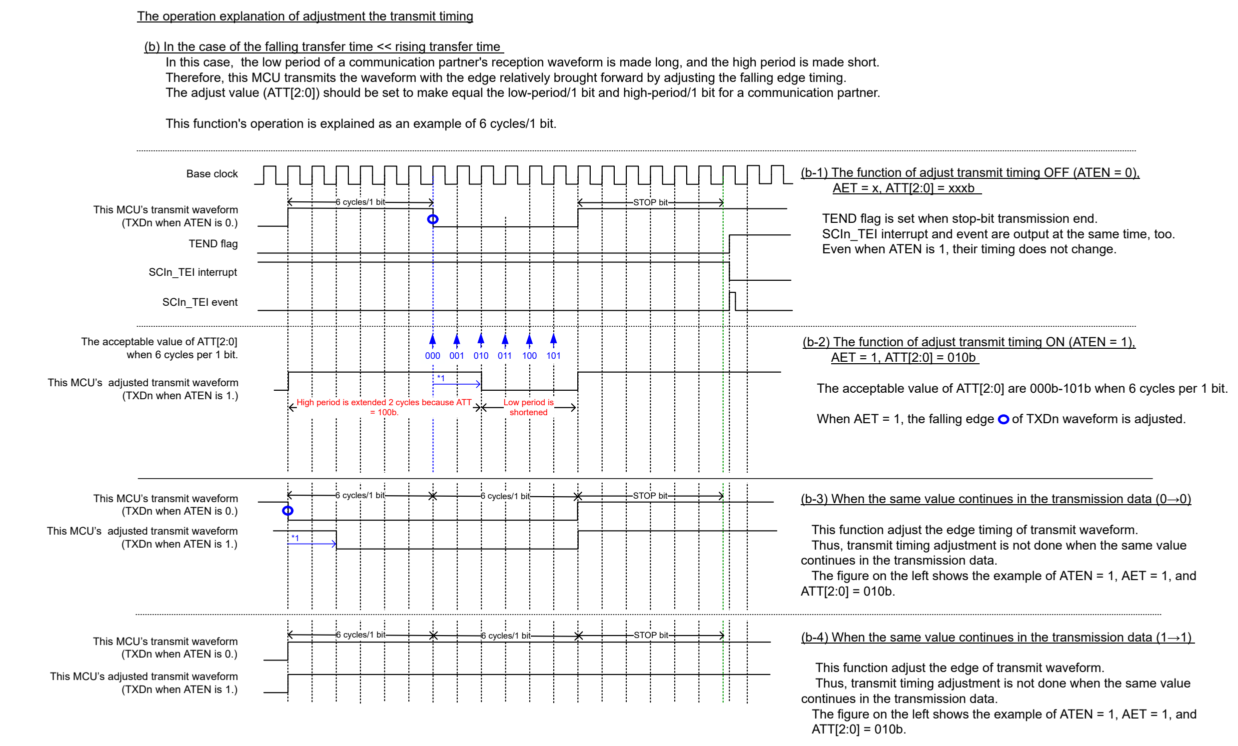


Figure 38.32 The explanation for the transmit waveform through a photo coupler



Note 1. The rising edge of transmit timing is shifted to back by the setting value of CCR4.ATT[2:0].

Figure 38.33 The adjustment operation explanation for the transmit timing when AET is 0



Note 1. The falling edge of transmit timing is shifted to back by the setting value of CCR4.ATT[2:0].

Figure 38.34 The adjustment operation explanation for the transmit timing when AET is 1

38.4 Multi-Processor Communication Function

The multi-processor communication function enables the SCI to transmit and receive data between multiple processors by sharing an asynchronous serial communication line that has an added multi-processor bit. In multi-processor communication, a unique ID code is allocated to each receiving station. Serial communication cycles consist of an ID transmission cycle to specify the receiving station and a data transmission cycle to transmit data to the specified receiving station.

The multi-processor bit is used to distinguish between the ID transmission cycle and the data transmission cycle:

- When the multi-processor bit is set to 1, the transmission cycle is the ID transmission cycle
- When the multi-processor bit is set to 0, the transmission cycle is the data transmission cycle

Figure 38.35 shows an example of communication between processors using a multi-processor format. First, a transmitting station transmits communication data in which the multi-processor bit set to 1 is added to the ID code of the receiving station. Next, the transmitting station transmits communication data in which the multi-processor bit set to 0 is added to the transmit data. After receiving communication data with the multi-processor bit set to 1, the receiving station compares the received ID with the ID of the receiving station itself. If the two match, the receiving station receives communication data that is subsequently transmitted. If the received ID does not match with the ID of the receiving station, the receiving station skips the communication data until it receives data in which the multi-processor bit is set to 1.

RTS control cannot be used at the time of multi-processor communication function use because this is a function corresponding to one-to-many communications.

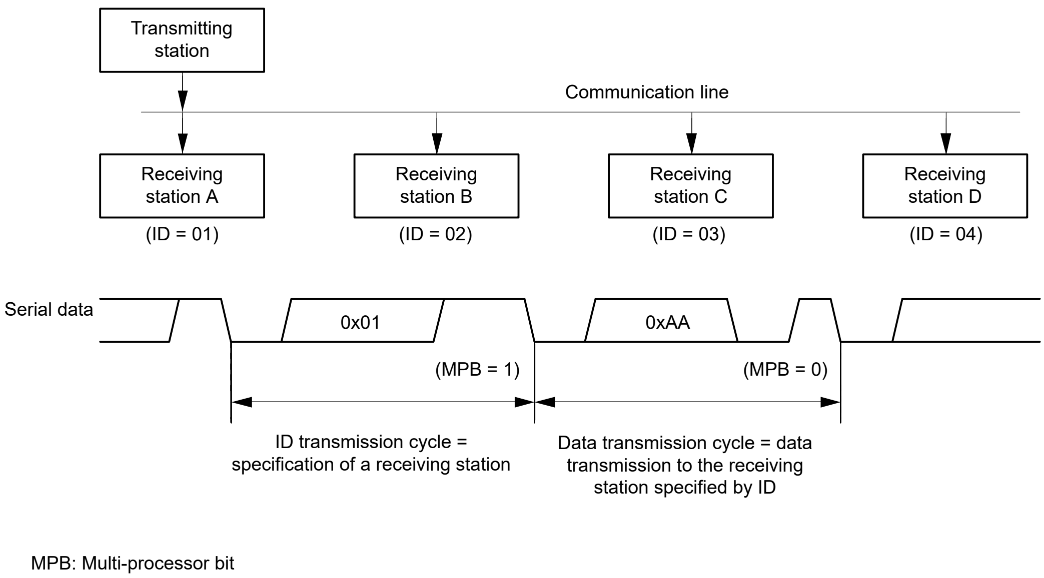


Figure 38.35 Example of communication using multi-processor format with transmission of data 0xAA to receiving station A

(1) Non-FIFO selected

To support this function, the SCI provides the CCR0.MPIE bit. When the MPIE bit is set to 1, the following operations are disabled until the reception of data in which the multi-processor bit is set to 1:

- Transfer of receive data from the RSR register to the RDR register
- Detection of a receive error
- Setting of the respective RDRF, ORER, and FER status flags in the CSR register

When the SCI receives a character in which the multi-processor bit is set to 1, the RDR.MPB bit is set to 1 and the CCR0.MPIE bit is automatically cleared, returning the SCI to non-multi-processor reception operation. If the CCR0.RIE bit is set to 1, an SCIn_RXI interrupt is generated.

When the multi-processor format is specified, the parity bit function is disabled. Apart from this, there is no difference from operation in non-multi-processor Asynchronous mode. The clock used for the multi-processor communication is the same as the clock used in non-multi-processor Asynchronous mode.

(2) FIFO selected

For data transmission, software must write data to TDR.MPBT (Multi-Processor Bit Transfer) that corresponds to transmit data in TDR.TDAT. For data reception, the multi-processor bit that is part of the receive data is written to RDR.MPB and receive data is written to RDR.RDAT.

When the MPIE bit is set to 1, the following operations are disabled until reception of data in which the multi-processor bit is set to 1:

- Transfer of receive data from the RSR register to the RDR.RDAT register
- Detection of a receive error
- Detection of DR
- Setting of the respective RDRF, ORER, and FER status flags in the CSR register

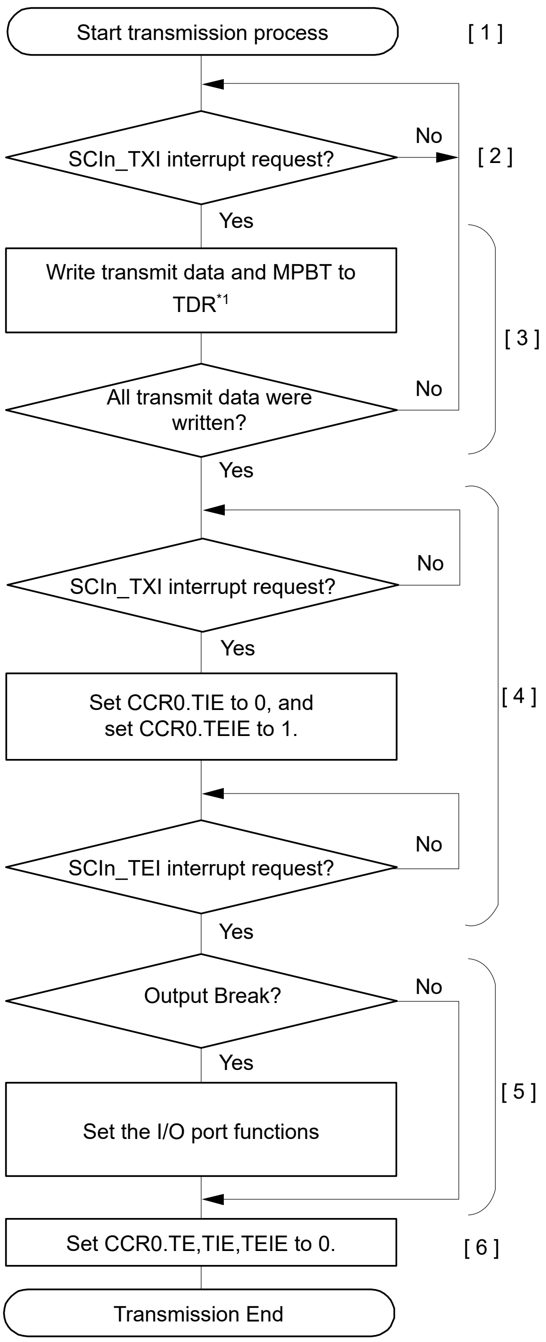
When the SCI receives a character in which the multi-processor bit is set to 1, the RDR.MPB bit is set to 1 and receive data is written to receive-FIFO(RDR.RDAT). The CCR0.MPIE bit is automatically cleared, returning the SCI to normal reception operation. If the CCR0.RIE bit is set to 1, an SCIn_RXI interrupt is generated.

When the multi-processor format is specified, the parity bit function is disabled. Apart from this, there is no difference from operation in normal Asynchronous mode with FIFO selected.

38.4.1 Multi-Processor Serial Data Transmission

(1) Non-FIFO selected

Figure 38.36 shows an example flow of multi-processor data transmission. In the ID transmission cycle, the ID must be transmitted with the TDR.MPBT bit set to 1. In the data transmission cycle, the data must be transmitted with the MPBT bit set to 0. The rest of the operations are the same as operations in Asynchronous mode.



[1] Start transmission process

[2] SCIn_TXI interrupt request:
The transmit-data-empty-interrupt(SCIn_TXI) is generated at following conditions.
When set TE and TIE to 0 at same time.
When transmission-data forwards to TSR from TDR.

[3] Serial transmission continuation procedure:
To continue serial transmission write transmit data and MPBT setting value to TDR*1 once by using a SCIn_TXI.
Transmit data and MPBT setting value can also be written by activating the DMAC or DTC.

[4] If you use SCIn_TEI, set TIE = 0 and TEIE = 1 after writing the last-transmit-data to TDR*1 and transmission is started.

[5] Break output at the end of serial transmission:
To output a break, it is necessary to set SPB2IO and SPB2DT, for output status setting when TE is 0.
(If you set these bits at initialization, you can skip this step.)

[6] Set CCR0.TE,TIE,TEIE to 0.

Note 1. In case of byte access with data length 9 bits or to write MPBT bit, write transmit data in order of TDR[15:8] and TDR[7:0].

Figure 38.36 Example flow of multi-processor serial transmission with non-FIFO selected

(2) FIFO selected

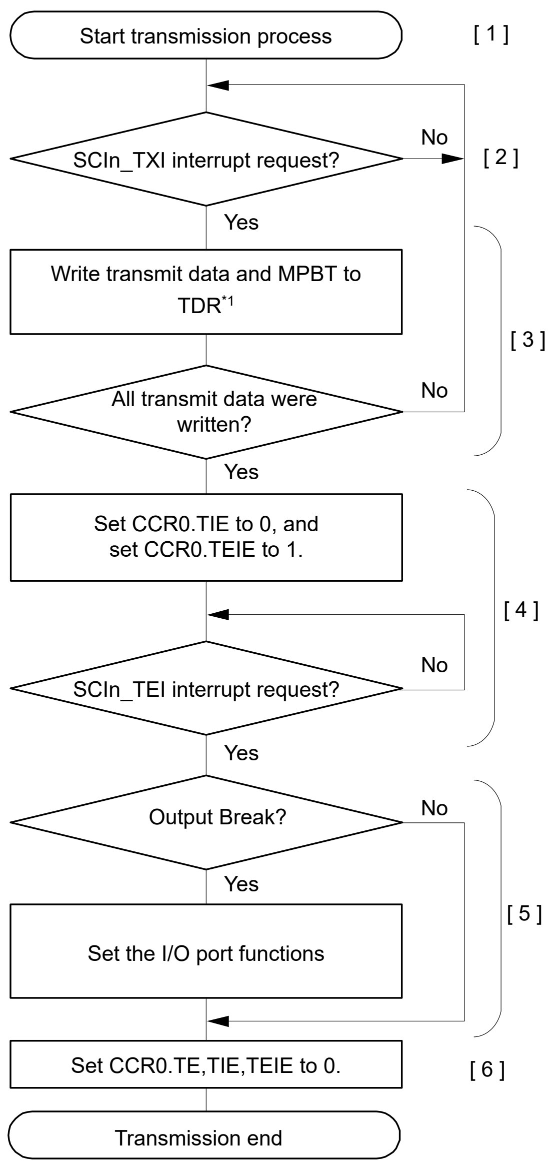
Figure 38.37 shows an example of data format that is written to transmit-FIFO (TDR) in multi-processor mode. The TDR.MPBT bit is set to 1. Data is set to transmit-FIFO (TDR) with the correct data length. Write 0 for unused bits.

Data Length	Register setting CCR3. CHR[1:0]		Transmit data in TDR[15:0] b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bit	1	1	–	–	–	–	–	–	MPBT	–	–	TDAT[6:0]						
8 bit	1	0	–	–	–	–	–	–	MPBT	–	TDAT[7:0]							
9 bit	0	Don't Care	–	–	–	–	–	–	MPBT	TDAT[8:0]								

Note: —: Invalid. The write value should be 0.

Figure 38.37 Data format written to transmit-FIFO (TDR) in multi-processor mode with FIFO selected

Figure 38.38 shows an example flow of multi-processor serial transmission with FIFO selected. In the ID transmission cycle, the ID must be transmitted with the TDR.MPBT bit set to 1. In the data transmission cycle, the data must be transmitted with the MPBT bit set to 0. The rest of the operations are the same as operations in Asynchronous mode with FIFO selected.



[1] Start transmission process
To enable SCIn_TXI interrupt at he start of transmission, set TE and TIE to 1 with one instructions at the same time. And set CCR3.FM,MP to 1.

[2] SCIn_TXI interrupt request:
The transmit-data-empty-interrupt(SCIn_TXI) is generated when transmission-data forwards to TSR from the transmit-FIFO(TDR) and the number of transmit-data written in the transmit-FIFO is equal to or less than the transmit-FIFO threshold.

[3] Serial transmission continuation procedure:
To continue serial transmission, write all transmit data and MPBT setting value to the transmit-FIFO(TDR) by using a SCIn_TXI interrupt. Transmit data and MPBT setting value can also be written to the transmit- FIFO(TDR) by activating the DMAC or DTC.

[4] If you use SCIn_TEI, set TIE = 0 and TEIE = 1 after writing the last-transmit-data to TDR*1.

[5] Break output at the end of serial transmission:
To output a break, it is necessary to set SPB2IO and SPB2DT, for output status setting when TE is 0.
(If you set these bits at initialization, you can skip this step.)

[6] Set CCR0.TE,TIE,TEIE to 0.

Note 1. In case of byte access with data length 9 bits or to write MPBT bit, write transmit data in order of TDR[15:8] and TDR[7:0].

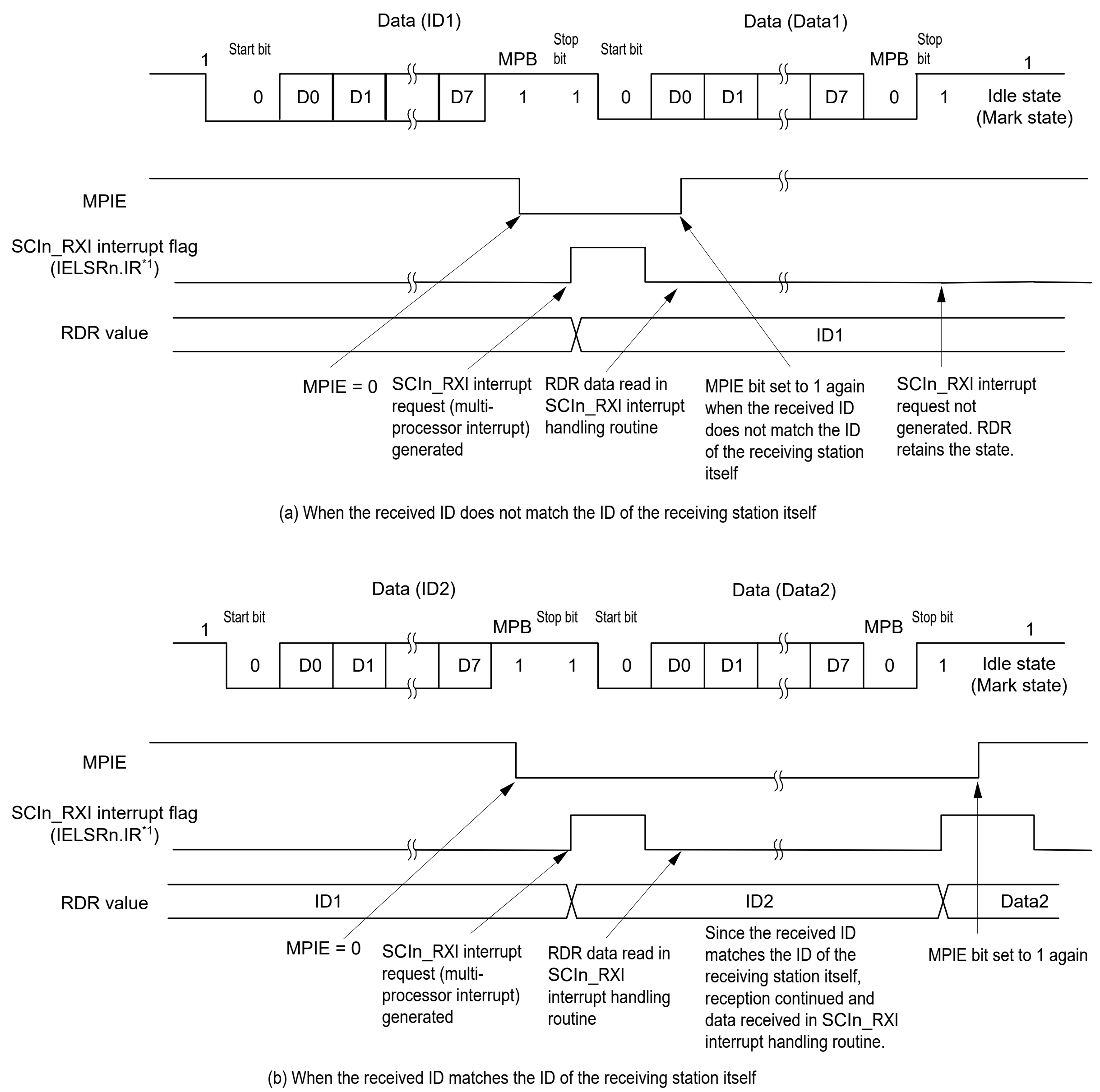
Figure 38.38 Example flow of serial transmission in multi-processor mode with FIFO selected

38.4.2 Multi-Processor Serial Data Reception

(1) Non-FIFO selected

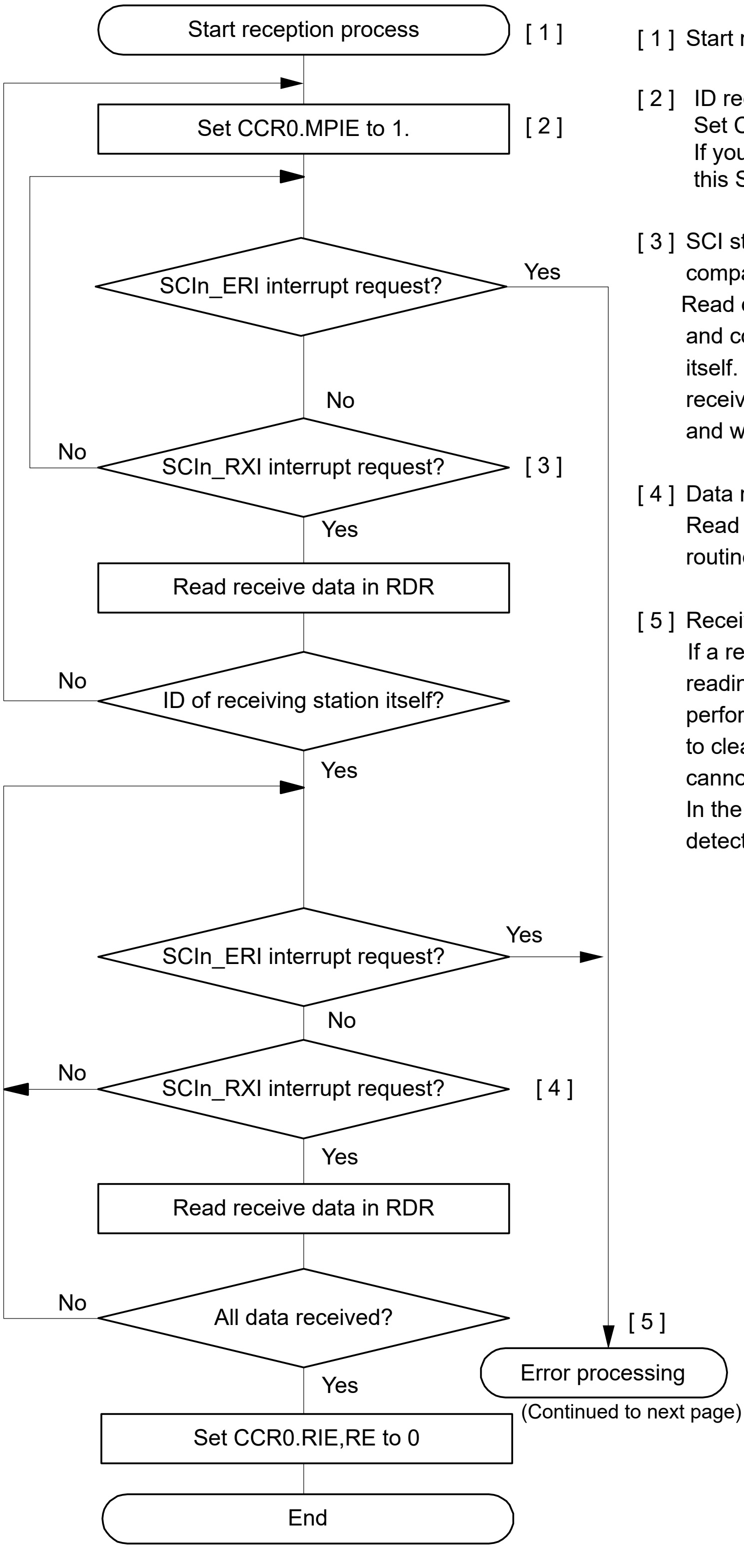
Figure 38.40 and Figure 38.41 are example flows of multi-processor serial reception. When the CCR0.MPIE bit is set to 1, reading communication data is skipped until reception of communication data in which the multi-processor bit is set to 1. When communication data in which the multi-processor bit is set to 1 is received, the received data is transferred to the RDR register, and the SCIn_RXI interrupt request is generated. The rest of the operations are the same as operations in Asynchronous mode.

Figure 38.39 shows an example operation for data reception.



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 38.39 Example of SCI reception with 8-bit data, multi-processor bit, and 1 stop bit



[1] Start reception process

[2] ID reception cycle:
Set CCR0.MPIE to 1. and wait for ID reception.
If you set this bits at initialization([1]), you can skip this Step.

[3] SCI status confirmation and reception and comparison of ID:
Read data in RDR at the first SCIn_RXI interrupt and compare it with the ID of the receiving station itself. If the ID does not match the ID of the receiving station itself, set the MPIE bit to 1 again, and wait for another SCIn_RXI interrupt request.

[4] Data reception at an SCIn_RXI interrupt:
Read data in RDR once in the SCIn_RXI interrupt routine.

[5] Receive error processing and break detection:
If a receive error occurs, an error is identified by reading ORER and FER flags in CSR. After performing the appropriate error processing be sure to clear the ORER and FER flags. Reception cannot be resumed if any of these flags is set to 1.
In the case of a framing error, a break can be detected by reading CSR.RXDMON bit.

Figure 38.40 Example flow of multi-processor serial reception with non-FIFO selected (1)

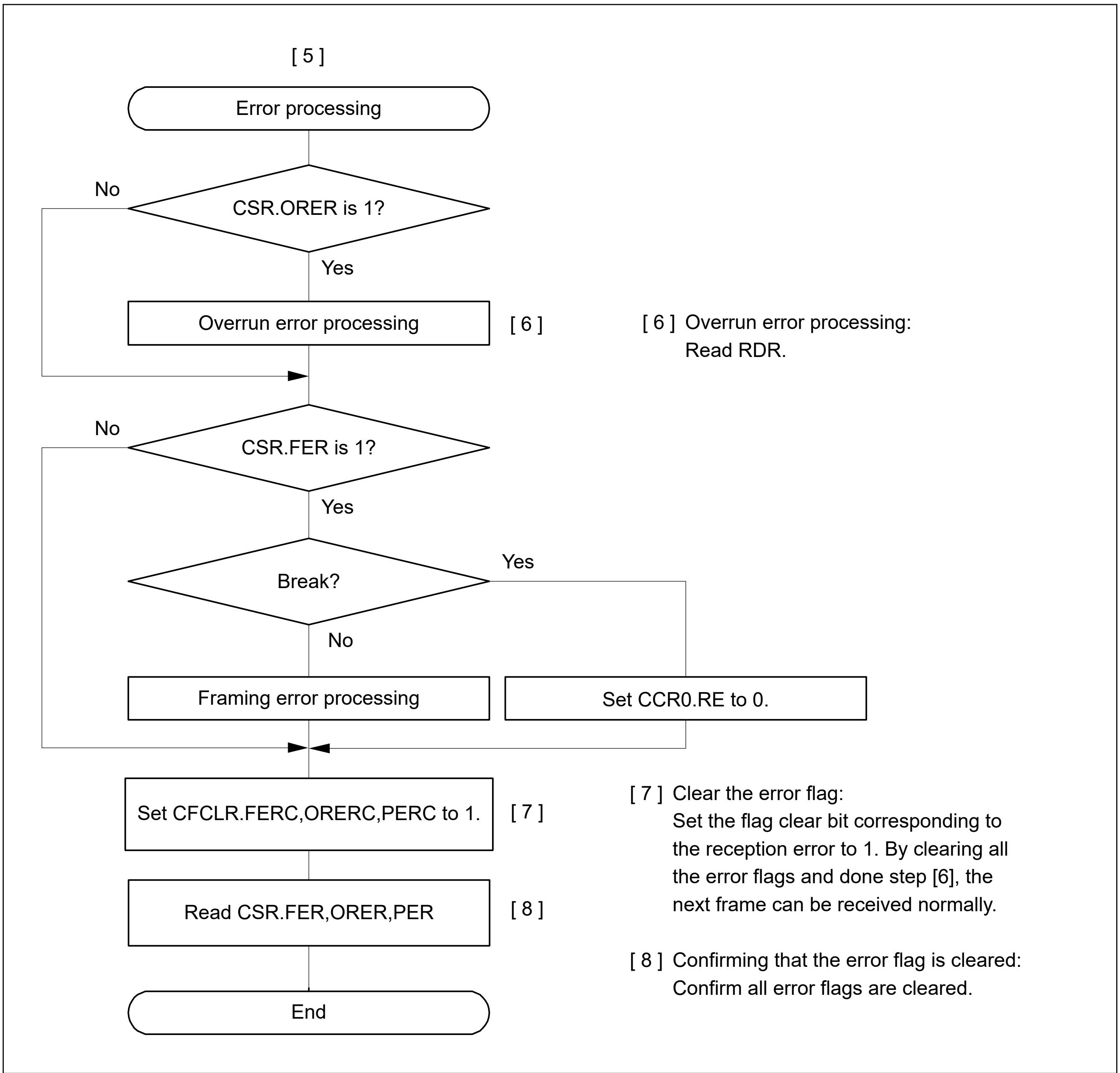


Figure 38.41 Example flow of multi-processor serial reception with non-FIFO selected (2)

(2) FIFO selected

Figure 38.42 shows an example of a data format that is written to receive-FIFO (RDR) in multi-processor mode.

In multi-processor mode, the MPB value that is a part of the receive data is written to the RDR.MPB bit. A value of 0 is written to the RDR.FPER and PER flags. Data is written to receive-FIFO (RDR) with the correct data length. Unused bits are written with 0. When software reads the receive-FIFO (RDR) register, the SCI updates RDR.FFER, FPER, and MPB flags, and receive data (RDAT[8:0]) in receive-FIFO (RDR) with the next data. The FER, PER, and ORER flags in the receive-FIFO (RDR) register always reflect the associated flags in the CSR and FRSR registers.

Data Length	Register Setting CCR3.CHR[1:0]		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7bit	1	1	0	0	0	FFER	FPER	DR	MPB	0	0	RDAT[6:0]						
8bit	1	0	0	0	0	FFER	FPER	DR	MPB	0	RDAT[7:0]							
9bit	0	Don't Care	0	0	0	FFER	FPER	DR	MPB	RDAT[8:0]								

Data Length	CCR3.CHR[1:0]		b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
7bit	1	1	0	0	0	FER	PER	0	0	ORER	0	0	0	0	0	0	0	0
8bit	1	0	0	0	0	FER	PER	0	0	ORER	0	0	0	0	0	0	0	0
9bit	0	Don't Care	0	0	0	FER	PER	0	0	ORER	0	0	0	0	0	0	0	0

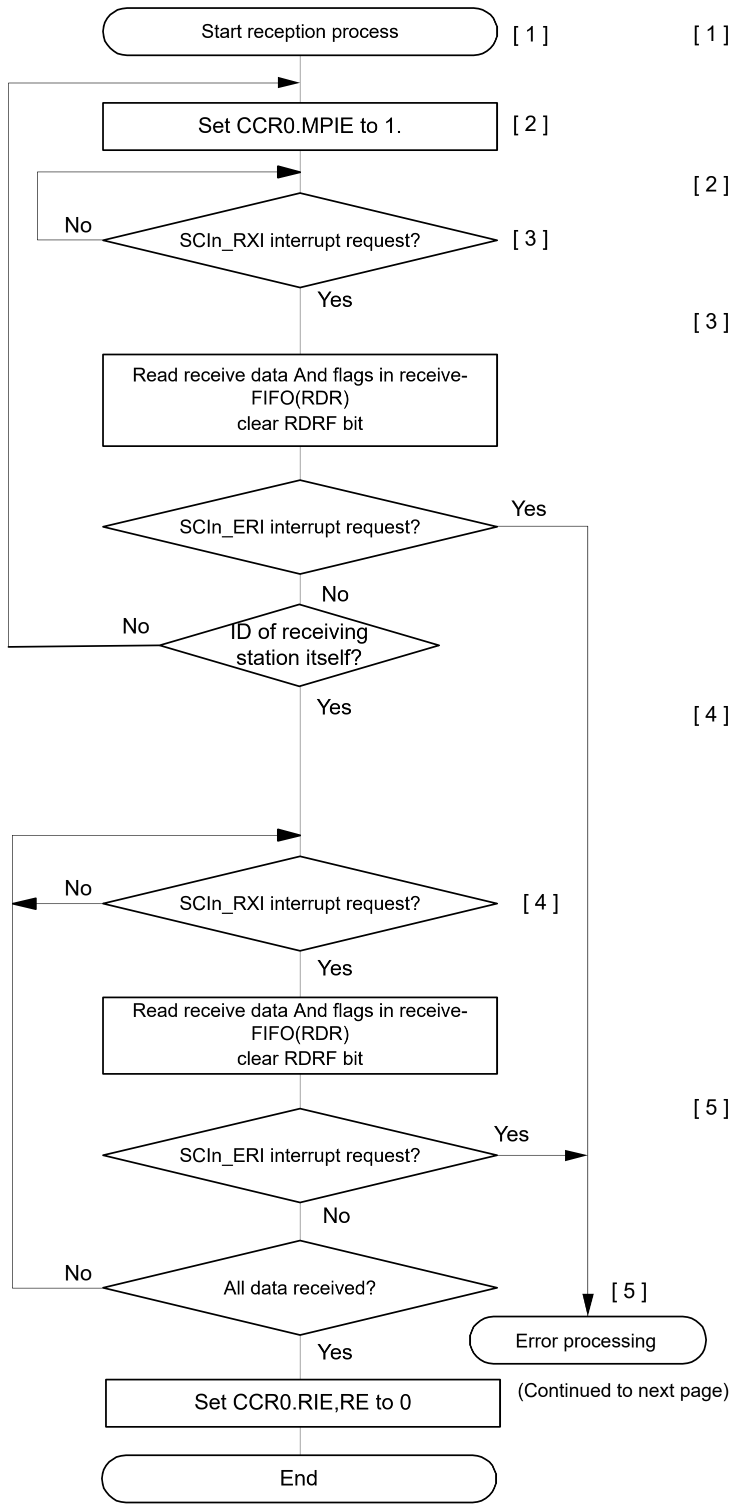
Receive data in RDR[31:0]

Note: When data length is 7bit, it can read always 0 in RDAT[8:7].
When data length is 8bit, it can read always 0 in RDAT[8].

Figure 38.42 Data format stored in receive-FIFO (RDR) in multi-processor mode with FIFO selected

Figure 38.43 shows an example flow of multi-processor data reception with FIFO selected. When the CCR0.MPIE is set to 1, reading communication data is skipped until reception of communication data in which the multi-processor bit is set to 1. When communication data in which the multi-processor bit is set to 1 is received, the received data, MPB and associated errors are transferred to the receive-FIFO (RDR) register. The CCR0.MPIE bit is automatically cleared and normal reception continues.

If a framing error occurs and the CSR.FER flag is set to 1, the SCI continues data reception. The rest of the operations are the same as operations in Asynchronous mode with FIFO selected.



(Continued to next page)

[1] Start reception process
To generate an SCIn_RXI interrupt request with MP detection, please set FCR.RTRG = 1 and FCR.DRES = 0.

[2] ID reception cycle:
Set CCR0.MPIE to 1. and wait for ID reception. If you set this bits at initialization([1]), you can skip this Step.

[3] SCI status confirmation and reception and comparison of ID:
Skip the communication data until the MPB bit receives data 1. When the MPB bit receives data 1, an SCIn_XRI interrupt request is generated. When an SCIn_RXI interrupt request is received, read the data in the receiving FIFO (RDR register) and compare it to the ID of my station. If it is not my station's ID, set the MPIE bit to 1 again and wait for the SCIn_RXI interrupt request. For my station's ID, set the received FIFO threshold to the necessary value.

[4] Data reception at an SCIn_RXI interrupt:
When the number of received data stored in the received FIFO (RDR register) exceeds the received FIFO threshold, the CSR.RDRF bit becomes 1 and an SCIn_RXI interrupt request is generated. In addition, if the number of received data stored in the received FIFO (RDR register) is 1 or more and less than the received FIFO threshold, the FRSR.DR bit becomes 1 after 15 etus, and an SCIn_RXI interrupt request is generated. The SCIn_RXI interrupt routine reads the data from the received FIFO (RDR register).

[5] Receive error processing and break detection:
If a receive error occurs, an error is identified by reading ORER and FER flags in CSR. After performing the appropriate error processing be sure to clear the ORER and FER flags. In the case of a framing error, a break can be detected by reading the value of the RXDn pin.

Figure 38.43 Example flow of serial reception in multi-processor mode with FIFO selected

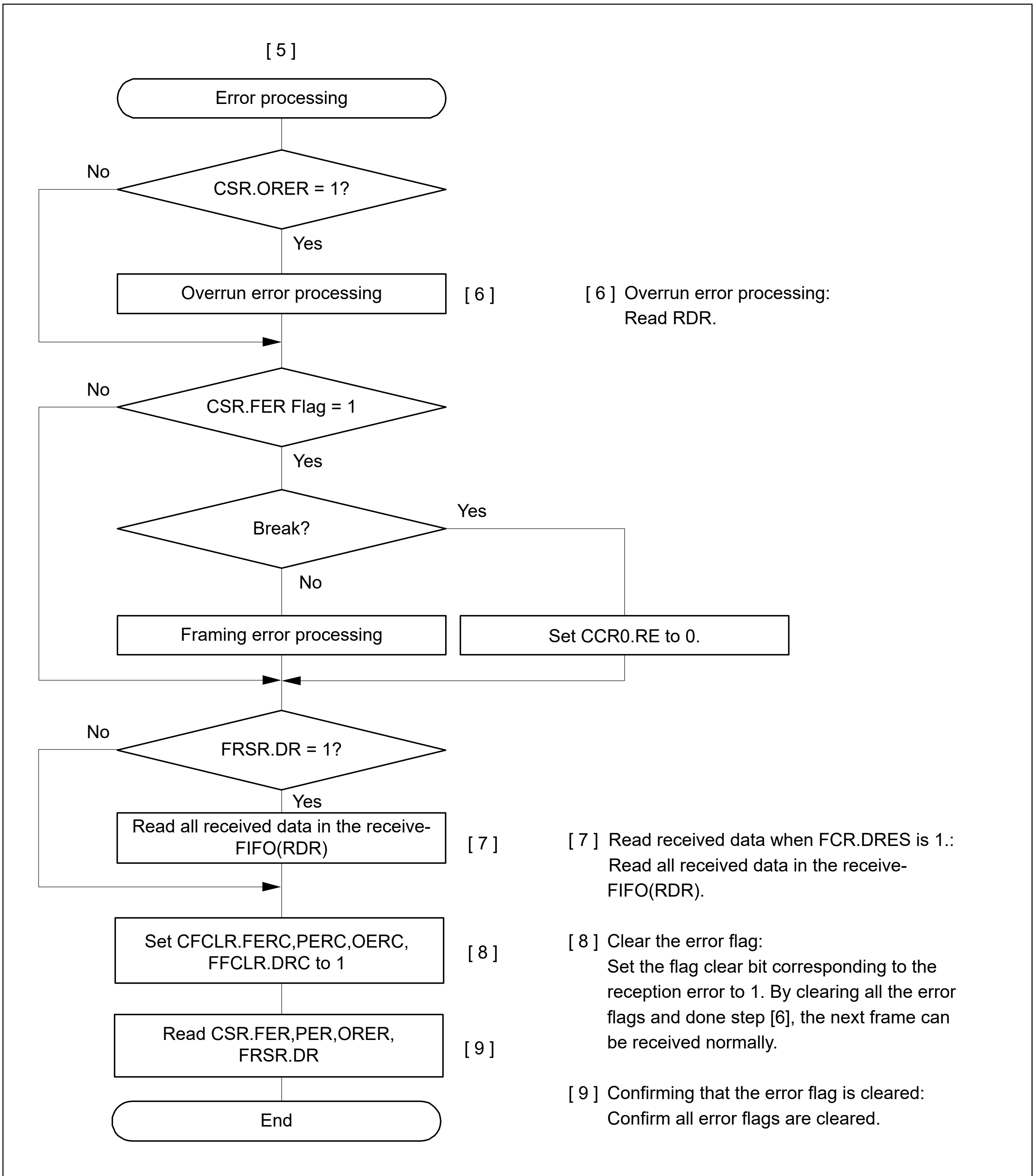


Figure 38.44 Example Flowchart of Serial Reception in Multi-Processor Mode (2) (FIFO selected)

38.5 Operation in Manchester Mode

In Manchester mode, the transmit or receive serial data is coded in Manchester encoding.

Figure 38.45 shows the conceptual image of Manchester encoding.

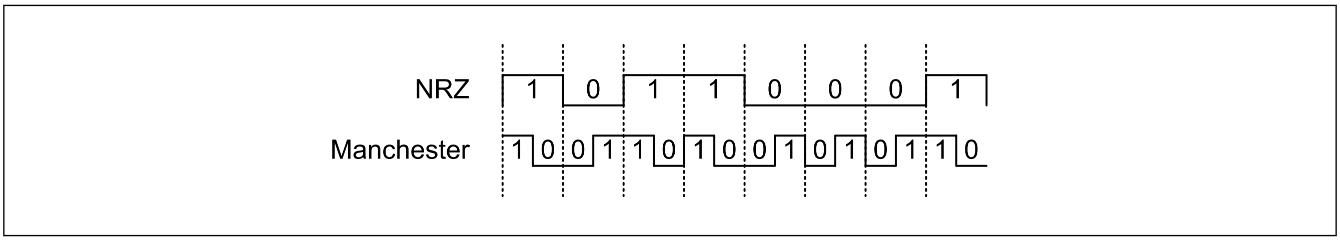


Figure 38.45 Example of Manchester Encoding

In Manchester mode, a preface and a start bit area are added to the transmit data in the register to configure a transmit frame. For transmission, data is encoded in Manchester encoding. When data is received, frames having the same format as transmitted frames are detected and Manchester decoding is performed.

For details on the frame format, see section 38.5.1. Frame Format.

38.5.1 Frame Format

Figure 38.46 shows the frame format in Manchester mode.

In the upper half of the figure, relevant setting registers are shown.

The preface area and the data area are encoded in Manchester encoding.

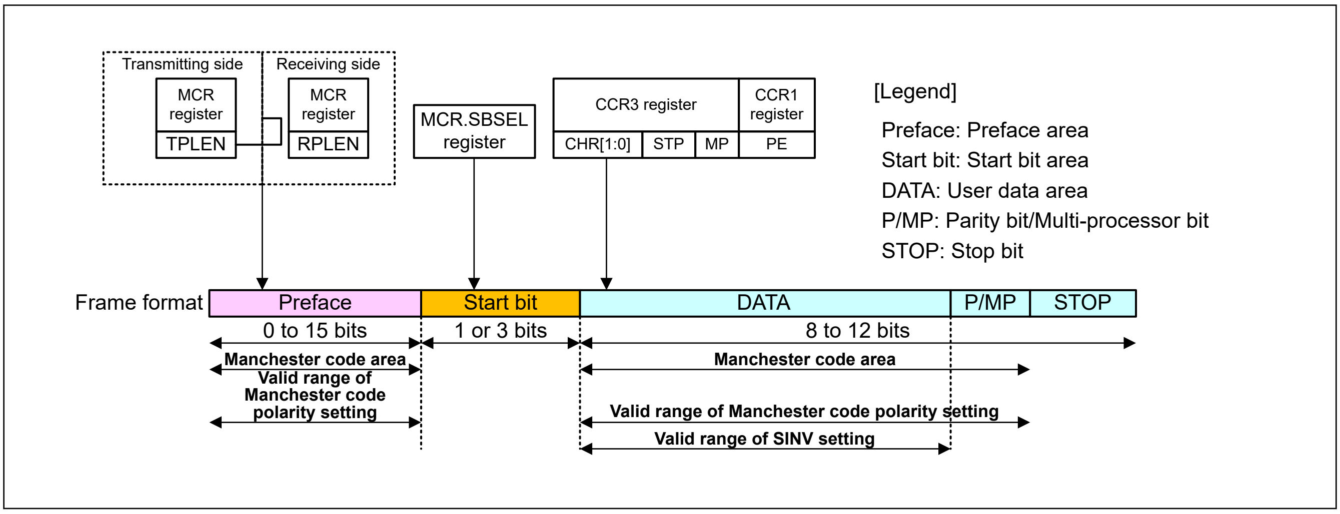


Figure 38.46 Frame Format in Manchester Mode

(1) Preface area

This is a fixed pattern area located at the beginning of each frame.

Different registers are used to set the preface area for transmission and reception. The preface length is determined by setting MCR.TPLEN[3:0] for transmission. It is determined by setting MCR.RPLEN[3:0] for reception.

If it is set to 0, the transmit preface is disabled and is not added.

If it is set to 1d to 15d, a preface whose length is determined by this setting is added.

(For example, if it is set to 1d, a 1-bit preface is added. If it is set to 15d, a 15-bit preface is added.)

The preface pattern is set with MCR.TPPAT[1:0] for transmission and MCR.RPPAT[1:0] for reception, and is selected from four types of patterns.

Figure 38.47 shows how the preface pattern is set. The preface area and the start bit area are added for each communication frame.

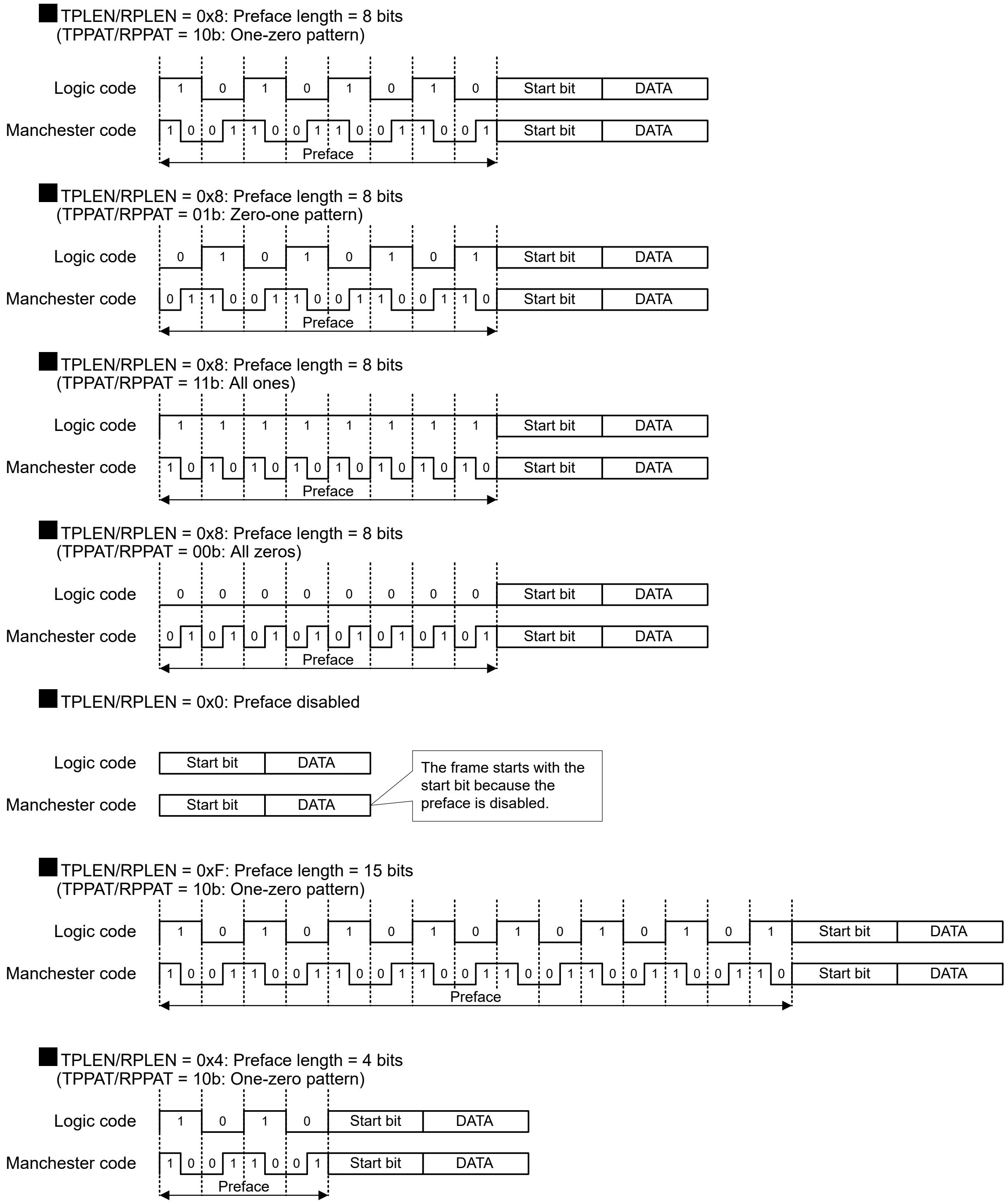


Figure 38.47 Preface Pattern Setting Example

(2) Start bit area

This is an area indicating the start of valid data in a frame. It is added after the preface area.

The start bit length is determined by MCR.SBSEL setting. When MCR.SBSEL = "0", the start bit length is 1 bit.

When MCR.SBSEL = "1", the start bit length is 3 bits.

When MCR.SBSEL = "1", the SYNC type can be selected from command SYNC and data SYNC.

Command SYNC means the three start bits are added as a one-to-zero transition.

Data SYNC means the three start bits are added as a zero-to-one transition.

The SYNC type is determined by the MCR.SYNSEL, MCR.SYNVAL and TDR.TSYNC settings.

(When receiving, the received result is applied to MSR.RSYNC.)

When MCR.SBSEL = “0”, the start bit is added as a zero-to-one or one-to-zero transition.

The selection is determined by the MCR.SYNVAL setting.

The MCR.SYNSEL bit specifies the destination to be referred to when setting for transmission.

When the MCR.SYNSEL bit is set to 1, the MCR.SYNVAL setting is referred to. When the MCR.SYNSEL bit is set to 0, the TDR.TSYNC setting is referred to.

Figure 38.48 shows the state of the start bit area according to the settings in the MCR.SYNSEL, MCR.SYNVAL and TDR.TSYNC registers in the case of transmission. Figure 38.49 shows that in the case of reception.

The start bit(s) is not affected by the MCR.TMPOL or MCR.RMPOL setting.

*d.c. = don’t care, Preface = preface area, START = start bit area, Data = user data area,
P = parity bit, MP = multi-processor bit, STOP = stop bit

Register setting				Output signal	
MCR			TDR	Output TXDn waveform	Output Start bit
SBSEL	SYNSEL	SYNVAL	TSYNC		
0	d.c.	0	d.c.		1bit 0 to 1 transition
0	d.c.	1	d.c.		1bit 1 to 0 transition
1	0	0	d.c.		3bit 0 to 1 transition (DATA SYNC)
1	0	1	d.c.		3bit 1 to 0 transition (COMMAND SYNC)
1	1	d.c.	0		3bit 0 to 1 transition (DATA SYNC)
1	1	d.c.	1		3bit 1 to 0 transition (COMMAND SYNC)

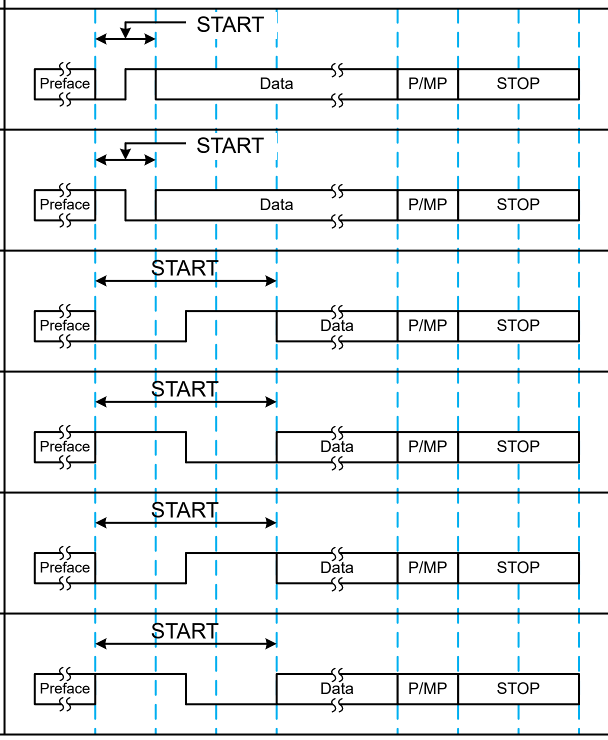
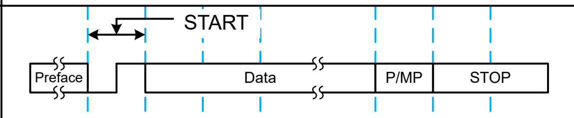
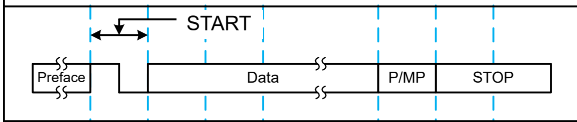
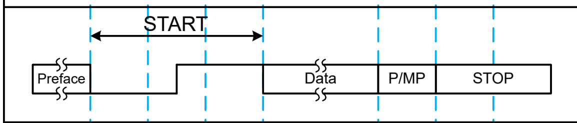
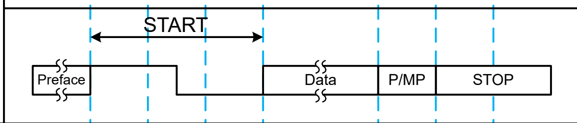
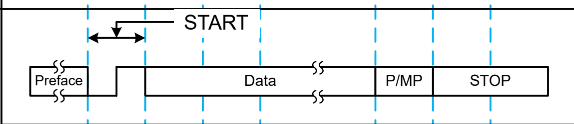
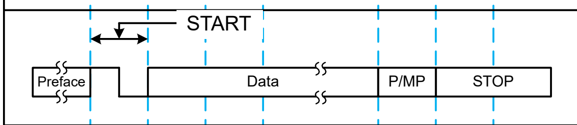
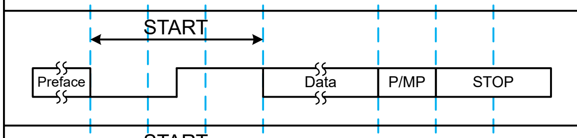
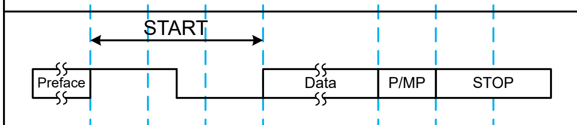
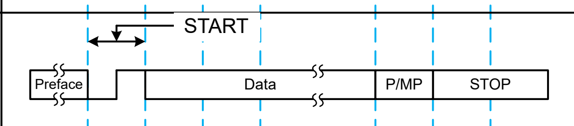
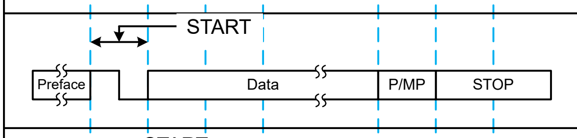
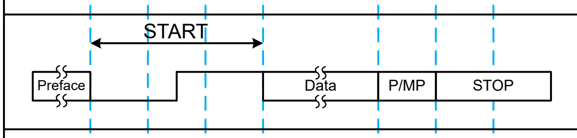
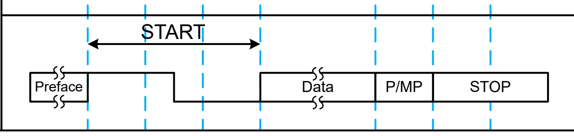


Figure 38.48 Settings Related to and Format of the Start Bit Area at Transmission

d.c. = don't care, Preface = Preface area, START = Start bit area, Data = Data area
P = Parity bit, MP = Multi-processor bit, STOP = Stop bit

Register setting				Input signal	Start bit detection result*1	Register indication
MCR			TDR	RXDn input waveform		MSR.RSYNC
SBSEL	SYNSEL	SYNVAL	TSYNC			
0	d.c.	0	d.c.		Normal start bit (1 bit: 0-to-1 transition)	0
					Start bit error	0
					Start bit error	0
					Start bit error	0
0	d.c.	1	d.c.		Start bit error	0
					Normal start bit (1 bit: 1-to-0 transition)	0
					Start bit error	0
					Start bit error	0
1	d.c.	d.c.	d.c.		Start bit error	0
					Start bit error	0
					Data SYNC	0
					Command SYNC	1

Note 1. Data other than the start bit is assumed to be normal.

Figure 38.49 Settings Related to and Judgment of the Start Bit Area at Reception

(3) DATA

Since the format of the data area is the same as that of the Asynchronous mode, see section 38.3.1. Serial Data Transfer Format.

As shown in Figure 38.45, Frame Format in Manchester Mode, the stop bit is not included in the Manchester encoding range.

38.5.2 Clock

As the transfer clock in Manchester mode, the clock generated by the on-chip baud rate generator is used by setting the CCR2.CKS[1:0] bit.

Also it is possible to set the oversampling (transfer rate of one-bit period) by CCR2.ABCS bit.

When the CCR2.ABCS bit is set to 0, oversampling x16 is selected with the one-bit period being 16 cycles of the base clock. When the CCR2.ABCS bit is set to 1, oversampling x8 is selected with the one-bit period being 8 cycles of the base clock.

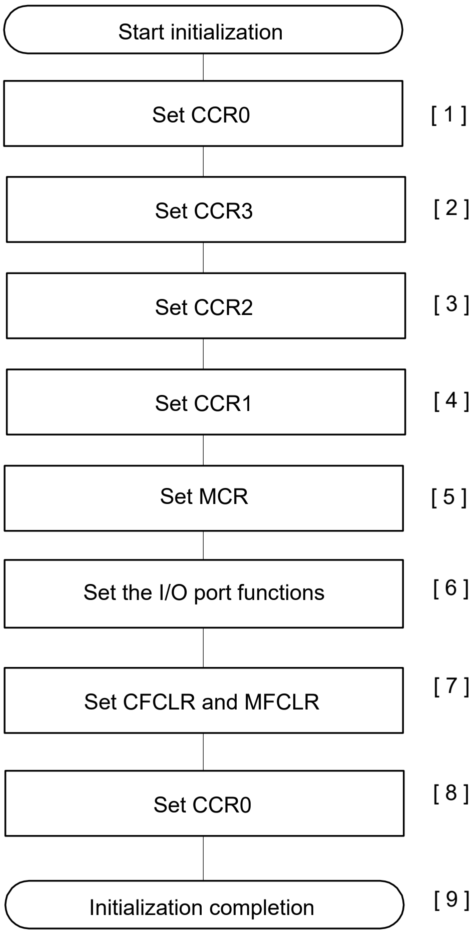
38.5.3 Initialization of the SCI in Manchester Mode

Before transferring data, write 0 to CCR0.TE and CCR0.RE (or write the initial value to CCR0 register) and initialize the SCI following the example of flowchart shown in Figure 38.50.

Whenever the operating mode or transfer format is changed, the CCR0 register must be initialized before the change is made.

Note that setting the CCR0.RE bit to 0 initializes none of the ORER, FER, PER, RDRF, and RDF flags in the CSR register, the SYER, PFER, MER and SBER flags in the MSR register, and the RDR registers.

Note also that switching the value of CCR0.TE from 0 to 1 when CCR0.TIE is 1 generates a SCIn_TXI interrupt request.



[1] Set the following bits of CCR0 to 0.
TEIE, TIE, RIE, TE, RE
If you have not changed from the initial settings, you can skip this step.

[2] Set the driver control, Multi-processor communication, the communication mode, transmission/reception format, and Clock enable.
Do not use function bits such as CPOL and CPHA at the initial value.

[3] Set the clock selection, Bit rate setting.
Leave unused function bits at their initial values.(ex. BRME)

[4] Set the noise filter, loopback control, communication pin status, parity check, and CTS / RTS function.

[5] Set Manchester transmission / reception setting.
Start bit setting, retiming setting, Manchester polarity setting, transmission preface setting, reception preface setting.

[6] Set the I / O port so that the required pin function is valid among the TXDn, RXDn, and SCKn pins.

[7] Set 1 to the following clear bits in the CFCLR and MFCLR registers to clear the target flag.
CFCLR.RDRFC, FERC, PERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
MFCLR.MERC, SBERC, SYERC, PFERC
This procedure can be omitted for the initial setting after reset.

[9] Set the TE bit or RE bit to 1.
To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time, and multiprocessor function. The TXDn and RXDn pins can be used by setting the TE and RE bits.

Figure 38.50 SCI Initialization Flow in Manchester Mode

38.5.4 Double-speed Operation

When the ABCS bit in CCR2 is set to 1 and eight pulses of the base clock for a 1-bit period is selected, the SCI operates on the bit rate twice that of when ABCS is set to 0.

When the BGDM bit in CCR2 is set to 1, the cycle of the base clock is reduced to half and the SCI operates on the bit rate twice that of when ABCS is set to 0.

When the ABCS and the BGDM bits in CCR2 are set to 1, the SCI operates on the bit rate four times that of when the ABCS and the BGDM bits in CCR2 are set to 0.

38.5.5 CTS and RTS Functions

The CTS function uses input on the CTSn pin in transmission control. Setting the CTSE bit in CCR1 to 1 enables the CTS function. The CTSn_RTSn pin can be set as a multiplexed pin which allows one pin to be used for either CTS or RTS function, or as dedicated pins with each pin at CTSn pin for CTS function and CTSn_RTSn pin for RTS function. Use the CTSPEN bit in CCR1 for this setting.

When the CTS function is enabled, reception starts only when the CTSn pin is at the low level.

Applying a high level to the CTSn pin after transmission starts does not affect transmission of the current frame, which continues.

The RTS function uses output on the CTSn_RTSn pin to request transmission. When the SCI is ready to receive, it outputs a low level to the CTSn_RTSn pin, Conditions for output of the low level and high level are as follows:

[Conditions for low-level output]

When all conditions listed below are satisfied:

- The value of the RE bit in CCR0 is 1.
- When The SCI is ready to receive next.
  - When there is no received data yet to be read and not receiving
  - All of the following flags are set to 0: CSR.ORER, FER, and PER, MSR.MER, SYER (when SYEREN = 1), PFER (when PFEREN = 1) and SBER flags (when SBEREN = 1).

[Conditions for high-level output]

- When the conditions for low output are not satisfied

38.5.6 Serial Data Transmission in Manchester Mode

The SCI encodes data in Manchester encoding and sends the resultant data in Manchester mode.

When the polarity setting (MCR.TMPOL) set to 0, logic 0 is coded as a zero-to-one transition in Manchester code and logic 1 is coded as a one-to-zero transition in Manchester code.

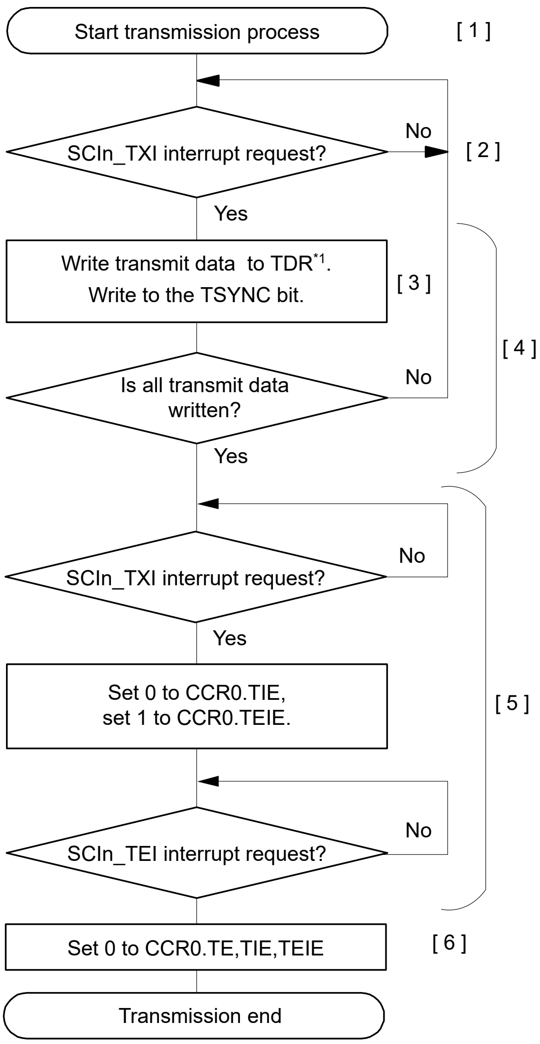
When the polarity setting (MCR.TMPOL) set to 1, logic 0 is coded as a one-to-zero transition in Manchester code and logic 1 is coded as a zero-to-one transition in Manchester code.

For this reason, a level transition occurs with the Manchester encoded data in the middle of individual logic data. (See Figure 38.45).

The transmitter constructs transmit frames in a specific format by adding a preface area to data and setting the start bit(s) according to the polarity setting and sends resultant serial data.

For details on the frame format, see section 38.5.1. Frame Format.

Figure 38.51 shows the flowchart in transmission. At transmission starts, set the CCR0.TIE and CCR0.TE bits to 1 simultaneously with one instruction. Then, a SCIn_TXI interrupt request is generated. Figure 38.52, Figure 38.53, and Figure 38.54 show examples of the operation for serial transmission in Manchester mode.



[1] Start transmission process

[2] SCIn_TXI interrupt:
A transmit data empty interrupt (SCIn_TXI) request is generated when CCR0.TIE = 1 and TE = 0->1, and when transmit data is transferred from TDR to TSR.

[3] TDR setting:
Write the transmit data to TDR.
When TSYNC is used, also set TDR.TSYNC.

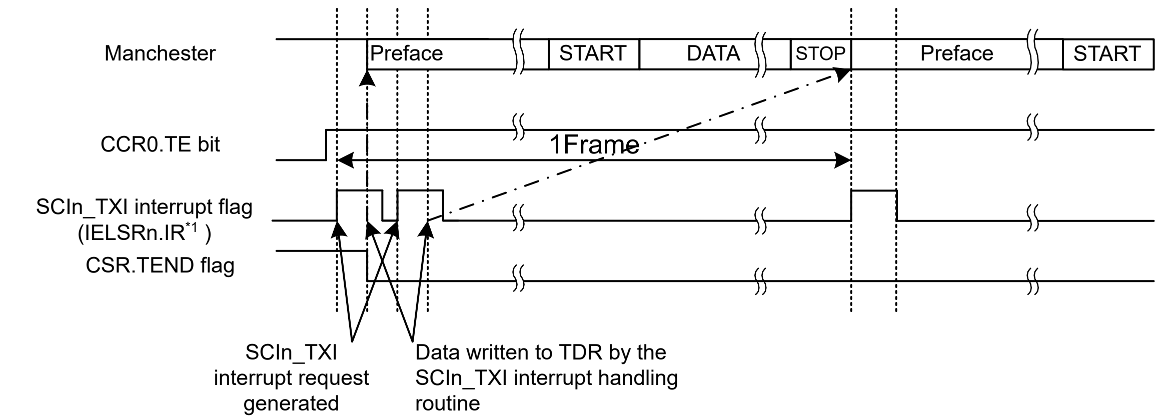
[4] Serial transmission continuation procedure:
To continue serial transmission, write the transmit data to TDR*1 once by using an SCIn_TXI interrupt.
The transmit data can also be written to TDR*1 by activating the DMAC or DTC in the SCIn_TXI interrupt request.

[5] When SCIn_TEI interrupt requests are used, set the CCR0.TIE bit to 0 and the CCR0.TEIE bit to 1 after the last of the data to be transmitted is written to TDR and it's transmission is started.

[6] Set 0 to CCR0.TE, TIE and TEIE.

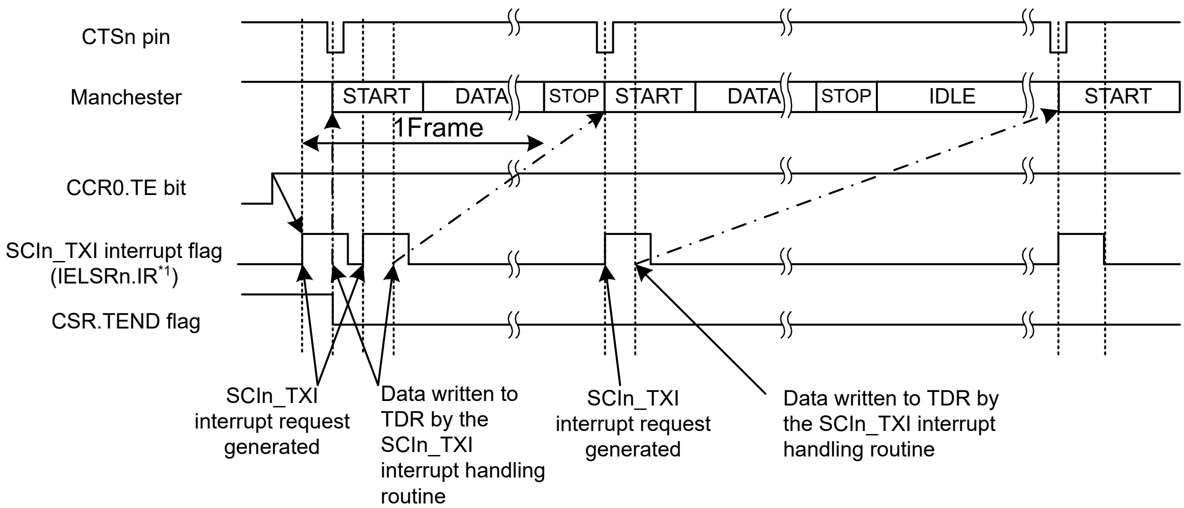
Note 1. If TDR[15:8] requires data write and byte access is required, write the transmit data in the order of TDR [15:8] and TDR [7:0].

Figure 38.51 Example of Serial Transmission Flowchart in Manchester Mode



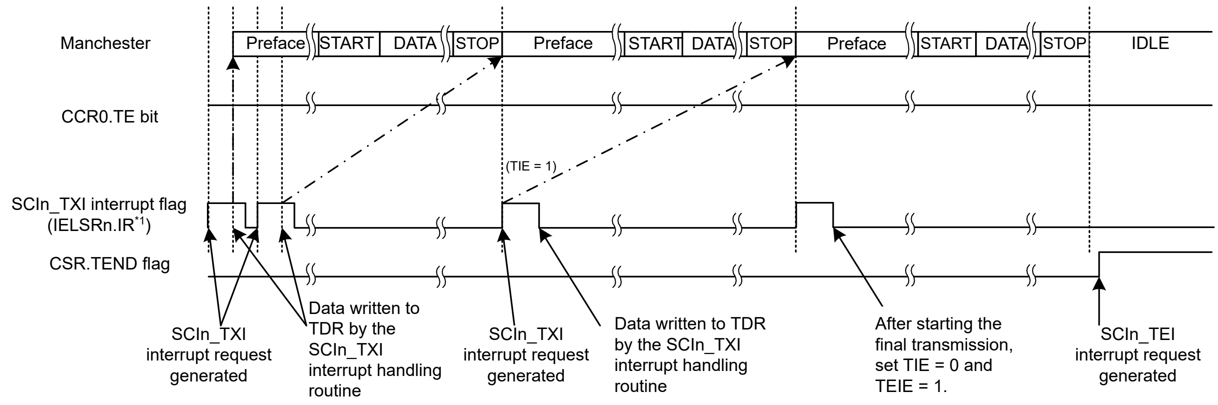
Note 1. For corresponding interrupt numbers, see section 14, Interrupt Controller Unit (ICU)

Figure 38.52 Example of Start-of-Transmission Operation for Serial Transmission in Manchester mode (with Preface but Without the CTS Function)



Note 1. For corresponding interrupt numbers, see section 14, Interrupt Controller Unit (ICU)

Figure 38.53 Example of Start-of-Transmission Operation for Serial Transmission in Manchester Mode (Without Preface but with the CTS Function)



Note 1. For corresponding interrupt numbers, see section 14, Interrupt Controller Unit (ICU)

Figure 38.54 Example of End-of-Transmission Operation for Serial Transmission in Manchester Mode (with Preface but Without the CTS Function)

38.5.7 Serial Data Reception in Manchester Mode

In Manchester mode, the SCI operates on a base clock with a frequency of 16 times*1 the bit rate. Reception starts by sampling the falling edges of received data at the base clock. As shown in Figure 38.55, reception starts at a falling edge of the received data and it continues if the received data keeps low for the duration of 1/4 bit. If the received data goes high within the duration of 1/4 bit, the SCI judges it as an error and waits for a falling edge again.

If a high level is expected in the first half of a bit in the received data, the SCI judges a low level that continues for one base clock cycle as an error and ignores the change to the low level.

Note 1. This is the case when CCR2.ABCS = 0. When CCR2.ABCS = 1, the SCI operates on a base clock with a frequency of 8 times the bit rate.

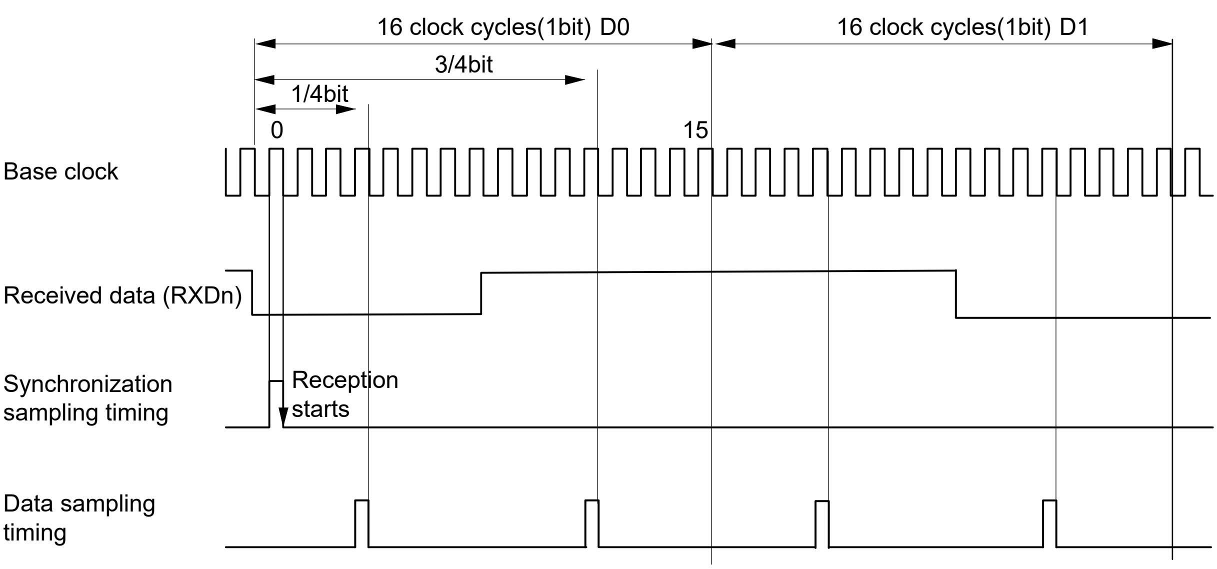


Figure 38.55 Data Reception Sampling Timing in Manchester Mode

In Manchester mode, data reception starts with detection of a preface and start bit area.

The SCI checks the input from the RXDn pin to see whether a preface is added based on the value of MCR.RPLEN.

If the preface is disabled (MCR.RPLEN = 0), it moves on to the detection of a start bit area without detecting a preface.

When a preface is enabled, it identifies a preface pattern setting according to the set value in MCR.RPPAT, and compares it with the RXDn input for a pattern match to detect a preface pattern.

Upon detection of a preface pattern match, it judges it as a normal preface and moves on to the detection of a start bit area.

If detecting a preface pattern mismatch or a Manchester code error in the preface area, it judges it as a preface error and asserts a preface error (PFER).

For start bit detection, the SCI selects an expected value based on the register settings (MCR.SBSEL and SYNVAL), compares it with the RXDn input for a pattern match to detect a start bit area. Upon detection of a start bit pattern match, it judges it as a normal start bit area and moves on to the data processing.

Only when a preface and a start bit area are detected normally, it moves on to the next phase of data reception.

Upon detection of a start bit pattern mismatch, it asserts a start bit error flag (SBER).

In data processing, the SCI shifts the data by the expected received data length based on the register settings (CCR3.CHR[1:0]) through the RSR register. If two sampling points in a bit of the received data are identical, the SCI judges this as a Manchester code error.

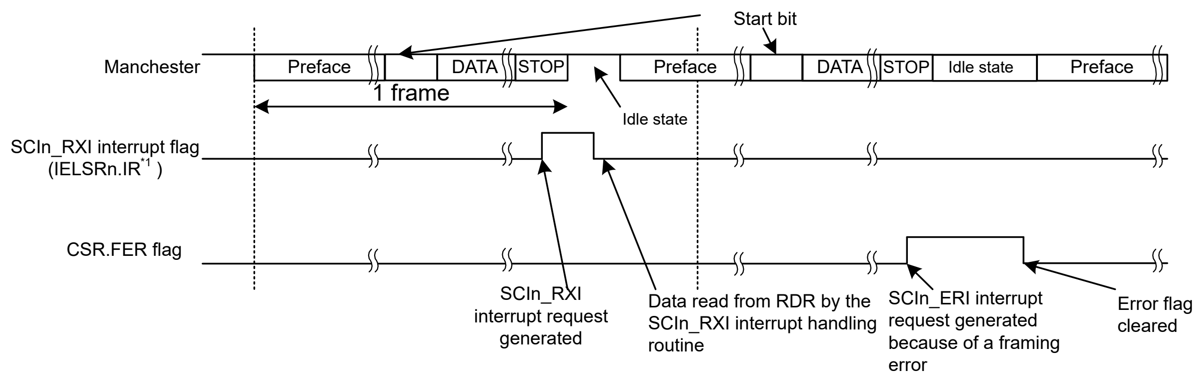
For details, see section 38.5.11. Errors in Manchester Mode (4).

When the parity function is disabled (CCR1.PE = 0), the SCI moves on to the next phase of stop bit detection. When the parity function is enabled (CCR1.PE = 1), the SCI performs parity checking. If detecting a parity error, it asserts a parity error flag (PER), and then moves on to stop bit detection.

In stop bit detection, the SCI checks the following in the stop bit area of the received frame:

It has two sampling points in a bit. If both points are at the high level, the bit is recognized as a normal stop bit and the data is stored in the RDR register. At least one low-level point is judged as an abnormal stop bit, causing a framing error flag (FER) to be set. Even when an error is detected, the received data is stored in the RDR register as abnormal data.

Figure 38.56 shows an example of the operation for serial data reception in Manchester mode.



Note 1. For corresponding interrupt numbers, see section 14, Interrupt Controller Unit (ICU).

Figure 38.56 Example of Operation for Serial Data Reception in Manchester mode (with a Preface)

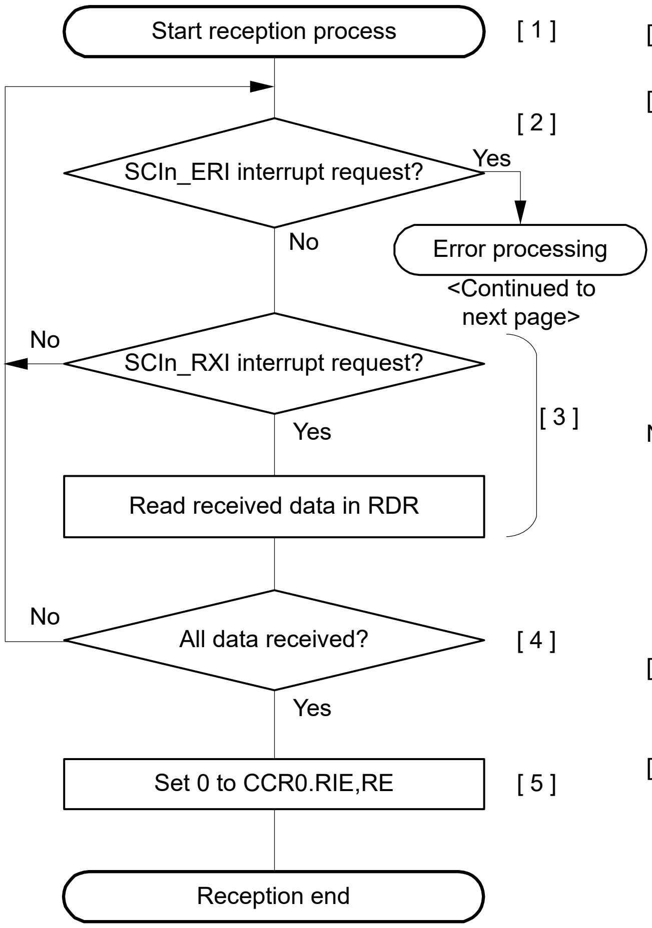
For the state of each status flag in the CCR0 register and RXDn input processing when a receive error is detected, see section 38.5.11. Errors in Manchester Mode.

If a receive error is detected, an SCIn_ERI interrupt request is generated but an SCIn_RXI interrupt request is not generated.

Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, FER, PER, MER, SYER*1, PFER*1, and SBER*1 flags to 0 before resuming reception. Also, be sure to read the RDR register during overrun error processing. When a reception is forcibly terminated by setting the CCR0.RE bit to 0 during operation, read the RDR register because received data which has not yet been read may be left in the RDR register.

Figure 38.57 and Figure 38.58 show examples of serial data reception flowchart in Manchester mode.

Note 1. Effective when the corresponding bit is enabled.



[1] Start reception process

[2] Receive error processing:
When a receive error occurs, SCI generates an SCIn_ERI interrupt . An error is identified by reading the ORER, PER, FER in CSR and MER, SBER, SYER, PFER in MSR. After performing the appropriate error processing, always set the ORER, PER, FER, MER, SBER, SYER, PFER flags to 0. Reception cannot be resumed if any of these flags is set to 1.

Note: Each of the PFER, SYER, SBER bits in MSR has an enable bit (PFEREN, SYEREN, and SBEREN in MECR). If the corresponding enable bit is set to 0 (disable), the flag value does not affect the communication. Perform error processing as required.

[3] Read the received data in RDR once in the SCIn_RXI interrupt handling routine.

[4] Serial reception continuation procedure:
To continue serial reception, before the stop bit of a frame is received, read data from RDR in the SCIn_RXI interrupt handling routine. The RDR data can also be read by activating the DMAC or DTC.

[5] Write 0 to CCR0.RIE and RE.

Figure 38.57 Example of Serial Data Reception Flowchart in Manchester Mode (Normal Reception)

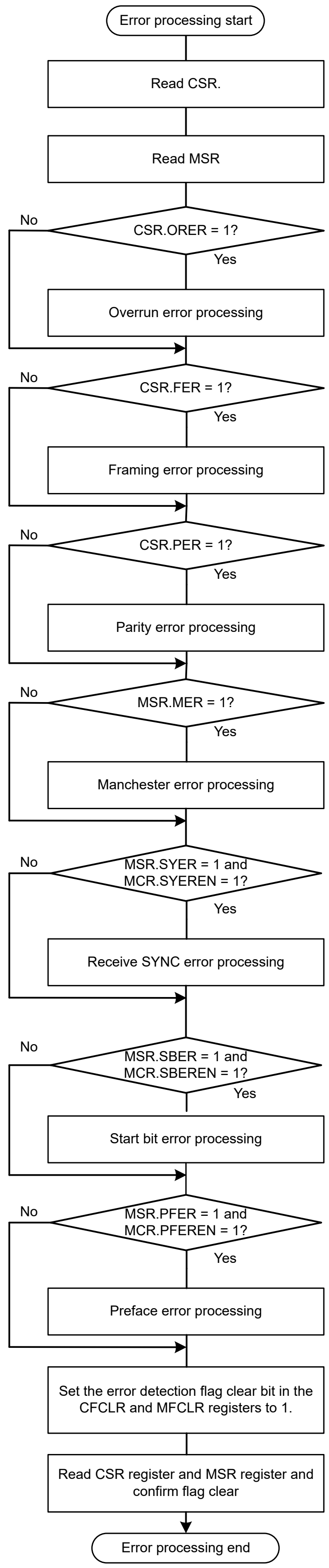


Figure 38.58 Example of Serial Reception Flowchart in Manchester Mode (Error Processing)

38.5.8 Operation When Multi-Processor Bit Is Used

See section 38.4. Multi-Processor Communication Function (1) for the operation in Manchester mode when using multi-processor mode because the operation is the same.

A preface and a start bit area are added to the frame format in Manchester mode. See Figure 38.58 for error processing in Manchester mode for the reception flowchart (Figure 38.41). See Table 38.36 for the operation status when detecting various errors.

38.5.9 Receive Retiming

This function corrects the timing for each central edge of the bit, taking advantage of the fact that each bit has an edge in the center in Manchester code.

The receive retiming function can be turned on or off by setting the ERTEN bit in the MCR register.

When the receive retiming function is turned off (MCR.ERTEN = 0), retiming is not performed, causing misalignment between the internal clock and the RXDn input to be accumulated and the receive margin to be reduced.

When the receive retiming function is turned on (MCR.ERTEN = 1), retiming is performed for the preface area, the start bit area[*1], and the data area (excluding the stop bit).

Note 1. Retiming is not performed for the start bit area if the preface length is 0 and the start bit length is 3.

As an example, the receive retiming when oversampling x16 is selected is shown below.

When detecting an RXDn input edge two to four cycles before the expected receive cycle, the receive processing is shortened by one sampling CLK cycle.

When detecting a RXDn input edge two to three cycles after the expected receive cycle, the receive processing is extended by one sampling CLK cycle.

(Even if the clock is misaligned with the data by more than two cycles, one cycle is corrected for each bit.)

Figure 38.59 shows the conceptual image of receive retiming range.

When detecting an edge in the tolerance area in the figure, data is received as is without making correction.

When detecting an edge in the SyncJump area in the figure, data is corrected for reception.

When detecting an edge in the SyncError area in the figure, data is received as abnormal data with no correction made.

For a Manchester code error (data matches at the 1/4-phase and 3/4-phase sampling points), the SCI reports a code error.

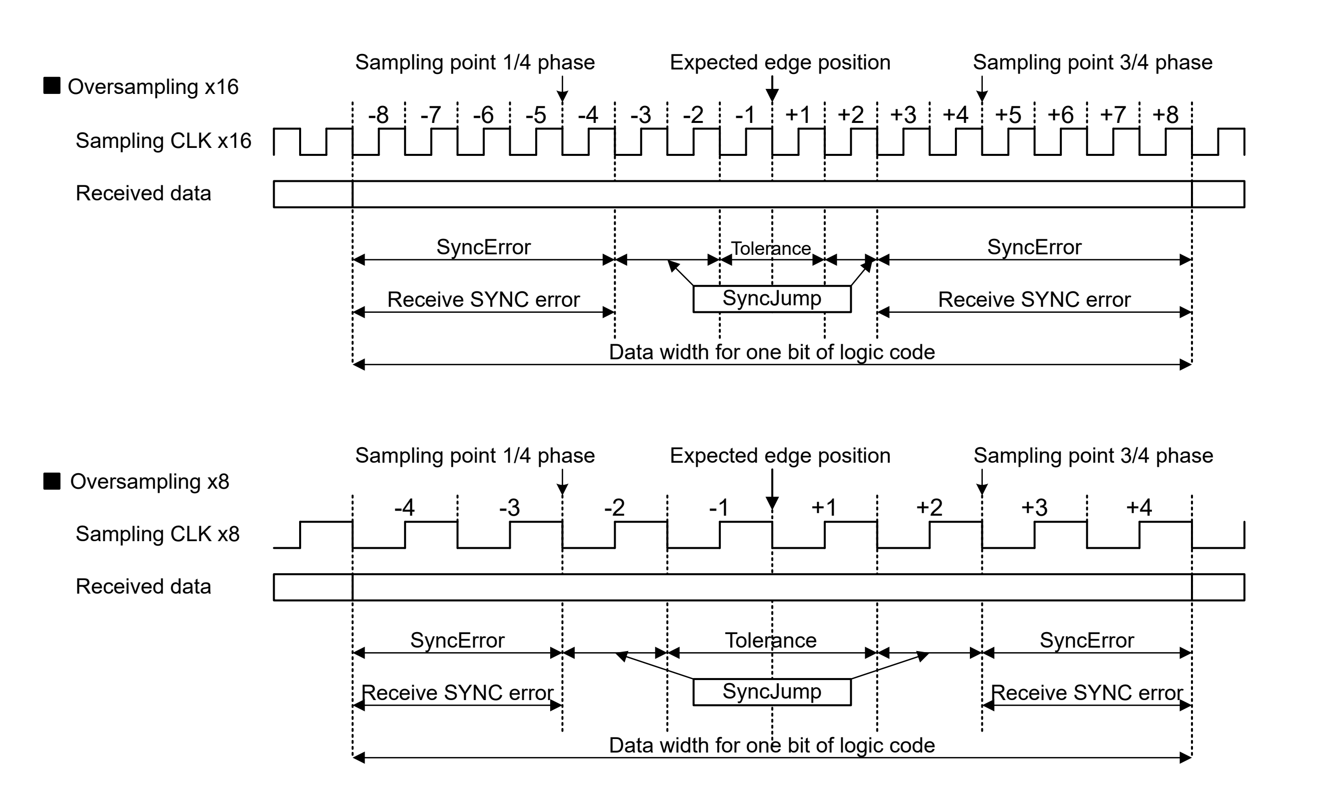


Figure 38.59 Conceptual Image of Reception Retiming Range

38.5.10 Polarity Setting for Manchester Code

The polarity of the Manchester code can be set with the Manchester Control Register (MCR).

It can be set separately for transmission and reception. Use the MCR.TMPOL bit to set the polarity for transmission and the MCR.RMPOL bit to set the polarity for reception.

The Manchester code polarity setting is valid for the preface area, the data area, and the parity or multi-processor area.

When the initial settings (TMPOL/RMPOL = 0) are used for the polarity of Manchester code, logic 0 is encoded as a zero-to-one transition in Manchester code and logic 1 is encoded as a one-to-zero transition in Manchester code. If the settings are changed to TMPOL/RMPOL = 1, logic 0 is encoded as a one-to-zero transition in Manchester code and logic 1 is encoded as a zero-to-one transition in Manchester code. Figure 38.60 shows the conceptual image of the settings and operation.

Separately from the function above, the transmitted and received data in the data area can be inverted by the transmitted/received data inversion function (CCR3.SINV). Since the polarity of Manchester code (MCR.TMPOL/RMPOL) can be set separately from the transmitted/received data invert function (CCR3.SINV), if both are set to inversion (MCR.TMPOL/RMPOL = 1 and CCR3.SINV = 1), the transmitted and received data are set to initial state (inversion + inversion = normal).

The polarity of the start bit area can be set by a register different from the ones mentioned above.

Since a different register is used, the polarity of the start bit area is not affected by the polarity setting for Manchester code mentioned above.

For details on the setting for the start bit area, see section 38.5.1. Frame Format (2).

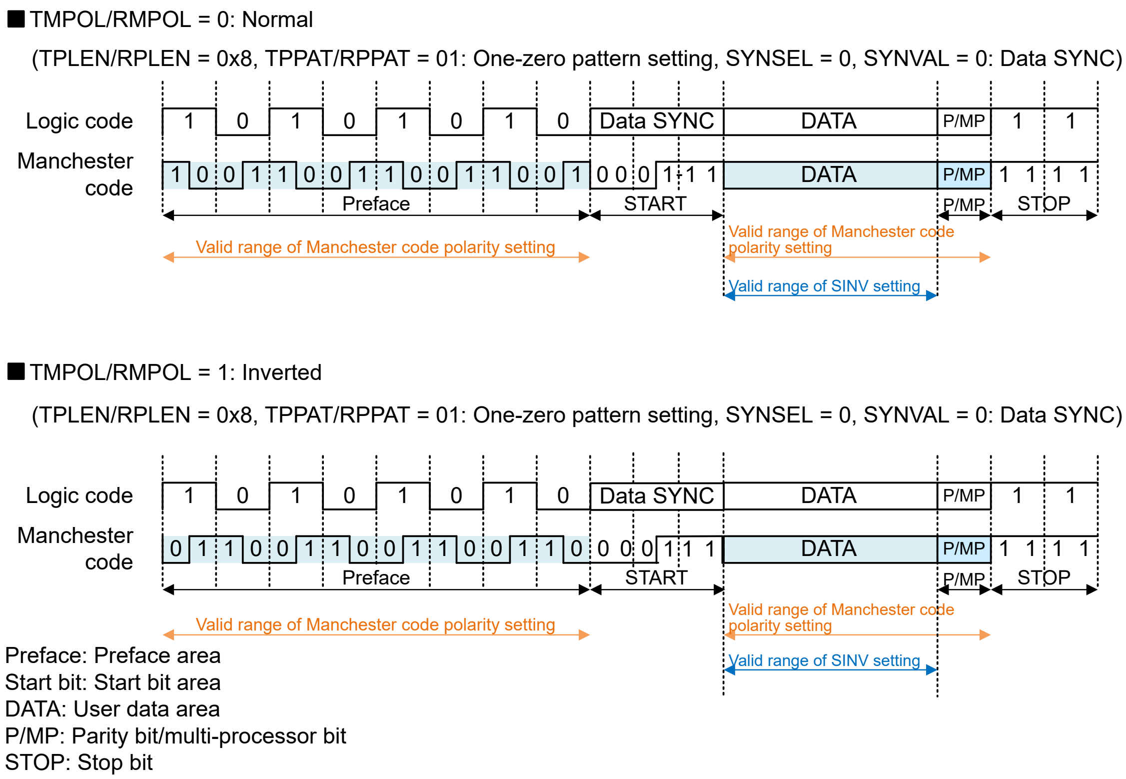


Figure 38.60 Valid Range of the Manchester Code Polarity Setting

38.5.11 Errors in Manchester Mode

There are the following errors in Manchester mode:

1. Parity error
2. Over run error
3. Framing error
4. Manchester error
5. Preface error
6. Start Bit error
7. Receive SYNC error

For errors (1) to (3), see section 38.3.9. Serial Data Reception in Asynchronous Mode (1) because they are the same as in Asynchronous mode.

Each error is judged in each area, but they are reflected on flags and operations at the timing of 3/4-bit sampling of the STOP bit area. If a preface error or start bit error is detected, subsequent data will not be received. Therefore, no other error detection is performed, and the error flag holds the previous information.

Table 38.34 lists the states of the serial status register when detecting errors and judgment about whether to store data in the RDR.

Table 38.35 lists the errors that can be detected in each area of a Manchester frame.

If a Preface error or Start bit error is detected, subsequent data will not be received. Therefore, no other error detection is performed, and the error flag holds the result of the previous frame reception. Also, if an error is detected in the previous frame, data will not be received, but errors in the pre-face area and start bit area will update that flag. Table 38.36 shows the flags and actions in this case.

(4) Manchester error

A Manchester error is generated when a Manchester code error is detected.

In Manchester code, there must be an edge (transition) in the center of the bit.

In the data area of a received frame (including the parity/multi-processor bit), the values of the 1/4-bit and 3/4-bit sampling points are checked in each received 1-bit data, and a Manchester code error is determined if these two values match.

If a Manchester code error is detected, the Manchester error flag (MSR.MER) is asserted.

If a Manchester error occurs, it is handled as an interrupt source and event source. If a Manchester error is detected, the next reception is not performed until the corresponding error flag is cleared.

(5) Preface error

A preface error is generated when the preface pattern does not match or a Manchester code error is detected in the preface area. If a preface error is detected, the preface error flag (MSR.PFER) is asserted.

It is possible to set whether to use this error flag as an interrupt source with the setting of the MCR register.

When MCR.PFEREN = 1, a preface error is handled as an interrupt source or event source. If a preface error is detected, the next reception is not performed until the corresponding error flag is cleared.

When MCR.PFEREN = 0, a preface error is not handled as an interrupt source or event source, and the next reception is not halted. However, a preface error is notified to MSR.PFER.

(6) Start bit error

A start bit error is generated when a mismatch is detected between the start bit area in the received frame and the preset start bit pattern. Upon detection of a start bit error, a start bit error flag (MSR.SBER) is asserted.

It is possible to set whether to use the start bit error as an interrupt source with the setting of the MCR register.

When MCR.SBEREN = 1, a start bit error is handled as an interrupt source or event source. If a start bit error is detected, the next reception is not performed until the corresponding error flag is cleared.

When MCR.SBEREN = 0, a start bit error is not handled as an interrupt source or event source, and the next reception is not halted. However, a start bit error is notified to MSR.SBER.

(7) Receive SYNC error

When the receive retiming function described in section 38.5.9. Receive Retiming is enabled, the receive retiming operation is performed.

If no edges are detected within the receive retiming range (SyncError area in Figure 38.59) when receive timing operation is being performed, a receive SYNC error is generated. Upon detection of a receive SYNC error, a receive SYNC error flag (MSR.SYER) is asserted. In areas not subject to retiming, receive SYNC errors are not detected.

The preface area[*1], the start bit area[*1,*2], and the data area (excluding the stop bit) for which receive retiming operation is performed are checked.

It is possible to set whether to use the receive SYNC error as an interrupt source with the setting of the MCR register.

When MCR.SYEREN = 1, a receive SYNC error is handled as an interrupt source or event source. If a receive SYNC error is detected, the next reception is not performed until the corresponding error flag is cleared.

When MCR.SYEREN = 0, a receive SYNC error is not handled as an interrupt source or event source, and the next reception is not halted. However, a receive SYNC error is notified to MSR.SYER.

Note 1. In the case of a frame that starts with a pattern that expects the first half of the bit to be High, it is excluded from retiming.

Note 2. In the start bit area, when there is no preface length and 3 bit start bit is set, it is not subject to retiming.

Also, the 1st bit and the 2nd bit in the start bit area when 3 bit start bit is set are not subject to retiming.

Table 38.34 Flags in the CSR Register and Receive Data Handling in Manchester Mode

Flag in the CSR register			Flag in the MRS register				Received data	Received error status (SCIn_ERI interrupt / event generation)
ORER	FER	PER	MER	SBER*1	PFER*1	SYER		
0	0	0	0	0	0	0	Transfer to RDR	No error
0	1	0	0	0	0	0	Transfer to RDR	Framing error
0	0	1	0	0	0	0	Transfer to RDR	Parity error
0	1	1	0	0	0	0	Transfer to RDR	Framing error + Parity error
0	0	0	1	0	0	0	Transfer to RDR	Manchester error
0	1	0	1	0	0	0	Transfer to RDR	Framing error + Manchester error
0	0	1	1	0	0	0	Transfer to RDR	Parity error + Manchester error
0	1	1	1	0	0	0	Transfer to RDR	Framing error + Parity error + Manchester error
1	0	0	0	0	0	0	Lost	Overrun error
1	1	0	0	0	0	0	Lost	Overrun error + Framing error
1	0	1	0	0	0	0	Lost	Overrun error + Parity error
1	1	1	0	0	0	0	Lost	Overrun error + Framing error + Parity error
1	0	0	1	0	0	0	Lost	Overrun error + Manchester error
1	1	0	1	0	0	0	Lost	Overrun error + Framing error + Manchester error
1	0	1	1	0	0	0	Lost	Overrun error + Parity error + Manchester error
1	1	1	1	0	0	0	Lost	Overrun error + Framing error + Parity error + Manchester error
0	Combination of above			0	0	1	Transfer to RDR	Errors above + Receive SYNC error*2
1				0	0	1	Lost	Errors above + Receive SYNC error*2
hold	hold	hold	hold	0	1	0	Lost	Preface error*3
hold	hold	hold	hold	1	0	0	Lost	Start bit error*3
hold	hold	hold	hold	0	1	1	Lost	Preface error*3 + Receive SYNC error*2
hold	hold	hold	hold	1	0	1	Lost	Start bit error*3 + Receive SYNC error*2

Note 1. Start bit error and Preface error never become 1 at the same time.
Note 2. When MCR.SYEREN = 1, SCIn_ERI interrupt / event is generated by SYER factor.
Note 3. If MCR.PFEREN = 1 or MCR.SBEREN = 1, an SCIn_ERI interrupt / event is generated when the corresponding flag is set.

Table 38.35 Errors Detectable in Each Area

	Preface error (PFER)	Start Bit error (SBER)	Manchester error (MER)	Receive SYNC error (SYER)	Parity error (PER)	Framing error (FER)
Preface area	✓	—	—*1	✓*2	—	—
Start Bit area	—	✓	—	✓*2	—	—
Data area	—	—	✓	✓	—	—
Parity area	—	—	✓	✓	✓	—
Multi-processor area	—	—	✓	✓	—	—
Stop Bit area	—	—	—	—	—	✓

Note: ✓: Detected, —: Not detected
Note 1. When a Manchester code error occurs in the preface area, it is defined as a preface error.
Note 2. It may not be subject to Receive SYNC error detection. For details see the text section 38.5.11. Errors in Manchester Mode (7)

Table 38.36 Operation status due to presence / absence of error in previous frame and operation status list in multiprocessor mode (1 of 2)

Previous frame	Each area of the Frame					PFEREN	SBEREN	SYEREN	Received data	Error flag	Interrupt request	Event signal
	preface	start bit	data	parity	stop							
No Error	PFER No SYER*1	No Error	Don't Care	Don't Care	Don't Care	0	Don't Care	Don't Care	Lost	Set PFER*1	Not output	Not output
						1					Output	Output
	No Error	SBER No SYER*1	Don't Care	Don't Care	Don't Care	Don't Care	0	Don't Care	Lost	Set SBER*1	Not output	Not output
							1				Output	Output
	SYER No PFER	No Error	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0	Transfer to RDR	Set SYER	Not output	Not output
								1	Lost		Output	Output
	No Error	SYER No SBER	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0	Transfer to RDR	Set SYER	Not output	Not output
								1	Lost		Output	Output
	No Error	No Error	SYER		No Error	Don't Care	Don't Care	0	Transfer to RDR	Set SYER	Not output	Not output
								1			Output	Output
	No Error	No Error	MER		No Error	Don't Care	Don't Care	Don't Care	Transfer to RDR	Set MER	Output	Output
	No Error	No Error	Don't Care	PER	No Error	Don't Care	Don't Care	Don't Care	Transfer to RDR	Set PER	Output	Output
	No Error	No Error	Don't Care	Don't Care	FER	Don't Care	Don't Care	Don't Care	Transfer to RDR	Set FER	Output	Output
	There is some error ORER					Don't Care	Don't Care	Don't Care	Lost	Set some flags*2	Output	Output
	No Error	No Error	No Error	No Error	No Error ORER	Don't Care	Don't Care	Don't Care	Lost	Set ORER	Output	Output

Table 38.36 Operation status due to presence / absence of error in previous frame and operation status list in multiprocessor mode (2 of 2)

Previous frame	Each area of the Frame: preface	start bit	data	parity	stop	PFEREN	SBEREN	SYEREN	Received data	Error flag	Interrupt request	Event signal
Some error*3 *6	PFER No SYER*1	No Error	Don't Care	Don't Care	Don't Care	0 1	Don't Care	Don't Care	Lost	Set PFER*1	Output*4	Not output*5
	No Error	SBER No SYER*1	Don't Care	Don't Care	Don't Care	Don't Care	0 1	Don't Care		Set SBER*1		
	SYER No PFER	No Error	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0 1		Set SYER		
	No Error	SYER No SBER	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0 1		Set SYER		
	No Error	No Error	SYER		No Error	Don't Care	Don't Care	0 1		Don't set any flags		
	No Error	No Error	MER		No Error	Don't Care	Don't Care	Don't Care				
	No Error	No Error	Don't Care	PER	No Error	Don't Care	Don't Care	Don't Care				
	No Error	No Error	Don't Care	Don't Care	FER	Don't Care	Don't Care	Don't Care				
	There is some error ORER					Don't Care	Don't Care	Don't Care				
	No Error	No Error	No Error	No Error	No Error ORER	Don't Care	Don't Care	Don't Care				

Note 1. If SYER is detected, the SYER flag is also set. Other operations are as shown in this table.
Note 2. Other detected error flags including ORER are also set.
Note 3. If all the error flags are cleared before the STOP bit is judged, the operation will be the same as the case where there is no error in the previous frame of this table.
Note 4. Since the SCIn_ERI interrupt request is level output, it remains active due to errors in the previous frame regardless of the presence or absence of error in the relevant frame.
Note 5. Since the error cause is continuously detected, the SCIn_ERI event is not newly output regardless of the presence or absence of errors in the relevant frame.
Note 6. For PFER, SBER, and SYER, when each enable bit is set to disable, it is treated as no error.

Table 38.37 Operation when MPIE = 1 in multi-processor mode (MPIE = 0)

MPB*1	Each area of the frame: preface	start bit	data	parity	stop	PFEREN	SBEREN	SYEREN	Received data	Error flag	Interrupt request	Event signal
1	No Error	No Error	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	Transfer to RDR	Set some flags	Output*2	Output*2
	No PFER	No SBER	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0				
	SYER*3	SYER*3						1	Lost	Don't set any flags	Not output	Not output
	PFER	No Error	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care				
	No Error	SBER	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care					

Note 1. If the received MPB bit is 0, it is not received the frame, and the operation is the same as lost of the reception data of this table.
Note 2. If no error is detected, SCIn_RXI interrupt request or event is output, and if it is detected, SCIn_ERI interrupt request or event is output.

Note 3. When SYER is detected in the preface area or the start bit area, the behavior of handling as an error depending on the SYEREN bit changes.

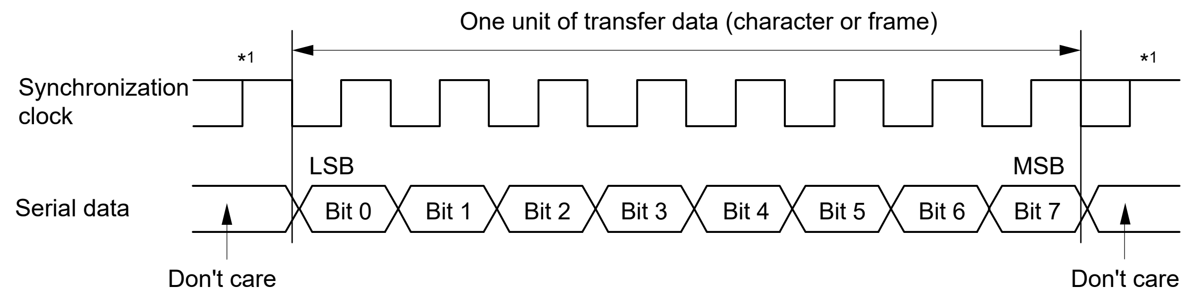
38.6 Operation in Clock Synchronous Mode

Figure 38.61 shows the data format for clock synchronous serial data communications.

In clock synchronous mode, data is transmitted or received in synchronization with clock pulses. For single-character data transfer, data consists of 8-bit. In clock synchronous mode, no parity bit can be added.

In data transmission when CPHA = 1 and CPOL = 1, the SCI outputs data from one falling edge of the synchronization clock to the next falling edge. In data reception, the SCI receives data in synchronization with the rising edge of the synchronization clock. After 8-bit data is output, the transmission line holds the last bit as output state. When the CPHA bit is 0 in slave mode, the transmission line holds the first bit output state.

Within the SCI, the transmitter and receiver are independent units, enabling full-duplex communications by using a shared communication clock of the transmitter and the receiver. Furthermore, because both the transmitter and the receiver also have a double-buffered structure, so that the next transmit data can be written during transmission or the previous receive data can be read during reception, enabling continuous data transfer.



Note 1. Holds a high level except during continuous transfer.

Figure 38.61 Data format in clock synchronous serial communications with LSB-first order

38.6.1 Clock

If the maximum speed of SCK = 1/2TCLK is set in Clock Synchronous and Simple SPI mode, Do not make PCLK less than half the speed of TCLK. If PCLK is made slower than this, malfunction may occur.

1. When the internal clock is selected
   When the SCI operates on an internal clock (CCR3.CKE[1:0] bits are set to 00b or 01b (master mode)), the synchronization clock is output from the SCKn pin. Eight synchronization clock pulses are output during single-character transmission/reception. When no data transfer is performed, the clock is held at high level.*1 In transmission-only or transmission/reception, the synchronization clock is not output unless transmit data is prepared.
   When the internal clock is selected, the clock with a delay from the SCKn signal is used for the master reception sampling clock. This ensures the data setup time and hold time for high-speed communication.

   Note 1. The signal is held high while (CCR3.CPHA = 0 and CCR3.CPOL = 1) or (CCR3.CPHA = 1 and CCR3.CPOL = 1). It is held low while (CCR3.CPHA = 0 and CCR3.CPOL = 0) or (CCR3.CPHA = 1 and CCR3.CPOL = 0).

2. When the external clock is selected
   When the CCR3.CKE[1:0] bits are set to 10b or 11b (slave mode), data is transmitted and received using the external clock that is input from the SCKn pin.

38.6.2 CTS and RTS Functions

In the CTS function, the CTSn_RTSn pin input controls the start of data reception or transmission when the clock source is the internal clock. Setting the CCR1.CTSE bit to 1 enables the CTS function. In clock synchronous communication, the CTS function can be used for the internal clock and the RTS function can be used for the external clock, so the CTS function and RTS function cannot be used at the same time. When the CTS function is enabled, setting the CTSn_RTSn pin low causes data reception or transmission to start.

Setting the CTSn_RTSn pin high while the data transmission or reception is in progress does not affect transmission or reception of the current frame.

In the RTS function, the CTSn_RTSn pin output is used to request the start of data reception or transmission when the clock source is an external synchronizing clock. The CTSn_RTSn output goes low when serial communication is enabled. Conditions for output of the CTSn_RTSn low and high are shown as follows:

[Conditions for low output]

Satisfaction of all the following conditions:

Non-FIFO selected when all of the following conditions are satisfied

- The value of the CCR0.RE bit or the CCR0.TE bit is 1
- Next serial communication is enabled.
  - No receive data is present before reading and not receiving. (when CCR0.RE bit = 1)
  - When the transmission data written in TDR is ready for transmission.*1 (when CCR0.TE bit = 1)
- The CSR.ORER flag is 0

Note 1. The CTSn_RTSn pin will be High after starting transmission.

FIFO selected when all of the following conditions are satisfied

- The value of the CCR0.RE bit or the CCR0.TE bit is 1
- Next serial communication is enabled.
  - The number of receive data stored in the receive FIFO (RDR register) is less than the value set in FCR.RSTRG[4:0] (when CCR0.RE bit = 1)
  - When the transmission data written in the transmission FIFO (TDR register) is ready for transmission.*1 (when CCR0.TE bit = 1)
- The CSR.ORER flag is 0

Note 1. The CTSn_RTSn pin will be High after the last data transmission starts.

[Condition for high output]

- The conditions for low output are not satisfied

38.6.3 SCI Initialization in Clock Synchronous Mode

Before transmitting and receiving data, start by writing the initial value 0x00 to the CCR0 register, then continue through the SCI initialization procedure given in the sections describing non-FIFO and FIFO selection in section 38.6.2. CTS and RTS Functions. Anytime the operating mode or transfer format is to be changed, the CCR0.TE and CCR0.RE must write to 0 before the change can be made.

Note: Setting the CCR0.RE bit to 0 initializes neither the ORER, FER, PER, and RDRF flags in CSR nor the RDR register. When the TE bit is set to 0, the TEND flag for the selected FIFO buffer is not initialized.

Note: Switching the value of the CCR0.TE bit from 1 to 0 when the CCR0.TIE bit is 1 generates an SCIn_TXI interrupt request.

Table 38.38 Example flow of SCI initialization in clock synchronous mode with non-FIFO selected (1 of 2)

No.	Step Name	Description
1	Start initialization	—
2	Set CCR0	Set the CCR0.TEIE, TIE, RIE, TE, and RE bits to 0*1. If you have not changed from the initial settings, you can skip this step.
3	Set FCR	Set the TFRST and RFRST to 1 to empty FIFO. Set the TTRG[4:0], RTRG[4:0], and RSTRG[4:0] bits.

Table 38.38 Example flow of SCI initialization in clock synchronous mode with non-FIFO selected (2 of 2)

No.	Step Name	Description
4	Set CCR3 except MOD[2:0]	Set CCR3 except of communication mode. ● FIFO use/no-use ● Transmission/reception format ● Clock setting ● Leave unused bits (CHR[1:0], STP, RXDSEL, MP, DE, ACS0, GM, BLK) at their initial values.
5	Set CCR3.MOD[2:0]	Set communication mode (MOD[2:0] = 010b)*2.
6	Set CCR2	Select clock, set bit rate *3. Leave unused bits (BCP[2:0], ABCS, ABCSE, BRME, MDDR[7:0]) at their initial values.
7	Set CCR1	Set up the loop-back function, communication pin status and the CTS/RTS function.
8	Set CCR4	Set up the adjust sampling timing function. Leave unused bits (CMPD[8:0]) at their initial values.
9	Set the I/O port functions	Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.
10	Set CFCLR, FFCLR	Write 1 to the following bits and clear the corresponding flag. CFCLR.RDRFC, FERC, PERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC FFCLR.BRKC, DRC
11	Set CCR0	Set the TE or RE bit to 1.*1 *4 To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time. Setting the TE and RE bits allows TXDn and RXDn to be used.
12	Initialization completion	—

Note 1. In simultaneous transmit and receive operations, the TE and RE bits in CCR0 should both be to 0 or set to 1 simultaneously.
Note 2. Set CPOL and CPHA before setting the communication mode.
Note 3. If you use an external clock, you do not need to set it.
Note 4. When using the internal clock (master), the setting of reception only is prohibited.

38.6.4 Serial Data Transmission in Clock Synchronous Mode

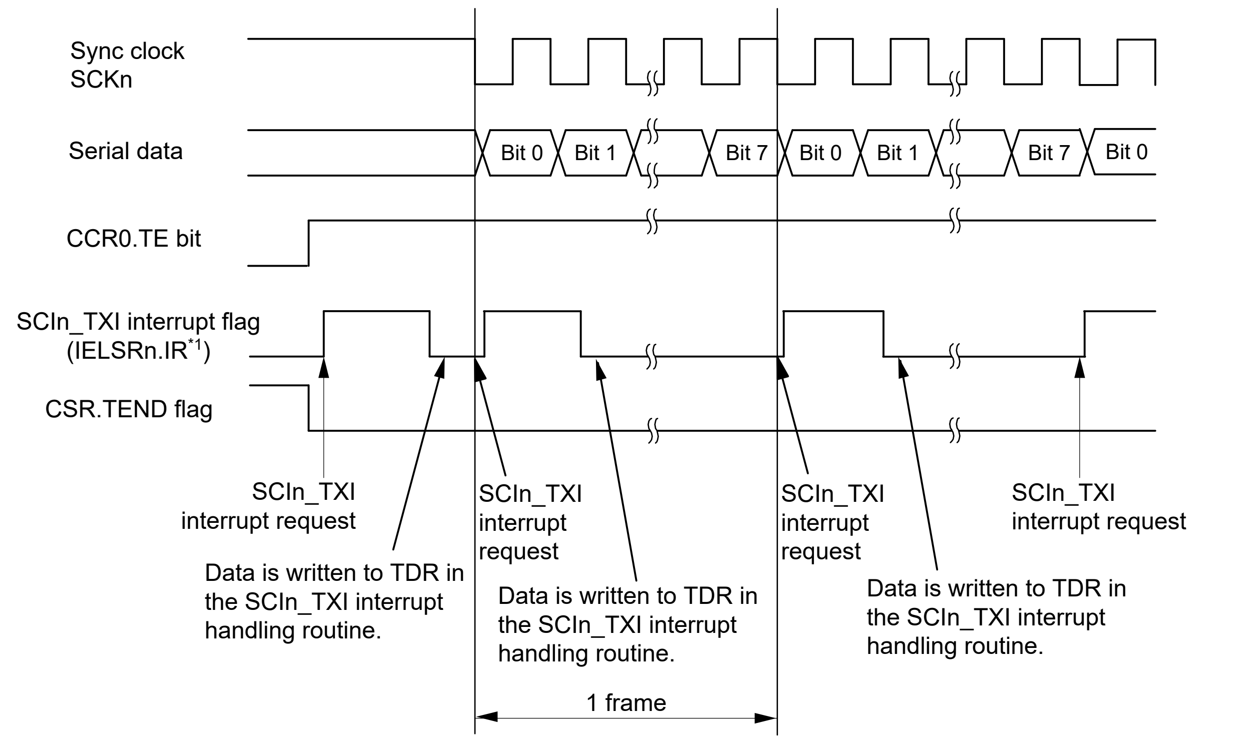
(1) Non-FIFO selected

Figure 38.62, Figure 38.63, and Figure 38.64 show examples of serial transmission in clock synchronous mode.

In serial data transmission, the SCI operates as follows:

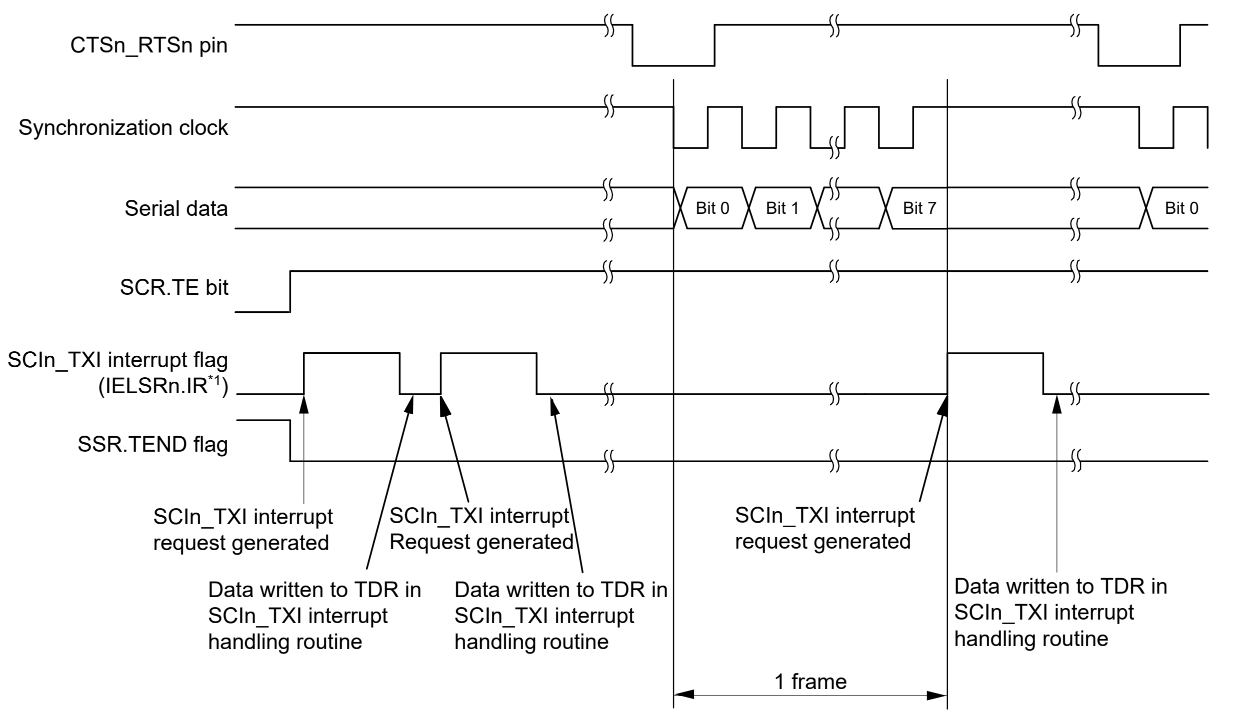
1. The SCI transfers data from the TDR register to the TSR register when data is written to TDR in the SCIn_TXI interrupt handling routine. When starting data transmission, set the CCR0.TIE bit and the CCR0.TE bit to 1 simultaneously by a single instruction. Then a SCIn_TXI interrupt request is generated.
2. After transferring data from TDR to TSR, the SCI starts transmission. When the CCR0.TIE bit is set to 1, an SCIn_TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to TDR in the SCIn_TXI interrupt handling routine before transmission of the current transmit data finishes. When SCIn_TEI interrupt requests are in use, set the CCR0.TIE bit to 0 and the CCR0.TEIE bit to 1 after the last of the data to be transmitted is written to the TDR register from the handling routine for SCIn_TXI requests.
3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when the clock output mode is specified and in synchronization with the input clock when the use of an external clock is specified. Output of the clock signal is suspended until the input CTS signal is low when the CCR1.CTSE bit is 1 (CTS function enabled).
4. The SCI checks for update to the TDR register on output of the last bit.
5. When the TDR register is updated, the next transmit data is transferred from TDR to TSR, and serial transmission of the next frame starts.
6. If TDR is not updated, the CSR.TEND flag is set to 1. The TXDn pin retains the output state of the last bit. If the CCR0.TEIE bit is 1, an SCIn_TEI interrupt request is generated and the SCKn pin is held high.

Figure 38.62, Figure 38.63, and Figure 38.64 show examples of serial data transmission.



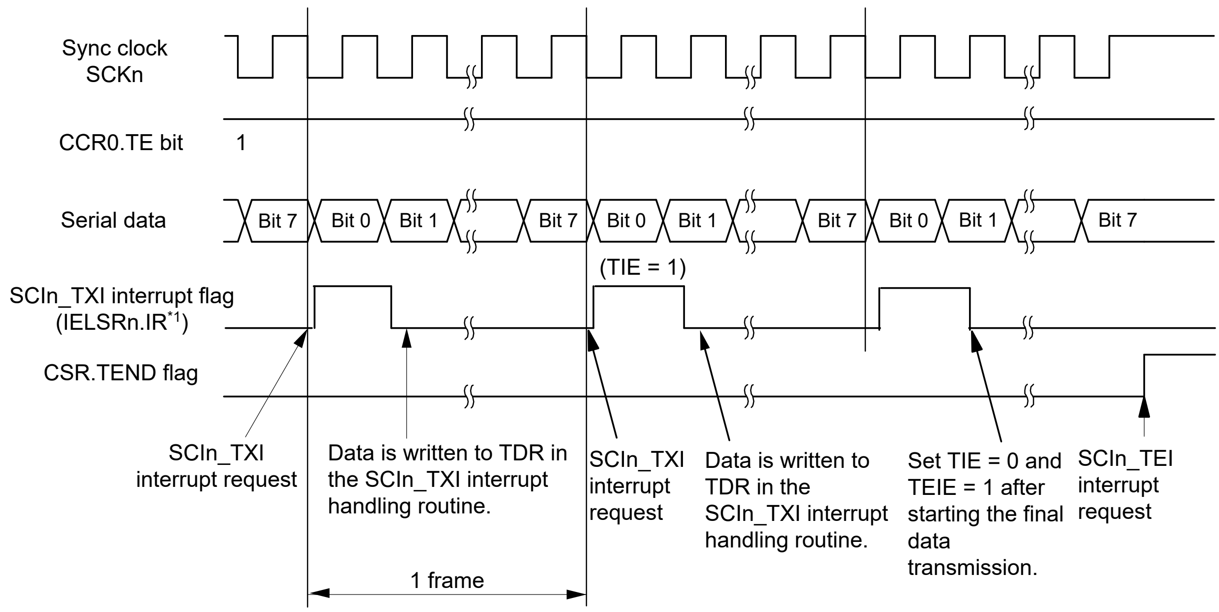
Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 38.62 Example of serial data transmission in clock synchronous mode when the CTS function is not used at the beginning of transmission



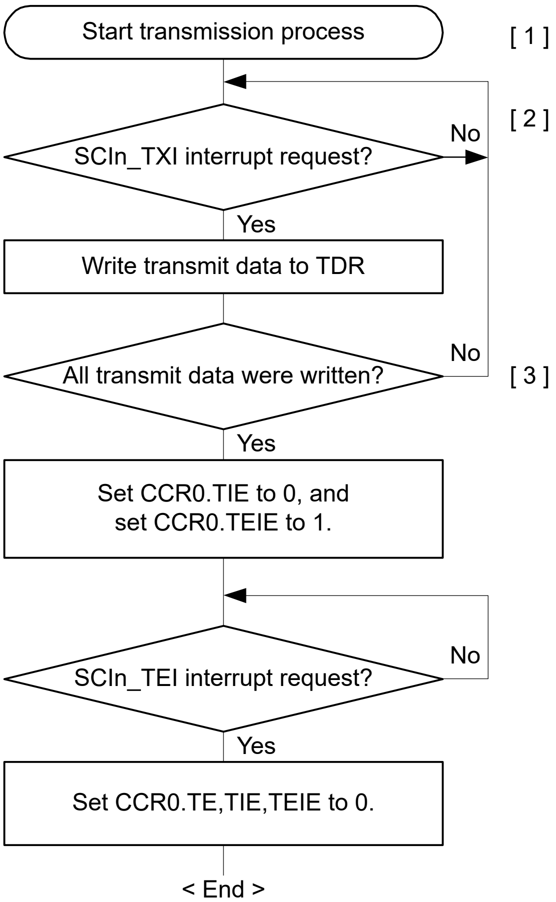
Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 38.63 Example of serial data transmission in clock synchronous mode when the CTS function is used at the beginning of transmission



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 38.64 Example of serial data transmission in clock synchronous mode from the middle of transmission until transmission completion



[1] Start transmission process

[2] Write transmit data to TDR by using SCIn_TXI:
The transmit-data-empty-interrupt(SCIn_TXI) is Generated when transmission-data forwards to TSR from TDR. Write transmit data to TDR once by using a SCIn_TXI.

[3] Serial transmission continuation procedure:
To continue serial transmission write transmit data to TDR by using a SCIn_TXI.
Transmit data can also be written by activating the DMAC or DTC.
If you use SCIn_TEI, set CCR0.TIE = 0 and CCR0.TEIE = 1 after writing the last-transmit-data to TDR.

Note: When the external clock is in use (the value of the CCR3.CKE[1:0] is 10b or 11b), the rising edge on the SCK pin for the last bit sets the CSR.TEND flag to 1. Setting TE to 0 immediately after this, then it may lead to insufficient received data hold time on the receiver side.

Figure 38.65 Example flow of serial transmission in clock synchronous mode with non-FIFO selected

(2) FIFO selected

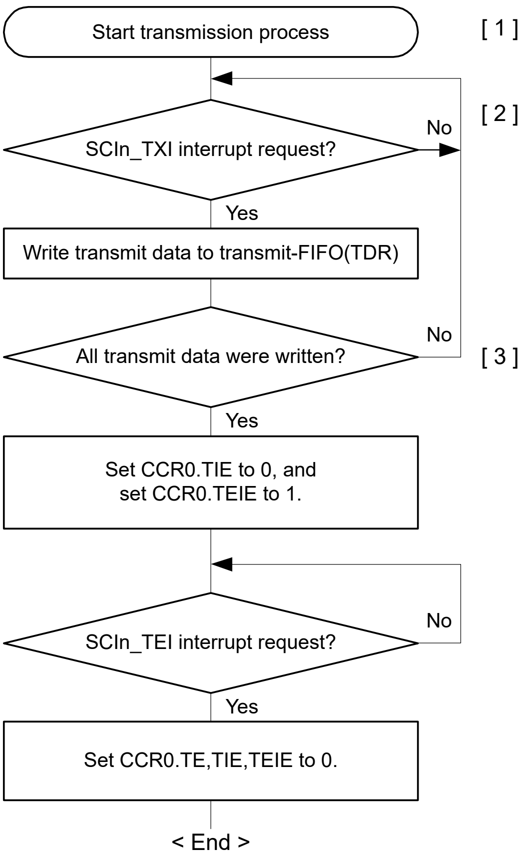
Figure 38.66 shows an example of serial transmission in clock synchronous mode with FIFO selected.

In serial data transmission, the SCI operates as follows:

1. The SCI transfers data from the transmit-FIFO (TDR register) to the TSR register when data is written to transmit-FIFO (TDR register) in the SCIn_TXI interrupt handling routine. The amount of data that can be written to transmit-FIFO (TDR register) is 16 - FTSR.T[5:0] bytes. In addition, when starting data transmission, set the CCR0.TIE bit and the CCR0.TE bit to 1 simultaneously by a single instruction. Then a SCIn_TXI interrupt request is generated.
2. After transferring data from transmit-FIFO (TDR register) to TSR, the SCI starts transmission. When the amount of transmit data written in transmit-FIFO (TDR register) is equal to or less than the specified transmit triggering number, the CSR.TDRE is set to 1. When the CCR0.TIE bit is set to 1, an SCIn_TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to transmit-FIFO (TDR register) in the SCIn_TXI interrupt handling routine before transmission of the current transmit data has finished. When SCIn_TEI interrupt requests are in use, set the CCR0.TIE bit to 0 and the CCR0.TEIE bit to 1 after the last of the data to be transmitted is written to the transmit-FIFO (TDR register) from the handling routine for SCIn_TXI requests.
3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when the clock output mode is specified and in synchronization with the input clock when the use of an external clock is specified. Output of the clock signal is suspended until the input CTS signal is low when the CCR1.CTSE bit is 1 (CTS function enabled).
4. The SCI checks whether non-transmitted data remains in transmit-FIFO (TDR register)*1 on output of the stop bit.
5. When data is remaining in the transmit-FIFO (TDR register), the next transmit data is transferred from transmit-FIFO (TDR register) to TSR and serial transmission of the next frame starts.

6. If no data is remaining in the transmit FIFO (TDR register), the CSR.TEND flag is set to 1. The TXDn pin retains the output state of the last bit. If the CCR0.TEIE bit is 1, an SCIn_TEI interrupt request is generated and the SCKn pin is held high.

Note 1. The number of unsent transmit data stored in the TDR register (transmit FIFO) can be monitored by reading the FTSR.T[5:0] bits.



[1] Start transmission process

[2] Write transmit data to transmit-FIFO(TDR) by using SCIn_TXI: The transmit-data-empty-interrupt(SCIn_TXI) is generated when transmission-data forwards to TSR from the transmit-FIFO(TDR) and the number of transmit-data written in the transmit-FIFO is equal to or less than the transmit-FIFO threshold.

[3] Serial transmission continuation procedure: To continue serial transmission, write transmit data to the transmit-FIFO(TDR) by using a SCIn_TXI interrupt and clear CSR.TDRE flag to 0. Transmit data can also be written to the transmit- FIFO(TDR) by activating the DMAC or DTC. In this case, the TDRE flag is cleared automatically, so please don't operate the TDRE flag. If you use SCIn_TEI, set CCR0.TIE = 0 and CCR0.TEIE = 1 after writing the last-transmit-data to TDR.

Note: When the external clock is in use (the value of the CCR3.CKE[1:0] is 10b or 11b), the rising edge on the SCK pin for the last bit sets the CSR.TEND flag to 1. Setting CCR0.TE to 0 immediately after this, then it may lead to insufficient received data hold time on the receiver side.

Figure 38.66 Example flow of serial transmission in clock synchronous mode with FIFO selected

38.6.5 Serial Data Reception in Clock Synchronous Mode

(1) Non-FIFO selected

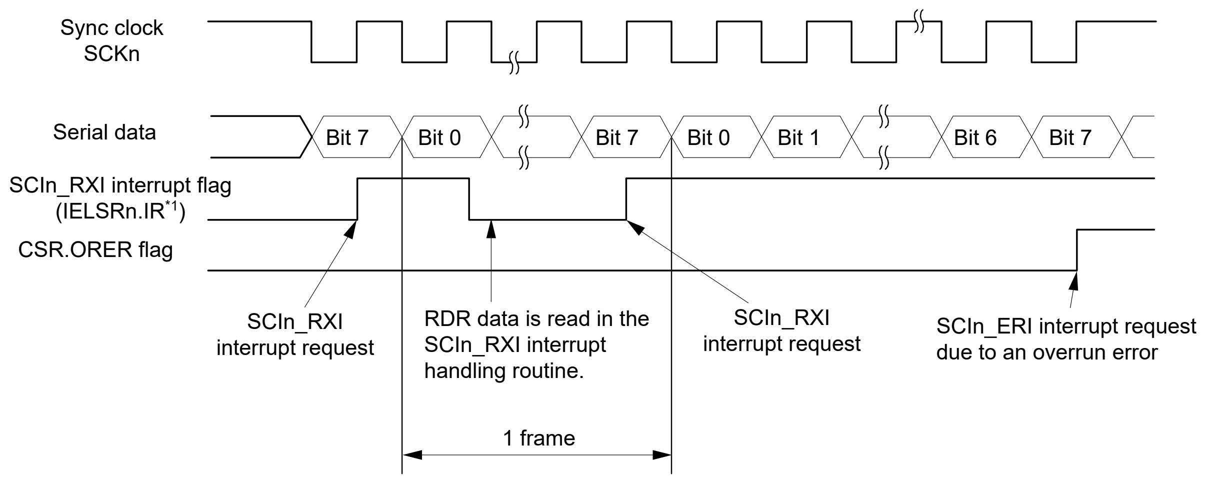
Figure 38.67 and Figure 38.68 show examples of SCI operation for serial reception in clock synchronous mode.

In serial data reception, the SCI operates as follows during serial data reception. Reception-only operation is possible only in slave mode. (In master mode, reception-only operation is prohibited.)

1. When the value of the CCR0.RE bit becomes 1, the CTSn_RTSn pin goes low (when the RTS function is used).
2. The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and stores the receive data in the RSR register.
3. If an overrun error occurs, the CSR.ORER flag is set to 1. If the CCR0.RIE bit is 1, an SCIn_ERI interrupt request is generated. Receive data is not transferred to the RDR register.
4. When reception completes successfully, receive data is transferred to the RDR register. If the CCR0.RIE bit is 1, an SCIn_RXI interrupt request is generated. Continuous reception is enabled by reading the received data transferred to the

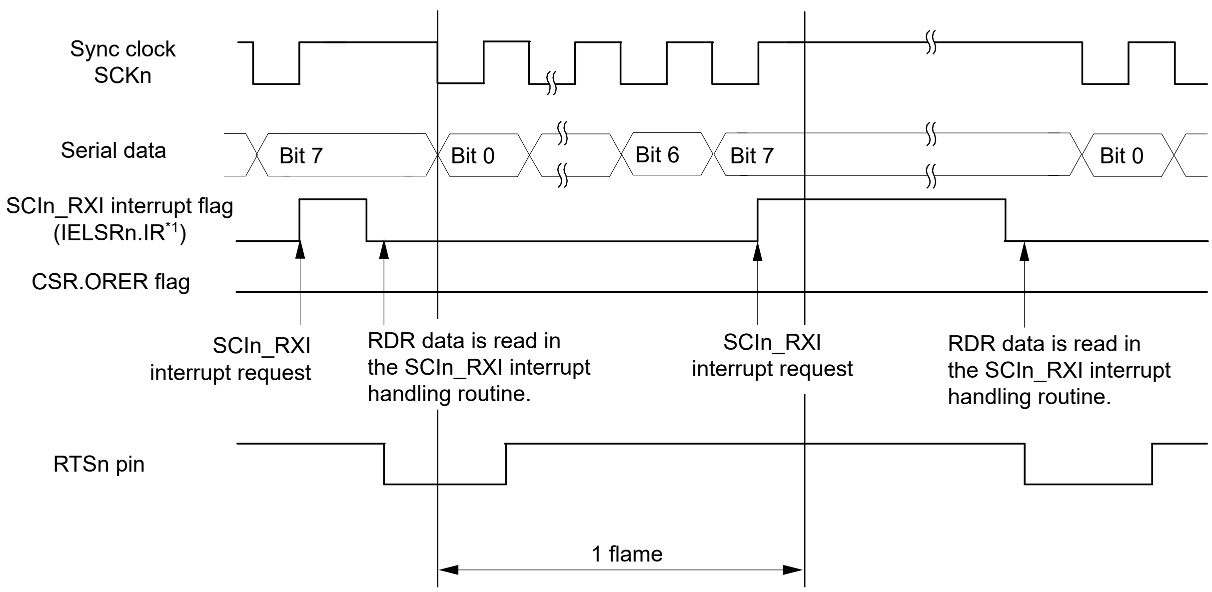
RDR register in the SCIn_RXI interrupt handling routine before reception of the next receive data completes. Reading the received data that is transferred to RDR causes the CTSn_RTSn pin to output low (when the RTS function is used).

If you want to prevent the CTSn_RTSn pin output from turning low level after the final data is received, clear the CCR0.RE bit to 0 and then read the RDR register.



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 38.67 Example operation for serial reception in clock synchronous mode (1) when the RTS function is not used



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 38.68 Example operation for serial reception in clock synchronous mode (2) when RTS function is used

Data transfer cannot resume while the receive error flag is 1. Therefore, clear the ORER, FER, and PER flags in the CSR register to 0 before resuming data reception. Additionally, always read the RDR register during overrun error processing. When a data reception is forced to terminate by a 0 write to the CCR0.RE bit during operation, read the RDR register because received data that is not yet read might be left in the RDR register.

Figure 38.69 shows an example flow of serial data reception.

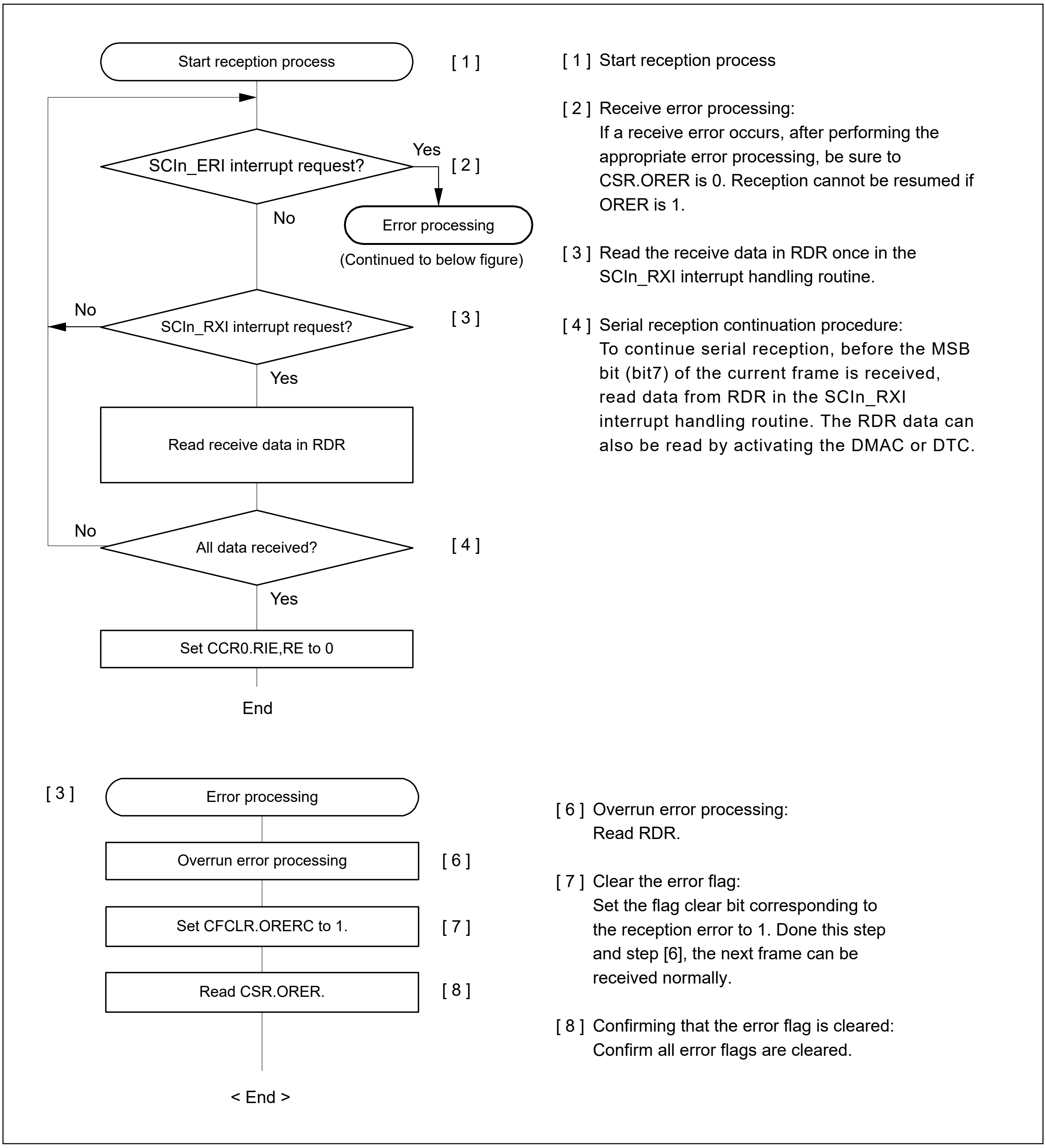


Figure 38.69 Example flow of serial reception in clock synchronous mode with non-FIFO selected

(2) FIFO selected

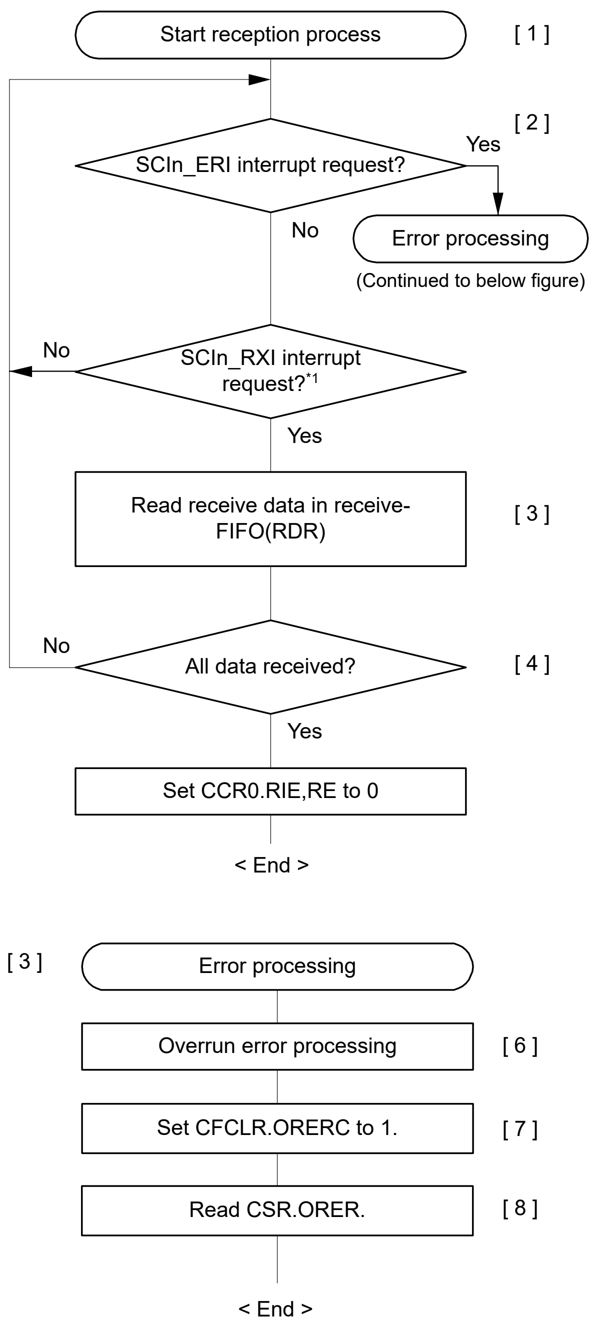
Figure 38.70 shows an example of serial reception in clock synchronous mode with FIFO selected.

In serial data reception, the SCI operates as follows during serial data reception. Reception-only operation is possible only in slave mode. (In master mode, reception-only operation is prohibited.)

1. When the value of the CCR0.RE bit becomes 1, the CTSn_RTSn pin goes low (when the RTS function is used).
2. The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and transfers the received data to the receive-FIFO (RDR register).
3. If an overrun error occurs, the CSR.ORER flag is set to 1. If the CCR0.RIE bit is 1, an SCIn_ERI interrupt request is generated. Received data is not transferred to the receive-FIFO (RDR register)*1.

4. When data reception completes successfully, the receive data is transferred to the receive-FIFO (RDR register)*1. The CSR.RDRF flag is set to 1 when the amount of the receive data stored in receive-FIFO (RDR register) is equal to or greater than the specified receive triggering number. If the CCR0.RIE bit is 1, an SCIn_RXI interrupt request is generated. Continuous data reception is enabled by reading the receive data transferred to receive-FIFO (RDR register)*1 in the SCIn_RXI interrupt handling routine before an overrun error occurs. If the amount of received data that is transferred to receive-FIFO (RDR register) is less than the specified receive triggering number, the CTSn_RTSn pin goes low (when the RTS function is used).

Note 1. In clock synchronous mode, RDR.RDAT[8] is not used.



[1] Start reception process

[2] Receive error processing:
If a receive error occurs, after performing the appropriate error processing, be sure to CSR.ORER is 0. Reception cannot be resumed if ORER is 1.

[3] Read the receive data in the receive-FIFO(RDR) in the SCIn_RXI interrupt handling routine. The reception data stored in the receive-FIFO(RDR) is read until the number of stored data will be below the FCR.RTRG value at least. You can confirm the number of the reception data in the FIFO by FRSR.R[5:0].

[4] Serial reception continuation procedure:
To continue serial reception, before the overrun error occurs, read data from RDR. The RDR data can also be read by activating the DMAC or DTC.

Note1:
Set all received data to an integer multiple of the receive FIFO threshold. If it is not an integral multiple, return to step [2] and it will become an infinite loop, so set a timeout time separately using a timer etc.

[6] Overrun error processing:
Read RDR.

[7] Clear the error flag:
Set the flag clear bit corresponding to the reception error to 1. Done this step and step [6], the next frame can be received normally.

[8] Confirming that the error flag is cleared:
Confirm all error flags are cleared.

Figure 38.70 Example flow of serial reception in clock synchronous mode with FIFO selected

38.6.6 Simultaneous Serial Data Transmission and Reception in Clock Synchronous Mode

(1) Non-FIFO selected

Figure 38.71 shows an example flow of simultaneous serial transmit and receive operations in clock synchronous mode. After initializing the SCI, use the following procedure for simultaneous serial data transmit and receive operations.

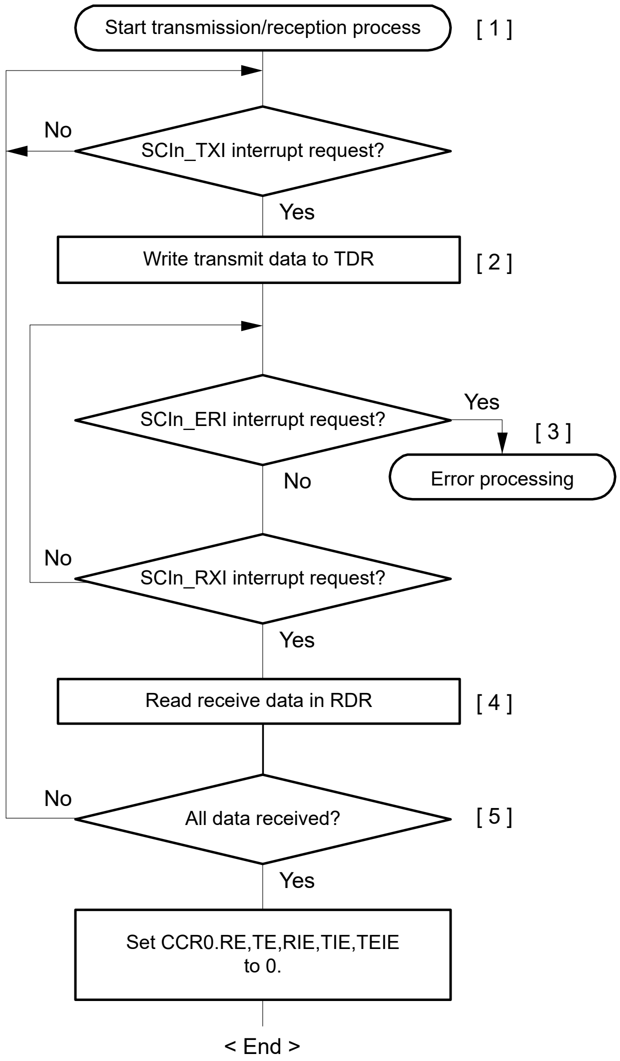
To switch from transmit mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the data transmission by verifying that the CSR.TEND flag is set to 1.
2. Initialize the CCR0 register, and then set the TIE, RIE, TE, and RE bits in the CCR0 register to 1 simultaneously by a single instruction.

To switch from receive mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the data reception.
2. Set the CCR0.TE and RE bits to 0, and then check that the receive error flag (ORER, FER, and PER) in the CSR register is 0.
3. Set the TIE, RIE, TE, and RE bits in the CCR0 register to 1 simultaneously by a single instruction.

When the RTS function is used in the concurrent transmission/reception operation, if you want to prevent the CTSn_RTSn pin output from turning to low after the final data is received as in the reception operation, clear the RE and TE bits in CCR0 to 0 simultaneously, and then read the RDR register.



[1] Start transmission/reception process

[2] Transmit data write:
Write transmit data to TDR in the SCIn_TXI Interrupt request handling routine.

[3] Receive error processing:
If a receive error occurs, after performing the appropriate error processing, be sure to CSR.ORER is 0. Reception cannot be resumed if ORER is 1.

[4] Reading receive data:
The reception data stored in RDR once in the SCIn_RXI interrupt request handling routine.

[5] Serial transmission/reception continuation procedure
To continue serial reception, before the MSB bit (bit7) of the current frame is received, read data from RDR in the SCIn_RXI interrupt handling routine. And before the MSB(bit7) of the current frame is transmitted, write transmit data to TDR in the SCIn_TXI interrupt handling. Transmit data can also be written by activating the DMAC or DTC. And the RDR data can also be read by activating the DMAC or DTC.

Figure 38.71 Example flow of simultaneous serial transmission and reception in clock synchronous mode with non-FIFO selected

(2) FIFO selected

Figure 38.72 shows an example flow of simultaneous serial transmit and receive operations in clock synchronous mode with FIFO selected.

After initializing the SCI, use the following procedure for simultaneous serial data transmit and receive operations.

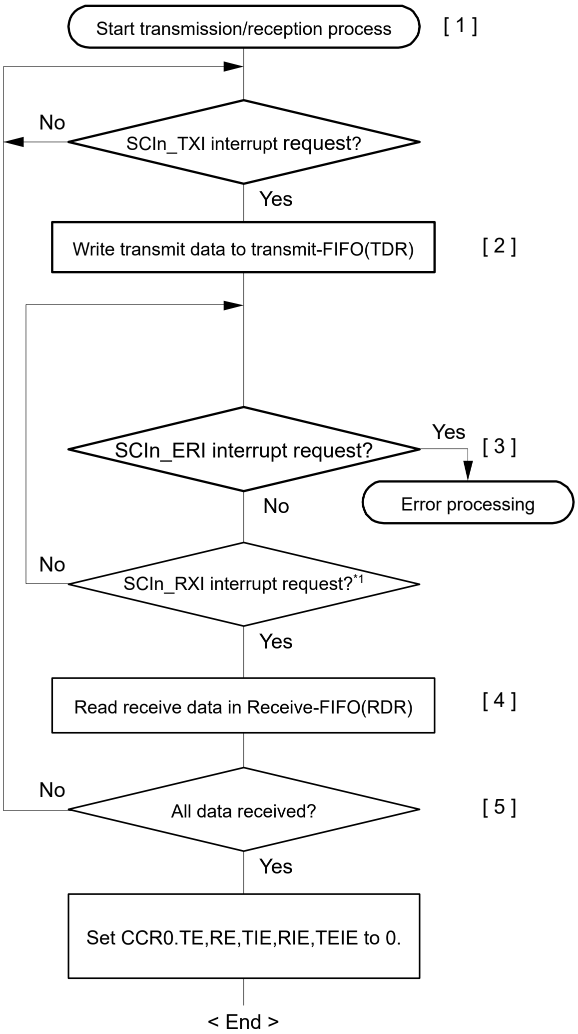
To switch from transmit mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the transmission by verifying that the CSR.TEND flag is set to 1.
2. Initialize the CCR0 register, then set the TIE, RIE, TE, and RE bits in the CCR0 register to 1 simultaneously by a single instruction.

To switch from receive mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the reception.
2. Set the CCR0.TE and RE bits to 0.
3. Check that the receive error flags (ORER, FER, and PER) in the CSR register are 0, and then set the TIE, RIE, TE, and RE bits in the CCR0 register to 1 simultaneously by a single instruction.

Since clock synchronous communication performs transmission and reception at the same time, make sure that the number of data to be transmitted and received is the same.



[1] Start transmission / reception process

[2] Transmit data write:
Write transmit data to Transmit-FIFO(TDR) in the SCIn_TXI Interrupt request handling routine.
Match the number of transmitted data with the FCR.RTRG.

[3] Receive error processing:
If a receive error occurs, after performing the appropriate error processing (Initialization such as stopping communication, reading all received data, clearing CSR.RDRF, etc.), be sure to ORER is 0. Reception cannot be resumed if ORER is 1.

[4] Read receive data:
Read the received data stored in receive-FIFO(RDR) for the number of the FCR.RTRG in the SCIn_RXI interrupt handling routine.

[5] Serial transmission/reception continuation procedure
To continue serial reception, before the overrun error occurs, read data from RDR and clear CSR.RDRF.
To continue serial transmission, write transmit data to the transmit-FIFO(TDR) by using a SCIn_TXI interrupt and clear TDRE flag to 0.
Transmit data can also be written to the transmit- FIFO(TDR) by activating the DMAC or DTC. And The RDR data can also be read by activating the DMAC or DTC. In this case, the TDRE and RDRF are cleared automatically, so please don't operate the TDRE And RDRF.

Note 1. Set all received data to an integer multiple of the receive FIFO threshold. If it is not an integral multiple, return to step [2] and it will become an infinite loop, so set a timeout time separately using a timer etc.

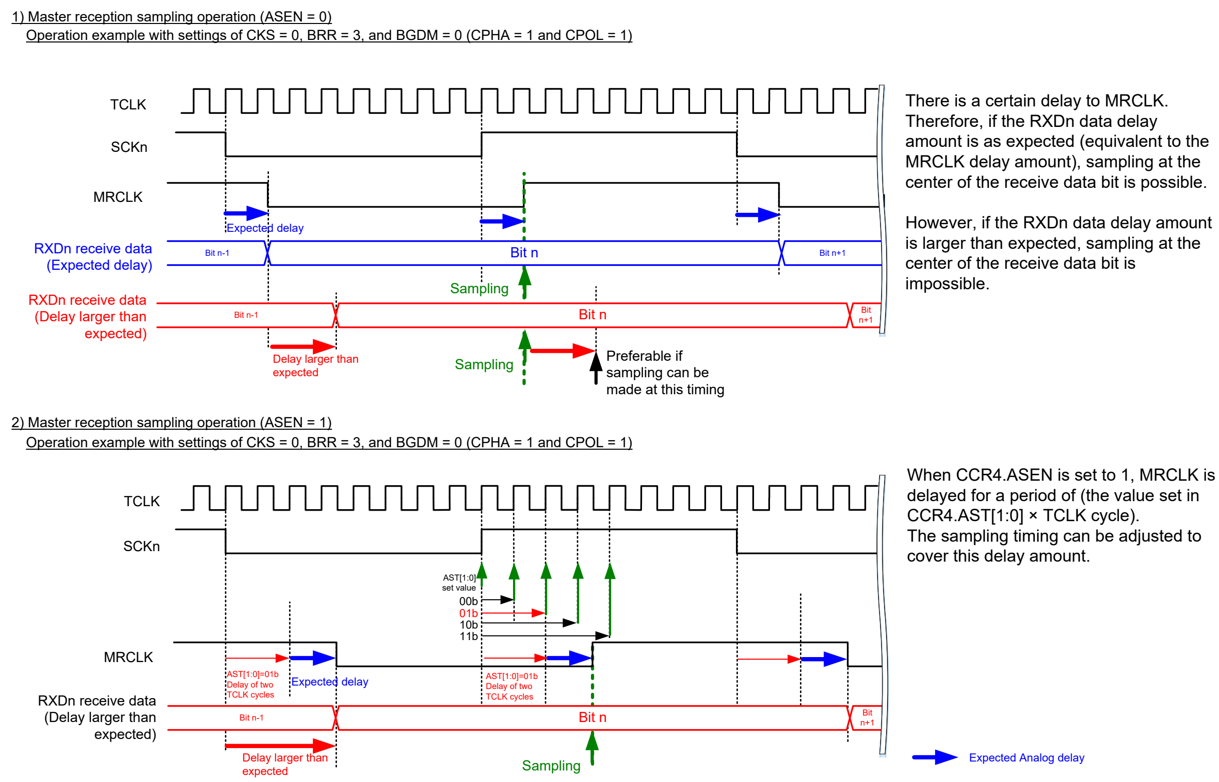
Figure 38.72 Example flow of simultaneous serial transmission and reception in clock synchronous mode with FIFO selected

38.6.7 Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with Internal Clock Used

When the clock synchronous internal clock is used (master mode), MRCLK is used as a reception sampling clock.

This function adjusts the reception sampling timing by delaying MRCLK by 1 to 4 TCLK and adding a digital delay. MRCLK's analog delay cannot be adjusted by this function.

Setting the CCR4.ASEN bit to 1 enables this function. The delay value is set in CCR4.AST[1:0].



Note: This timing chart illustrates the reception sampling timing adjustment function.

Figure 38.73 Reception Sampling Timing Adjustment Operation in Clock Synchronous Mode (Master)

38.7 Operation in Smart Card Interface Mode

The SCI supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards), as an extended function of the SCI.

Smart card interface mode can be selected using the appropriate register.

38.7.1 Example Connection

Figure 38.74 shows an example connection between a smart card (IC card) and the MCU. As shown in Figure 38.74, because the MCU communicates with an IC card using a single transmission line, interconnect the TXDn and RXDn pins and pull up the data transmission line to VCC using a resistor.

Setting the CCR0.TE and CCR0.RE bits to 1 with an IC card disconnected enables closed-loop transmission or reception, allowing self-diagnosis. To supply an IC card with the clock pulses generated by the SCI, input the SCKn pin output to the CLK pin of an IC card.

An output port of the MCU can be used to output a reset signal.

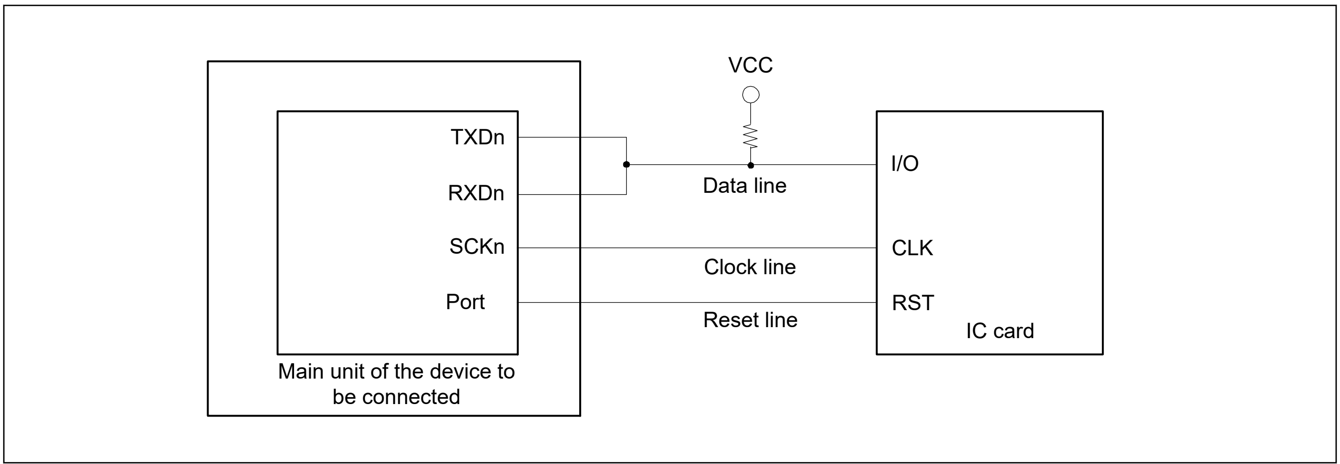


Figure 38.74 Example connection with a smart card (IC card)

38.7.2 Data Format (Except in Block Transfer Mode)

Figure 38.75 shows the data transfer formats in smart card interface mode:

- One frame consists of 8-bit data and a parity bit in Asynchronous mode.
- During transmission, at least 2 etu (elementary time unit – the time required for transferring 1 bit) is set as a guard time from the end of the parity bit until the start of the next frame.
- If a parity error is detected during reception, a low error signal is output for 1 etu after 10.5 etu elapse from the start bit.
- If an error signal is sampled during transmission, the same data is automatically retransmitted after at least 2 etu.

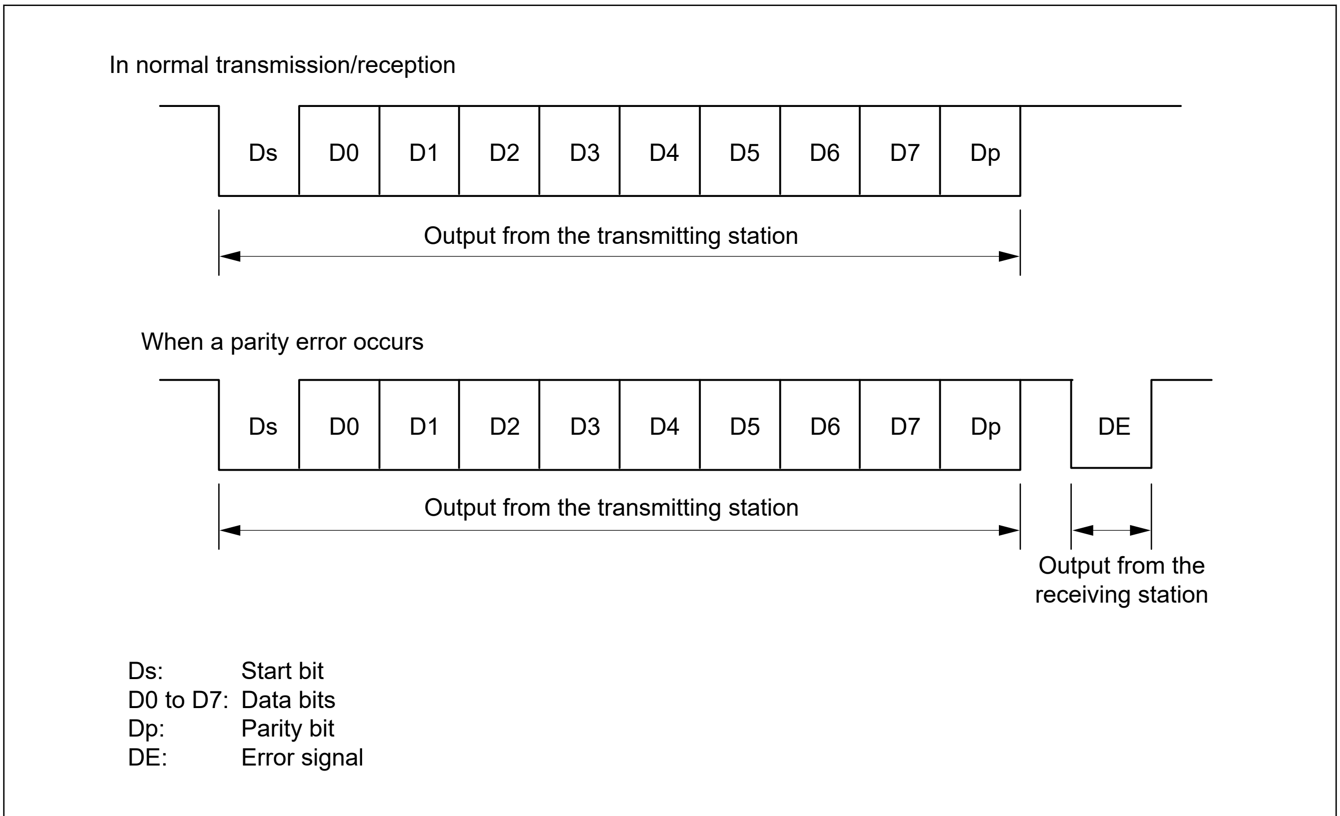


Figure 38.75 Data formats in smart card interface mode

For communications with IC cards of the direct convention type and inverse convention type, follow the procedures in this section.

(1) Direct Convention Type

For the direct convention type, logic levels 1 and 0 indicate the Z and A states, respectively, and data is transferred with LSB-first for the start character, as shown in Figure 38.76. Therefore, data in the start character in the figure is 0x3B.

When using the direct convention type, write 1 to the CCR3.LSBF and write 0 to the CCR3.SINV. Write 0 to the CCR1.PM bit to use even parity, which is prescribed by the smart card standard.

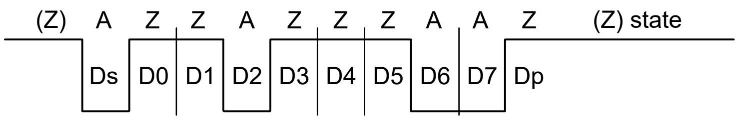


Figure 38.76 Direct convention with LSBF in CCR3 = 1, SINV in CCR3 = 0, and PM in CCR1 = 0

(2) Inverse Convention Type

For the inverse convention type, logic levels 1 and 0 indicate the A and Z states, respectively, and data is transferred with MSB-first for the start character, as shown in Figure 38.77. Therefore, data in the start character in the figure is 0x3F.

When using the inverse convention type, write 0 to the CCR3.LSBF and write 1 to the CCR3.SINV. The parity bit is logic level 0 to produce even parity, which is prescribed by the smart card standard, and corresponds to the Z state. Because the SINV bit of the MCU only inverts data bits D7 to D0, write 1 to the PM bit in CCR1 to invert the parity bit for both transmission and reception.

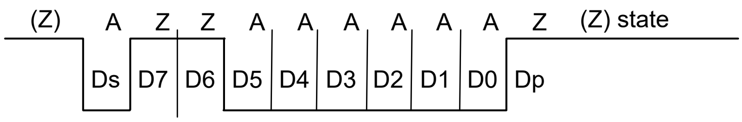


Figure 38.77 Inverse convention with LSBF in CCR3 = 0, SINV in CCR3 = 1, and PM in CCR1 = 1

38.7.3 Block Transfer Mode

Block transfer mode differs from normal smart card interface mode as follows:

- Even if a parity error is detected during reception, no error signal is output. Because the PER flag in CSR is set by error detection, clear the PER flag before receiving the parity bit of the next frame.
- During transmission, at least 1 etu is set as a guard time from the end of the parity bit until the start of the next frame
- Because the same data is not retransmitted, the TEND flag in CSR is set to 11.5 etu after transmission starts
- In block transfer mode, the ERS flag in CSR indicates the error signal status as in normal smart card interface mode, but the flag is read as 0 because no error signal is transferred

38.7.4 Receive Data Sampling Timing and Reception Margin

Only the internal clock generated by the on-chip baud rate generator can be used as a transfer clock in smart card interface mode.

In this mode, the SCI can operate on a base clock with a frequency of 32, 64, 372, 256, 93, 128, 186, or 512 times the bit rate set up in the CCR2.BCP[2:0] bits. The frequency is always 16 times the bit rate in normal Asynchronous mode.

For data reception, the falling edge of the start bit is sampled with the base clock to perform internal synchronization.

Receive data is sampled on the 16th, 32nd, 186th, 128th, 46th, 64th, 93rd, and 256th rising edges of the base clock so that it can be latched at the middle of each bit as shown in Figure 38.78. The reception margin is determined by the following formula:

$$M = \left|\left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F)\right| \times 100\ [\%]$$

M: Reception margin (%)

N: Ratio of bit rate to clock (N = 32, 64, 372, 256)

D: Duty cycle of clock (D = 0 to 1.0)

L: Frame length (L = 10)

F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5, and N = 372 in the specified formula, the reception margin is determined using the following formula:

$$M = \{0.5 - 1/(2 \times 372)\} \times 100\ [\%] = 49.866\ \%$$

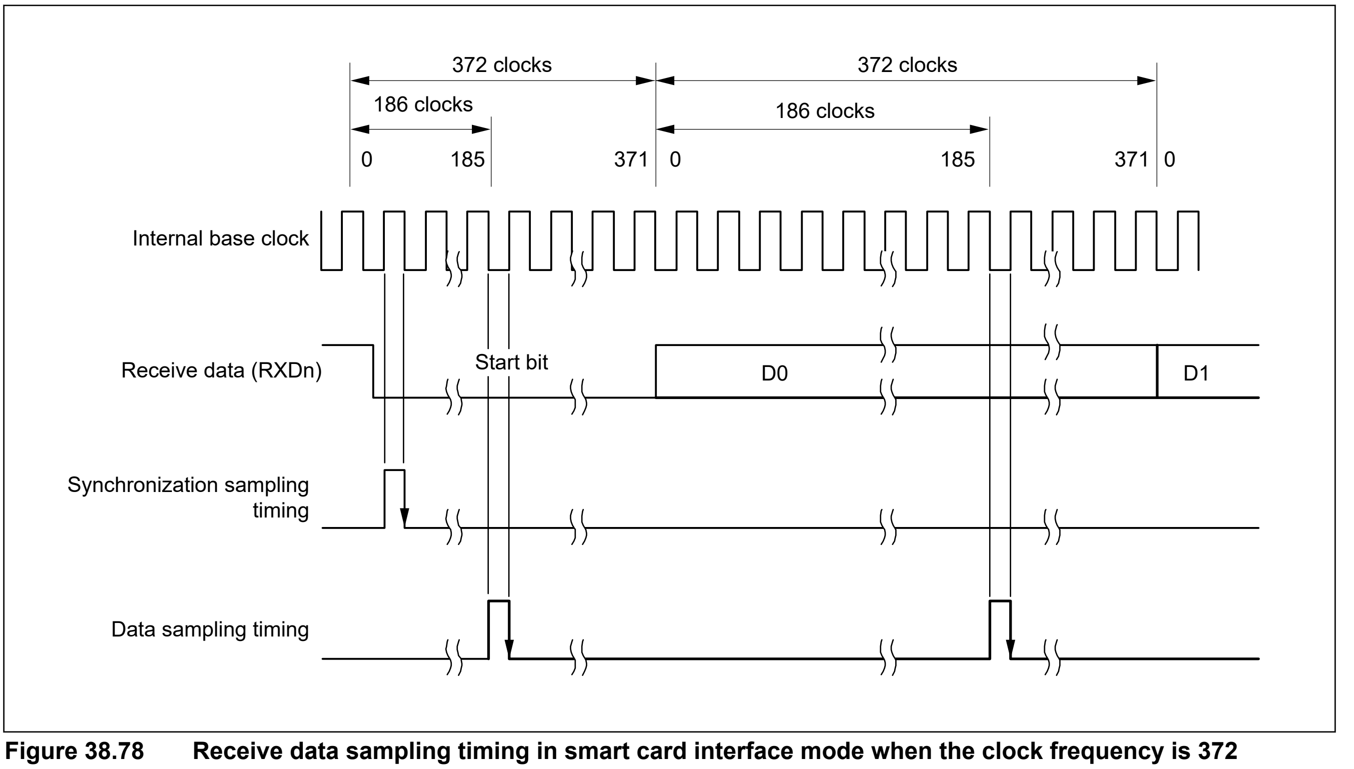


Figure 38.78 Receive data sampling timing in smart card interface mode when the clock frequency is 372 times the bit rate

38.7.5 SCI Initialization (Smart Card Interface Mode)

Before transmitting and receiving data, write the initial value 0x00 in the CCR0 register and initialize the SCI following the example flow shown in Table 38.39.

Always set the initial value in the TIE, RIE, TE, RE, TEIE bits in the CCR0 register before switching from transmission to reception mode or from reception to transmission mode. When CCR0.RE is set to 0, the RDR register is not initialized.

In transmission mode, set 1 to the CCR0.TE bit and CCR0.TIE bit simultaneously, then the SCIn_TXI interrupt request is generated.

To change from reception mode to transmission mode, first check that reception has completed, then initialize the SCI. At the end of initialization, set CCR0.TE = 1 and CCR0.RE = 0. Reception completion can be verified by reading the SCIn_RXI request, ORER, or PER flag in CSR.

To change transmission mode to reception mode, first check that transmission has completed, then initialize the SCI. At the end of initialization, set CCR0.TE = 0 and CCR0.RE = 1. Transmission completion can be verified by reading the TEND flag in CSR.

Table 38.39 Example flow of SCI initialization in smart card interface mode

No.	Step Name	Description
1	Start initialization	—
2	Set CCR0	Set CCR0.TEIE, TIE, RIE, TE, RE to 0. If you have not changed from the initial settings, you can skip this step.
3	Set CCR3	Set communication mode (MOD[2:0] = 001b), BLK, GM, and SINV. Leave other bits at their initial values.
4	Set CCR2	Set clock-select and bit-rate. Set BRME to 0.
5	Set CCR1	Set up the Noise filter function, communication terminal status. Set NFEN, PE, CTSE to 0 and set PE to 1.
6	Set the I/O port functions	Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.
7	Set CCR3	Set CKE[1:0]. When the CKE[0] bit is set to "1" due on GM setting value, the clock is output from the SCKn pin.
8	Set CFCLR	Write 1 to the following bits and clear the corresponding flag. CFCLR.RDRFC, FERC, PERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
9	Set CCR0	Set the TE or RE to 1. And set the TIE and RIE. Do not simultaneously set the TE and RE bits to 1 if self-diagnosis in not used.
10	Initialization completed	—

Figure 38.79 is a timing chart when data transmission is performed by making transition to the Smart Card Interface mode according to the above flow chart. The figure shows the case when CCR3.GM bit is 0. As shown in the figure, when the pin function is set to the SCKn pin, the SCKn pin is high impedance because the CCR3.CKE [0] bit is 0. When the TXDn pin is set, the TXDn pin is high impedance because the CCR0.TE bit is 0. Start clock output to the SCKn pin with the clock output setting CCR3.CKE [0] to 1, start data transmission by writing transmit data after setting CCR0.TE to 1.

In the smart card interface mode, even if not communicating at CCR0.TE = 0 and CCR0.RE = 0, the clock is continuously output if the clock output setting is used.

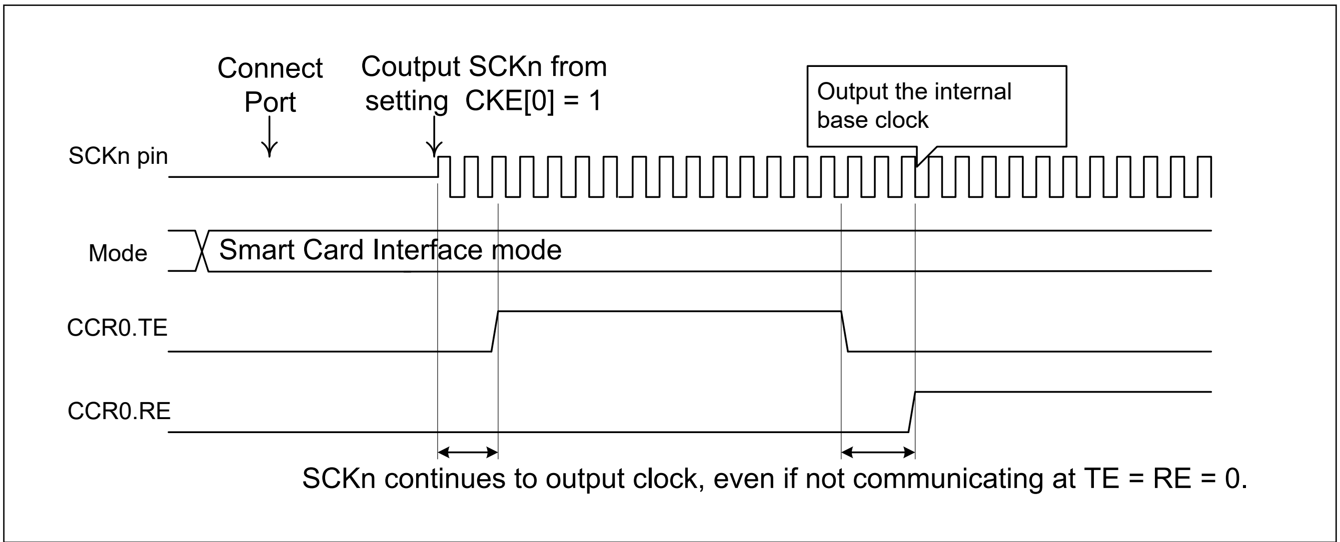


Figure 38.79 Example of Timing chart of data transmission in Smart Card Interface Mode

38.7.6 Serial Data Transmission (Except in Block Transfer Mode)

Serial data transmission in smart card interface mode (except in block transfer mode) is different from that in non-smart card interface mode, in that an error signal is sampled and data can be re-transmitted in smart card mode. Figure 38.80 shows the data re-transfer operation during transmission.

1. When an error signal from the receiver end is sampled after 1-frame data is transmitted, the CSR.ERS flag is set to 1. If the CCR0.RIE bit is 1, an SCIn_ERI interrupt request is generated. Clear the ERS flag to 0 before the next parity bit is sampled.

2. For a frame in which an error signal is received, the CSR.TEND flag is not set. Data is re-transferred from TDR to TSR, allowing automatic data retransmission.
3. If no error signal is returned from the receiver, the ERS flag is not set to 1.
4. In this case, the SCI determines that transmission of 1-frame data, including the re-transfer, is complete, and the TEND flag is set. If the CCR0.TIE bit is 1, an SCIn_TXI interrupt request is generated. Write transmit data to the TDR to start transmission of the next data.

Figure 38.82 shows an example flow of serial transmission. All the processing steps are automatically performed using an SCIn_TXI interrupt request to activate the DTC or DMAC.

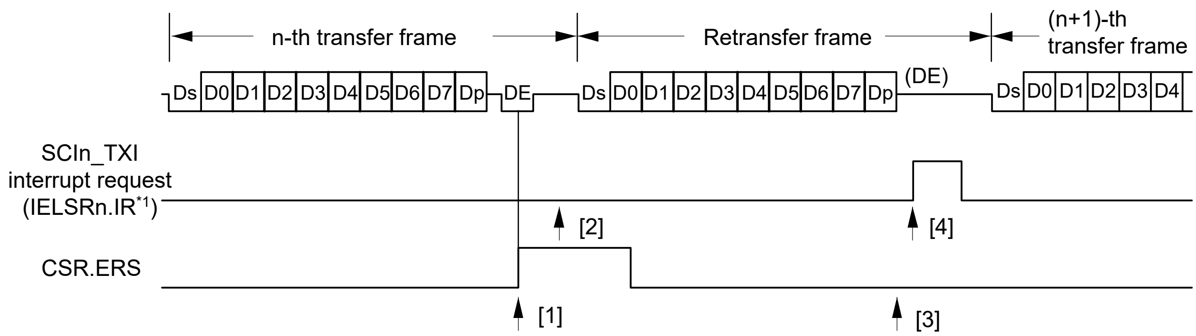
When the CSR.TEND flag is set to 1 in transmission and when the CCR0.TIE bit is 1, an SCIn_TXI interrupt request is generated.

The DTC or DMAC is activated by an SCIn_TXI interrupt request if the SCIn_TXI interrupt request is previously specified as a source of DTC or DMAC activation, allowing the transfer of transmit data. The TEND flag is automatically set to 0 when the DTC or DMAC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During this retransmission, the TEND flag is kept at 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC automatically transmit the specified number of bytes, including retransmission when an error occurs. Because the ERS flag is not automatically cleared, set the RIE bit to 1 before enabling an SCIn_ERI interrupt request to be generated if an error occurs, and clear the ERS flag to 0.

When transmitting or receiving data using the DTC or DMAC, always enable the DTC or DMAC before making the SCI settings.

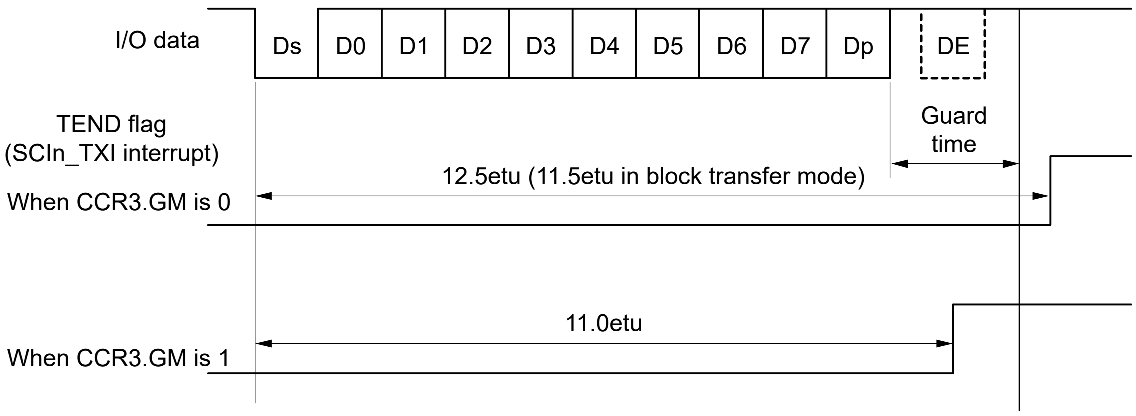
For DTC or DMAC settings, see section 18, Data Transfer Controller (DTC), section 17, DMA Controller (DMAC).



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 38.80 Data re-transfer operation in smart card interface transmission mode

The CSR.TEND flag is set at different timings depending on the CCR3.GM bit setting. Figure 38.81 shows the TEND flag generation timing.



Ds : Start bit
D0 to D7 : Data bit
Dp : Parity bit
DE : Error signal

Figure 38.81 CSR.TEND flag generation timing during transmission

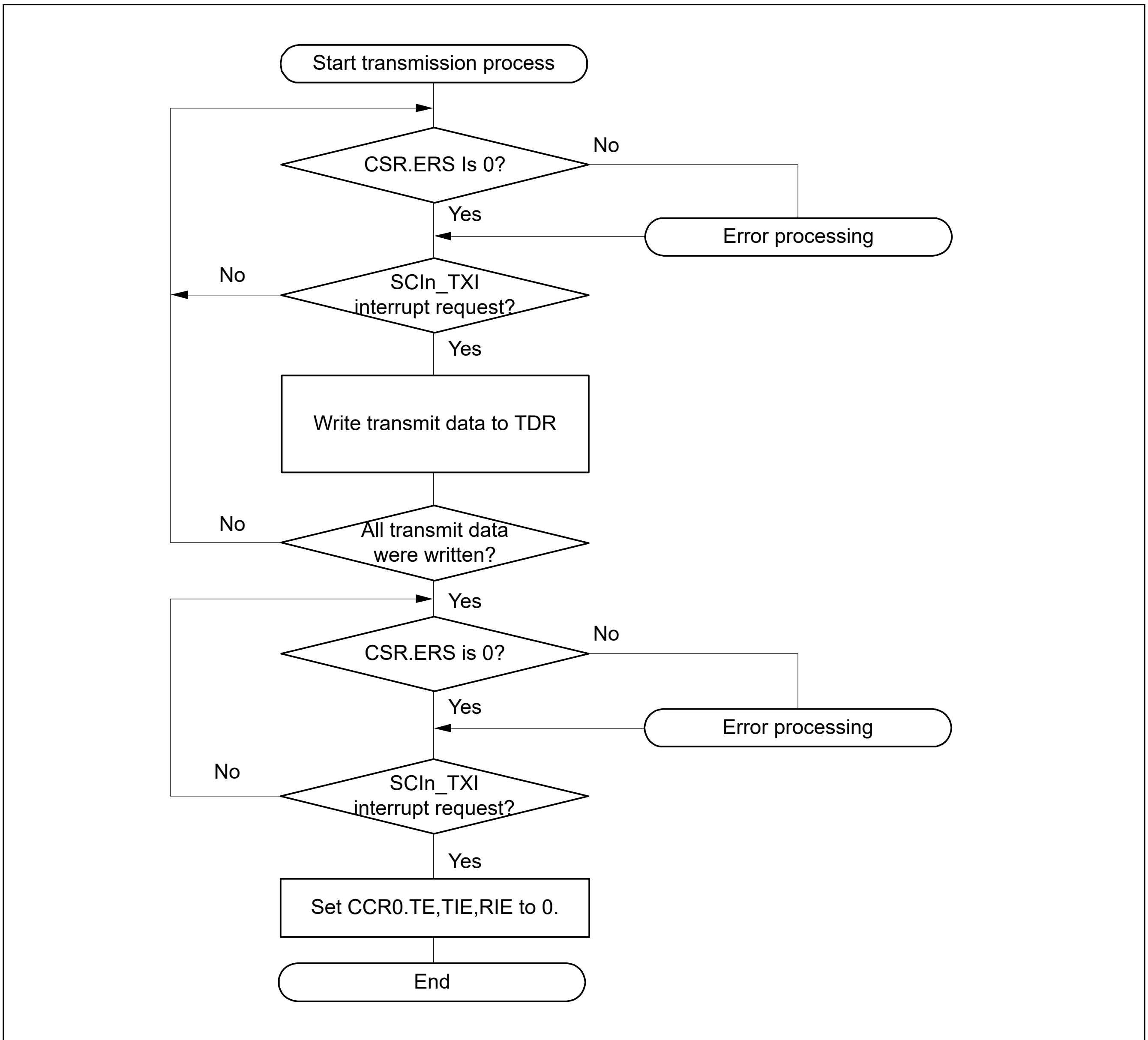


Figure 38.82 Example flow of smart card interface transmission

38.7.7 Serial Data Reception (Except in Block Transfer Mode)

Serial data reception in smart card interface mode is similar to that in non-smart card interface mode. Figure 38.83 shows the data re-transfer operation in reception mode.

1. If a parity error is detected in the receive data, the CSR.PER flag is set to 1. When the CCR0.RIE bit is 1, an SCIn_ERI interrupt request is generated. Clear the PER flag to 0 before the next parity bit is sampled.
2. For a frame in which a parity error is detected, no SCIn_RXI interrupt is generated.
3. When no parity error is detected, the CSR.PER flag is not set to 1.
4. In this case, data is determined to be received successfully. When the CCR0.RIE bit is 1, an SCIn_RXI interrupt request is generated.

Figure 38.84 shows an example flow of serial data reception. All the processing steps are automatically performed using an SCIn_RXI interrupt request to activate the DTC or DMAC.

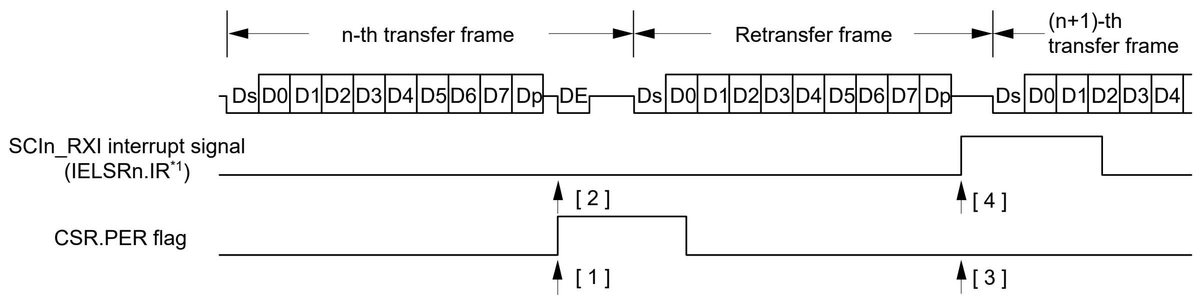
In reception, setting the RIE bit to 1 allows an SCIn_RXI interrupt request to be generated. The DTC or DMAC is activated by an SCIn_RXI interrupt request if the SCIn_RXI interrupt request is previously specified as a source of DTC or DMAC activation, allowing the transfer of receive data.

If an error occurs during reception and either the ORER or PER flag in CSR is set to 1, a receive error interrupt (SCIn_ERI) request is generated. Clear the error flag after the error occurrence. If an error occurs, the DTC or DMAC is not activated and receive data is skipped. Therefore, the number of bytes of receive data specified in the DTC or DMAC is transferred.

If a parity error occurs and the PER flag is set to 1 during reception, the receive data is transferred to RDR, allowing the data to be read.

When a reception is forced to terminate by setting CCR0.RE to 0 during operation, read the RDR register because the received data that is not yet read might be left in the RDR.

Note: For operations in block transfer mode, see section 38.3.9. Serial Data Reception in Asynchronous Mode.



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 38.83 Data re-transfer operation in smart card interface reception mode

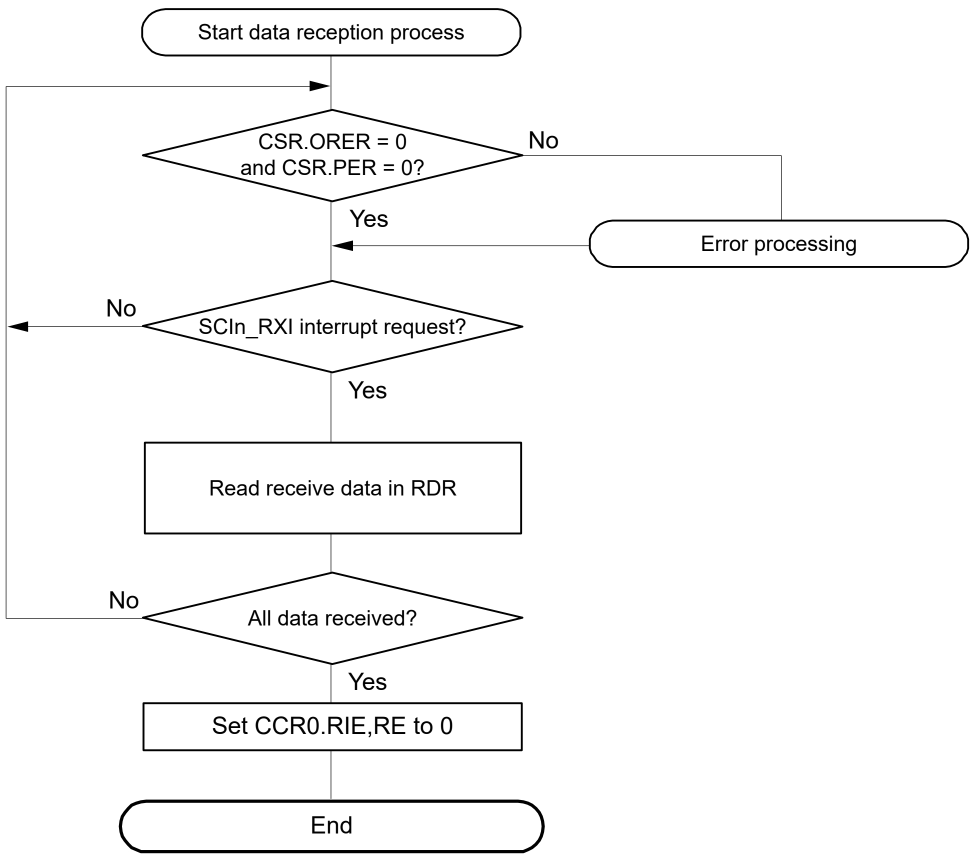


Figure 38.84 Example flow of smart card interface reception

38.7.8 Clock Output Control

When the GM bit in CCR3 is set to 1, the clock output can be controlled by the CKE[1:0] bits in CCR3. For details on the CKE[1:0] bits, see section 38.2.8. CCR3 : Common Control Register 3. When setting the clock output, the base clock described in section 38.7.4. Receive Data Sampling Timing and Reception Margin the bit rate is set by CCR2.CKS, CCR2.BCP[2:0] and BRR[7:0].

Figure 38.85 shows an example timing for the clock output control when the CKE[1] bit in CCR3 is set to 0 and the CKE[0] bit in CCR3 is controlled.

When the GM bit in CCR3 is 0, output control by the CKE[0] bit in CCR3 is immediately reflected on the SCKn pin, so there is a possibility that pulses with an unintended width may be output from the SCKn pin.

When the GM bit in CCR3 is 1, the output pulse control by the CCR3.CKE [0] controls the pulse width set to be based on the state of the base clock.

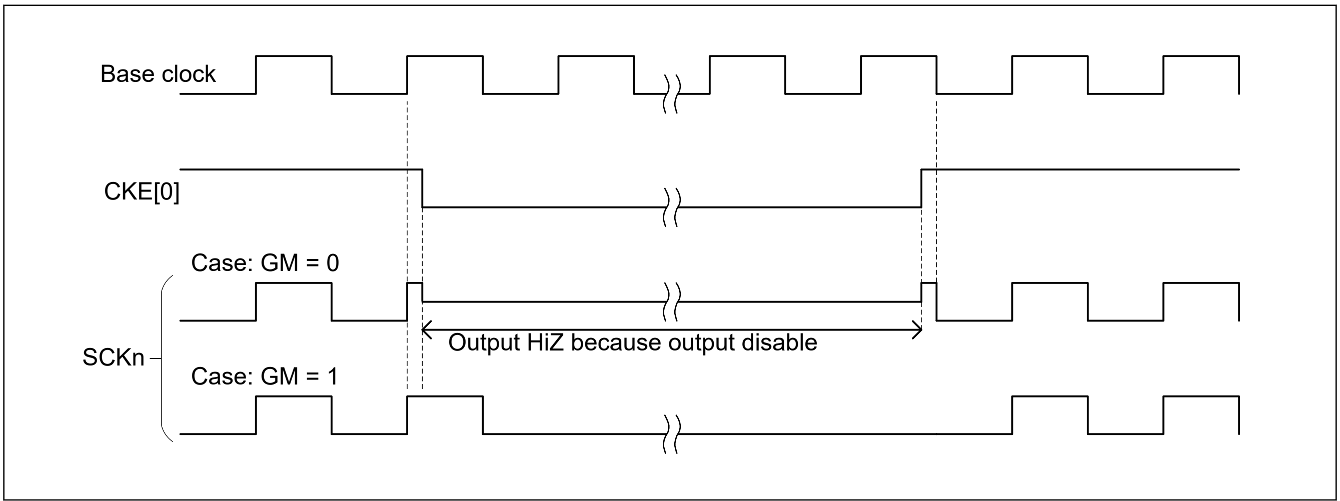


Figure 38.85 Clock Output timing

38.8 Operation in Simple IIC Mode

Simple IIC mode format is composed of 8 data bits and an acknowledge bit. By continuing into a slave-address frame after a start condition or restart condition, a master device can specify a slave device as the partner for communications. The currently specified slave device remains valid until a new slave device is specified or a stop condition is satisfied. The 8 data bits in all frames are transmitted in order from the MSB.

The I^2C bus format and timing of the I^2C bus are shown in Figure 38.86 and Figure 38.87.

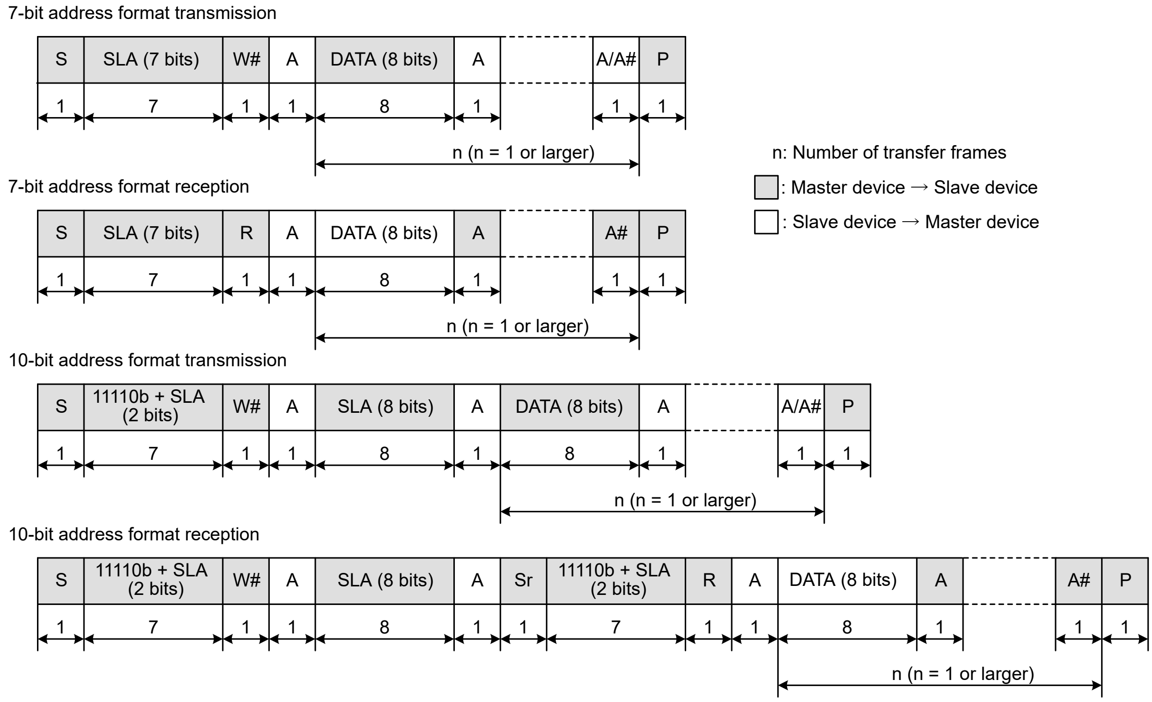


Figure 38.86 I²C bus format

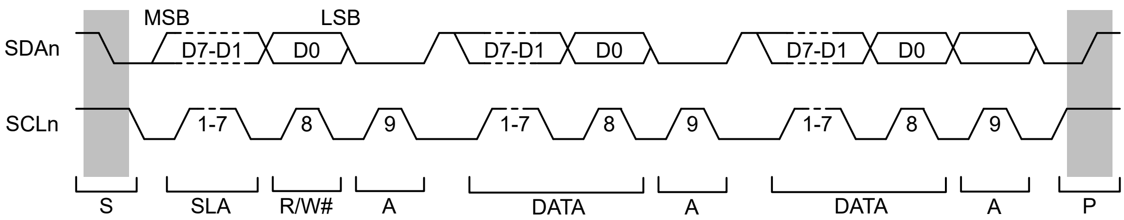


Figure 38.87 I²C bus timing when SLA is 7 bits

- S: Indicates a start condition, when the master device changes the level on the SDAn line from high to low while the SCLn line is high
- SLA: Indicates a slave address, by which the master device selects a slave device
- R/W#: Indicates the direction of transfer (reception or transmission). The value 1 indicates transfer from the slave device to the master device and 0 indicates transfer from the master device to the slave device.
- A/A#: Indicates an acknowledge bit. This is returned by the slave device for master transmission and by the master device for master reception. Return low indicates ACK and return high indicates NACK.
- Sr: Indicates a restart condition, when the master device changes the level on the SDAn line from high to low while the SCLn line is high and after the setup time elapses
- DATA: Indicates the data being received or transmitted
- P: Indicates a stop condition, when the master device changes the level on the SDAn line from low to high while the SCLn line is high

38.8.1 Generation of Start, Restart, and Stop Conditions

Writing 1 to the ICR.IICSTAREQ bit causes the generation of a start condition. The generation of a start condition proceeds through the following operations:

- The level on the SDAn line falls (from the high level to the low level) and the SCLn line is kept in the released state
- The hold time for the start condition is set as half of a bit period at the bit rate determined by the CCR2.BRR setting
- The level on the SCLn line falls (from the high level to the low level), the IICSTAREQ bit in ICR is set to 0, and a start-condition generated interrupt is output

Writing 1 to the IICRSTAREQ bit in ICR causes the generation of a restart condition. The generation of a restart condition proceeds through the following operations:

- The SDAn line is released and the SCLn line is kept at the low level
- The period at low level for the SCLn line is set as half of a bit period at the bit rate determined by the CCR2.BRR setting
- The SCLn line is released (transition from the low to the high level)
- When a high level is detected on the SCLn line, the setup time for the restart condition is set as half of a bit period at the bit rate determined by the CCR2.BRR setting
- The level on the SDAn line falls (from the high level to the low level)
- The hold time for the restart condition is set as half of a bit period at the bit rate determined by the CCR2.BRR setting
- The level on the SCLn line falls (from the high level to the low level), the ICR.IICRSTAREQ bit is set to 0, and a restart-condition generated interrupt is output

Writing 1 to the ICR.IICSTPREQ bit causes the generation of a stop condition. The generation of a stop condition proceeds through the following operations:

- The level on the SDAn line falls (from the high level to the low level) and the SCLn line is kept at the low level
- The period at low level for the SCLn line is set as half of a bit period at the bit rate determined by the CCR2.BRR setting
- The SCLn line is released (transition from the low to the high level)
- When a high level is detected on the SCLn line, the setup time for the stop condition is set as half of a bit period at the bit rate determined by the CCR2.BRR setting
- The SDAn line is released (transition from the low to the high level), the ICR.IICSTPREQ bit is set to 0, and a stop-condition generated interrupt is output

Figure 38.88 shows the timing of operations in the generation of start, restart, and stop conditions.

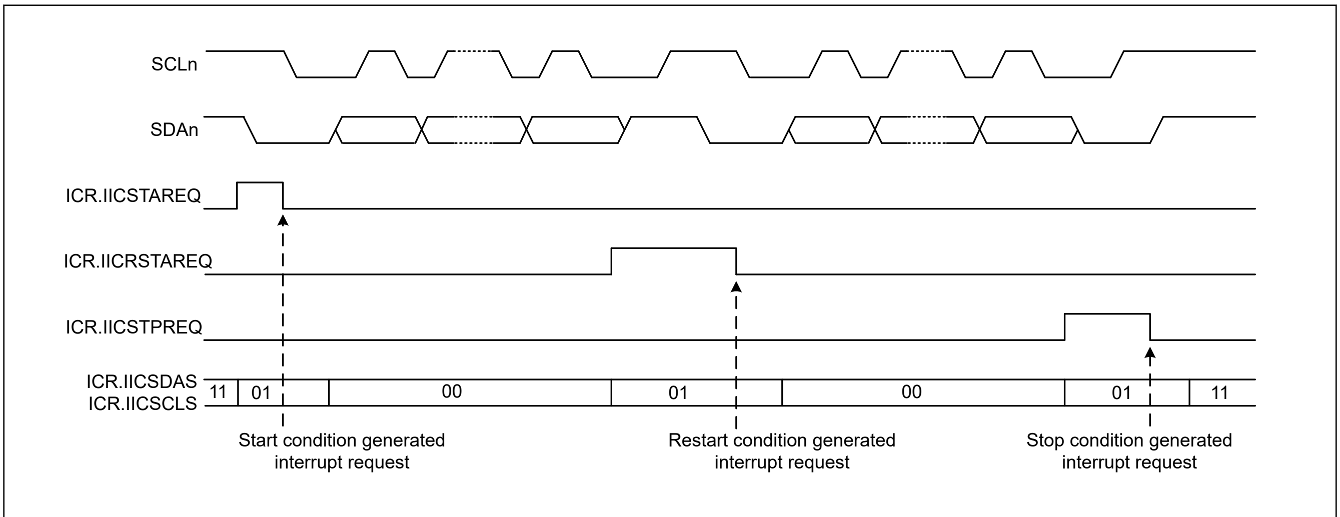


Figure 38.88 Timing of operations in generation of start, restart, and stop conditions

38.8.2 Clock Synchronization

The SCLn line can be driven low if a wait is inserted by a slave device at the other side of the transfer. Setting the ICR.IICCSC bit to 1 applies control to obtain synchronization when a difference arises between the levels of the internal SCLn clock signal and the level being input on the SCLn pin.

When the ICR.IICCSC bit is set to 1, the level of the internal SCLn clock signal changes from low to high. Counting to determine the period at a high level stops while the low level is being input on the SCLn pin. Counting to determine the period at a high level starts after the transition of the input on the SCLn pin to the high level.

The interval from this time until counting to determine the period at high level starts on the transition of the SCLn pin to the high level, is the total time which contains the SCLn input delay, delay for noise filtering of the input on the SCLn pin (2 or 3 cycles of sampling clock for the noise filter), and delay for internal processing (1 or 2 cycles of PCLK). The period at high level of the internal SCLn clock is extended even when other devices do not place the low level on the SCLn line.

If the ICR.IICCSC bit is 1, synchronization is obtained for the transmission and reception of data by taking the logical AND of the input on the SCLn pin and the internal SCLn clock. If the ICR.IICCSC bit is 0, synchronization with the internal SCLn clock is obtained for the transmission and reception of data.

If a slave device inserts a wait period into the interval until the transition of the internal SCLn clock signal from the low to the high level after a request for the generation of a start, restart, or stop condition is issued, the time until generation is prolonged by that period.

If a slave device inserts a wait period after the transition of the internal SCLn clock signal from the low to the high level, although the generation-completed interrupt is issued without stopping the waiting period, generation of the condition itself is not guaranteed.

Figure 38.89 shows an example operation for synchronizing the clocks.

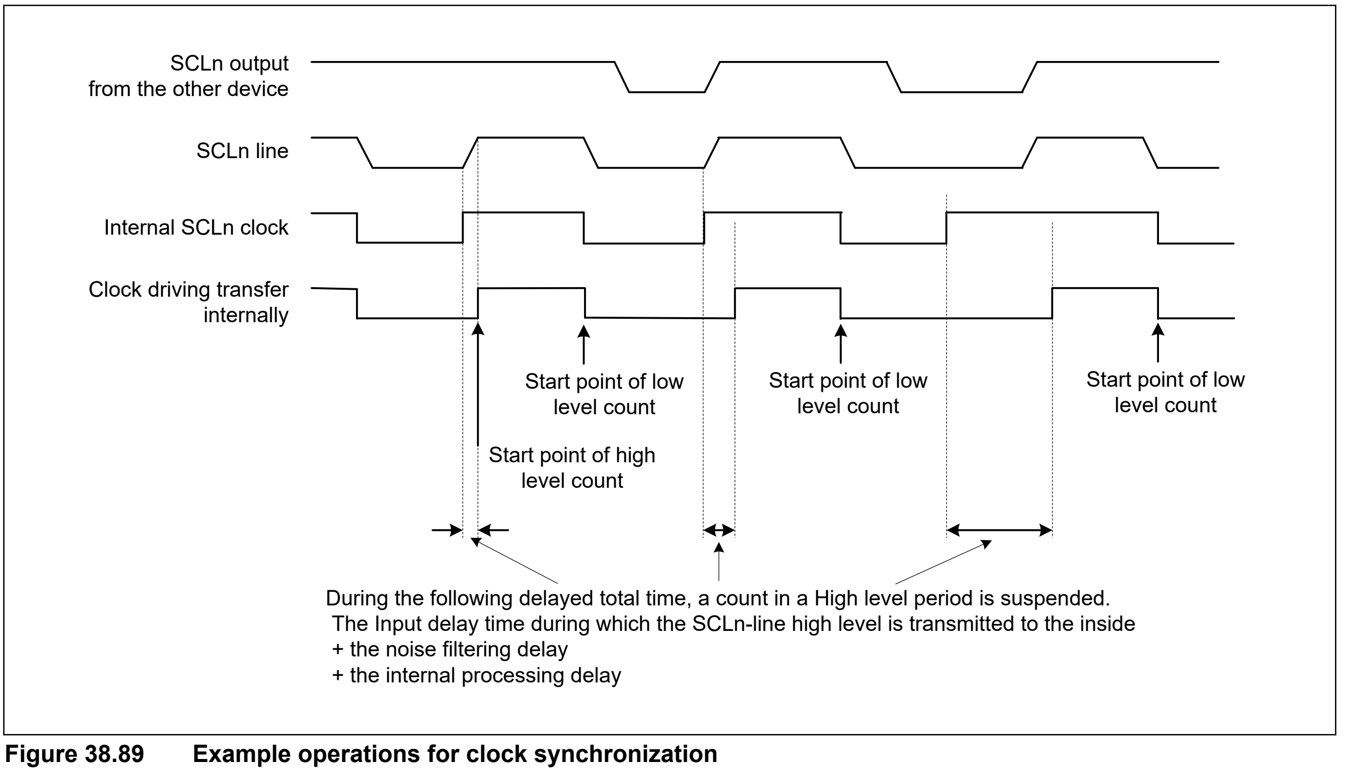


Figure 38.89 Example operations for clock synchronization

38.8.3 SDAn Output Delay

The ICR.IICDL[4:0] bits can be used to set a delay for output on the SDAn pin relative to falling edges of output on the SCLn pin. Delay settings from 0 to 31 are selectable, representing periods of the corresponding numbers of cycles of the clock signal from the on-chip baud rate generator (derived by frequency-dividing the base clock, TCLK, by the divisor selected in the CCR2.CKS[1:0] bits). A delay for output on the SDAn pin applies to the start condition/restart condition/stop condition signal, 8-bit transmit data, and acknowledge bit.

If the SDAn output delay is shorter than the time for the level on the SCLn pin to fall, the change of the output on the SDAn pin starts while the output level on the SCLn pin is falling, creating a possibility of erroneous operation for slave devices. Ensure that settings for the delay of output on the SDAn pin specify times greater than the time output on the SCLn pin takes to fall (300 ns for IIC in normal mode and fast mode).

Figure 38.90 shows the timing of delays in SDAn output.

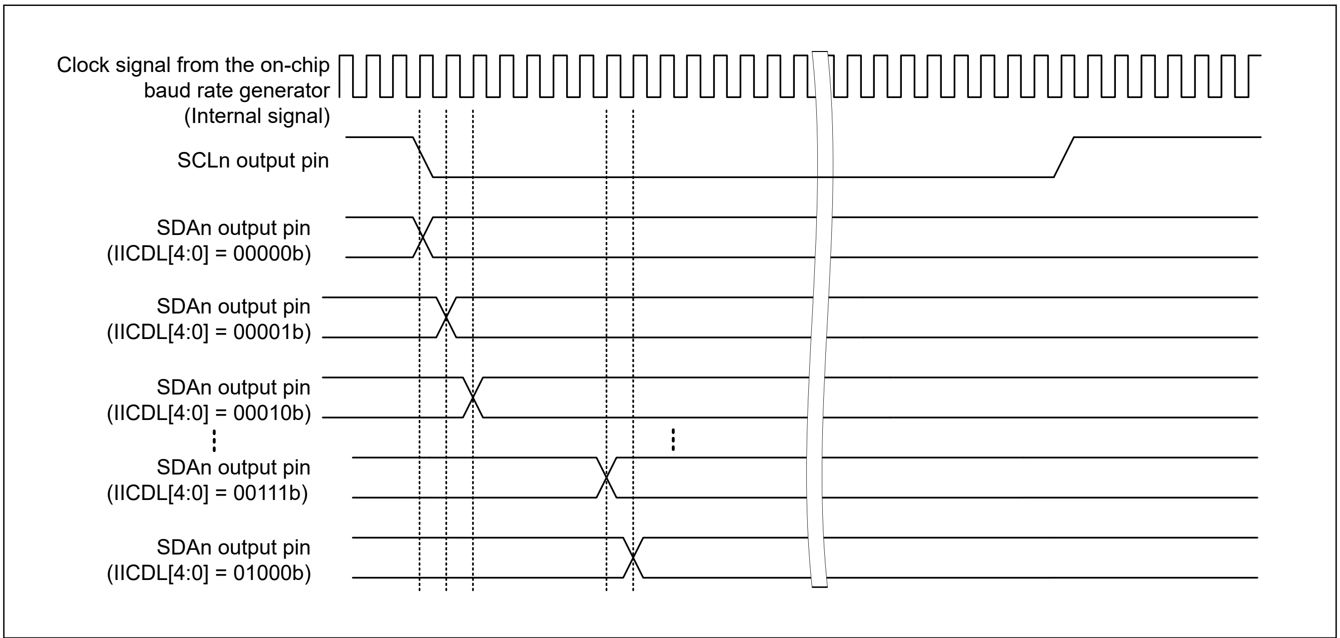


Figure 38.90 Timing of delays in SDAn output

38.8.4 SCI Initialization in Simple IIC Mode

Before transferring data, write the initial value 0x00 to CCR0 and initialize the interface following the example shown in Table 38.40.

Before making any changes to the operating mode or transfer format, be sure to set CCR0 to its initial value. In simple IIC mode, the open-drain setting for the communication ports should be made on the port side.

Table 38.40 Example flow of SCI initialization in simple IIC mode

No.	Step Name	Description
1	Start initialization	—
2	Set CCR0	Set CCR0.TEIE, TIE, RIE, TE, RE to 0. If you have not changed from the initial settings, you can skip this step.
3	Set ICR	Set the IICSDAS[1:0] and IICSCLS[1:0] to 11b. Set the IICDL[4:0] and IICINTM as required. Set the IICACKT and the IICCSC bits to 1.
4	Set CCR3	Set the transmission / reception format as the communication mode (MOD [2:0] = 100b) and CKE [1:0] = 00b.
5	Set CCR2	Set the bit rate modulation function*1, the clock selection, and the bit rate.
6	Set CCR1	Set noise filter, communication pin status, parity check, and CTSn / RTSn function.
7	Set the I/O port functions	Set I/O port settings that allow use (on NMOS open-drain output pins and Hi-Z) of the SCLn and SDAn pin functions.
8	Set CFCLR, ICFCLR	Write 1 to the following bits and clear the corresponding flag. CFCLR.RDRFC, FERC, PERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC ICFCLR.IICSTIFC
9	Set CCR0 (TE, RE, TIE, RIE)	Set the TE and RE bits to 1. To enable interrupts, set the TE, TIE, RE and RIE bits to 1 with one instruction at the same time (for transmission and when the IICINTM bit is 1, clear the RIE bit). Setting the TE and RE bits to 1 makes the SCLn and SDAn pins functions available.
10	Initialization completed	—

Note: Set the CCR0.TE and RE bits to 0 or 1 at the same time.

Note 1. If you do not use the bit rate modulation function, you do not need to set it.

38.8.5 Operation in Master Transmission in Simple IIC Mode

Figure 38.91 and Figure 38.92 show examples of master transmission and Figure 38.93 shows an example flow of data transmission.

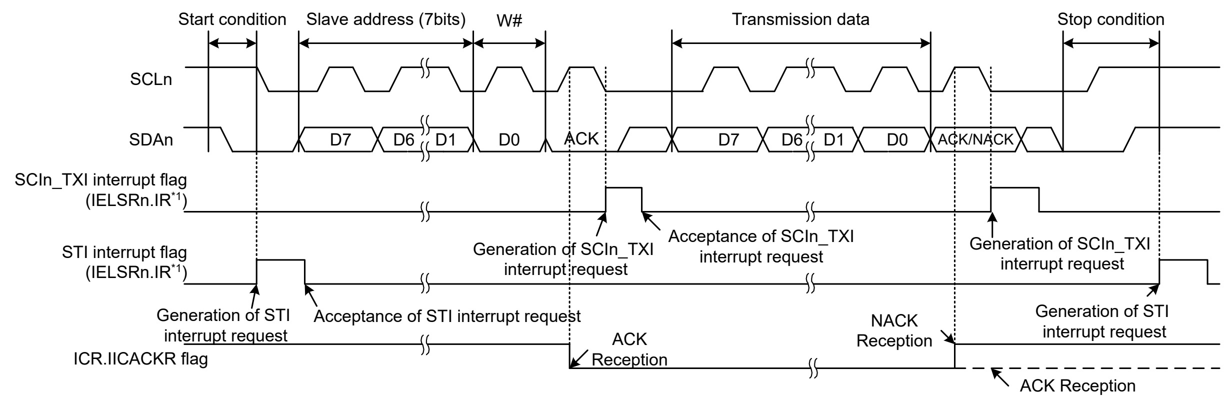
Figure 38.91 shows the operation example when ICR.IICINTM bit is 1 (use reception and transmission interrupts). In this case, you can start DMAC or DTC by SCIn_TXI interrupt. However, if you use DMAC or DTC, ACK/NACK cannot be confirmed. If you want to confirm ACK/NACK, prepare the transmit data by CPU.

In simple IIC mode, SCIn_TXI interrupt is generated when communication of one frame is completed. And it is not used SCIn_RXI interrupt in master transmission, so the CCR0.RIE set to 0.

See Table 38.45 for more information on the STI interrupt.

Figure 38.93 shows a flow chart in the case of ICR.IICINTM is 1 and address transmission by CPU and data transmission by DTC or DMAC. When 10-bit slave addresses are in use, steps [3] and [4] are repeated twice.

In simple IIC mode, the transmit data empty interrupt (SCIn_TXI) is generated when communication of one frame is complete, unlike the SCIn_TXI interrupt request generation timing during clock synchronous transmission.



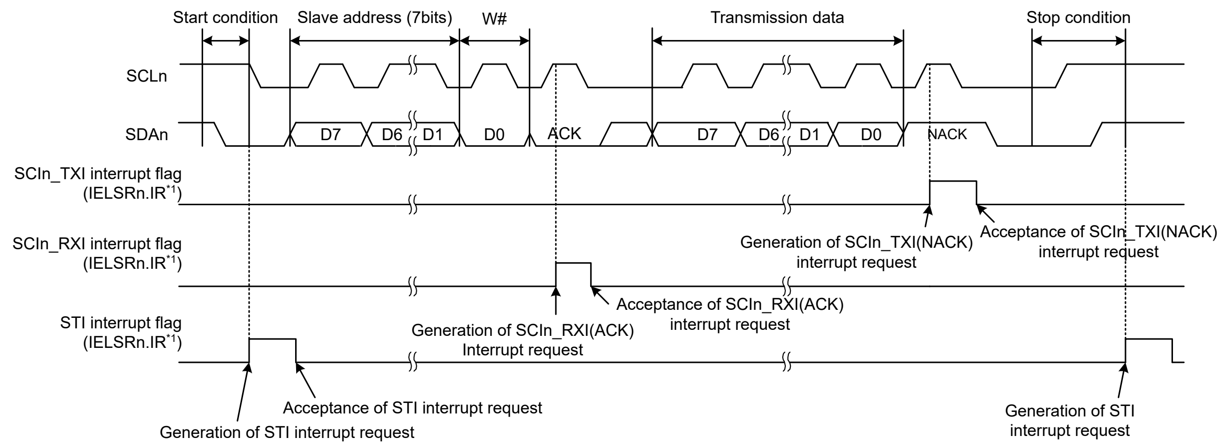
Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 38.91 Example 1 of operations for master transmission in simple IIC mode with 7-bit slave addresses, transmission interrupts, and reception interrupts (ICR.IICINTM = 1)

When the ICR.IICINTM bit is set to 0 (use ACK/NACK interrupts) during master transmission, the DTC or DMAC is activated by the ACK interrupt as the trigger and required number of data bytes are transmitted. When the NACK is received, error processing such as transmission stop and retransmission is performed using the NACK interrupt as the trigger.

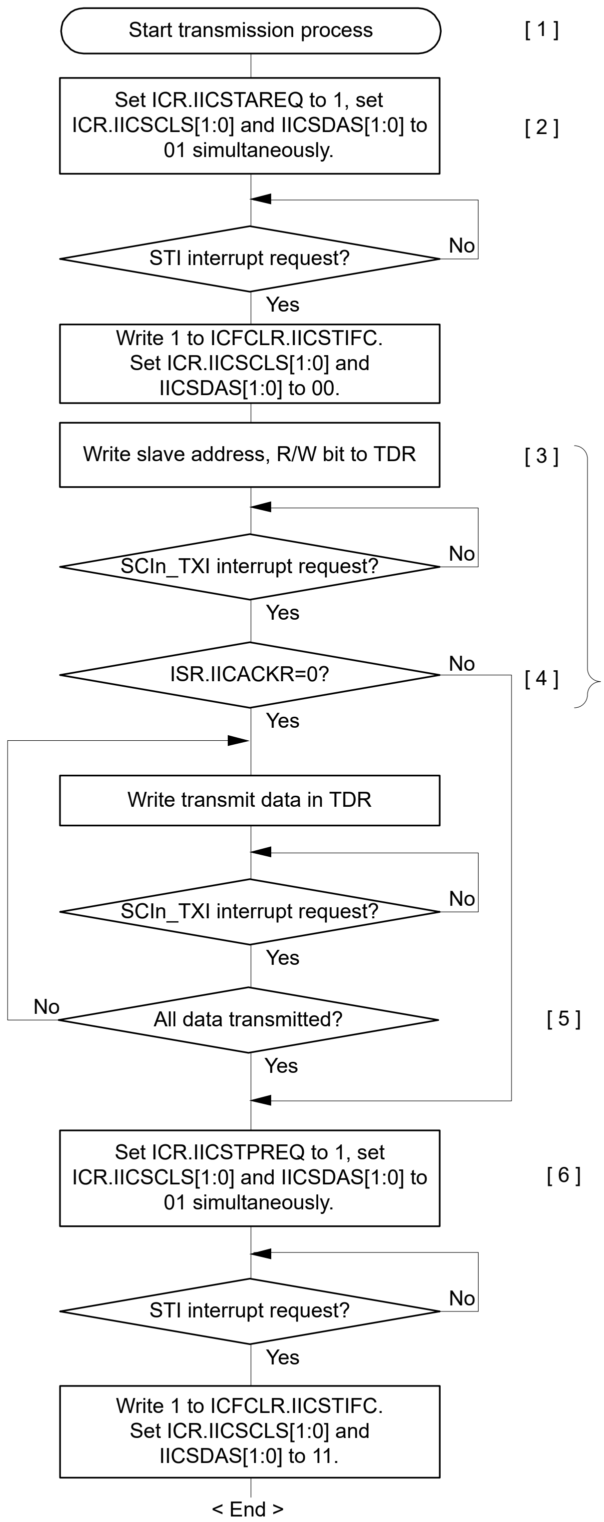
To restart communication for some reason after writing data in the TDR register, use the following procedure:

1. Set the TE and RE bits in the CCR0 register to 0 to stop communication.
2. Set ICR.IICSCLS[1:0] and ICR.IICSDAS[1:0] bits to 11b, release the I^2C bus, and clear the generation of a condition.
3. If the RDRF flag in the CSR register is set to 1, the RDR register is read by dummy and the RDRF bit is set to 0.
4. Set the TE and RE bits in the CCR0 register to 1 and start the next communication.



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 38.92 Example 2 of operations for master transmission in simple IIC mode with 7-bit slave addresses, ACK interrupts, and NACK interrupts (ICR.IICINTM = 0)



[1] Start transmission process
For transmission, set the CCR0.RIE to 0.
(SCIn_RXI and SCIn_ERI interrupts requests are disabled)
[2] Generate a start condition.
[3] Writing to TDR:
Writing the slave address and value for the R/W bit to TDR.
[4] Confirm ACK response from the slave device:
Check the ISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.
[5] Steps for continuing with serial transmission:
When transmission is to continue, write further transmit data to TDR. Except for the first data to be transmitted, a SCIn_TXI interrupt request can activate the DMAC or DTC to handle writing of data to TDR.
But it cannot be confirmed status of ACK or NACK in this case.
[6] Generation of a stop condition.

If 10-bits slave address are in use, processing from [3] to [4] is repeated twice.

Note: In simple IIC mode, the SCIn_TXI interrupt request is generated when communication is completed.

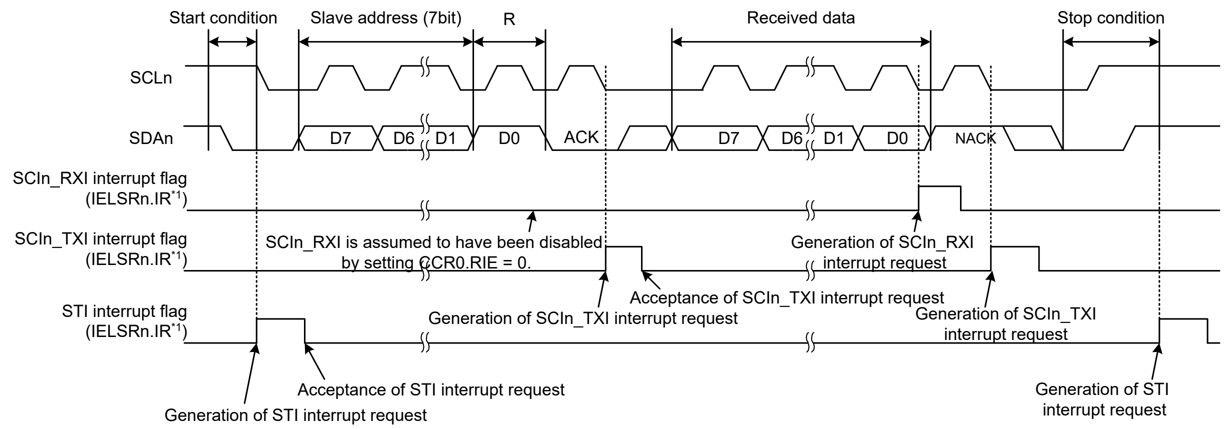
Figure 38.93 Example flow of master transmission in simple IIC mode with transmission interrupts and reception interrupts

38.8.6 Master Reception in Simple IIC Mode

Figure 38.94 shows an example operation in simple IIC mode master reception and Figure 38.96 shows an example flow of master reception.

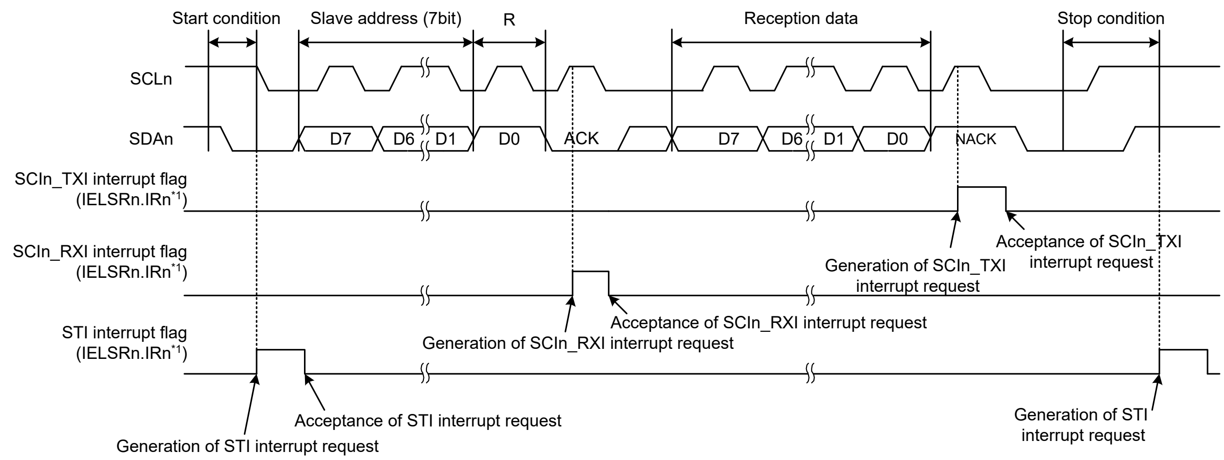
The value of the ICR.IICINTM bit is assumed to be 1 (use reception and transmission interrupts) and 0 (use ACK and NACK interrupts).

In simple IIC mode, the transmit data empty interrupt (SCIn_TXI) is generated when communication of one frame is complete, unlike the SCIn_TXI interrupt request generation timing during clock synchronous transmission.



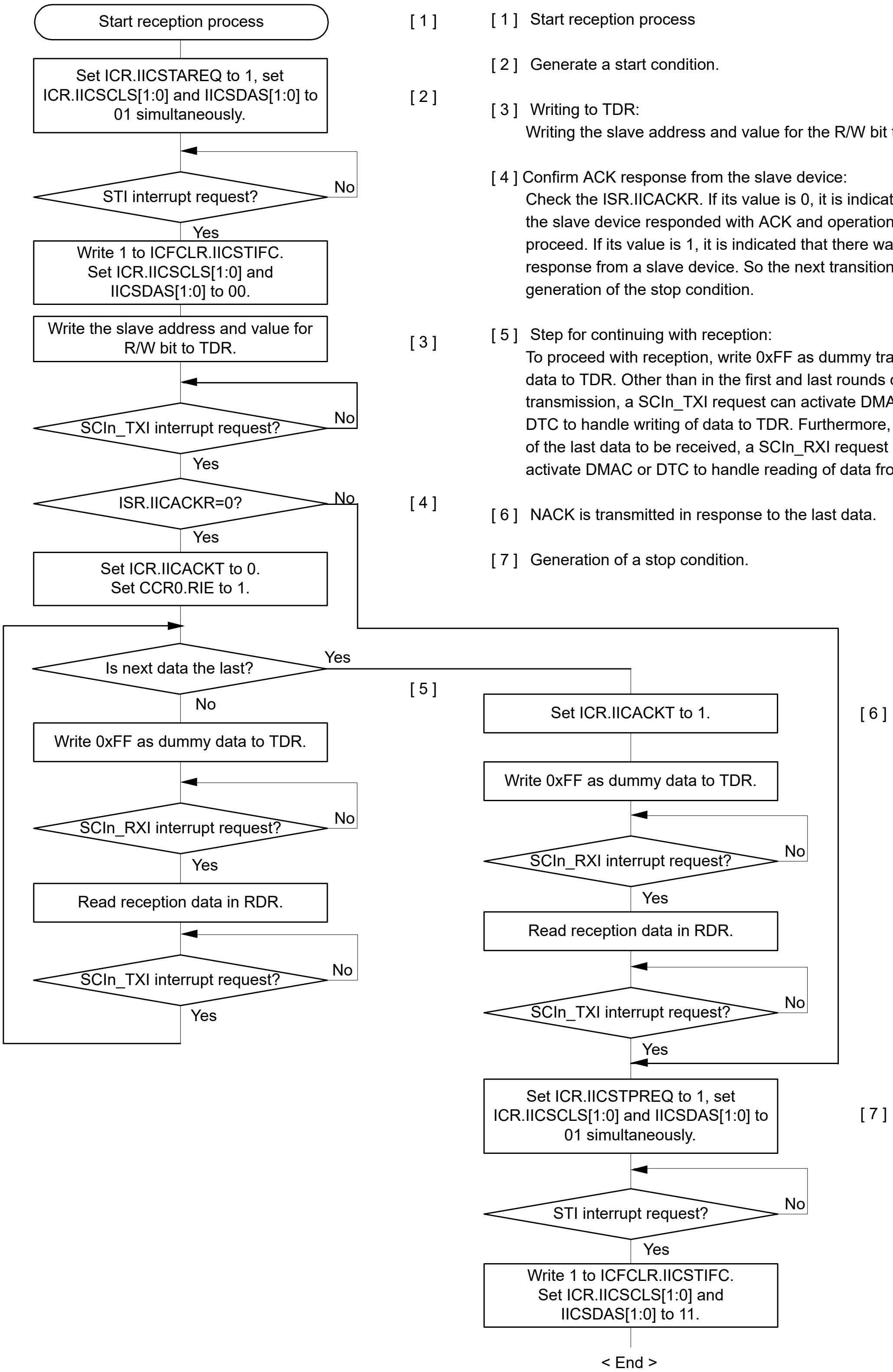
Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 38.94 Example operations for master reception in simple IIC mode with 7-bit slave addresses, transmission interrupts, and reception interrupts (ICR.IICINTM = 1)



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 38.95 Example of Operations for Master Reception in Simple IIC Mode (7bit Slave address, ACK and NACK interrupt in use (ICR.IICINTM = 0))



[1] Start reception process

[2] Generate a start condition.

[3] Writing to TDR:
Writing the slave address and value for the R/W bit to TDR.

[4] Confirm ACK response from the slave device:
Check the ISR.IICACKR. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device. So the next transition is to generation of the stop condition.

[5] Step for continuing with reception:
To proceed with reception, write 0xFF as dummy transmit data to TDR. Other than in the first and last rounds of transmission, a SCIn_TXI request can activate DMAC or DTC to handle writing of data to TDR. Furthermore, except of the last data to be received, a SCIn_RXI request can activate DMAC or DTC to handle reading of data from RDR.

[6] NACK is transmitted in response to the last data.

[7] Generation of a stop condition.

Note: In simple IIC mode, the SCIn_TXI interrupt request is generated when communication is completed.

Figure 38.96 Example flow of master reception in simple IIC mode with transmission interrupts and reception interrupts (ICR.IICINTM = 1)

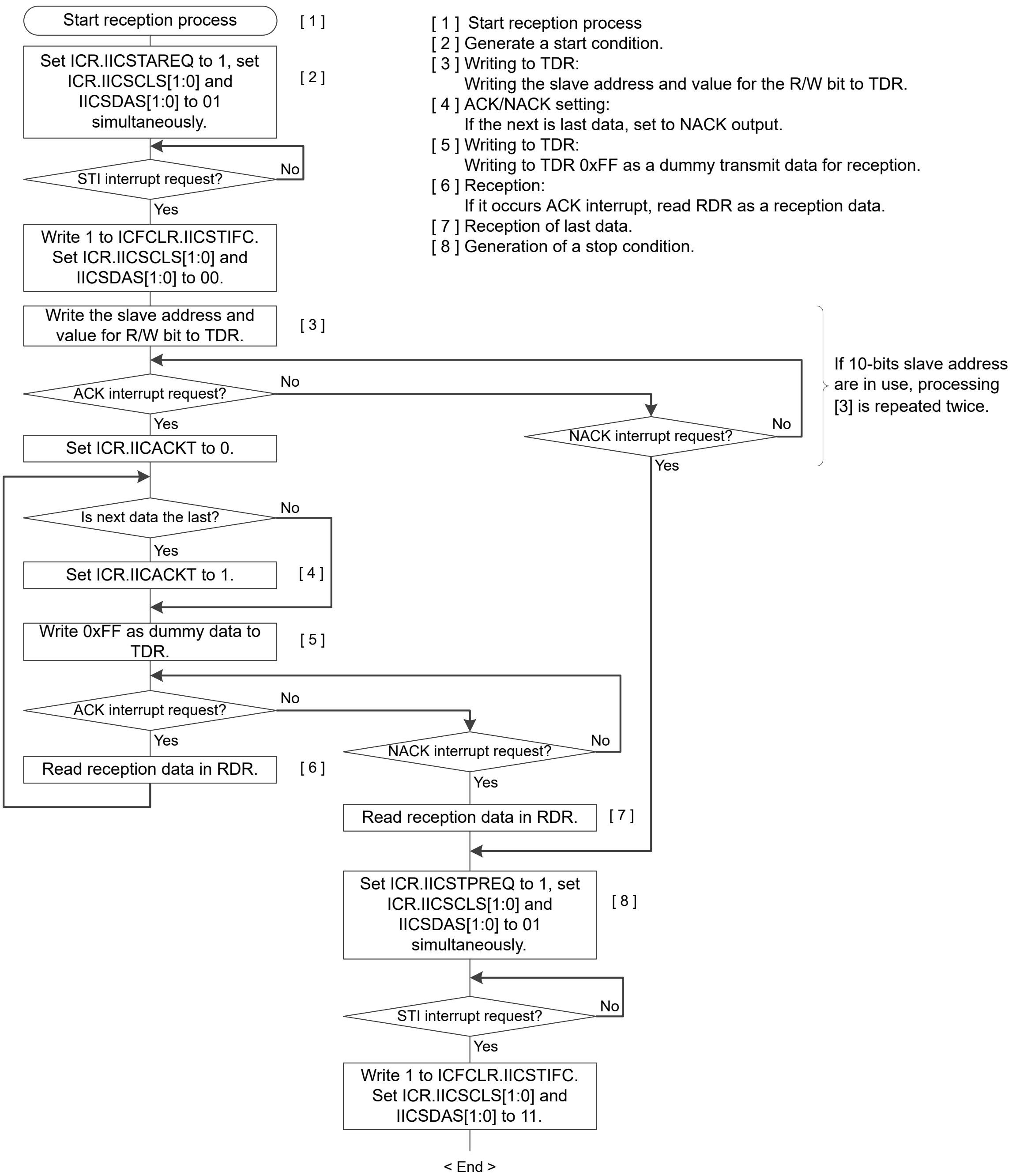


Figure 38.97 Example flow of master reception in simple IIC mode with ACK Interrupts and NACK Interrupts (ICR.IICINTM = 0)

38.9 Operation in Simple SPI Mode

As an extended function, the SCI supports a simple SPI mode that handles transfer among one or multiple master devices and multiple slave devices.

Using the settings for Simple SPI mode setting (CCR3.MOD[2:0] bit = 011b) and setting the CCR0.SSE bit to 1 place the SCI in simple SPI mode. However, the SSn pin function on the master side is not required for connection of the device used as the master in simple SPI mode when the configuration only has a single master. Therefore, set the CCR0.SSE bit to 0 in such cases.

Figure 38.98 shows an example of connections for simple SPI mode. Control a general port pin to produce the SSn output signal from the master.

In simple SPI mode, data is transferred in synchronization with clock pulses in the same way as in clock synchronous mode. One character of data for transfer consists of 8 bits of data, and parity bits cannot be appended. The data can be inverted by setting the CCR3.SINV bit to 1.

Because the receiver and transmitter are independent of each other within the SCI module, full-duplex communications are possible, with a shared clock signal. Additionally, because both the transmitter and receiver have a buffered structure, writing the next transmit data while transmission is in progress and reading previously received data while reception is in progress are both possible. This enables continuous transfer.

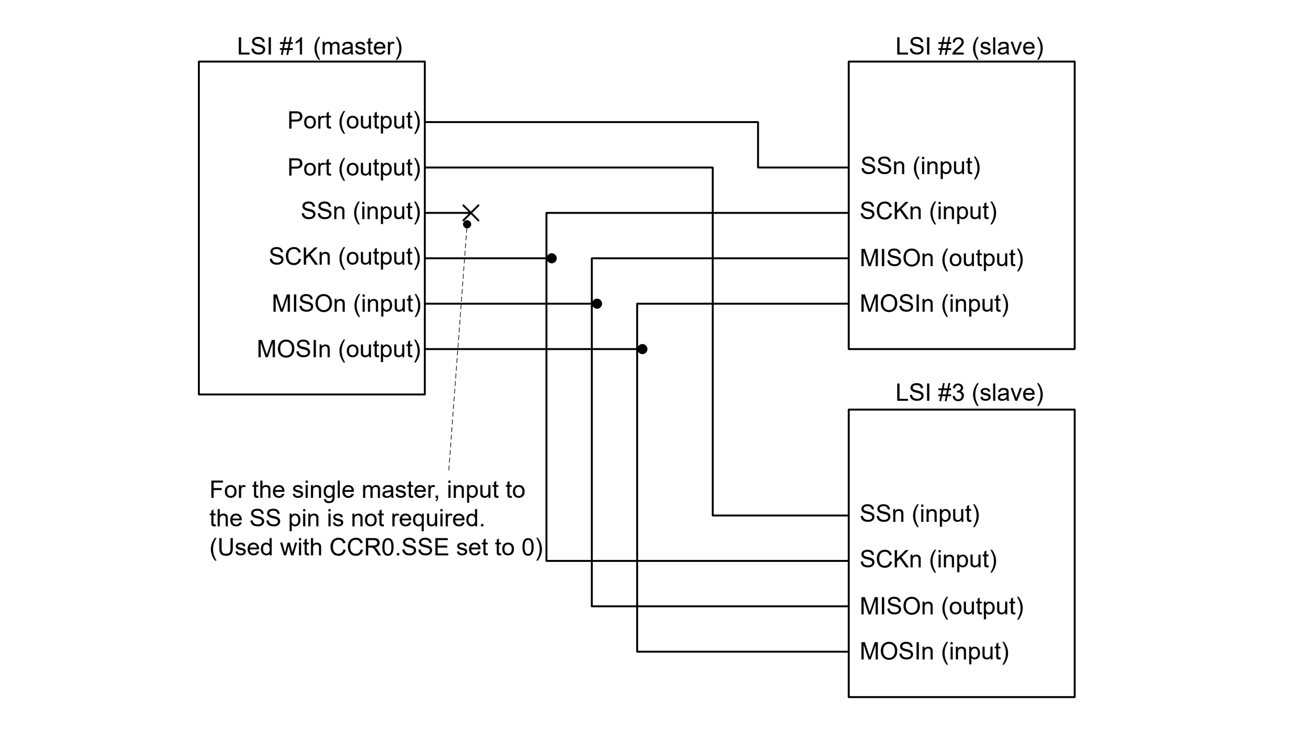


Figure 38.98 Example connections using simple SPI mode in single master mode with CCR0.SSE bit = 0

38.9.1 States of Pins in Master and Slave Modes

The direction (input or output) of pins for the simple SPI mode interface differs according to whether the device is a master (CCR3.CKE[1:0] = 00b or 01b) or slave (CCR3.CKE[1:0] = 10b or 11b).

Table 38.41 lists the relationship between the pin states, mode, and level on the SSn pin.

Table 38.41 States of pins by mode and input level on SSn pin

Mode	Input on SSn pin	State of MOSIn pin	State of MISOn pin	State of SCKn pin
Master mode[*1]	High level (transfer can proceed)	Output for data transmission[*2]	Input for received data	Clock output[*3]
	Low level (transfer cannot proceed)	High-impedance	Input for received data (but disabled)	High-impedance
Slave mode	High level (transfer cannot proceed)	Input for received data (but disabled)	High-impedance	Clock input (but disabled)
	Low level (transfer can proceed)	Input for received data	Output for data transmission	Clock input

Note 1. When there is only a single master (CCR0.SSE = 0), transfer is possible regardless of the input level on the SSn pin. This is equivalent to input of a high level on the SSn pin. The SSn pin is not used and is available for other purposes.

Note 2. The MOSIn pin output is in the high-impedance state when serial transmission is disabled (CCR0.TE bit = 0).

Note 3. The SCKn pin output is in the high-impedance state when serial transmission is disabled (CCR0.TE = 0 and CCR0.RE = 0) in a multi-master configuration (CCR0.SSE = 1).

38.9.2 SS Function in Master Mode

Setting the CKE[1:0] bits in the CCR3 to 00b or 01b selects master mode operation. The SSn pin is not used in single-master configurations (CCR0.SSE = 0), so transmission or reception can proceed regardless of the value of the SSn pin.

When the level on the SSn pin is high in a multi-master configuration (CCR0.SSE = 1), a master device outputs clock signals from the SCKn pin before starting transmission or reception to indicate that there are no other masters or another master is performing reception or transmission.

When the level on the SSn pin is low in a multi-master configuration (CCR0.SSE = 1), there are other masters, and a transmission or reception is in progress. The MOSIn output and SCKn pins are placed in the high-impedance state and starting transmission or reception is not possible. In addition, the value of the CSR.MFF bit is 1, indicating a mode fault error. In a multi-master configuration, start error processing by reading CSR.MFF flag. If a mode fault error occurs while transmission or reception is in progress, transmission or reception does not stop, but the MOSIn and SCKn outputs are in the high-impedance state after completion of the transfer.

When the SSn pin input becomes high level, the SCKn pin outputs a clock signal and the MOSIn outputs data. Even if the SCKn pin and the MOSIn pin are in the high impedance state, internal transmission or reception operation continues, but it stops after transmission or reception of a single character is complete. In this case, any of SCIn_TXI, SCIn_RXI, and SCIn_TEI interrupts occurs.

Use a general port pin to produce the SS output signal from the master.

38.9.3 SS Function in Slave Mode

Setting the CCR3.CKE[1:0] bits to 10b or 11b selects slave operation. When the SSn pin is high, the MISOn output pin is in the high-impedance state and clock input through the SCKn pin is ignored. When the SSn pin is low, clock input through the SCKn pin is valid and transmission or reception can proceed.

If the input on the SSn pin changes from low to high during transmission or reception, the MISOn output pin is placed in the high-impedance state. Transmission / reception operation is immediately suspended. If the transmission is in progress, the CSR.TEND flag will not be set, a transmit end interrupt will not be output, and an abnormal stop status will occur. So, do not negate the SSn pin during slave transmission / reception. If an abnormal stop occurs, set CCR0.RE and CCR0.TE to 0 to stop transmission / reception. To resume transmission / reception, set CCR0.RE and CCR0.TE to 1 after at least TCLK × 3 cycles + PCLK × 3 cycles.

38.9.4 Relationship Between Clock and Transmit/Receive Data

The CPOL and CPHA bits in the CCR3 register can be used to set up the clock for use in transmission and reception in four different ways. The relation between the clock signal and the transmission and reception of data is shown in Figure 38.99. The relation is the same for both master and slave operation. This is the same as when the level on the SSn pin is high.

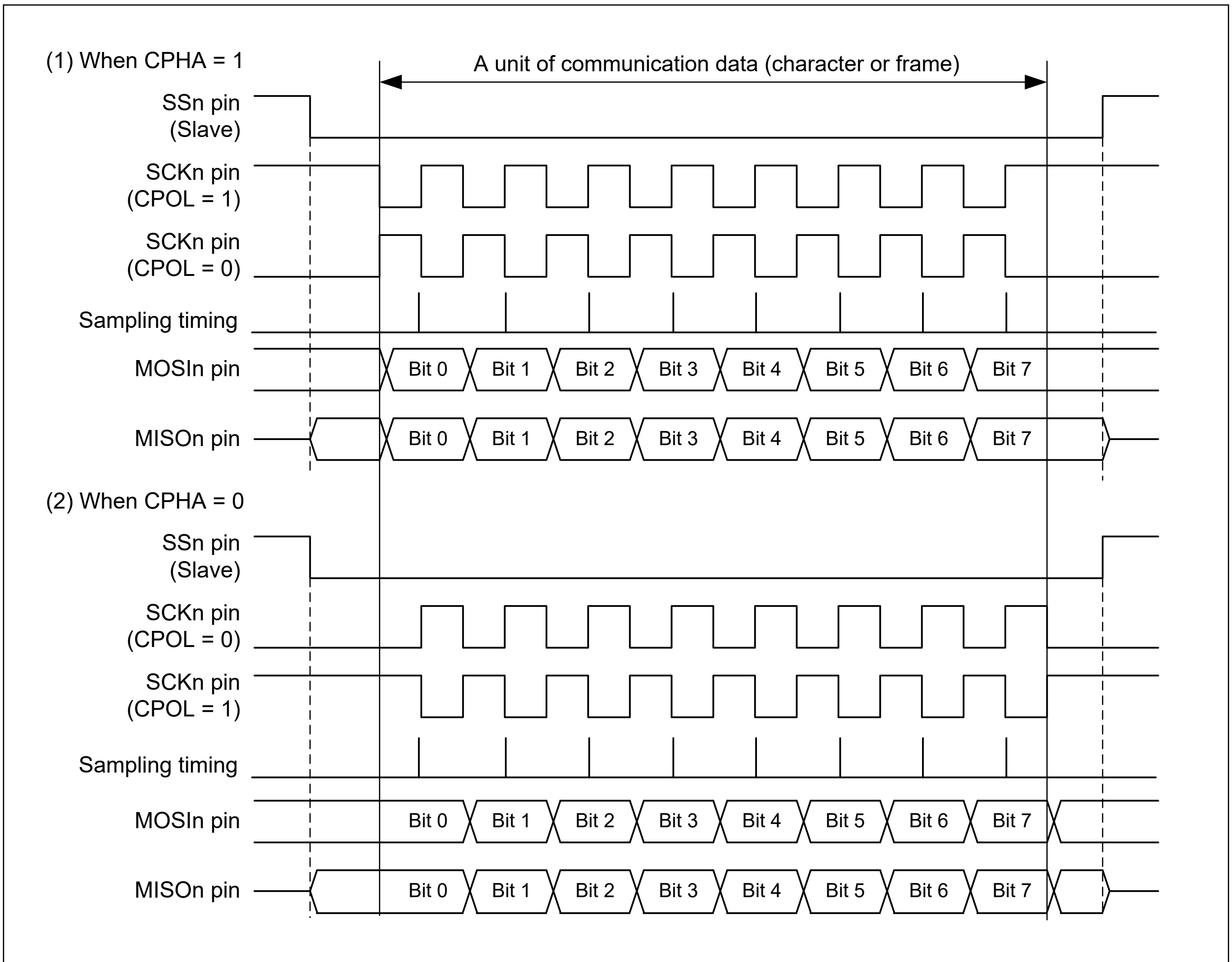


Figure 38.99 Relation between clock signal and transmit or receive data in simple SPI mode

38.9.5 SCI Initialization in Simple SPI Mode

Initialization in simple SPI mode is the same as in clock synchronous mode. See section 38.6.3. SCI Initialization in Clock Synchronous Mode for an example initialization flow. The CPOL and CPHA bits in the CCR3 register must be set to ensure that the clock signal is suitable for both master and slave devices.

Always initialize the CCR0 register before making any changes to the operating mode or transfer format.

Note: Only the RE bit is set to 0. The CSR.ORER, FER, PER, and RDR flags are not initialized.

Changing the value of the TE bit from 1 to 0 or from 0 to 1 when the TIE bit in the CCR0 register is 1 at the same time, leads to the generation of a transmit data empty interrupt (SCIn_TXI).

38.9.6 Transmission and Reception of Serial Data in Simple SPI Mode

In master operation, ensure that the SSn pin of the slave device on the other side of the transfer is at the low level before starting the transfer and at the high level on completion of the transfer. In multiple master operation with CCR0.SSE = 1 even in master mode, a mode fault error will occur if the SSn pin goes low. Therefore, make sure that no mode fault error has occurred before starting communication, and start communication, and make sure that no mode fault error has occurred even after communication ends. If a mode fault error has occurred, communication may be incomplete, so measures such as retransmission are required. Otherwise, the procedures are the same as in clock synchronous mode.

In slave mode, it operates according to the SSn pin input level. Other steps are the same as those of clock synchronous mode.

38.9.7 Reception Sampling Timing Adjustment Function in Simple SPI Mode with Internal Clock Used

The reception sampling timing adjustment function in simple SPI mode is the same as the reception sampling timing adjustment function in clock synchronous mode. For the description of operation, see section 38.6.7. Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with Internal Clock Used.

38.10 Bit Rate Modulation Function

Using the bit rate modulation function, the bit rate can be evenly corrected using the number specified in the MDDR register among 256 clock cycles of internal clocks which is selected by the CKS[1:0] bits in CCR2.

Figure 38.100 shows an example where the PCLK is selected in the CKS[1:0] bits in CCR2, the BRR bit is set to 0, and the MDDR is set to 160 in Asynchronous mode. In this example, the cycle of the base clock is evenly corrected (256/160) and the bit rate is also corrected (160/256).

Note: Enabling an internal clock causes bias, and expansion and contraction are generated in the pulse width of the internal base clock.

Do not use this function in clock synchronous mode, simple SPI mode, Smart Card Interface mode, Manchester mode and Simple LIN mode.

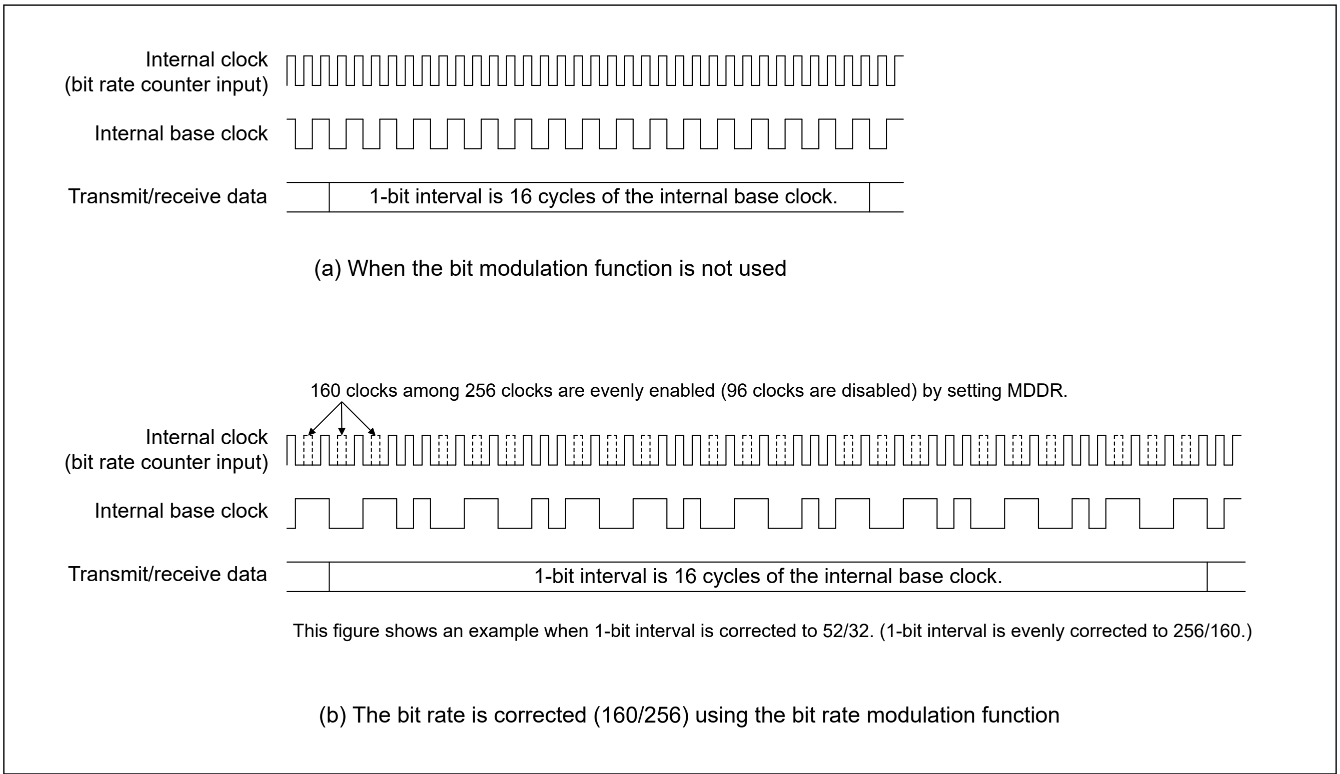


Figure 38.100 Example internal base clock when bit rate modulation function is used

38.11 Simple LIN Mode

As an extended function of the SCI, the SCI supports the serial communication protocol (Figure 38.101) consisting of a Start Frame and an Information Frame as Simple LIN. Simple LIN mode is enabled by the CCR3.MOD[2:0] = 110b. Since the Simple LIN mode uses the same circuit as the Asynchronous mode for transmission / reception control other than Break Field, the basic communication settings required for the Asynchronous mode are also required for the Simple LIN mode.

(For the setting value when using simple LIN, see the explanation in section 38.2. Register Descriptions. In particular, note that CCR3.RXDESEL needs to be changed from the initial value and set to 1.)

The Start Frame consists of a Break Field, Control Field 0, and Control Field 1. The Information Frame can be configured with some Data Fields, a CRC16 Upper Field, and a CRC16 Lower Field.

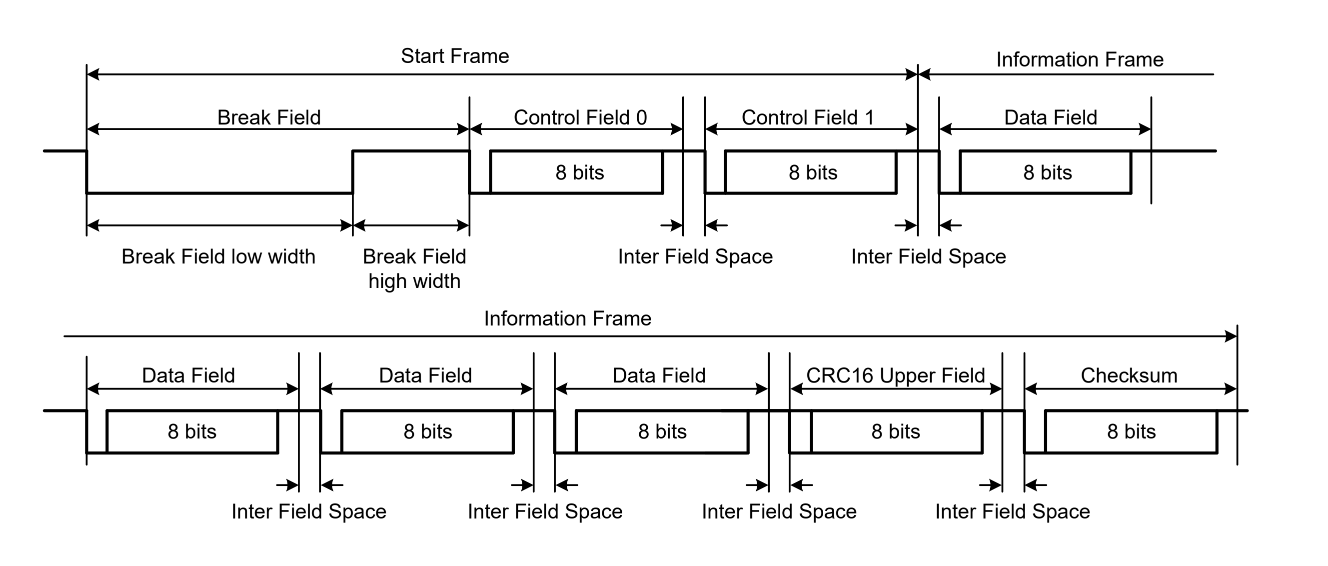


Figure 38.101 Simple LIN Protocol Example

The following describes operations when the Simple LIN is used. In this section, operations are described with the following conditions:

Communication pin (RXDn / TXDn) level inversion function: OFF (RINV = TINV = 0)

When using the simple LIN with the communication pin (RXDn / TXDn) level inversion function enabled, replace the RXDn and TXDn signal levels with their inverted levels.

38.11.1 Simple LIN Start Frame Transmission

Figure 38.102 shows an example of transmission of the Start Frame consisting of a Break Field, Control Field0, and Control Filed1. (Omit Break Field and Control Field0 according to the Start Frame configuration.)

Figure 38.103 shows a flowchart for Start Frame transmission.

The SCI operates as follows during Start Frame transmission.

1. Make the initial settings for the SCI according to the SCI initialization flow (Figure 38.68) in Asynchronous mode. In Simple LIN mode, do not set CCR0.TE and TIE to 1 at the same time to avoid SCIn_TXI output before the Break Field. Therefore, perform the following two steps sequentially to set the SCI initialization flow (Asynchronous mode) procedure [9].
   - Set the bits except CCR0.TIE. (CCR0.TIE = 0, CCR0.TE = 1, and CCR0.RE = 0)
   - Set CCR0.TIE to 1.
2. When 1 is written to TCST, the Break Field output timer starts counting and outputs a low level (Break Field) from the TXDn pin for the period set in XCR2.BFLW[15:0]. A timer count clock source can be selected by XCR0.TCSS[1:0]. Writing 0 to XCR1.TCST suspends output of the Break Field. After the suspension, set CCR0.TE = 0 and turn off the transmission.
3. When the Simple LIN module timer count value matches the set XCR2.BFLW[15:0] value, the timer stops counting and inverts the TXDn pin output level, and the XSR0.BFOF flag is set to 1[*1]. Furthermore, if XCR0.BFOIE has been set to 1 at this time, a SCIn_TXI interrupt is generated.
4. After the SCIn_TXI interrupt and confirming XSR0.BFOF = 1, write the transmitted data then the Control Field 0 data is transmitted using the SCI[*2].
5. After the Control Field 0 data has been transmitted, write the Control Field 1 data to TDR. And it is transmitted.
6. After the Control Field 1 data has been transmitted, the Information Frame data is transmitted.

Note 1. After XSR0.BFOF is set to 1, if 1 is written to XCR1.TCST without clearing it, no SCIn_TXI interrupt is output at the end of Break Field transmission. Clear XSR0.BFOF before writing 1 to XCR1.TCST.

Note 2. LIN communication requires a Break Delimiter (IDLE period) of 1 bit or more from the end of Break Field output until the next data transmission starts. For this reason, the Break Delimiter length is counted upon completion of Break

Field output. If transmit data is written while the Break Delimiter length is being counted, transmission does not start until the Break Delimiter length counting is completed. When transmit data is written after the Break Delimiter length has been counted, transmission starts at the same timing as normal data transmission.

Break Delimiter length count time after Break Field output:

1-bit to 2-bit length (CCR3.STP = 0)

2-bit to 3-bit length (CCR3.STP = 1)

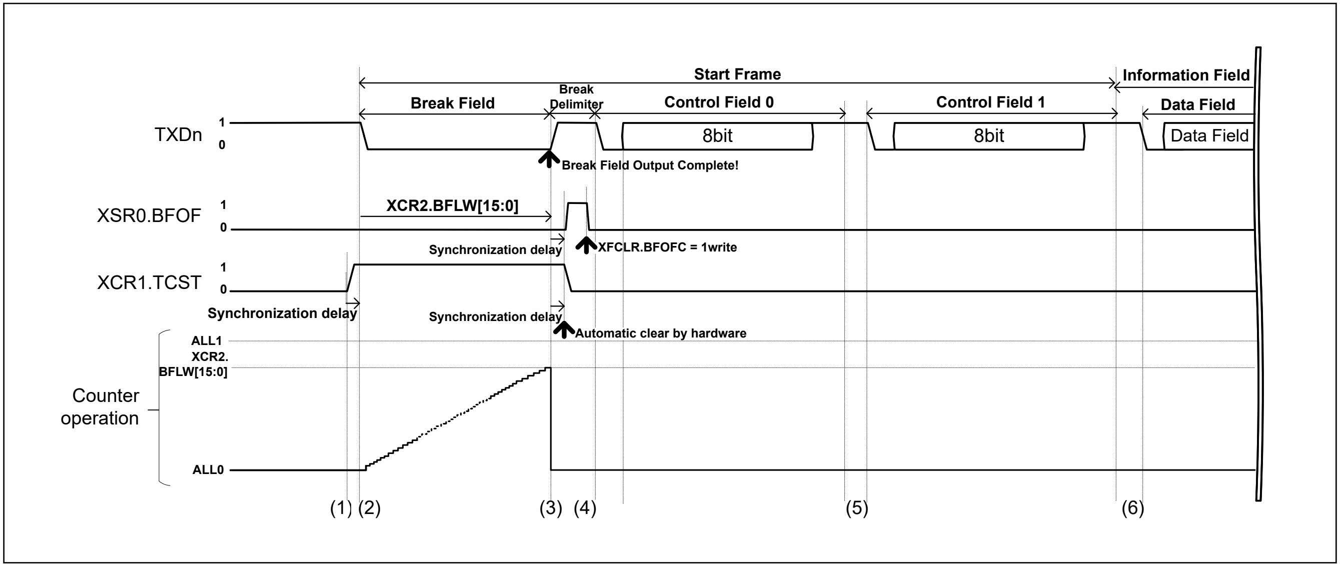


Figure 38.102 Start Frame Transmission Example

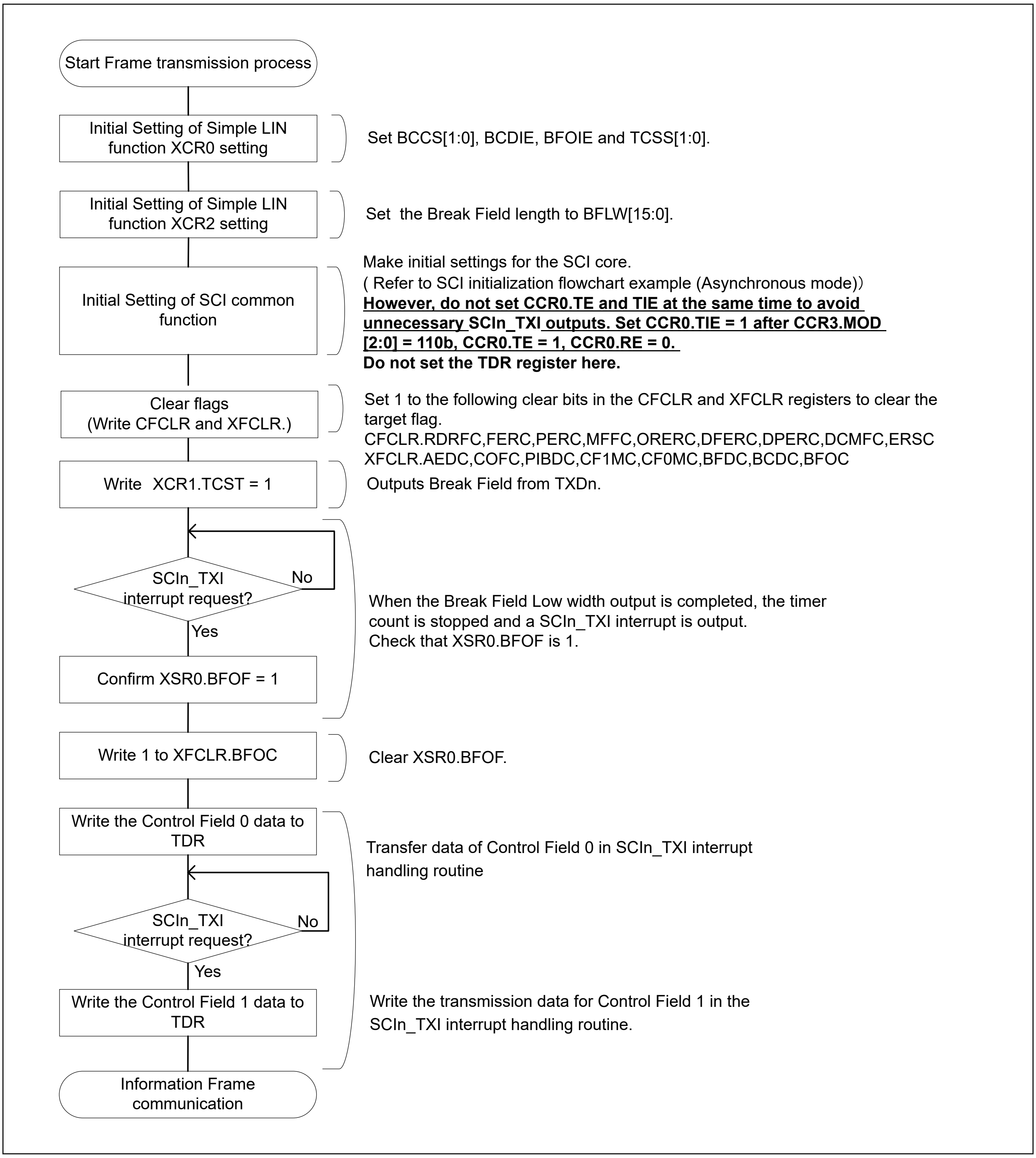


Figure 38.103 Start Frame Transmission Flowchart Example

38.11.2 Simple LIN Start Frame Reception

The SCI can detect Start Frames configured as shown in Figure 38.104.

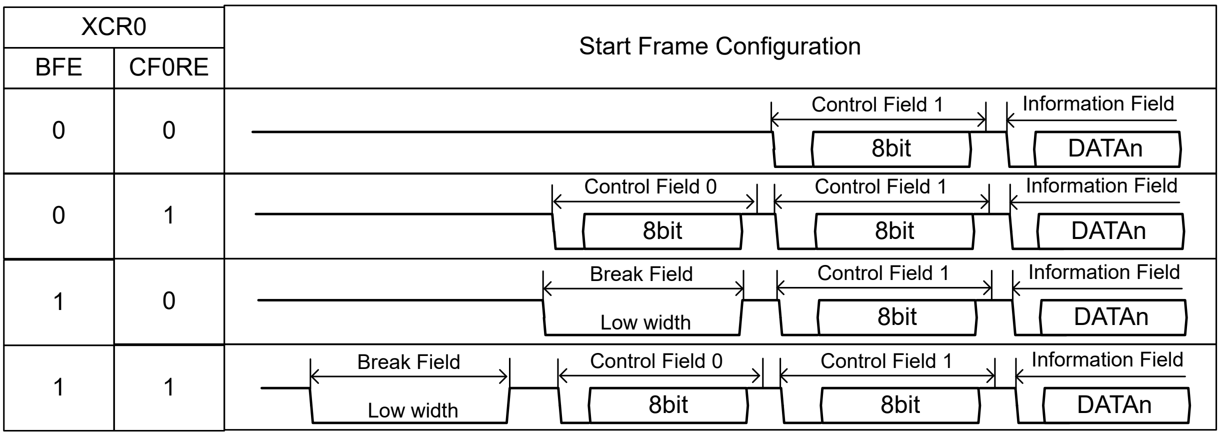


Figure 38.104 Start Frame Configuration

(1) Simple LIN normal reception of Start Frame (PIB not used)

Figure 38.105 shows an example of normal reception of the Start Frame consisting of a Break Field, Control Field0, and Control Field1. Figure 38.106 shows an example of reception to detect the Break Field during Control Field 1. Figure 38.107 shows a flowchart to receive the Start Frame, and Figure 38.108 shows a state transition diagram.

When receiving the Start Frame, the SCI operates as follows. Omit the processing of Break Field and Control Field0 according to the Start Frame configuration.

1. Writing 1 to XCR1.SDST makes it possible to detect the Start Frame. When XCR0.BFE = 1, RXDn input to the SCI core is disabled until the Break Field is detected (because XSR0.RXDSF is set to 1). Once the Break Field is detected, RXDn input can be received to the SCI core (XSR0.RXDSF = 0).
2. When a low level is input from the RXDn pin, the Break Field detection count starts. A timer count clock source can be selected by XCR0.TCSS[1:0].
3. When a low level that is equal to or longer than the period set in XCR2.BFLW[15:0] is input from the RXDn pin, it is determined as Break Field. At this time, XSR0.BFDF is set to 1. If XCR0.BFDIE has been set to 1 at this time, a SCIn_BFD interrupt is generated.
   The timer continues counting until the RXDn rising edge or counter overflow.
4. After the Break Field is detected, when the input level from the RXDn pin becomes high, the count value is captured to XSR1.TCNT[15:0] when BMEN = 0. At this time, XSR0.RXDSF is cleared to 0 and the SCI core starts receiving the RXDn input.
5. The SCI core starts receiving Control Field 0. Because the simple LIN continuously counts the edge interval, it determines a low level that is equal to or longer than the period set in XCR2.BFLW[15:0] as detection of the Break Field. When the Break Field is detected in the Control Field 0 phase, the SCI core waits for reception of Control Field 0 again (Figure 38.106).
6. When Control Field 0 has been received, an SCIn_RXI interrupt is generated and the Control Field 0 data is stored in XSR0.CF0RD[7:0]. When the received data matches the set XCR2.CF0D[7:0] value, XSR0.CF0MF is set to 1. If the received data differs from the set XCR2.CF0D[7:0] value, the SCI transitions to the state before the Break Field is detected.
7. The SCI core starts receiving Control Field 1. When BFE = 1, the Break Field detection function is continuously enabled while SDST = 1 as in the case of Control Field 0. When the Break Field is detected in the Control Field 1 phase, the SCI core waits for reception of Control Field 0 again.
8. When Control Field 1 has been received, an SCIn_RXI interrupt is generated and the Control Field 1 data is stored in XSR0.CF1RD[7:0]. When the received data matches the set XCR1.PCF1D[7:0] value or the set XCR1.SCF1D[7:0] value, XSR0.CF1MF is set to 1. If the received Control Field 1 data matches neither the set XCR1.PCF1D[7:0] value nor the set XCR1.SCF1D[7:0] value, the SCI transitions to the state before the Break Field is detected.
9. The SCI core performs Information Frame communication.
10. When communication is completed, write 0 to XCR1.SDST and 0 to CCR0.RE to stop reception.

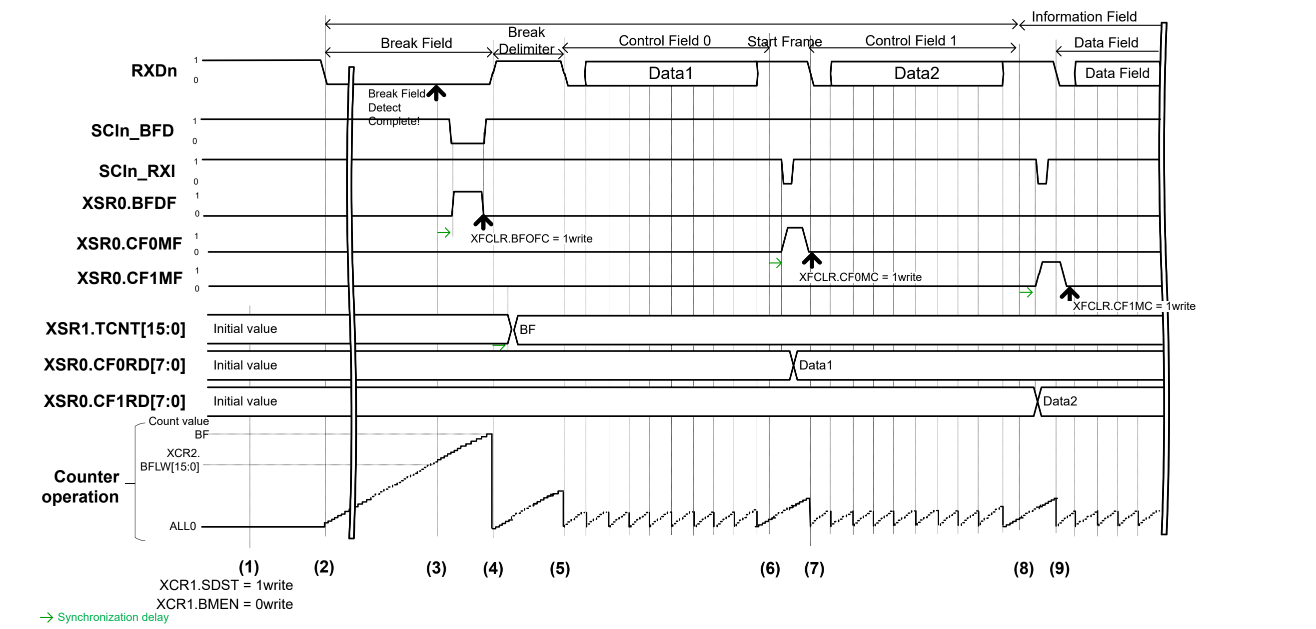


Figure 38.105 Normal Reception Example of Start Frame (PIB Not Used)

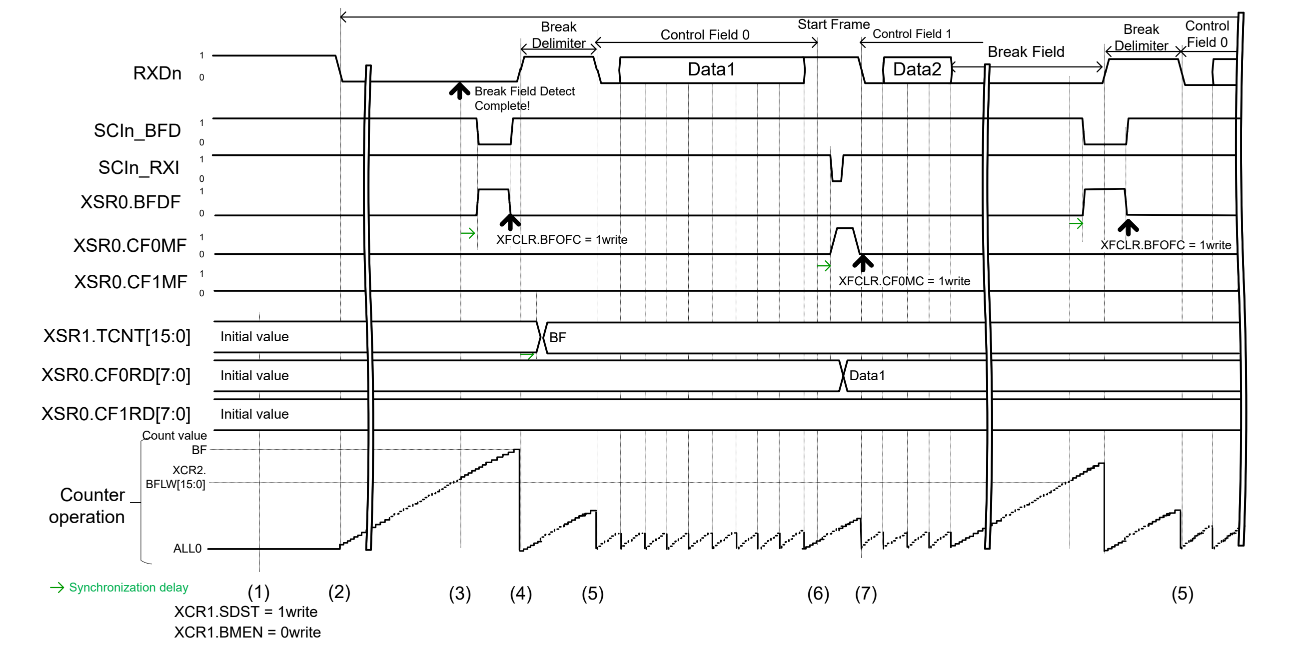


Figure 38.106 Start Frame Reception Example (PIB Not Used) Break Field Detected during Control Field 1

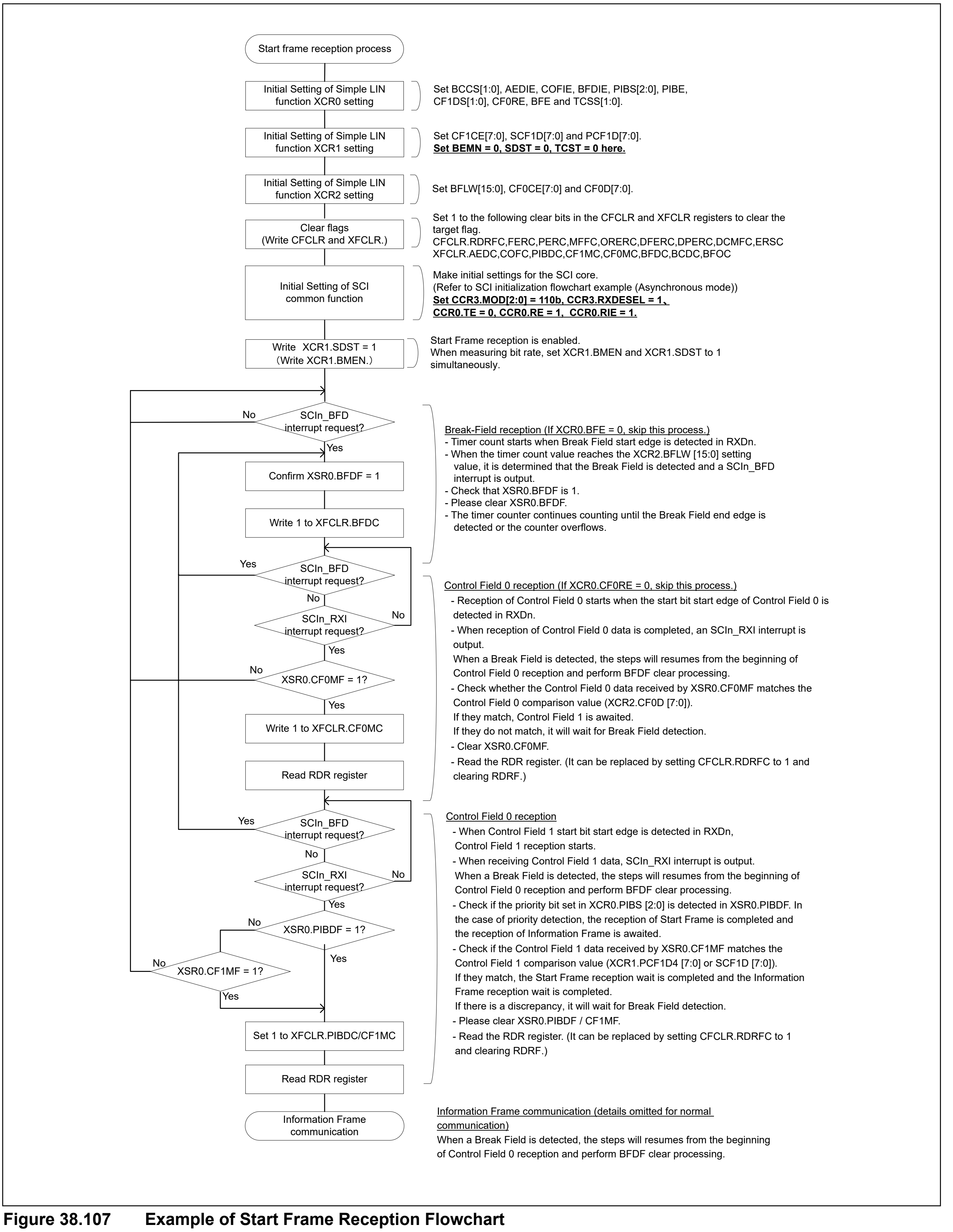


Figure 38.107 Example of Start Frame Reception Flowchart

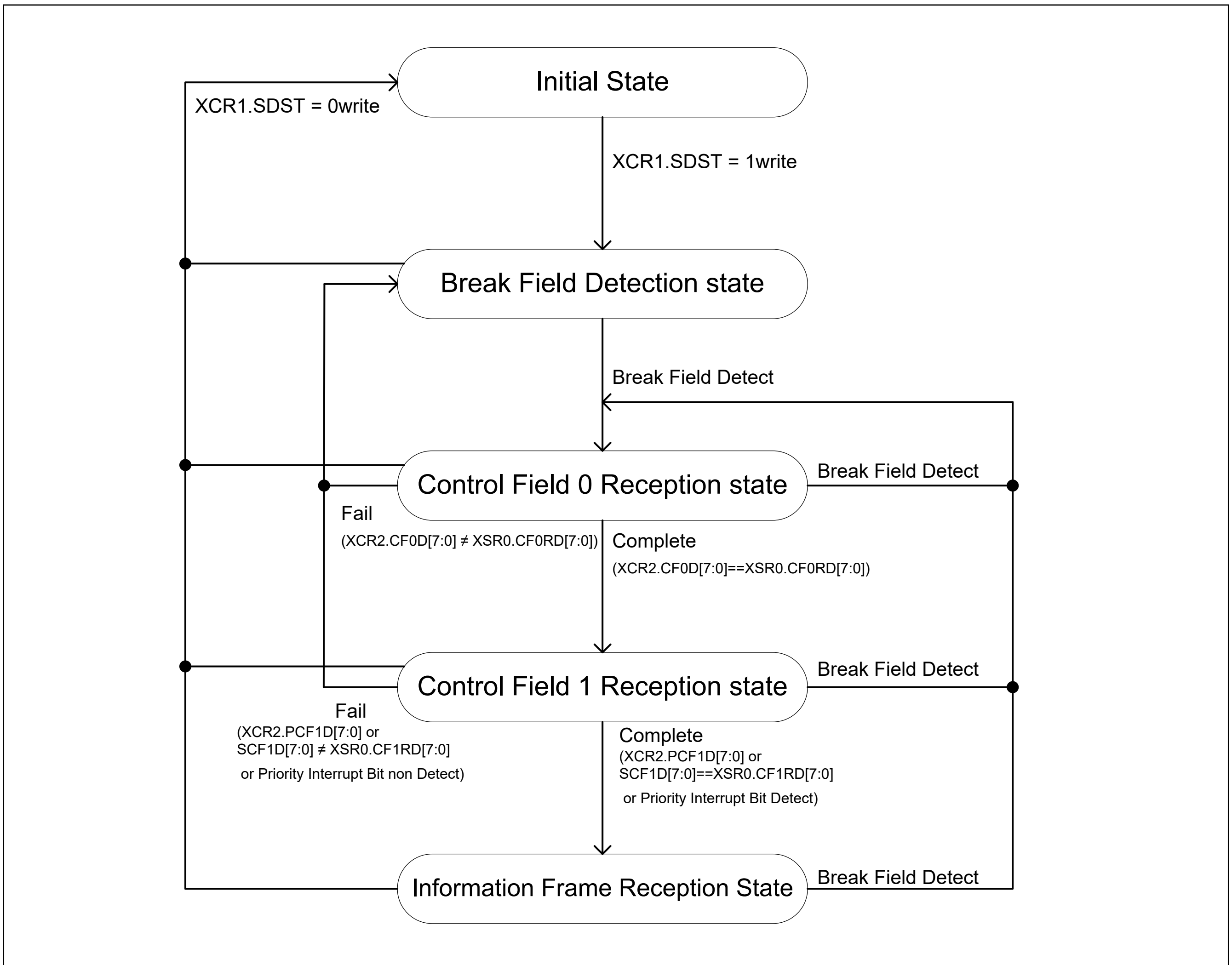


Figure 38.108 State Transition Diagram of Start Frame Reception

(2) Simple LIN Start Frame reception (using the priority interrupt bit)

Figure 38.109 shows an example of Start Frame reception using the priority interrupt bit. The priority interrupt bit is enabled by setting XCR0.PIBE to 1.

The SCI operates as follows during Start Frame reception using the priority interrupt bit.

Steps (1) to (7) are the same as steps (1) to (7) in the Start Frame reception example in Figure 38.105.

(8) When the value specified in the XCR0.PIBS[2:0] bits matches the set XCR1.PCF1D[7:0] value, XSR0.PIBDF is set to 1 and the SCI core performs communication of the Information Frame. If the data received in Control Field 1 matches neither the set XCR1.PCF1D[7:0] value nor the set XCR1.SCF1D[7:0] value and the priority interrupt bit is not detected, the SCI transitions to the state before the Break Field is detected.

(9) Communicate information frame at the SCI core.

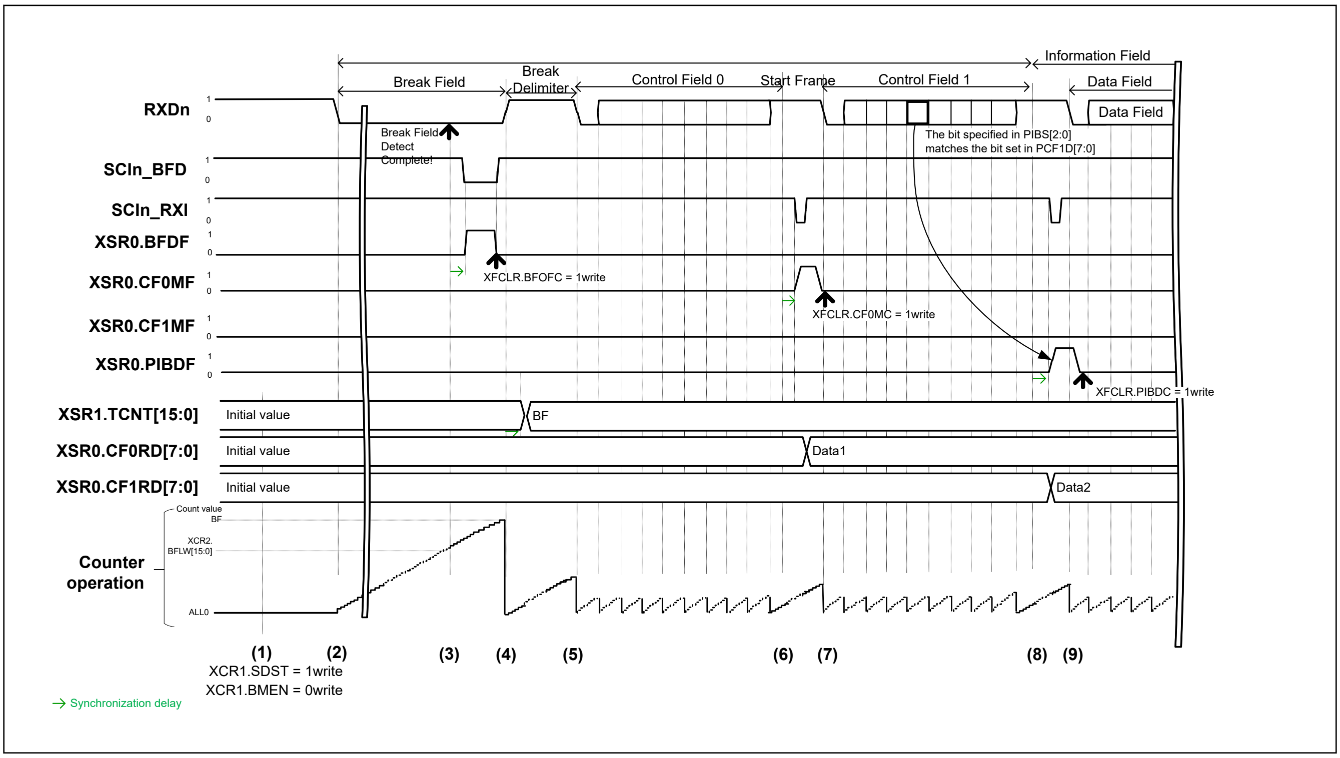


Figure 38.109 Start Frame Reception Example (Priority Interrupt Bit Used)

38.11.3 Simple LIN Bus Conflict Detection Function

In Simple LIN mode (CCR3MOD[2:0] = 110) when TE = 1, the bus conflict detection function works during Break Field output and during data transmission.

Figure 38.110 shows an operation example of the bus conflict detection function. The TXDn pin output and the RXDn pin input are sampled by the bus conflict detection clock set in XCR0.BCCS[1:0]. When a mismatch occurs three times in a row, XSR0.BCDF is set to 1, and if XCR0.BCDIE has been set to 1 at this time, an SCIn_ERI interrupt is generated.

When an SCIn_ERI interrupt is generated, stop transmission according to Figure 38.111. Check the bus state to decide whether to resume transmission.

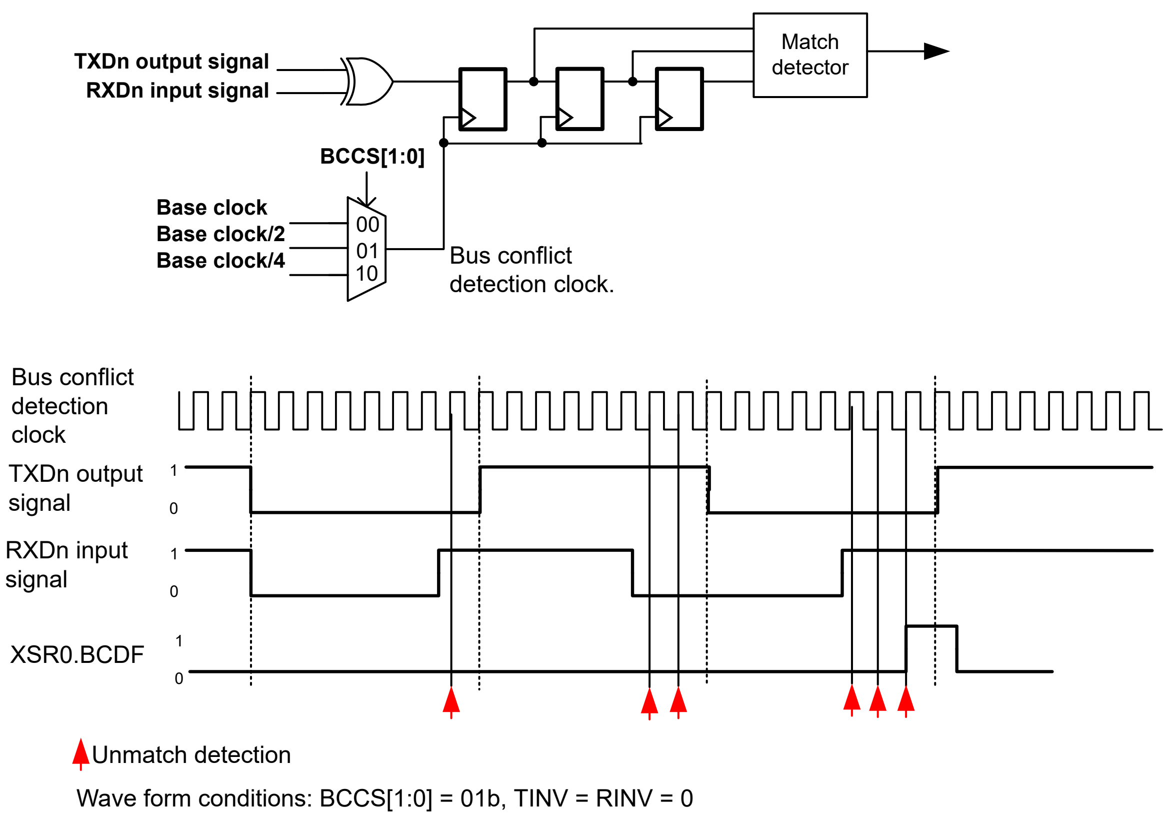


Figure 38.110 Operation Example of the Bus Conflict Detection Function

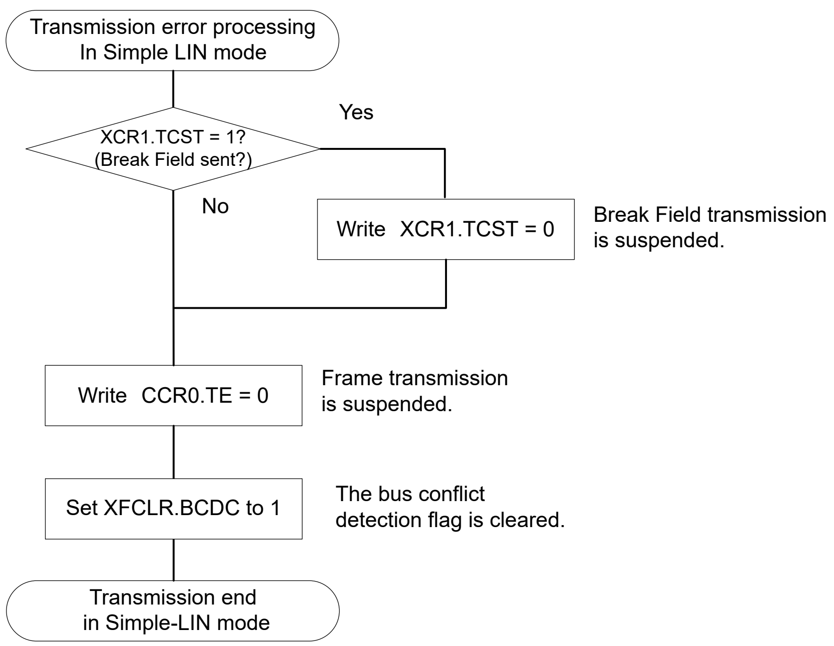


Figure 38.111 SCIn_ERI Interrupt handling flow at transmission in Simple LIN mode

38.11.4 Simple LIN Bit Rate Measurement Function

This function measures a bit rate between the effective edges of the input signal from the RXDn pin. Figure 38.112 shows an operation example of the bit rate measurement function.

1. Writing 1 to XCR1.SDST and XCR1.BMEN enables bit rate measurement. When this bit is set to 1, the valid edge interval of Control Field 0 and Control Field 1 data is measured. However, bit rate is not measured between the Break Field and the Break Delimiter. Set XCR1.BMEN and XCR1.SDST to 1 simultaneously, only when measuring bit rate.
2. Because bit rate is not measured in the Break Field, the effective edge detection flag is not set to 1 at the rising edge at the end of the Break Field, and the counter capture value is not stored in XSR1.TCNT[15:0].
3. The counter starts counting from the falling edge of the start bit in Control Field 0. The Break Delimiter count value is not captured in XSR1.TCNT[15:0].
4. The rising edge of the start bit is detected as an effective edge, and then the XSR0.AEDF flag is set to 1. If XCR0.AEDIE has been set to 1 at this time, an SCIn_AED interrupt is output. The start bit count value is stored in XSR1.TCNT[15:0]. The XSR1.TCNT[15:0] value is retained until the effective capture value is read.
5. Even if an effective edge is input from the RXDn input pin, the count value of this effective edge timing is not captured because the XRS1.TCNT[15:0] value has not been read and retention has not been released. In this case, an SCIn_AED interrupt is not output.
6. The XSR1.TCNT[15:0] value is read. Then the retention of XSR1.TCNT[15:0] is released and the XSR0.AEDF flag is cleared by hardware.
7. Because the retention of XSR1.TCNT[15:0] has been released, the count value is captured at the effective edge and is retained. At the same time, the XSR0.AEDF flag is set to 1, and if XCR0.AEDIE has been set to 1, an SCIn_AED interrupt is output. The bit rate can be adjusted by calculating it from the count value between effective edges by software and by changing the SCI settings.
8. To disable bit rate measurement, write 0 to XCR1.BMEN.
9. The XSR0.AEDF value and the XSR1.TCNT[15:0] value remain unchanged at the effective edge timing because the bit rate measurement function is disabled.

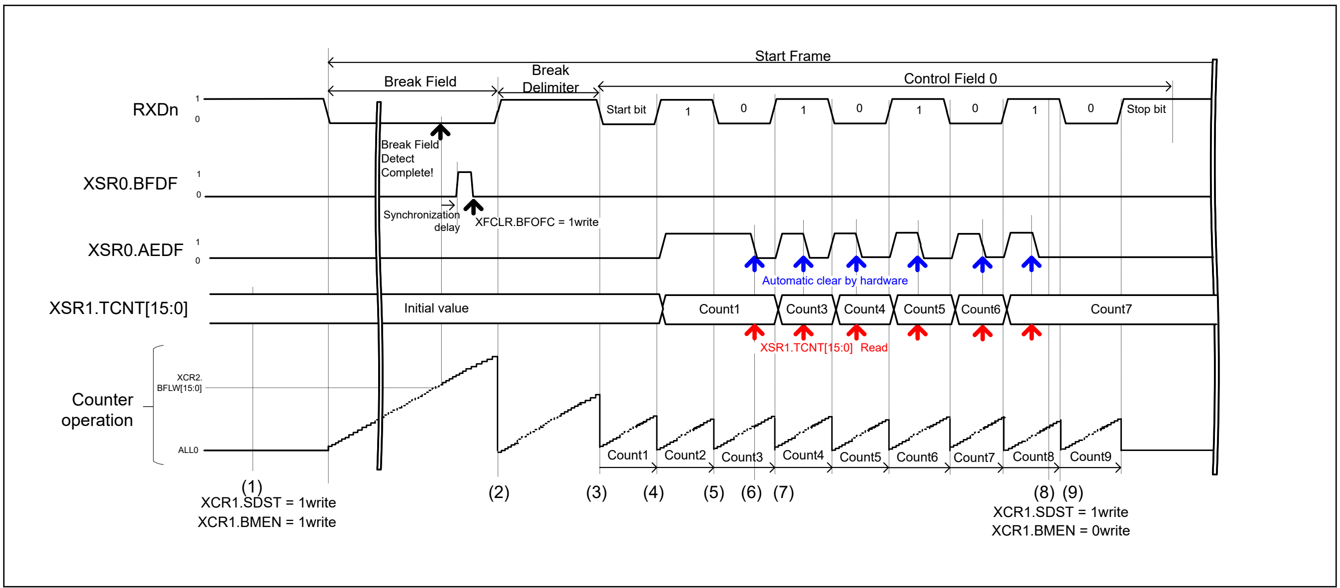


Figure 38.112 Operation Example of the Bit Rate Measurement Function

38.12 Interrupt Sources

38.12.1 Buffer Operation for SCIn_TXI and SCIn_RXI Interrupts

If the conditions for an SCIn_TXI and SCIn_RXI interrupt are satisfied while the interrupt status flag in the ICU is 1, the ICU does not output the interrupt request but saves it internally with a capacity for retention of one request per source.

38.12.2 Interrupts in Asynchronous, Manchester, Clock Synchronous, and Simple SPI Modes

(1) Non-FIFO selected

Table 38.42 lists interrupt sources in Asynchronous mode, Manchester mode, clock synchronous mode, and simple SPI mode.

A different interrupt vector can be assigned to each interrupt source. Individual interrupt sources can be enabled or disabled with the enable bits in the CCR0 register.

If the CCR0.TIE bit is 1, an SCIn_TXI interrupt request is generated when transmit data is transferred from the TDR register to the TSR register. An SCIn_TXI interrupt request can also be generated by using a single instruction to set the CCR0.TE and CCR0.TIE bits to 1 at the same time. An SCIn_TXI interrupt request can activate the DTC or DMAC to handle data transfer.

An SCIn_TXI interrupt request is not generated by setting the CCR0.TE bit to 1 when CCR0.TIE is 0 or by setting the CCR0.TIE bit to 1 when the CCR0.TE is 1.*1

When new data is not written by the time of transmission of the last bit of the current transmit data and CCR0.TEIE is 1, the CSR.TEND flag is set to 1 and an SCIn_TEI interrupt request is generated. Additionally, when CCR0.TE is 1, the CSR.TEND flag retains the value 1 until more transmit data is written to the TDR register, and setting CCR0.TEIE to 1 leads to the generation of an SCIn_TEI interrupt request.

Writing data to the TDR register leads to clearing of the CSR.TEND flag and, after a certain time, discarding of the SCIn_TEI interrupt request.

If the CCR0.RIE bit is 1, an SCIn_RXI interrupt request is generated when received data is stored in the RDR register. An SCIn_RXI interrupt request can activate the DTC or DMAC to handle data transfer.

Setting any of the CSR.ORER, FER, PER or MSR.MER*2, SYER*2, PFER*2, and SBER*2 flags to 1 when the CCR0.RIE bit is 1 leads to the generation of an SCIn_ERI interrupt request.

An SCIn_RXI interrupt request is not generated in this case. Clearing all these flags (ORER, FER, PER, MER*2, SYER*2, PFER*2 and SBER*2) leads to discarding of the SCIn_ERI interrupt request.

Note 1. To temporarily prohibit SCIn_TXI interrupts on transmission of the last of the data when a new round of transmission is to be started, after handling the transmission-completed interrupt, control activation of the interrupt by using the interrupt request enable bit in the ICU rather than using the CCR0.TIE bit. This approach can prevent the suppression of SCIn_TXI interrupt requests in the transfer of new data.

Note 2. MER, SYER, PFER, and SBER work as a factor of SCIn_ERI interrupt only in Manchester mode. SYER, PFER, and SBER also only work if its enable bits (SYEREN, PFEREN, SBEREN in MECR) are set to "1".

(2) FIFO selected

Table 38.43 lists interrupt sources in FIFO selected mode.

If the CCR0.TIE bit is 1, an SCIn_TXI interrupt request is generated when the stored amount of data in the transmit-FIFO (TDR) register becomes the threshold value indicated in FCR.TTRG or below. An SCIn_TXI interrupt request can also be generated by using a single instruction to set the CCR0.TIE and CCR0.TE bits to 1 simultaneously or by setting CCR0.TIE to 1 when CCR0.TE is 1.

An SCIn_TXI interrupt request is not generated by setting CCR0.TE to 1 when CCR0.TIE is 0 or by setting the CCR0.TIE bit to 1 while the setting of the CCR0.TE bit is 1.

If CCR0.TEIE is 1 and if the next data is not written to the transmit-FIFO (TDR) register by the time the last bit of the transmit data is sent, the CSR.TEND flag is set to 1 and the SCIn_TEI interrupt request is generated.

If CCR0.RIE is 1, the SCIn_RXI interrupt request is generated when the stored amount of data in the transmit-FIFO (TDR) register is equal to or greater than the threshold value indicated in FCR.RTRG. When RTRG is 0, an SCIn_RXI interrupt does not occur even when the amount of data in the receive FIFO is equal to 0.

If the CCR0.RIE bit is 1, when the CSR.ORER flag is set to 1 or data with a framing error or a parity error is stored in the transmit-FIFO (TDR) register, the SCIn_ERI interrupt request is generated. When the amount of data stored in the transmit-FIFO (TDR) register is at the threshold value or above, the SCIn_RXI interrupt request is also generated. The SCIn_ERI interrupt request can be canceled, in which case CSR.ORER, FER, and PER flags are all cleared.

Table 38.42 SCI interrupt sources with non-FIFO selected

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC or DMAC activation
SCIn_ERI (n = 0 to 9)	Receive error	CSR.ORER, CSR.FER, CSR.PER, CSR.DFER, CSR.DPER, (MSR.MER, MSR.SYER, MSR.PFER, MSR.SBER)*1	CCR0.RIE	Not possible
SCIn_RXI (n = 0 to 9)	Receive data full	CSR.RDRF	CCR0.RIE	Possible
	Address match	CSR.DCMF	CCR0.RIE	Possible
SCIn_TXI (n = 0 to 9)	Transmit data empty	CSR.TDRE	CCR0.TIE	Possible
	TE = 0->1 detection			
SCIn_TEI (n = 0 to 9)	Transmit end	CSR.TEND	CCR0.TEIE	Not possible

Note: Only SCI0 supports Manchester mode.
Note 1. MER, SYER, PFER, and SBER work as a factor of SCIn_ERI interrupt only in Manchester mode. SYER, PFER, and SBER also only work if its enable bits (SYEREN, PFEREN, SBEREN in MCR) are set to 1.

Table 38.43 SCI interrupt sources with FIFO selected

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC or DMAC activation
SCIn_ERI (n = 0 to 9)	Receive error	CSR.ORER, CSR.FER, CSR.PER, CSR.DFER, CSR.DPER	CCR0.RIE	Not possible
		FRSR.DR (when FCR.DRES = 1)	CCR0.RIE	Not possible
SCIn_RXI (n = 0 to 9)	Receive data full	CSR.RDRF	CCR0.RIE	Possible
	Receive data ready	FRSR.DR (when FCR.DRES = 0)	CCR0.RIE	Possible
	Address match	CSR.DCMF	CCR0.RIE	Possible
SCIn_TXI (n = 0 to 9)	Transmit data empty	CSR.TDFE	CCR0.TIE	Possible
	TE=0->1 detection			
SCIn_TEI (n = 0 to 9)	Transmit end	CSR.TEND	CCR0.TEIE	Not possible

Note: Only SCI0 supports Manchester mode.

38.12.3 Interrupts in Smart Card Interface Mode

Table 38.44 lists interrupt sources in smart card interface mode. A transmit end interrupt (SCIn_TEI) request and an address match (SCIn_AM) request cannot be used in this mode.

Table 38.44 SCI Interrupt sources in Smart Card Interface Mode

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC or DMAC activation
SCIn_ERI (n = 0 to 9)	Receive error or error signal detection	CSR.ORER, CSR.PER, CSR.ERS	CCR0.RIE	Not possible
SCIn_RXI (n = 0 to 9)	Receive data full	CSR.RDRF	CCR0.RIE	Possible
SCIn_TXI (n = 0 to 9)	Transmit data empty	CSR.TEND	CCR0.TIE	Possible
	When set TE = 0->1			

Data transmission or reception using the DTC or DMAC is also possible in smart card interface mode, similar to normal SCI mode. In transmission, when the CCR0.TEND flag is set to 1, an SCIn_TXI interrupt request is generated. This SCIn_TXI interrupt request activates the DTC or DMAC, allowing transfer of transmit data if the SCIn_TXI request is previously specified as a source of DTC or DMAC activation. The TEND flag is automatically set to 0 when the DTC or DMAC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During the retransmission, the TEND flag is kept at 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC automatically transmit the specified number of bytes, including retransmission after an error occurrence. However, the CSR.ERS flag is not automatically set to 0 at error occurrence. Therefore, the ERS flag must be cleared by previously setting the CCR0.RIE bit to 1 to enable an SCIn_ERI interrupt request to be generated at error occurrence.

When transmitting or receiving data using the DTC or DMAC, always enable the DTC or DMAC before making the SCI settings. For DTC or DMAC settings, see section 18, Data Transfer Controller (DTC), section 17, DMA Controller (DMAC).

In reception, an SCIn_RXI interrupt request is generated when receive data is set to the RDR register. This SCIn_RXI interrupt request activates the DTC or DMAC, allowing transfer of the receive data if the SCIn_RXI request is previously specified as a source of DTC or DMAC activation. If an error occurs, the error flag is set. Therefore, the DTC or DMAC is not activated and an SCIn_ERI interrupt request is issued to the CPU instead. The error flag must be cleared.

38.12.4 Interrupts in Simple IIC Mode

Table 38.45 lists the interrupt sources in simple IIC mode. The STI interrupt is allocated to the transmit end interrupt (SCIn_TEI) request. The receive error interrupt (SCIn_ERI) and the address match (SCIn_AM) request cannot be used.

The DTC or DMAC can also be used to handle transfer in simple IIC mode.

When the ICR.IICINTM bit is 1:

- An SCIn_RXI request is generated on the falling edge of the SCLn signal for the 8th bit. If SCIn_RXI is previously set up as an activation source for the DTC or DMAC, the SCIn_RXI request activates the DTC or DMAC to handle transfer of the received data.
- An SCIn_TXI request is generated on the falling edge of the SCLn signal for the 9th bit (acknowledge bit). If SCIn_TXI is previously set up as an activation source for the DTC or DMAC, the SCIn_TXI request activates the DTC or DMAC to handle transfer of the transmit data.

When the ICR.IICINTM bit is 0:

- An SCIn_RXI request (ACK detection) is generated if the input on the SDAn pin is low on the rising edge of the SCLn signal for the 9th bit (acknowledge bit)
- An SCIn_TXI request (NACK detection) is generated if the input on the SDAn pin is high on the rising edge of the SCLn signal for the 9th bit (acknowledge bit)
- If SCIn_RXI is previously set up as an activation source for the DTC or DMAC, the SCIn_RXI request activates the DTC or DMAC to handle transfer of the received data.

If the DTC or DMAC is used for data transfer in reception or transmission, always set up and enable the DTC or DMAC before setting up the SCI.

When the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits in ICR are used to generate a start condition, restart condition, or stop condition, the STI request is issued when generation is complete.

Table 38.45 SCI interrupt sources in Simple IIC Mode

Name	Interrupt source		Interrupt flag	Interrupt enable	DTC or DMAC activation
	ICR.IICINTM = 1	ICR.IICINTM = 0			
SCIn_RXI (n = 0 to 9)	Reception end	—	—	CCR0.RIE	Possible*1
	—	ACK detection	—		Possible
SCIn_TXI (n = 0 to 9)	Transmission end	—	—	CCR0.TIE	Possible*1
	—	NACK detection	—		Possible
SCIn_TEI(STIn) (n = 0 to 9)	Completion of generation of a start, restart, or stop condition		ICR.IICSTIF	CCR0.TEIE	Not possible

Note 1. If the DMAC or DTC are being used, you cannot confirm whether ACK or NACK.

38.12.5 Interrupts in Simple LIN Mode

Table 38.46 lists interrupt sources in Simple LIN mode.

Table 38.46 SCI interrupt sources in Simple LIN mode

Name	Interrupt Sources	Interrupt Flag	Flag the needs to be confirmed	Interrupt Enable	DTC/DMAC Activation
SCIn_ERI	Receive error	CSR.ORER, CSR.FER, CSR.PER	—	CCR0.RIE	Not Possible
		XSR0.BCDF		XCR0.BCDIE	
		XSR0.COF		CCR0.RIE, XCR0.COFIE	
SCIn_RXI	Receive data full flag	CSR.RDRF	XSR0.CF0MF XSR0.CF1MF XSR0.PIBDF	CCR0.RIE	XSR0.SFSF = 0: Possible XSR0.SFSF = 1: Not Possible
SCIn_AED	Active edge detection	XSR0.AEDF	—	XCR0.AEDIE	Possible
SCIn_TXI	Transmit data empty interrupt	CSR.TDRE	—	CCR0.TIE	Possible
	When set TE = 0->1				
	Break Field output completion	XSR0.BFOF		CCR0.TIE, XCR0.BFOIE	
SCIn_TEI	Transmit end	CSR.TEND	—	CCR0.TEIE	Not Possible
SCIn_BFD	Break Field Detection	XSR0.BFDF	—	XCR0.BFDIE	Not Possible (Unnecessary)

In Simple LIN mode, in addition to reception errors (ORER, FER, PER), an SCIn_ERI interrupt request is output when a bus conflict is detected during transmission, or when a counter overflow of the Simple LIN module occurs. At this time, a SCIn_RXI interrupt request is not output. The SCIn_ERI interrupt request can be canceled by clearing all the flags.

When transmitting Start Frame, if CCR0.TIE = 1 and XCR0.BFOIE = 1, a SCIn_TXI interrupt request is output when Break Field transmission is completed. When Control Field 0 data is written to the TDR register, data transmission starts. Therefore, transmission using DTC or DMAC is possible.

Set CCR0.TEIE = 1 after writing the last transmit data to the TDR register and transmission starts.

During Start Frame reception (XSR0.SFSF = 1), reception using DTC or DMAC by SCIn_RXI interrupt is not possible. Check the CSR register and XSR0 register, check the reception status (See Figure 38.83), and then clear the flag. Also read the RDR register (if you do not need to check the received data value, clear the RDRF flag without reading the RDR register). When reception of Control Field 1 is completed (XSR0.CF1MF = 1), Start Frame detection is disabled (XSR0.SFSF = 0) and reception using DTC or DMAC is possible. Be sure to read the RDR register.

When Start Frame / Break Field detection is enabled (XCR1.SDST = 1), if a Break Field longer than the period set in XCR2.BFLW [15:0] is received, the BFDF flag is set and a SCIn_BFD interrupt request is output. Then SCI becomes the Start Frame reception state. Clear the BFDF flag.

When Start Frame / Break Field detection is enabled (XCR1.SDST = 1) and the bit rate measurement function is enabled (XCR1.BMEN = 1), an SCIn_AED interrupt factor is output when an active edge is detected. Read the timer count capture value (XSR1.TCNT [15:0]).

38.13 Event Linking

By using interrupt request signals as event signals, the SCIn can provide linked operation through the ELC for modules selected in advance.

Event signals can be output regardless of the values of the associated interrupt request enable bits.

(1) Error event output (receive error or error signal detected) (SCIn_ERI, n = 0 to 9)

- Indicates abnormal termination because of a parity error during reception in Asynchronous mode
- Indicates abnormal termination because of a framing error during reception in Asynchronous mode
- Indicates abnormal termination because of an overrun error during reception
- Indicates abnormal termination due to a Manchester error during reception (Only in Manchester mode).

- Indicates that a preface error occurred upon reception and abnormal termination occurred (only in Manchester mode and MCR.PFEREN = 1).
- Indicates that a start bit error occurred during reception and abnormal termination occurred (only in Manchester mode and MCR. SBEREN = 1).
- Indicates that a reception sync error occurred during reception and abnormal termination occurred (only in Manchester mode and only when MCR.SYEREN = 1).
- Indicates detection of the error signal during transmission in smart card interface mode
- The CSR.FER and PER flags are 0, and receive data less than the receive FIFO data trigger number is set in a reception FIFO buffer, and it indicates that 15 etu elapse when FIFO is selected and the FCR.DRES bit is 1
- In Simple LIN mode, indicates that the 16-bit counter in the Simple LIN module has overflowed.
- In Simple LIN mode, a bus collision is detected during transmission (CCR0.TE = 1).

(2) Receive data full event output (SCIn_RXI, n = 0 to 9)

- Indicates that ACK is detected if the ICR.IICINTM bit is 0 in simple IIC mode
- Indicates that the 8th-bit SCLn falling edge is detected if the ICR.IICINTM bit is 1 in simple IIC mode
- When the ICR.IICINTM bit is 1 during master transmission in simple IIC mode, set the ELC so that receive data full events are not used

Non-FIFO selected

- Indicates that received data is set in the Receive Data Register (RDR).

FIFO selected

- Using this event output is prohibited.

(3) Transmit data empty event output (SCIn_TXI, n = 0 to 9)

- Indicates that the CCR0.TE bit is changed from 0 to 1
- Indicates that transmission is complete in smart card interface mode
- Indicates that NACK is detected if the ICR.IICINTM bit is 0 in simple IIC mode
- Indicates that the 9th-bit SCLn falling edge is detected if the ICR.IICINTM bit is 1 in simple IIC mode
- In Simple LIN mode, indicates that Break Field output is complete.

Non-FIFO selected

- Indicates that transmit data is transferred from the Transmit Data Register (TDR) to the Transmit Shift Register (TSR).

FIFO selected

- Using this event output is prohibited.

(4) Transmit end event output (SCIn_TEI, n = 0 to 9)

- Indicates the completion of transmission
- Indicates that the starting condition, resumption condition, or termination condition is generated in simple IIC mode
- In Smart Card mode, the transmit end event is not output.

Note: When FIFO is selected, using this event output is prohibited

(5) Address match event output (SCIn_AM, n = 0 to 9)

- Indicates a match of the comparison data (CCR4.CMPD) with one frame of receive data when CCR0.DCME is set to 1 in Asynchronous mode, including multi-processor mode.

(6) Active edge detection event output (SCIn_AED, n= 0, 1)

- In Simple LIN mode, when CCR1.BMEN is 1, it indicates that a valid edge has been detected in the RXD input signal.

38.14 Noise Cancellation Function

Figure 38.113 shows the configuration of the noise filter used for noise cancellation. The noise filter consists of a 2-stage flip-flop circuit and a match detection circuit. When the input signals of the noise filter and the output signals of the 2-stage flip-flop circuits completely match, the matched level is conveyed as an internal signal. Unless otherwise matched, the previous value is retained. When the same level is retained for 3 cycles or longer on the sampling clock of the noise filter, it is considered as a valid receive signal. A change in pulse for 3 cycles or shorter is considered as noise, not as a receive signal. (When CCR1.NFM = 0)

When CCR2.ABCSE2 = 1 in asynchronous mode, the above noise cancellation cannot be used because one bit has only 4 base clocks. Therefore, only in this case, set the noise cancellation mode to transmit the majority value internally by a majority vote of 3 sampling values instead of matching 3 consecutive sampling values. (CCR1.NFM = 1)

In Asynchronous mode, Manchester and Simple LIN modes, the noise cancellation function can be applied to the receive signal input to the RXDn pin. The sampling period of the noise filter can be selected from the base clock period and the divided clock of the baud rate generator clock source by CCR1.NFCS[2:0].

- When CCR1.NFCS[2:0] = 000b, CCR2.ABCS = 0 and CCR2.ABCSE = 0, the cycle is 1/16 of a 1-bit period.
- When CCR1.NFCS[2:0] = 000b, CCR2.ABCS = 1 and CCR2.ABCSE = 0, the cycle is 1/8 of a 1-bit period.
- When CCR1.NFCS[2:0] = 000b, CCR2.ABCSE = 1, the cycle is 1/6 of a 1-bit period.
- When CCR1.NFCS[2:0] = 000b, CCR2.ABCSE2 = 1, the cycle is 1/4 of a period 1 transfer bit.
- For the CCR1.NFCS value, select “000b” or “001b” if ABCSE = 1 or ABCSE2 = 1. At least, the sampling period of Noise Filter should not be greater than the base clock due to the CCR2.BRR setting value.

In simple IIC mode, this function can be used for each input on SDAn and SCLn. The sampling clock is selected from divided clock of baud rate generator settings by CCR1.NFCS[2:0].

If the base clock is stopped once with the noise filter enabled and then the base clock input is restarted again, the noise filter operation resumes from the state where the clock was stopped. When CCR0.TE and CCR0.RE are set to 0 during base clock input, all of the noise filter flip-flop values are initialized to 1. Accordingly, if the input data is 1 when reception operation resumes, the function determines that a level match is detected and the result is conveyed as an internal signal. If the input data is 0, the Noise Filter output holds its initial value until all three sampled pin levels match (CCR1.NFM = 0) or two or more match (CCR1.NFM = 1).

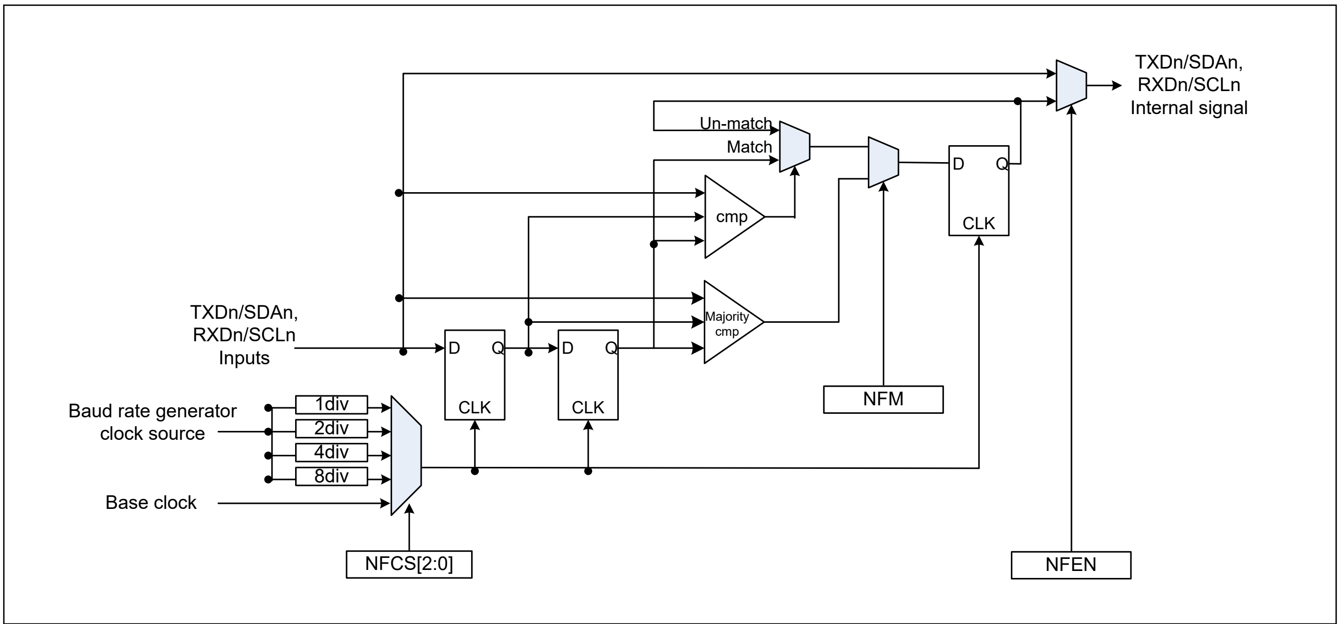


Figure 38.113 Digital noise filter circuit block diagram

38.15 RS-485 Driver Control Function

Setting the DEN bit in the SCI common control register3 (CCR3) to 1 enables the RS-485 driver control function and generates a DEn (Driver Enable) signal that enables the external transceiver transmission mode. The DEn signal outputs a valid level for the period with driver assertion time and driver negate time added before and after data transmission. The DEn signal valid level is set by the DEPOL bit in the driver control register (DCR).

The driver assertion time is the time from when the DEn signal is valid until the start bit starts. Set by DEAST [4:0] of driver control register (DCR).

The driver negate time is the time from the end of the last stop bit of the transmitted message to the invalidation of the DEn signal. Set with DENGT [4:0] of the driver control register (DCR).

DEAST and DENGT are expressed in base clock period (1/4, 1/6, 1/8, or 1/16 bit period, see Table 38.6). For details, see section 38.2.13. DCR : Driver Control Register.

When this function is used (CCR3.DEN = 1), the CSR.TEND set timing and SCIn_TEI interrupt output timing are at the end of the driver negation time.

When transmission is completed and the next transmission data is not written before the DEn signal is negated, the DEn signal is negated once. If the timing for writing the next transmit data is not in time, assert the DEn signal after negating it again, insert the driver assertion time, and transmit the next data. If you want to perform the next transmission with the DEn signal asserted, write the next transmission data to the TDR quickly enough in consideration of the synchronization delay time of the register.

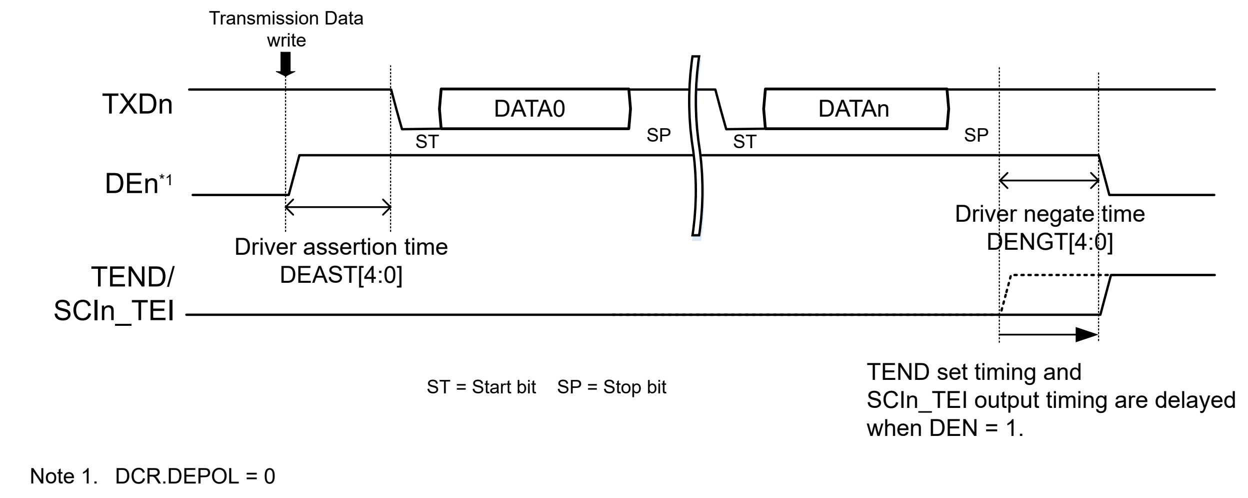


Figure 38.114 The image waveform for RS-485 driver control DE signal output

38.16 Loopback Function

The loopback function can be used in Asynchronous mode with the internal clock, and Manchester mode with the internal clock, and Clock synchronous mode with the internal clock.

When 1 is written to the SPLP bit in the CCR1 register, SCI blocks the external input (RXDn) path and connects the output path of the transmit data register and the input path of the receive data register.

When this function is used with TINV bit = 1, inversion of transmission data becomes reception data. However, this function can be used with TINV = 1 only when operating in clock synchronous mode internal clock.

Table 38.47 shows the relationship between the TINV and SPLP bit settings and the received data.

Table 38.47 TINV and SPLP bit settings and received data

CCR1.TINV	CCR1.SPLP	Receive Data	Communication mode		
			Asynchronous	Manchester	Clock synchronous
			internal clock	internal clock	internal clock
—	0	Receive Data from RXDn pin	Possible	Possible	Possible
0	1	Transmit Data	Possible	Possible	Possible
1	1	Inverted transmit data	Impossible	Impossible	Possible

Note: —: don't care

Figure 38.115 shows the configuration of the shift register input / output path in loopback mode.

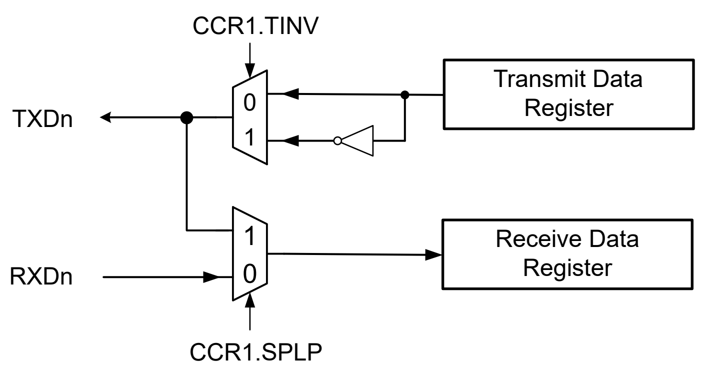


Figure 38.115 Shift register input output configuration image in loopback mode

38.17 Half-Duplex Communication Function

Do not use the half-duplex communication function in Simple IIC, Simple SPI and Smart Card Interface modes.

In other communication modes, if the CCR1.SHARPS bit is set to 1, half-duplex communication using the TXDn pin is possible. When half-duplex communication is used, transmission and reception must be performed exclusively. Transmission and reception settings (CCR0.TE = 1 and CCR0.RE = 1) is prohibited.

However, if half-duplex communication is performed as the master reception in clock synchronous mode, perform transmission / reception settings (CCR0.TE = 1 and CCR0.RE = 1) and perform dummy transmission. By dummy transmission (arbitrary transmission data is written to TDR), SCKn is output and reception is enabled. The dummy transmission data is discarded inside the IP and is not actually transmitted.

During half-duplex communication, only the TXDn pin is used as the communication pin. Output when CCR0.TE = 1, input when CCR0.TE = 0.

38.18 Synchronizer Bypass Function

The SCI has a bus clock (PCLK) and the operation clock (TCLK). And these have each operating circuit. Therefore, there is a synchronization circuit for signal transfer between different clocks, and synchronization delay time is required for signal propagation between different clocks.

However, the synchronization circuit can be bypassed by the CCR3.BPEN bit only when the same clock is input to the bus clock and the operation clock. In this case, eliminates synchronization delay time and improves responsiveness. Figure 38.116 shows the image waveform of the bypass function.

The SCI also has a synchronization circuit between the communication clock (SCKn) and the operation clock (TCLK), but this synchronization circuit cannot be bypassed.

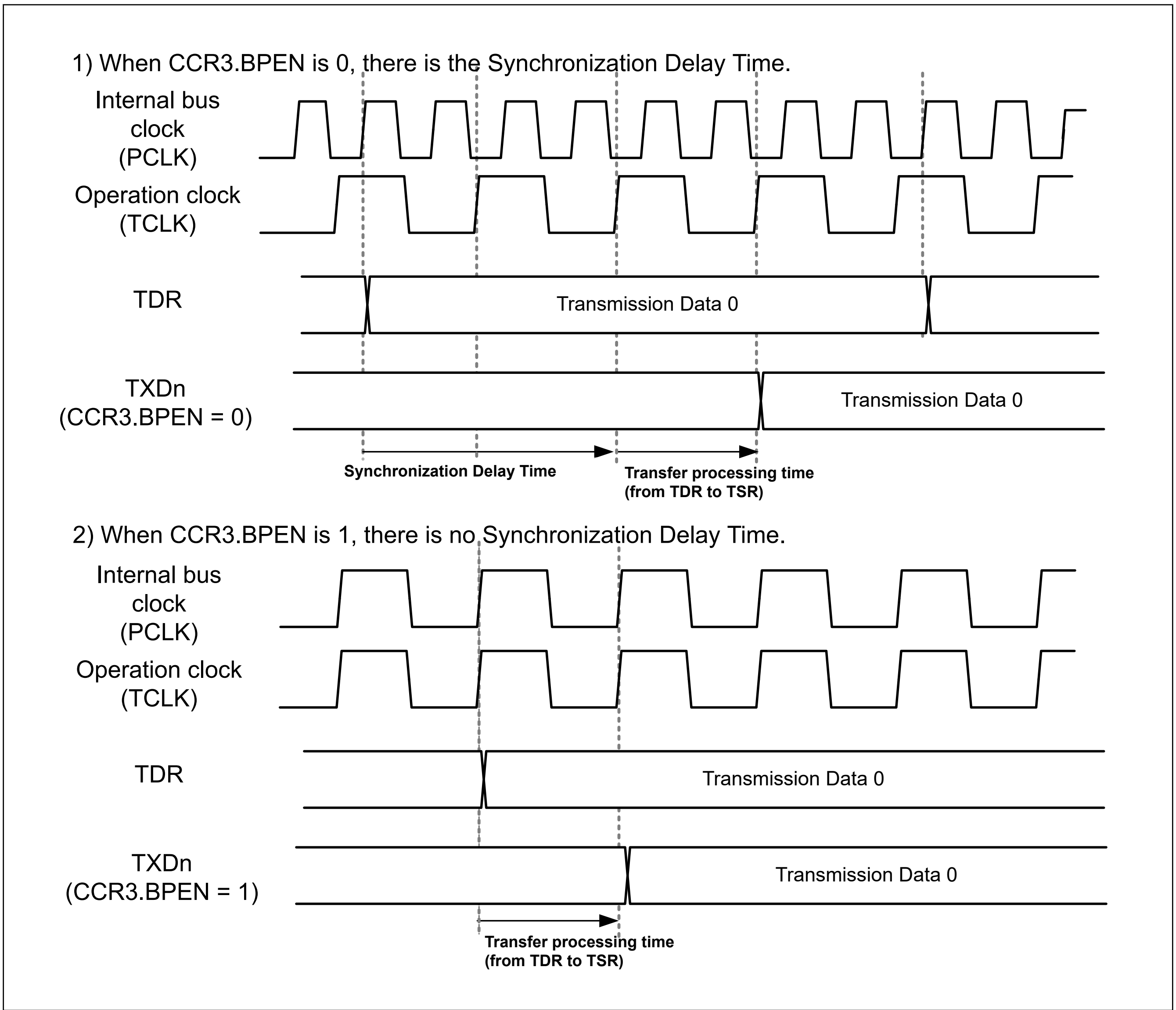


Figure 38.116 Image waveform of Synchronizer bypass function

38.19 Usage Notes

38.19.1 Settings for the Module-Stop Function

The Module Stop Control Register B (MSTPCRB) can enable or disable SCI operation. The SCI is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Mode.

38.19.2 Software Standby Mode

(1) Transmission

Before using the power consumption reduction function to reduce SCI's power consumption, do the following to confirm transmission end (CSR.TEND = 1):

- Set the output pin state after transmission operation is stopped by CCR1.SPB2DT, SPB2IO.
- Stop the transmission (CCR0.TIE = 0, TE = 0, TEIE = 0)

When transitions to these states are made during transmission, the data being transmitted become indeterminate.

To transmit data in the same operation mode after cancellation of the low power consumption state, set the TE bit to 1, read CSR, and write data to TDR sequentially to start data transmission. To transmit data with a different operation mode, initialize the SCI first.

To start transmission using the DMAC or DTC after cancellation from software standby mode, set the CCR0.TE and CCR0.TIE bit to at the same time. Then SCIn_TXI interrupt flag is generated, which causes the DMAC or DTC to write the transmit data, which starts transmission.

Figure 38.117 shows a sample flowchart for transition to software standby mode during transmission. Figure 38.118 and Figure 38.119 show the port pin states during transition to software standby mode.

(2) Reception

When Address match function is non used as condition of resumption (wake-up)

Before specifying the module stop state or making a transition to software standby mode, stop the receive operations (CCR0.RE = 0). If transition is made during data reception, the data being received will be invalid.

Figure 38.120 shows a sample flowchart for reception to software standby mode during reception.

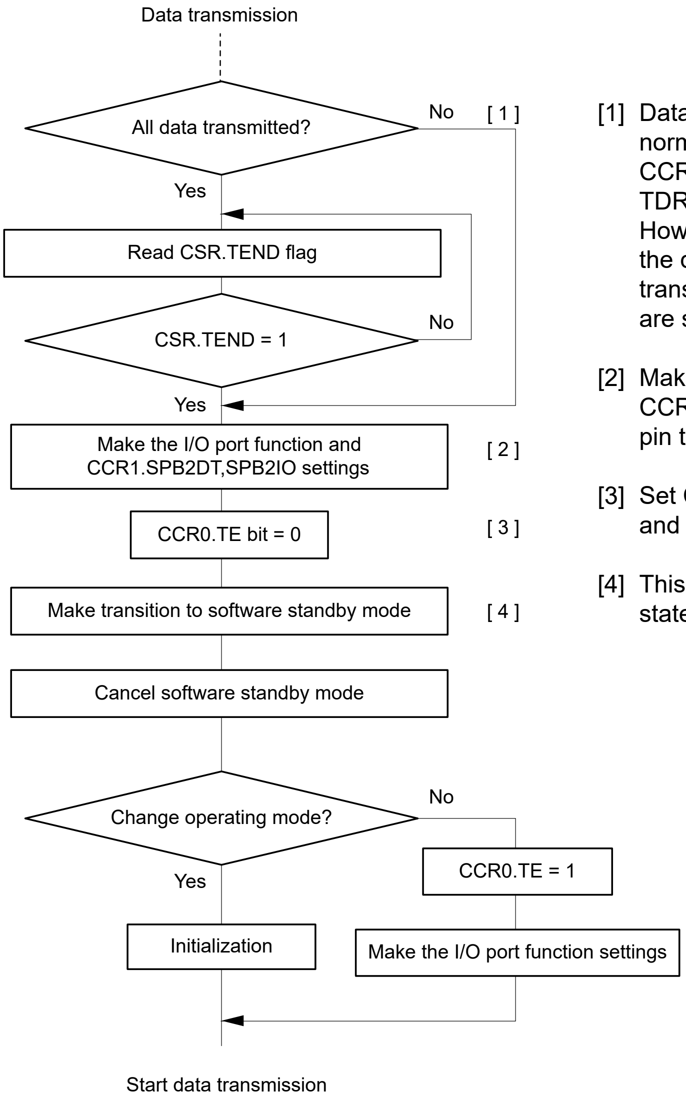
When Address match function is used as condition of resumption (wake-up)

When using the power consumption reduction function to reduce SCI's power consumption, do the following:

- Set the operation mode to Asynchronous mode
- Set the compare data to CCR4.CMPD and set 1 to CCR0.DCME
- Set the receive operation (CCR0.RE = 1)

Set CCR3.RXDESEL = 0. Because there is a possibility that a start bit (fall edge of RXDn pin) cannot be detected at the time of low power consumption mode release.

Figure 38.121 shows a sample flowchart for reception to software standby mode during reception.



[1] Data being transmitted is lost halfway. Data can be normally transmitted from the CPU by setting the CCR0.TE to "1", reading CSR, and writing data to TDR after canceling software stand by mode. However, if the DMAC or DTC has been activated, the data remaining in the DMAC or DTC will be transmitted when both the TE and TIE bits in CCR0 are set to "1".

[2] Make the I/O port function and CCR1.SPB2DT,SPB2IO settings to switch the TxDn pin to operate as a general I/O port.

[3] Set CCR0.TE bit to "0". Set also 0 to CCR0.TIE and CCR0.TEIE if you set these bits were 1.

[4] This includes the setting for the module stopped state.

Figure 38.117 Example of Flowchart for Transition to Software Standby Mode during Transmission

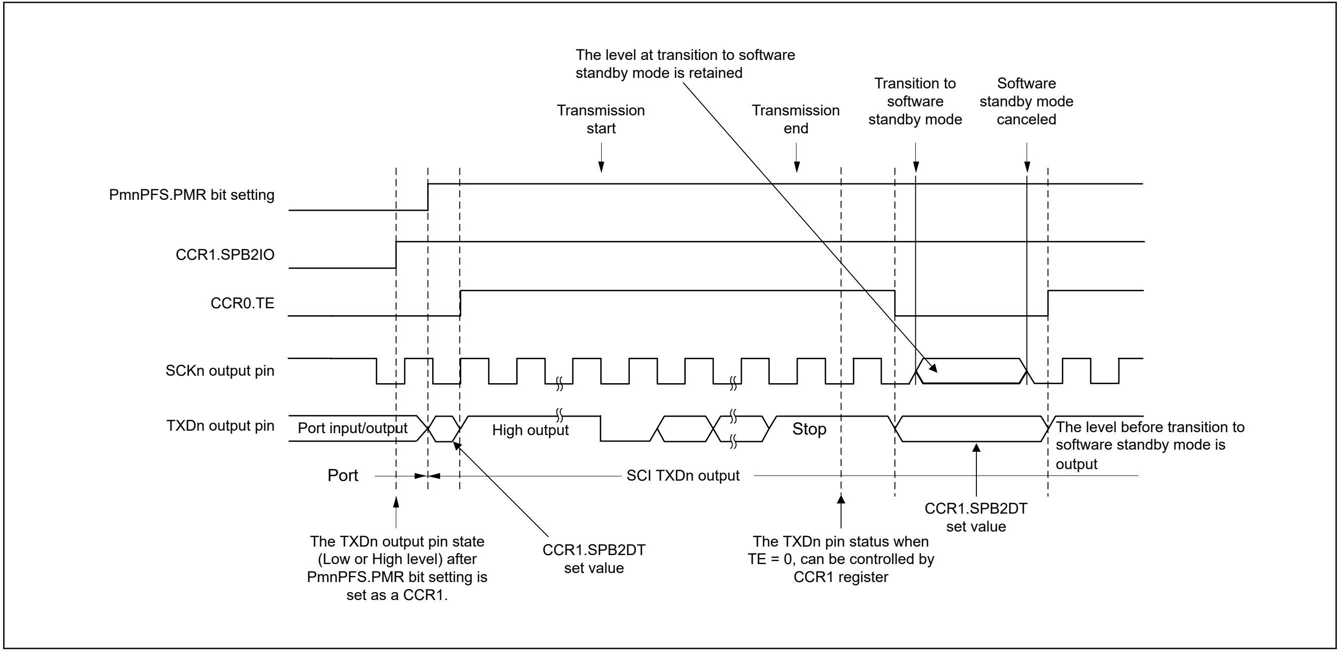


Figure 38.118 Port Pin States during Transition to Software Standby Mode (Internal Clock, Asynchronous Transmission)

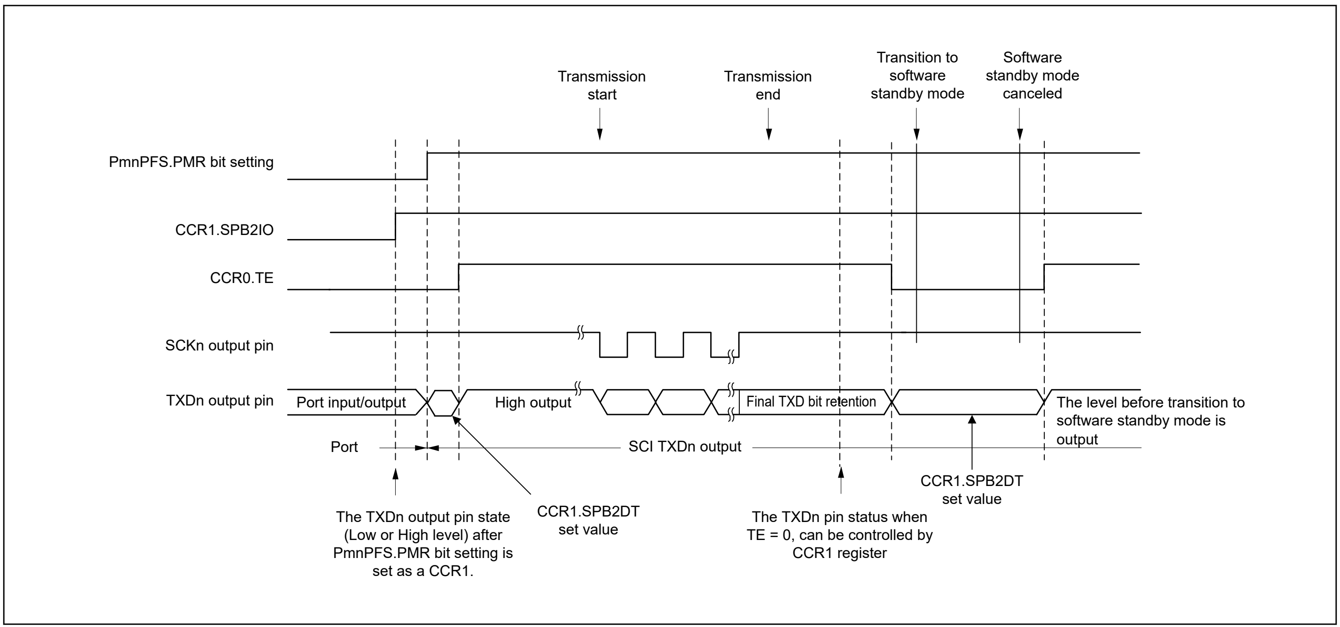


Figure 38.119 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clock Synchronous Transmission)

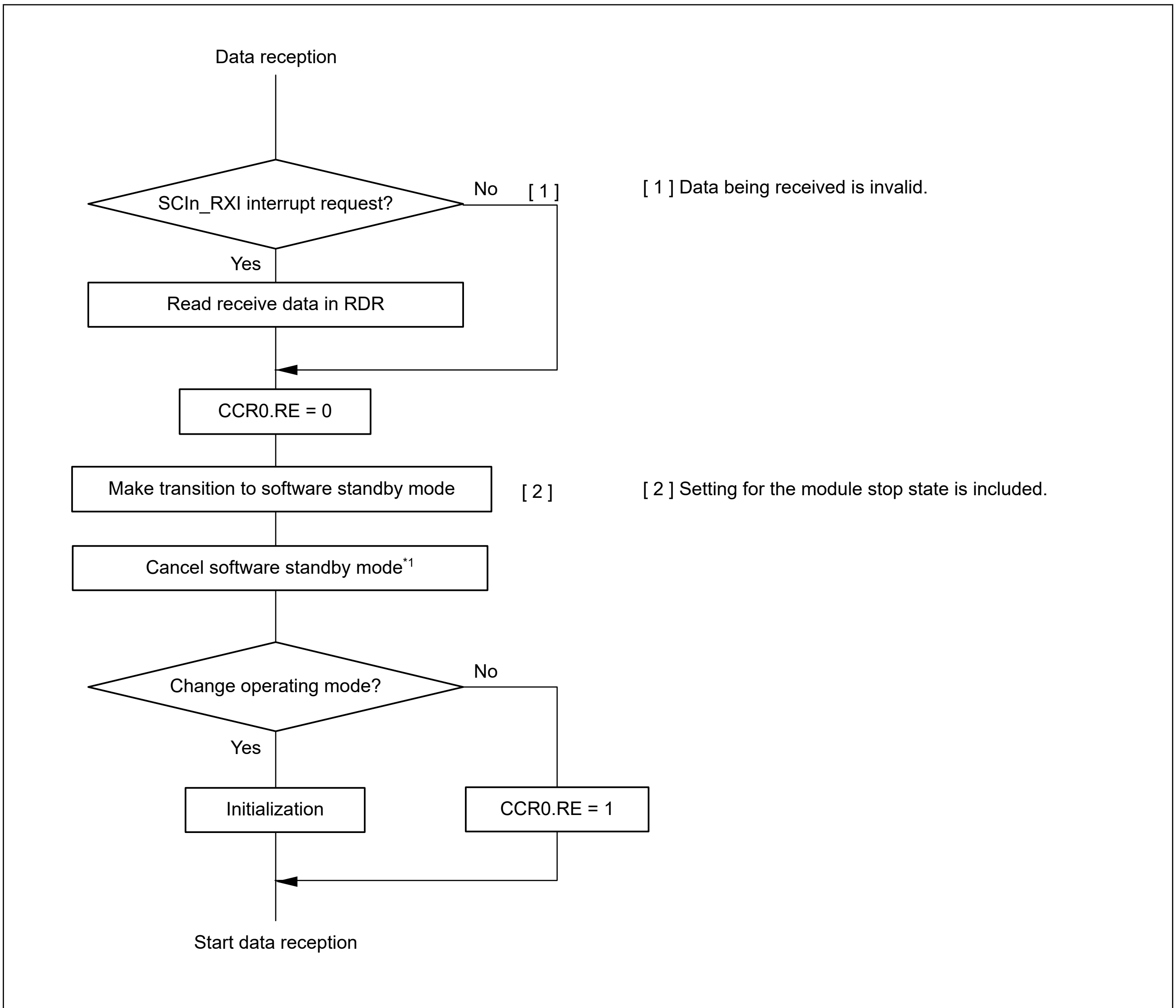


Figure 38.120 Example of Flowchart for Reception to Software Standby Mode during Reception

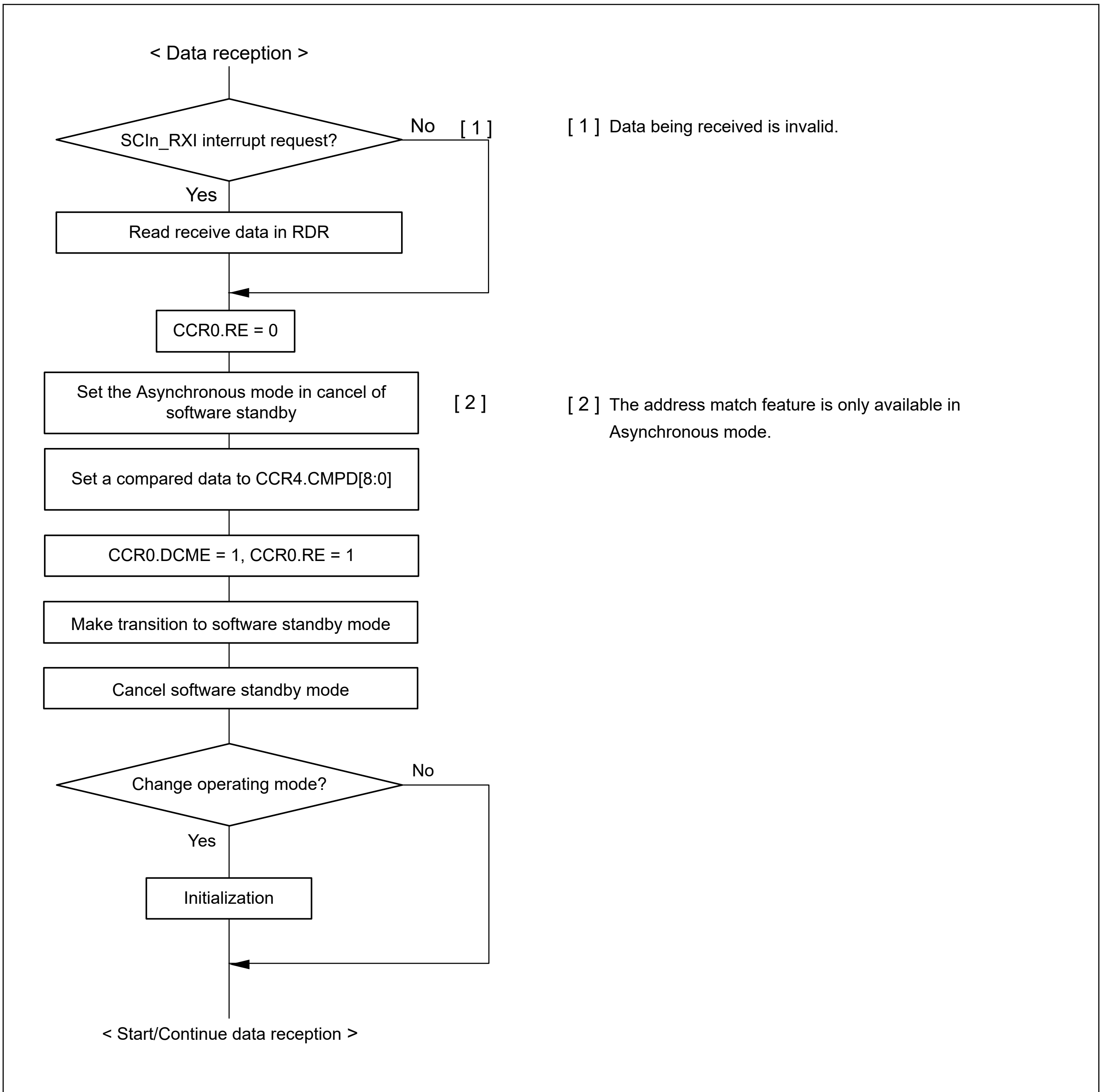


Figure 38.121 Example of Flowchart for Reception to Software Standby Mode during Reception with Address match

38.19.3 Break Detection and Processing

(1) Non-FIFO selected

When a framing error is detected, a break can be detected by reading CSR.RXDMON bit value. In a break, the input from the RXDn pin becomes all 0s, and the CSR.FER flag is set to 1 to indicate a framing error, and the CSR.PER flag might also be set to 1 to indicate a parity error. The SCI continues the receive operation even after a break is received. Therefore, even if the FER flag is 0, indicating that no framing error occurred, it is set to 1 again. When the CCR3.RXDESEL bit is 1, the SCI sets the CSR.FER flag to 1 and stops receiving operations until a start bit of the next data frame is detected. If the CSR.FER flag is set to 0, the CSR.FER flag retains 0 during the break.

When the RXDn pin is set to 1 and the break ends, detecting the beginning of the start bit on the first falling edge of the RXDn pin allows the SCI to start the receiving operation.

(2) FIFO selected

After a framing error is detected and when the SCI detects that continuous receive data is 0 for 1 frame, reception stops. When a framing error is detected, a break can be detected by reading the CSR.RXDMON flag value. After the RXDn signal is in high and the break is finished, data reception to the receive-FIFO (RDR) register resumes.

38.19.4 Mark State and Production of Breaks

When the CCR0.TE bit is 0, disabling serial transmission, the state of the TXDn pin can be set using the CCR1.SPB2IO and CCR1.SPB2DT bits. With this approach, a TXDn pin can be placed in the mark state to transmit a break.

Before setting the CCR0.TE bit to 1, enabling serial transmission, set the SPB2IO and SPB2DT bits to put the communication line in the mark state (the state of 1), and change the TXDn pin using I/O port function. To output a break on data transmission, after setting the TXDn pin to output 0 by setting the SPB2IO and SPB2DT bits, change the TXDn pin using the I/O port function and set the CCR0.TE bit to 0. When the CCR0.TE bit is set to 0, the transmitter is initialized regardless of the current state of transmission.

38.19.5 Receive Error Flags and Transmit Operations (Clock Synchronous Mode and Simple SPI Mode)

Transmission can be start by writing transmit-data to TDR even if CSR.ORER is 1. However, reception cannot be started. Note also that the receive error flags cannot be set to 0 even if the CCR0.RE is set to 0 (serial reception is disabled).

38.19.6 Writing Data to TDR

(1) Non-FIFO selected

Data can be written to TDR anytime when CCR0.TE is 1. However, if new data is written to TDR when transmit data is remaining in TDR, the previous data in TDR is lost because it has not been transferred to TSR yet. If you use DTC or DMAC, be sure to write transmit data to TDR in the SCIn_TXI interrupt request handling routine.

(2) FIFO selected

Data can be written to transmit-FIFO(TDR) when CCR0.TE is 1. Check the number of writable data with the FTSR.T [5:0] bit.

38.19.7 Restrictions on Clock Synchronous Transmission (Clock Synchronous Mode and Simple SPI Mode)

When the external clock source is used as a synchronization clock, the following restrictions apply.

(1) Start of transmission

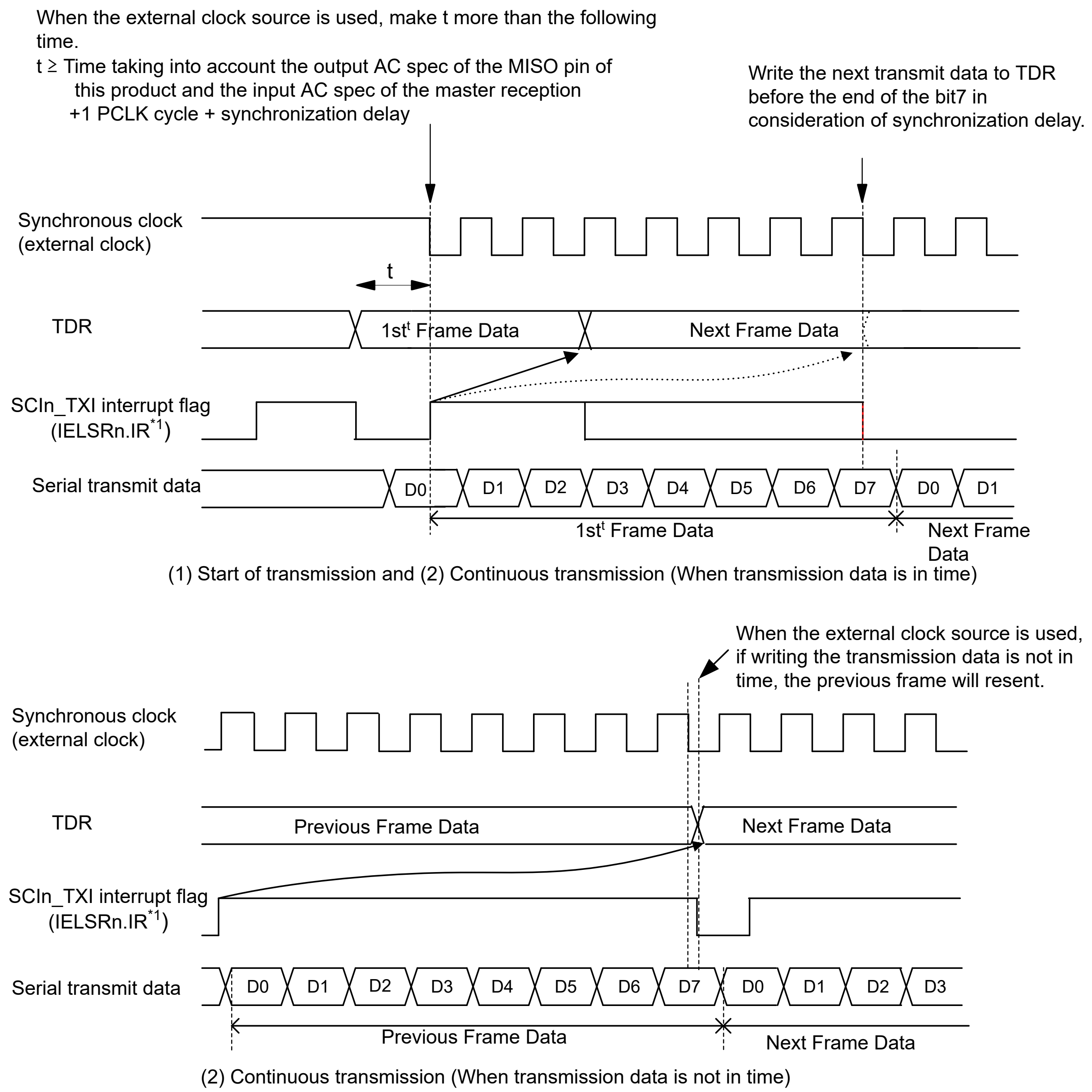
Update TDR by the CPU, DMAC, or DTC and wait at least the following time until the start of the external clock input: (See Figure 38.122)

Take the following time into account: the output AC spec of the MISO pin of this product and the input AC spec of the master reception + 1 PCLK cycle + synchronization delay.

(2) Continuous transmission

Write the next transmit data to TDR before the falling edge[*1] of the transmit clock for bit 7. Write the transmit data to TDR in consideration of synchronization delay. If the transmit data cannot be written in time, the previous frame data is resent. (See Figure 38.122)

Note 1. When CCR3.CPOL = 1 and CCR3.CPHA = 0, or CCR3.CPOL = 0 and CCR3.CPHA = 1. In the case of CCR3.CPOL = 0 and CCR3.CPHA = 0, or CCR3.CPOL = 1 and CCR3.CPHA = 1, it's the rising edge.



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 38.122 Restrictions on Use of External Clock in Clock Synchronous Transmission

38.19.8 Restrictions on Using DMAC or DTC

When using the DMAC or DTC to read RDR, be sure to set the receive data full interrupt (SCIn_RXI) as the activation source of the relevant SCI.

During the operation in transmission / reception using the DMAC or DTC, it should not set transfer information of DMAC or DTC.

38.19.9 Notes on Starting Transfer

At the point where transfer starts when the interrupt status flag (IELSRn.IR flag) in the ICU is 1, follow the procedure in this section to clear interrupt requests before permitting operations (by setting the CCR0.TE or CCR0.RE bit to 1). For details on the interrupt status flag, see section 14, Interrupt Controller Unit (ICU).

1. Confirm that transfer has stopped (the CCR0.TE or CCR0.RE bit is 0)

2. Set the associated interrupt enable bit (CCR0.TIE or CCR0.RIE bit) to 0
3. Read the associated interrupt enable bit (CCR0.TIE or CCR0.RIE bit) to check that it actually becomes 0
4. Set the interrupt status flag, IELSRn.IR, in the ICU to 0

38.19.10 Limitations on Simple SPI Mode

(1) Master mode

- Use a resistor to pull up or pull down the clock line matching the initial settings for the transfer clock set in the CCR3.CPHA and CPOL bits when the CCR0.SSE bit is 1.

  This prevents the clock line from being placed in the high-impedance state when the CCR0.TE bit is set to 0 or unexpected edges from being generated on the clock line when the CCR0.TE bit changes from 0 to 1. When the CCR0.SSE bit is 0 in single master mode, pulling up or pulling down the clock line is not required because the clock line is not placed in the high-impedance state even when the SCR.TE bit is set to 0.
- For the clock delay setting (CCR3.CPHA bit is 0), the receive data full interrupt (SCIn_RXI) is generated before the final clock edge on the SCKn pin as indicated in Figure 38.123. If the TE and RE bits in the CCR0 register become 0 before the final edge of the clock signal on the SCKn pin, the SCKn pin is placed in the high-impedance state, so the width of the last clock pulse of the transfer clock is shortened. Additionally, an SCIn_RXI interrupt might lead to the input signal on the SSn pin of a connected slave going to the high level before the final edge of the clock signal on the SCKn pin, leading to incorrect operation of the slave.
- In a multi-master configuration, the SCKn pin output goes to high-impedance while the input on the SSn pin is at the low level if a mode fault error occurs while a character is being transferred, stopping supply of the clock signal to the connected slave. Reset the connected slave to avoid misaligned bits when transfer is restarted.

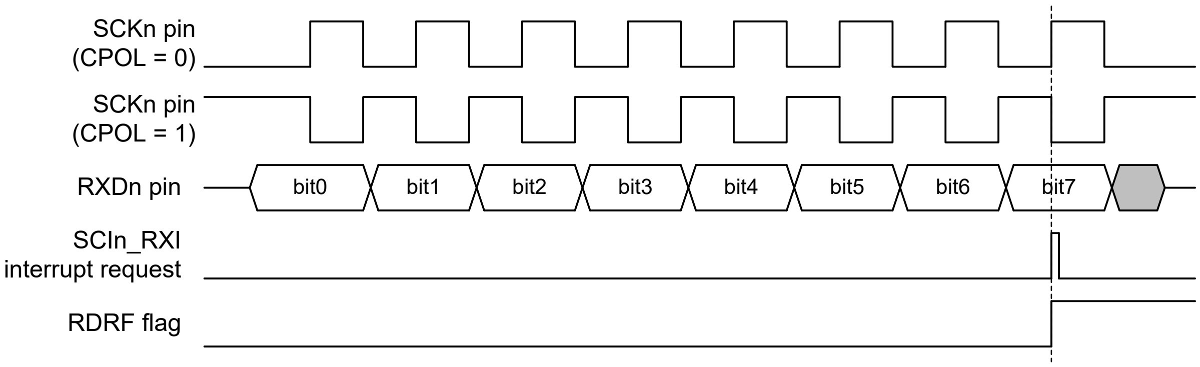


Figure 38.123 Timing of SCIn_RXI interrupt in simple SPI mode with clock delay

(2) Slave mode

- It takes 1PCLK + synchronization delay time + data output delay time (AC spec) from writing the transmit data to the TDR register until the data is output to the RXDn pin. Take these into account when starting external clock input.
- Provide an external clock signal to the master the same as the data length for transfer
- Secure the SSn input setup time (AC spec) from the SSn low-level input to the start of external clock input.
- Control the input on the SSn pin before the start and after the end of data transfer
- When the input level on the SSn pin is to be changed from low to high while a character is being transferred, set the TE and RE bits in the CCR0 register to 0 and, after restoring the settings, restart transfer of the first byte

38.19.11 Notes on Transmit Enable bit (CCR0.TE)

In initial register value, when CCR0.TE = 0, the state of the TXDn pin is high impedance. The TXDn line should not be high impedance by the following one of ways.

1. The pull-up or pull-down resistance is connected to the TXDn line.

2. Set CCR1 and decided level of TXDn terminal during TE is 0.

38.19.12 Notes on Simple LIN Mode

In Simple LIN mode (CCR3.MOD[2:0] = 110), the following functions cannot be used.

- Multi-processor communication function
- Bit Rate Modulation function
- Loopback function
- FIFO buffer

38.19.13 Notes on RS-485 Driver Control Function

RS-485 Driver control function is valid only in Asynchronous mode.

When RS-485 Driver control function is active (CCR3.DEN = 1), the CSR.TEND set timing / SCIn_TEI output timing changes as follows. Wait for the SCIn_TEI interrupt and set the CCR0.TE bit in SCI to 0.

When RS-485 Driver control function is inactive: When STOP bit output is completed.

When RS-485 Driver control function is active: At the end of DEn negation time.

38.19.14 Notes on Loopback Function

The Loopback function is valid in Asynchronous mode with internal clock, in Manchester mode with internal clock and Clock synchronous mode with internal clock.

38.19.15 Notes Regarding Register Access When Operation Clock (TCLK) Is Slower than Bus Clock (PCLK)

If the operating clock (TCLK) is slower than the bus clock (PCLK), the time until this information is transmitted internally after writing to the CCR0.TE and CCR0.RE registers is slower than the bus access time. In particular, when trying to change the setting register after writing 0 and interrupting communication, do not change the setting register before the signal inside the IP is in the communication stopped state. To prevent this, after setting CCR0.TE and CCR0.RE to 0, check the CESR.TIST and CESR.RIST bits until they are 0 before setting the next register.

38.19.16 Notes on Interrupting Operation

If 0 is written to CCR0.RE during data reception and the reception operation is interrupted, there is a possibility of an invalid state, do not use the received data (RDR register stored value) and the flag value of each status register. To interrupt the reception operation, stop the interrupt or event link reception side and then write 0 to the CCR0.RE bit.

38.19.17 Notes on CCR3.BPEN Bit Setting

Set the BPEN bit only once when setting the CCR3 register in the SCI initialization flow.

This bit cannot be changed after the initialization.

When this bit setting is changed, start from the SCI initialization flow again.

39. I^{2}C Bus Interface (IIC)

39.1 Overview

The I^{2}C bus interface (IIC) has 3 channels. The IIC module conforms with and provides a subset of the NXP I^{2}C (Inter-Integrated Circuit) bus interface functions.

Table 39.1 lists the IIC specifications, Figure 39.1 shows a block diagram, and Figure 39.2 shows an example of I/O pin connections to external circuits, with an I^{2}C bus configuration. Table 39.2 lists the I/O pins.

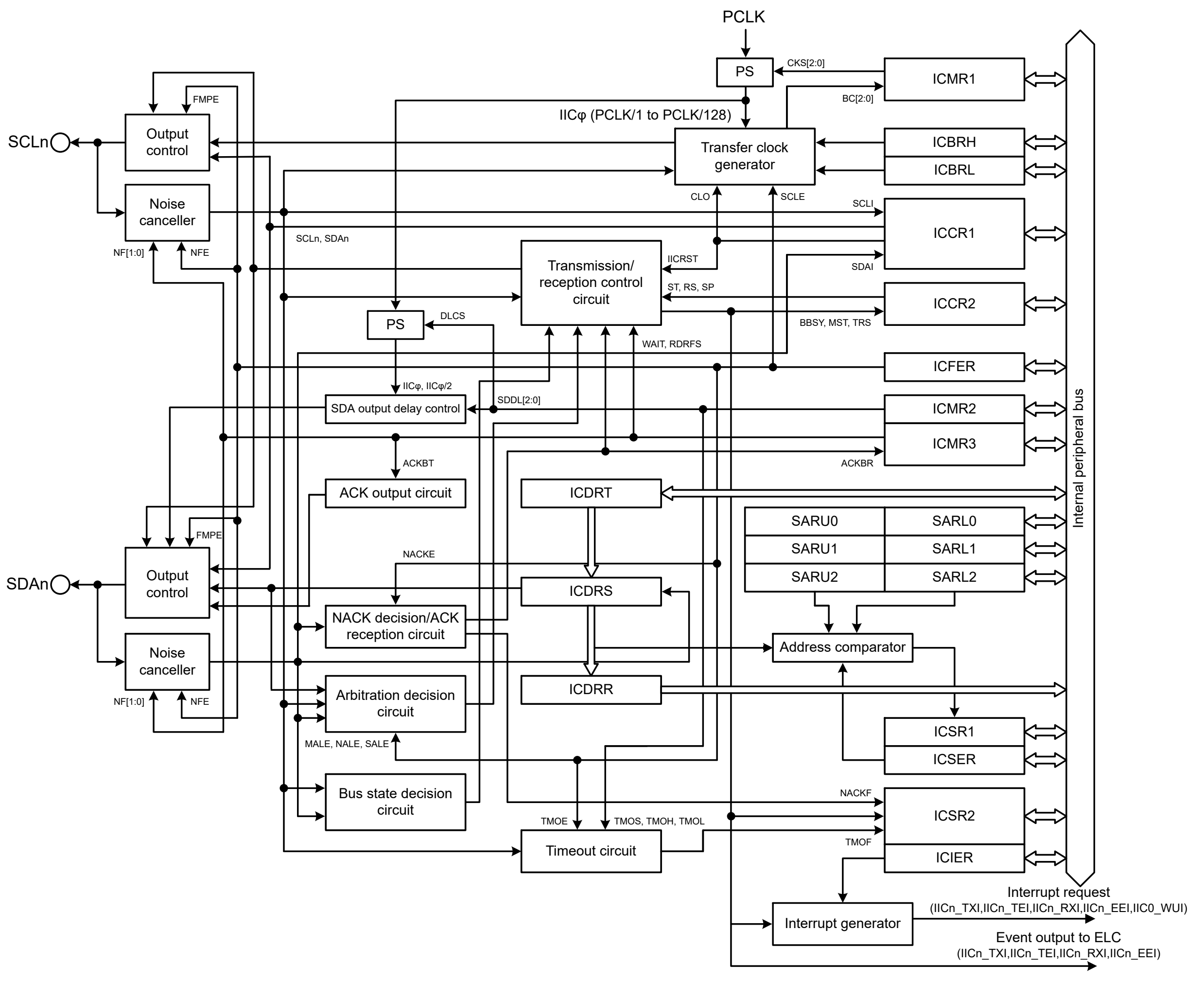
Table 39.1 IIC specifications (1 of 2)

Parameter	Specifications
Communications format	• I^{2}C-bus format or SMBus format • Master or slave mode selectable • Automatic securing of the setup times, hold times, and bus-free times for the transfer rate
Transfer rate	• Fast-mode Plus supported, up to 1 Mbps
SCL clock	For master operation, the duty cycle of the SCL clock is selectable in the range from 4% to 96%
Issuing and detecting conditions	• Start, restart, and stop conditions are automatically generated • Start conditions (including restart conditions) and stop conditions are detectable
Slave address	• Configurable for up to three different slave addresses • 7- and 10-bit address formats supported, including simultaneous use • General call addresses, device ID addresses, and SMBus host addresses detectable
Acknowledgment	• For transmission, automatic loading of the acknowledge bit Transfer of the next transmit data can be automatically suspended on detection of a not-acknowledge bit. • For reception, automatic transmission of the acknowledge bit If a wait between the 8th and 9th clock cycles is selected, the software can control the value in the acknowledge field in response to the received value.
Wait function	During reception, the following wait periods are available by holding the SCL clock low: • Waiting between the eighth and ninth clock cycles • Waiting between the ninth clock cycle and the 1st clock cycle of the next transfer
SDA output delay function	Output timing of transmitted data, including the acknowledge bit, can be delayed
Arbitration	• For multi-master operation: – SCL clock synchronization is possible when conflict occurs with the SCL signal from another master – When issuing the start condition creates conflict on the bus, loss of arbitration is detected by testing for a mismatch between the internal signal for the SDA line and the level on the SDA line – In master operation, loss of arbitration is detected by testing for non-matching between the signal on the SDA line and the internal signal for the SDA line • Loss of arbitration because the start condition occurs while the bus is busy is detectable, to prevent the issuing of double start conditions • Loss of arbitration is detectable on transfer of a not-acknowledge bit because the internal signal for the SDA line and the level on the SDA line do not match • Loss of arbitration because a mismatch of internal and line levels for data is detectable in slave transmission
Timeout function	Internal detection of long-interval stops of the SCL clock
Noise cancellation	• Digital noise filters for both the SCL and SDA signals • Programmable window for noise cancellation by the filters
Interrupt sources	• Transfer error or event occurrence (arbitration-lost, NACK, timeout, start or restart condition, or stop condition) • Receive data full, including matching with a slave address • Transmit data empty, including matching with a slave address • Transmit end
Module-stop function	Module-stop state can be set
IIC operating modes	• Master transmit • Master receive • Slave transmit • Slave receive

Table 39.1 IIC specifications (2 of 2)

Parameter	Specifications
Event link function (output)	● Transfer error or event occurrence (arbitration-lost, NACK, timeout, start or restart condition, or stop condition) ● Receive data full, including matching with a slave address ● Transmit data empty, including matching with a slave address ● Transmit end
Wakeup function*1	CPU can return from Software Standby mode using a wakeup event
TrustZone Filter	Security and Privilege attribution can be set for each channels

Note 1. This function is only available for IIC channel IIC0. IIC1 and IIC2 are not supported.



Note: n = 0 to 2
PCLK = PCLKB

Figure 39.1 IIC block diagram

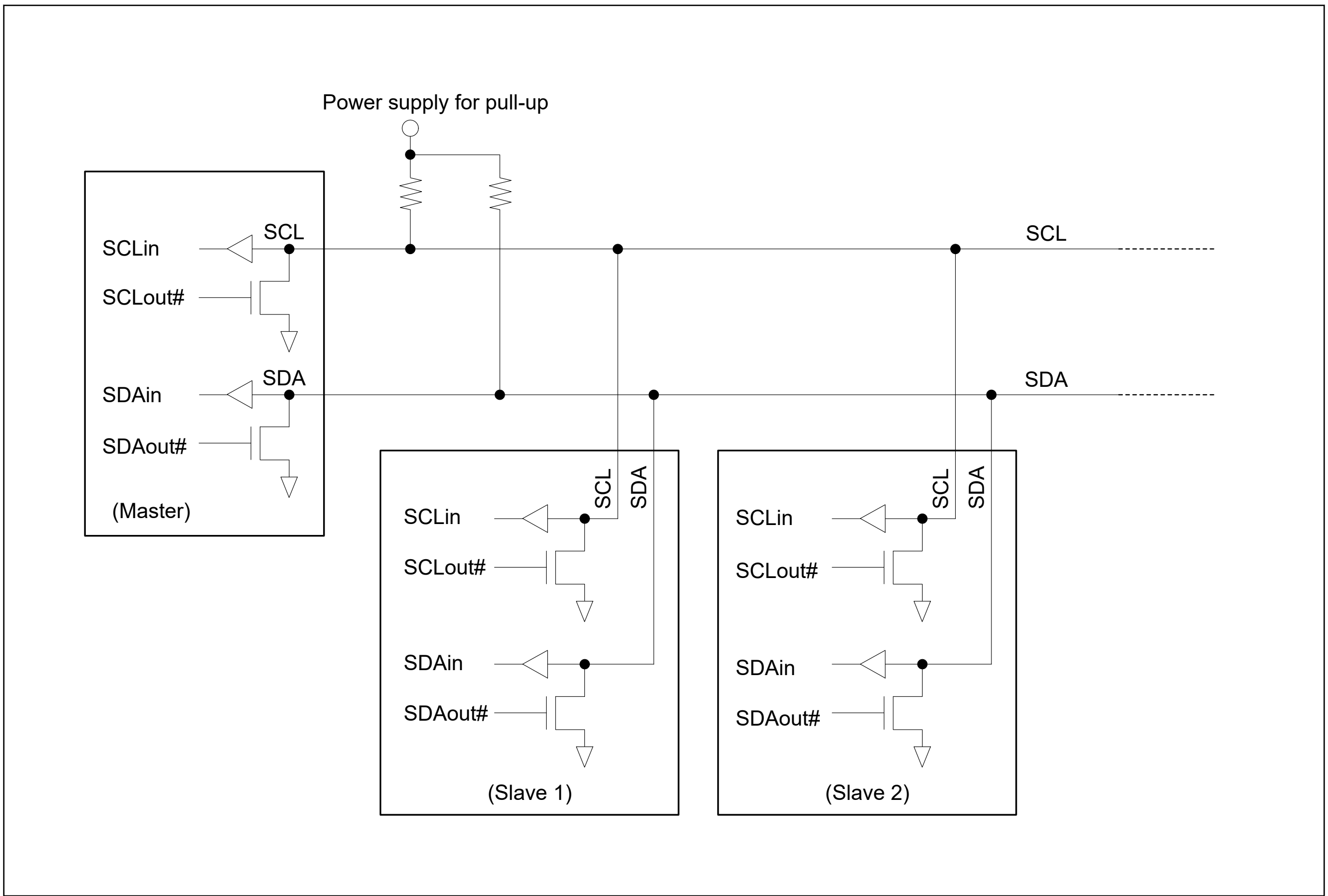


Figure 39.2 I/O pin connection to an external circuit (I^{2}C bus configuration example)

The input level of the signals for IIC is CMOS when I^{2}C bus is selected (ICMR3.SMBS = 0), or TTL when SMBus is selected (ICMR3.SMBS = 1).

Table 39.2 IIC I/O pins

Channel	Pin name	I/O	Function
IICn	SCLn	I/O	IICn serial clock I/O pin
	SDAn	I/O	IICn serial data I/O pin

Note: n = 0 to 2

39.2 Register Descriptions

39.2.1 ICCR1 : I^{2}C Bus Control Register 1

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 2)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 2)

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Function	R/W
0	SDAI	SDA Line Monitor 0: SDAn line is low 1: SDAn line is high	R

Bit	Symbol	Function	R/W
1	SCLI	SCL Line Monitor 0: SCLn line is low 1: SCLn line is high	R
2	SDAO	SDA Output Control/Monitor 0: Read: IIC drives SDAn pin low Write: IIC drives SDAn pin low 1: Read: IIC releases SDAn pin Write: IIC releases SDAn pin	R/W
3	SCLO	SCL Output Control/Monitor Use an external pull-up resistor to drive the signal high. 0: Read: IIC drives SCLn pin low Write: IIC drives SCLn pin low 1: Read: IIC releases SCLn pin Write: IIC releases SCLn pin	R/W
4	SOWP	SCLO/SDAO Write Protect This bit is read as 1. 0: Write enable SCLO and SDAO bits 1: Write protect SCLO and SDAO bits	W
5	CLO	Extra SCL Clock Cycle Output This bit clears automatically after 1 clock cycle is output. 0: Do not output extra SCL clock cycle (default) 1: Output extra SCL clock cycle	R/W
6	IICRST	I^{2}C Bus Interface Internal Reset This setting clears the bit counter and the SCLn/SDAn output latch. 0: Release IIC reset or internal reset 1: Initiate IIC reset or internal reset	R/W
7	ICE	I^{2}C Bus Interface Enable Used in combination with the IICRST bit to select either IIC or internal reset. 0: Disable (SCLn and SDAn pins in inactive state) 1: Enable (SCLn and SDAn pins in active state)	R/W

Note: S-TYPE3, P-TYPE3

SDAO bit (SDA Output Control/Monitor) and SCLO bit (SCL Output Control/Monitor)

The SDAO and SCLO bits directly control the SDAn and SCLn signals output from the IIC. When writing to these bits, also write 0 to the SOWP bit. Setting these bits results in input to the IIC by the input buffer. When slave mode is selected, a start condition might be detected and the bus might be released, depending on the bit settings.

Do not rewrite these bits during a start condition, stop condition, restart condition, transmission, or reception. Operation after rewriting under these conditions is not guaranteed. When reading these bits, the state of signals output from the IIC can be read.

CLO bit (Extra SCL Clock Cycle Output)

The CLO bit allows output of an extra SCL clock cycle for debugging or error processing. Normally, set this bit to 0. Setting the bit to 1 in a normal communication state causes a communication error. For details on this function, see section 39.12.2. Extra SCL Clock Cycle Output Function.

IICRST bit (I^{2}C Bus Interface Internal Reset)

The IICRST bit initiates an internal state reset of the IIC. Setting this bit to 1 initiates an IIC reset or internal reset. Whether an IIC reset or internal reset is initiated is determined by the settings of this bit in combination with the ICE bit. Table 39.3 lists the IIC resets.

The IIC reset initializes all registers except ICCR1.ICE and ICCR1.IICRST bits and internal states of the IIC. In addition to the internal states of the IIC, the internal reset initializes the following:

- Bit counter (ICMR1.BC[2:0] bits)
- I^{2}C Bus Shift Register (ICDRS)
- I^{2}C Bus Status Registers (ICSR1 and ICSR2)
- SDAO and SCLO Output Control/Monitor (ICCR1.SDAO and ICCR1.SCLO bits)

- I^2C Bus Control Register 2 (except ICCR2.BBSY bit)

For the reset conditions for each register, see section 39.15. State of Registers When Issuing Each Condition.

An internal reset initiated with the IICRST bit set to 1 during operation (with the ICE bit set to 1) resets the internal states of the IIC without initializing the port settings and the control and setting registers of the IIC. If the IIC hangs in a low-level output state, resetting the internal states cancels the low-level output state and releases the bus with the SCLn pin and SDAn pin at high impedance.

Note: If an internal reset is initiated using the IICRST bit for a bus hang-up that occurs during communication with the master device in slave mode, the slave and master devices might enter different states, because the bit counter information differs. For this reason, do not initiate an internal reset in slave mode. Initiate recovery processing from the master device. If an internal reset is required because the IIC hangs with the SCLn line in a low-level output state in slave mode, initiate an internal reset, then issue a restart condition from the master device, or issue a stop condition and resume communication from the start condition. If communication is restarted by initiating a reset solely in the slave device without issuing a start or restart condition from the master device, synchronization is lost because the master and slave devices operate asynchronously.

Table 39.3 IIC resets

IICRST	ICE	State	Specifications
1	0	IIC reset	Resets all registers except ICCR1.IICRST and ICCR1.ICE bits, and the internal states of the IIC
	1	Internal reset	Reset the following: ● ICMR1.BC[2:0] bits ● ICSR1, ICSR2, ICDRS registers ● SDAO and SCLO Output Control/Monitor (ICCR1.SDAO and ICCR1.SCLO bits) ● I^2C Bus Control Register 2 (except ICCR2.BBSY bit) ● Internal states of the IIC

ICE bit (I^2C Bus Interface Enable)

The ICE bit selects the active or inactive state of the SCLn and SDAn pins. It can also be combined with the IICRST bit to initiate two types of resets. See Table 39.3 for the reset descriptions.

Set the ICE bit to 1 when using the IIC. The SCLn and SDAn pins are placed in the active state when the ICE bit is set to 1. Set the ICE bit to 0 when the IIC is not used. The SCLn and SDAn pins are placed in the inactive state when the ICE bit is set to 0. Do not assign the SCLn or SDAn pin to the IIC when setting up the pin function control. Slave address comparison is performed if the pins are assigned to the IIC.

39.2.2 ICCR2 : I^2C Bus Control Register 2

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 2)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 2)

Offset address: 0x01

Bit position:	7	6	5	4	3	2	1	0
Bit field:	BBSY	MST	TRS	—	SP	RS	ST	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	ST	Start Condition Issuance Request 0: Do not issue a start condition request 1: Issue a start condition request	R/W
2	RS	Restart Condition Issuance Request 0: Do not issue a restart condition request 1: Issue a restart condition request	R/W

Bit	Symbol	Function	R/W
3	SP	Stop Condition Issuance Request 0: Do not issue a stop condition request 1: Issue a stop condition request	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	TRS	Transmit/Receive Mode 0: Receive mode 1: Transmit mode	R/W*1
6	MST	Master/Slave Mode 0: Slave mode 1: Master mode	R/W*1
7	BBSY	Bus Busy Detection Flag 0: I^2C bus released (bus free state) 1: I^2C bus occupied (bus busy state)	R

Note: S-TYPE3, P-TYPE3
Note 1. The MST and TRS bits can be written to when the ICMR1.MTWP bit is set to 1.

ST bit (Start Condition Issuance Request)

The ST bit requests transition to master mode and triggers a start condition. When this bit is set to 1, a start condition is issued when the BBSY flag is set to 0 (bus free state). For details on this function, see section 39.11. Start, Restart, and Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the ST bit.

[Clearing conditions]

- When 0 is written to the ST bit
- When a start condition is issued (a start condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Only set the ST bit to 1 (start condition request) when the BBSY flag is set to 0 (bus free state). Arbitration might be lost if the ST bit is set to 1 (start condition request) when the BBSY flag is 1 (bus busy state).

RS bit (Restart Condition Issuance Request)

The RS bit requests that a restart condition be issued in master mode. When this bit is set to 1 to request a restart condition, a restart condition is issued when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode). For details on this function, see section 39.11. Start, Restart, and Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the RS bit with the BBSY flag in ICCR2 set to 1.

[Clearing conditions]

- When 0 is written to the RS bit
- When a restart condition is issued (a start condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Do not set the RS bit to 1 while issuing a stop condition.
Note: If 1 (restart condition request) is written to the RS bit in slave mode, the restart condition is not issued, but the RS bit remains set to 1. If the operating mode changes to master mode without the bit being cleared, a restart condition might be issued.

SP bit (Stop Condition Issuance Request)

The SP bit requests that a stop condition be issued in master mode. When this bit is set to 1, a stop condition is issued when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode). For details on this function, see section 39.11. Start, Restart, and Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the SP bit with both the BBSY flag and the MST bit in ICCR2 set to 1.

[Clearing conditions]

- When 0 is written to the SP bit
- When a stop condition is issued (a stop condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When a start condition and a restart condition are detected
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Writing to the SP bit is not possible while the BBSY flag is 0 (bus free state).
Note: Do not set the SP bit to 1 while a restart condition is being issued.

TRS bit (Transmit/Receive Mode)

The TRS bit indicates transmit or receive mode. The IIC is in receive mode when the TRS bit is 0 and in transmit mode when the bit is 1. The combination of this bit and the MST bit indicates the IIC operating mode.

The value of the TRS bit automatically changes to 1 for transmit mode or 0 for receive mode when a start condition is issued or detected and the R/W# bit is set. Although writing to the TRS bit is possible when the MTWP bit in ICMR1 is set to 1, writing to this bit is not required during normal usage.

[Setting conditions]

- When a start condition is issued normally because of a start condition request (when a start condition is detected with the ST bit set to 1)
- When a restart condition is issued normally because of a restart condition request (when a restart condition is detected with the RS bit set to 1)
- When the R/W# bit appended to the slave address is set to 0 in master mode
- When the address received in slave mode matches the address enabled in ICSER, with the R/W# bit set to 1
- When 1 is written to the TRS bit with the MTWP bit in ICMR1 set to 1.

[Clearing conditions]

- When a stop condition is detected
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When the R/W# bit appended to the slave address is set to 1 in master mode
- In slave mode, on a match between the received address and the address enabled in ICSER when the value of the received R/W# bit is 0, including when the received address is the general call address
- In slave mode, when a restart condition is detected (a restart condition is detected with ICCR2.BBSY = 1 and ICCR2.MST = 0)
- When 0 is written to the TRS bit with the MTWP bit in ICMR1 set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

MST bit (Master/Slave Mode)

The MST bit indicates master or slave mode. The IIC is in slave mode when the MST bit is 0 and is in master mode when the bit is 1. The combination of this bit and the TRS bit indicates the IIC operating mode.

The value of the MST bit automatically changes to 1 for master mode or 0 for slave mode when a start condition is issued or a stop condition is issued or detected. Although writing to the MST bit is possible when the MTWP bit in ICMR1 is set to 1, writing to this bit is not required during normal usage.

[Setting conditions]

- When a start condition is issued normally because of a start condition request (when a start condition is detected with the ST bit set to 1)
- When 1 is written to the MST bit with the MTWP bit in ICMR1 set to 1.

[Clearing conditions]

- When a stop condition is detected
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 0 is written to the MST bit with the MTWP bit in ICMR1 set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

BBSY flag (Bus Busy Detection Flag)

The BBSY flag indicates whether the I^2C bus is occupied (bus busy state) or released (bus free state). The flag is set to 1 when the SDAn line changes from high to low when the SCLn line is high, assuming that a start condition was issued. The flag is set to 0 if a start condition is not detected for the bus free time (ICBRL setting), assuming that a stop condition was issued.

[Setting condition]

- When a start condition is detected.

[Clearing conditions]

- When a start condition is not detected for the bus free time (ICBRL setting) after detecting a stop condition
- When 1 is written to the IICRST bit in ICCR1 with the ICE bit in ICCR1 set to 0 (IIC reset).

39.2.3 ICMR1 : I^2C Bus Mode Register 1

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 2)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 2)

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MTWP	CKS[2:0]			BCWP	BC[2:0]		
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
2:0	BC[2:0]	Bit Counter 0 0 0: 9 bits 0 0 1: 2 bits 0 1 0: 3 bits 0 1 1: 4 bits 1 0 0: 5 bits 1 0 1: 6 bits 1 1 0: 7 bits 1 1 1: 8 bits	R/W*1
3	BCWP	BC Write Protect This bit is read as 1. 0: Write enable BC[2:0] bits 1: Write protect BC[2:0] bits	W*1
6:4	CKS[2:0]	Internal Reference Clock Select Select the internal reference clock source (IICφ) for the IIC. IICφ = (PCLKB / $2^{CKS[2:0]}$) clock	R/W
7	MTWP	MST/TRS Write Protect 0: Write protect MST and TRS bits in ICCR2 1: Write enable MST and TRS bits in ICCR2	R/W

Note: S-TYPE3, P-TYPE3

Note 1. Rewrite the BC[2:0] bits and set the BCWP bit to 0 at the same time.

BC[2:0] bits (Bit Counter)

The BC[2:0] bits function as a counter indicating the number of bits remaining to be transferred on detection of a rising edge on the SCLn line. Although BC[2:0] are read/write bits, it is not required to access these bits under normal conditions.

To write to these bits, specify the number of bits to be transferred plus one, for an additional acknowledge bit, between transferred frames when the SCLn line is at a low level. The value in the BC[2:0] bits returns to 000b at the end of a data transfer, including the acknowledge bit, or when a start or restart condition is detected.

39.2.4 ICMR2 : I^{2}C Bus Mode Register 2

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 2)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 2)

Offset address: 0x03

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DLCS	SDDL[2:0]			—	TMOH	TMOL	TMOS
Value after reset:	0	0	0	0	0	1	1	0

Bit	Symbol	Function	R/W
0	TMOS	Timeout Detection Time Select 0: Select long mode 1: Select short mode	R/W
1	TMOL	Timeout L Count Control 0: Disable count while SCLn line is low 1: Enable count while SCLn line is low	R/W
2	TMOH	Timeout H Count Control 0: Disable count while SCLn line is high 1: Enable count while SCLn line is high	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	SDDL[2:0]	SDA Output Delay Counter 0 0 0: No output delay 0 0 1: 1 IICφ cycle (When ICMR2.DLCS = 0 (IICφ)) 1 or 2 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 0 1 0: 2 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 3 or 4 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 0 1 1: 3 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 5 or 6 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 1 0 0: 4 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 7 or 8 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 1 0 1: 5 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 9 or 10 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 1 1 0: 6 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 11 or 12 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 1 1 1: 7 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 13 or 14 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2))	R/W
7	DLCS	SDA Output Delay Clock Source Select 0: Select internal reference clock (IICφ) as the clock source for SDA output delay counter 1: Select internal reference clock divided by 2 (IICφ/2) as the clock source for SDA output delay counter*1	R/W

Note: S-TYPE3, P-TYPE3

Note 1. The setting DLCS = 1 (IICφ/2) is only valid when SCL is low. When SCL is high, the DLCS = 1 setting becomes invalid and the clock source becomes the internal reference clock (IICφ).

TMOS bit (Timeout Detection Time Select)

The TMOS bit selects long or short mode for the timeout detection time when the timeout function is enabled (ICFER.TMOE = 1). When this bit is set to 0, long mode is selected. When it is set to 1, short mode is selected. In long mode, the timeout detection internal counter functions as a 16 bit-counter. In short mode, the counter functions as a

14- bit counter. While the SCLn line is in the state that enables this counter as specified in the TMOH and TMOL bits, the counter counts up in synchronization with the internal reference clock (IICφ) as a count source. For details on this function, see section 39.12.1. Timeout Function.

TMOL bit (Timeout L Count Control)

The TMOL bit enables or disables up-counting on the internal counter of the timeout function while the SCLn line is held low and the timeout function is enabled (ICFER.TMOE = 1).

TMOH bit (Timeout H Count Control)

The TMOH bit enables or disables up-counting on the internal counter of the timeout function while the SCLn line is held high and the timeout function is enabled (ICFER.TMOE = 1).

SDDL[2:0] bits (SDA Output Delay Counter)

The SDDL[2:0] bits can be used to delay the SDA output. This counter works with the clock source selected in the DLCS bit. This setting can be used for all types of SDA output, including transmission of the acknowledge bit.

Set the SDA output delay to meet the I^2C bus standard for the data enable time/acknowledge enable time,[*1] or the SMBus standard, within [data hold time (300 ns or more + the SCL clock low-level period) - the data setup time (250 ns)]. If a value outside the standard is set, communication between devices might malfunction or falsely indicate a start or stop condition, depending on the bus state.

For details on this function, see section 39.5. SDA Output Delay Function.

Note 1. Data enable time/acknowledge enable time.
3,450 ns for up to 100 kbps: Standard mode (Sm)
900 ns for up to 400 kbps: Fast mode (Fm)
450 ns for up to 1 Mbps: Fast-mode Plus (Fm+)

39.2.5 ICMR3 : I^2C Bus Mode Register 3

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 2)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 2)

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SMBS	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	NF[1:0]	Noise Filter Stage Select 0 0: Filter out noise of up to 1 IICφ cycle (single-stage filter) 0 1: Filter out noise of up to 2 IICφ cycles (2-stage filter) 1 0: Filter out noise of up to 3 IICφ cycles (3-stage filter) 1 1: Filter out noise of up to 4 IICφ cycles (4-stage filter)	R/W
2	ACKBR	Receive Acknowledge 0: 0 received as the acknowledge bit (ACK reception) 1: 1 received as the acknowledge bit (NACK reception)	R
3	ACKBT	Transmit Acknowledge 0: Send 0 as the acknowledge bit (ACK transmission) 1: Send 1 as the acknowledge bit (NACK transmission)	R/W[*1]
4	ACKWP	ACKBT Write Protect 0: Write protect ACKBT bit 1: Write enable ACKBT bit	R/W

Bit	Symbol	Function	R/W
5	RDRFS	RDRF Flag Set Timing Select Low-hold is released by writing to ACKBT. 0: Set the RDRF flag on the rising edge of the 9th SCL clock cycle. The SCLn line is not held low on the falling edge of the 8th clock cycle. 1: Set the RDRF flag on the rising edge of the 8th SCL clock cycle. The SCLn line is held low on the falling edge of the 8th clock cycle.	R/W*2
6	WAIT	Low-hold is released by reading ICDRR. 0: No wait (The SCLn line is not held low during the period between the 9th clock cycle and the 1st clock cycle.) 1: Wait (The SCLn line is held low during the period between the 9th clock cycle and the 1st clock cycle.)	R/W*2
7	SMBS	SMBus/I^{2}C Bus Select 0: Select I^{2}C Bus 1: Select SMBus	R/W

Note: S-TYPE3, P-TYPE3

Note 1. Write to the ACKBT bit only while the ACKWP bit is already 1. If the application writes 1 to the ACKWP and ACKBT bits at the same time, the ACKBT bit is not set to 1.

Note 2. The WAIT and RDRFS bits are only valid in receive mode (invalid in transmit mode).

NF[1:0] bits (Noise Filter Stage Select)

The NF[1:0] bits select the number of stages in the digital noise filter. For details on this function, see section 39.6. Digital Noise Filter Circuits

Note: Set the noise range to be filtered within a range less than the SCLn line high- or low-level period. If the noise range is set to a value of [SCL clock width: high- or low-level period, whichever is shorter] - [1.5 internal reference clock (IICφ) cycles + analog noise filter: 120 ns (reference values)] or more, the SCL clock is regarded as noise, which might prevent the IIC from operating normally.

ACKBR bit (Receive Acknowledge)

The ACKBR bit stores the acknowledge bit information received from the receive device in transmit mode.

[Setting condition]

- When 1 is received as the acknowledge bit with the TRS bit in ICCR2 set to 1.

[Clearing conditions]

- When 0 is received as the acknowledge bit with the TRS bit in ICCR2 set to 1
- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 is 0 (IIC reset).

ACKBT bit (Transmit Acknowledge)

The ACKBT bit sets the acknowledge bit to be sent in receive mode

[Setting condition]

- When 1 is written to this bit with the ACKWP bit set to 1.

[Clearing conditions]

- When 0 is written to this bit with the ACKWP bit set to 1
- When stop condition request is detected with the SP bit in ICCR2 set to 1
- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 is 0 (IIC reset).

ACKWP bit (ACKBT Write Protect)

The ACKWP bit controls write enabling of the ACKBT bit.

RDRFS bit (RDRF Flag Set Timing Select)

The RDRFS bit selects the RDRF flag set timing in receive mode and also selects whether to hold the SCLn line low on the falling edge of the 8th SCL clock cycle.

When the RDRFS bit is 0, the SCLn line is not held low on the falling edge of the 8th SCL clock cycle, and the RDRF flag is set to 1 on the rising edge of the 9th SCL clock cycle.

When the RDRFS bit is 1, the RDRF flag is set to 1 on the rising edge of the 8th SCL clock cycle, and the SCLn line is held low on the falling edge of the 8th SCL clock cycle. The low-hold of the SCLn line is released by a write to the ACKBT bit.

After data is received with this setting, the SCLn line is automatically held low before the acknowledge bit is sent. This enables processing to send ACK (ACKBT = 0) or NACK (ACKBT = 1), based on the receive data.

WAIT bit (WAIT)

The WAIT bit controls whether to force a low-hold between the ninth SCL clock cycle and the first SCL clock cycle, until the receive data buffer (ICDRR) is completely read each time single-byte data is received in receive mode.

When the WAIT bit is 0, the receive operation continues without a low-hold between the ninth and the first SCL clock cycle. When both the RDRFS and WAIT bits are 0, continuous receive operation is enabled with the double buffer.

When the WAIT bit is 1, the SCLn line is held low from the falling edge of the ninth clock cycle until the ICDRR value is read each time single-byte data is received. This enables receive operation in byte units.

Note: When the value of the WAIT bit is to be read, always read ICDRR first.

SMBS bit (SMBus/I²C Bus Select)

Setting the SMBS bit to 1 selects the SMBus and enables the HOAE bit in ICSER.

39.2.6 ICFER : I²C Bus Function Enable Register

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 2)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 2)

Offset address: 0x05

Bit position:	7	6	5	4	3	2	1	0
Bit field:	FMPE	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
Value after reset:	0	1	1	1	0	0	1	0

Bit	Symbol	Function	R/W
0	TMOE	Timeout Function Enable 0: Disable 1: Enable	R/W
1	MALE	Master Arbitration-Lost Detection Enable 0: Disable the arbitration-lost detection function and disable automatic clearing of the MST and TRS bits in ICCR2 when arbitration is lost 1: Enable the arbitration-lost detection function and enable automatic clearing of the MST and TRS bits in ICCR2 when arbitration is lost	R/W
2	NALE	NACK Transmission Arbitration-Lost Detection Enable 0: Disable 1: Enable	R/W
3	SALE	Slave Arbitration-Lost Detection Enable 0: Disable 1: Enable	R/W
4	NACKE	NACK Reception Transfer Suspension Enable 0: Do not suspend transfer operation during NACK reception (disable transfer suspension) 1: Suspend transfer operation during NACK reception (enable transfer suspension)	R/W
5	NFE	Digital Noise Filter Circuit Enable 0: Do not use the digital noise filter circuit 1: Use the digital noise filter circuit	R/W
6	SCLE	SCL Synchronous Circuit Enable 0: Do not use the SCL synchronous circuit 1: Use the SCL synchronous circuit	R/W

Bit	Symbol	Function	R/W
7	FMPE*1	Fast-Mode Plus Enable 0: Do not use the Fm+ slope control circuit for the SCLn and SDAn pins 1: Use the Fm+ slope control circuit for the SCLn and SDAn pins.	R/W

Note: S-TYPE3, P-TYPE3

Note 1. The Fast-mode Plus enable bit (FMPE) is supported by IIC0 (SCL0_A, SDA0_A), IIC1 (SCL1_A, SDA1_A) and IIC2 (SCL2_A, SDA2_A). Bit [7] is the reserved bit in the not supported channel.

TMOE bit (Timeout Function Enable)

The TMOE bit enables or disables the timeout function. For details on this function, see section 39.12.1. Timeout Function.

MALE bit (Master Arbitration-Lost Detection Enable)

The MALE bit specifies whether to use the arbitration-lost detection function in master mode. Normally, set this bit to 1.

NALE bit (NACK Transmission Arbitration-Lost Detection Enable)

The NALE bit specifies whether to cause arbitration to be lost when ACK is detected during transmission of NACK in receive mode, for example when slaves with the same address exist on the bus or when two or more masters select the same slave device simultaneously with a different number of receive bytes.

SALE bit (Slave Arbitration-Lost Detection Enable)

The SALE bit specifies whether to cause arbitration to be lost when a value different from the value being transmitted is detected on the bus in slave transmit mode, for example when slaves with the same address exist on the bus or when a mismatch with the transmit data occurs because of noise.

NACKE bit (NACK Reception Transfer Suspension Enable)

The NACKE bit specifies whether to continue or discontinue the transfer operation when NACK is received in transmit mode. Normally, set this bit to 1.

When NACK is received with the NACKE bit set to 1, the next transfer operation is suspended. When the NACKE bit is 0, the next transfer operation continues regardless of the received acknowledge content.

For details, see section 39.9.2. NACK Reception Transfer Suspension Function.

SCLE bit (SCL Synchronous Circuit Enable)

The SCLE bit specifies whether to synchronize the SCL clock with the SCL input clock. Normally, set this bit to 1.

When the SCLE bit is set to 0 (no SCL synchronous circuit used), the IIC does not synchronize the SCL clock with the SCL input clock. With this setting, the IIC outputs the SCL clock at the transfer rate set in ICBRH and ICBRL, regardless of the SCLn line state. For this reason, if the bus load of the I^2C bus line is much larger than the specification value, or if the SCL clock output overlaps in multiple masters, a short-cycle SCL clock that does not meet the specification might be output. When no SCL synchronous circuit is used, it also affects the issuing of the start, restart, and stop conditions, and the continuous output of extra SCL clock cycles.

Do not set this bit to 0 except when checking the output of the set transfer rate.

FMPE bit (Fast-Mode Plus Enable)

The FMPE bit specifies whether to use a slope control circuit for Fast-mode Plus (Fm+).

When this bit is set to 1, a slope control circuit conforming to the I^2C bus Fast-mode Plus (Fm+) standard (tof) is selected. When this bit is set to 0, a slope control circuit conforming to the I^2C bus Standard-mode (Sm) and Fast-mode (Fm) standards (tof) is selected.

Set this bit to 1 when using transmission rates up to 1 Mbps (Fast-mode Plus (Fm+) standard). Set it to 0 when using other transmission rates (up to 100 kbps (Sm) or up to 400 kbps (Fm)) or for SMBus (10 to 100 kbps).

39.2.7 ICSER : I^{2}C Bus Status Enable Register

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 2)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 2)

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	HOAE	—	DIDE	—	GCAE	SAR2E	SAR1E	SAR0E
Value after reset:	0	0	0	0	1	0	0	1

Bit	Symbol	Function	R/W
0	SAR0E	Slave Address Register 0 Enable 0: Disable slave address in SARL0 and SARU0 1: Enable slave address in SARL0 and SARU0	R/W
1	SAR1E	Slave Address Register 1 Enable 0: Disable slave address in SARL1 and SARU1 1: Enable slave address in SARL1 and SARU1	R/W
2	SAR2E	Slave Address Register 2 Enable 0: Disable slave address in SARL2 and SARU2 1: Enable slave address in SARL2 and SARU2	R/W
3	GCAE	General Call Address Enable 0: Disable general call address detection 1: Enable general call address detection	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	DIDE	Device-ID Address Detection Enable 0: Disable device-ID address detection 1: Enable device-ID address detection	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	HOAE	Host Address Enable 0: Disable host address detection 1: Enable host address detection	R/W

Note: S-TYPE3, P-TYPE3

SARyE bit (Slave Address Register y Enable) (y = 0 to 2)

The SARyE bit enables or disables the received slave address and the slave address set in SARLy and SARUy.

When this bit is set to 1, the slave address set in SARLy and SARUy is enabled and is compared with the received slave address. When this bit is set to 0, the slave address set in SARLy and SARUy is disabled and is ignored even if it matches the received slave address.

GCAE bit (General Call Address Enable)

The GCAE bit specifies whether to ignore the general call address (0000 000b + 0 [W]: All 0) when it is received.

When this bit is set to 1, if the received slave address matches the general call address, the IIC recognizes the received slave address as the general call address independently of the slave addresses set in SARLy and SARUy (y = 0 to 2) and performs the data receive operation. When this bit is set to 0, the received slave address is ignored even if it matches the general call address.

DIDE bit (Device-ID Address Detection Enable)

The DIDE bit specifies whether to recognize and execute the device-ID address when a device ID (1111 100b) is received in the first frame after a start or restart condition is detected.

When this bit is set to 1, if the received first frame matches the device ID, the IIC recognizes that the device-ID address was received. When the next R/W# bit is 0 (W), the IIC recognizes the second and the subsequent frames as slave addresses and continues the receive operation. When this bit is set to 0, the IIC ignores the received first frame even if it matches the device-ID address, and it recognizes the first frame as a normal slave address.

For details on this function, see section 39.7.3. Device-ID Address Detection.

HOAE bit (Host Address Enable)

The HOAE bit specifies whether to ignore the received host address (0001 000b) when the SMBS bit in ICMR3 is 1.

When this bit is set to 1 while the SMBS bit in ICMR3 is 1, if the received slave address matches the host address, the IIC recognizes the received slave address as the host address independently of the slave addresses set in SARLy and SARUy (y = 0 to 2) and performs the receive operation.

When the SMBS bit in ICMR3 or the HOAE bit is set to 0, the received slave address is ignored even if it matches the host address.

39.2.8 ICIER : I^{2}C Bus Interrupt Enable Register

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 2)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 2)

Offset address: 0x07

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMOIE	Timeout Interrupt Request Enable 0: Disable timeout interrupt (TMOI) request 1: Enable timeout interrupt (TMOI) request	R/W
1	ALIE	Arbitration-Lost Interrupt Request Enable 0: Disable arbitration-lost interrupt (ALI) request 1: Enable arbitration-lost interrupt (ALI) request	R/W
2	STIE	Start Condition Detection Interrupt Request Enable 0: Disable start condition detection interrupt (STI) request 1: Enable start condition detection interrupt (STI) request	R/W
3	SPIE	Stop Condition Detection Interrupt Request Enable 0: Disable stop condition detection interrupt (SPI) request 1: Enable stop condition detection interrupt (SPI) request	R/W
4	NAKIE	NACK Reception Interrupt Request Enable 0: Disable NACK reception interrupt (NAKI) request 1: Enable NACK reception interrupt (NAKI) request	R/W
5	RIE	Receive Data Full Interrupt Request Enable 0: Disable receive data full interrupt (IICn_RXI) request 1: Enable receive data full interrupt (IICn_RXI) request	R/W
6	TEIE	Transmit End Interrupt Request Enable 0: Disable transmit end interrupt (IICn_TEI) request 1: Enable transmit end interrupt (IICn_TEI) request	R/W
7	TIE	Transmit Data Empty Interrupt Request Enable 0: Disable transmit data empty interrupt (IICn_TXI) request 1: Enable transmit data empty interrupt (IICn_TXI) request	R/W

Note: S-TYPE3, P-TYPE3

TMOIE bit (Timeout Interrupt Request Enable)

The TMOIE bit enables or disables timeout interrupt (TMOI) requests when the TMOF flag in ICSR2 is 1. To cancel a TMOI interrupt request, set the TMOF flag or the TMOIE bit to 0.

ALIE bit (Arbitration-Lost Interrupt Request Enable)

The ALIE bit enables or disables arbitration-lost interrupt (ALI) requests when the AL flag in ICSR2 is 1. To cancel an ALI interrupt request, set the AL flag or the ALIE bit to 0.

STIE bit (Start Condition Detection Interrupt Request Enable)

The STIE bit enables or disables start condition detection interrupt (STI) requests when the START flag in ICSR2 is 1. To cancel an STI interrupt request, set the START flag or the STIE bit to 0.

SPIE bit (Stop Condition Detection Interrupt Request Enable)

The SPIE bit enables or disables stop condition detection interrupt (SPI) requests when the STOP flag in ICSR2 is 1. To cancel an SPI interrupt request, set the STOP flag or the SPIE bit to 0.

NAKIE bit (NACK Reception Interrupt Request Enable)

The NAKIE bit enables or disables NACK reception interrupt (NAKI) requests when the NACKF flag in ICSR2 is 1. To cancel an NAKI interrupt request, set the NACKF flag or the NAKIE bit to 0.

RIE bit (Receive Data Full Interrupt Request Enable)

The RIE bit enables or disables receive data full interrupt (IICn_RXI) requests when the RDRF flag in ICSR2 is 1.

TEIE bit (Transmit End Interrupt Request Enable)

The TEIE bit enables or disables transmit end interrupt (IICn_TEI) requests when the TEND flag in ICSR2 is 1. To cancel an IICn_TEI interrupt request, set the TEND flag or the TEIE bit to 0.

TIE bit (Transmit Data Empty Interrupt Request Enable)

The TIE bit enables or disables transmit data empty interrupt (IICn_TXI) requests when the TDRE flag in ICSR2 is 1.

39.2.9 ICSR1 : I^{2}C Bus Status Register 1

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 2)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 2)

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	HOA	—	DID	—	GCA	AAS2	AAS1	AAS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AAS0	Slave Address 0 Detection Flag 0: Slave address 0 not detected 1: Slave address 0 detected	R/(W)*1
1	AAS1	Slave Address 1 Detection Flag 0: Slave address 1 not detected 1: Slave address 1 detected	R/(W)*1
2	AAS2	Slave Address 2 Detection Flag 0: Slave address 2 not detected 1: Slave address 2 detected	R/(W)*1
3	GCA	General Call Address Detection Flag 0: General call address not detected 1: General call address detected	R/(W)*1
4	—	This bit is read as 0. The write value should be 0.	R/W
5	DID	Device-ID Address Detection Flag This bit is set to 1 when the first frame received immediately after a start condition is detected matches a value of (device ID (1111 100b) + 0[W]). 0: Device-ID command not detected 1: Device-ID command detected	R/(W)*1
6	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7	HOA	Host Address Detection Flag This bit is set to 1 when the received slave address matches the host address (0001 000b). 0: Host address not detected 1: Host address detected	R/(W)[*1]

Note: S-TYPE3, P-TYPE3
Note 1. Only 0 can be written to clear the flag.

AASy flag (Slave Address y Detection flag) (y = 0 to 2)

The AASy flag indicates whether slave address y was detected.

[Setting conditions]

For 7-bit address format (SARUy.FS = 0):

- When the received slave address matches the SVA[6:0] value in SARLy, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

For 10-bit address format: (SARUy.FS = 1):

- When the received slave address matches a value of (11110b + SVA[1:0] in SARUy), and the subsequent address matches the SARLy value, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the AASy flag after reading AASy = 1
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

For 7-bit address format (SARUy.FS = 0):

- When the received slave address does not match the SVA[6:0] value in SARLy, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.

For 10-bit address format (SARUy.FS = 1):

- When the received slave address does not match a value of (11110b + SVA[1:0] in SARUy), with the SARyE bit in ICSER set to 1 (slave address y detection enabled)
  The AASy flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When the received slave address matches a value of (11110b + SVA[1:0] in SARUy), and the subsequent address does not match the SARLy value, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.

GCA flag (General Call Address Detection Flag)

The GCA flag indicates whether the general call address was detected.

[Setting condition]

- When the received slave address matches the general call address (0000 000b + 0 [W]), with the GCAE bit in ICSER set to 1 (general call address detection enabled).
  The GCA flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the GCA flag after reading GCA = 1
- When a stop condition is detected
- When the received slave address does not match the general call address (0000 000b + 0 [W]), with the GCAE bit in ICSER set to 1 (general call address detection enabled)
  The GCA flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.

- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

DID flag (Device-ID Address Detection Flag)

The DID flag indicates whether the device-ID address was detected.

[Setting condition]

- When the first frame received immediately after a start or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]), with the DIDE bit in ICSER set to 1 (device-ID address detection enabled).
  The DID flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the DID flag after reading DID = 1
- When a stop condition is detected
- When the first frame received immediately after a start or restart condition is detected does not match a value of (device ID (1111 100b)), with the DIDE bit in ICSER set to 1 (device-ID address detection enabled)
  The DID flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When the first frame received immediately after a start or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]), and the second frame does not match any slave address from 0 to 2, with the DIDE bit in ICSER set to 1 (device-ID address detection enabled)
  The DID flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

HOA flag (Host Address Detection Flag)

The HOA flag indicates whether the host address was detected.

[Setting condition]

- When the received slave address matches the host address (0001 000b), with the HOAE bit in ICSER set to 1 (host address detection enabled).
  The HOA flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the HOA flag after reading HOA = 1
- When a stop condition is detected
- When the received slave address does not match the host address (0001 000b), with the HOAE bit in ICSER set to 1 (host address detection enabled)
  The HOA flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

39.2.10 ICSR2 : I^{2}C Bus Status Register 2

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 2)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 2)

Offset address: 0x09

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMOF	Timeout Detection Flag 0: Timeout not detected 1: Timeout detected	R/(W)*1

Bit	Symbol	Function	R/W
1	AL	Arbitration-Lost Flag 0: Arbitration not lost 1: Arbitration lost	R/(W)[*1]
2	START	Start Condition Detection Flag 0: Start condition not detected 1: Start condition detected	R/(W)[*1]
3	STOP	Stop Condition Detection Flag 0: Stop condition not detected 1: Stop condition detected	R/(W)[*1]
4	NACKF	NACK Detection Flag 0: NACK not detected 1: NACK detected	R/(W)[*1]
5	RDRF	Receive Data Full Flag 0: ICDRR contains no receive data 1: ICDRR contains receive data	R/(W)[*1]
6	TEND	Transmit End Flag 0: Data being transmitted 1: Data transmit complete	R/(W)[*1]
7	TDRE	Transmit Data Empty Flag 0: ICDRT contains transmit data 1: ICDRT contains no transmit data	R

Note: S-TYPE3, P-TYPE3
Note 1. Only 0 can be written, to clear the flag.

TMOF flag (Timeout Detection Flag)

The TMOF flag is set to 1 when the IIC detects a timeout because the SCLn line state remains unchanged for the set period.

[Setting condition]

- When the SCLn line state remains unchanged for the period specified in the ICMR2.TMOH, TMOL, and TMOS bits while the ICFER.TMOE bit is 1 (timeout function enabled) in master or in slave mode and the received slave address matches.

[Clearing conditions]

- When 0 is written to the TMOF flag after reading TMOF = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

AL flag (Arbitration-Lost Flag)

The AL flag indicates that bus mastership was lost in arbitration because of a bus conflict or some other reason when a start condition was issued or an address and data was transmitted. The IIC monitors the level on the SDAn line during transmission and, if the level on the line does not match the value of the bit being output, is set the value of the AL flag to 1 to indicate that the bus is occupied by another device.

The IIC can also set the AL flag to indicate the detection of arbitration loss during NACK transmission or during data transmission.

[Setting conditions]

When master arbitration-lost detection is enabled (ICFER.MALE = 1):

- When the internal SDA output state does not match the SDAn line level on the rising edge of the SCL clock except for the ACK period during data transmission in master transmit mode
- When a start condition is detected while the ST bit in ICCR2 is 1 (start condition requested) or the internal SDA output state does not match the SDAn line level
- When the ST bit in ICCR2 is 1 (start condition requested), with the BBSY flag in ICCR2 set to 1.

When NACK arbitration-lost detection is enabled (ICFER.NALE = 1):

- When the internal SDA output state does not match the SDAn line level on the rising edge of the SCL clock in the ACK period during NACK transmission in receive mode.

When slave arbitration-lost detection is enabled (ICFER.SALE = 1):

- When the internal SDA output state does not match the SDAn line level on the rising edge of the SCL clock, except for the ACK period during data transmission in slave transmit mode.

[Clearing conditions]

- When 0 is written to the AL flag after reading AL = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Table 39.4 Relationship between arbitration-lost generation sources and arbitration-lost enable functions

ICFER			ICSR2		
MALE	NALE	SALE	AL	Error	Arbitration-lost generation source
1	x	x	1	Start condition issuance error	When internal SDA output state does not match SDAn line level when a start condition is detected, while the ST bit in ICCR2 is 1
					When ST in ICCR2 is set to 1 while BBSY in ICCR2 is 1
			1	Transmit data mismatch	When transmit data (including slave address) does not match the bus state in master transmit mode
x	1	x	1	NACK transmission mismatch	When ACK is detected during transmission of NACK in master or slave receive mode
x	x	1	1	Transmit data mismatch	When transmit data does not match the bus state in slave transmit mode

x: Don’t care

START flag (Start Condition Detection Flag)

The START flag indicates whether a start or restart condition was detected.

[Setting condition]

- When a start or restart condition is detected.

[Clearing conditions]

- When 0 is written to the START flag after reading START = 1
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

STOP flag (Stop Condition Detection Flag)

The STOP flag indicates whether a stop condition was detected.

[Setting condition]

- When a stop condition is detected.

[Clearing conditions]

- When 0 is written to the STOP flag after reading STOP = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

NACKF flag (NACK Detection Flag)

The NACKF flag indicates whether a NACK was detected.

[Setting condition]

- When acknowledge is not received (NACK received) from the receive device in transmit mode, with the NACKE bit in ICFER set to 1 (transfer suspension enabled).

[Clearing conditions]

- When 0 is written to the NACKF flag after reading NACKF = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: When the NACKF flag is set to 1, the IIC suspends data transmission and reception. Writing to ICDRT in transmit mode or reading from ICDRR in receive mode with the NACKF flag set to 1 does not enable data transmit or receive operation. To restart data transmission or reception, set the NACKF flag to 0.

RDRF flag (Receive Data Full Flag)

The RDRF flag indicates whether the ICDRR contains receive data.

[Setting conditions]

- When receive data is transferred from ICDRS to ICDRR
  The RDRF flag is set to 1 on the rising edge of the eighth or ninth SCL clock cycle (selected in the RDRFS bit in ICMR3).
- When the received slave address matches after a start or restart condition is detected with the TRS bit in ICCR2 set to 0.

[Clearing conditions]

- When 0 is written to the RDRF flag after reading RDRF = 1
- When data is read from ICDRR
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

TEND flag (Transmit End Flag)

The TEND flag indicates completion of transmission.

[Setting condition]

- On the rising edge of the ninth SCL clock cycle while the TDRE flag is 1.

[Clearing conditions]

- When 0 is written to the TEND flag after reading TEND = 1
- When data is written to ICDRT
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

TDRE flag (Transmit Data Empty Flag)

The TDRE flag indicates whether the ICDRT contains transmit data.

[Setting conditions]

- When data is transferred from ICDRT to ICDRS and ICDRT becomes empty
- When the TRS bit in ICCR2 is set to 1
- When the received slave address matches while the TRS bit is 1.

[Clearing conditions]

- When data is written to ICDRT
- When the TRS bit in ICCR2 is set to 0
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: When the NACKF flag is set to 1 while the NACKE bit in ICFER is 1, the IIC suspends data transmission and reception. In this case, if the TDRE flag is 0 (next transmit data written), data is transferred to the ICDRS register and the ICDRT register becomes empty on the rising edge of the 9th clock cycle, but the TDRE flag does not set to 1.

39.2.11 ICWUR : I^{2}C Bus Wakeup Unit Register

Base address: IIC0WU = 0x4025_E014
IIC0WU_NS = 0x5025_E014

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	WUE	WUIE	WUF	WUACK	—	—	—	WUAFA
Value after reset:	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
0	WUAFA	Wakeup Analog Filter Additional Selection 0: Do not add the wakeup analog filter 1: Add the wakeup analog filter	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	WUACK	ACK Bit for Wakeup Mode Choice of four response modes in combination with ICCR1.IICRST and WUACK. See Table 39.5.	R/W
5	WUF	Wakeup Event Occurrence Flag 0: Slave address not matching during wakeup 1: Slave address matching during wakeup	R/W
6	WUIE	Wakeup Interrupt Request Enable 0: Disable wakeup interrupt request (IIC0_WUI) 1: Enable wakeup interrupt request (IIC0_WUI)	R/W
7	WUE	Wakeup Function Enable 0: Disable wakeup function 1: Enable wakeup function	R/W

Note: S-TYPE3, P-TYPE3

Table 39.5 Wakeup mode

IICRST	WUACK	Operation mode	Description
0	0	Normal wakeup mode 1	ACK response on 9th SCL, and SCL low-hold after 9th SCL.
0	1	Normal wakeup mode 2	No ACK response immediately and SCL low-hold between 8th and 9th SCL. SCL low-hold release and ACK response on 9th SCL.
1	0	Command recovery mode	ACK response on 9th SCL and no SCL low-hold.
1	1	EEP response mode	NACK response on 9th SCL and no SCL low-hold.

WUF flag (Wakeup Event Occurrence Flag)

The WUF flag indicates whether the slave address is matching during wakeup.

[Setting condition]

- When PCLKB is supplied after the slave address of I^{2}C bus is matched with the address whose enable bits of ICSER (except Device ID address) are set to 1 (enable), the bit is set.

[Clearing conditions]

- When 0 is written to the WUF flag after reading WUF = 1 (when WUSYF flag is set to 1).
- When ICCR1.ICE = 0 and IICRST = 1.

39.2.12 ICWUR2 : I^2C Bus Wakeup Unit Register 2

Base address: IIC0WU = 0x4025_E014
IIC0WU_NS = 0x5025_E014

Offset address: 0x03

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	WUSYF	WUASYF	WUSEN
Value after reset:	1	1	1	1	1	1	0	1

Bit	Symbol	Function	R/W
0	WUSEN	Wakeup Function Synchronous Enable 0: IIC asynchronous circuit enable 1: IIC synchronous circuit enable	R/W
1	WUASYF	Wakeup Function Asynchronous Operation Status Flag 0: IIC synchronous circuit enable condition 1: IIC asynchronous circuit enable condition	R
2	WUSYF	Wakeup Function Synchronous Operation Status Flag 0: IIC asynchronous circuit enable condition 1: IIC synchronous circuit enable condition	R
7:3	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE3, P-TYPE3

WUSEN bit (Wakeup Function Synchronous Enable)

The WUSEN bit is used in combination with the WUASYF flag (or WUSYF flag) to switch between the PCLKB synchronous and asynchronous operation, when the wakeup effective function is enabled (ICWUR.WUE = 1).

The PCLKB operation switches from synchronous to asynchronous operation:

When the ICCR2.BBSY flag is 0, if 0 is written to the WUSEN bit while the WUASYF flag is 0, the reception occurs independently of the operation of PCLKB (with PCLKB stopped) after it switches to the PCLKB asynchronous operation, on wakeup event detection.

The PCLKB operation switches from asynchronous to synchronous operation:

- When 1 is written to the WUSEN bit, with the WUASYF flag at 1, when a wakeup event is detected. After writing 1, the WUASYF flag immediately becomes 0.
- When the stop condition is detected with a wakeup event undetected.

[Setting condition]

- When 1 is written to the WUSEN bit.
- ICCR1.ICE = 0 and IICRST = 1 (IIC reset)
- ICWUR.WUE = 0

[Clearing conditions]

- When 0 is written to the WUSEN bit.

WUASYF flag (Wakeup Function Asynchronous Operation Status Flag)

The WUASYF flag can place the IIC in PCLKB asynchronous operation when the wakeup effective function is enabled (ICWUR.WUE = 1).

[Setting condition]

- When the ICCR2.BBSY flag is 0, and the WUSEN bit is set to 0 with the ICWUR.WUE bit set to 1.

[Clearing conditions]

- When 1 is written to the WUSEN bit after detecting the wake-up event with ICWUR.WUE bit set to 1.

- When a stop condition is detected with WUSEN bit set to 1 before detecting the wake-up event with WUASY flag set to 1 with ICWUR.WUE bit set to 1.
- When you write 1 in the WUSEN bit with the WUASYF flag detected 1 and the wake-up event in the state of ICWUR.WUE = 1.
- ICCR1.ICE = 0 and ICCRST = 1 (ICC reset)
- ICWUR.WUE = 0.

WUSYF flag (Wakeup Function Synchronous Operation Status Flag)

It is shown that IIC is in the PCLKB synchronous operation at wake-up effective function (ICWUR.WUE = 1). This flag is a value in which the WUASYF flag is always reserved.

[Setting conditions]

- When 1 is written to the WUSEN bit after detecting the wake-up event with ICWUR.WUE bit set to 1 with WUSYF flag set to 0 with ICWUR.WUE bit set to 1.
- When a stop condition is detected with WUSEN bit set to 1 before detecting the wake-up event with WUSYF flag set to 0 with ICWUR.WUE bit set to 1.
- ICCR1.ICE = 0 and ICCRST = 1 (ICC reset)
- ICWUR.WUE = 0.

[Clearing condition]

- When the ICCR2.BBSY flag is 0 with the ICWUR.WUE bit set to 1 after writing 0 to the WUSEN bit.

39.2.13 SARLy : Slave Address Register Ly (y = 0 to 2)

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 2)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 2)

Offset address: 0x0A + 0x02 × y

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SVA[6:0]							SVA0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SVA0	10-bit Address LSB Slave address setting.	R/W
7:1	SVA[6:0]	7-bit Address/10-bit Address Lower Bits Slave address setting.	R/W

Note: S-TYPE3, P-TYPE3

SVA0 bit (10-bit Address LSB)

When the 10-bit address format is selected (SARUy.FS = 1), the SVA0 bit functions as the LSB of a 10-bit address and is combined with the SVA[6:0] bits to form the lower 8 bits of a 10-bit address.

This bit is valid when the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1. When the SARUy.FS or SARyE bit is 0, the setting in this bit is ignored.

SVA[6:0] bits (7-bit Address/10-bit Address Lower Bits)

When the 7-bit address format is selected (SARUy.FS = 0), the SVA[6:0] bits function as a 7-bit address. When the 10-bit address format is selected (SARUy.FS = 1), these bits combine with the SVA0 bit to form the lower 8 bits of a 10-bit address.

When the SARyE bit in ICSER is 0, the setting in these bits is ignored.

39.2.14 SARUy : Slave Address Register Uy (y = 0 to 2)

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 2)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 2)

Offset address: 0x0B + 0x02 × y

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SVA[1:0]		FS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FS	7-bit/10-bit Address Format Select 0: Select 7-bit address format 1: Select 10-bit address format	R/W
2:1	SVA[1:0]	10-bit Address Upper Bits Slave address setting.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

FS bit (7-bit/10-bit Address Format Select)

The FS bit selects 7- or 10-bit format for slave address y (in SARLy and SARUy).

When the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 0, the 7-bit address format is selected for slave address y, the SVA[6:0] setting in SARLy is valid, and the SVA[1:0] and SVA0 settings in SARLy are ignored.

When the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, the 10-bit address format is selected for slave address y and the SVA[1:0] and SARLy settings are valid.

When the SARyE bit in ICSER is 0 (SARLy and SARUy disabled), the SARUy.FS setting is invalid.

SVA[1:0] bits (10-bit Address Upper Bits)

When the 10-bit address format is selected (FS = 1), the SVA[1:0] bits function as the upper 2 bits of a 10-bit address.

These bits are valid when the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1. When the SARUy.FS or SARyE bit is 0, the setting in these bits is ignored.

39.2.15 ICBRL : I^{2}C Bus Bit Rate Low-Level Register

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 2)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 2)

Offset address: 0x10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	BRL[4:0]				
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
4:0	BRL[4:0]	Bit Rate Low-Level Period Low-level period of SCL clock.	R/W
7:5	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE3, P-TYPE3

BRL[4:0] bits (Bit Rate Low-Level Period)

The BRL[4:0] bits set the low-level period of the SCL clock. ICBRL counts the low-level period with the internal reference clock source (IICφ) specified by the CKS[2:0] bits in ICMR1. ICBRL also generates the data setup time for automatic SCL low-hold operation, see section 39.9. Automatic Low-Hold Function for SCL. When the IIC is used in slave mode, the BRL[4:0] bits must be set to a value longer than the data setup time[*1].

If the digital noise filter is enabled (NFE bit in ICFER is 1), set the BRL[4:0] bits to a value at least one greater than the number of stages in the noise filter. For details on the number of stages, see the description of the NF[1:0] bits in section 39.2.5. ICMR3 : I²C Bus Mode Register 3.

Note 1. Data setup time (t_{SU}: DAT)

250 ns for up to 100 kbps: Standard-mode (Sm)

100 ns for up to 400 kbps: Fast-mode (Fm)

50 ns for up to 1 Mbps: Fast-mode Plus (Fm+)

39.2.16 ICBRH : I²C Bus Bit Rate High-Level Register

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 2)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 2)

Offset address: 0x11

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	BRH[4:0]				
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
4:0	BRH[4:0]	Bit Rate High-Level Period High-level period of SCL clock.	R/W
7:5	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE3, P-TYPE3

BRH[4:0] bits (Bit Rate High-Level Period)

The BRH[4:0] bits set the high-level period of SCL clock. BRH[4:0] bits are valid in master mode. If the IIC is used only in slave mode, do not set the BRH[4:0] bits.

ICBRH counts the high-level period with the internal reference clock source (IICφ) specified in the CKS[2:0] bits in ICMR1.

If the digital noise filter is enabled (the NFE bit in ICFER is 1), set these bits to a value at least one greater than the number of stages in the noise filter. For the number of stages in the noise filter, see the description of the NF[1:0] bits in section 39.2.5. ICMR3 : I²C Bus Mode Register 3.

The IIC transfer rate and the SCL clock duty are calculated using the following expressions (1) to (5):

1. ICFER.SCLE = 0
   Transfer rate = $1/[\{(BRH + 1) + (BRL + 1)\}/IIC\varphi^{*1} + tr^{*2} + tf^{*2}]$
   Duty cycle = $[tr + \{(BRH + 1)/IIC\varphi\}]/[tr + tf + \{(BRH + 1) + (BRL + 1)\}/IIC\varphi]$
2. ICFER.SCLE = 1 and ICFER.NFE = 0 and CKS[2:0] = 000b (IICφ = PCLKB)
   Transfer rate = $1/[\{(BRH + 3) + (BRL+ 3)\}/IIC\varphi + tr + tf]$
   Duty cycle = $[tr + \{(BRH + 3)/IIC\varphi\}]/[tr + tf + \{(BRH + 3) + (BRL + 3)\}/IIC\varphi]$
3. ICFER.SCLE = 1 and ICFER.NFE = 1 and CKS[2:0] = 000b (IICφ = PCLKB)
   Transfer rate = $1/[\{(BRH + 3 + nf^{*3}) + (BRL + 3 + nf)\}/IIC\varphi + tr + tf]$
   Duty cycle = $[tr + \{(BRH + 3 + nf)/IIC\varphi\}]/[tr + tf + \{(BRH + 3 + nf) + (BRL + 3 + nf)\}/IIC\varphi]$
4. ICFER.SCLE = 1 and ICFER.NFE = 0 and CKS[2:0] ≠ 000b
   Transfer rate = $1/[\{(BRH + 2) + (BRL + 2)\}/IIC\varphi + tr + tf]$
   Duty cycle = $[tr + \{(BRH + 2)/IIC\varphi\}]/[tr + tf + \{(BRH + 2) + (BRL + 2)\}/IIC\varphi]$
5. ICFER.SCLE = 1 and ICFER.NFE = 1 and CKS[2:0] ≠ 000b
   Transfer rate = $1/[\{(BRH + 2 + nf) + (BRL + 2 + nf)\}/IIC\varphi + tr + tf]$
   Duty cycle = $[tr + \{(BRH + 2 + nf)/IIC\varphi\}]/[tr + tf + \{(BRH + 2 + nf) + (BRL + 2 + nf)\}/IIC\varphi]$

Note 1. IICφ = PCLKB × division ratio
Note 2. The SCLn line rise time (tr) and SCLn line fall time (tf) depend on the total bus line capacitance (Cb) and the pull-up resistor (Rp). For details, see the I2C bus standard from NXP Semiconductors.
Note 3. nf = Number of digital noise filters selected in the ICMR3.NF bit.

Table 39.6 Example of ICBRH/ICBRL Settings for Transfer Rate IIC when SCLE = 0

Transfer rate (kbps)	CKS[2:0] (ICMR1)	BRH[4:0] (ICBRH)	BRL[4:0] (ICBRL)	PCLKB (MHz)	NF[1:0]	Computation expression
100	011b	24 (0xF8)	30 (0xFE)	50	—	(1)
400	010b	7 (0xE7)	15 (0xEF)	50	—	(1)
1000	000b	12 (0xEC)	24 (0xF8)	50	—	(1)

Table 39.7 Example of ICBRH/ICBRL Settings for Transfer Rate when SCLE = 1 and NFE = 0

Transfer rate (kbps)	CKS[2:0] (ICMR1)	BRH[4:0] (ICBRH)	BRL[4:0] (ICBRL)	PCLKB (MHz)	NF[1:0]	Computation expression
100	100b	11 (0xEB)	13 (0xED)	50	—	(4)
400	001b	13 (0xED)	31 (0xFF)	50	—	(4)
1000	000b	10 (0xEA)	22 (0xF6)	50	—	(2)

Table 39.8 Example of ICBRH/ICBRL Settings for Transfer Rate when SCLE = 1 and NFE = 1

Transfer rate (kbps)	CKS[2:0] (ICMR1)	BRH[4:0] (ICBRH)	BRL[4:0] (ICBRL)	PCLKB (MHz)	NF[1:0]	Computation expression
100	011b	21 (0xF5)	26 (0xFA)	50	01b	(5)
400	001b	11 (0xEB)	29 (0xFD)	50	01b	(5)
1000	000b	8 (0xE8)	20 (0xF4)	50	01b	(3)

39.2.17 ICDRT : I2C Bus Transmit Data Register

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 2)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 2)

Offset address: 0x12

Bit position:	7	6	5	4	3	2	1	0
Bit field:								
Value after reset:	1	1	1	1	1	1	1	1

When ICDRT detects a space in the I2C Bus Shift Register (ICDRS), it transfers the transmit data that was written to ICDRT to ICDRS and starts transmitting data in transmit mode. The double-buffer structure of ICDRT and ICDRS allows continuous transmit operation if the next transmit data is written to ICDRT while the ICDRS data is being transmitted.

ICDRT can always be read and written to. Write transmit data to ICDRT once when a transmit data empty interrupt (IICn_TXI) request is generated.

39.2.18 ICDRR : I2C Bus Receive Data Register

Base address: IICn = 0x4025_E000 + 0x0100 × n (n = 0 to 2)
IICn_NS = 0x5025_E000 + 0x0100 × n (n = 0 to 2)

Offset address: 0x13

Bit position:	7	6	5	4	3	2	1	0
Bit field:								
Value after reset:	0	0	0	0	0	0	0	0

When 1 byte of data is received, the received data is transferred from the I^{2}C Bus Shift Register (ICDRS) to ICDRR to enable the next data to be received. The double-buffer structure of ICDRS and ICDRR allows continuous receive operation if the received data is read from ICDRR while ICDRS is receiving data. ICDRR cannot be written to. Read data from ICDRR once when a receive data full interrupt (IICn_RXI) request is generated.

If ICDRR receives the next receive data before the current data is read from ICDRR (while the RDRF flag in ICSR2 is 1), the IIC automatically holds the SCL clock low 1 cycle before the RDRF flag is set to 1 next.

39.2.19 ICDRS : I^{2}C Bus Shift Register

Base address: n/a

Offset address: n/a

Bit position:	7	6	5	4	3	2	1	0
Bit field:								
Value after reset:	—	—	—	—	—	—	—	—

ICDRS is an 8-bit shift register for data transmit and receive. During transmission, transmit data is transferred from ICDRT to ICDRS and is transmitted from the SDAn pin. During reception, data is transferred from ICDRS to ICDRR after 1 byte of data is received. ICDRS cannot be accessed directly.

39.3 Operation

39.3.1 Communication Data Format

The I^{2}C bus format consists of 8-bit data and 1-bit acknowledge. The frame following a start or restart condition is an address frame that specifies a slave device with which the master device communicates. The specified slave is valid until a new slave is specified or a stop condition is issued.

Figure 39.3 shows the I^{2}C bus format, and Figure 39.4 shows the I^{2}C bus timing.

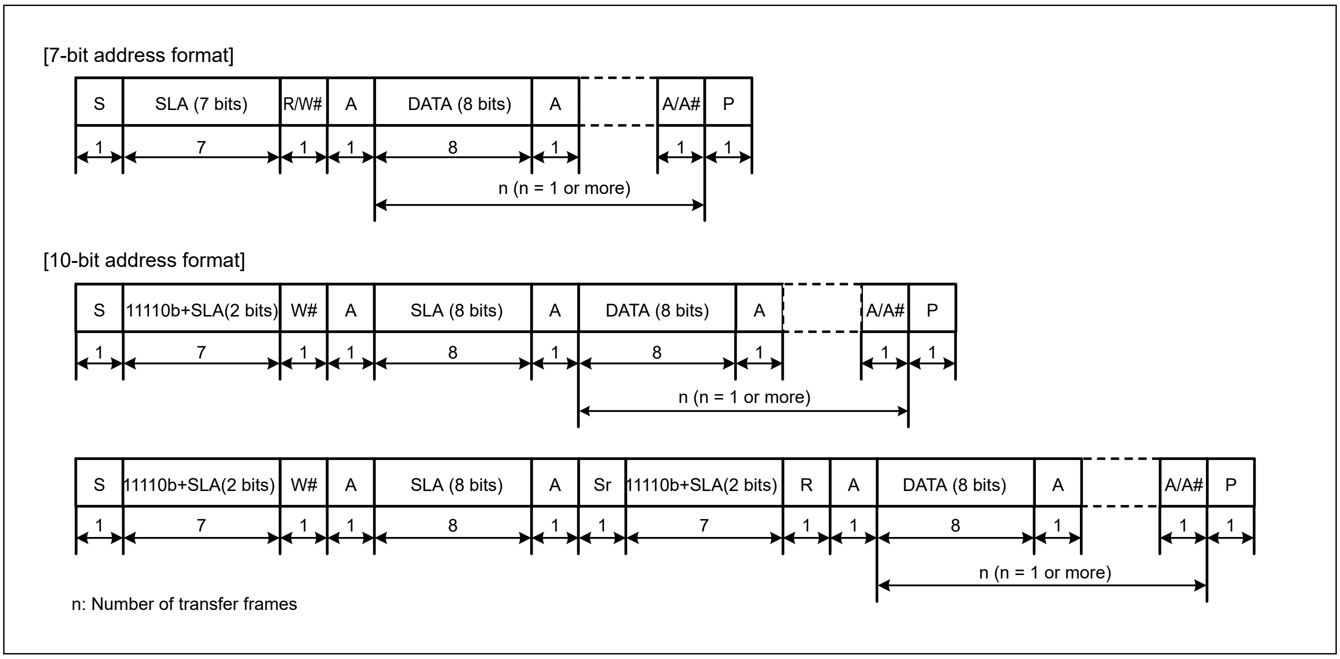
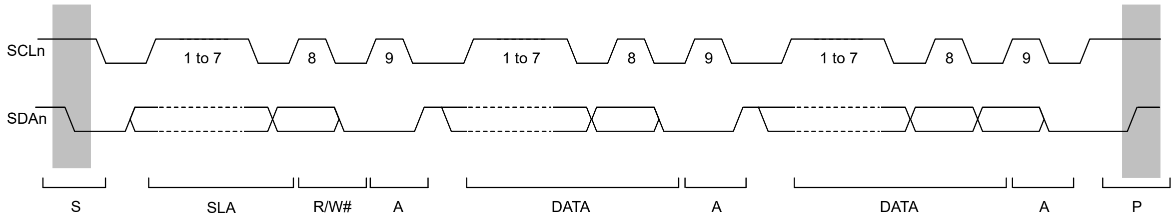


Figure 39.3 I^{2}C bus format



Note: n = 0, 1

Figure 39.4 I²C bus timing when the SLA setting = 7 bits

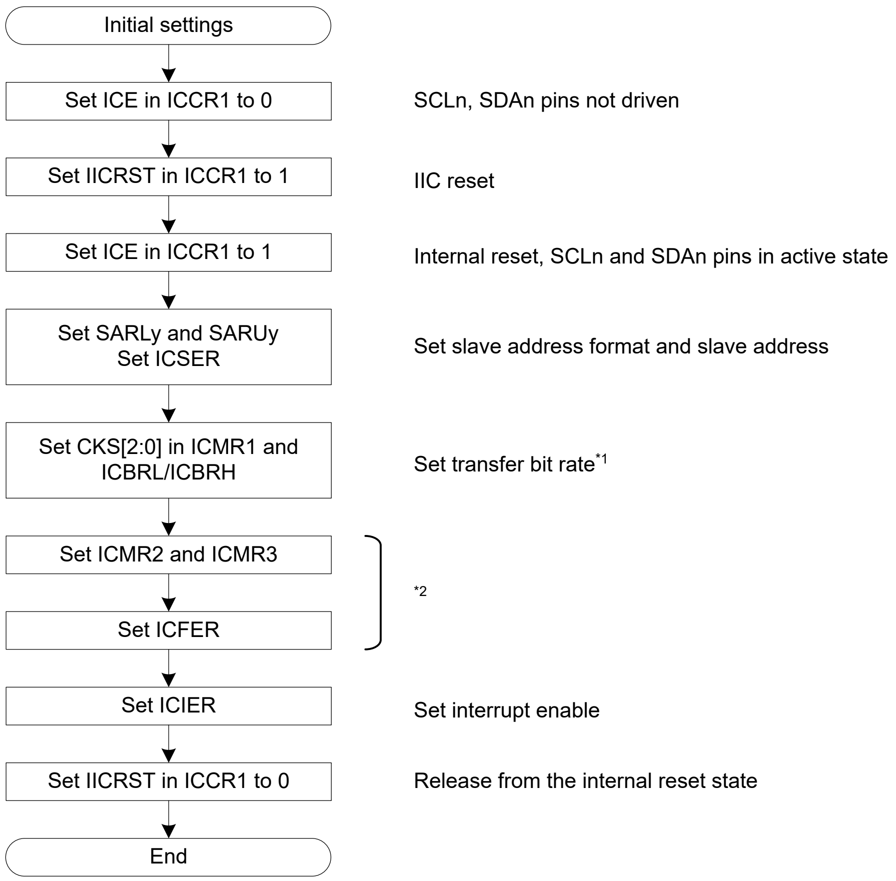
S:	Start condition. The master device drives the SDAn line low from high while the SCLn line is high.
SLA:	Slave address, by which the master device selects a slave device.
R/W#:	Indicates the direction of data transfer: from the slave device to the master device when R/W# is 1, or from the master device to the slave device when R/W# is 0.
A:	Acknowledge. The receive device drives the SDAn line low. In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.
A#:	Not Acknowledge. The receive device drives the SDAn line high.
Sr:	Restart condition. The master device drives the SDAn line low from the high level after the setup time has elapsed with the SCLn line high.
DATA:	Transmitted or received data.
P:	Stop condition. The master device drives the SDAn line high from low while the SCLn line is high.

39.3.2 Initial Settings

Before starting data transmission or reception, initialize the IIC using the procedure shown in Figure 39.5.

1. Set the ICCR1.ICE bit to 0 to set the SCLn and SDAn pins to the inactive state.
2. Set the ICCR1.IICRST bit to 1 to initiate IIC reset.
3. Set the ICCR1.ICE bit to 1 to initiate internal reset.
4. Set the SARLy, SARUy, ICSER, ICMR1, ICBRH, and ICBRL registers (y = 0 to 2), and set the other registers as required. For initial settings of the IIC, see Figure 39.5.
5. When the required register settings are complete, set the ICCR1.IICRST bit to 0 to release the IIC reset.

This procedure is not required if the IIC initialization is already complete.



Note: n = 0, 1, y = 0 to 2
Note 1. When the IIC is used only in slave mode, set the ICBRL register to a value longer than the data setup time.
Note 2. Set these registers as required.

Figure 39.5 Example IIC initialization flow

39.3.3 Master Transmit Operation

In master transmit operation, the IIC outputs the SCL clock and transmitted data signals as the master device, and the slave device returns acknowledgments. Figure 39.6 shows an example of master transmission, and Figure 39.7 to Figure 39.9 show the operation timing in master transmission.

To set up and perform master transmission:

1. Process initial settings. For details, see section 39.3.2. Initial Settings.
2. Read the BBSY flag in ICCR2 to check that the bus is free, and then set the ST bit in ICCR2 to 1 (start condition request). On receiving the request, the IIC issues a start condition. At the same time, the BBSY and START flags in ICSR2 automatically set to 1 and the ST bit automatically is set to 0. At this time, if the start condition is detected and the internal levels for the SDA output state and the levels on the SDAn line match while the ST bit is 1, the IIC recognizes that issuance of the start condition as requested by the ST bit is successfully complete, and the MST and TRS bits in ICCR2 automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 also automatically is set to 1 in response to the setting of the TRS bit to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDRT. When the transmit data is written to ICDRT, the TDRE flag automatically is set to 0, the data is transferred from ICDRT to ICDRS, and the TDRE flag again is set to 1. After the byte containing the slave address and R/W# bit is transmitted, the value of the TRS bit automatically updates to select master transmit or master receive mode based on the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the IIC continues in master transmit mode.

Because the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to issue a stop condition.
For data transmission with an address in the 10-bit format, start by writing 11110b, the 2 higher-order bits of the slave address, and W to ICDRT as the first address transmission. For the second address transmission, write the 8 lower-order bits of the slave address to ICDRT.

4. Check that the TDRE flag in ICSR2 is 1, and then write the transmit data to the ICDRT register. The IIC automatically holds the SCLn line low until the transmit data is ready or a stop condition is issued.
5. After all bytes of transmit data are written to the ICDRT register, wait until the value in the TEND flag in ICSR2 returns to 1, and then set the SP bit in ICCR2 to 1 (stop condition requested). On receiving a stop condition request, the IIC issues the stop condition. Regarding issuing a stop condition, see section 39.11.3. Issuing a Stop Condition.
6. On detecting the stop condition, the IIC automatically sets the MST and TRS bits in ICCR2 to 00b and enters slave receive mode. Additionally, it automatically sets the TDRE and TEND flags to 0, and sets the STOP flag in ICSR2 to 1.
7. Check that the ICSR2.STOP flag is 1, and then set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

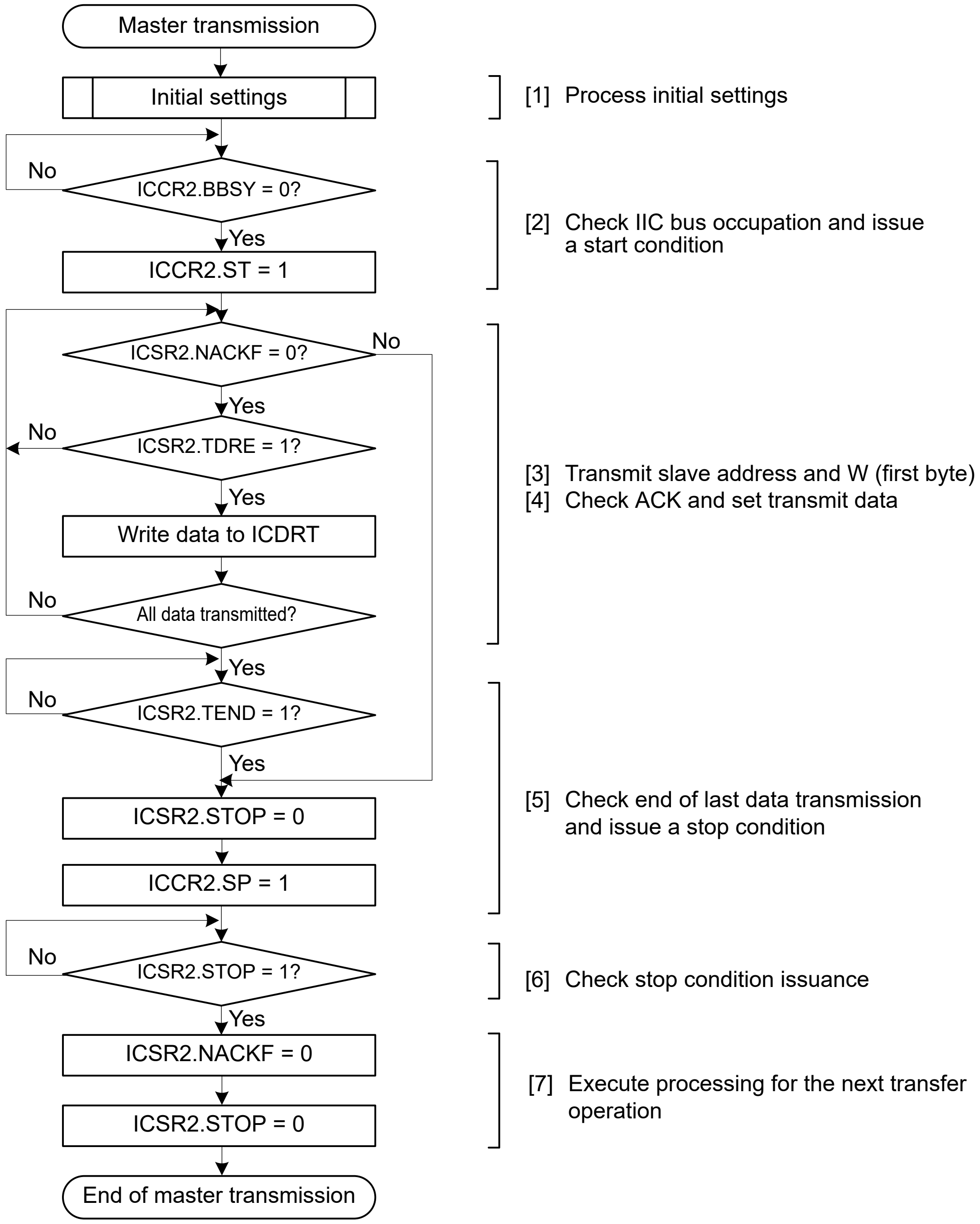


Figure 39.6 Example master transmission flow

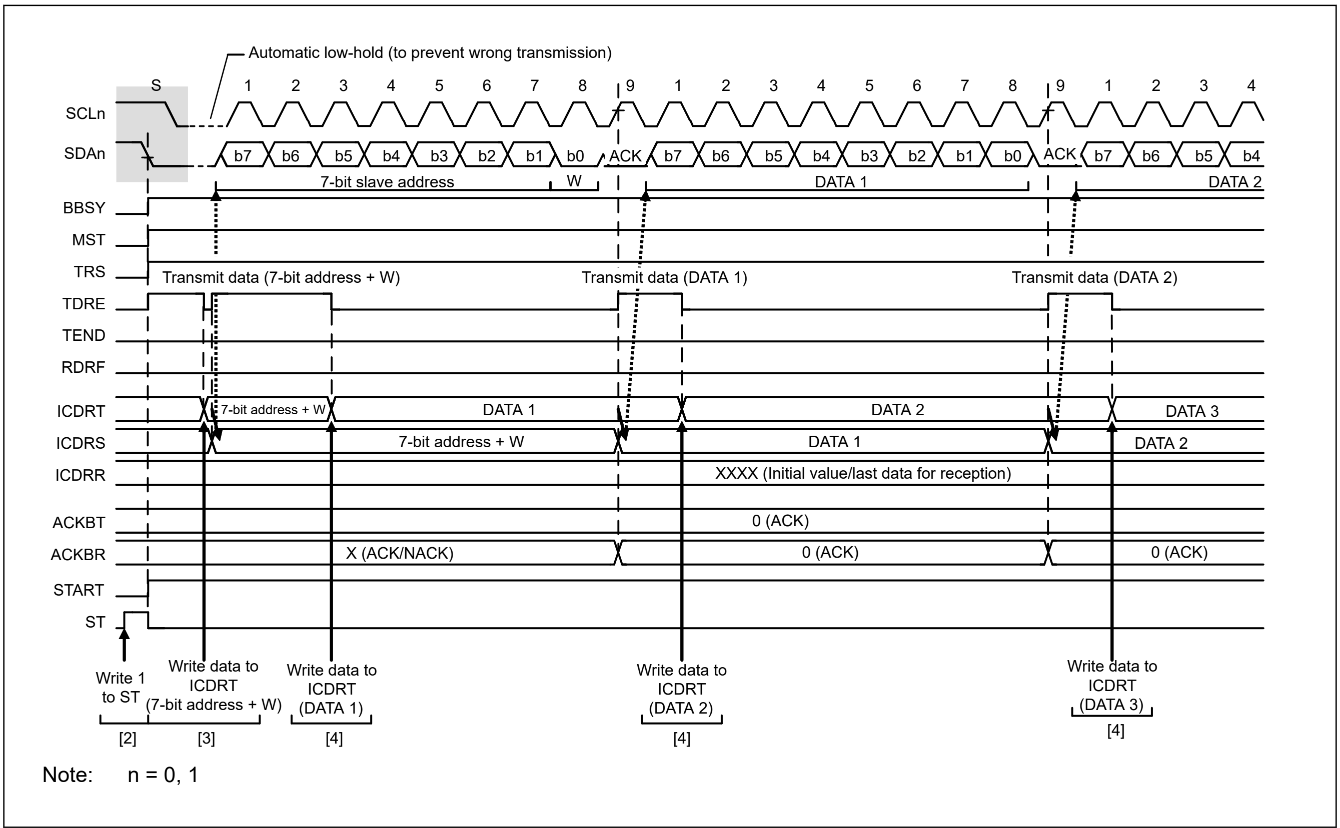


Figure 39.7 Master transmit operation timing (1) with 7-bit address format

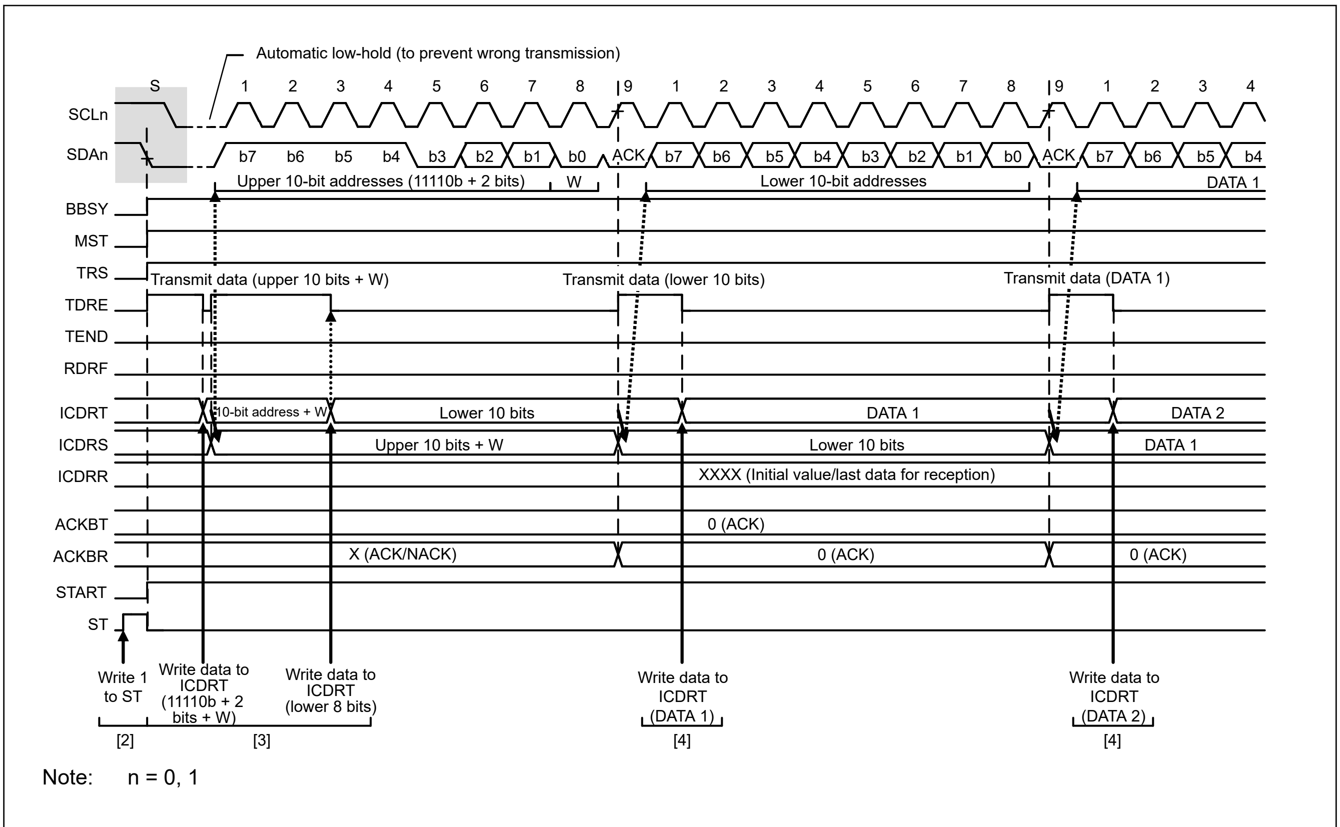
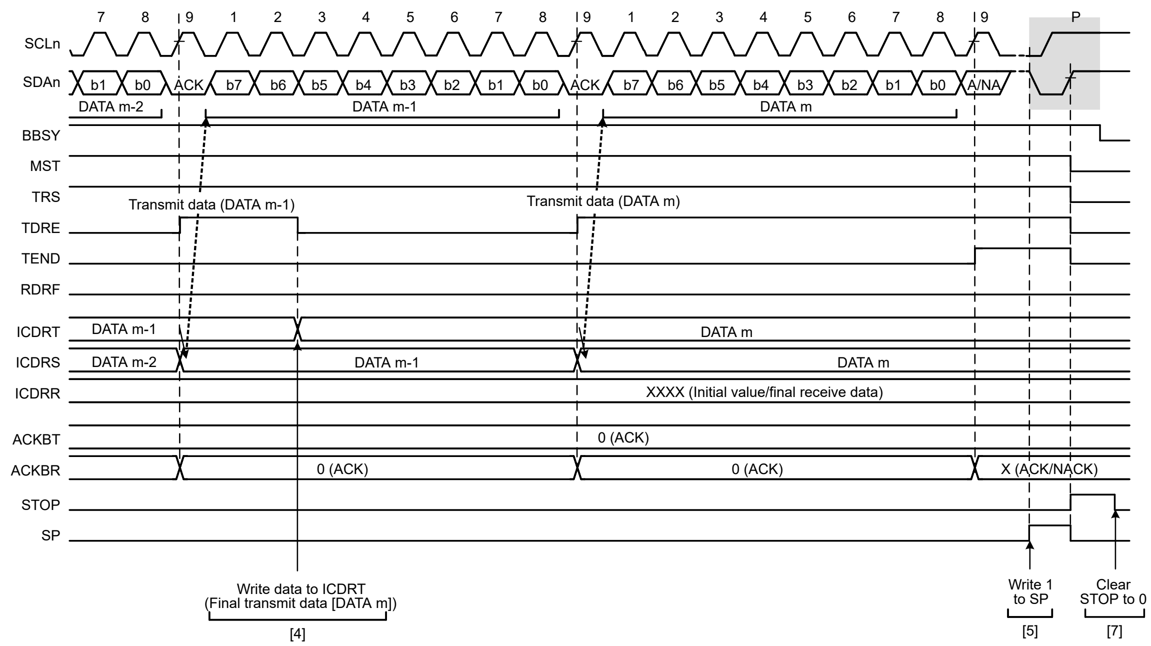


Figure 39.8 Master transmit operation timing (2) with 10-bit address format



Note: n = 0, 1

Figure 39.9 Master transmit operation timing (3)

39.3.4 Master Receive Operation

In master receive operation, the IIC as a master device outputs the SCL clock, receives data from the slave device, and returns acknowledgments. Because the IIC must start by sending a slave address to the associated slave device, this part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.

Figure 39.10 and Figure 39.11 show examples of master reception (7-bit address format), and Figure 39.12 to Figure 39.14 show the operation timing in master reception.

To set up and perform master reception:

1. Process initial settings. For details, see section 39.3.2. Initial Settings.
2. Read the BBSY flag in ICCR2 to check that the bus is free, and then set the ST bit in ICCR2 to 1 (start condition request). On receiving the request, the IIC issues a start condition. When the IIC detects the start condition, the BBSY and START flags in ICSR2 automatically set to 1, and the ST bit automatically is set to 0. At this time, if the start condition is detected and the levels for the SDA output and the levels on the SDAn line match while the ST bit is 1, the IIC recognizes that issuance of the start condition as requested by the ST bit is successfully complete, and the MST and TRS bits in ICCR2 automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 also automatically is set to 1 in response to the setting of the TRS bit to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to ICDRT. When the transmit data is written to ICDRT, the TDRE flag automatically is set to 0, the data is transferred from ICDRT to ICDRS, and the TDRE flag again is set to 1. When the byte containing the slave address and R/W# bit is transmitted, the value of the ICCR2.TRS bit automatically updates to select transmit or receive mode based on the value of the transmitted R/W# bit. If the value of the R/W# bit is 1, the TRS bit is set to 0 on the rising edge of the ninth cycle of the SCL clock, placing the IIC in master receive mode. At this time, the TDRE flag is set to 0 and the ICSR2.RDRF flag automatically is set to 1.
   Because the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to issue a stop condition.
   For master reception from a device with a 10-bit address, start by using master transmission to issue the 10-bit address, and then issue a restart condition. After that, transmitting 11110b, the two higher-order bits of the slave address, and the R bit places the IIC in master receive mode.

4. Dummy read ICDRR after confirming that the RDRF flag in ICSR2 is 1. This makes the IIC start output of the SCL clock and start data reception.
5. After 1 byte of data is received, the RDRF flag in ICSR2 is set to 1 on the rising edge of the 8th or 9th cycle of the SCL clock, as selected in the RDRFS bit in ICMR3. Reading ICDRR at this time produces the received data, and the RDRF flag is automatically set to 0 at the same time. Additionally, the value of the acknowledgment field received during the ninth cycle of the SCL clock is returned as the value set in the ICMR3.ACKBT bit. If the next byte to be received is the second-to-last byte, set the ICMR3.WAIT bit to 1 for wait insertion before reading ICDRR, containing the second-to-last byte. In addition to enabling NACK output, even when interrupts or other operations result in delays in setting the ICMR3.ACKBT bit to 1 (NACK) in step (6), this fixes the SCLn line to the low level on the rising edge of the ninth clock cycle in reception of the last byte, which enables the issuing of a stop condition.
6. When the ICMR3.RDRFS bit is 0, and the slave device must be notified that it is to end transfer for data reception after transfer of the next and final byte, set the ICMR3.ACKBT bit to 1 (NACK).
7. After reading the second-to-last byte from the ICDRR register, if the value of the ICSR2.RDRF flag is 1, write 1 to the SP bit in ICCR2 (stop condition requested), and then read the last byte from ICDRR. When ICDRR is read, the IIC is released from the wait state and issues the stop condition after low-level output in the ninth clock cycle is complete or the SCLn line is released from the low-hold state.
8. On detecting the stop condition, the IIC automatically sets the MST and TRS bits in ICCR2 to 00b and enters slave receive mode. Additionally, detection of the stop condition sets the ICSR2.STOP flag to 1.
9. Check that the ICSR2.STOP flag is 1, then set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

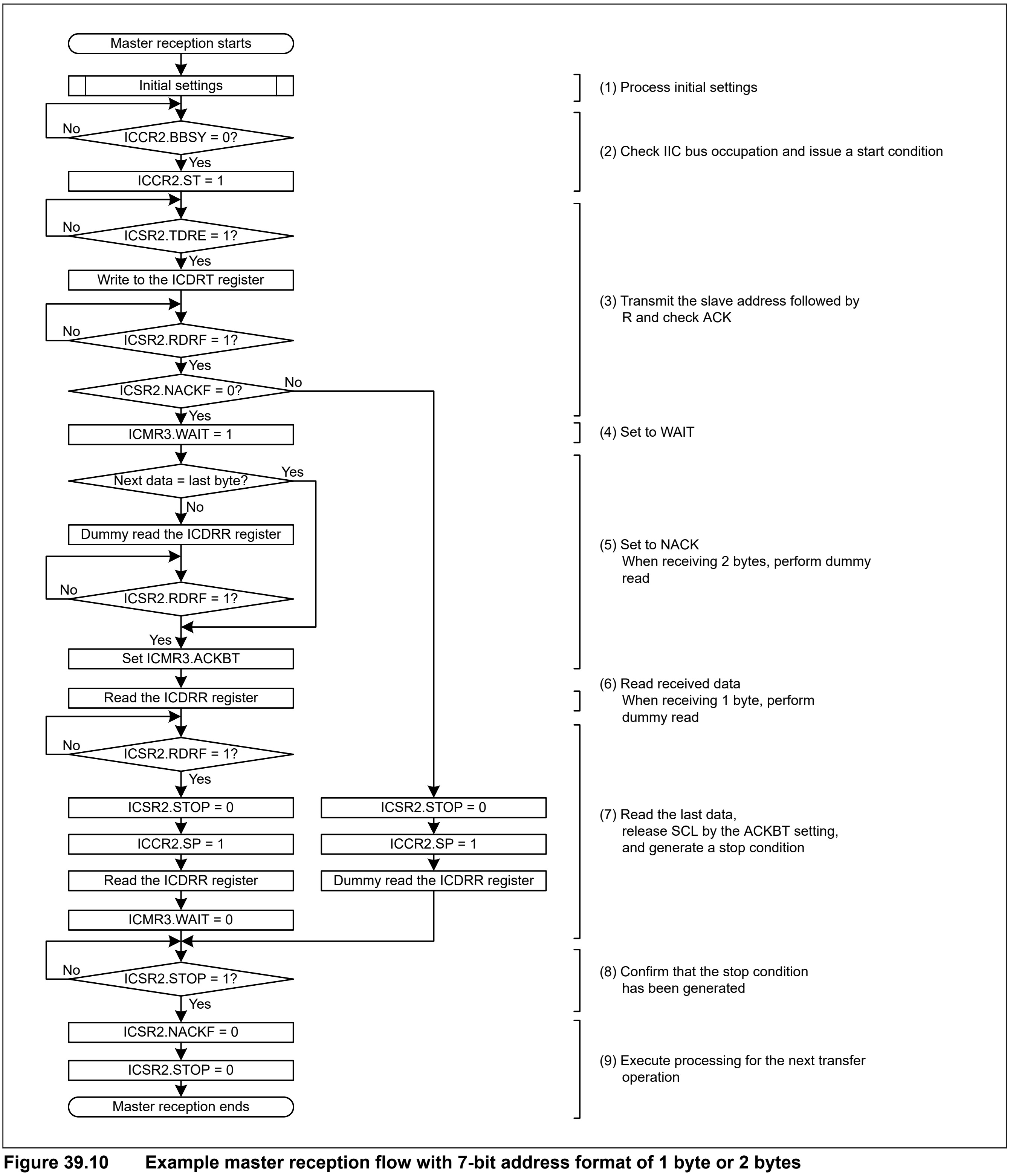


Figure 39.10 Example master reception flow with 7-bit address format of 1 byte or 2 bytes

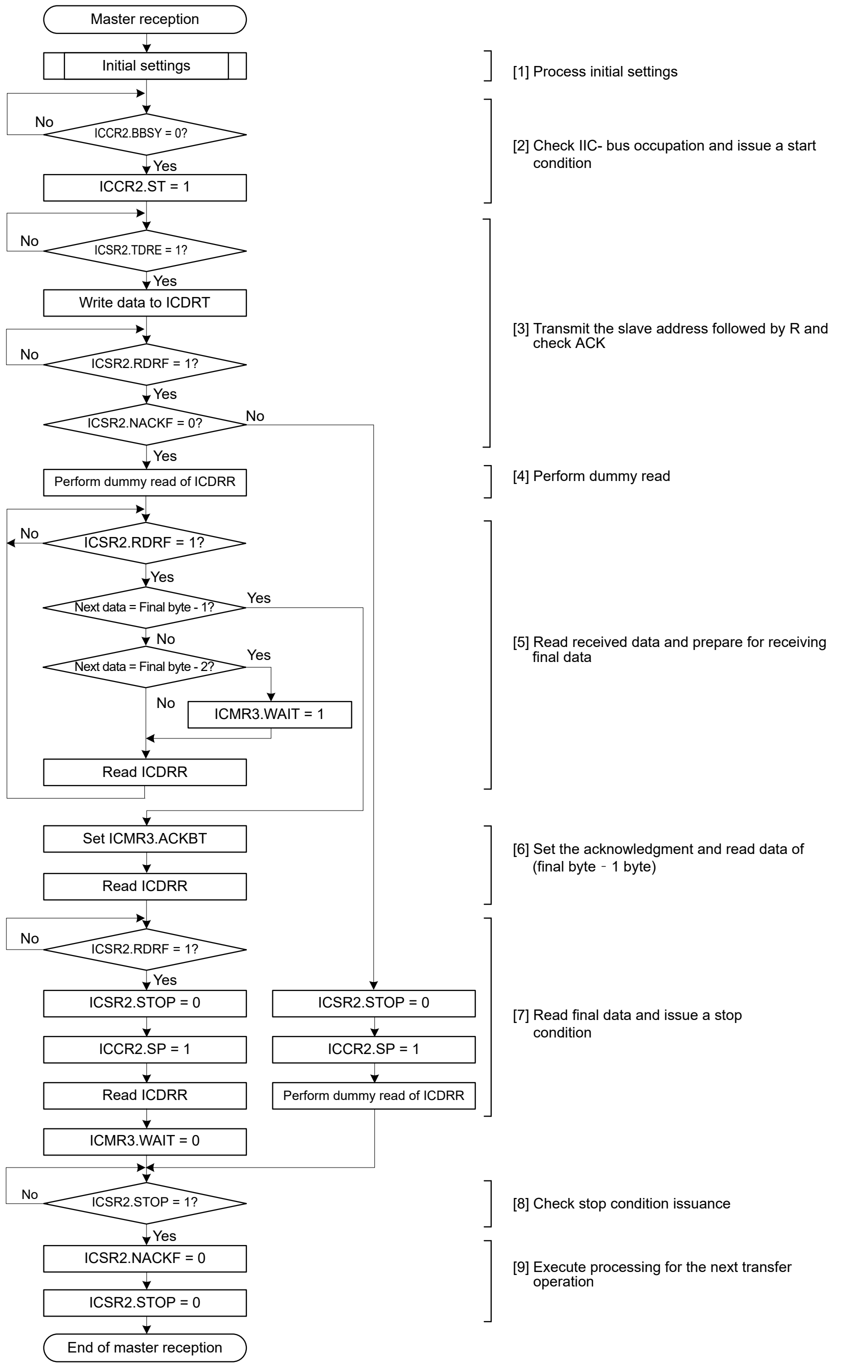
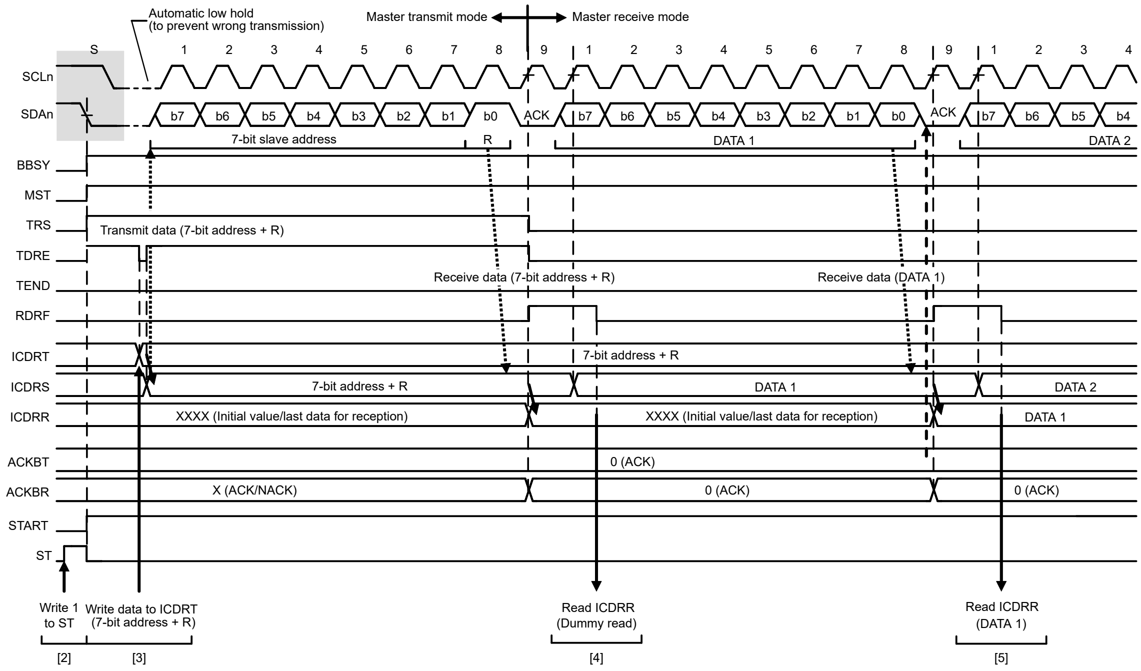
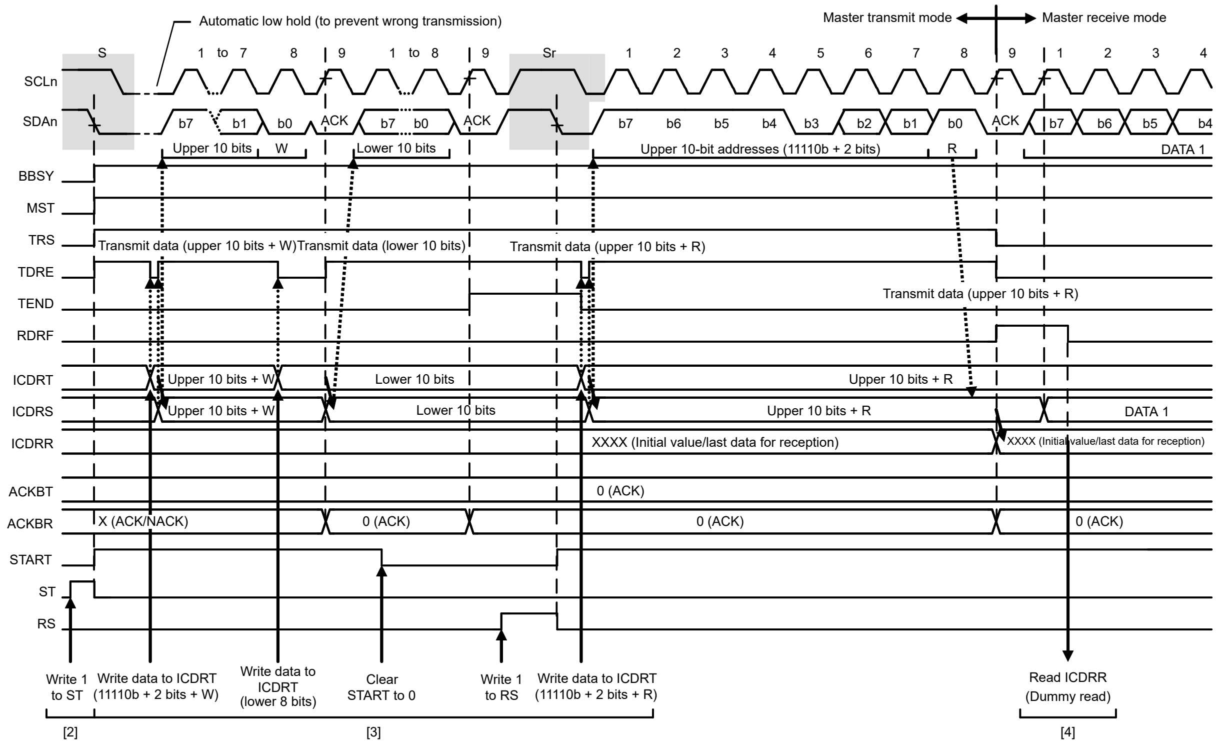


Figure 39.11 Example master reception flow with 7-bit address format of 3 or more bytes



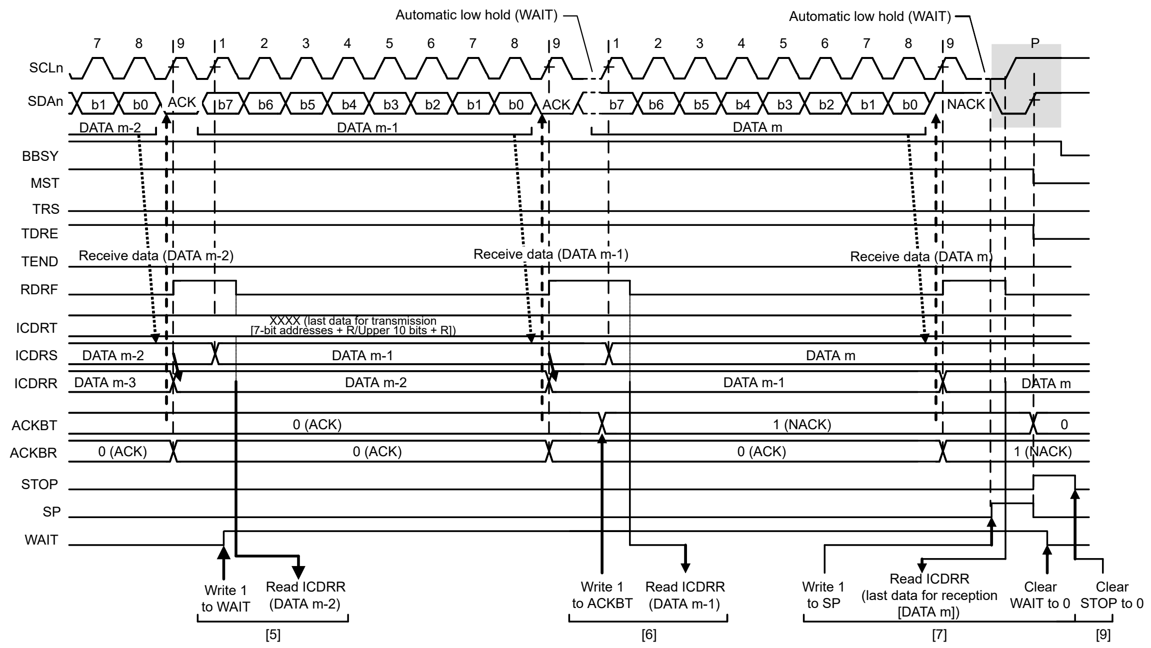
Note: n = 0, 1

Figure 39.12 Master receive operation timing (1) with 7-bit address format when RDRFS = 0



Note: n = 0, 1

Figure 39.13 Master receive operation timing (2) with 10-bit address format when RDRFS = 0



Note: n = 0, 1

Figure 39.14 Master receive operation timing (3) when RDRFS = 0

39.3.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL clock, the IIC transmits data as a slave device, and the master device returns acknowledgments.

Figure 39.15 shows an example of slave transmission, and Figure 39.16 and Figure 39.17 show the operation timing in slave transmission.

To set up and perform slave transmission:

1. Initialize the IIC using the procedure in section 39.3.2. Initial Settings.
   After initialization, the IIC stays in the standby state until it receives a slave address that matches.
2. After receiving a matching slave address, the IIC sets one of the associated ICSR1.HOA, GCA, and AASy flags (y = 0 to 2) to 1 on the rising edge of the ninth cycle of the SCL clock and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the ninth cycle of the SCL clock. If the value of the R/W# bit is 1, the IIC automatically places itself in slave transmit mode by setting both the ICCR2.TRS bit and the ICSR2.TDRE flag to 1.
3. Check that the ICSR2.TDRE flag is 1, then write the transmit data to the ICDRT register. If the IIC receives no acknowledge from the master device (receives an NACK signal) while the ICFER.NACKE bit is 1, the IIC suspends transfer of the next data.
4. Wait until the ICSR2.TEND flag is set to 1 while the ICSR2.TDRE flag is 1, after the ICSR2.NACKF flag is set to 1 or the last byte for transmission is written to the ICDRT register. When the ICSR2.NACKF flag or the TEND flag is 1, the IIC drives the SCLn line low on the ninth falling edge of the SCL clock.
5. When the ICSR2.NACKF or ICSR2.TEND flag is 1, dummy read ICDRR to complete the processing. This releases the SCLn line.
6. On detecting the stop condition, the IIC automatically sets the ICSR1.HOA, GCA, and AASy flags (y = 0 to 2), the ICSR2.TDRE and TEND flags, and the ICCR2.TRS bit to 0, and enters slave receive mode.
7. Check that the ICSR2.STOP flag is 1, then set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

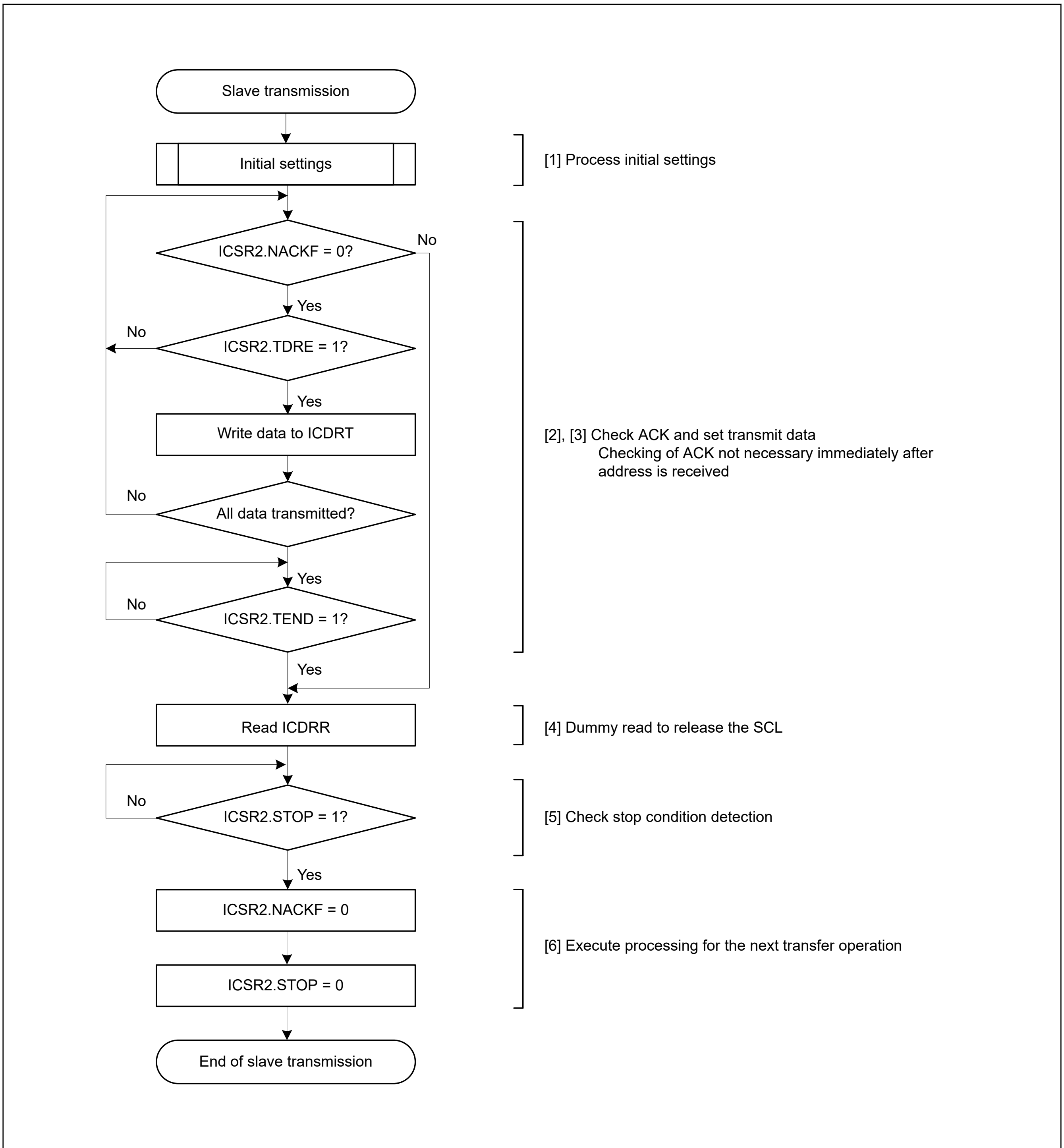


Figure 39.15 Example slave transmission flow

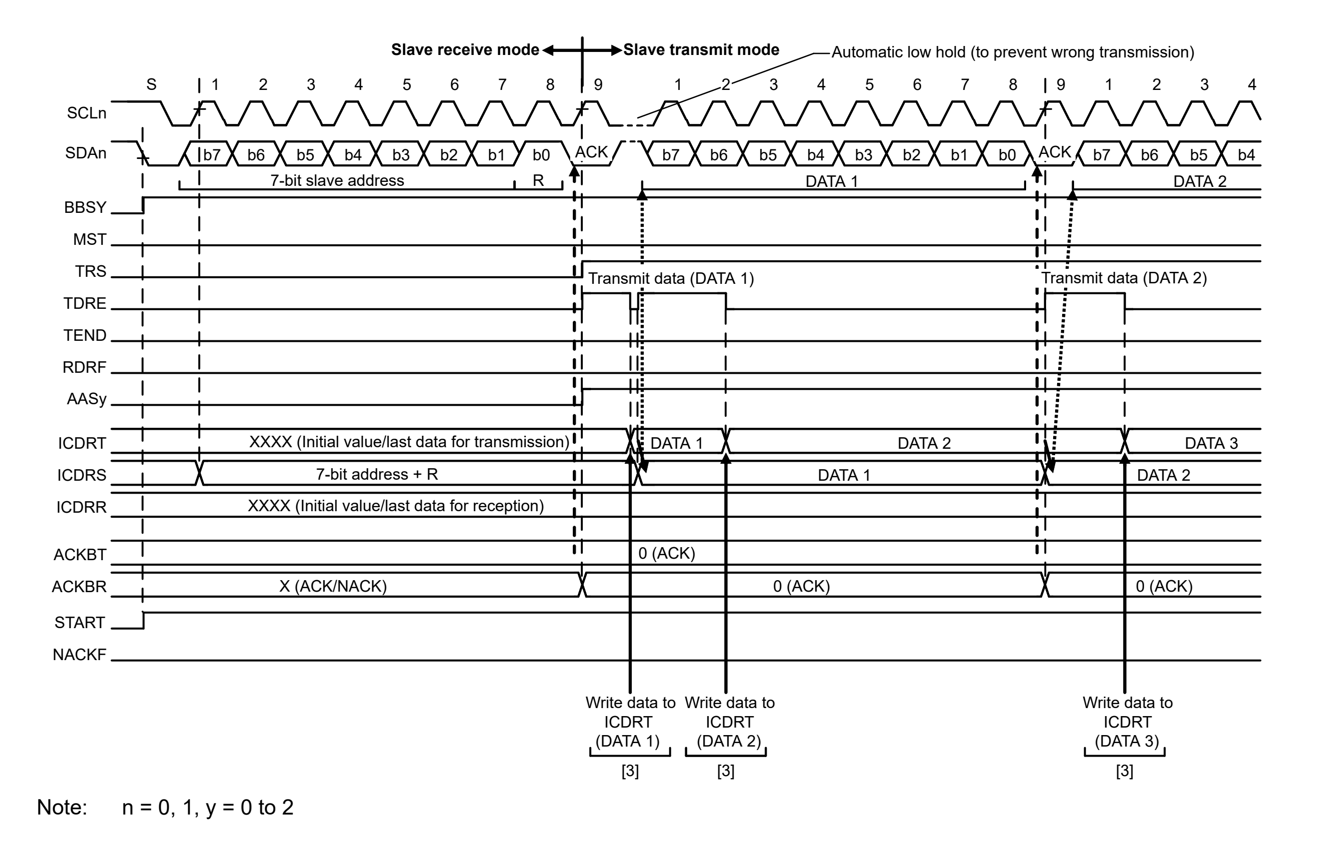


Figure 39.16 Slave transmit operation timing (1) with 7-bit address format

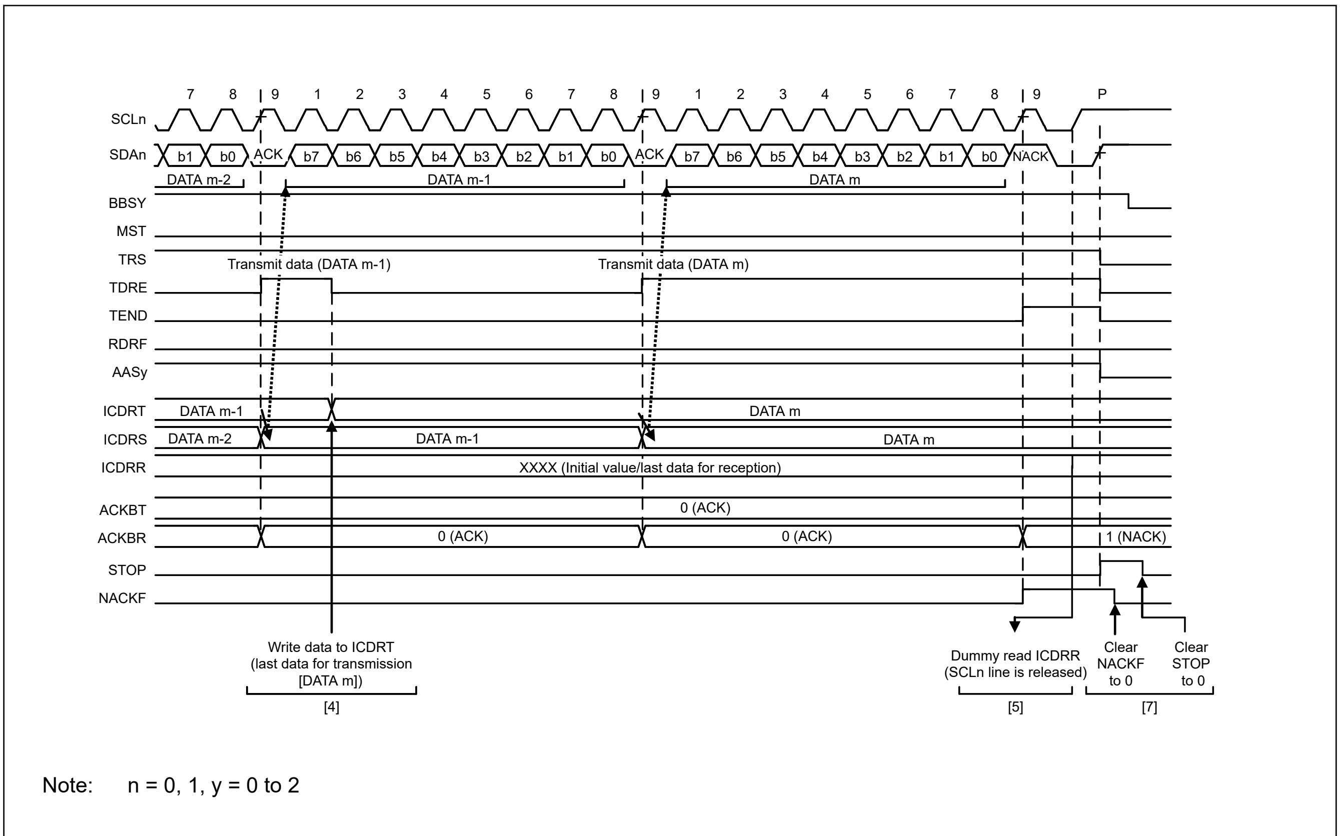


Figure 39.17 Slave transmit operation timing (2)

39.3.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL clock and transmit data, and the IIC returns acknowledgments as a slave device.

Figure 39.18 shows an example of slave reception, and Figure 39.19 and Figure 39.20 show the operation timing in slave reception.

To set up and perform slave reception:

1. Initialize the IIC using the procedure in section 39.3.2. Initial Settings.
   After initialization, the IIC stays in the standby state until it receives a slave address that matches.
2. After receiving a matching slave address, the IIC sets one of the associated ICSR1.HOA, GCA, and AASy flags (y = 0 to 2) to 1 on the rising edge of the ninth cycle of the SCL clock and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the ninth cycle of the SCL clock. If the value of the R/W# bit is 0, the IIC continues to place itself in slave receive mode and sets the RDRF flag in ICSR2 to 1.
3. Check that the ICSR2.STOP flag is 0 and the ICSR2.RDRF flag is 1, then dummy read ICDRR. The dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower 8 bits when the 10-bit address format is selected.
4. When ICDRR is read, the IIC automatically sets the ICSR2.RDRF flag to 0. If reading of ICDRR is delayed and a next byte is received while the RDRF flag is still set to 1, the IIC holds the SCLn line low until 1 SCL cycle before the point where RDRF must be set. In this case, reading ICDRR releases the SCLn line from being held low.
   When the ICSR2.STOP flag is 1 and the ICSR2.RDRF flag is also 1, read ICDRR until all the data is completely received.
5. On detecting the stop condition, the IIC automatically clears the ICSR1.HOA, GCA, and AASy flags (y = 0 to 2) to 0.
6. Check that the ICSR2.STOP flag is 1, then set the ICSR2.STOP flag to 0 for the next transfer operation.

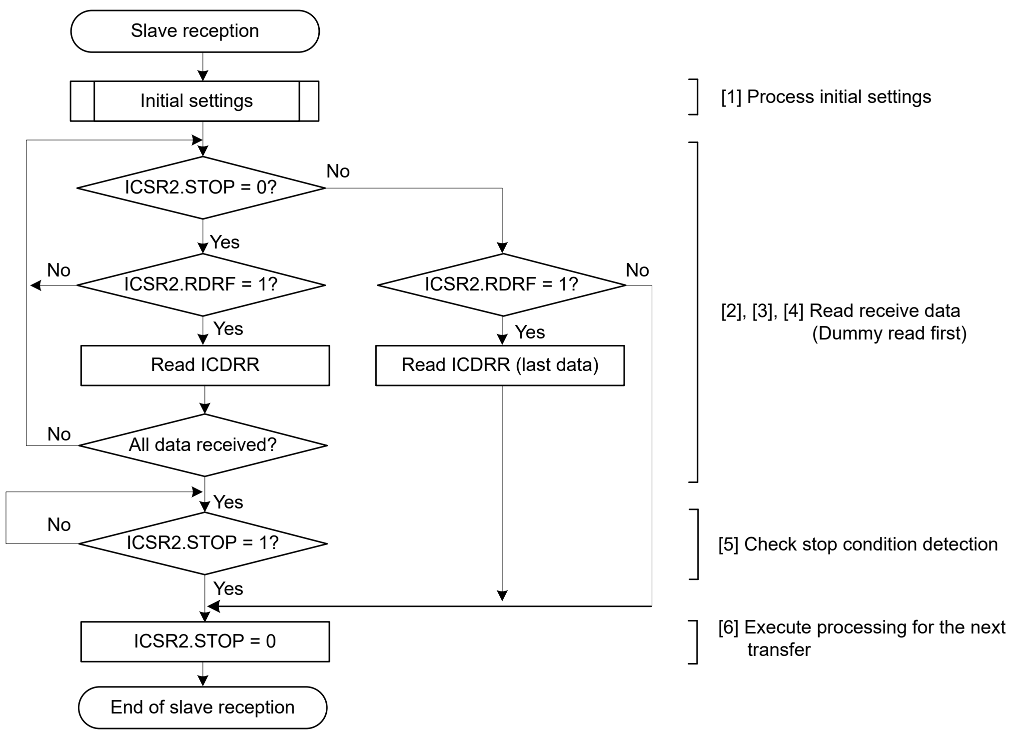
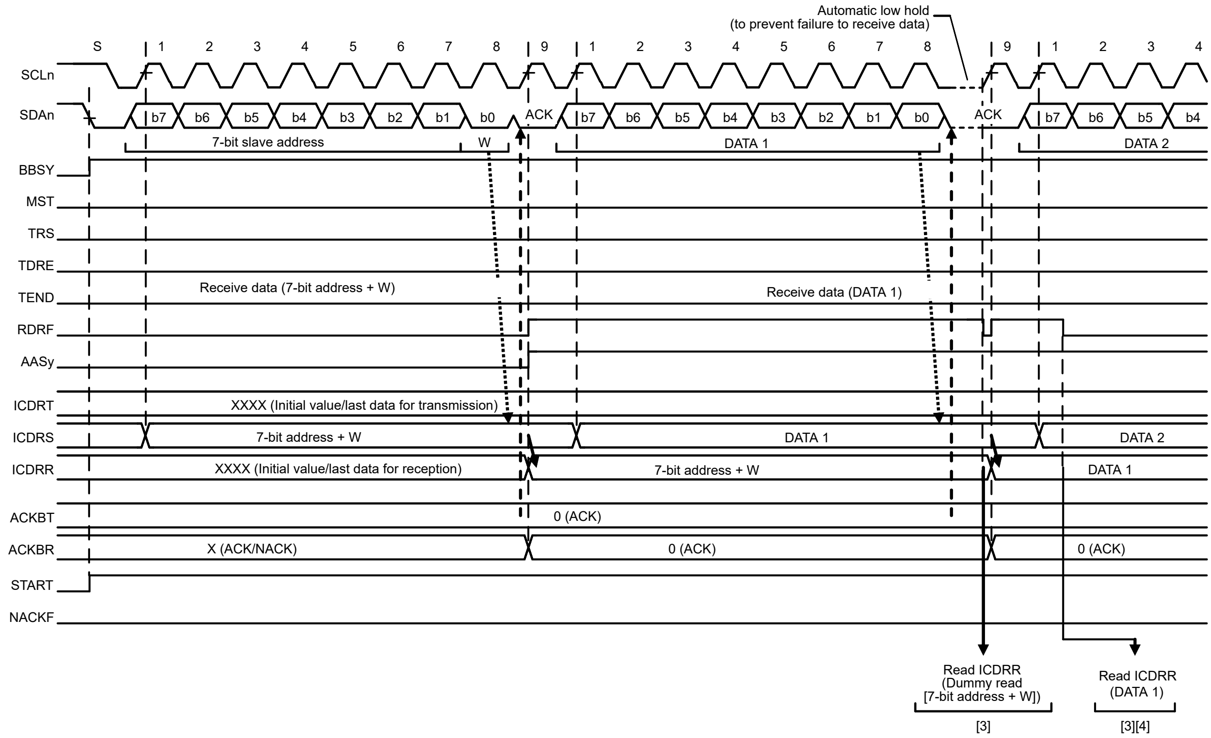
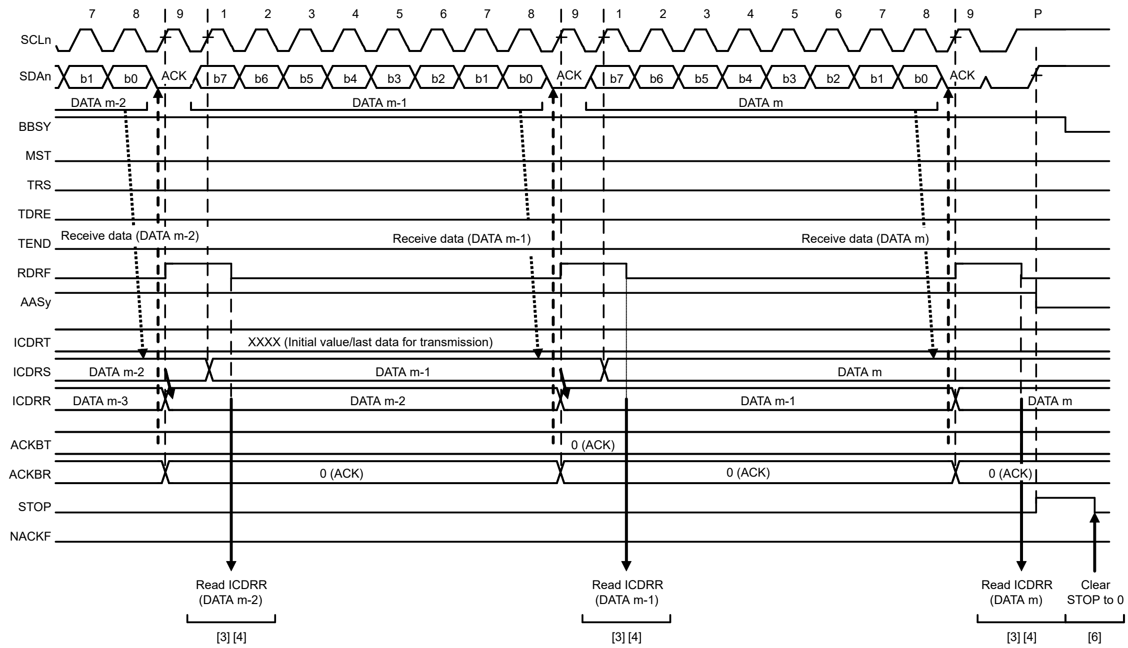


Figure 39.18 Example slave reception flow



Note: n = 0, 1, y = 0 to 2

Figure 39.19 Slave receive operation timing (1) with 7-bit address format when RDRFS = 0



Note: n = 0, 1, y = 0 to 2

Figure 39.20 Slave receive operation timing (2) when RDRFS = 0

39.4 SCL Synchronization Circuit

For generation of the SCL clock, the IIC starts counting the value for the high-level period specified in ICBRH when it detects a rising edge on the SCLn line, and it drives the SCLn line low when it completes counting. When the IIC detects the falling edge of the SCLn line, it starts counting the value for the low-level period specified in ICBRL, and then it stops driving the SCLn line, releasing the line, when it completes counting. The IIC repeats this process to generate the SCL clock.

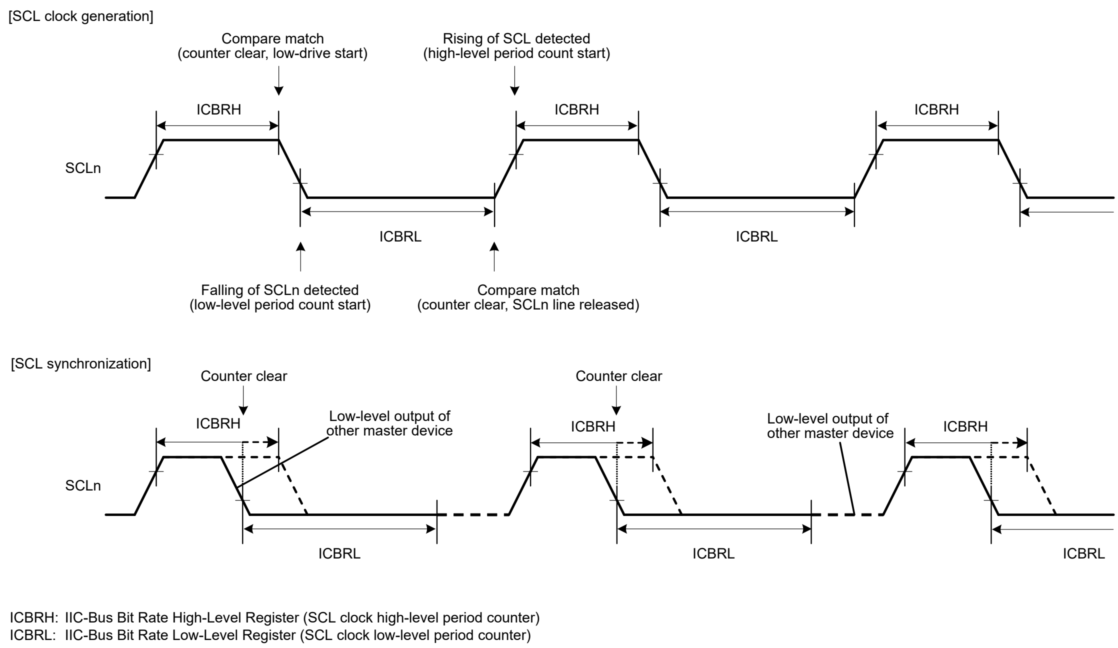
If multiple master devices are connected to the I^2C bus, a collision of SCL signals might arise because of contention with another master device. In such cases, the master devices must synchronize their SCL signals. Because this synchronization of SCL signals must be bit-by-bit, the IIC is equipped with an SCL synchronization circuit to obtain bit-by-bit synchronization of the SCL clock signals by monitoring the SCLn line while in master mode.

When the IIC detects a rising edge on the SCLn line and starts counting the high-level period specified in ICBRH.BRH[4:0], and the level on the SCLn line falls because an SCL signal is being generated by another master device, the IIC performs the following:

1. Stops counting when it detects the falling edge.
2. Drives the level on the SCLn line low.
3. Starts counting the low-level period specified in ICBRL.BRL[4:0].

When the IIC finishes counting the low-level period, it stops driving the SCLn line low to release the line. If the low-level period of the SCL clock signal from the other master device is longer than the low-level period set in the IIC, the low-level period of the SCL signal is extended. When the low-level period for the other master device ends, the SCL signal rises because the SCLn line is released.

When the IIC finishes outputting the low-level period of the SCL clock, the SCLn line is released and the SCL clock rises. That is, when SCL signals from more than one master are contending, the high-level period of the SCL signal is synchronized with that of the clock with the narrower period, and the low-level period of the SCL signal is synchronized with that of the clock with the broader period. However, such synchronization of the SCL signal is only enabled when the SCLE bit in ICFER is set to 1.



Note: n = 0, 1

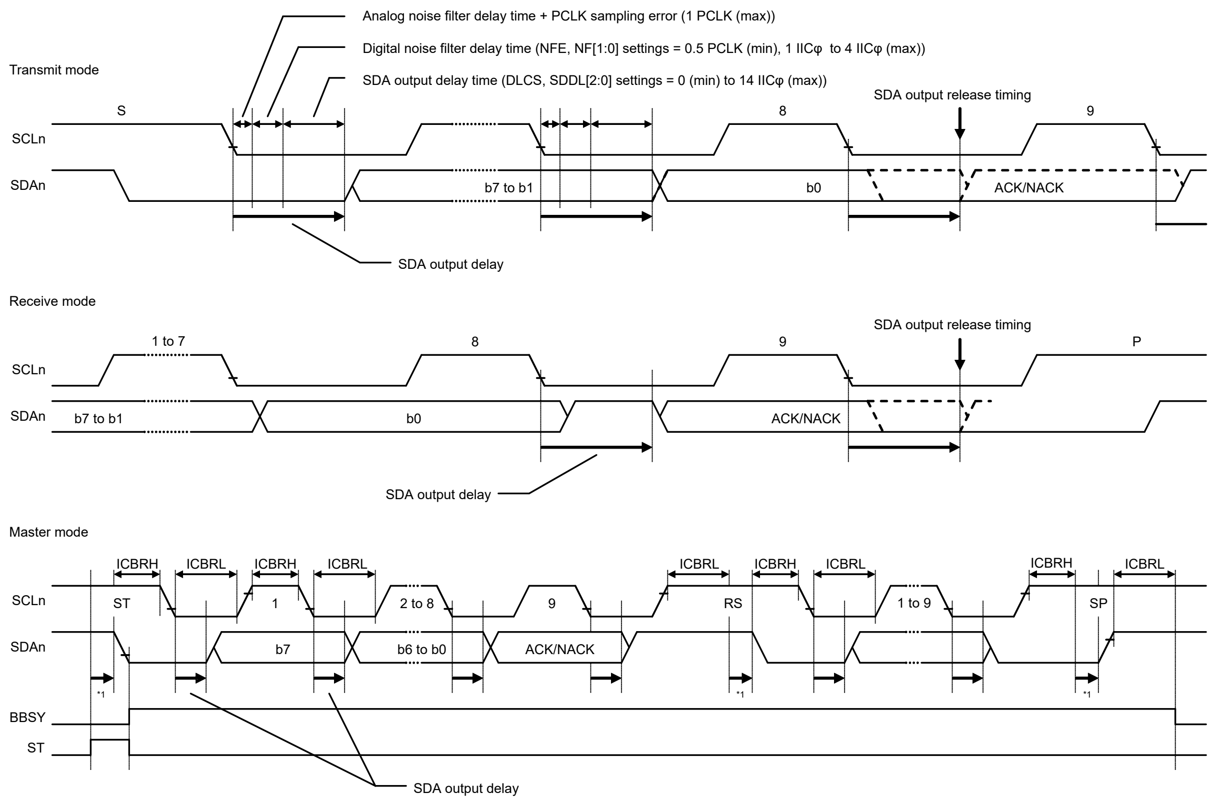
Figure 39.21 Generation and synchronization of SCL signal from IIC

39.5 SDA Output Delay Function

The IIC module incorporates a function for delaying output on the SDA line. The delay can be applied to all output on the SDA line, including issuing of the start, restart, and stop conditions, data, and the ACK and NACK signals.

With this function, SDA output is delayed from the detection of a falling edge of the SCL signal to ensure that the SDA signal is output within the interval during which the SCL clock is low. This approach helps prevent erroneous operation of communications devices, with the aim of satisfying the 300-ns minimum data-hold time requirement of the SMBus specification. The output delay function is enabled by setting the SDDL[2:0] bits in ICMR2 to a value other than 000b, and disabled by setting the same bits to 000b.

When the SDA output delay function is enabled, for example, the DLCS bit in ICMR2 selects the clock source for the SDA output delay counter, either as the internal base clock (IICφ) for the IIC module or as the internal base clock divided by 2 (IICφ/2). The counter counts the number of cycles set in the SDDL[2:0] bits in ICMR2. When the delay cycles count is reached, the IIC module places the required output (start, restart, or stop condition, data, or an ACK or NACK signal) on the SDA line.



Note: PCLK = PCLKB
Note: n = 0, 1
Note 1. The output delay function is set by the DLCS and SDDL[2:0] bits when a start (ST), restart (RS), or stop (SP) condition is issued.

Figure 39.22 SDA output delay function

39.6 Digital Noise Filter Circuits

The internal circuitry sees the states of the SCLn and SDAn pins through analog and digital noise-filter circuits. Figure 39.23 shows a block diagram of the digital noise-filter circuit.

The on-chip digital noise-filter circuit of the IIC consists of four flip-flop circuit stages connected in series and a match-detection circuit. The number of valid stages in the digital noise filter is selected in the NF[1:0] bits in ICMR3. The selected number of valid stages determines the noise-filtering capability as a period from 1 to 4 IICφ cycles.

The input signal to the SCLn pin (or SDAn pin) is sampled on falling edges of the IICφ signal. When the input signal level matches the output level of the number of valid flip-flop circuit stages as selected in the NF[1:0] bits in ICMR3, the signal level is seen in the subsequent stage. If the signal levels do not match, the previous value is saved.

If the ratio between the frequency of the internal operating clock (PCLKB) and the transfer rate is small, for example, if data transfer is 400 kbps with PCLKB = 4 MHz, the characteristics of the digital noise filter might lead to the elimination of required signals as noise. In such cases, it is possible to disable the digital noise-filter circuit by setting the ICFER.NFE bit to 0, and use only the analog noise filter circuit.

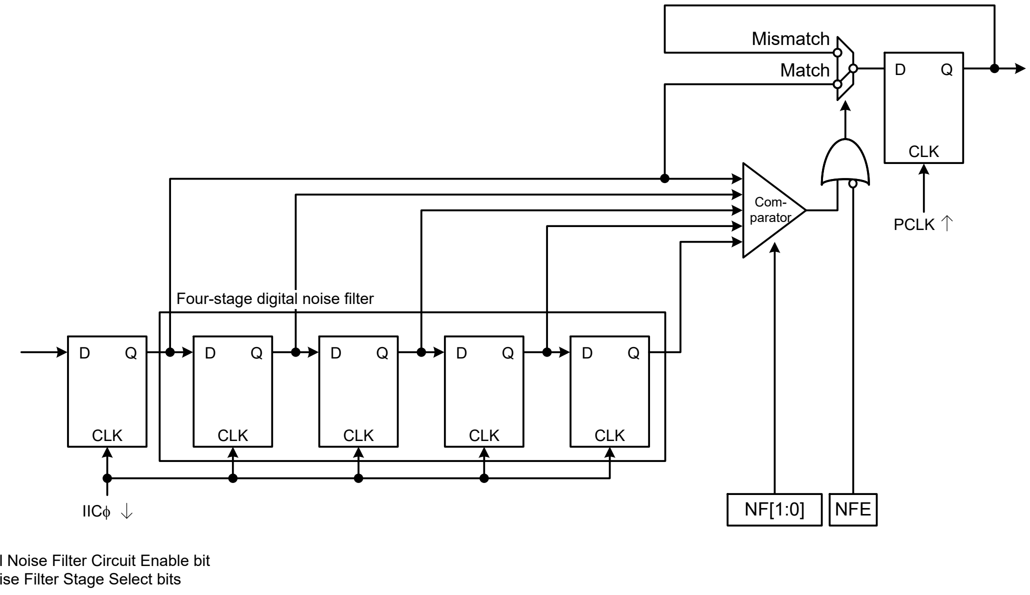


Figure 39.23 Digital noise filter circuit block diagram

39.7 Address Match Detection

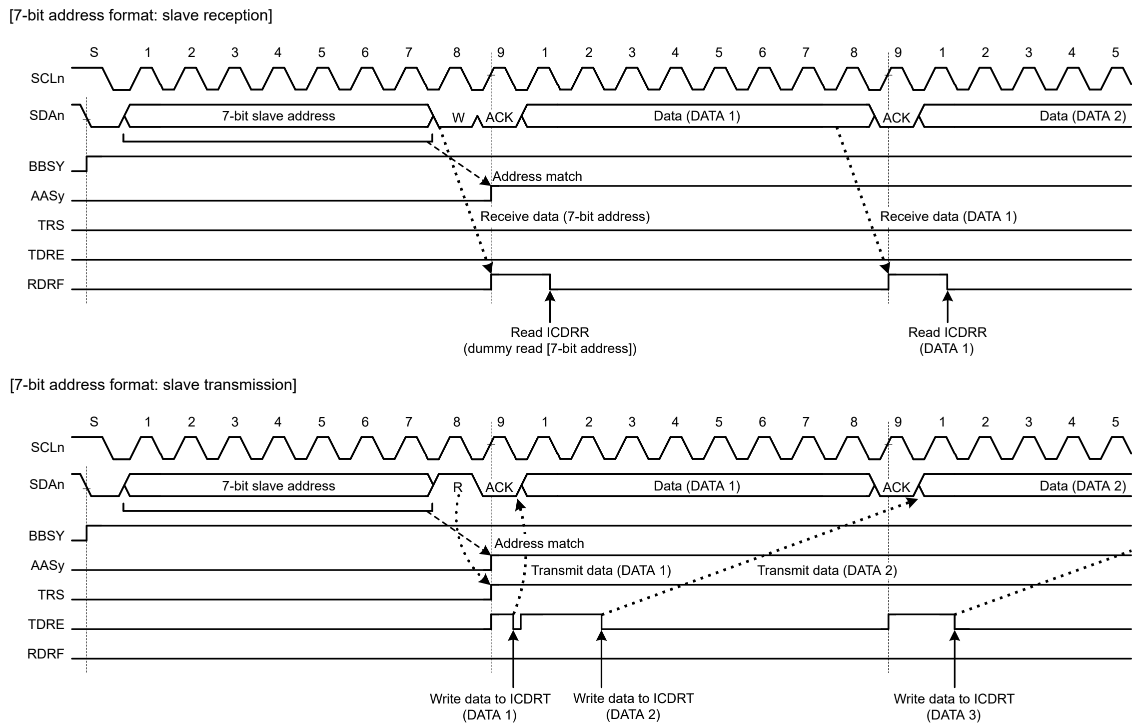
The IIC can set three unique slave addresses in addition to the general call address and host address. The slave addresses can be 7-bit or 10-bit slave addresses.

39.7.1 Slave-Address Match Detection

The IIC can set three unique slave addresses and has a slave address detection function for each unique slave address. When the SARyE bit (y = 0 to 2) in ICSER is set to 1, the slave addresses set in SARUy and SARLy (y = 0 to 2) can be detected.

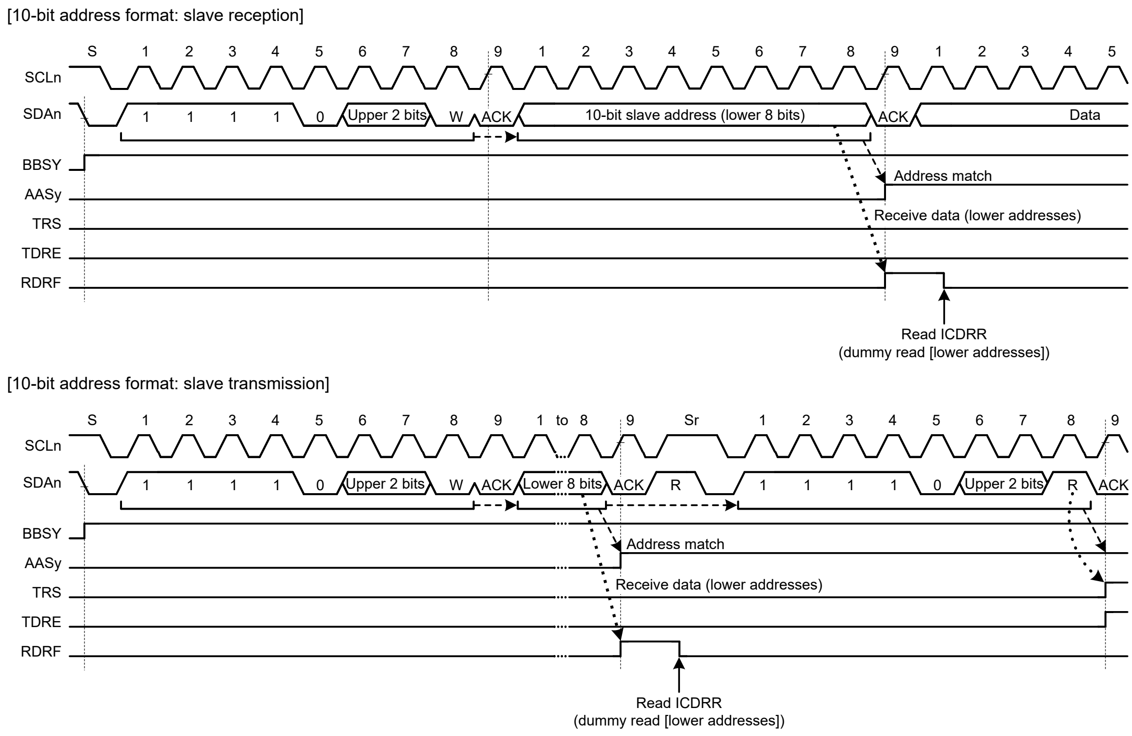
When the IIC detects a match of the set slave address, the associated AASy flag (y = 0 to 2) in ICSR1 is set to 1 on the rising edge of the ninth SCL clock cycle, and the RDRF flag in ICSR2 or the TDRE flag in ICSR2 is set to 1 by the subsequent R/W# bit. This causes a receive data full interrupt (IICn_RXI) or transmit data empty interrupt (IICn_TXI) to be generated. The AASy flag identifies which slave address is specified.

Figure 39.24 to Figure 39.26 show the AASy flag set timing in three cases.



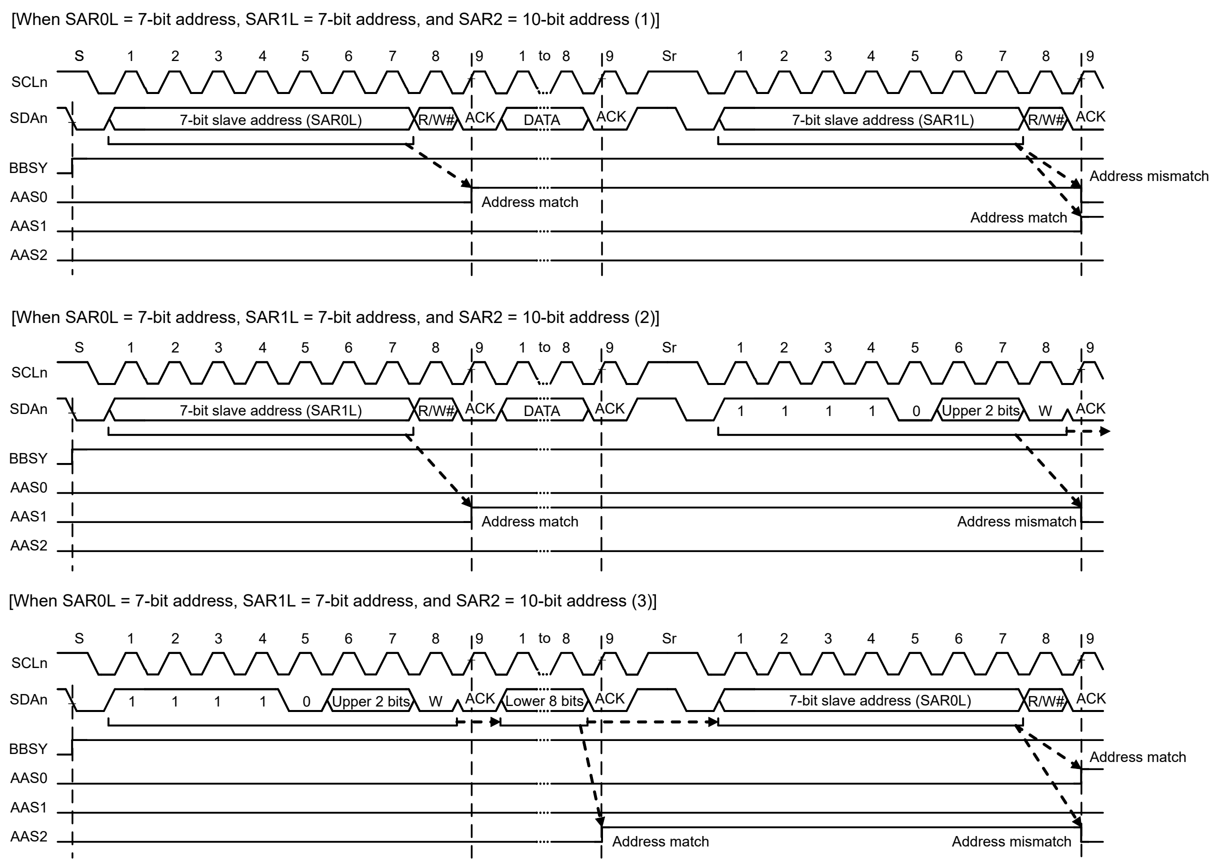
Note: n = 0, 1, y = 0 to 2

Figure 39.24 AASy flag set timing with 7-bit address format



Note: n = 0, 1, y = 0 to 2

Figure 39.25 AASy flag set timing with 10-bit address format



Note: n = 0, 1

Figure 39.26 AASy flag set and clear timing with mixed 7-bit and 10-bit address formats

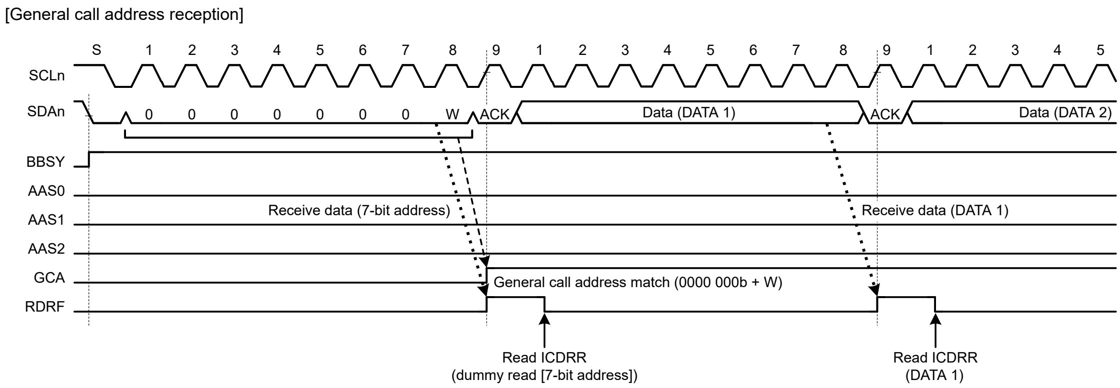
39.7.2 Detection of General Call Address

The IIC provides detection of the general call address (0000 000b + 0 [W]). This is enabled by setting the GCAE bit in ICSER to 1.

If the address received after a start or restart condition is issued is 0000 000b + 1[R] (start byte), the IIC recognizes this as the address of a slave device with an all-zero address, but not as the general call address.

When the IIC detects the general call address, both the GCA flag in ICSR1 and the RDRF flag in ICSR2 set to 1 on the rising edge of the ninth cycle of the SCL clock. This leads to the generation of a receive data full interrupt (IICn_RXI). The value of the GCA flag can be checked to confirm that the general call address was transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.



Note: n = 0, 1

Figure 39.27 Timing of GCA flag setting during reception of general call address

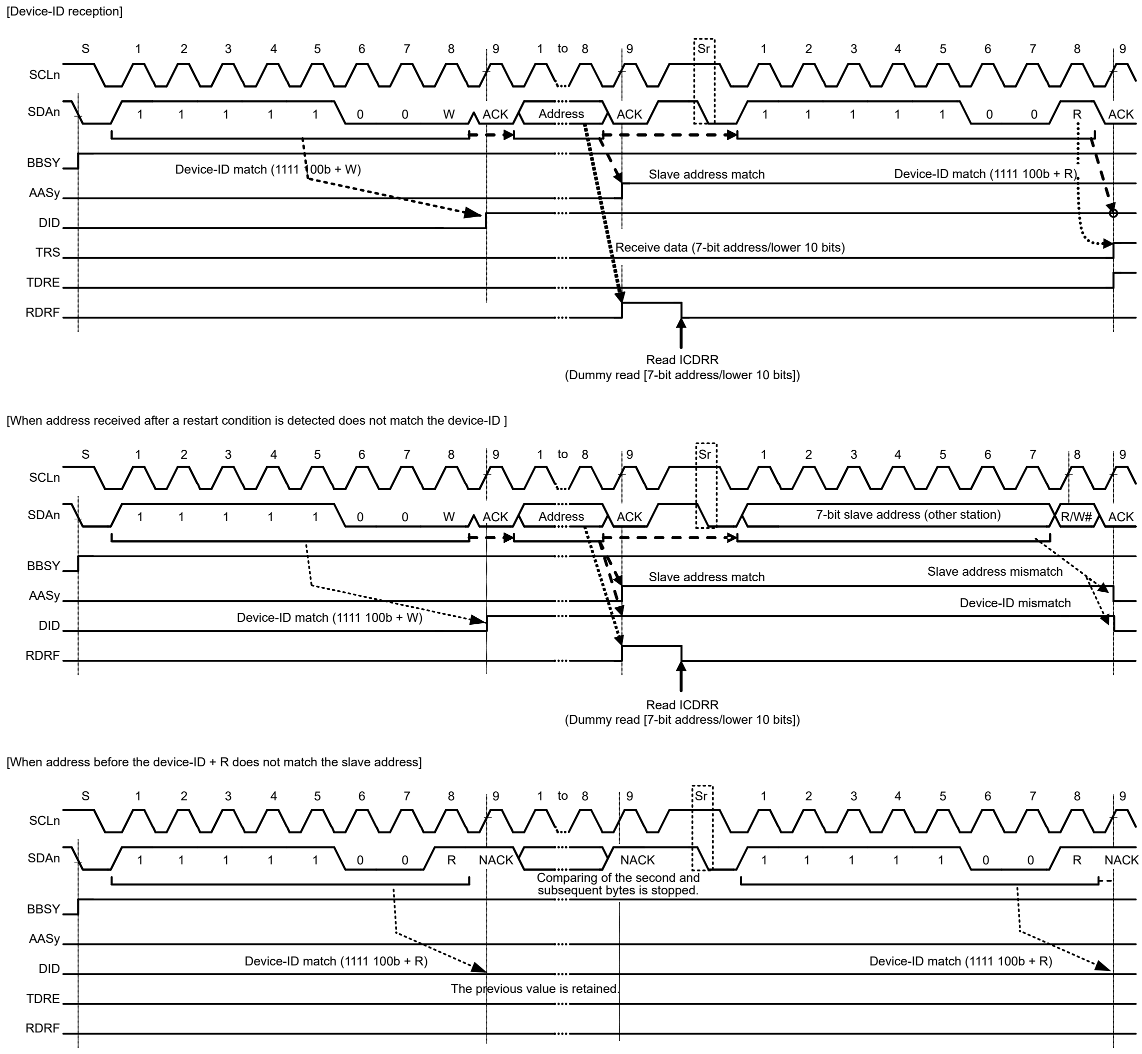
39.7.3 Device-ID Address Detection

The IIC module provides detection of device-ID address compliant with the I^{2}C bus specification (revision 03). When the IIC receives 1111 100b as the first byte after a start or restart condition is issued with the DIDE bit in ICSER set to 1, it recognizes the address as a device ID, sets the DID flag in ICSR1 to 1 on the rising edge of the ninth SCL clock cycle when the subsequent R/W# bit is 0, then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, the IIC sets the associated AASy flag (y = 0 to 2) in ICSR1 to 1.

When the first byte received after the issue of a start or restart condition matches the device ID address (1111 100b) again and the subsequent R/W# bit is 1, the IIC does not compare the second and subsequent bytes and sets the ICSR2.TDRE flag to 1.

In the device ID address detection function, the IIC sets the DID flag to 0 if a match with the IIC slave address is not obtained or a match with the device ID address is not obtained after a match with the IIC slave address and a restart condition is not detected. If the first byte after detection of a start or restart condition matches the device ID address (1111 100b), and the R/W# bit is 0, the IIC sets the DID flag to 1 and compares the second and subsequent bytes with the slave address of the IIC. If the R/W# bit is 1, the DID flag holds the previous value and the IIC does not compare the second and subsequent bytes. Therefore, the reception of a device ID address can be checked by reading the DID flag after confirming that TDRE = 1.

Additionally, prepare the device ID fields (3 bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal transmit data. For details on the information that must be included in device ID fields, contact NXP Semiconductors.



Note: n = 0, 1, y = 0 to 2

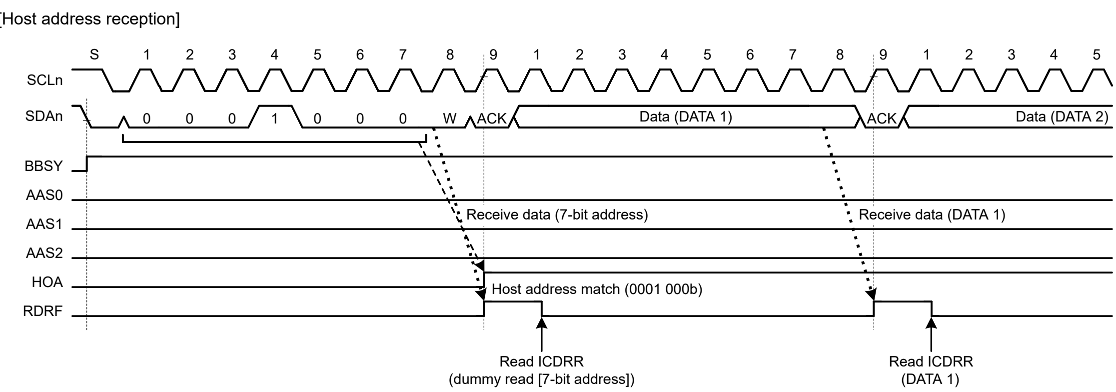
Figure 39.28 AASy and DID flag set and clear timing during reception of device ID

39.7.4 Host Address Detection

The IIC provides host address detection when operating in SMBus. When the HOAE bit in ICSER is set to 1 while the SMBS bit in ICMR3 is 1, the IIC can detect the host address (0001 000b) in slave receive mode (MST and TRS bits = 00b in ICCR2).

When the IIC detects the host address, the HOA flag in ICSR1 is set to 1 on the rising edge of the 9th SCL clock cycle, and at the same time, the RDRF flag in ICSR2 is set to 1 when the R/W# bit is 0 (Wr bit). This causes a receive data full interrupt (IICn_RXI) to be generated. The HOA flag indicates that the host address was sent from another device.

If the bit following the host address (0001 000b) is an Rd bit (R/W# bit = 1), the IIC can also detect the host address. After the host address is detected, the IIC operates in the same manner as in normal slave operation.



Note: n = 0, 1

Figure 39.29 HOA flag set timing during reception of host address

39.8 Wakeup Function

The IIC provides a wakeup function that causes the MCU to transition from Software Standby mode to normal operation. The wakeup function enables the reception of data when the peripheral module clock (PCLKB) is stopped, and generates a wakeup interrupt signal on a match of the slave address of the received data. This wakeup interrupt signal triggers the return to normal operation. After the wakeup interrupt occurs, switch the IIC to PCLKB synchronous operation so that communication can continue.

The wakeup function has four operation modes:

- Normal wakeup mode 1
- Normal wakeup mode 2
- Command recovery mode
- EEP response mode

Table 39.9 describes the behavior in these modes.

Table 39.9 Wakeup operation modes

Operation mode	ACK response timing	ACK response before wakeup to PCLKB synchronous operation	SCL state during wakeup to PCLKB synchronous operation
Normal wakeup mode 1	Before wakeup to PCLKB synchronous operation*1	ACK	Fixed low
Normal wakeup mode 2	After wakeup to PCLKB synchronous operation*2	Before wakeup: no response (NACK level retained) After wakeup: ACK response	Fixed low
Command recovery mode	Before wakeup to PCLKB synchronous operation*1	ACK	Open
EEP response mode	Before recovery to PCLKB synchronous operation*1	NACK	Open

Note 1. Switching timing from PCLKB asynchronous operation to PCLKB synchronous operation is the falling edge of the 9th clock of the SCL.

Note 2. Switching timing from PCLKB asynchronous operation to PCLKB synchronous operation is the falling edge of the 8th clock of the SCL.

The following can be selected as wakeup interrupt sources:

- Host address detection (valid when ICSER.HOAE = 1)
- General call address detection (valid when ICSER.GCAE = 1)

- Slave address 0*1 detection (valid when ICSER.SAR0E = 1)
- Slave address 1*1 detection (valid when ICSER.SAR1E = 1)
- Slave address 2*1 detection (valid when ICSER.SAR2E = 1)

Note 1. Only 7-bit address can be set. Set the FS bit in SARUy (y = 0 to 2) to 0.

Precautions on the use of the wakeup function

- Do not change the content of the IIC registers except the WUSEN bit in ICWUR2 while the WUASYF flag in ICWUR2 is 1 (during PCLKB asynchronous operation).
- Set ICWUR.WUE and ICWUR.WUIE to 1, and ICCR2.MST and ICCR2.TRS to 0 (slave reception mode) before switching to PCLKB asynchronous mode.
- The device ID and the 10-bit slave address cannot be selected for the wakeup interrupt source. Set the DIDE bit in ICSER and FS bit in SARUy (y = 0 to 2) to 0.
- Set bits TIE, TEIE, RIE, NAKIE, SPIE, STIE, ALIE, and TMOIE in the ICIER register to 0 (interrupt disabled) before switching to the asynchronous operation.
- When the wakeup function is enabled, do not use the timeout function (ICWUR.WUE = 1)
- Even when a wakeup interrupt is generated during PCLKB asynchronous operation (when ICWUR2.WUASYF = 1), if the slave addresses match in PCLKB synchronous mode (ICWUR2.WUASYF = 0), the wakeup interrupt does not occur and the WUF flag is not set.
- If the timing of writing 0 to the ICWUR2.WUSEN bit and the timing of detecting a start condition conflict, the IIC might start the next reception in PCLKB synchronous operation mode. In this case, ICWUR2.WUASYF flag becomes 1 (switch to PCLKB asynchronous mode) when data communication is complete, a stop condition is detected, and detection of a wakeup event starts.
- If you want to switch from PCLKB asynchronous operation to PCLKB synchronous operation without address match detection, it will switch in the stop condition detection. When the ICWUR2.WUSEN bit was set to 1 in a bus free state, it is continued PCLKB asynchronous operation (Reception operation: waiting communication frame). ICWUR2.WUSYF flag becomes to 1 when IIC detect the stop condition of the next communication frame, and IIC switches to PCLKB synchronous operation.
- After writing 0 to the WUSEN bit in ICWUR2, do not change registers relate to the IIC operation mode setting (ICMR3, ICSER, and SARLy) until the mode is switched to PCLKB asynchronous operation from PCLKB synchronous operation (while the ICWUR2.WUASYF flag is 1). If the register value changes during this period by an interrupt handling or another factor, the IIC might malfunction before switching to the asynchronous operation.
- During PCLKB asynchronous operation (ICWUR2.WUSYF = 0 (or WUASYF = 1)), do not refer to each flag of ICSR1, ICSR2 register and ICCR2.BBSY flag.

39.8.1 Normal Wakeup Mode 1

This section describes the behavior, the timing, and an example operation of normal wakeup mode 1.

In normal wakeup mode 1, a wakeup interrupt triggered by the match of the slave address initiates the transition to normal operation as follows:

Before wakeup:	ACK is sent in response to the data received with its own slave address of the IIC.
During wakeup:	ACK response is made on the 9th clock cycle of SCL, after which SCL is held low*1.
After wakeup:	Normal operation continues.

Note 1. Between the 9th clock cycle and 1st clock cycle during wakeup, ICMR1.WAIT = 1 is invalid.

If the slave address does not match, the SCL line is not held low after the 9th clock cycle of SCL, and the slave operation continues. Figure 39.30 shows an operation example, and Figure 39.32 shows the detailed timing.

If the transition from Software Standby mode is triggered by an interrupt other than a wakeup interrupt, for example the IRQn, the WUF flag is not set to 1. Figure 39.31 shows an operation example.

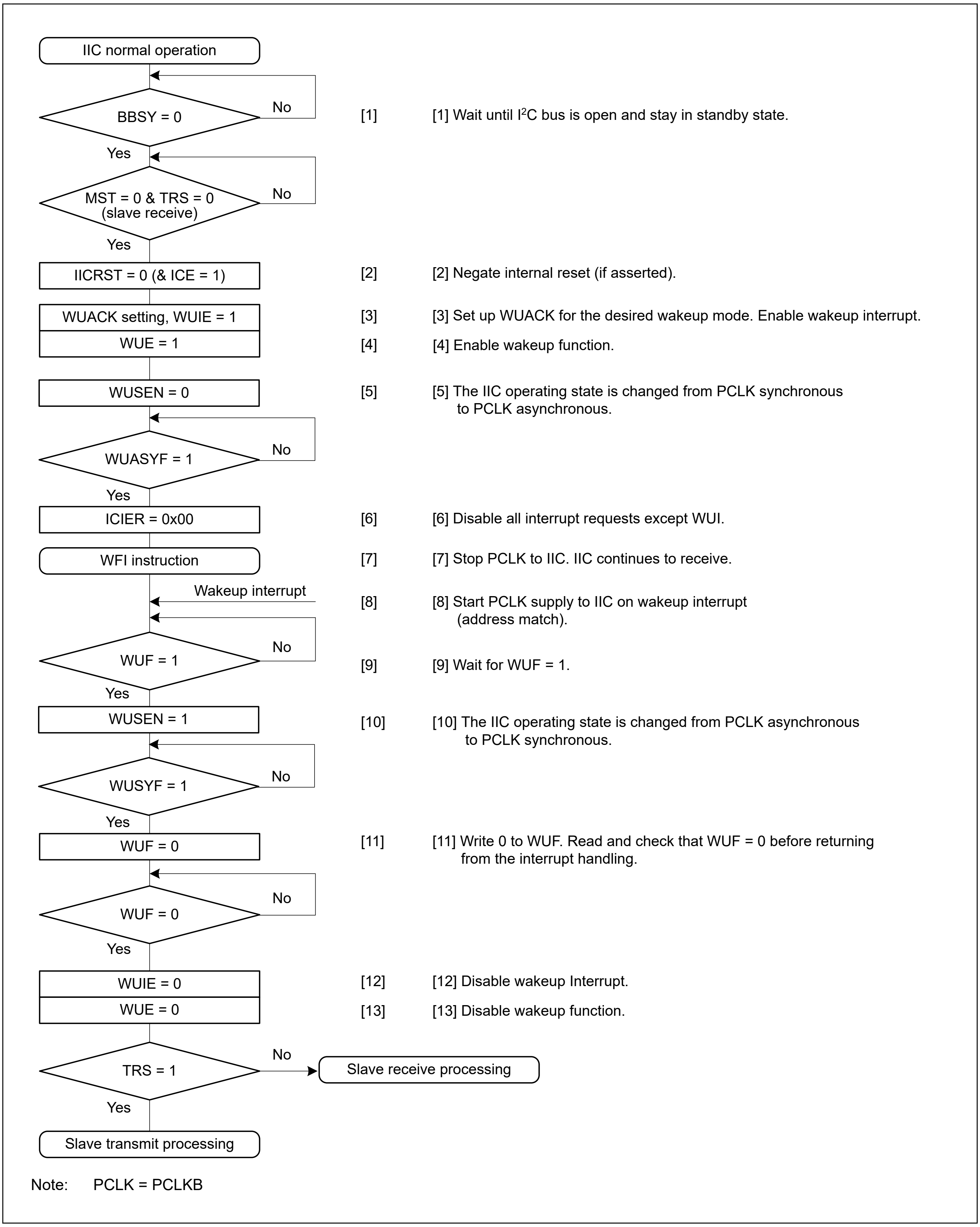
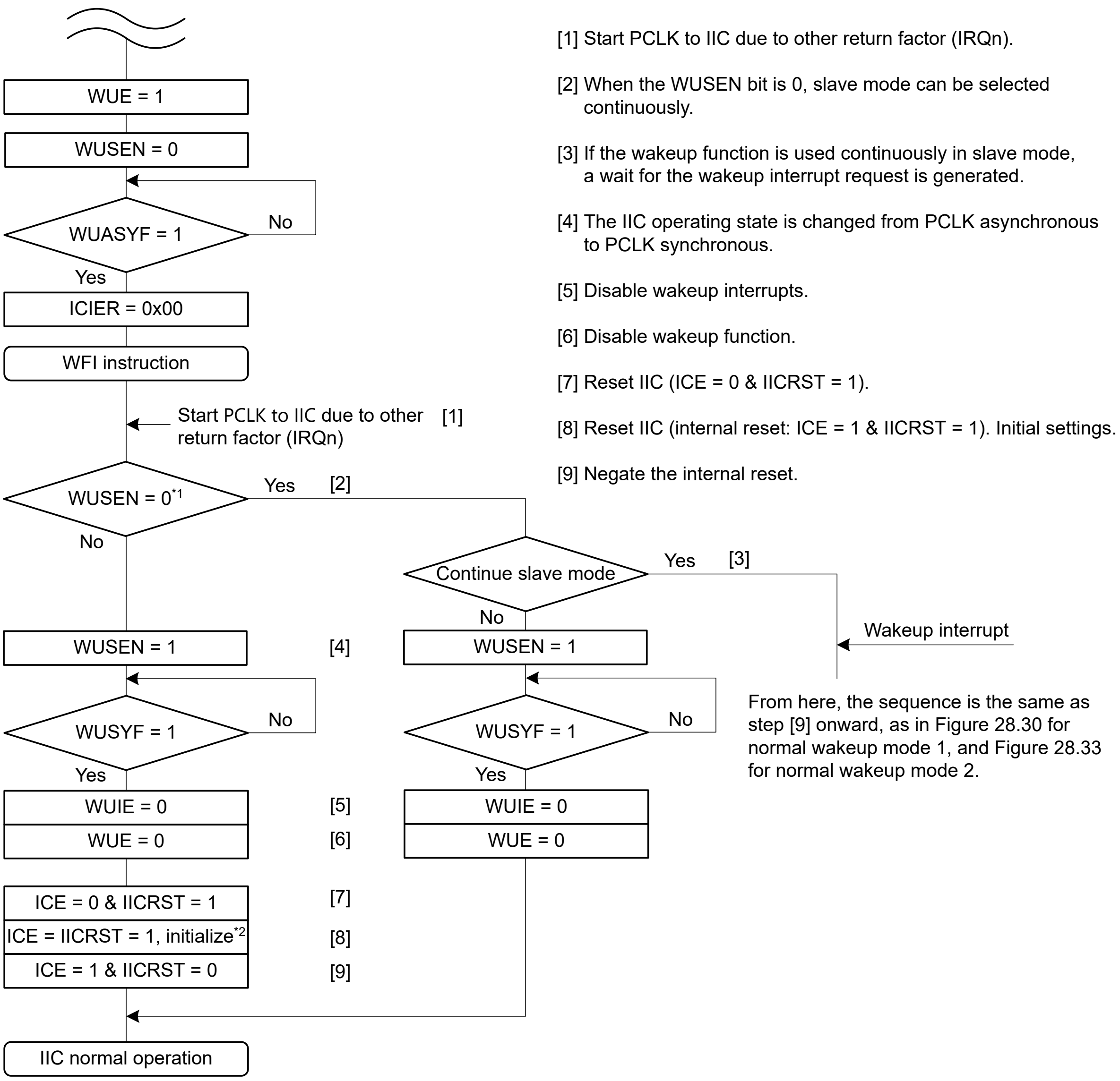


Figure 39.30 Example operation of normal wakeup mode 1 when wakeup is triggered by a wakeup interrupt on match of the slave address

Note: See Precautions on the use of the wakeup function.



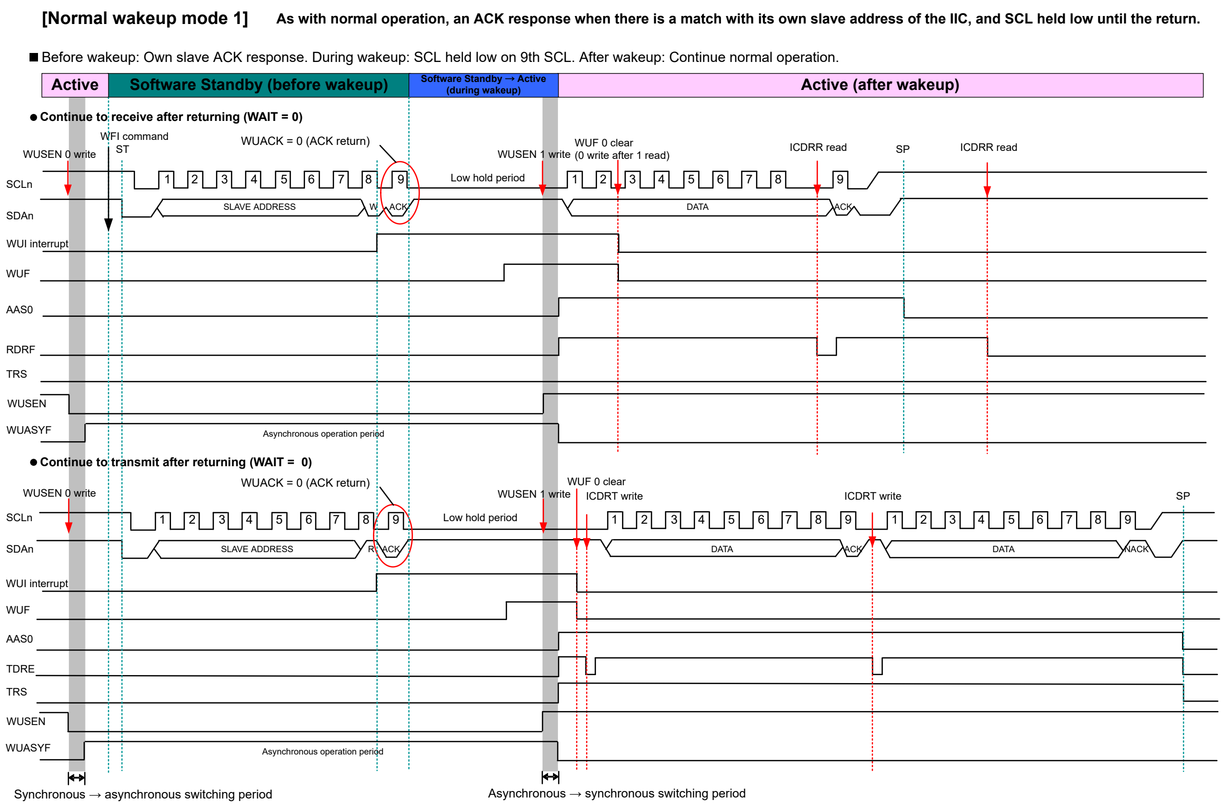
Note: PCLK = PCLKB

Note 1. In the MCU, WUSEN bit is always 0 in the IIC.

Note 2. For details on the IIC initial settings, see section 39.3.2. Initial Settings.

Figure 39.31 Example operation of normal wakeup modes 1 and 2 when wakeup is triggered by an interrupt other than IIC wakeup interrupt, for example, the IRQn

Note: For details on the IIC initial settings, see section 39.3.2. Initial Settings.



Note: n = 0

Figure 39.32 Timing of normal wakeup mode 1

39.8.2 Normal Wakeup Mode 2

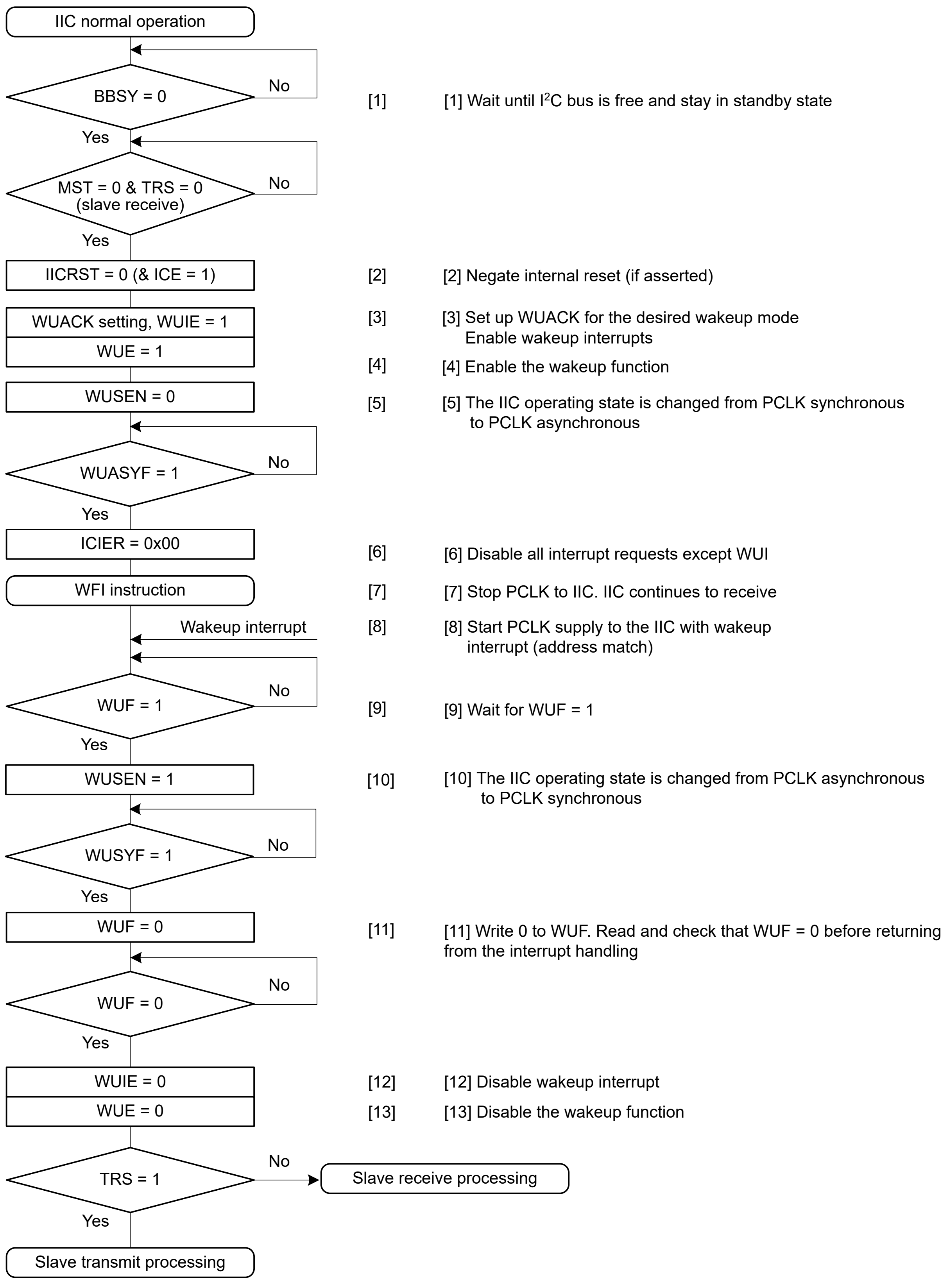
This section describes the behavior, the timing, and an example operation of normal wakeup mode 2.

In normal wakeup mode 2, a wakeup interrupt triggered by a match of the slave address initiates the transition to normal operation as follows:

Before wakeup:	No response to data received with its own slave address until the end of the 8th SCL cycle.
During wakeup:	SCL line held low during the 8th and 9th clock cycles.
After wakeup:	ACK returns on the 9th clock cycle of SCL, and normal operation continues.

If the slave address does not match, the SCL line is not held low after the 8 th SCL clock cycle, and the slave operation continues. Figure 39.33 shows an example operation, and Figure 39.34 shows the detailed timing.

If the transition from Software Standby mode is triggered by an interrupt other than a wakeup interrupt, such as the IRQn, for example, the WUF flag is not set to 1. Figure 39.31 shows an operation example.



Note: PCLK = PCLKB

Figure 39.33 Example operation of normal wakeup mode 2 when wakeup is triggered by a wakeup interrupt on match of the slave address

Note: See Precautions on the use of the wakeup function.

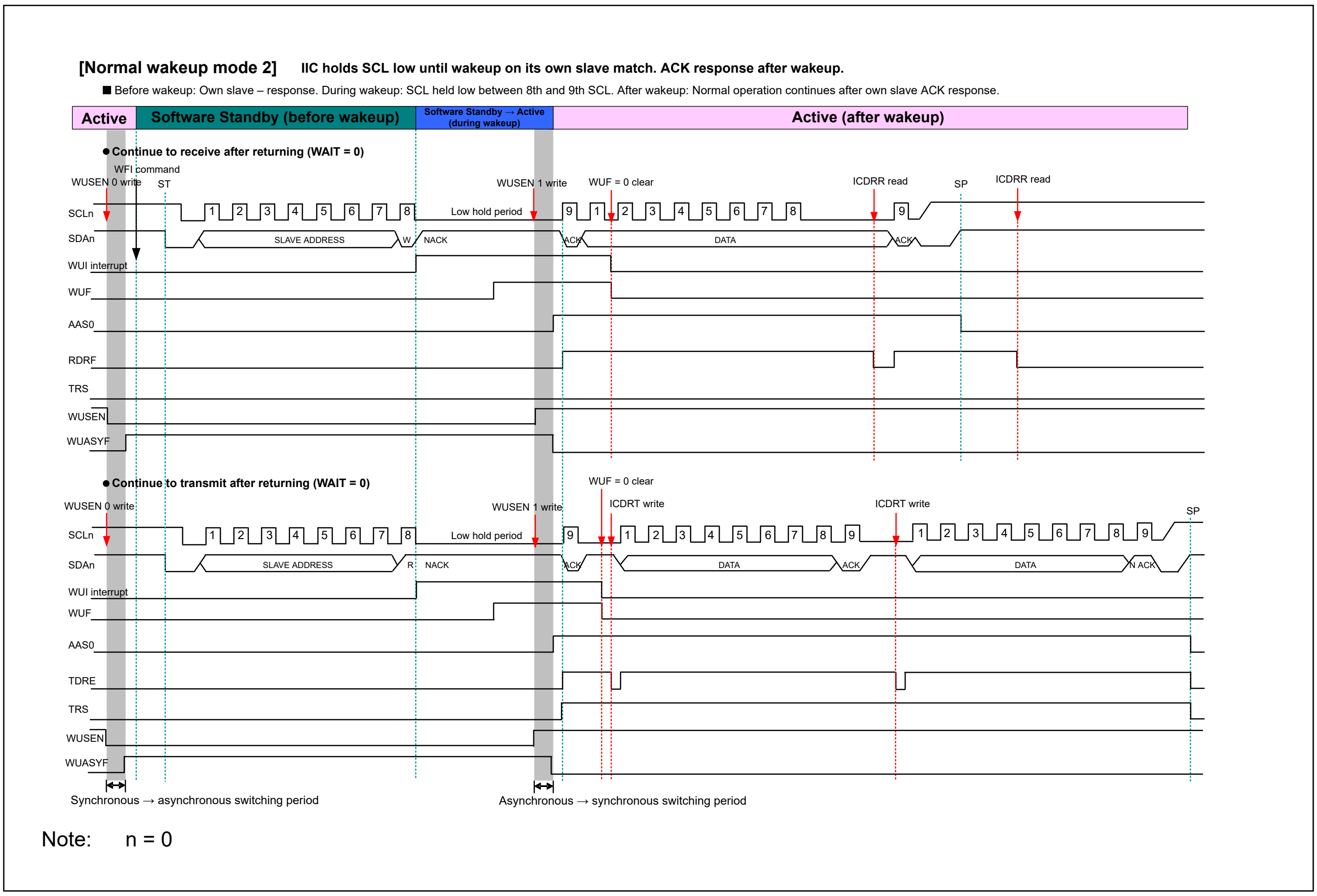


Figure 39.34 Timing of normal wakeup mode 2

39.8.3 Command Recovery Mode and EEP Response Mode (Special Wakeup Modes)

This section describes the behavior, the timing, and example operations of the command recovery and EEP response modes.

In the command recovery and EEP response modes, the SCL line is not held low during the wakeup period (after the rise of the 9th clock cycle of SCL). Therefore, other IIC devices can use the I^2C bus during this period.

A wakeup interrupt triggered by a match of the slave address initiates the transition to normal operation as follows:

Before wakeup:	In response to the data received with its own slave address, the IIC returns ACK (command recovery mode) or NACK (EEP response mode).
During wakeup:	The SCL line is not held low.
After wakeup:	Normal operation continues after IIC initialization.

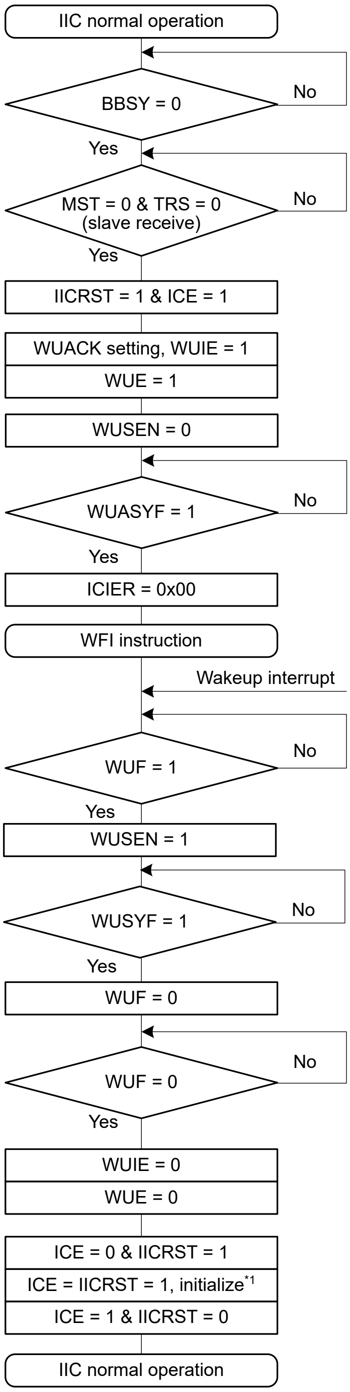
If the slave address does not match, the slave operation continues.

Note: Because the SCL line is not held low during wakeup, transmission or reception of the data that follows the slave address is not possible.

Note: The command recovery and EEP response modes are internal reset states (ICE = IICRST = 1). Therefore, the match of the slave address does not set the flags, HOA, GCA, ASS0, ASS1, and ASS2 in the ICSR1 register.

Figure 39.35 shows an example operation in recovery and EEP response modes. Figure 39.37 shows the detailed timing.

If the transition from Software Standby mode is triggered by an interrupt other than a wakeup interrupt, such as IRQn for example, the WUF flag is not set to 1. Follow the processing shown in Figure 39.36.



[1] [1] Wait until I^{2}C bus is open and stay in standby state.

[2] [2] Internal reset is asserted.

[3] [3] Set up WUACK for the desired wakeup mode. Enable wakeup interrupt.

[4] [4] Enable wakeup function.

[5] [5] The IIC operating state is changed from PCLK synchronous to PCLK asynchronous.

[6] [6] Disable all interrupt requests except WUI.

[7] [7] Stop PCLK to IIC. IIC continues to receive.

[8] [8] Start PCLK supply to the IIC with wakeup interrupt (address match).

[9] [9] Wait for WUF = 1.

[10] [10] The IIC operating state is changed from PCLK asynchronous to PCLK synchronous.

[11] [11] Write 0 to WUF. Read and check that WUF = 0 before returning from the interrupt handling.

[12] [12] Disable wakeup interrupt.

[13] [13] Disable wakeup function.

[14] [14] Reset IIC (ICE = 0 & IICRST = 1).

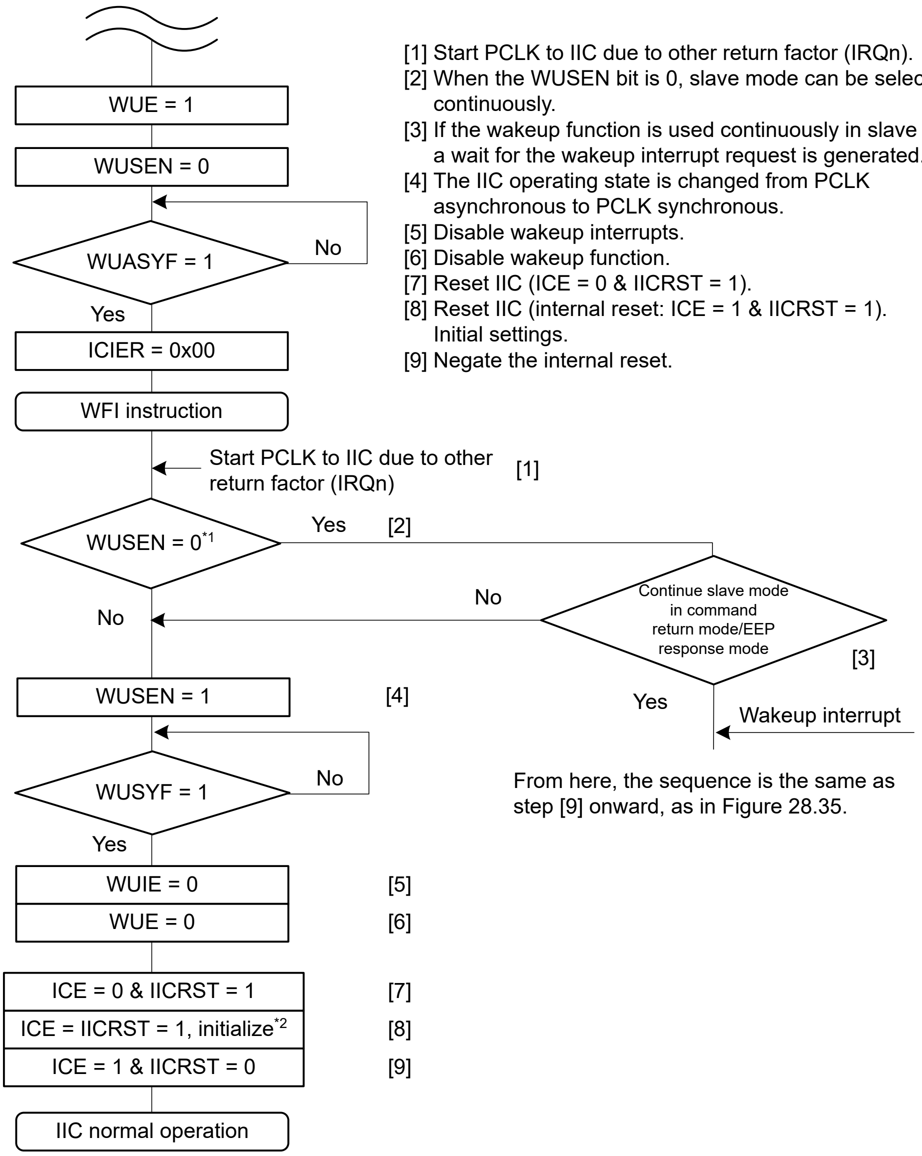
[15] [15] Reset IIC (internal reset: ICE = 1 & IICRST = 1). Initial settings.

[16] [16] Negate the internal reset.

Note: PCLK = PCLKB
Note 1. For details on IIC initialization, see section 39.3.2. Initial Settings.

Figure 39.35 Example operation of command recovery mode and EEP response mode when wakeup is triggered by a wakeup interrupt on match of the slave address

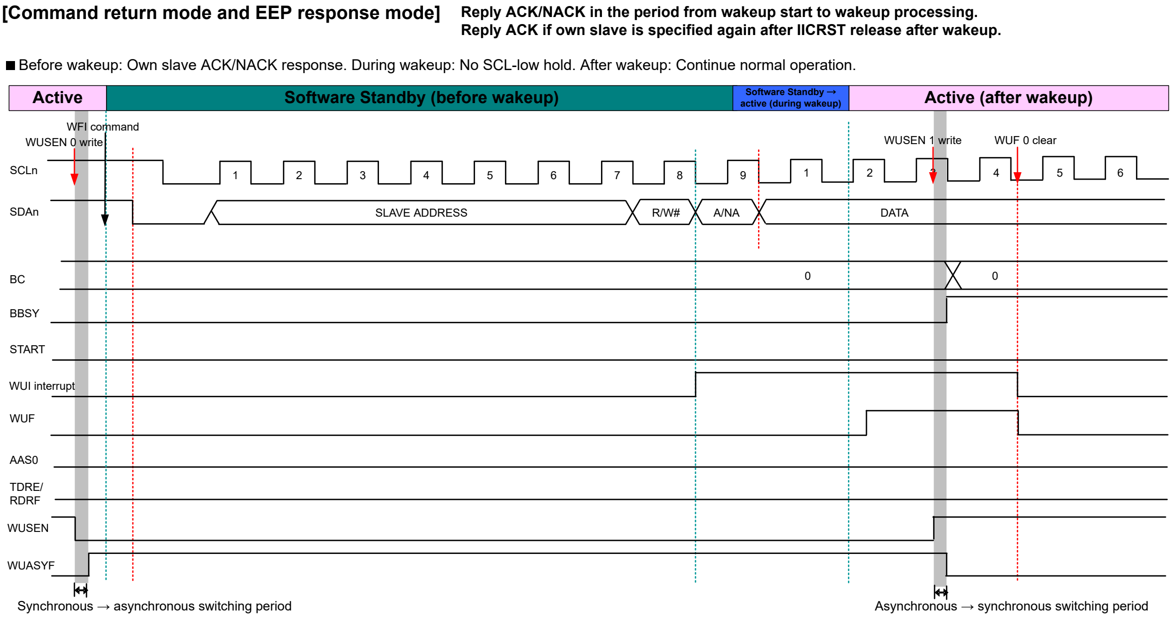
Note: See Precautions on the use of the wakeup function.



[1] Start PCLK to IIC due to other return factor (IRQn).
[2] When the WUSEN bit is 0, slave mode can be selected continuously.
[3] If the wakeup function is used continuously in slave mode, a wait for the wakeup interrupt request is generated.
[4] The IIC operating state is changed from PCLK asynchronous to PCLK synchronous.
[5] Disable wakeup interrupts.
[6] Disable wakeup function.
[7] Reset IIC (ICE = 0 & IICRST = 1).
[8] Reset IIC (internal reset: ICE = 1 & IICRST = 1). Initial settings.
[9] Negate the internal reset.

Note: PCLK = PCLKB
Note 1. In the MCU, WUSEN bit is always 0 in the IIC.
Note 2. For details on IIC initialization, see section 39.3.2. Initial Settings.

Figure 39.36 Example operation of command recovery and EEP response modes when wakeup is triggered by an interrupt other than IIC wakeup interrupt, for example, the IRQn



Note: n = 0

Figure 39.37 Timing of command recovery and EEP response modes

39.9 Automatic Low-Hold Function for SCL

39.9.1 Function to Prevent Wrong Transmission of Transmit Data

If the I²C Bus Shift Register (ICDRS) is empty and data has not been written to the I²C Bus Transmit Data Register (ICDRT) with the IIC in transmission mode (TRS bit = 1 in ICCR2), the SCLn line is automatically held low over the subsequent intervals. This low-hold period is extended until the transmit data is written, which prevents the unintended transmission of erroneous data.

Master transmit mode:

- Low-level interval after a start or restart condition is issued
- Low-level interval between the 9th clock cycle of one transfer and the 1st clock cycle of the next.

Slave transmit mode:

- Low-level interval between the 9th clock cycle of one transfer and the 1st clock cycle of the next.

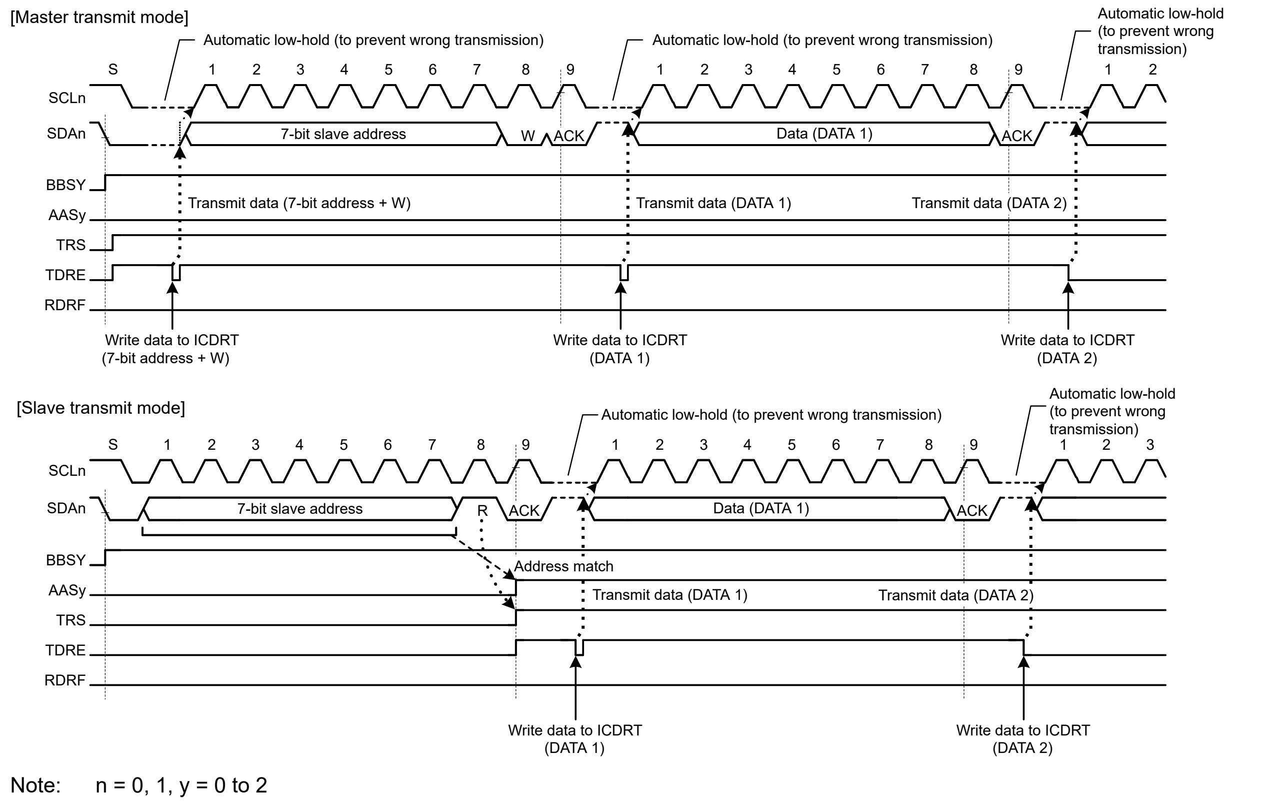


Figure 39.38 Automatic low-hold operation in transmit mode

39.9.2 NACK Reception Transfer Suspension Function

This function suspends transfer operation when NACK is received in transmit mode (TRS bit = 1 in ICCR2). This function is enabled when the NACKE bit in ICFER is set to 1. If the next transmit data is already written (TDRE flag = 0 in ICSR2) when NACK is received, the next data transmission on the falling edge of the 9th SCL clock cycle is automatically suspended. This prevents the SDAn line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is suspended by this function (NACKF flag = 1 in ICSR2), transmit and receive operations are discontinued. To restore transmit or receive operation, after issuing the restart condition, you need to set the NACKF flag to 0 and try again, or set the NACKF flag to 0 after issuing the stop condition and then start again from issuing the start condition.

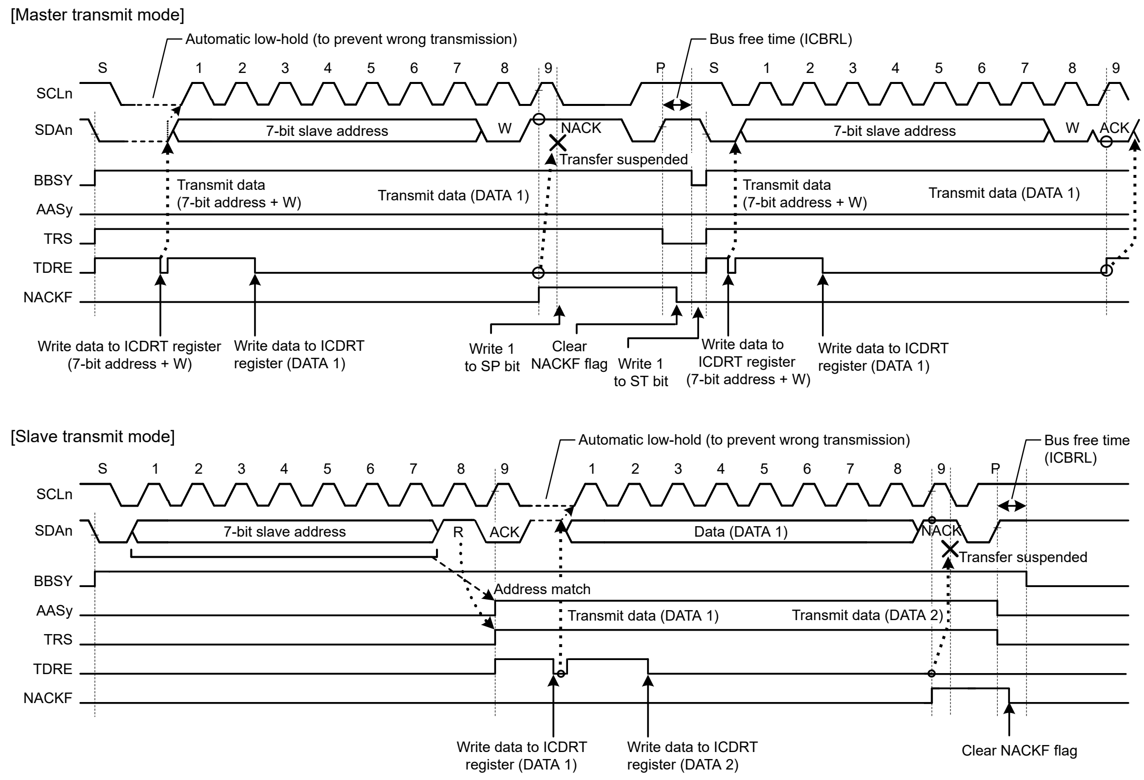


Figure 39.39 Suspension of data transfer when NACK is received, when NACKE = 1

39.9.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (ICDRR) read is delayed for a period of one transfer frame or more with receive data full (RDRF flag = 1 in ICSR2) in receive mode (TRS = 0 in ICCR2), the IIC holds the SCLn line low automatically immediately before the next data is received to prevent a failure to receive data.

This function is enabled even if the read processing of the final receive data is delayed and, in the meantime, the IIC slave address is designated after a stop condition is issued. This function does not interfere with other communication because the IIC does not hold the SCLn line low when a mismatch with its own slave address occurs after a stop condition is issued.

Periods in which the SCLn line is held low can be selected with a combination of the WAIT and RDRFS bits in ICMR3.

(1) 1-byte receive operation and automatic low-hold function using the WAIT bit

When the WAIT bit in ICMR3 is set to 1, the IIC performs a 1-byte receive operation using the WAIT bit function. Additionally, when the ICMR3.RDRFS bit is 0, the IIC automatically sends the ACKBT bit value in ICMR3 for the acknowledge bit in the period from the falling edge of the 8th SCL clock cycle to the falling edge of the 9th SCL clock cycle, and automatically holds the SCLn line low on the falling edge of the 9th SCL clock cycle using the WAIT bit function. This low-hold is released by reading data from ICDRR, which enables byte-wise receive operation.

The WAIT bit function is enabled for receive frames after a match with the IIC slave address, including the general call address and host address, is obtained in master or slave receive mode.

(2) 1-byte receive operation (ACK/NACK transmission control) and automatic low-hold function using the RDRFS bit

When the RDRFS bit in ICMR3 is set to 1, the IIC performs a 1-byte receive operation using the RDRFS bit function. When the RDRFS bit is set to 1, the RDRF flag in ICSR2 is set to 1 (receive data full) on the rising edge of the eighth SCL clock cycle, and the SCLn line is automatically held low on the falling edge of the eighth SCL clock cycle. This low-hold is released by writing a value to the ACKBT bit in ICMR3, but cannot be released by reading data from ICDRR, which enables receive operation through the ACK or NACK transmission control based on the data received in byte units.

The RDRFS bit function is enabled for receive frames after a match with the IIC slave address, including the general call address and host address, is obtained in master or slave receive mode.

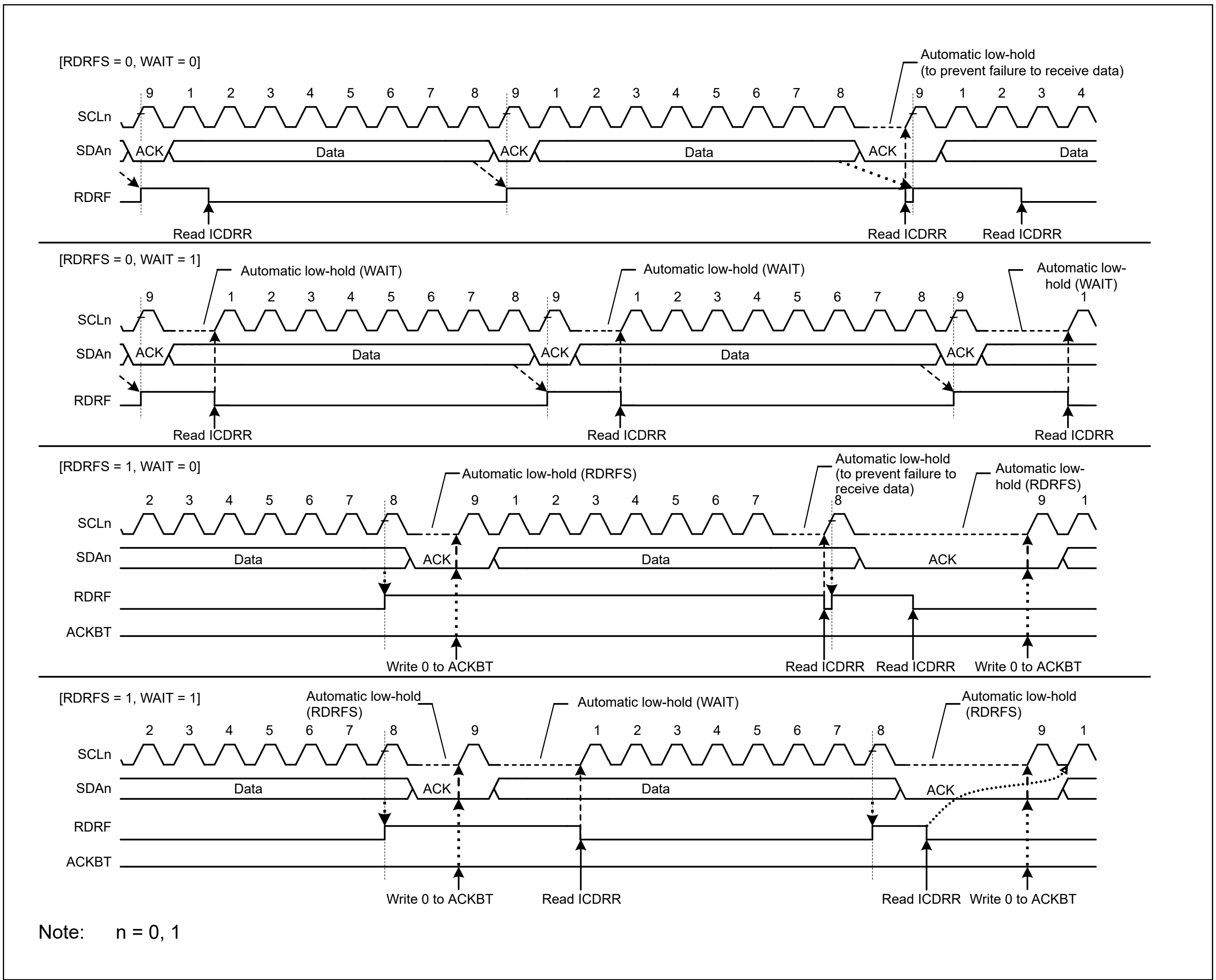


Figure 39.40 Automatic low-hold operation in receive mode using the RDRFS and WAIT bits

39.10 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I^2C bus standard, the IIC provides functions to prevent double-issue of a start condition, detect arbitration-lost during transmission of NACK, and detect arbitration-lost in slave transmit mode.

39.10.1 Master Arbitration-Lost Detection (MALE Bit)

The IIC drives the SDAn line low to issue a start condition. However, if the SDAn line was already driven low by another master device issuing a start condition, the IIC regards its own start condition as an error and considers this a loss in arbitration. Priority is given to transfer by the other master device. Similarly, if a request to issue a start condition is made by setting the ST bit in ICCR2 to 1 while the bus is busy (BBSY flag = 1 in ICCR2), the IIC regards this as a double-issuing-of-start-condition error and considers itself to have lost in arbitration. This prevents a failure of transfer resulting from a start condition being issued while transfer is in progress.

When a start condition is issued successfully, if the transmit data including the address bits (internal SDA output level) and the level on the SDAn line do not match (high output as the internal SDA output, meaning the SDAn pin is in the high-impedance state and a low level is detected on the SDAn line), the IIC loses in arbitration.

After a loss in arbitration of mastership, the IIC immediately enters slave receive mode. If a slave address, including the general call address, matches its own address at this time, the IIC continues in slave operation.

A loss in arbitration of mastership is detected when the following conditions are met while the MALE bit in ICFER is 1 (master arbitration-lost detection enabled).

[Master arbitration-lost conditions]

- Mismatching of the internal level for output on SDA and the level on the SDAn line after a start condition was issued by setting the ST bit in ICCR2 to 1 while the BBSY flag in ICCR2 is set to 0 (erroneous issuing of a start condition)
- Setting of the ST bit in ICCR2 to 1 (start condition double-issue error) while the BBSY flag is 1
- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the SDAn line in master transmit mode (MST and TRS bits = 11b in ICCR2).

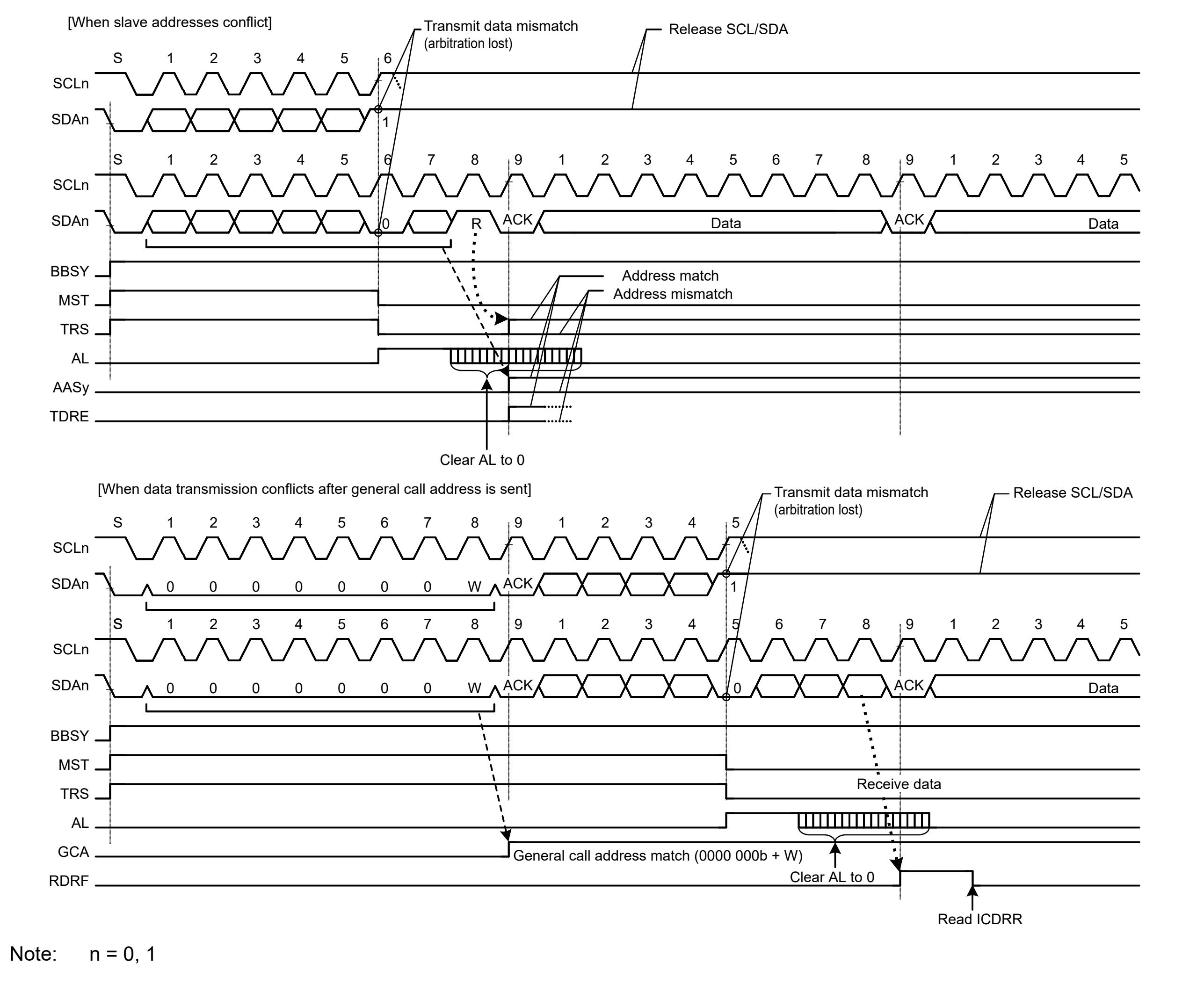
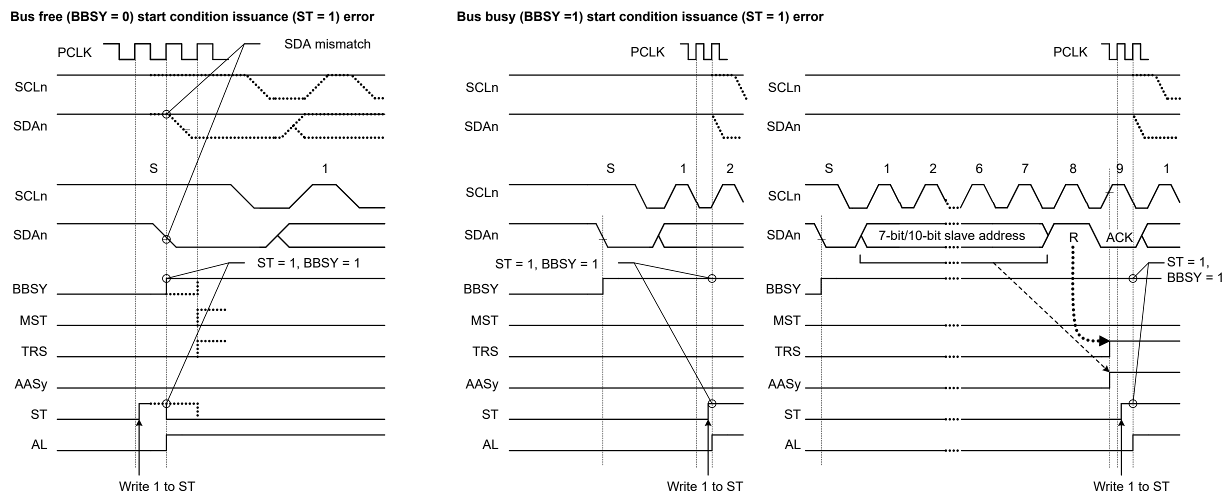


Figure 39.41 Examples of master arbitration-lost detection when MALE = 1

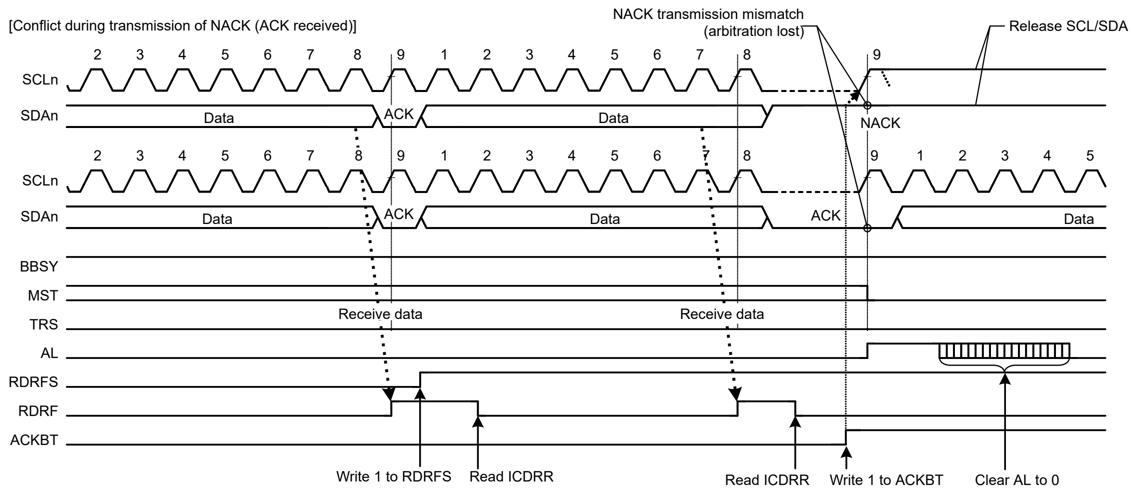


Note: PCLK = PCLKB
Note: n = 0, 1, y = 0 to 2

Figure 39.42 Arbitration-lost when start condition is issued when MALE = 1

39.10.2 Function to Detect Loss of Arbitration During NACK Transmission (NALE Bit)

This function causes arbitration to be lost if the internal SDA output level does not match the level on the SDAn line (high output as the internal SDA output, meaning the SDAn pin is in the high-impedance state) and the low level is detected on the SDAn line during transmission of NACK in receive mode. Arbitration is lost because of a conflict between NACK and ACK transmissions when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such a conflict occurs when multiple master devices send or receive the same information through a single slave device. Figure 39.43 shows an example of arbitration-lost detection during transmission of NACK.



Note: n = 0, 1

Figure 39.43 Example of arbitration-lost detection during transmission of NACK when NALE = 1

The following explains arbitration-lost detection using an example in which two master devices (master A and master B) and a single slave device are connected through the bus. In this example, master A receives 2 bytes of data from the slave device, and master B receives 4 bytes of data from the slave device.

If master A and master B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in either master A or master B during access to the slave device. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. Master A sends NACK when it has received 2 final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received the required 4 bytes of data. At this time, the NACK transmission from master A and the ACK transmission from master B conflict. In general, if a conflict like this occurs, master A cannot detect the ACK transmitted by master B and issues a stop condition. The stop condition issue conflicts with the SCL clock output of master B, which disrupts communication.

When the IIC receives ACK during transmission of NACK, it detects a defeat in conflict with other master devices and causes arbitration to be lost. If arbitration is lost during transmission of NACK, the IIC immediately cancels the slave match condition and enters slave receive mode. This prevents a stop condition from being issued, preventing a communication failure on the bus.

Similarly, in the ARP command processing of SMBus, the function to detect loss of arbitration during transmission of NACK is also available for eliminating the extra clock cycle processing, such as 0xFF transmission processing, which is required if the UDID (Unique Device Identifier) of the assigned address does not match in the Get UDID general processing after the Assign Address command.

The IIC detects arbitration-lost during transmission of NACK when the following condition is met with the NALE bit in ICFER set to 1 (arbitration-lost detection during NACK transmission enabled).

[Condition for arbitration-lost during NACK transmission]

- When the internal SDA output level does not match the SDAn line (ACK is received) during transmission of NACK (ACKBT bit = 1 in ICMR3).

39.10.3 Slave Arbitration-Lost Detection (SALE Bit)

This function causes arbitration to be lost if the transmit data (internal SDA output level) and the level on the SDAn line do not match (high output as the internal SDA output, meaning the SDAn pin is in the high-impedance state), and the low level is detected on the SDAn line in slave transmit mode. This arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identifier) over an SMBus.

When the IIC loses slave arbitration, the IIC is immediately released from the slave-matched state and enters slave receive mode. This function can detect conflicts of data during transmission of UDIDs over an SMBus and eliminates subsequent redundant processing for the transmission of 0xFF.

The IIC detects slave arbitration-lost when the following condition is met with the SALE bit in ICFER set to 1 (slave arbitration-lost detection enabled).

[Condition for slave arbitration-lost]

- When transmit data excluding acknowledge (internal SDA output level) does not match the SDAn line in slave transmit mode (MST and TRS bits = 01b in ICCR2).

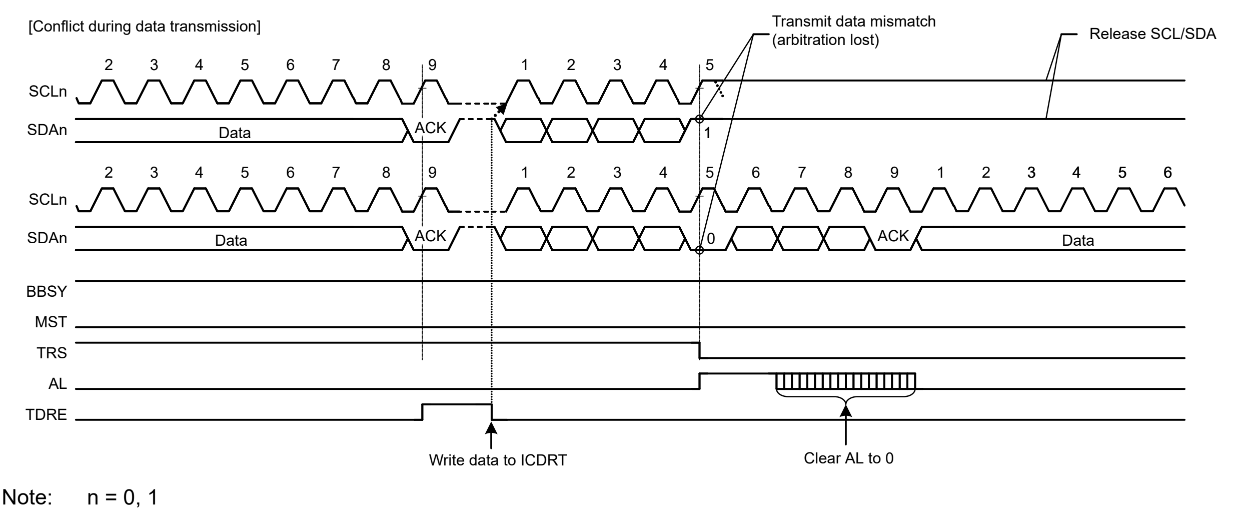


Figure 39.44 Example of slave arbitration-lost detection when SALE = 1

39.11 Start, Restart, and Stop Condition Issuing Function

39.11.1 Issuing a Start Condition

The IIC issues a start condition when the ST bit in ICCR2 is set to 1. When the ST bit is set to 1, a start condition request is made, and the IIC issues a start condition when the BBSY flag in ICCR2 is 0 (bus free state). When a start condition is issued normally, the IIC automatically shifts to the master transmit mode.

To issue a start condition:

1. Drive the SDAn line low (high level to low level).
2. Ensure that the time set in ICBRH and the start condition hold time elapse.
3. Drive the SCLn line low (high level to low level).
4. Detect low level on the SCLn line and ensure the low-level period of the SCLn line set in ICBRL elapses.

39.11.2 Issuing a Restart Condition

The IIC issues a restart condition when the RS bit in ICCR2 is set to 1. When the RS bit is set to 1, a restart condition request is made, and the IIC issues a restart condition when the BBSY flag in ICCR2 is 1 (bus busy state) and the MST bit in ICCR2 is 1 (master mode).

To issue a restart condition:

1. Release the SDAn line.
2. Ensure the low-level period of the SCLn line set in ICBRL elapses.
3. Release the SCLn line (low level to high level).
4. Detect a high level on the SCLn line and ensure the time set in ICBRL and the restart condition setup time elapse.
5. Drive the SDAn line low (high level to low level).
6. Ensure the time set in ICBRH and the restart condition hold time elapse.
7. Drive the SCLn line low (high level to low level).
8. Detect a low level on the SCLn line and ensure the low-level period of the SCLn line set in ICBRL elapses.

Note: When issuing restart condition requests, write the slave address to ICDRT after confirming that ICCR2.RS = 0. Data written while ICCR2.RS = 1 is not forwarded because of the retransmission condition before the occurrence.

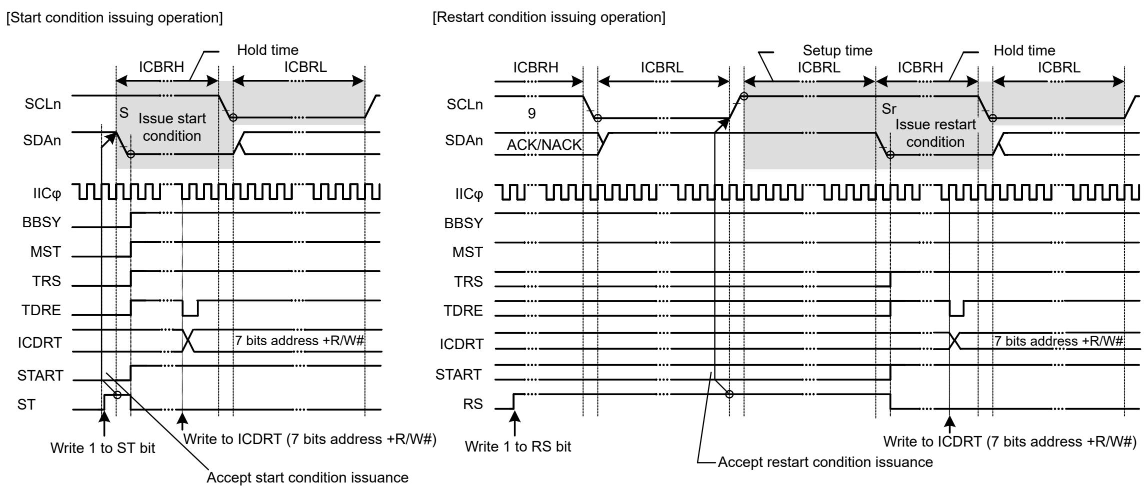
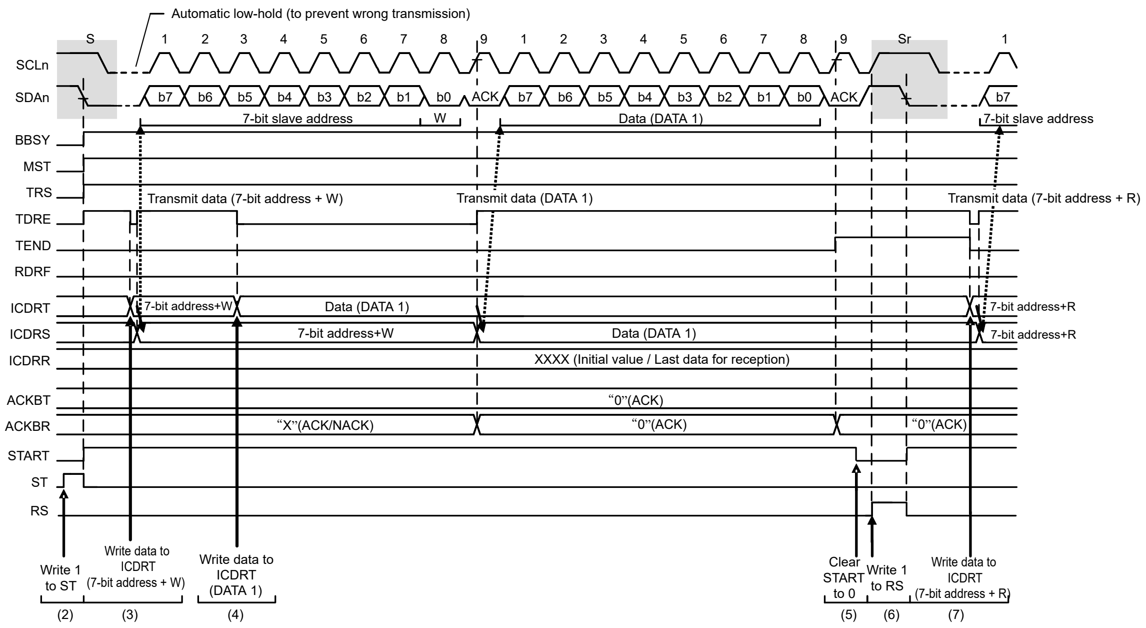


Figure 39.45 Start and restart condition issue timing using the ST and RS bits

Figure 39.46 shows the operation timing when a restart condition is issued after the master transmission.

[To issue a restart condition after the master transmission:]

1. Initialize the IIC using the procedure in section 39.3.2. Initial Settings.
2. Read the BBSY flag in ICCR2 to check that the bus is open, then set the ST bit in ICCR2 to 1 (start condition issuance request). On receiving the request, the IIC issues a start condition. At the same time, the BBSY and the START flags in ICSR2 are automatically set to 1 and the ST bit is automatically set to 0. If the start condition is detected and the internal levels for the SDA output state and the levels on the SDAn line match while the ST bit is 1, the IIC recognizes that a start condition is successfully issued as requested by the ST bit has been successfully completed. The MST and TRS bits in ICCR2 are automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 is also automatically set to 1 when the TRS bit is set to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDRT. After the data for transmission is written to ICDRT, the TDRE flag is automatically set to 0, data is transferred from ICDRT to ICDRS, and the TDRE flag is again set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRS bit is automatically updated to select master transmit or master receive mode according to the value of the transmitted R/W# bit. If the value of the R/W# bit is 0, the IIC continues in master transmit mode. If the ICSR2.NACKF flag is 1 at this time, indicating that no slave device recognized the address or that there was an error in communications, write 1 to ICCR2.SP bit to issue a stop condition. To transmit data with an address in the 10-bit format, start by writing 1111 0b, the 2 upper bits of the slave address, and W to ICDRT as the first address transmission. Then, as the second address transmission, write the 8 lower bits of the slave address to ICDRT.
4. After confirming that the TDRE flag in ICSR2 is 1, write data for transmission to the ICDRT register. The IIC automatically holds the SCLn line low until data for transmission is ready, a restart condition is issued or a stop condition is issued.
5. After all bytes of data for transmission are written to the ICDRT register, wait until the value of the TEND flag in ICSR2 returns to 1. Then after checking that the START flag in ICSR2 is 1, set the START flag in ICSR2 to 0.
6. Set the RS bit in ICCR2 to 1 (restart condition issue request). On receiving the request, the IIC issues a restart condition.
7. After checking that the START flag in ICSR2 is 1, write the value for transmission (the slave address and the R/W# bit) to ICDRT.



Note: n = 0, 1

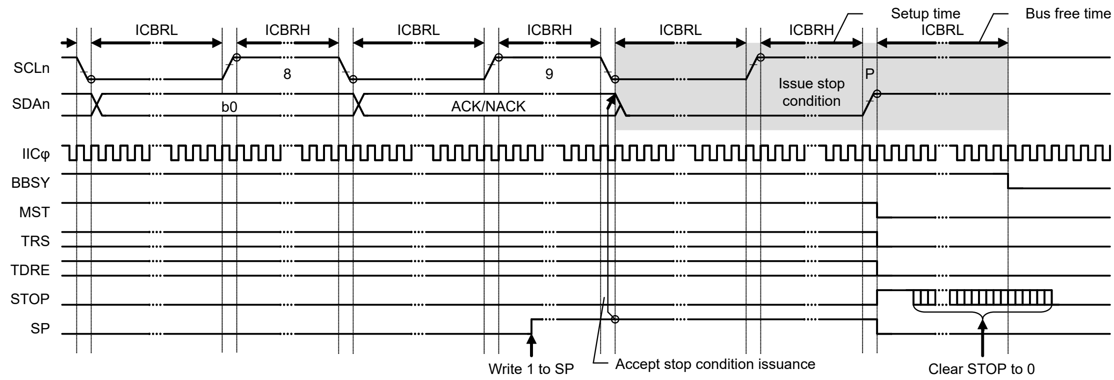
Figure 39.46 Restart condition issue timing after master transmission.

39.11.3 Issuing a Stop Condition

The IIC issues a stop condition when the SP bit in ICCR2 is set to 1. When the SP bit is set to 1, a stop condition request is made, and the IIC issues a stop condition when the BBSY flag in ICCR2 is 1 (bus busy state) and the MST bit in ICCR2 is 1 (master mode).

To issue a stop condition:

1. Drive the SDAn line low (high level to low level).
2. Ensure the low-level period of the SCLn line set in ICBRL elapses.
3. Release the SCLn line (low level to high level).
4. Detect a high level on the SCLn line and ensure the time set in ICBRH and the stop condition setup time elapse.
5. Release the SDAn line (low level to high level).
6. Ensure the time set in ICBRL and the bus free time elapse.
7. Clear the BBSY flag to 0 to release the bus mastership.



Note: n = 0, 1

Figure 39.47 Stop condition issue timing using the SP bit

39.12 Bus Hanging

If the clock signals from the master and slave devices are out of synchronization because of noise or other factors, the I^2C bus might hang with a fixed level on the SCLn line or SDAn line.

To manage bus hanging, the IIC has a timeout function to detect hanging by monitoring the SCLn line, and a function for outputting an extra SCL clock cycle to release the bus from:

- A timeout function to detect hanging by monitoring the SCLn line
- The IIC reset function
- An internal reset function.

By checking the SCLO, SDAO, SCLI, and SDAI bits in ICCR1, it is possible to see whether the IIC or its communicating partner is placing the low level on the SCLn or SDAn line.

39.12.1 Timeout Function

The timeout function can detect when the SCLn line is stuck longer than the predetermined time. The IIC can detect an abnormal bus state by monitoring that the SCLn line is stuck low or high for a predetermined time.

The timeout function monitors the SCLn line state and counts the low- or high-level period using the internal counter. The timeout function resets the internal counter each time the SCLn line changes (rises or falls), but continues to count unless the SCLn line changes. If the internal counter overflows because no SCLn line changes, the IIC can detect the timeout and report the bus hung state.

This timeout function is enabled when the ICFER.TMOE bit is 1. It detects a hung state when the SCLn line is stuck low or high during the following conditions:

- The bus is busy (ICCR2.BBSY flag is 1) in master mode (ICCR2.MST bit is 1)
- The IIC slave address is detected (ICSR1 register is not 0x00) and the bus is busy (ICCR2.BBSY flag is 1) in slave mode (ICCR2.MST bit is 0)
- The bus is open (ICCR2.BBSY flag is 0) while a start condition is requested (ICCR2.ST bit is 1).

The internal counter of the timeout function uses the internal reference clock (IICφ) set in the CKS[2:0] bits in ICMR1 as a count source. It functions as a 16-bit counter when long mode is selected (TMOS bit = 0 in ICMR2) or a 14-bit counter when short mode is selected (TMOS bit = 1).

The SCLn line level (low, high, or both levels) during which this counter is activated can be selected in the TMOH and TMOL bits in ICMR2. If both TMOL and TMOH bits are set to 0, the internal counter is disabled.

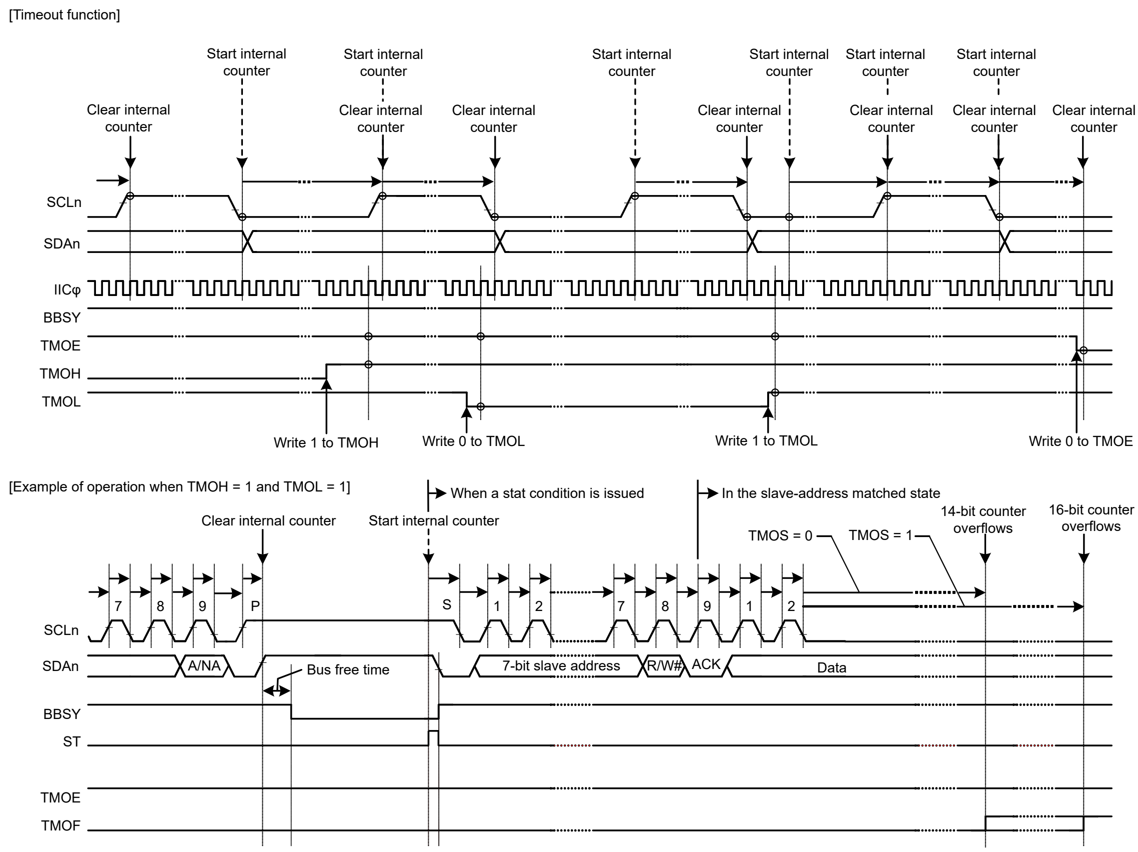


Figure 39.48 Timeout function using the TMOE, TMOS, TMOH, and TMOL bits

39.12.2 Extra SCL Clock Cycle Output Function

In master mode, this function outputs extra SCL clock cycles to release the SDAn line of the slave device from being held low because the master is out of synchronization with the slave device. This function is mainly used in master mode to release the SDAn line of the slave device from being fixed low by including extra cycles of SCL output from the IIC. It uses single cycles of the SCL clock for a bus error where the IIC cannot issue a stop condition because the slave device is holding the SDAn line at the low level. Do not use this function in normal situations. Using it when communications are proceeding correctly leads to malfunctions.

When the CLO bit in ICCR1 is set to 1 in master mode, a single cycle of the SCL clock at the frequency specified in the CKS[2:0] bits in ICMR1, and in the ICBRH and ICBRL registers, is output as an extra clock cycle. After output of this single cycle of the SCL clock, the CLO bit is automatically set to 0. At this time, if ICCR2.BBSY = 1, the SCL pin goes low, and when ICCR2.BBSY = 0, the SCL pin goes high. After confirming that the CLO bit is 0 by software, write 1 to the CLO bit to output the additional clock continuously.

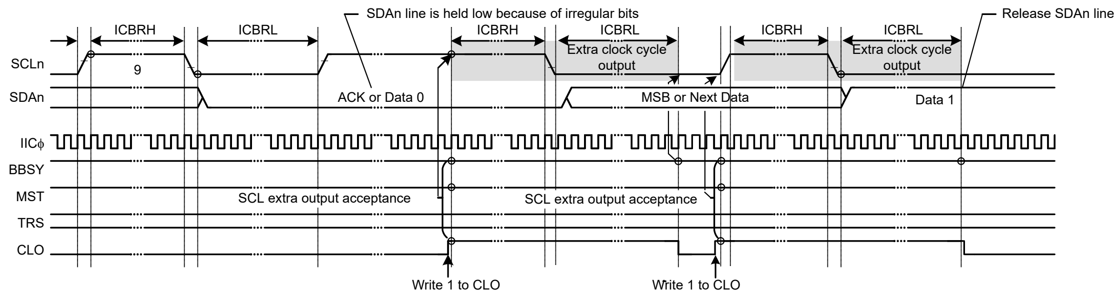
When the IIC module is in master mode and the slave device is holding the SDAn line low because synchronization with the slave device is lost because of effects like noise, the output of a stop condition is not possible. This function can be used to output extra cycles of SCL one by one to make the slave device release the SDAn line from being held low, and so recover the bus from an unusable state. Release of the SDAn line by the slave device can be monitored by reading the SDAI bit in ICCR1. After confirming the release of the SDAn line by the slave device, complete communications by reissuing the stop condition.

[Output conditions for using the CLO bit in ICCR1]

- When the bus is open (BBSY flag in ICCR2 = 0) or in master mode (MST bit = 1 and BBSY flag = 1 in ICCR2)

- When the communication device does not hold the SCLn line low.

Figure 39.49 shows the operation timing of the extra SCL clock cycle output function (CLO bit).



Note: n = 0, 1

Figure 39.49 Extra SCL clock cycle output function using the CLO bit

39.12.3 IIC Reset and Internal Reset

The IIC module incorporates a function for resetting itself. It uses two types of resets:

- An IIC reset, which initializes all registers, including the BBSY flag in ICCR2.
- An internal reset, which releases the IIC from the slave-address matched state and initializes the internal counter while saving other settings.

After issuing a reset, always set the IICRST bit in ICCR1 to 0. Both types of resets are valid for release from bus-hung states, because both restore the output state of the SCLn and SDAn pins to the high-impedance state.

Issuing a reset during slave operation might lead to a loss of synchronization between the master device clock and the slave device clock, so avoid this when possible. In addition, monitoring of the bus state, such as for the presence of a start condition, is not possible during an IIC reset (ICE and IICRST bits = 01b in ICCR1).

For a detailed description of the IIC and internal resets, see section 39.15. State of Registers When Issuing Each Condition.

39.13 SMBus Operation

The IIC supports data communication conforming to the SMBus Specification (version 2.0). To perform SMBus communication, set the SMBS bit in ICMR3 to 1. To use the transfer rate within a range of 10 to 100 kbps of the SMBus standard, set the CKS[2:0] bits in ICMR1, the ICBRH, and ICBRL registers. In addition, specify the values in the DLCS bit in ICMR2 and the SDDL[2:0] bits in ICMR2 to meet the data hold time specification of 300 ns or more. When the IIC is used only as a slave device, the transfer rate setting is not required, but ICBRL must be set to a value longer than the data setup time (250 ns).

For the SMBus device default address (1100 001b), use one of the slave address registers L0 to L2 (SARL0, SARL1, and SARL2), and set the associated FS bit (7- or 10-bit address format select) in SARUy (y = 0 to 2) to 0 (7-bit address format).

When transmitting the UDID (Unique Device Identifier), set the SALE bit in ICFER to 1 to enable the slave arbitration-lost detection function.

39.13.1 SMBus Timeout Measurement

(1) Measuring slave device timeout

The following period (timeout interval: $T_{LOW:\ SEXT}$) must be measured for slave devices in SMBus communication:

- From start condition to stop condition.

To measure timeout for slave devices, measure the period from start condition detection to stop condition detection with the GPT using the IIC start condition detection interrupt (STIn) and stop condition detection interrupt (SPIn). The measured timeout period must be within the total clock low-level period [slave device] $T_{LOW:\ SEXT}$: 25 ms (maximum) of the SMBus standard.

If the time measured with the GPT exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (minimum) of the SMBus standard, the slave device must release the bus by writing 1 to the IICRST bit in ICCR1 to issue an internal reset of the IIC. When an internal reset is issued, the IIC stops driving the bus for the SCLn and SDAn pins, making them output high-impedance, which releases the bus.

(2) Measuring master device timeout

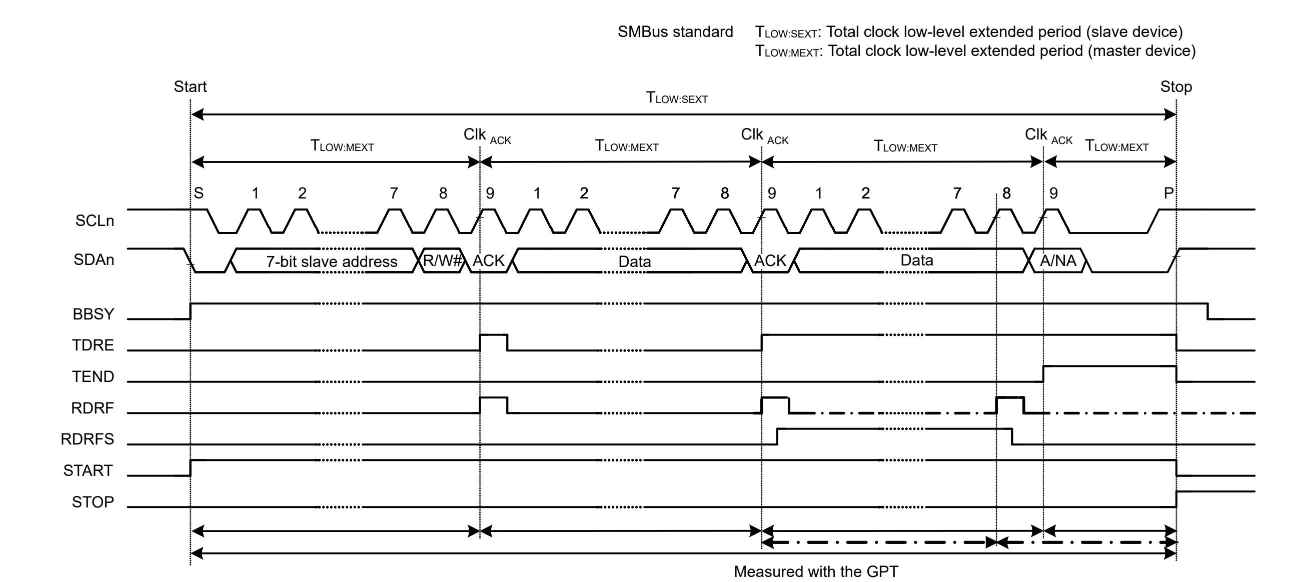
The following periods (timeout interval: $T_{LOW:\ MEXT}$) must be measured for master devices in SMBus communication:

- From start condition to acknowledge bit
- Between acknowledge bits
- From acknowledge bit to stop condition.

To measure timeout for master devices, measure these periods with the GPT using the IIC start condition detection interrupt (STIn), stop condition detection interrupt (SPIn), transmit end interrupt (IICn_TEI), or receive data full interrupt (IICn_RXI). The measured timeout period must be within the total clock low-level extended period (master device) $T_{LOW:\ MEXT}$: 10 ms (maximum) of the SMBus standard, and the total of all $T_{LOW:\ MEXT}$ values from start condition to stop condition must be within $T_{LOW:\ SEXT}$: 25 ms (maximum).

For the ACK receive timing (rising edge of the 9th SCL clock cycle), monitor the TEND flag in ICSR2 in master transmit mode (master transmitter) and the RDRF flag in ICSR2 in master receive mode (master receiver). Perform byte-wise transmit operations in master transmit mode, and hold the RDRFS bit in ICMR3 at 0 until the byte immediately before reception of the final byte in master receive mode. While the RDRFS bit is 0, the RDRF flag is set to 1 on the rising edge of the 9th SCL clock cycle.

If the period measured with the GPT exceeds the total clock low-level extended period (master device) $T_{LOW:\ MEXT}$: 10 ms (maximum) of the SMBus standard or the total of measured periods exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (minimum) of the SMBus standard, the master device must stop the transaction by issuing a stop condition. In master transmit mode, immediately stop the transmit operation (stop writing data to ICDRT).



Note: n = 0, 1

Figure 39.50 SMBus timeout measurement

39.13.2 Packet Error Code (PEC)

The MCU provides a CRC calculator that enables transmission of a Packet Error Code (PEC) or allows checking of the received data in SMBus data communication for the IIC. For the CRC-generating polynomials of the CRC calculator, see section 48, Cyclic Redundancy Check (CRC).

In master transmit mode, the PEC data can be generated by writing all transmit data to the CRC Data Input Register (CRCDIR) in the CRC calculator.

In master receive mode, the PEC data can be checked by writing all receive data to CRCDIR in the CRC calculator and comparing the obtained value in the CRC Data Output Register (CRCDOR) with the received PEC data.

To send ACK or NACK based on the match or mismatch result when the final byte is received as a result of the PEC code check, set the RDRFS bit in ICMR3 to 1 before the rising edge of the 8th SCL clock cycle during reception of the final byte, and hold the SCLn line low on the falling edge of the 8th clock cycle.

39.13.3 SMBus Host Notification Protocol (Notify ARP Master Command)

In communicating over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of its own slave address, or to request its own slave address from the SMBus host.

For a product using the MCU to operate as an SMBus host or ARP master, the host address (0001 000b) sent from the slave device must be detected as a slave address, and so the IIC provides a function for detecting the host address. To detect the host address as a slave address, set the SMBS bit in ICMR3 and the HOAE bit in ICSER to 1. Operation after the host address is detected is the same as normal slave operation.

39.14 Interrupt Sources

The IIC issues five types of interrupt requests:

- Transfer error or event generation (arbitration-lost, NACK detection, timeout detection, start condition detection, and stop condition detection)
- Receive data full
- Transmit data empty
- Transmit end
- Address match during wakeup function

Table 39.10 lists details about the interrupt requests. The receive data full and transmit data empty interrupts can activate data transfer by the DTC or DMAC.

Table 39.10 Interrupt sources

Symbol	Interrupt source	Interrupt flag	DTC or DMAC activation	Interrupt condition
IICn_EEI*5	Transfer error or event occurrence	AL	Not possible	AL = 1, ALIE = 1
		NACKF		NACKF = 1, NAKIE = 1
		TMOF		TMOF = 1, TMOIE = 1
		START		START = 1, STIE = 1
		STOP		STOP = 1, SPIE = 1
IICn_RXI*2 *5	Receive data full	RDRF	Possible	RDRF = 1, RIE = 1
IICn_TXI*1 *5	Transmit data empty	RDRF	Possible	TDRE = 1, TIE = 1
IICn_TEI*3 *5	Transmit end	TEND	Not possible	TEND = 1, TEIE = 1
IIC0_WUI*4	Slave address match during wakeup function	WUF	Not possible	Slave address match Slave receive complete RWAK operation ASY0 = 1 WUIE = 1

Note: There is a delay between the execution of a write instruction for a peripheral module by the CPU and the actual writing to the module. When an interrupt flag is cleared or masked, read the relevant flag again to check whether clearing or masking is complete, then return from interrupt handling. Not doing so creates the possibility of repeated processing of the same interrupt.

Note 1. Because IICn_TXI is edge-detected, it does not require clearing. Additionally, the TDRE flag in ICSR2 (condition for IICn_TXI) is automatically set to 0 when transmit data is written to the ICDRT register or a stop condition is detected (STOP flag = 1 in ICSR2).
Note 2. Because IICn_RXI is edge-detected, it does not require clearing. Additionally, the RDRF flag in ICSR2 (condition for IICn_RXI) is automatically set to 0 when data is read from ICDRR.
Note 3. When using the IICn_TEI interrupt, clear the TEND flag in ICSR2 in the IICn_TEI interrupt handling. The TEND flag in ICSR2 automatically is set to 0 when transmit data is written to the ICDRT register or a stop condition is detected (STOP flag = 1 in ICSR2).
Note 4. Only channel 0 has a wakeup function, so IIC0_WUI is for channel 0 only.
Note 5. Channel number (n = 0 to 2).

Clear or mask each flag during interrupt handling.

39.14.1 Buffer Operation for IICn_TXI and IICn_RXI Interrupts

If the conditions for generating an IICn_TXI or IICn_RXI interrupt are satisfied while the associated IR flag is 1, the interrupt request is not output for the ICU but is saved internally. One request per source can be saved internally.

An interrupt request that is saved in the ICU is output when the ICU.IELSRn.IR flag becomes 0. Internally saved interrupt requests are automatically cleared under normal conditions. They can also be cleared by writing 0 to the interrupt enable bit within the associated peripheral module.

39.15 State of Registers When Issuing Each Condition

The IIC has two dedicated resets, IIC reset and Internal reset. Table 39.11 lists the registers states when issuing each condition.

Table 39.11 Register states when issuing each condition (1 of 2)

Registers		Reset	IIC reset (ICE = 0, IICRST = 1)	Internal reset (ICE = 1, IICRST = 1)	Start or restart condition detection	Stop condition detection
ICCR1	ICE, IICRST	In reset	Saved	Saved	Saved	Saved
	SCLO, SDAO		In reset	In reset		
	Others			Saved		
ICCR2	BBSY	In reset	In reset	Saved	Set	In reset
	ST, RS			In reset	In reset	Saved
	SP					In reset
	TRS				Set or saved	
	MST					
ICMR1	BC[2:0]	In reset	In reset	In reset	In reset	Saved
	Others			Saved	Saved	
ICMR2		In reset	In reset	Saved	Saved	Saved
ICMR3	ACKBT	In reset	In reset	Saved	Saved	In reset
	Others					Saved
ICFER		In reset	In reset	Saved	Saved	Saved
ICSER		In reset	In reset	Saved	Saved	Saved
ICIER		In reset	In reset	Saved	Saved	Saved
ICSR1		In reset	In reset	In reset	Saved	In reset
ICSR2	TEND	In reset	In reset	In reset	Saved	In reset
	TDRE				Set or saved	
	START				Set	
	STOP				Saved	Set
	Others				Saved	Saved
ICWUR		In reset	In reset	Saved	Saved	Saved

Table 39.11 Register states when issuing each condition (2 of 2)

Registers		Reset	IIC reset (ICE = 0, IICRST = 1)	Internal reset (ICE = 1, IICRST = 1)	Start or restart condition detection	Stop condition detection
SARL0, SARL1, SARL2 SARU0, SARU1, SARU2		In reset	In reset	Saved	Saved	Saved
ICBRH, ICBRL		In reset	In reset	Saved	Saved	Saved
ICDRT		In reset	In reset	Saved	Saved	Saved
ICDRR		In reset	In reset	Saved	Saved	Saved
ICDRS		In reset	In reset	In reset	Saved	Saved
Timeout function		In reset	In reset	In reset	Operating	Operating
Bus free time measurement		In reset	In reset	Operating	Operating	Operating
ICWUR2	WUSEN	In reset	In reset	Saved	Saved	Saved
	Others					Saved or set or reset

39.16 Event Link Output

The IIC0 module handles the event output for the Event Link Controller (ELC) for the following sources:

(1) Transfer error event

When a transfer error event occurs, the associated event signal can be output to another module by the ELC.

(2) Receive data full

When a receive data register becomes full, the associated event signal can be output to another module by the ELC.

(3) Transmit data empty

When a transmit data register becomes empty, the associated event signal can be output to another module by the ELC.

(4) Transmit end

On completion of the transfer, the associated event signal can be output to another module by the ELC.

39.16.1 Interrupt Handling and Event Linking

Each of the IIC interrupt types (see Table 39.10) has an enable bit to control enabling and disabling of the associated interrupt signal. An interrupt request signal is output to the CPU when an interrupt source condition is satisfied while the associated enable bit is set.

The associated event link output signals are sent to other modules as event signals by the ELC when the interrupt source conditions are satisfied, regardless of the interrupt enable bit settings. For details on interrupt sources, see Table 39.10.

39.17 Usage Notes

39.17.1 Settings for the Module-Stop Function

The Module Stop Control Register B (MSTPCRB) can enable or disable IIC operation. The IIC is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Mode.

39.17.2 Notes on Starting Transfer

If the IR flag associated with the IIC interrupt is 1 when transfer is started (ICCR1.ICE bit = 1), follow the procedure in this section to clear the interrupts before enabling operations. Starting transfer with the IR flag set to 1 while the ICCR1.ICE bit is 1 leads to an interrupt request being internally saved after transfer starts, and this can lead to unexpected behavior of the IR flag.

To clear interrupts before starting transfer operation:

1. Confirm that the ICCR1.ICE bit is 0.

2. Set the relevant interrupt enable bits, such as ICIER.TIE to 0.
3. Read the relevant interrupt enable bits, such as ICIER.TIE, and confirm that the value is 0.
4. Set the IR flag to 0.

39.17.3 Software Standby Mode

When using a wakeup function, the following setting is not required:

- When the IIC is in communication, set ICCR1.ICE to 0 to stop communication before transitioning to Software Standby mode.
- After returning from Software Standby mode, set ICCR1.ICE to 1 to enable communication.

40. I3C Bus Interface (I3C)

Specific HDR Modes are not covered by conformance test suite (CTS).

When CTS for HDR Modes is available, Renesas will confirm CTS tests.

40.1 Overview

40.1.1 Functional Overview

The I3C bus interface (I3C) has 1 channel. The I3C module conform with and provide a subset of the NXP I^2C (Inter-Integrated Circuit) bus interface functions and a subset of the MIPI I3C.

In this section, PCLK refers to PCLKA, TCLK refers to I3CCLK.

Table 40.1 lists the I^2C specifications, and Table 40.2 lists the I3C specifications.

Table 40.1 I^2C specifications (1 of 2)

Parameter	Specifications
Operation mode	Master mode and slave mode selectable
Data handler	Single buffer transfer

Communication protocol	• I^2C bus format
	  – Standard-mode (Sm) : 0 to 100 kbps
	  – Fast-mode (Fm) : 0 to 400 kbps
	  – Fast-mode Plus (Fm+) : 0 to 1 Mbps
	  – High-speed mode (Hs-mode) : 0 to 3.4 Mbps
	• SMBus format : 10 to 100 kbps

Address format	• 7-bit address • 10-bit address
Address detection	• Slave address (static address) (max 3 address) • General call address • Hs-mode master code • Device ID • Host address • 10-bit slave addressing
Clock stretching	Clock stretching capability
Noise-filter	• Analog noise-filter • Digital noise-filter
Interrupt source	• Normal Rx data buffer full • Normal Tx data buffer empty • START condition detection • STOP condition detection • Transmit end • NACK detection • Arbitration lost • Timeout detection • Wake-up condition detection
Error detection	• Non-recoverable internal error • NACK received • Receive overflow or transfer underflow error • Arbitration lost error • Timeout error
Event link output	• Normal Rx data buffer full event • Normal Tx data buffer empty event • START condition event • STOP condition event • Transmit end event • NACK event • Arbitration lost event • Timeout event
Wake-up source	Address detection of slave address

Table 40.1 I^2C specifications (2 of 2)

Parameter	Specifications
Module-stop function	Module-stop state can be set to reduce power consumption
Trust Zone Filter	Security and Privilege attribution can be set

Table 40.2 I3C specifications (1 of 2)

Parameter	Specifications
Operation mode	Master (main master/secondary master) mode and slave mode selectable

Data handler	• Master :
	  – High priority FIFO buffer transfer
	  – Normal FIFO buffer transfer
	• Slave :
	  – Normal FIFO buffer transfer

Communication protocol	• SDR (I3C Single Data Rate) mode : up to 12.5 Mbps
	  – Private message
	  – Broadcast message (common command code)
	  – Direct message (common command code)
	• HDR (I3C High Data Rate) mode
	  – HDR-DDR (Dual Data Rate) mode : up to 25 Mbps
	  – HDR-TSL (Ternary Symbol Legacy) mode : up to 27.5 Mbps
	  – HDR-TSP (Ternary Symbol Pure-bus) mode : up to 39.5 Mbps
	• Legacy I^2C message
	  – Fast-mode (Fm) : 0 to 400 kbps
	  – Fast-mode Plus (Fm+) : 0 to 1 Mbps

In-band interrupt	• Slave interrupt request • Master ship request (secondary master only)
Address format	7-bit address
Address detection	• Slave address (static address or dynamic address) • Broadcast address (0x7E)
Clock stalling	Clock stalling capability

Timing control	• Synchronous timing control
	  – Sync mode : Synchronous basic mode
	• Asynchronous timing control
	  – Async mode 0 : Asynchronous basic mode
	  – Async mode 1 : Asynchronous advanced mode

Interrupt source	• Non-recoverable internal error • Normal Transfer error • Normal Transfer abort • Normal Response queue full • Normal Command queue empty • Normal IBI Queue Empty/Full • Normal Rx Data buffer full • Normal Tx Data buffer empty • Normal Receive status queue full • High Priority Response queue full • High Priority Command queue empty • High Priority Rx Data buffer full • High Priority Tx Data buffer empty • High Priority Transfer Error • High Priority Transfer Abort • START condition detection • STOP condition detection • HDR exit pattern detection • Timeout detection • Synchronous Timing • MREF Counter Overflow • MREF Capture • Additional Master-initiated bus Event • Wake-up condition detection

Table 40.2 I3C specifications (2 of 2)

Parameter	Specifications
Error detection	• Non-recoverable internal error • CRC error • Parity error • Frame error • Address header error • Address NACKed or dynamic address assignment NACKed • Receive overflow or transfer underflow error • Aborted • NACK received for the I^2C write data transfer • Timeout error
Event link output	• Normal Response queue full event • Normal Command queue empty event • Normal IBI Queue Empty/Full • Normal Rx data buffer full event • Normal Tx data buffer empty event • Normal Receive Status queue full event • High Priority Response queue full event • High Priority Command queue empty event • High Priority Rx Data buffer full event • High Priority Tx Data buffer empty event • Non-recoverable internal error event • Normal Transfer Error event • Normal Transfer Abort event • High Priority Transfer Error event • High Priority Transfer Abort event • START condition event • STOP condition event • HDR exit pattern event • Timeout event • Synchronous timing event • MREF counter overflow event • MREF capture event • Additional master-initiated bus Event
Wake-up source	• Master : SDA assert of IBI (START condition detection) • Slave : Address detection of broadcast address (0x7E) and slave address
Module-stop function	Module-stop state can be set to reduce power consumption
Trust Zone Filter	Security and Privilege attribution can be set

Table 40.3 I3C I/O pins

Channel	Pin name	I/O	Function
I3C	I3C_SCL0	I/O	I3C serial clock I/O pin
	I3C_SDA0	I/O	I3C serial data I/O pin

Note: In this section, I3C_SCL refers to I3C_SCL0, I3C_SDA refers to I3C_SDA0.

40.1.2 Block Diagram

Figure 40.1 shows the main components of this I3C.

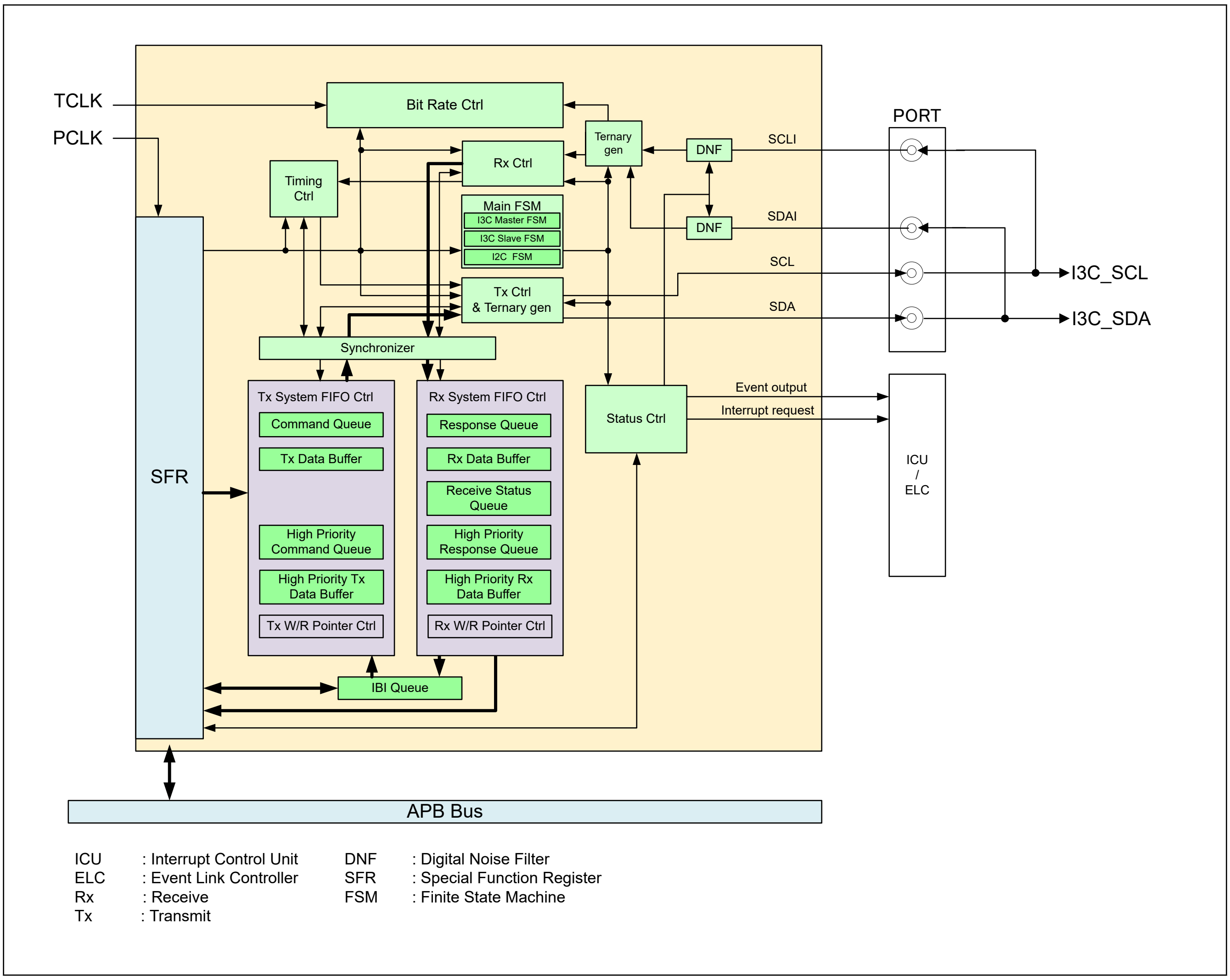


Figure 40.1 I3C block diagram

40.2 Registers

40.2.1 PRTS : Protocol Selection Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PRTMD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	PRTMD	Protocol Mode 0: I3C protocol mode 1: I^2C protocol mode	R/W

Bit	Symbol	Function	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

PRTMD bit (Protocol Mode)

PRTMD = 0 : I3C FIFO buffer transfer (Equivalent to HCI)

PRTMD = 1 : I^2C single buffer transfer

There are notes about rewriting the PRTS.PRTMD bit, see section 40.7.4. Notes on Rewriting PRTS.PRTMD Bit.

40.2.2 BCTL : Bus Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	BUSE	RSM	ABT	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	BMDS	—	—	—	—	—	—	INCBA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	INCBA	Include I3C Broadcast Address*1 0: Do not include I3C broadcast address for private transfers 1: Include I3C broadcast address for private transfers	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	BMDS	Bus Mode Selection*1 0: Legacy inclusive Bus mode disabled 1: Legacy inclusive (mix) Bus mode enabled	R/W
28:8	—	These bits are read as 0. The write value should be 0.	R/W
29	ABT	Abort*1 0: I3C is running. 1: I3C has aborted a transfer.	R/W
30	RSM	Resume*2 Values when read: 0: I3C is running. 1: I3C is suspended.	R/W
31	BUSE	Bus Enable 0: I3C bus operation is disabled. 1: I3C bus operation is enabled.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports I3C master mode and I3C secondary master mode.
Note 2. This bit supports all I3C mode.

INCBA bit (Include I3C Broadcast Address)

This bit controls whether the I3C broadcast address (0x7E) is included for private transfers.

If the I3C broadcast address is not included for private transfers, then IBIs driven from Slaves might not win the arbitration, potentially delaying acceptance of the IBIs.

BMDS bit (Bus Mode Selection)

Indicates whether or not any Legacy I^2C Devices are present on the I3C Bus. Whenever this bit is set, I3C shall use the HDR-TSL protocol (Ternary Symbol Legacy Inclusive Bus) for HDR-TS Transfers.

The specific mode of the transfer (SDR, DDR, HDRTS) is controlled on a per-Command basis.

I3C shall use this field to select either TSL (1) or TSP (0) for HDR-TS transfers.

ABT bit (Abort)

When set to 1, this bit allows I3C to relinquish control of the I3C Bus before completing the currently issued transfer.

In response to an ABORT request, I3C issues the STOP condition on the I3C Bus after the complete data byte is transferred or received.

The Driver shall clear the ABT bit to allow operation on the Bus.

If BCTL.ABT is set and ABORT processing is performed, please ignore ERR_STATUS of Response Descriptor.

RSM bit (Resume)

This bit is used to resume I3C operation following the Halt state.

I3C enters the Halt state (as indicated in register PRSTDBG) as a result of any type of error occurring in a transfer.

The error type is indicated by the field ERR_STATUS in register NRSPQP, HRSPQP, NRSQP and NIBIQP.

After I3C has entered the Halt state, the application must write the value 1 to the RSM bit to resume I3C operation. I3C shall auto-clear the RSM bit once it has resumed making transfers (it has initiated the next Command).

BUSE bit (Bus Enable)

Enables or disables the operation on the I3C Bus by I3C.

Set the BUSE bit to 1 when using I3C. The I3C_SCL and I3C_SDA pins are placed in the active state when the BUSE bit is set to 1. Set the BUSE bit to 0 when I3C is not to be used. The I3C_SCL and I3C_SDA pins are placed in the inactive state when the BUSE bit is set to 0.

If the software sets this bit, then it also confirms that initialization is done, and that I3C can use the programmed register values (For example, generation of SCL on IBI detection, etc.). If this bit is not set, then I3C shall not generate SCL for incoming IBI.

Software may disable I3C bus operation while it is active, However:

- If a disable request occurs while receiving IBI, the actual disabling will not occur until reception of the IBI is complete.
- When the software reads the value 0 from this field, this indicates that I3C bus operation disable operation has completed.
  If commands remain in the command queue, do not set BUSE = 0.

40.2.3 MSDVAD : Master Device Address Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x018

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MDYADV	—	—	—	—	—	—	—	—	MDYAD[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
22:16	MDYAD[6:0]	Master Dynamic Address	R/W
30:23	—	These bits are read as 0. The write value should be 0.	R/W
31	MDYADV	Master Dynamic Address Valid 0: The master dynamic address field is not valid. 1: The master dynamic address field is valid.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode.

MDYAD[6:0] bits (Master Dynamic Address)

This field is used to program I3C master dynamic address. I3C uses this address to respond to master transactions in I3C interface mode (slave or secondary master role).

In I3C main master mode, the software shall program the dynamic address as it self-assigns its dynamic address.

MDYADV bit (Master Dynamic Address Valid)

This bit indicates whether or not the value in the MDYAD field is valid.

In I3C main master mode, the user sets this bit to 1 as it self-assigns its dynamic address.

Note: After setting MSDVAD, and setting BCTL. BUSE = 1, the device will act as main master.

Without setting MSDVAD, setting SVDCT.TBCR76[1:0] = 00b (Device Role Slave), and setting BCTL.BUSE = 1, the device will act as slave.

Without setting MSDVAD, setting MSDCTm.RBCR76[1:0] = 01b (Device Role Master), and setting BCTL.BUSE = 1, the device will act as slave.

40.2.4 RSTCTL : Reset Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x020

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	INTLRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	HRDBRST	HTDBRST	HRSPQRST	HCMDQRST	—	—	RSQRST	IBIQRST	RDBRST	TDBRST	RSPQRST	CMDQRST	RI3CRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RI3CRST	I3C Software Reset 0: Release I3C reset. 1: Initiate I3C reset.	R/W
1	CMDQRST	Normal Command Queue Software Reset*1 0: The Normal Command Queues in I3C is not flushed. 1: The Normal Command Queues in I3C is flushed.	R/W
2	RSPQRST	Normal Response Queue Software Reset*1 0: The Normal Response Queues in I3C is not flushed. 1: The Normal Response Queues in I3C is flushed.	R/W
3	TDBRST	Normal Tx Data Buffer Software Reset*1 0: The Normal Tx Data buffers in I3C is not flushed. 1: The Normal Tx Data buffers in I3C is flushed.	R/W

Bit	Symbol	Function	R/W
4	RDBRST	Normal Rx Data Buffer Software Reset*1 0: The Normal Rx Data buffers in I3C is not flushed. 1: The Normal Rx Data buffers in I3C is flushed.	R/W
5	IBIQRST	Normal IBI Queue Software Reset*1 0: The Normal IBI Queues in I3C is not flushed. 1: The Normal IBI Queues in I3C is flushed.	R/W
6	RSQRST	Normal Receive Status Queue Software Reset*2 0: The Normal Receive Status Queue in I3C is not flushed. 1: The Normal Receive Status Queue in I3C is flushed.	R/W
8:7	—	These bits are read as 0. The write value should be 0.	R/W
9	HCMDQRST	High Priority Command Queue Software Reset*3 0: The High Priority Command Queues in I3C is not flushed. 1: The High Priority Command Queues in I3C is flushed.	R/W
10	HRSPQRST	High Priority Response Queue Software Reset*3 0: The High Priority Response Queues in I3C is not flushed. 1: The High Priority Response Queues in I3C is flushed.	R/W
11	HTDBRST	High Priority Tx Data Buffer Software Reset*3 0: The High Priority Tx Data buffers in I3C is not flushed. 1: The High Priority Tx Data buffers in I3C is flushed.	R/W
12	HRDBRST	High Priority Rx Data Buffer Software Reset*3 0: The High Priority Rx Data buffers in I3C is not flushed. 1: The High Priority Rx Data buffers in I3C is flushed.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
16	INTLRST	Internal Software Reset 0: Releases of some registers and internal state. 1: Resets of some registers and internal state.	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports all I3C mode.
Note 2. This bit supports I3C secondary master mode and I3C slave mode.
Note 3. This bit supports I3C master mode and I3C secondary master mode.

For details on reset for each register, see section 40.6. Reset Descriptions.

RI3CRST bit (I3C Software Reset)

On Driver setting this bit to 1, I3C shall be reset and disabled.

All registers shall return to their reset values, and the software shall re-initialize I3C.

This field is cleared automatically upon I3C reset completion. This field also resets all Queues in I3C.

Note: Programming this field while it contains a value of 1 may result in undefined behavior.

CMDQRST bit (Normal Command Queue Software Reset)

On software setting this bit to 1, the Normal Command Queues in I3C shall be flushed.

This field shall be cleared automatically upon Normal Command Queue reset completion.

RSPQRST bit (Normal Response Queue Software Reset)

On software setting this bit to 1, the Normal Response Queues in I3C shall be flushed.

This field shall be cleared automatically upon Normal Response Queue reset completion.

TDBRST bit (Normal Tx Data Buffer Software Reset)

On software setting this bit to 1, the Normal Tx Data Buffers in I3C shall be flushed.

This field shall be cleared automatically upon Normal Tx Data Buffer reset completion.

RDBRST bit (Normal Rx Data Buffer Software Reset)

On software setting this bit to 1, the Normal Rx Data Buffers in I3C shall be flushed.

This field shall be cleared automatically upon completion of Normal Rx Data Buffer reset.

IBIQRST bit (Normal IBI Queue Software Reset)

On software setting this bit to 1, the Normal IBI Queues in I3C shall be flushed.

This field shall be cleared automatically upon completion of Normal IBI Queue reset.

RSQRST bit (Normal Receive Status Queue Software Reset)

On software setting this bit to 1, the Normal Receive Status Queues in I3C shall be flushed.

This field shall be cleared automatically upon Normal Receive Status Queue reset completion.

HCMDQRST bit (High Priority Command Queue Software Reset)

On software setting this bit to 1, the High Priority Command Queues in I3C shall be flushed.

This field shall be cleared automatically upon High Priority Command Queue reset completion.

HRSPQRST bit (High Priority Response Queue Software Reset)

On software setting this bit to 1, the High Priority Response Queues in I3C shall be flushed.

This field shall be cleared automatically upon High Priority Response Queue reset completion.

HTDBRST bit (High Priority Tx Data Buffer Software Reset)

On software setting this bit to 1, the High Priority Tx Data Buffers in I3C shall be flushed.

This field shall be cleared automatically upon High Priority Tx Data Buffer reset completion.

HRDBRST bit (High Priority Rx Data Buffer Software Reset)

On software setting this bit to 1, the High Priority Rx Data Buffers in I3C shall be flushed.

This field shall be cleared automatically upon completion of High Priority Rx Data Buffer reset.

INTLRST bit (Internal Software Reset)

When set to 1, some of the registers are reset. For details on the registers to be reset, see section 40.6. Reset Descriptions.

Note: When set internal software reset during bus operation enable, use DISEC CCC in advance to disable IBI transmission to I3C Slave in order to avoid conflict with IBI from I3C Slave connected to I3C Bus.

40.2.5 PRSST : Present State Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x024

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	PRSSTWP	—	—	TRMD	—	CRMS	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
2	CRMS	Current Master*2 0: The Master is not the Current Master, and must request and acquire bus ownership before initiating any transfer. 1: The Master is the Current Master, and as a result can initiate transfers.	R/W*1
3	—	This bit is read as 0. The write value should be 0.	R/W
4	TRMD	Transmit/Receive Mode*3 0: Receive mode 1: Transmit mode	R
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	PRSSTWP	Present State Write Protect*2 0: CRMS bit is protected. 1: CRMS bit can be written when writing simultaneously with the value of the target bit.	W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. When the PRSSTWP bit is set to 1, the CRMS bit can be written to.
Note 2. This bit supports I^{2}C, I3C master, and I3C secondary master mode.
Note 3. This bit supports I^{2}C mode.

CRMS bit (Current Master)

Indicates the set condition and reset condition of each operation mode.

Operation Mode [I^{2}C/I3C common]

[Clearing conditions]

- When 0 written to the PRSST.CRMS by the software.

[Setting conditions]

- When 1 written to the PRSST.CRMS by the software.

Operation Mode [I^{2}C]

[Clearing conditions]

- When STOP is issued.
- When Master Arbitration-Lost.

[Setting conditions]

- When START is issued.

Operation Mode [I3C Main Master]

[Clearing conditions]

- When 0 written to the MSDVAD.MDYADV by the software.
- When GETACCMST transmission is successfully completed by issuing STOP, after responding ACK to the Mastership-Request received from the Secondary Master.

[Setting conditions]

- When 1 written to the MSDVAD.MDYADV by the software.
- When GETACCMST reception is successfully completed by issuing STOP, after the ACK is responded to the Mastership-Request transmitted to the Secondary Master.

Operation Mode [I3C Secondary Master]

[Clearing condition]

- When GETACCMST transmission is successfully completed by issuing STOP, after responding ACK to the Mastership-Request received from the Non-Current Master.

[Setting condition]

- When GETACCMST reception is successfully completed by issuing STOP, after the ACK is responded to the Mastership-Request transmitted to the Current Master.

The PRSST register returns I3C current state.

State has two parts: this register which is mandatory, and an additional optional PRSST_DEBUG register intended for debug purposes (see the Debug Capability registers in the Extended Capabilities list).

TRMD bit (Transmit/Receive Mode)

This bit indicates transmit or receive mode.

I3C is in receive mode when the TRMD bit is set to 0 and is in transmit mode when the bit is set to 1. Combination of this bit and the CRMS bit indicates the operating mode of I3C.

The value of TRMD bit is automatically changed to 1 for transmit mode or 0 for receive mode by issuing or detection of a START condition and setting of the R/W# bit.

[Setting conditions]

- When a START condition is issued normally according to the START condition issuance request (when a START condition is detected with the CNDCTL.STCND bit set to 1).
- When a Repeated START condition is issued normally according to the Repeated START condition issuance request (when a Repeated START condition is detected with the CNDCTL.SRCND bit set to 1).
- When the R/W# bit added to the slave address is set to 0 in master mode.
- When the address received in slave mode matches the address enabled in SVCTL, with the R/W# bit set to 1.

[Clearing conditions]

- When a STOP condition is detected.
- The ALF (arbitration-lost) flag in BST being set to 1.
- In master mode, reception of a slave address to which an R/W# bit with the value 1 is appended.
- In slave mode, a match between the received address and the address enabled in SVCTL when the value of the received R/W# bit is 0 (including cases where the received address is the general call address).
- In slave mode, a Repeated START condition is detected (a Repeated START condition is detected with BCST.BFREF = 0 and CRMS = 0).

PRSSTWP bit (Present State Write Protect)

PRSSTWP is always 0 when reading.

When writing to PRSST, writing 1 to this bit at the same time enables writing to CRMS bit.

40.2.6 INST : Internal Status Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x030

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	INEF	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	—	These bits are read as 0. The write value should be 0.	R/W
10	INEF	Internal Error Flag 0: I3C Internal Error has not detected. 1: I3C Internal Error has detected.	R/W*1
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports all I3C mode.
Note 1. Clearing (to 0) condition : Writing 0 after the 1 state is read.

The Interrupt Status register reflects the status of outstanding interrupts.

The status fields are either write 0 to clear, or else are cleared based on queue operations.

INEF bit (Internal Error Flag)

When this bit is 1, it indicates that I3C Internal Error has detected.

When this bit is 0, it indicates that I3C Internal Error has not detected.

[Setting conditions]

- The following 1 is satisfied and any of the following 2 to 9 are satisfied.
  1. The INSTE.INEE bit = 1
  2. When transmit data is written to the Tx Data Buffer that is completely full.
  3. When received data is read from the Rx Data Buffer that is completely empty.
  4. When Command Descriptor is written to the Command Queue that is completely full.
  5. When Response Descriptor is read from the Response Queue that is completely empty.
  6. When Receive Status Descriptor is read from the Receive Status Queue that is completely empty.
  7. When IBI Status Descriptor is read from the IBI Queue under the condition that the IBI Queue is completely empty and PRSST.CRMS = 1.
  8. When IBI Data is written to the IBI Queue under the condition that the IBI Queue is completely full and PRSST.CRMS = 0.
  9. When the Response Queue, IBI Queue or Receive Status Queue overflows.

[Clearing condition]

- When 0 is written to the INEF bit after reading INEF bit = 1.

40.2.7 INSTE : Internal Status Enable Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x034

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	INEE	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
10	INEE	Internal Error Enable 0: Disable INST.INEF 1: Enable INST.INEF	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports all I3C mode.

INEE bit (Internal Error Enable)

When this bit set to 1, it enables detection of I3C Internal Error.

When this bit set to 0, it disables detection of I3C Internal Error.

40.2.8 INIE : Internal Interrupt Enable Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x038

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	INEIE	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	—	These bits are read as 0. The write value should be 0.	R/W
10	INEIE	Internal Error Interrupt Enable 0: Disables Non-recoverable Internal Error Interrupt Signal. 1: Enables Non-recoverable Internal Error Interrupt Signal.	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports all I3C mode.

INEIE bit (Internal Error Interrupt Enable)

When set to 1 and register INEF is set, the hardware Controller asserts an interrupt to the Host.

40.2.9 INSTFC : Internal Status Force Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x03C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	INEFC	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	—	The write value should be 0.	W
10	INEFC	Internal Error Force 0: Not force a specific interrupt 1: Force a specific interrupt	W
31:11	—	The write value should be 0.	W

Note: S-TYPE3, P-TYPE3
Note: This register supports all I3C mode.

INEFC bit (Internal Error Force)

For debug, helps to force this interrupt.

40.2.10 DVCT : Device Characteristic Table Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x044

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	IDX[4:0]					—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
18:0	—	These bits are read as 0.	R
23:19	IDX[4:0]	DCT Table Index Current index of the DCT, which is used as the starting index for the I3C ENTDAA CCC.	R
31:24	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

IDX[4:0] bits (DCT Table Index)

Once the complete characteristics of device that won the arbitration are written to the DCT (during ENTDAA using Address Assignment Command) this index is incremented by 1.

Note: How to check the progress of ENTDAA using this bit:

1. Read the value of this bit before setting the Command Descriptor for issuing the ENTDAA command.
2. After starting the ENTDAA command, until the value of this bit is updated (that is, it changes from the value read in advance), it indicates that the Dynamic Address is being assigned to the device specified by the first index value (value set in DEV_INDEX[4:0] of Command Descriptor).
3. After the value of this bit is updated, it indicates that Dynamic Address is being assigned according to the value set in DEV_INDEX[4:0] and DEV_COUNT[3:0] of Command Descriptor to the device of the first index value or later.

40.2.11 IBINCTL : IBI Notify Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x058

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	NRSIRCTL	—	NRMRCTL	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	NRMRCTL	Notify Rejected Master Request Control 0: Do not pass rejected IBI Status to the Normal IBI Queue, if the incoming Master Request is NACKed and is auto-disabled based on DVMRRJ field in relevant DAT entry. 1: Pass rejected IBI Status to the Normal IBI Queue if the incoming Master Request is NACKed and is auto-disabled based on DVMRRJ field in relevant DAT entry.	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	NRSIRCTL	Notify Rejected Slave Interrupt Request Control 0: Do not pass rejected IBI Status to the Normal IBI Queue, if the incoming SIR is NACKed and is auto-disabled based on DVSIRRJ field in relevant DAT entry. 1: Pass rejected IBI Status to the Normal IBI Queue, if the incoming SIR is NACKed and is auto-disabled based on DVSIRRJ field in relevant DAT entry.	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

NRMRCTL bit (Notify Rejected Master Request Control)

Enables or disables reporting rejection of individual Master Requests.

NRSIRCTL bit (Notify Rejected Slave Interrupt Request Control)

Enables or disables reporting rejection of individual Slave Interrupt Requests (SIR).

40.2.12 BFCTL : Bus Function Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x060

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	HSME	FMPE	—	SMBS	—	—	—	SCSYNE	—	—	—	—	—	SALE	NALE	MALE
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	MALE	Master Arbitration-Lost Detection Enable 0: Master arbitration-lost detection disables. Disables the arbitration-lost detection function and does not clear the CRMS and TRMD bits in PRSST automatically when arbitration is lost. 1: Master arbitration-lost detection enables. Enables the arbitration-lost detection function and clears the CRMS and TRMD bits in PRSST automatically when arbitration is lost.	R/W
1	NALE	NACK Transmission Arbitration-Lost Detection Enable 0: NACK transmission arbitration-lost detection disables. 1: NACK transmission arbitration-lost detection enables.	R/W
2	SALE	Slave Arbitration-Lost Detection Enable 0: Slave arbitration-lost detection disables. 1: Slave arbitration-lost detection enables.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	SCSYNE	SCL Synchronous Circuit Enable 0: No SCL synchronous circuit uses. 1: An SCL synchronous circuit uses.	R/W
11:9	—	These bits are read as 0. The write value should be 0.	R/W
12	SMBS	SMBus/I^{2}C Bus Selection 0: The I^{2}C bus select. 1: The SMBus select.	R/W
13	—	This bit is read as 0. The write value should be 0.	R/W
14	FMPE	Fast-mode Plus Enable 0: No Fm+ slope control circuit uses for the I3C_SCL pin and I3C_SDA pin. (n = 0) 1: An Fm+ slope control circuit uses for the I3C_SCL pin and I3C_SDA pin. (n = 0)	R/W
15	HSME	High Speed Mode Enable 0: Disable High Speed Mode. 1: Enable High Speed Mode.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports for I2C mode.

MALE bit (Master Arbitration-Lost Detection Enable)

This bit is used to specify whether to use the arbitration-lost detection function in master mode. Normally, set this bit to 1.

NALE bit (NACK Transmission Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when ACK is detected during transmission of NACK in receive mode (such as when slaves with the same address exist on the bus or when two or more masters select the same slave device simultaneously with different number of receive bytes).

SALE bit (Slave Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when a value different from the value being transmitted is detected on the bus in slave transmit mode (such as when slaves with the same address exist on the bus or when a mismatch with the transmit data occurs due to noise).

SCSYNE bit (SCL Synchronous Circuit Enable)

This bit is used to specify whether to synchronize the SCL clock with the SCL input clock. Normally, set this bit to 1.

When the SCSYNE bit set to 0 (no SCL synchronous circuit used), I3C does not synchronize the SCL clock with the SCL input clock. In this setting, I3C outputs the SCL clock with the transfer rate set in STDBR and EXTBR regardless of the I3C_SCL line state. For this reason, if the bus load of the I^{2}C bus line is much larger than the specification value or if the SCL clock output overlaps in multiple masters, the short-cycle SCL clock that does not meet the specification may be output. When no SCL synchronous circuit uses, it also affects the issuance of a START condition, Repeated START condition, and STOP condition, and the continuous output of extra SCL clock cycles.

This bit must not be set to 0 except for checking the output of the set transfer rate.

FMPE bit (Fast-mode Plus Enable)

This bit is used to specify whether to use a slope control circuit for Fast-mode Plus [Fm+].

When this bit is set to 1, a slope control circuit conforming to the Fast-mode Plus [Fm+] slope control specification (tof) of the I3C-bus is selected. When this bit is set to 0, a slope control circuit conforming to the Standard-mode [Sm] and Fast-mode [fm] slope control specification (tof) of the I3C-bus is selected.

Set this bit to 1 when using the transmission rate within a range up to 1 Mbps (Fast-mode Plus [Fm+]) of the I3C-bus specification. Set this bit to 0 when using the transmission rate at other rates (up to 100 kbps [Sm], up to 400 kbps [Fm]) or for SMBus (10 to 100 kbps).

Note: When communicating in Hs-mode, set as follows.

- Set FMPE to 0 when sending Hs-mode master code (0000 1XXXb) with Fast-mode.
- Set FMPE to 1 when sending Hs-mode master code (0000 1XXXb) with Fast-mode Plus.

HSME bit (High Speed Mode Enable)

This bit is used for communicating in Hs-mode.

When this bit is set to 1, the Hs-mode master code is recognized and Hs-mode communication is possible.

After the START condition is detected, if Hs-mode master code (0000 1XXXb) transmission is recognized, Hs-mode communication starts from Repeated START after receiving the NACK response.

It communicates at the bit rate set in STDBR until the NACK response, and automatically switches from Repeated START condition issuance after receiving the NACK response to the bit rate set in EXTBR.

Hs-mode continues until a STOP condition is detected.

When the STOP condition is detected, the bit rate is automatically switched to the bit rate set in STDBR.

Note: When this bit is set to 1, the BST.NACKDF bit will not be set even if a NACK response is received after sending the Hs-mode master code.

40.2.13 SVCTL : Slave Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x064

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SVAE[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	HOAE	—	—	—	—	—	—	—	—	DVIDE	HSMCE	—	—	—	—	GCAE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	GCAE	General Call Address Enable*1 0: General call address detection disables. 1: General call address detection enables.	R/W
4:1	—	These bits are read as 0. The write value should be 0.	R/W
5	HSMCE	Hs-mode Master Code Enable*1 0: Hs-mode Master Code Detection disables. 1: Hs-mode Master Code Detection enables.	R/W
6	DVIDE	Device-ID Address Enable*1 0: Device-ID address detection disables. 1: Device-ID address detection enables.	R/W
14:7	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
15	HOAE	Host Address Enable*1 0: Host address detection disables. 1: Host address detection enables.	R/W
18:16	SVAE[2:0]	Slave Address Enable n (n = 0 to 2)*2 0: Slave n disables 1: Slave n enables	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

Note 1. This bit supports I2C mode.

Note 2. These bits support I2C, I3C secondary master, and I3C slave mode.

GCAE bit (General Call Address Enable)

This bit is used to specify whether to ignore the general call address (0000 000 + 0 (write): All 0) when it is received. When this bit is set to 1, if the received slave address matches the general call address, I3C recognizes the received slave address as the general call address independently of the slave addresses set in the SVDVADn.SVAD[9:0] bits (n = 0 to 2) and performs data receive operation.

When this bit is set to 0, the received slave address is ignored even if it matches the general call address.

HSMCE bit (Hs-mode Master Code Enable)

This bit is used to specify whether to recognize and execute the Hs-mode master code (00001xxxb) is received in the first byte after a START condition is detected.

When this bit is set to 1, if the received first byte matches the Hs-mode master code, I3C recognizes that the Hs-mode master code has been received.

The first byte after Repeated START after NACK response to Hs-mode master code is recognized as a slave address and compared with the slave address set by SVDVADn.SVAD[9:0]bits (n = 0 to 2).

If the addresses match, the transmission / reception operation continues according to the R/W# bit value.

Hs-mode continues until a STOP condition is detected.

When this bit is set to 0, I3C will ignore the pattern until a STOP condition is detected, even if it matches the Hs-mode master code.

Note: When this bit is set to 1, SCSTRCTL.ACKTWE bit must be set to 0 and SCSTRCTL.RWE bit must be set to 1.

DVIDE bit (Device-ID Address Enable)

This bit is used to specify whether to recognize and execute the Device-ID address when a device ID (1111 100) is received in the first byte after a START condition or Repeated START condition is detected.

When this bit is set to 1, if the received first byte matches the Device-ID, I3C recognizes that the Device-ID address has been received. When the following R/W# bit is 0 (write), I3C recognizes the second and the following bytes as slave addresses and continues the receive operation.

When this bit is set to 0, I3C ignores the received first byte even if it matches the Device ID address and recognizes the first byte as a normal slave address.

For details on the Device-ID address detection, see (3) Device-ID Address Detection [I2C mode].

HOAE bit (Host Address Enable)

This bit is used to specify whether to ignore received host address (0001 000) when the BFCTL.SMBS bit = 1.

When this bit is set to 1 while the SMBS bit = 1, if the received slave address matches the host address, I3C recognizes the received slave address as the host address independently of the slave addresses set in the SVDVADn.SVAD[9:0] bits (n = 0 to 2) and performs the receive operation.

When the SMBS bit or the HOAE bit is set to 0, the received slave address is ignored even if it matches the host address.

SVAE[2:0] bits (Slave Address Enable n (n = 0 to 2))

This bit is used to enable or disable the slave address set in the SVDVADn.SVAD[9:0] bits.

When this bit is set to 1, the slave address set in the SVAD[9:0] bits is enabled and is compared with the received slave address.

When this bit is set to 0, the slave address set in the SVAD[9:0] bits is disabled and is ignored even if it matches the received slave address.

40.2.14 REFCKCTL : Reference Clock Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x070

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2		0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	IREFCKS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	IREFCKS[2:0]*1	Internal Reference Clock Selection Selects the internal reference clock source (I3Cφ) for I3C. 0 0 0: TCLK/1 clock 0 0 1: TCLK/2 clock 0 1 0: TCLK/4 clock 0 1 1: TCLK/8 clock 1 0 0: TCLK/16 clock 1 0 1: TCLK/32 clock 1 1 0: TCLK/64 clock 1 1 1: TCLK/128 clock	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. Set the IREFCKS[2:0] bit to 000 in I3C mode.

40.2.15 STDBR : Standard Bit Rate Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x074

Bit position:	31	30	29					24	23	22	21					16
Bit field:	DSBRPO	—	SBRHP[5:0]						—	—	SBRLP[5:0]					
Value after reset:	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1

Bit position:	15							8	7							0
Bit field:	SBRHO[7:0]								SBRLO[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	SBRLO[7:0]	Standard Bit Rate Low-Level Period Open-Drain Count value of the low-level period of SCL clock*1	R/W
15:8	SBRHO[7:0]	Standard Bit Rate High-Level Period Open-Drain Count value of the high-level period of SCL clock*1	R/W

Bit	Symbol	Function	R/W
21:16	SBRLP[5:0]	Standard Bit Rate Low-level Period Push-Pull*2 Count value of the low-level period of SCL clock	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
29:24	SBRHP[5:0]	Standard Bit Rate High-Level Period Push-Pull*3 Count value of the high-level period of SCL clock	R/W
30	—	This bit is read as 0. The write value should be 0.	R/W
31	DSBRPO	Double the Standard Bit Rate Period for Open-Drain*4 0: The time period set for SBRHO[7:0] and SBRLO[7:0] is not doubled. 1: The time period set for SBRHO[7:0] and SBRLO[7:0] is doubled.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. These bits support I^{2}C, I3C master, and I3C secondary master mode.
Note 2. These bits support I3C master mode and I3C secondary master mode.
Note 3. These bits support all I3C mode.
Note 4. This bit supports I^{2}C, I3C master, and I3C secondary master mode.

The STDBR register sets the bit rate according to the operating speed.

- I^{2}C mode: Bit rate setting when communicating with Standard-mode / Fast-mode / Fast-mode plus
- I3C master mode: Bit rate setting selected by MODE bit of command descriptor
- I3C slave mode: I3C bit rate setting

The I^{2}C transfer rate and the SCL clock duty are calculated using the following expression.

Transfer rate = 1 / {[(High-Level Period + α*1) + (Low-Level Period + α)] / I3Cφ*2 + I3C_SCL line rising time [tr]*3 + I3C_SCL line falling time [tf]*3}

Duty cycle = {I3C_SCL line rising time [tr] + (High-Level Period + α) / I3Cφ} / {I3C_SCL line falling time [tf] + (Low-Level Period + α) / I3Cφ}

Note 1. α depend on the number of stages in the noise filter.
Note 2. I3Cφ = TCLK × Division ratio
Note 3. The I3C_SCL line rising time [tr] and I3C_SCL line falling time [tf] depend on the total bus line capacitance [Cb] and the pull-up resistor [Rp]. For details, see the I^{2}C-bus specification from NXP Semiconductors.

The I3C transfer rate and the SCL clock duty are calculated using the following expression.

Transfer rate = 1 / [(High-Level Period + Low-Level Period) / I3Cφ + I3C_SCL line rising time [tr] + I3C_SCL line falling time [tf]]

Duty cycle = [I3C_SCL line rising time [tr] + High-Level Period / I3Cφ] / [I3C_SCL line falling time [tf] + Low-Level Period / I3Cφ]

SBRLO[7:0] bits (Standard Bit Rate Low-Level Period Open-Drain)

The SBRLO[7:0] bits are used to set the low-level period of SCL clock in Open-Drain mode.

I3C counts the low-level period with the internal reference clock source (I3Cφ) specified by the REFCKCTL.IREFCKS[2:0] bits. It also works to generate the data setup time for automatic SCL low-hold operation (see section 40.3.2.3.6. Clock Stretching [I^{2}C Mode]); when I3C is used in I^{2}C slave mode, these bits need to be set to a value longer than the data setup time*1.

If the digital noise filter is enabled (INCTL.DNFE = 1), set the SBRLO[7:0] bits to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the INCTL.DNFS[3:0] bits.

Note 1. Data setup time (tSU: DAT)
250 ns (up to 100 kbps: Standard-mode [Sm])
100 ns (up to 400 kbps: Fast-mode [Fm])
50 ns (up to 1 Mbps: Fast-mode plus [Fm+])
10 ns (up to 3.4 Mbps: Hs-mode [HS])

SBRHO[7:0] bits (Standard Bit Rate High-Level Period Open-Drain)

The SBRHO[7:0] bits are used to set the high-level period of SCL clock in Open-Drain mode. SBRHO[7:0] bits are valid in master mode. If I3C is used only in I^2C slave mode, these bits need not set the high-level period.

I3C counts the high-level period with the internal reference clock source (I3Cφ) specified by the REFCKCTL.IREFCKS[2:0] bits.

If the digital noise filter is enabled (the INCTL.DNFE bit = 1), set the SBRHO[7:0] bits to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the INCTL.DNFS[3:0] bits.

SBRLP[5:0] bits (Standard Bit Rate Low-level Period Push-Pull)

SBRLP[5:0] bits are used to set the low-level period of SCL clock in Push-Pull.

I3C counts the low-level period with the internal reference clock source (I3Cφ) specified by the REFCKCTL.IREFCKS[2:0] bits.

If the digital noise filter is enabled (the INCTL.DNFE bit = 1), set the SBRLP[5:0] bits to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the INCTL.DNFS[3:0] bits.

SBRHP[5:0] bits (Standard Bit Rate High-Level Period Push-Pull)

SBRHP[5:0] bits is used to set the high-level period of SCL clock in Push-Pull mode.

SBRHP[5:0] bits are valid in master mode. If I3C is used only in I^2C slave mode, these bits need not set the high-level period.

I3C counts the high-level period with the internal reference clock source (I3Cφ) specified by the REFCKCTL.IREFCKS[2:0] bits.

If the digital noise filter is enabled (the INCTL.DNFE bit = 1), set the SBRHP[5:0] bits to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the INCTL.DNFS[3:0] bits.

DSBRPO bit (Double the Standard Bit Rate Period for Open-Drain)

When DSBRPO = 1, double the high-level period that is set in SBRHO[7:0] and double the low-level period that is set in SBRLO[7:0].

Table 40.4 Requirement and usage of setting in each mode

Bit name	Device mode				
	I^2C master	I^2C slave	I3C Master	I3C Secondary Master	I3C Slave
SBRHP[5:0]	Do not use	Do not use	Setting required*3	Setting required*4	Setting required*6
SBRLP[5:0]	Do not use	Do not use	Setting required*3	Setting required*5	Do not use
SBRHO[7:0]	Setting required*1	Do not use	Setting required*3	Setting required*5	Do not use
SBRLO[7:0]	Setting required*1	Setting required*2	Setting required*3	Setting required*5	Do not use

Note 1. The setting value is used for the data rate of ST, FM, and FM+ mode.
Note 2. The setting value is used for the data setup time of automatic SCL low-hold operation.
Note 3. The setting value is used for the data rate of each communication.
Note 4. When operating with I3C Master, the setting value is used for the data rate of each communication.
When operating with I3C Slave, the setting value is used for the data rate of HDR-TSP/TSL mode.
Note 5. When operating with I3C Master, the setting value is used for the data rate of each communication.
When operating with I3C Slave, do not use.
Note 6. The setting value is used for the data rate of HDR-TSP/TSL mode.

40.2.16 EXTBR : Extended Bit Rate Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x078

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	EBRHP[5:0]						—	—	EBRLP[5:0]						EBRHO[7:0]								EBRLO[7:0]							
Value after reset:	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	EBRLO[7:0]	Extended Bit Rate Low-Level Period Open-Drain[*1] Count value of the low-level period of SCL clock	R/W
15:8	EBRHO[7:0]	Extended Bit Rate High-Level Period Open-Drain[*1] Count value of the high-level period of SCL clock	R/W
21:16	EBRLP[5:0]	Extended Bit Rate Low-Level Period Push-Pull[*2] Count value of the low-level period of SCL clock	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
29:24	EBRHP[5:0]	Extended Bit Rate High-Level Period Push-Pull[*2] Count value of the high-level period of SCL clock	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. These bits support I^{2}C, I3C master, and I3C secondary master mode.
Note 2. These bits support I3C master mode and I3C secondary master mode.

The EXTBR register sets the bit rate according to the operating speed.

- I^{2}C mode: Bit rate setting for communicating in high-speed mode
- I3C master mode: Bit rate setting selected by MODE bit of command descriptor
- I3C slave mode: unused

EBRLO[7:0] bits (Extended Bit Rate Low-Level Period Open-Drain)

See SBRLO[7:0] bits of section 40.2.15. STDBR : Standard Bit Rate Register for details. Watch SBRHO, SBRLO as EBRHO[7:0], EBRLO[7:0].

EBRHO[7:0] bits (Extended Bit Rate High-Level Period Open-Drain)

See SBRHO[7:0] bits of section 40.2.15. STDBR : Standard Bit Rate Register for details. Watch SBRHO, SBRLO as EBRHO[7:0], EBRLO[7:0].

EBRLP[5:0] bits (Extended Bit Rate Low-Level Period Push-Pull)

See SBRLP[5:0] bits of section 40.2.15. STDBR : Standard Bit Rate Register for details. Watch SBRHP, SBRLP as EBRHP[5:0], EBRLP[5:0].

EBRHP[5:0] bits (Extended Bit Rate High-Level Period Push-Pull)

See SBRHP[5:0] bits of section 40.2.15. STDBR : Standard Bit Rate Register for details. Watch SBRHP, SBRLP as EBRHP[5:0], EBRLP[5:0].

Table 40.5 Requirement and usage of setting in each mode (1 of 2)

Bit name	Device mode				
	I^{2}C master	I^{2}C slave	I3C Master	I3C Secondary Master	I3C Slave
EBRHP[5:0]	do not use	do not use	Setting required[*3]	Setting required[*4]	do not use
EBRLP[5:0]	do not use	do not use	Setting required[*3]	Setting required[*4]	do not use

Table 40.5 Requirement and usage of setting in each mode (2 of 2)

Bit name	Device mode				
	I2C master	I2C slave	I3C Master	I3C Secondary Master	I3C Slave
EBRHO[7:0]	Setting required*1	do not use	Setting required*3	Setting required*4	do not use
EBRLO[7:0]	Setting required*1	Setting required*2	Setting required*3	Setting required*4	do not use

Note 1. The setting value is used for the data rate of High-Speed mode.
Note 2. The setting value is used for the data setup time of automatic SCL low-hold operation in Hs-mode.
Note 3. The setting value is used for the data rate of each communication.
Note 4. When operating with I3C Master, the setting value is used for the data rate of each communication.
When operating with I3C Slave, do not use.

40.2.17 BFRECDT : Bus Free Condition Detection Time Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x07C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FRECYC[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	FRECYC[8:0]	Bus Free Condition Detection Cycle The count value is a period for detecting the Bus free condition.	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

FRECYC[8:0] bits (Bus Free Condition Detection Cycle)

I3C counts the period for detecting the Bus free condition with the I3Cφ.

These bits set the Bus Free period. This Bus Free period is counted by the internal reference clock (I3Cφ) selected by the REFCKCTL.IREFCKS[2:0] bits. See the BCST.BFREF flag for Bus Free detection behavior.

40.2.18 BAVLCDT : Bus Available Condition Detection Time Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x080

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AVLCYC[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	AVLCYC[8:0]	Bus Available Condition Detection Cycle The count value is a period for detecting the Bus available condition.	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports all I3C mode.

AVLCYC[8:0] bits (Bus Available Condition Detection Cycle)

I3C counts the period for detecting the Bus available condition with the I3Cφ.

These bits set the Bus Available period. This Bus Available period is counted by the internal reference clock (I3Cφ) selected by the REFCKCTL.IREFCKS[2:0] bits. See the BCST.BAVLF flag for Bus Available detection behavior.

40.2.19 BIDLCDT : Bus Idle Condition Detection Time Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x084

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IDLCYC[17:0]																	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
17:0	IDLCYC[17:0]	Bus Idle Condition Detection Cycle The count value is a period for detecting the Bus idle condition.	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports all I3C mode.

IDLCYC[17:0] bits (Bus Idle Condition Detection Cycle)

I3C counts the period for detecting the Bus idle condition with the I3Cφ.

These bits set the Bus Idle period. This Bus Idle period is counted by the internal reference clock (I3Cφ) selected by the REFCKCTL.IREFCKS[2:0] bits. See the BCST.BIDLF flag for Bus Available detection behavior.

40.2.20 OUTCTL : Output Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x088

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SDODCS	—	—	—	—	SDOD[2:0]			—	—	—	EXCYC	—	SOCWP	SCOC	SDOC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	SDOC	SDA Output Control*1 0: I3C drives the I3C_SDA pin low. 1: I3C releases the I3C_SDA pin.	R/W
1	SCOC	SCL Output Control*1 High level output is achieved through an external pull-up resistor. 0: I3C drives the I3C_SCL pin low. 1: I3C releases the I3C_SCL pin.	R/W
2	SOCWP	SCL/SDA Output Control Write Protect*1 0: Bits SCOC and SDOC are protected. 1: Bits SCOC and SDOC can be written (When writing simultaneously with the value of the target bit). This bit is read as 0.	W
3	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
4	EXCYC	Extra SCL Clock Cycle Output*3 The EXCYC bit is cleared automatically after one clock cycle is output. 0: Does not output an extra SCL clock cycle (default). 1: Outputs an extra SCL clock cycle.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
10:8	SDOD[2:0]	SDA Output Delay*2 0 0 0: No output delay 0 0 1: 1 I3Cφ cycle (When OUTCTL.SDODCS = 0 (I3Cφ)) 1 or 2 I3Cφ cycles (When OUTCTL.SDODCS = 1 (I3Cφ/2)) 0 1 0: 2 I3Cφ cycles (When OUTCTL.SDODCS = 0 (I3Cφ)) 3 or 4 I3Cφ cycles (When OUTCTL.SDODCS = 1 (I3Cφ/2)) 0 1 1: 3 I3Cφ cycles (When OUTCTL.SDODCS = 0 (I3Cφ)) 5 or 6 I3Cφ cycles (When OUTCTL.SDODCS = 1 (I3Cφ/2)) 1 0 0: 4 I3Cφ cycles (When OUTCTL.SDODCS = 0 (I3Cφ)) 7 or 8 I3Cφ cycles (When OUTCTL.SDODCS = 1 (I3Cφ/2)) 1 0 1: 5 I3Cφ cycles (When OUTCTL.SDODCS = 0 (I3Cφ)) 9 or 10 I3Cφ cycles (When OUTCTL.SDODCS = 1 (I3Cφ/2)) 1 1 0: 6 I3Cφ cycles (When OUTCTL.SDODCS = 0 (I3Cφ)) 11 or 12 I3Cφ cycles (When OUTCTL.SDODCS = 1 (I3Cφ/2)) 1 1 1: 7 I3Cφ cycles (When OUTCTL.SDODCS = 0 (I3Cφ)) 13 or 14 I3Cφ cycles (When OUTCTL.SDODCS = 1 (I3Cφ/2))	R/W
14:11	—	These bits are read as 0. The write value should be 0.	R/W
15	SDODCS	SDA Output Delay Clock Source Selection*3 0: The internal reference clock (I3Cφ) is selected as the clock source of the SDA output delay counter. 1: The internal reference clock divided by 2 (I3Cφ/2) is selected as the clock source of the SDA output delay counter.*4	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

Note 1. This bit supports I^{2}C, I3C master, and I3C secondary master mode.

Note 2. These bits support I^{2}C mode.

Note 3. This bit supports I^{2}C mode.

Note 4. The setting SDODCS = 1 (I3Cφ/2) only becomes valid when SCL is at the low level. When SCL is at the high level, the setting SDODCS = 1 becomes invalid and the clock source becomes the internal reference clock (I3Cφ).

SDOC bit (SDA Output Control) and SCOC bit (SCL Output Control)

These bits are used to directly control the I3C_SDA and I3C_SCL signals output from this I3C.

When writing to these bits, also write 1 to the SOCWP bit at the same time.

The result of setting these bits is input to I3C via the input buffer. When slave mode is selected, a START condition may be detected and the bus may be released depending on the bit settings.

Do not rewrite these bits during a START condition, STOP condition, Repeated START condition, or during transmission or reception. Operation after rewriting under the above conditions is not guaranteed.

EXCYC bit (Extra SCL Clock Cycle Output)

This bit is used to output an extra SCL clock cycle for debugging or error processing.

Normally, set the bit to 0. Setting the bit to 1 in a normal communication state causes a communication error.

For details on this function, see section 40.3.2.3.11. Port Control , (1) Extra SCL Clock Cycle Output Function.

40.2.21 INCTL : Input Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x08C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	DNFE	DNFS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0

Bit	Symbol	Function	R/W
3:0	DNFS[3:0]	Digital Noise Filter Stage Selection 0x0: Noise of up to one I3Cφ cycle is filtered out (single-stage filter). 0x1: Noise of up to two I3Cφ cycles is filtered out (2-stage filter). 0x2: Noise of up to three I3Cφ cycles is filtered out (3-stage filter). 0x3: Noise of up to four I3Cφ cycles is filtered out (4-stage filter). 0x4: Noise of up to five I3Cφ cycles is filtered out (5-stage filter). ⋮ 0xF: Noise of up to sixteen I3Cφ cycles is filtered out (16-stage filter).	R/W
4	DNFE	Digital Noise Filter Circuit Enable 0: No digital noise filter circuit is used. 1: A digital noise filter circuit is used.	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
7:6	—	These bits are read as 1. The write value should be 1.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I^{2}C mode.

DNFS[3:0] bits (Digital Noise Filter Stage Selection)

These bits are used to select the number of stages in the digital noise filter.

For details on the digital noise filter function, see section 40.3.2.6.3. Digital Noise-Filter Circuits [I^{2}C Mode].

In I^{2}C High Speed mode, I3C changes the number of noise filter stage to a quarter of the number of noise filter stage automatically.

Note:
- Set the noise range to be filtered out by the noise filter within a range less than the I3C_SCL line high-level period or low-level period. If the noise range is set to a value of (SCL clock width: high-level period or low-level period, whichever is shorter) - [1.5 internal reference clock (I3Cφ) cycles] or more, the SCL clock is regarded as noise by the noise filter function of I3C, which may prevent I3C from operating normally.
- In I^{2}C High Speed mode, the lower 2 bits of the DNFS [3:0] bits are ignored, and the number of filter stages for 1 to 4 stages is selected by the upper 2 bits.

40.2.22 TMOCTL : Timeout Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x090

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	TOMDS[1:0]		TOHCTL	TOLCTL	—	—	TODTS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	Symbol	Function	R/W
1:0	TODTS[1:0]	Timeout Detection Time Selection 0 0: 16bit-timeout 0 1: 14bit-timeout 1 0: 8bit-timeout 1 1: 6bit-timeout	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	TOLCTL	Timeout L Count Control 0: Count is disabled while the I3C_SCL line is at a low level. 1: Count is enabled while the I3C_SCL line is at a low level.	R/W
5	TOHCTL	Timeout H Count Control 0: Count is disabled while the I3C_SCL line is at a high level. 1: Count is enabled while the I3C_SCL line is at a high level.	R/W
7:6	TOMDS[1:0]	Timeout Operation Mode Selection 0 0: Timeout is detected during the following conditions: ● The bus is busy (BCST.BFREF = 0) in master mode. ● I3C's own slave address is detected and the bus is busy in slave mode. ● The bus is free (BCST.BFREF = 1) while generation of a START condition is requested (CNDCTL.STCND = 1). 0 1: Timeout is detected while the bus is busy. 1 0: Timeout is detected while the bus is free. 1 1: Setting prohibited	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

TODTS[1:0] bits (Timeout Detection Time Selection)

These bits are used to select for the timeout detection time when the timeout function is enabled (BSTE.TODE bit = 1).

When these bits are set to 00b, the timeout detection internal counter functions as a 16-bit counter.

When these bits are set to 01b, the counter functions as a 14-bit counter.

When these bits are set to 10b, the counter functions as an 8-bit counter.

When these bits are set to 11b, the counter functions as a 6-bit counter.

While the I3C_SCL line is in the state that enables this counter as specified by bits TOHCTL and TOLCTL, the counter counts up in synchronization with the internal reference clock (I3Cφ) as a count source.

For details on the timeout function, see section 40.3.2.4.5. Timeout Error Detection .

TOLCTL bit (Timeout L Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the I3C_SCL line is held low when the timeout function is enabled (BSTE.TODE = 1).

TOHCTL bit (Timeout H Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the I3C_SCL line is held high when the timeout function is enabled (BSTE.TODE = 1).

TOMDS[1:0] bits (Timeout Operation Mode Selection)

These bits are used to select the detection condition for timeout when the timeout function is enabled.

Note: When working with I^2C Slave, during 10-bit address communication, the timeout count starts when the upper address match is detected.

40.2.23 WUCTL : Wake Up Unit Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x098

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	WUFE	WUFSYNE	—	WUANFS	—	—	—	WUACKS
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	WUACKS	Wake-Up Acknowledge Selection*1 Choice of four response mode with a combination of RSTCTL.INTLRST bit and WUACKS bit. Shown in Table 40.6.	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	WUANFS	Wake-Up Analog Noise Filter Selection*1 0: Do not add the Wake-up analog filter. 1: Add the Wake-up analog filter.	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	WUFSYNE	Wake-Up function PCLK Synchronous Enable 0: I3C asynchronous circuit enable 1: I3C synchronous circuit enable	R/W
7	WUFE	Wake-up function Enable Do not set WUFE = 0 during Wake-up operation. 0: Wake-up function disables 1: Wake-up function enables	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports I^2C mode.

Table 40.6 Wake-Up Mode

INTLRST	WUACKS	Operation mode	Description
0	0	Normal Wake-Up mode 1	ACK response at 9th SCL and SCL low hold after at 9th SCL.
0	1	Normal Wake-Up mode 2	No ACK response immediately and SCL low hold between 8th and 9th SCL. Release SCL low hold and ACK response at 9th SCL.
1	0	Command recovery mode	ACK response at 9th SCL and not SCL low hold.
1	1	EEP response mode	NACK response at 9th SCL and not SCL low hold.

Note: In Wake-Up mode 2, HS mode cannot be used.

WUFSYNE bit (Wake-Up function PCLK Synchronous Enable)

This bit is used to switch between the PCLK synchronous operation and the PCLK asynchronous operation.

The bit is used in combination with the WUASYNF flag at Wake-Up effective function (WUCTL.WUFE bit = 1).

[When switching from the PCLK synchronous operation to the PCLK asynchronous operation]

I3C operation changes into the PCLK asynchronous operation during BCST.BFREF flag = 1, when the WUASYNF flag set to 1 during WUFSYNE = 0.

The reception can operate without depending on the state of operation of PCLK (With PCLK stopped) after it switches to the PCLK asynchronous operation (Wake-Up event detection operation).

[When switching from the PCLK asynchronous operation to the PCLK synchronous operation]

I3C operation changes into the PCLK synchronous operation at the following conditions. (At the same timing when WUFSYNE flag becomes 0)

In the case Wake-Up event detects : right after WUFSYNE bit is set to 1.

In the case Wake-Up event does not detect : when STOP condition is detected after WUFSYNE bit is set to 1.

[Setting condition]

- When 1 is written to the WUFSYNE bit.
- WUCTL.WUFE = 0

[Clearing conditions]

- When 0 is written to the WUFSYNE bit.

40.2.24 ACKCTL : Acknowledge Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x0A0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	ACKTWP	ACKT	ACKR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ACKR	Acknowledge Reception 0: A 0 is received as the acknowledge bit (ACK reception). 1: A 1 is received as the acknowledge bit (NACK reception).	R
1	ACKT	Acknowledge Transmission 0: A 0 is sent as the acknowledge bit (ACK transmission). 1: A 1 is sent as the acknowledge bit (NACK transmission).	R/W
2	ACKTWP	ACKT Write Protect 0: The ACKT bit is protected. 1: The ACKT bit can be written (when writing simultaneously with the value of the target bit). This bit is read as 0.	W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I^2C mode.

ACKR bit (Acknowledge Reception)

This bit is used to store the acknowledge bit information received from the receive device in transmit mode.

[Setting condition]

- When 1 is received as the acknowledge bit with the PRSST.TRMD bit set to 1.

[Clearing condition]

- When 0 is received as the acknowledge bit with the PRSST.TRMD bit set to 1.

ACKT bit (Acknowledge Transmission)

[Setting condition]

- When 1 is written to the ACKT bit and 1 is written to the ACKTWP bit at the same time.

[Clearing conditions]

- When 0 is written to the ACKT bit and 1 is written to the ACKTWP bit at the same time.
- When a STOP condition is detected. (when a STOP condition is detected with the CNDCTL.SPCND bit set to 1.)

Note: Set the ACKT bit to 0 in I^2C Slave mode.

ACKTWP bit (ACKT Write Protect)

This bit is used to control the modification of the ACKT bit.

When changing the ACKT bit, setting this bit to 1 at the same time can change the ACKT bit.

When this bit is read, 0 is always read.

40.2.25 SCSTRCTL : SCL Stretch Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x0A4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RWE	ACKTWE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ACKTWE	Acknowledge Transmission Wait Enable 0: NTST.RDBFF0 is set at the rising edge of the ninth SCL clock cycle. (The I3C_SCL line is not held low at the falling edge of the eighth clock cycle.) 1: NTST.RDBFF0 is set at the rising edge of the eighth SCL clock cycle. (The I3C_SCL line is held low at the falling edge of the eighth clock cycle.) Low-hold is released by writing a value to the ACKCTL.ACKT bit.	R/W
1	RWE	Receive Wait Enable 0: No WAIT (The period between ninth clock cycle and first clock cycle is not held low.) 1: WAIT (The period between ninth clock cycle and first clock cycle is held low.) Low-hold is released by reading NTDTBP0.	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I^2C mode.

ACKTWE bit (Acknowledgement Transmission Wait Enable)

This bit is used to select the NTST.RDBFF0 flag set timing in receive mode and also to select whether to hold the I3C_SCL line low at the falling edge of the eighth SCL clock cycle.

When ACKTWE = 0, the I3C_SCL line is not held low at the falling edge of the eighth SCL clock cycle, and the NTST.RDBFF0 flag is set to 1 at the rising edge of the ninth SCL clock cycle.

When ACKTWE = 1, the NTST.RDBFF0 flag is set to 1 at the rising edge of the eighth SCL clock cycle and the I3C_SCL line is held low at the falling edge of the eighth SCL clock cycle. The low-hold of the I3C_SCL line is released by writing a value to the ACKCTL.ACKT bit.

After data is received with this setting, the I3C_SCL line is automatically held low before the acknowledge bit is sent. This enables processing to send ACK (ACKCTL.ACKT = 0) or NACK (ACKCTL.ACKT = 1) according to receive data.

RWE bit (Receive Wait Enable)

This bit is used to control whether to hold the period between the ninth SCL clock cycle and the first SCL clock cycle low until the receive data buffer (NTDTBP0) is completely read each time single-byte data is received in receive mode.

When RWE = 0, the receive operation is continued without holding the period between the ninth and the first SCL clock cycle low. When both the ACKTWE and RWE bits = 0, continuous receive operation is enabled with the double buffer.

When RWE = 1, the I3C_SCL line is held low from the falling edge of the ninth clock cycle until the NTDTBP0 value is read each time single-byte data is received.

This enables receive operation in byte units.

Note: When the value of the RWE bit is to be read, be sure to read the NTDTBP0 beforehand.

40.2.26 SCSTLCTL : SCL Stalling Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x0B0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ACKPE	PARPE	—	AAPE	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15															0
Bit field:	STLCYC[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	STLCYC[15:0]	Stalling Cycle Counter setting of stall period (I3Cφ cycle). Common use for each phase.	R/W
27:16	—	These bits are read as 0. The write value should be 0.	R/W
28	AAPE	Assigned Address Phase Enable Enable bit that allows stall by the first bit at address assignment 0: Does not stall the SCL clock during the address assignment phase. 1: Stall the SCL clock during address assignment phase.	R/W
29	—	This bit is read as 0. The write value should be 0.	R/W
30	PARPE	Parity Phase Enable Stall enable bit in parity bit period 0: Does not stall the SCL clock during the parity bit period. 1: Stall the SCL clock during the parity bit period.	R/W
31	ACKPE	ACK phase Enable Stall enable bit during ACK/NACK phase 0: Does not stall the SCL clock during the ACK/NACK phase. 1: Stall the SCL clock during the ACK/NACK phase.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

When setting this register, follow Chapter 5.1.2.5 Master Clock Stalling of MIPI I3C Spec V1.0, and use it only when necessary because of its negative impacts on bus performance.

STLCYC[15:0] bits (Stalling Cycle)

These bits set the SCL stall period. The SCL stall period is counted by the internal reference clock (I3Cφ). This is a counter common to the enable bits of each phase.

AAPE bit (Assigned Address Phase Enable)

The master can stall SCL during the low period of the first bit of the assigned address phase of the Enter Dynamic Address Assignment CCC command. It can gain time in assigning dynamic address to the device based on the BCR and DCR of the slave. However, because the Dynamic Address Assignment procedure sends the dynamic address set in the DATBASm (m = 0 to 7) register in sequence, it is not necessary to set this bit and it is prohibited.

PARPE bit (Parity Phase Enable)

The parity bit of the transmission data of I3C write transfer can be used for SCL stalling to avoid underrun of the transmission data FIFO. However, when the transmission data FIFO of the I3C master becomes empty, SCL stalling is performed regardless of the setting of this bit, it is not necessary to set this bit and it is prohibited. It is necessary to set this bit when the I3C slave requires preparation time to receive data.

ACKPE bit (ACK phase Enable)

Determine the need to perform SCL stalling in the ACK/NACK phase based on the following criteria:

- It is necessary to set this bit when the I3C and I2C slaves connected to the bus require preparation time to receive or transmit data.
- In legacy I^2C communication, if there is a possibility that the data FIFO of the I3C master might underrun or overflow, it is not necessary to set this bit because SCL Stalling is performed by FIFO Empty or Full regardless of the setting of this bit.
- Other than legacy I^2C communication, the data FIFO of I3C master might underrun or overflow, and if SCL stalling is required in ACK phase, this bit can be set. However, it is necessary to build the software so that the FIFO does not underrun or overflow due to the interrupt generated according to the FIFO threshold setting (NQTHCTL, NTBTHCTL0, NRQTHCTL, HQTHCTL, HTBTHCTL).
- When I3C master responds ACK/NACK to IBI, it is not necessary to set this bit because ACK/NACK response can be set in advance by BCTL.HJACK, DATBASm.DVMRRJ and DATBASm.DVS IRRJ (m = 0 to 7).
- It is necessary to set this bit when the I3C slave connected to the bus requires preparation time to transmit data for Direct GET CCC.

40.2.27 SVTDLG0 : Slave Transfer Data Length Register 0

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x0C0

Bit position:	31															16	15															0
Bit field:	STDLG[15:0]																—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
31:16	STDLG[15:0]	Slave Transfer Data Length Indicates the number of bytes to be transferred.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

40.2.28 STCTL : Synchronous Timing Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x120

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STOE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STOE	Synchronous Timing output Enable 0: Disable 1: Enable	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports all I3C mode.

40.2.29 ATCTL : Asynchronous Timing Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x124

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CDIV[7:0]								—	—	—	—	—	AMEOE	MREFOE	ATTRGS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ATTRGS	Asynchronous Timing Trigger Select*1 0: Software trigger 1: Hardware trigger	R/W
1	MREFOE	MREF Output Enable (Capture Event / Counter Overflow)*2 0: Disable 1: Enable	R/W
2	AMEOE	Additional Master-initiated bus Event Output Enable*2 0: Disable 1: Enable	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
15:8	CDIV[7:0]	TCLK Counter Divide Setting*3	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports I3C secondary master mode and I3C slave mode.
Note 2. This bit supports I3C master mode and I3C secondary master mode.
Note 3. These bits support all I3C mode.

40.2.30 ATTRG : Asynchronous Timing Trigger Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x128

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ATSTRG
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ATSTRG	Asynchronous Timing Software Trigger 0: Do nothing 1: Software trigger (one-shot pulse) output This bit is always read as 0.	W
31:1	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

40.2.31 ATCCNTE : Asynchronous Timing Control Counter Enable Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x12C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ATCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ATCE	Asynchronous Timing Counter Enable for MREF, MC2, SC1, SC2. 0: Disable 1: Enable	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports all I3C mode.

40.2.32 CNDCTL : Condition Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x140

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SPCND	SRCND	STCND
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STCND	START (S) Condition Issuance 0: Does not request to issue a START condition. 1: Requests to issue a START condition.	R/W
1	SRCND	Repeated START (Sr) Condition Issuance 0: Does not request to issue a Repeated START condition. 1: Requests to issue a Repeated START condition.	R/W
2	SPCND	STOP (P) Condition Issuance 0: Does not request to issue a STOP condition. 1: Requests to issue a STOP condition.	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

Note: This register supports I^2C mode.

STCND bit (START (S) Condition Issuance)

This bit is used to request transition to master mode and issuance of a START condition.

For details on the START condition issuance, see section 40.3.2.3.3. START Condition / Repeated START Condition / STOP Condition Issuing Function .

[Setting condition]

- When 1 is written to the STCND bit

[Clearing conditions]

- When 0 is written to the STCND bit
- When a START condition has been issued (A START condition is detected)
- When the BST.ALF (arbitration-lost) flag is set to 1

Note: Set the STCND bit to 1 (START condition issuance request) when the BCST.BFREF flag is set to 1 (bus free state).

Note that arbitration may be lost due to a START condition issuance error if the STCND bit is set to 1 (START condition issuance request) when the BFREF flag is set to 0 (bus busy state).

SRCND bit (Repeated START (Sr) Condition Issuance)

This bit is used to request that a Repeated START condition be issued in master mode.

When this bit is set to 1 to request to issue a Repeated START condition, a Repeated START condition is issued when the BFREF flag is set to 0 (bus busy state) and the PRSST.CRMS bit is set to 1 (master mode).

For details on the Repeated START condition issuance, see section 40.3.2.3.3. START Condition / Repeated START Condition / STOP Condition Issuing Function .

[Setting condition]

- When 1 is written to the SRCND bit with the BCST.BFREF flag set to 0

[Clearing conditions]

- When 0 is written to the SRCND bit
- When a Repeated START condition has been issued (A Repeated START condition is detected)
- When the BST.ALF (arbitration-lost) flag is set to 1

Note: Do not set the SRCND bit to 1 while issuing a STOP condition.

Note: If 1 (requests to issue a Repeated START condition) is written to the SRCND bit in slave mode, the Repeated START condition is not issued but the SRCND bit remains set to 1.

If the operating mode changes to master mode with the bit not being cleared, note that the Repeated START condition may be issued.

SPCND bit (STOP (P) Condition Issuance)

This bit is used to request that a STOP condition be issued in master mode.

When this bit is set to 1 to request to issue a STOP condition, a STOP condition is issued when the BCST.BFREF flag is set to 0 (bus busy state) and the PRSST.CRMS bit is set to 1 (master mode).

For details on the STOP condition issuance, see section 40.3.2.3.3. START Condition / Repeated START Condition / STOP Condition Issuing Function .

[Setting condition]

- When 1 is written to the SPCND bit with the BCST.BFREF flag set to 0 and the PRSST.CRMS bit set to 1

[Clearing conditions]

- When 0 is written to the SPCND bit
- When a STOP condition has been issued (A STOP condition is detected)
- When the BST.ALF (arbitration-lost) flag is set to 1
- When a START condition and a Repeated START condition are detected

Note: Writing to the SPCND bit is not possible while the setting of the BCST.BFREF flag = 1 (bus free state).

Note: Do not set the SPCND bit to 1 while a Repeated START condition is being issued.

40.2.33 NCMDQP : Normal Command Queue Port Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x150

Bit position: 31 ... 0

Bit field: (blank)

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	Normal Command Queue Port	W

Note: S-TYPE3, P-TYPE3
Note: This register supports all I3C mode.

32-bit mailbox register NCMDQP contains a command descriptor structure that depends on the requested transfer type:

1. Address Assignment Command (see section 40.3.1.1.1. Address Assign Command)
2. Immediate Data Transfer (see section 40.3.1.1.2. Immediate Transfer Command)
3. Regular Data Transfer (see section 40.3.1.1.3. Regular Transfer Command)
4. Write + Write/Read Combo Transfer (see section 40.3.1.1.4. Combo Transfer Command)
5. Internal Control Command (see section 40.3.1.1.5. Internal Control Command)

Within the command descriptor, DWORDs appear starting with the Least Significant DWORD, in order until the Most Significant DWORD.

40.2.34 NRSPQP : Normal Response Queue Port Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x154

Bit position:	31 … 0
Bit field:	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	Normal Response Queue Port	R

Note: S-TYPE3, P-TYPE3
Note: This register supports all I3C mode.

32-bit mailbox register NRSPQP contains a response structure (see section 40.3.1.4. Receive Status Descriptor).

40.2.35 NTDTBP0/NTDTBP0_BY : Normal Transfer Data Buffer Port Register 0

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x158

Bit position:	31 … 0
Bit field:	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	Normal Transfer Data Buffer Port NTDTBP0 is a 32-bit read/write register. NTDTBP0_BY (NTDTBP0[7:0]) is an 8-bit read/write register.	R/W

Note: S-TYPE3, P-TYPE3

Note: NTDTBP0 is 32-bit access in I3C mode.
NTDTBP0_BY is 8-bit access in I^{2}C mode.

32-bit mailbox register NTDTBP0 is a 32-bit bidirectional data transfer register which is used both to read from the Normal Rx Data Buffer, and to write to the Normal Tx Data Buffer.

In other words, the Normal Rx Data Buffer and the Normal Tx Data Buffer have the same offset, forming a single bidirectional port for transmitting or receiving I3C data.

Read Operations:

[I3C protocol mode]

Data Read from the Normal Rx Data Buffer. It should be read based on Normal Queue Status Level indications. The Rx data is always aligned to a 4-byte boundary, and stored in the Normal Rx Data Buffer. If the length of the data transfer is not aligned to a 4-byte boundary, then there will be extra (unused) bytes at the end of the transferred data. The valid data must be identified using the DATA_LENGTH field in the Response Descriptor.

[I2C protocol mode]

When 1 byte of data has been received, the received data is transferred from the internal shift register to NTDTBP0 to enable the next data to be received. The double-buffer structure of the internal shift register and NTDTBP0 allows continuous receive operation if the received data has been read from NTDTBP0 while the internal shift register is receiving data. Read data from NTDTBP0 once when a Normal Rx Data buffer full interrupt (I3C_RX) request is generated. If NTDTBP0 receives the next Rx data before the current data is read from NTDTBP0 (while the RDBFF0 flag in NTST is 1), this module automatically holds the SCL clock low one cycle before the RDBFF0 flag is set to 1 next. The lower 8 bits of the read 32-bit data are valid as received data.

Write Operations:

[I3C protocol mode]

Data Written to the Normal Tx Data Buffer. Data DWORDs written to the Normal Tx Data Buffer are placed onto the I3C bus one byte at a time, with the DWORD LSB first. Within each byte, bits are placed onto the I3C bus in big-endian order, with bit 7 going out first on the bus. The Tx data should always start aligned to a 4-byte boundary, and written to the NTDTBP0 register. If the length of the transfer is not aligned to a 4-byte boundary, then there will be extra (unused) bytes at the end of the transferred data. I3C shall only send the valid number of bytes indicated in the DATA_LENGTH field of the Command Descriptor.

[I2C protocol mode]

When NTDTBP0 detects a space in the internal shift register, it transfers the Tx data that has been written to NTDTBP0 to the internal shift register and starts transmitting data in transmit mode. The double-buffer structure of NTDTBP0 and the internal shift register allows continuous transmit operation if the next Tx data has been written to NTDTBP0 while the internal shift register data is being transmitted. Write Tx data to NTDTBP0 once when a Normal Tx Data buffer empty interrupt (I3C_TX) request is generated. The lower 8 bits of the written 32-bit data are valid as transmission data.

40.2.36 NIBIQP : Normal IBI Queue Port Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x17C

Bit position: 31 … 0

Bit field: (blank)

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	Normal IBI Queue Port	R/W

Note: S-TYPE3, P-TYPE3

Note: This register supports all I3C mode.

When receiving an IBI, 32-bit mailbox register NIBIQP is used for both:

- Read the IBI status descriptor (see section 40.3.1.3. IBI Status Descriptor)
- Read the IBI data (which is raw/opaque data).

The IBI status descriptor is a read-only structure describing an IBI event received from a Slave device on the I3C bus.

Note: If the I3C HCI auto-read feature is used, then the IBI data includes the data received from the auto-generated private read operation.

Even if LAST_STATUS is set to 0, the driver software still evaluates the data payload length by examining the CHUNKS field.

40.2.37 NRSQP : Normal Receive Status Queue Port Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x180

Bit position:	31 … 0
Bit field:	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	Normal Receive Status Queue Port	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

32-bit mailbox register NRSQP contains a receive status structure (see section 40.3.1.4. Receive Status Descriptor).

40.2.38 HCMDQP : High Priority Command Queue Port Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x184

Bit position:	31 … 0
Bit field:	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	High Priority Command Queue Port	W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

32-bit mailbox register HCMDQP contains a command descriptor structure that depends on the requested transfer type:

1. Address Assignment Command (see section 40.3.1.1.1. Address Assign Command)
2. Immediate Data Transfer (see section 40.3.1.1.2. Immediate Transfer Command)
3. Regular Data Transfer (see section 40.3.1.1.3. Regular Transfer Command)
4. Write + Write/Read Combo Transfer (see section 40.3.1.1.4. Combo Transfer Command)
5. Internal Control Command (see section 40.3.1.1.5. Internal Control Command)

Within the command descriptor, DWORDs appear starting with the least significant DWORD, in order until the most significant DWORD.

40.2.39 HRSPQP : High Priority Response Queue Port Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x188

Bit position:	31 … 0
Bit field:	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	High Priority Response Queue Port	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

32-bit mailbox register HRSPQP contains a response structure. (see section 40.3.1.2. Response Descriptor)

40.2.40 HTDTBP : High Priority Transfer Data Buffer Port Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x18C

Bit position:	31–0
Bit field:	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	High Priority Transfer Data Buffer Port	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

The HTDTBP register is a 32-bit bidirectional data transfer register which is used both to read from the high priority Rx data, and to write to the high priority Tx data.

For Read Operation:

To receive data from the High Priority Rx Data Buffer, read from the HTDTBP register. It should be read based on queue status indication.

The Rx data is always aligned to a 4-byte boundary, and stored in the High Priority Rx Data Buffer.

If the length of the data transfer is not aligned to a 4-byte boundary, then there will be extra (unused) bytes at the end of the transferred data.

The valid data must be identified using the DATA_LENGTH field in the response descriptor.

For Write Operation:

To send data to the High Priority Tx Data Buffer, write to the HTDTBP register. Data DWORDs written to the Data port are placed onto the I3C bus one byte at a time, with the DWORD’s LSB first. Within each byte, bits are placed onto the I3C bus in big-endian order, with bit 7 going out first on the bus.

The High Priority Tx Data Buffer Port is mapped to the High Priority Tx Data Buffer.

The Tx data should always start aligned to a 4byte boundary, and written to the Tx data port register.

If the length of the transfer is not aligned to a 4-byte boundary, then there will be extra (unused) bytes at the end of the transferred data.

I3C shall only send the valid number of bytes indicated in the DATA_LENGTH field of the command descriptor.

40.2.41 NQTHCTL : Normal Queue Threshold Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x190

Bit position:	31–24	23–16	15–8	7–0
Bit field:	IBIQTH[7:0]	IBIDSSZ[7:0]	RSPQTH[7:0]	CMDQTH[7:0]
Value after reset:	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 1

Bit	Symbol	Function	R/W
7:0	CMDQTH[7:0]	Normal Command Queue Threshold[*1] 0x00: Interrupt is issued when Normal Command Queue is completely empty. Others: Interrupt is issued when Normal Command Queue contains N empties. (N = CMDQTH[7:0])	R/W
15:8	RSPQTH[7:0]	Normal Response Queue Threshold[*1] 0x00: Interrupt is issued when Normal Response Queue contains 1 entry (DWORD). Others: Interrupt is triggered when Normal Response Queue contains N+1 entries (DWORD). (N = CMDQTH[7:0])	R/W
23:16	IBIDSSZ[7:0]	Normal IBI Data Segment Size[*2] Supported Values: Minimum: 1 (4 bytes) Maximum: 63 (252 bytes), provided that the configured IBI Queue depth is 64 or more. When ATCCNTE.ATCE = 1, restrict to the number of slices ≥ 2.	R/W
31:24	IBIQTH[7:0]	Normal IBI Queue Threshold[*1] 0x00: I3C Protocol mode (Master): Interrupt is generated when the Outstanding IBI Status count is 1 or more. I3C Protocol mode (Slave): Interrupt is issued when IBI Data Buffer is completely empty. Others: I3C Protocol mode (Master): Interrupt is generated when the Outstanding IBI Status count is N + 1 or more. (N = CMDQTH[7:0]) I3C Protocol mode (Slave): Interrupt is issued when IBI Data Buffer contains N empties.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. These bits support all I3C mode.
Note 2. These bits support I3C master mode and I3C secondary master mode.

The Queue Threshold Control register controls the interrupt trigger thresholds for the Command Queue, the Response Queue, and the IBI Queue.

The specific reset values are indicative, and could be hardware implementation specific.

CMDQTH[7:0] bits (Normal Command Queue Threshold)

Controls the minimum number of Normal Command Queue empties needed to trigger the I3C_CMD interrupt.

If this field is greater than (Normal Command Queue size[*1] – 1), then only the number of bits required to address the full buffer depth will be considered.

RSPQTH[7:0] bits (Normal Response Queue Threshold)

Controls the minimum number of Normal Response Queue entries needed to trigger the I3C_RESP interrupt.

If this field is greater than (Normal Response Queue size[*2] – 1), then only the number of bits required to address the full buffer depth will be considered.

IBIDSSZ[7:0] bits (Normal IBI Data Segment Size)

This is the IBI data segment size, in DWORDs (4 bytes).

In PIO mode, this field allows the incoming Normal IBI data to be sliced into multiple segments generating status individually, to support cut-through readout of a long IBI payload data.

When Asynchronous Timing Control mode is supported, this field should be set to a value other than 1 or 3 to allow the single data segment to contain the entire Master time-stamp value (for example, both MREF and MC2).

IBIQTH[7:0] bits (Normal IBI Queue Threshold)

For I3C protocol mode (Master): PRTS.PRTMD = 0 and PRSST.CRMS = 1.

Controls generation of the I3C_IBI interrupt, based on the value of the Normal IBI Queue’s Outstanding IBI status count.

Each IBI status entry can represent either the complete IBI payload (if the IBI payload byte size is 4×IBIDSSZ or less), or a segment of the IBI payload (if the IBI payload byte size is more than 4×IBIDSSZ).

For I3C protocol mode (Slave) : PRTS.PRTMD bit = 0, PRSST.CRMS bit = 0.

Controls the minimum number of IBI Data Buffer empties needed to trigger the I3C_IBI interrupt.

If this field is greater than (IBI Data Buffer size*3 - 1), then only the number of bits required to address the full buffer depth will be considered.

Note 1. Normal Command Queue size is 4.
Note 2. Normal Response Queue size is 4.
Note 3. IBI Data Buffer size is 8.

Note: It is assumed that I3C has exactly one Command Queue, exactly one Response Queue, and exactly one IBI Queue.

40.2.42 NTBTHCTL0 : Normal Transfer Data Buffer Threshold Control Register 0

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x194

Bit position:	31	30	29	28	27	26		24	23	22	21	20	19	18		16
Bit field:	—	—	—	—	—	RXSTTH[2:0]			—	—	—	—	—	TXSTTH[2:0]		
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit position:	15	14	13	12	11	10		8	7	6	5	4	3	2		0
Bit field:	—	—	—	—	—	RXDBTH[2:0]			—	—	—	—	—	TXDBTH[2:0]		
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
2:0	TXDBTH[2:0]	Normal Tx Data Buffer Threshold*1 0 0 0: Interrupt triggers at 2 Tx Buffer empties, DWORDs 0 0 1: Interrupt triggers at 4 Tx Buffer empties, DWORDs 0 1 0: Interrupt triggers at 8 Tx Buffer empties, DWORDs 0 1 1: Interrupt triggers at 16 Tx Buffer empties, DWORDs Others: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
10:8	RXDBTH[2:0]	Normal Rx Data Buffer Threshold*1 0 0 0: Interrupt triggers at 2 Rx Buffer entries, DWORDs 0 0 1: Interrupt triggers at 4 Rx Buffer entries, DWORDs 0 1 0: Interrupt triggers at 8 Rx Buffer entries, DWORDs 0 1 1: Interrupt triggers at 16 Rx Buffer entries, DWORDs Others: Setting prohibited	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
18:16	TXSTTH[2:0]	Normal Tx Start Threshold*2 0 0 0: Wait for 2 entry DWORDs 0 0 1: Wait for 4 entry DWORDs 0 1 0: Wait for 8 entry DWORDs 0 1 1: Wait for 16 entry DWORDs Others: Setting prohibited	R/W
23:19	—	These bits are read as 0. The write value should be 0.	R/W
26:24	RXSTTH[2:0]	Normal Rx Start Threshold*2 0 0 0: Wait for 2 empty DWORDs 0 0 1: Wait for 4 empty DWORDs 0 1 0: Wait for 8 empty DWORDs 0 1 1: Wait for 16 empty DWORDs Others: Setting prohibited	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. These bits support all I3C mode.
Note 2. These bits support I3C master mode and I3C secondary master mode.

The Data Buffer Control register controls the interrupt trigger thresholds for the Rx Data Buffer Queue and the Tx Data Buffer Queue.

TXDBTH[2:0] bits (Normal Tx Data Buffer Threshold)

Minimum number of Tx Data Buffer empties, in DWORDs, that will trigger the I3C_TX interrupt.

The software must program a value less than Tx Data Buffer size in this register.

RXDBTH[2:0] bits (Normal Rx Data Buffer Threshold)

Minimum number of Rx Data Buffer entries in DWORDs that will trigger the I3C_RX interrupt.

The software must program a value less than Rx Data Buffer size in this register.

TXSTTH[2:0] bits (Normal Tx Start Threshold)

When preparing to initiate a Write Transfer on the I3C Bus, I3C shall wait until the Transmit Buffer has at least the indicated number of locations available.

Two optional configurable Modes are available:

1. Store and Forward Mode
   If the TXSTTH[2:0] field is set to the Transmit Buffer size, then I3C shall delay initiation of the Write Command as follows:
   - If the data length to be transferred is more than the Transmit Buffer size, then this module shall wait until the Tx Data Buffer is completely full.
   - If the data length to be transferred is less than the Transmit Buffer size, then I3C shall wait until enough Tx Data Buffer locations are available to store the data to be transferred.
2. Threshold Mode
   If the TXSTTH[2:0] field value is less than the Transmit Buffer size, then I3C shall initiate the Write Command as soon as the indicated number of Tx Data Buffer locations are entries.

RXSTTH[2:0] bits (Normal Rx Start Threshold)

When preparing to initiate a Read Transfer on the I3C bus, I3C shall wait until the Receive Buffer has at least the indicated number of empty locations in DWORDs.

Two optional configurable Modes are available:

1. Store and Forward Mode
   If the RXSTTH[2:0] field is set to the Receive Buffer size, then I3C shall delay initiation of the Read Command as follows:
   - If the data length to be transferred is more than the Receive Buffer size, then this module shall wait until the Rx Data Buffer is completely empty.
   - If the data length to be transferred is less than the Receive Buffer size, then I3C shall wait until enough Rx Data Buffer locations are available to store the data to be transferred.
2. Threshold Mode
   If the RXSTTH[2:0] field value is less than the Receive Buffer size, then I3C shall initiate the Read Command as soon as the indicated number of Rx Data Buffer locations are empty.

40.2.43 NRQTHCTL : Normal Receive Status Queue Threshold Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x1C0

Bit position:	31																							8	7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RSQTH[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
7:0	RSQTH[7:0]	Normal Receive Status Queue Threshold 0x00: Interrupt is issued when Normal Receive Status Queue contains 1 entry (DWORD). Others: Interrupt is triggered when Normal Receive Status Queue contains N+1 entries (DWORD). (N = RSQTH[7:0])	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports for I3C secondary master and I3C slave mode.

RSQTH[7:0] bits (Normal Receive Status Queue Threshold)

Controls the minimum number of normal receive status queue entries needed to trigger the I3C_RCV interrupt.

If this field is greater than (Normal Receive Status Queue size*1 – 1), then only the number of bits required to address the full buffer depth will be considered.

Note 1. Normal Receive Status Queue size is 2.

40.2.44 HQTHCTL : High Priority Queue Threshold Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x1C4

Bit position:	31																15 – 8	7 – 0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RSPQTH[7:0]	CMDQTH[7:0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 1

Bit	Symbol	Function	R/W
7:0	CMDQTH[7:0]	High Priority Command Queue Threshold 0x00: Interrupt is issued when High Priority Command Queue is completely empty. Others: Interrupt is issued when High Priority Command Queue contains N entries. (N = CMDQTH[7:0])	R/W
15:8	RSPQTH[7:0]	High Priority Response Queue Threshold 0x00: Interrupt is issued when High Priority Response Queue contains 1 entry (DWORD). Others: Interrupt is triggered when High Priority Response Queue contains N+1 entries (DWORD). (N = RSPQTH[7:0])	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

The High Priority Queue Threshold Control register controls the interrupt trigger thresholds for the High Priority Command Queue, the High Priority Response Queue, and the IBI Queue.

The specific reset values are indicative, and could be hardware implementation specific.

CMDQTH[7:0] bits (High Priority Command Queue Threshold)

Controls the minimum number of empty High Priority Command Queue entries needed to trigger the I3C_HCMD interrupt.

If this field is greater than (High Priority Command Queue size*1 – 1), then only the number of bits required to address the full buffer depth will be considered.

RSPQTH[7:0] bits (High Priority Response Queue Threshold)

Controls the minimum number of High Priority Response Queue entries needed to trigger the I3C_HRESP interrupt.

If this field is greater than (High Priority Response Queue size*2 – 1), then only the number of bits required to address the full buffer depth will be considered.

Note 1. High Priority Command Queue size is 2.
Note 2. High Priority Response Queue size is 2.

Note: It is assumed that I3C has exactly one High Priority Command Queue, exactly one High Priority Response Queue, and exactly one IBI Queue.

40.2.45 HTBTHCTL : High Priority Transfer Data Buffer Threshold Control Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x1C8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	RXSTTH[2:0]			—	—	—	—	—	TXSTTH[2:0]		
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	RXDBTH[2:0]			—	—	—	—	—	TXDBTH[2:0]		
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
2:0	TXDBTH[2:0]	High Priority Tx Data Buffer Threshold 0 0 0: Interrupt triggers at 2 High Priority Tx Buffer empties, DWORDs 0 0 1: Reserved Others Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
10:8	RXDBTH[2:0]	High Priority Rx Data Buffer Threshold 0 0 0: Interrupt triggers at 2 High Priority Rx Buffer entries, DWORDs 0 0 1: Reserved Others Setting prohibited	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
18:16	TXSTTH[2:0]	High Priority Tx Start Threshold 0 0 0: Wait for 2 entry DWORDs 0 0 1: Reserved Others Setting prohibited	R/W
23:19	—	These bits are read as 0. The write value should be 0.	R/W
26:24	RXSTTH[2:0]	High Priority Rx Start Threshold 0 0 0: Wait for 2 empty DWORDs 0 0 1: Reserved Others Setting prohibited	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

TXDBTH[2:0] bits (High Priority Tx Data Buffer Threshold)

Minimum number of High Priority Tx Data Buffer empties, in DWORDs, that will trigger the I3C_HTX interrupt. The software must program a value less than High Priority Tx Data Buffer size in this register.

RXDBTH[2:0] bits (High Priority Rx Data Buffer Threshold)

Minimum number of High Priority Rx Data Buffer entries in DWORDs that will trigger the I3C_HRX interrupt. The software must program a value less than High Priority Rx Data Buffer size in this register.

TXSTTH[2:0] bits (High Priority Tx Start Threshold)

When preparing to initiate a Write Transfer on the I3C bus, I3C shall wait until the High Priority Transmit Buffer has at least the indicated number of locations available.

Two optional configurable modes are available:

1. Store and Forward Mode

If the TXSTTH[2:0] field is set to the High Priority Transmit Buffer size, then I3C shall delay initiation of the write command as follows:

- If the data length to be transferred is more than the High Priority Transmit Buffer size, then I3C shall wait until the High Priority Tx Data Buffer is completely full.
- If the data length to be transferred is less than the High Priority Transmit Buffer size, then I3C shall wait until enough High Priority Tx Data Buffer locations are available to store the data to be transferred.

2. Threshold mode
   If the TXSTTH[2:0] field value is less than the High Priority Transmit Buffer size, then I3C shall initiate the write command as soon as the indicated number of High Priority Tx Data Buffer locations are empty.

RXSTTH[2:0] bits (High Priority Rx Start Threshold)

When preparing to initiate a Read Transfer on the I3C bus, I3C shall wait until the High Priority Receive Buffer has at least the indicated number of empty locations in DWORDs.

Two optional configurable modes are available:

1. Store and forward mode
   If the RXSTTH[2:0] field is set to the High Priority Receive Buffer size, then I3C shall delay initiation of the read command as follows:
   - If the data length to be transferred is more than the High Priority Receive Buffer size, then I3C shall wait until the High Priority Rx Data Buffer is completely empty.
   - If the data length to be transferred is less than the High Priority Receive Buffer size, then I3C shall wait until enough High Priority Rx Data Buffer locations are available to store the data to be transferred.
2. Threshold mode
   If the RXSTTH[2:0] field value is less than the High Priority Receive Buffer size, then I3C shall initiate the read command as soon as the indicated number of High Priority Rx Data Buffer locations are empty.

40.2.46 BST : Bus Status Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x1D0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	WUCNDDF	—	—	—	TODF	—	—	—	ALF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	TENDF	—	—	—	NACKDF	—	HDREXDF	SPCNDDF	STCNDDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STCNDDF	START Condition Detection Flag 0: START condition is not detected. 1: START condition is detected.	R/W*3
1	SPCNDDF	STOP Condition Detection Flag 0: STOP condition is not detected. 1: STOP condition is detected.	R/W*3
2	HDREXDF	HDR Exit Pattern Detection Flag*1 0: HDR Exit Pattern is not detected 1: HDR Exit Pattern is detected.	R/W*3
3	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
4	NACKDF	NACK Detection Flag*2 0: NACK is not detected. 1: NACK is detected.	R/W*3
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	TENDF	Transmit End Flag*2 0: Data is being transmitted. 1: Data has been transmitted.	R/W*3
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	ALF	Arbitration Lost Flag*2 0: Arbitration is not lost 1: Arbitration is lost.	R/W*3
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	TODF	Timeout Detection Flag 0: Timeout is not detected. 1: Timeout is detected.	R/W*3
23:21	—	These bits are read as 0. The write value should be 0.	R/W
24	WUCNDDF	Wake-Up Condition Detection Flag 0: Wake-Up is not detected. 1: Wake-Up is detected.	R/W*3
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports all I3C mode.
Note 2. This bit supports I^2C mode.
Note 3. Clearing (to 0) condition : Writing 0 after 1 is read.

STCNDDF bit (START Condition Detection Flag)

[Setting conditions]

- All of the followings are satisfied:
  1. The BSTE.STCNDDE bit = 1.
  2. When a START condition (or a Repeated START condition) is detected.

[Clearing conditions]

- When 0 is written to the STCNDDF flag after reading STCNDDF flag = 1.
- When a STOP condition is detected.

SPCNDDF bit (STOP Condition Detection Flag)

[Setting conditions]

- All of the followings are satisfied:
  1. The BSTE.SPCNDDE bit = 1.
  2. When a STOP condition is detected.

[Clearing condition]

- When 0 is written to the SPCNDDF flag after reading SPCNDDF flag = 1.

HDREXDF bit (HDR Exit Pattern Detection Flag)

[Setting conditions]

- All of the followings are satisfied:
  1. The BSTE.HDREXDE bit = 1.
  2. When an HDR EXIT pattern is detected.

[Clearing condition]

- When 0 is written to the HDREXDF flag after reading HDREXDF flag = 1.

NACKDF bit (NACK Detection Flag)

[Setting conditions]

- All of the followings are satisfied:
  1. The PRTS.PRTMD bit = 1 (I^2C protocol mode).
  2. The BSTE.NACKDE bit = 1 (Enables NACK detection interrupt status logging).
  3. When acknowledge is not received (NACK is received) from the receive device in transmit mode.

[Clearing condition]

- When 0 is written to the NACKDF flag after reading NACKDF flag = 1.

TENDF bit (Transmit End Flag)

[Setting conditions]

- All of the followings are satisfied:
  1. The PRTS.PRTMD bit = 1 (I^2C protocol mode).
  2. The BSTE.TENDE bit = 1 (Enables Transmit End Interrupt Status logging).
  3. At the rising edge of the ninth SCL clock cycle while the NTST.TDBEF0 flag = 1.
     Excluding when sending an address.

[Clearing conditions]

- When 0 is written to the TENDF flag after reading TENDF flag = 1.
- When data is written to the NTDTBP0 register.
- When a STOP condition is detected.

ALF bit (Arbitration Lost Flag)

[Setting conditions]

When master arbitration-lost detection is enabled: BSTE.ALE bit = 1, BFCTL.MALE = 1.

- When the internal SDA output state does not match the I3C_SDA line level at the rising edge of SCL clock except for the ACK period during data (including slave address) transmission in master transmit mode (when the I3C_SDA line is driven low while the internal SDA output is at a high level (the I3C_SDA pin is in the high-impedance state)).
- All of the followings are satisfied.
  1. When the START condition is detected while the CNDCTL.STCND bit = 1.
  2. When the internal SDA output state does not match the I3C_SDA line level.
- When the CNDCTL.STCND bit is set to 1 (START condition issuance request) while the BCST.BFREF flag = 0.

When NACK arbitration-lost detection is enabled: BSTE.ALE bit = 1, BFCTL.NALE = 1.

- When the internal SDA output state does not match the I3C_SDA line level at the rising edge of SCL clock in the ACK period during NACK transmission in receive mode.

When slave arbitration-lost detection is enabled: BSTE.ALE bit = 1, BFCTL.SALE = 1.

- When the internal SDA output state does not match the I3C_SDA line level at the rising edge of SCL clock except for the ACK period during data transmission in slave transmit mode.

[Clearing condition]

- When 0 is written to the ALF flag after reading ALF flag = 1.

TODF bit (Timeout Detection Flag)

[Setting conditions]

- All of the followings are satisfied.
  1. The BSTE.TODE bit = 1 (Enables Timeout Detection Interrupt Status logging).
  2. When the master mode or the received slave address matches the slave address n (n = 0 to 2) in Slave mode.
  3. When the I3C_SCL line state remains unchanged for the period specified by TMOCTL register.

[Clearing condition]

- When 0 is written to the TODF flag after reading TODF flag = 1.

WUCNDDF bit (Wake-Up Condition Detection Flag)

[Setting condition]

For I^2C protocol mode: PRTS.PRTMD bit = 1

- When PCLK and TCLK are supplied after all of the followings are satisfied.
  1. The WUCTL.WUFE bit = 1 (Wake-up function is enabled).
  2. The BSTE.WUCNDDE bit = 1 (Enables Wake-up Condition Detection Status logging).
  3. The WUST.WUASYNF flag = 1.
  4. When the address received in slave mode matches the address of slave enabled in the SVCTL.SVAE[2:0] bit (except for the Device-ID address).

For I3C Protocol mode (Master): PRTS.PRTMD bit = 0, PRSST.CRMS bit = 1.

- When PCLK and TCLK are supplied after all of the followings are satisfied.
  1. The WUCTL.WUFE bit = 1 (Wake-up function is enabled).
  2. The BSTE.WUCNDDE bit = 1 (Enables Wake-up Condition Detection Status logging).
  3. The WUST.WUASYNF flag = 1.
  4. When low level of the I3C_SDA line is detected (When the START condition is detected).

For I3C Protocol mode (Slave) : PRTS.PRTMD bit = 0, PRSST.CRMS bit = 0.

- When PCLK and TCLK are supplied after all of the followings are satisfied.
  1. The WUCTL.WUFE bit = 1 (Wake-up function is enabled).
  2. The BSTE.WUCNDDE bit = 1 (Enables Wake-up Condition Detection Status logging).
  3. The WUST.WUASYNF flag = 1.
  4. When the broadcast address (0x7E) is detected after a START (or Repeated START) condition and the own dynamic address is detected after the Repeated START condition following the broadcast address.

[Clearing condition]

- When 0 is written to the WUCNDDF flag after reading WUCNDDF flag = 1 while the WUST.WUASYNF flag = 0.

40.2.47 BSTE : Bus Status Enable Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x1D4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	WUCNDDE	—	—	—	TODE	—	—	—	ALE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	TENDE	—	—	—	NACKDE	—	HDREXDE	SPCNDDE	STCNDDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STCNDDE	START Condition Detection Enable 0: Disables START condition Detection Interrupt Status logging. 1: Enables START condition Detection Interrupt Status logging.	R/W
1	SPCNDDE	STOP Condition Detection Enable 0: Disables STOP condition Detection Interrupt Status logging. 1: Enables STOP condition Detection Interrupt Status logging.	R/W
2	HDREXDE	HDR Exit Pattern Detection Enable*1 0: Disables HDR Exit Pattern Detection Interrupt Status logging. 1: Enables HDR Exit Pattern Detection Interrupt Status logging.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	NACKDE	NACK Detection Enable*2 0: Disables NACK Detection Interrupt Status logging. 1: Enables NACK Detection Interrupt Status logging.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	TENDE	Transmit End Enable*2 0: Disables Transmit End Interrupt Status logging. 1: Enables Transmit End Interrupt Status logging.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	ALE	Arbitration Lost Enable*2 0: Disables Arbitration Lost Interrupt Status logging. 1: Enables Arbitration Lost Interrupt Status logging.	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	TODE	Timeout Detection Enable 0: Disables Timeout Detection Interrupt Status logging. 1: Enables Timeout Detection Interrupt Status logging.	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
24	WUCNDDE	Wake-up Condition Detection Enable 0: Disables Wake-up Condition Detection Status logging. 1: Enables Wake-up Condition Detection Status logging.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports all I3C mode.
Note 2. This bit supports I^{2}C mode.

STCNDDE bit (START Condition Detection Enable)

When this bit is 1, operation of BST.STCNDDF is enabled. For the setting conditions and clearing conditions of the BST.STCNDDF flag, see the details of BST.STCNDDF.

SPCNDDE bit (STOP Condition Detection Enable)

When this bit is 1, operation of BST.SPCNDDF is enabled. For the setting conditions and clearing conditions of the BST.SPCNDDF flag, see the details of BST.SPCNDDF.

HDREXDE bit (HDR Exit Pattern Detection Enable)

When this bit is 1, the operation of BST.HDREXDF is enabled. For the setting conditions and clearing conditions of the BST.HDREXDF flag, see the details of BST.HDREXDF.

NACKDE bit (NACK Detection Enable)

When this bit is 1, the operation of BST.NACKDF is enabled. This bit is used to specify whether to continue or discontinue the transfer operation when NACK is received from the slave device in transmit mode. Normally, set this bit to 1. For the setting conditions and clearing conditions of the BST.NACKDF flag, see the details of BST.NACKDF.

TENDE bit (Transmit End Enable)

When this bit is 1, the operation of BST.TENDF is enabled. For the setting conditions and clearing conditions of the BST.TENDF flag, see the details of BST.TENDF.

ALE bit (Arbitration Lost Enable)

When this bit is 1, the operation of BST.ALF is enabled. For the setting conditions and clearing conditions of the BST.ALF flag, see the details of BST.ALF.

TODE bit (Timeout Detection Enable)

When this bit is 1, the operation of BST.TODF is enabled. For the setting conditions and clearing conditions of the BST.TODF flag, see the details of BST.TODF.

WUCNDDE bit (Wake-up Condition Detection Enable)

When this bit is 1, the operation of BST.WUCNDDF is enabled. For the setting conditions and clearing conditions of the BST.WUCNDDF flag, see the details of BST.WUCNDDF.

40.2.48 BIE : Bus Interrupt Enable Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x1D8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	WUCNDDIE	—	—	—	TODIE	—	—	—	ALIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	TENDIE	—	—	—	NACKDIE	—	HDREXDIE	SPCNDDIE	STCNDDIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STCNDDIE	START Condition Detection Interrupt Enable 0: Disables START condition Detection Interrupt Signal. 1: Enables START condition Detection Interrupt Signal.	R/W
1	SPCNDDIE	STOP Condition Detection Interrupt Enable 0: Disables STOP condition Detection Interrupt Signal. 1: Enables STOP condition Detection Interrupt Signal.	R/W
2	HDREXDIE	HDR Exit Pattern Detection Interrupt Enable*1 0: Disables HDR Exit Pattern Detection Interrupt Signal. 1: Enables HDR Exit Pattern Detection Interrupt Signal.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
4	NACKDIE	NACK Detection Interrupt Enable*2 0: Disables NACK Detection Interrupt Signal. 1: Enables NACK Detection Interrupt Signal.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	TENDIE	Transmit End Interrupt Enable*2 0: Disables Transmit End Interrupt Signal. 1: Enables Transmit End Interrupt Signal.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	ALIE	Arbitration Lost Interrupt Enable*2 0: Disables Arbitration Lost Interrupt Signal. 1: Enables Arbitration Lost Interrupt Signal.	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	TODIE	Timeout Detection Interrupt Enable 0: Disables Timeout Detection Interrupt Signal. 1: Enables Timeout Detection Interrupt Signal.	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
24	WUCNDDIE	Wake-Up Condition Detection Interrupt Enable 0: Disables Wake-Up Condition Detection Interrupt Signal. 1: Enables Wake-Up Condition Detection Interrupt Signal.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports all I3C mode.
Note 2. This bit supports I^{2}C mode.

The BIE register enables signaling of outstanding bus interrupts received by I3C.

STCNDDIE bit (START Condition Detection Interrupt Enable)

This bit enables or disables the START Condition Detection interrupt requests when the BST.STCNDDF flag is set to 1.

SPCNDDIE bit (STOP Condition Detection Interrupt Enable)

This bit enables or disables the STOP Condition Detection interrupt requests when the BST.SPCNDDF flag is set to 1.

HDREXDIE bit (HDR Exit Pattern Detection Interrupt Enable)

This bit enables or disables the HDR Exit Pattern Detection interrupt requests when the BST.HDREXDF flag is set to 1.

NACKDIE bit (NACK Detection Interrupt Enable)

This bit enables or disables the NACK Detection interrupt requests when the BST.NACKDF flag is set to 1.

TENDIE bit (Transmit End Interrupt Enable)

This bit enables or disables the Transmit End interrupt (I3C_TEND) requests when the BST.TENDF flag is set to 1.

ALIE bit (Arbitration Lost Interrupt Enable)

This bit enables or disables the Arbitration lost interrupt requests when the BST.ALF flag is set to 1.

TODIE bit (Timeout Detection Interrupt Enable)

This bit enables or disables the Timeout Detection interrupt requests when the BST.TODF flag is set to 1.

WUCNDDIE bit (Wake-Up Condition Detection Interrupt Enable)

This bit enables or disables the Wake-up Condition Detection interrupt (I3C_WU) requests when the BST.WUCNDDF flag is set to 1.

40.2.49 BSTFC : Bus Status Force Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x1DC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	WUCNDDFC	—	—	—	TODFC	—	—	—	ALFC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	TENDFC	—	—	—	NACKDFC	—	HDREXDFC	SPCNDDFC	STCNDDFC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STCNDDFC	START condition Detection Force 0: Not Force START condition Detection Interrupt for software testing. 1: Force START condition Detection Interrupt for software testing.	W
1	SPCNDDFC	STOP condition Detection Force 0: Not Force STOP condition Detection Interrupt for software testing. 1: Force STOP condition Detection Interrupt for software testing.	W
2	HDREXDFC	HDR Exit Pattern Detection Force*1 0: Not Force HDR Exit Pattern Detection Interrupt for software testing. 1: Force HDR Exit Pattern Detection Interrupt for software testing.	W
3	—	This bit is read as 0.	R
4	NACKDFC	NACK Detection Force*2 0: Not Force NACK Detection Interrupt for software testing. 1: Force NACK Detection Interrupt for software testing.	W
7:5	—	These bits are read as 0.	R
8	TENDFC*3	Transmit End Force*2 0: Not Force Transmit End Interrupt for software testing. 1: Force Transmit End Interrupt for software testing.	W
15:9	—	These bits are read as 0.	R
16	ALFC	Arbitration Lost Force*2 0: Not Force Arbitration Lost Interrupt for software testing. 1: Force Arbitration Lost Interrupt for software testing.	W
19:17	—	These bits are read as 0.	R
20	TODFC	Timeout Detection Force 0: Not Force Timeout Detection Interrupt for software testing. 1: Force Timeout Detection Interrupt for software testing.	W
23:21	—	These bits are read as 0.	R
24	WUCNDDFC	Wake-Up Condition Detection Force 0: Not Force Wake-Up Condition Detection Interrupt for software testing. 1: Force Wake-Up Condition Detection Interrupt for software testing.	W
31:25	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports all I3C mode.
Note 2. This bit supports I^2C mode.
Note 3. TENDFC does not work unless TDBEF0 = 1.

40.2.50 NTST : Normal Transfer Status Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x1E0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	RSQFF	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TEF	—	—	—	TABTF	RSPQFF	CMDQEF	IBIQEFF	RDBFF0	TDBEF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TDBEF0	Normal Tx Data Buffer Empty Flag 0 0: For I2C protocol mode: PRTS.PRTMD bit = 1. Normal Tx Data Buffer contains transmit data. For I3C protocol mode: PRTS.PRTMD bit = 0. The number of empties in the Normal Tx Data Buffer is less than the NTBTHCTL0.TXDBTH[2:0] threshold. 1: For I2C protocol mode: PRTS.PRTMD bit = 1. Normal Tx Data Buffer contains no transmit data. For I3C protocol mode: PRTS.PRTMD bit = 0. The number of empties in the Normal Tx Data Buffer is the NTBTHCTL0.TXDBTH [2:0] threshold or more.	R/W*3
1	RDBFF0	Normal Rx Data Buffer Full Flag 0 0: For I2C protocol mode: PRTS.PRTMD bit = 1. Normal Rx Data Buffer contains no receive data. For I3C Protocol mode: PRTS.PRTMD bit = 0. The number of entries in the Normal Rx Data Buffer is less than the NTBTHCTL0.RXDBTH[2:0] threshold. 1: For I2C protocol mode: PRTS.PRTMD bit = 1. Normal Rx Data Buffer contains receive data. For I3C Protocol mode: PRTS.PRTMD bit = 0. The number of entries in the Normal Rx Data Buffer is the NTBTHCTL0.RXDBTH [2:0] threshold or more.	R/W*3
2	IBIQEFF	Normal IBI Queue Empty/Full Flag*1 0: For I3C protocol mode (Master): PRTS.PRTMD bit = 0, PRSST.CRMS bit = 1. The number of Normal IBI Queue entries is the NQTHCTL.IBIQTH threshold or less. For I3C protocol mode (Slave) : PRTS.PRTMD bit = 0, PRSST.CRMS bit = 0. If the NQTHCTL.IBIQTH = 0: The number of IBI Data Buffer empties is less than the IBI Data Buffer size. If the NQTHCTL.IBIQTH is other than 0: The number of IBI Data Buffer empties is less than the NQTHCTL.IBIQTH threshold. 1: For I3C protocol mode (Master): PRTS.PRTMD bit = 0, PRSST.CRMS bit = 1. The number of Normal IBI Queue entries is more than the NQTHCTL.IBIQTH threshold. For I3C protocol mode (Slave) : PRTS.PRTMD bit = 0, PRSST.CRMS bit = 0. If the NQTHCTL.IBIQTH = 0: The number of IBI Data Buffer empties is the IBI Data Buffer size. If the NQTHCTL.IBIQTH is other than 0: The number of IBI Data Buffer empties is the NQTHCTL.IBIQTH threshold or more.	R/W*3

Bit	Symbol	Function	R/W
3	CMDQEF	Normal Command Queue Empty Flag*1 0: If the NQTHCTL.CMDQTH = 0: The number of Normal Command Queue empties is less than the Normal Command Queue size. If the NQTHCTL.CMDQTH is other than 0: The number of Normal Command Queue empties is less than the NQTHCTL.CMDQTH threshold. 1: If the NQTHCTL.CMDQTH = 0: The number of Normal Command Queue empties is the Normal Command Queue size. If the NQTHCTL.CMDQTH is other than 0: 1: The number of Normal Command Queue empties is the NQTHCTL.CMDQTH threshold or more.	R/W*3
4	RSPQFF	Normal Response Queue Full Flag*1 0: The number of Normal Response Queue entries is the NQTHCTL.RSPQTH threshold or less. 1: The number of Normal Response Queue entries is more than the NQTHCTL.RSPQTH threshold.	R/W*3
5	TABTF	Normal Transfer Abort Flag*1 0: Normal Transfer Abort does not occur. 1: Normal Transfer Abort occur. To clear, write 0 to this bit after 1 state is read.	R/W*3
8:6	—	These bits are read as 0. The write value should be 0.	R/W
9	TEF	Normal Transfer Error Flag*1 0: Normal Transfer Error does not occur. 1: Normal Transfer Error occurs. To clear, write 0 to this bit after 1 state is read.	R/W*3
19:10	—	These bits are read as 0. The write value should be 0.	R/W
20	RSQFF	Normal Receive Status Queue Full Flag*2 0: The number of Normal Receive Status Queue entries is the NRQTHCTL.RSQTH threshold or less. 1: The number of Normal Receive Status Queue entries is more than the NRQTHCTL.RSQTH threshold.	R/W*3
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports all I3C mode.
Note 2. This bit supports I3C secondary master mode and I3C slave mode.
Note 3. Clearing (to 0) condition : Writing 0 after the 1 state is read.

TDBEF0 bit (Normal Tx Data Buffer Empty Flag 0)

[Setting conditions]

For I^{2}C Protocol mode: PRTS.PRTMD bit = 1.

The following condition 1 is satisfied and any of the following conditions 2 to 4 are satisfied:

1. The NTSTE.TDBEE0 bit = 1 (enables Normal Tx Data Buffer Empty Interrupt Status logging).
2. When data has been transferred from the Normal Tx Data Buffer to the Shift Register and the Normal Tx Data Buffer becomes empty*1.
3. When the PRSST.TRMD bit is set to 1.
4. When the received slave address matches while the TRMD bit = 1.

For I3C Protocol mode: PRTS.PRTMD bit = 0.

The following conditions 1 and 2 are satisfied:

1. The NTSTE.TDBEE0 bit = 1 (enables Normal Tx Data Buffer Empty Interrupt Status logging).
2. When the number of empties in the Normal Tx Data Buffer is the NTBTHCTL0.TXDBTH[2:0] threshold or more (see NTBTHCTL0 register).

[Clearing conditions]

For I^{2}C Protocol mode: PRTS.PRTMD bit = 1.

- When data is written to NTDTBP0.
- When the TRMD bit in PRSST is set to 0.

For I3C protocol mode: PRTS.PRTMD bit = 0.

- Write 0 to this bit after 1 is read.
- On completion of the last write access to Normal Tx Data by DMAC/DTC.

Note 1. When the BST.NACKDF flag is set to 1 while the BSTE.NACKDE bit = 1, I3C aborts data transmission/reception. If the TDBEF0 flag = 0 (next transmit data has been written), data is transferred to the Shift Register and the Normal Tx Data Buffer register becomes empty at the rising edge of the 9th clock cycle, but the TDBEF0 flag is not set to 1.

RDBFF0 bit (Normal Rx Data Buffer Full Flag 0)

[Setting conditions]

For I^{2}C Protocol mode: PRTS.PRTMD bit = 1.

The following condition 1 is satisfied and any of the following condition 2 or 3 is satisfied:

1. The NTSTE.RDBFE0 bit = 1 (enables Normal Rx Data Buffer Full Interrupt Status logging).
2. When Rx data is transferred from Shift Register to Normal Rx Data Buffer.
   The RDBFF0 flag is set to 1 on the rising edge of the 8th or 9th SCL clock cycle (selected in the ACKTWE bit in SCSTRCTL).
3. When the received slave address matches after a START (or Repeated START) condition is detected with the TRMD bit in PRSST set to 0.

For I3C Protocol mode: PRTS.PRTMD bit = 0.

The following conditions 1 and 2 are satisfied:

1. The NTSTE.RDBFE0 bit = 1 (enables Normal Rx Data Buffer Full Interrupt Status logging).
2. When the number of Normal Rx Data Buffer entries is the NTBTHCTL0.RXDBTH[2:0] threshold or more (see NTBTHCTL0 register).

[Clearing conditions]

For I^{2}C Protocol mode: PRTS.PRTMD bit = 1.

- When data is read from NTDTBP0.

For I3C Protocol mode: PRTS.PRTMD bit = 0.

- Write 0 to this bit after 1 is read.
- On completion of the last read access to Normal Rx Data by DMAC/DTC.

IBIQEFF bit (Normal IBI Queue Empty/Full Flag)

[Setting conditions]

The following 2 conditions are satisfied:

1. The NTSTE.IBIQEFE bit = 1 (enables Normal IBI Queue Empty/Full Interrupt Status logging)
2. For I3C protocol mode (master): PRTS.PRTMD bit = 0, PRSST.CRMS bit = 1.
   - When the number of Normal IBI Queue entries is more than the NQTHCTL.IBIQTH threshold (see NQTHCTL register).

   For I3C protocol mode (slave) : PRTS.PRTMD bit = 0, PRSST.CRMS bit = 0.
   If the NQTHCTL.IBIQTH = 0:
   - When IBI Data Buffer is completely empty.

If the NQTHCTL.IBIQTH is other than 0:

- When the number of IBI Data Buffer empties is the NQTHCTL.IBIQTH threshold or more (see NQTHCTL register).

[Clearing conditions]

For I3C protocol mode (master): PRTS.PRTMD bit = 0, PRSST.CRMS bit = 1.

- Write 0 to this bit after 1 is read.
- On completion of the last read access to IBI Status by DMAC/DTC.

For I3C protocol mode (slave): PRTS.PRTMD bit = 0, PRSST.CRMS bit = 0.

- Write 0 to this bit after 1 is read.
- On completion of the last write access to IBI Status by DMAC/DTC.

CMDQEF bit (Normal Command Queue Empty Flag)

[Setting conditions]

The following 2 conditions are satisfied:

1. The NTSTE.CMDQEE bit = 1 (enables Normal Command Queue Empty Interrupt Status logging).
2. If the NQTHCTL.CMDQTH = 0:
   - When Normal Command Queue is completely empty.

   If the NQTHCTL.CMDQTH is other than 0:
   - When the number of Normal Command Queue empties is the NQTHCTL.CMDQTH threshold or more (see NQTHCTL register).

[Clearing conditions]

- Write 0 to this bit after 1 is read.
- On completion of the last write access to Normal Command by DMAC/DTC.

RSPQFF bit (Normal Response Queue Full Flag)

[Setting conditions]

The following 2 conditions are satisfied:

1. The NTSTE.RSPQFE bit = 1 (enables Normal Response Queue Full Interrupt Status logging).
2. When the number of Normal Response Queue entries is more than the NQTHCTL.RSPQTH threshold (see NQTHCTL register).

[Clearing conditions]

- Write 0 to this bit after 1 is read.
- On completion of the last read access to Normal Receive Status by DMAC/DTC.

TABTF bit (Normal Transfer Abort Flag)

[Setting conditions]

The following 2 conditions are satisfied:

1. The NTSTE.TABTE bit = 1 (enables Normal Transfer Abort Interrupt Status logging).
2. When any transfer is aborted.

[Clearing condition]

- Write 0 to this bit after 1 is read.

TEF bit (Normal Transfer Error Flag)

[Setting conditions]

The following 2 conditions are satisfied:

1. The NTSTE.TEE bit = 1 (enables Normal Transfer Error Interrupt Status logging).
2. When any transfer error occurs on the I3C bus. The Error type for this error is available in the Response or Receive Status structure corresponding to the Transfer command.

[Clearing condition]

- Write 0 to this bit after 1 is read.

RSQFF bit (Normal Receive Status Queue Full Flag)

[Setting conditions]

The following 2 conditions are satisfied:

1. The NTSTE.RSQFE bit = 1 (Normal Receive Status Queue Full Enable).
2. When the number of Normal Receive Status Queue entries is more than the NRQTHCTL.RSQTH threshold (see NRQTHCTL register).

[Clearing conditions]

- Write 0 to this bit after 1 is read.
- On completion of the last read access to Normal Receive Status by DMAC/DTC.

40.2.51 NTSTE : Normal Transfer Status Enable Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x1E4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	RSQFE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TEE	—	—	—	TABTE	RSPQFE	CMDQEE	IBIQEFE	RDBFE0	TDBEE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TDBEE0	Normal Tx Data Buffer Empty Enable 0 0: Disables Normal Tx Data Buffer Empty Interrupt Status logging. 1: Enables Normal Tx Data Buffer Empty Interrupt Status logging.	R/W
1	RDBFE0	Normal Rx Data Buffer Full Enable 0 0: Disables Normal Rx Data Buffer Full Interrupt Status logging. 1: Enables Normal Rx Data Buffer Full Interrupt Status logging.	R/W
2	IBIQEFE	Normal IBI Queue Empty/Full Enable*1 0: Disables Normal IBI Queue Empty/Full Interrupt Status logging. 1: Enables Normal IBI Queue Empty/Full Interrupt Status logging.	R/W
3	CMDQEE	Normal Command Queue Empty Enable*1 0: Disables Normal Command Queue Empty Interrupt Status logging. 1: Enables Normal Command Queue Empty Interrupt Status logging.	R/W
4	RSPQFE	Normal Response Queue Full Enable*1 0: Disables Normal Response queue Full Interrupt Status logging. 1: Enables Normal Response queue Full Interrupt Status logging.	R/W

Bit	Symbol	Function	R/W
5	TABTE	Normal Transfer Abort Enable*1 0: Disables Normal Transfer Abort Interrupt Status logging. 1: Enables Normal Transfer Abort Interrupt Status logging.	R/W
8:6	—	These bits are read as 0. The write value should be 0.	R/W
9	TEE	Normal Transfer Error Enable*1 0: Disables Normal Transfer Error Interrupt Status logging. 1: Enables Normal Transfer Error Interrupt Status logging.	R/W
19:10	—	These bits are read as 0. The write value should be 0.	R/W
20	RSQFE	Normal Receive Status Queue Full Enable*2 0: Disables Normal Receive Status Queue Full Interrupt Status logging. 1: Enables Normal Receive Status Queue Full Interrupt Status logging.	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports all I3C mode.
Note 2. This bit supports I3C secondary master mode and I3C slave mode.

TDBEE0 bit (Normal Tx Data Buffer Empty Enable 0)

When this bit is 1, the operation of NTST.TDBEF0 is enabled.

For the setting conditions and clearing conditions of the NTST.TDBEF0 flag, see the details of NTST.TDBEF0.

RDBFE0 bit (Normal Rx Data Buffer Full Enable 0)

When this bit is 1, the operation of NTST.RDBFF0 is enabled.

For the setting conditions and clearing conditions of the NTST.RDBFF0 flag, see the details of NTST.RDBFF0.

IBIQEFE bit (Normal IBI Queue Empty/Full Enable)

When this bit is 1, the operation of NTST.IBIQEFF is enabled.

For the setting conditions and clearing conditions of the NTST.IBIQEFF flag, see the details of NTST.IBIQEFF.

CMDQEE bit (Normal Command Queue Empty Enable)

When this bit is 1, the operation of NTST.CMDQEF is enabled.

For the setting conditions and clearing conditions of the NTST.CMDQEF flag, see the details of NTST.CMDQEF.

RSPQFE bit (Normal Response Queue Full Enable)

When this bit is 1, the operation of NTST.RSPQFF is enabled.

For the setting conditions and clearing conditions of the NTST.RSPQFF flag, see the details of NTST.RSPQFF.

TABTE bit (Normal Transfer Abort Enable)

When this bit is 1, the operation of NTST.TABTF is enabled.

For the setting conditions and clearing conditions of the NTST.TABTF flag, see the details of NTST.TABTF.

TEE bit (Normal Transfer Error Enable)

When this bit is 1, the operation of NTST.TEF is enabled.

For the setting conditions and clearing conditions of the NTST.TEF flag, see the details of NTST.TEF.

RSQFE bit (Normal Receive Status Queue Full Enable)

When this bit is 1, the operation of NTST.RSQFF is enabled.

For the setting conditions and clearing conditions of the NTST.RSQFF flag, see the details of NTST.RSQFF.

40.2.52 NTIE : Normal Transfer Interrupt Enable Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x1E8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	RSQFIE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TEIE	—	—	—	TABTIE	RSPQFIE	CMDQEIE	IBIQEFIE	RDBFIE0	TDBEIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TDBEIE0	Normal Tx Data Buffer Empty Interrupt Enable 0 0: Disables Normal Tx Data Buffer Empty Interrupt Signal. 1: Enables Normal Tx Data Buffer Empty Interrupt Signal.	R/W
1	RDBFIE0	Normal Rx Data Buffer Full Interrupt Enable 0 0: Disables Normal Rx Data Buffer Full Interrupt Signal. 1: Enables Normal Rx Data Buffer Full Interrupt Signal.	R/W
2	IBIQEFIE	Normal IBI Queue Empty/Full Interrupt Enable*1 0: Disables Normal IBI Queue Empty/Full Interrupt Signal. 1: Enables Normal IBI Queue Empty/Full Interrupt Signal.	R/W
3	CMDQEIE	Normal Command Queue Empty Interrupt Enable*1 0: Disables Normal Command Queue Empty Interrupt Signal. 1: Enables Normal Command Queue Empty Interrupt Signal.	R/W
4	RSPQFIE	Normal Response Queue Full Interrupt Enable*1 0: Disables Normal Response Queue Full Interrupt Signal. 1: Enables Normal Response Queue Full Interrupt Signal.	R/W
5	TABTIE	Normal Transfer Abort Interrupt Enable*1 0: Disables Normal Transfer Abort Interrupt Signal. 1: Enables Normal Transfer Abort Interrupt Signal.	R/W
8:6	—	These bits are read as 0. The write value should be 0.	R/W
9	TEIE	Normal Transfer Error Interrupt Enable*1 0: Disables Normal Transfer Error Interrupt Signal. 1: Enables Normal Transfer Error Interrupt Signal.	R/W
19:10	—	These bits are read as 0. The write value should be 0.	R/W
20	RSQFIE	Normal Receive Status Queue Full Interrupt Enable*2 0: Disables Normal Receive Status Queue Full Interrupt Signal. 1: Enables Normal Receive Status Queue Full Interrupt Signal.	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports all I3C mode.
Note 2. This bit supports I3C secondary master mode and I3C slave mode.

The PIO Interrupt Signal Enable register enables signaling of outstanding interrupts received by I3C.

TDBEIE0 bit (Normal Tx Data Buffer Empty Interrupt Enable 0)

This bit is used to enable or disable the Normal Tx Data buffer empty interrupt (I3C_TX) requests when the NTST.TDBEF0 flag is set to 1.

RDBFIE0 bit (Normal Rx Data Buffer Full Interrupt Enable 0)

This bit is used to enable or disable the Normal Rx Data buffer full interrupt (I3C_RX) requests when the NTST.RDBFF0 flag is set to 1.

IBIQEFIE bit (Normal IBI Queue Empty/Full Interrupt Enable)

This bit is used to enable or disable the Normal IBI Queue Empty/Full interrupt (I3C_IBI) requests when the NTST.IBIQEFF flag is set to 1.

CMDQEIE bit (Normal Command Queue Empty Interrupt Enable)

This bit is used to enable or disable the Normal Command queue empty interrupt (I3C_CMD) requests when the NTST.CMDQEF flag is set to 1.

RSPQFIE bit (Normal Response Queue Full Interrupt Enable)

This bit is used to enable or disable the Normal Response queue full interrupt (I3C_RESP) requests when the NTST.RSPQFF flag is set to 1.

TABTIE bit (Normal Transfer Abort Interrupt Enable)

This bit is used to enable or disable the Normal Transfer Abort interrupt (I3C_EEI) requests when the NTST.TABTF flag is set to 1.

TEIE bit (Normal Transfer Error Interrupt Enable)

This bit is used to enable or disable the Normal Transfer Error interrupt (I3C_EEI) requests when the NTST.TEF flag is set to 1.

RSQFIE bit (Normal Receive Status Queue Full Interrupt Enable)

This bit is used to enable or disable the Normal Receive Status queue full interrupt (I3C_RCV) requests when the NTST.RSQFF flag is set to 1.

40.2.53 NTSTFC : Normal Transfer Status Force Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x1EC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	RSQFFC	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TEFC	—	—	—	TABTFC	RSPQFFC	CMDQEFC	IBIQEFFC	RDBFFC0	TDBEFC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TDBEFC0	Normal Tx Data Buffer Empty Force 0 0: Not Force Normal Tx Data Buffer Empty Interrupt for software testing. 1: Force Normal Tx Data Buffer Empty Interrupt for software testing.	W
1	RDBFFC0	Normal Rx Data Buffer Full Force 0 0: Not Force Normal Rx Data Buffer Full Interrupt for software testing. 1: Force Normal Rx Data Buffer Full Interrupt for software testing.	W
2	IBIQEFFC	Normal IBI Queue Empty/Full Force*1 0: Not Force Normal IBI Queue Empty/Full Interrupt for software testing. 1: Force Normal IBI Queue Empty/Full Interrupt for software testing.	W
3	CMDQEFC	Normal Command Queue Empty Force*1 0: Not Force Normal Command Queue Empty Interrupt for software testing. 1: Force Normal Command Queue Empty Interrupt for software testing.	W
4	RSPQFFC	Normal Response Queue Full Force*1 0: Not Force Normal Response Queue Full Interrupt for software testing. 1: Force Normal Response Queue Full Interrupt for software testing.	W

Bit	Symbol	Function	R/W
5	TABTFC	Normal Transfer Abort Force*1 0: Not Force Normal Transfer Abort Interrupt for software testing. 1: Force Normal Transfer Abort Interrupt for software testing.	W
8:6	—	These bits are read as 0.	R
9	TEFC	Normal Transfer Error Force*1 0: Not Force Normal Transfer Error Interrupt for software testing. 1: Force Normal Transfer Error Interrupt for software testing.	W
19:10	—	The write value should be 0.	W
20	RSQFFC	Normal Receive Status Queue Full Force*2 0: Not Force Normal Receive Status Queue Full Interrupt for software testing. 1: Force Normal Receive Status Queue Full Interrupt for software testing.	W
31:21	—	The write value should be 0.	W

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports all I3C mode.
Note 2. This bit supports I3C secondary master mode and I3C slave mode.

The PIO Interrupt Force register is used to force specific interrupt. It can be used for debug purposes.

TDBEFC0 bit (Normal Tx Data Buffer Empty Force 0)

For software testing, when set to 1, forces the corresponding interrupt, subject to TDBEE0 and TDBEIE0 configuration.

RDBFFC0 bit (Normal Rx Data Buffer Full Force 0)

For software testing, when set to 1, forces the corresponding interrupt, subject to RDBFE0 and RDBFIE0 configuration.

IBIQEFFC bit (Normal IBI Queue Empty/Full Force)

For software testing, when set to 1, forces the corresponding interrupt, subject to IBIQEFE and IBIQEFIE configuration.

CMDQEFC bit (Normal Command Queue Empty Force)

For software testing, when set to 1, forces the corresponding interrupt, subject to CMDQEE and CMDQEIE configuration.

RSPQFFC bit (Normal Response Queue Full Force)

For software testing, when set to 1, forces the corresponding interrupt, subject to RSPQFE and RSPQFIE configuration.

TABTFC bit (Normal Transfer Abort Force)

For software testing, forces the corresponding interrupt, subject to TABTE and TABTIE configuration.

TEFC bit (Normal Transfer Error Force)

For software testing, when set to 1, forces the corresponding interrupt, subject to TEE and TEIE configuration.

RSQFFC bit (Normal Receive Status Queue Full Force)

For software testing, when set to 1, forces the corresponding interrupt, subject to RSQFE and RSQFIE configuration.

40.2.54 HTST : High Priority Transfer Status Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x200

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TEF	—	—	—	TABTF	RSPQFF	CMDQEF	—	RDBFF	TDBEF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TDBEF	High Priority Tx Data Buffer Empty Flag 0: The number of empties in the High Priority Tx Data Buffer is less than the HTBTHCTL.TXDBTH[2:0] threshold. 1: The number of empties in the High Priority Tx Data Buffer is or more than the HTBTHCTL.TXDBTH[2:0] threshold.	R/W*1
1	RDBFF	High Priority Rx Data Buffer Full Flag 0: The number of entries in the High Priority Rx Data Buffer is less than the HTBTHCTL.RXDBTH[2:0] threshold. 1: The number of entries in the High Priority Rx Data Buffer is or more than the HTBTHCTL.RXDBTH[2:0] threshold.	R/W*1
2	—	This bit is read as 0. The write value should be 0.	R/W
3	CMDQEF	High Priority Command Queue Empty Flag 0: If HQTHCTL.CMDQTH is 0, the number of High Priority Command Queue empties is less than the Command Queue size. If HQTHCTL.CMDQTH is other than 0, the number of High Priority Command Queue empties is less than the HQTHCTL.CMDQTH threshold. 1: If HQTHCTL.CMDQTH is 0, the number of High Priority Command Queue empties is the Command Queue size. If HQTHCTL.CMDQTH is other than 0, the number of High Priority Command Queue empties is or more than the HQTHCTL.CMDQTH threshold.	R/W*1
4	RSPQFF	High Priority Response Queue Full Flag 0: The number of High Priority Response Queue entries is less than the HQTHCTL.RSPQTH threshold. 1: The number of High Priority Response Queue entries is or more than the HQTHCTL.RSPQTH threshold.	R/W*1
5	TABTF	High Priority Transfer Abort Flag 0: High Priority Transfer Abort does not occur. 1: High Priority Transfer Abort occurs. To clear, write 0 to this bit after 1 is read.	R/W*1
8:6	—	These bits are read as 0. The write value should be 0.	R/W
9	TEF	High Priority Transfer Error Flag 0: High Priority Transfer Error does not occur. 1: High Priority Transfer Error occurs. To clear, write 0 to this bit after 1 is read.	R/W*1
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.
Note 1. Clearing (to 0) condition : Writing 0 after the 1 state is read.

TDBEF bit (High Priority Tx Data Buffer Empty Flag)

[Setting condition]

The following 2 conditions are satisfied:

1. The HTSTE.TDBEE bit = 1 (Enables High Priority Tx Data Buffer Empty Interrupt Status Logging).
2. When the number of empties in the High Priority Tx Data Buffer is the HTBTHCTL.TXDBTH[2:0] threshold (see register HTBTHCTL) or more.

[Clearing condition]

- Write 0 to this bit after 1 is read.
- On completion of the last write access to “High Priority Tx Data” by DMAC/DTC.

RDBFF bit (High Priority Rx Data Buffer Full Flag)

[Setting condition]

The following 2 conditions are satisfied:

1. The HTSTE.RDBFE bit = 1 (Enables High Priority Rx Data Buffer Full Interrupt Status Logging).
2. When the number of High Priority Rx Data Buffer entries is >= the HTBTHCTL.RXDBTH[2:0] threshold (see register HTBTHCTL).

[Clearing condition]

- Write 0 to this bit after 1 is read.
- On completion of the last read access to “High Priority Rx Data” by DMAC/DTC.

CMDQEF bit (High Priority Command Queue Empty Flag)

[Setting condition]

The following 2 conditions are satisfied:

1. The HTSTE.CMDQEE bit = 1 (Enables High Priority Command Queue Empty Interrupt Status Logging).
2. If the HQTHCTL.CMDQTH = 0:
   - When High Priority Command Queue is completely empty

   If the HQTHCTL.CMDQTH is other than 0:
   - When the number of High Priority Command Queue empties is the HQTHCTL.CMDQTH threshold (see register HQTHCTL) or more.

[Clearing condition]

- Write 0 to this bit after 1 is read.
- On completion of the last write access to “High Priority Command” by DMAC/DTC.

RSPQFF bit (High Priority Response Queue Full Flag)

[Setting condition]

The following 2 conditions are satisfied:

1. The HTSTE.RSPQFE bit = 1 (Enables High Priority Response Queue Full Interrupt Status Logging).
2. When the number of High Priority Response Queue entries is > the HQTHCTL.RSPQTH threshold (see register HQTHCTL).

[Clearing condition]

- Write 0 to this bit after 1 is read.
- On completion of the last read access to “High Priority Receive Status” by DMAC/DTC.

See the description of the RSPQFF bit for the elements used.

TABTF bit (High Priority Transfer Abort Flag)

[Setting condition]

The following 2 conditions are satisfied:

1. The HTSTE.TABTE bit = 1 (Enables High Priority Transfer Abort Interrupt Status Logging).
2. When any transfer is aborted.

[Clearing condition]

- Write 0 to this bit after 1 is read.

TEF bit (High Priority Transfer Error Flag)

[Setting condition]

The following 2 conditions are satisfied:

1. The HTSTE.TEE bit = 1 (Enables High Priority Transfer Error Interrupt Status Logging).
2. When any transfer error occurs on the I3C Bus. The error type for this error is available in the Response structure corresponding to this transfer/command.

[Clearing condition]

- Write 0 to this bit after 1 is read.

40.2.55 HTSTE : High Priority Transfer Status Enable Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x204

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TEE	—	—	—	TABTE	RSPQFE	CMDQEE	—	RDBFE	TDBEE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TDBEE	High Priority Tx Data Buffer Empty Enable 0: Disables High Priority Tx Data Buffer Empty Interrupt Status logging. 1: Enables High Priority Tx Data Buffer Empty Interrupt Status logging.	R/W
1	RDBFE	High Priority Rx Data Buffer Full Enable 0: Disables High Priority Rx Data Buffer Full Interrupt Status logging. 1: Enables High Priority Rx Data Buffer Full Interrupt Status logging.	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	CMDQEE	High Priority Command Queue Empty Enable 0: Disables High Priority Command Queue Empty Interrupt Status logging. 1: Enables High Priority Command Queue Empty Interrupt Status logging.	R/W
4	RSPQFE	High Priority Response Queue Full Enable 0: Disables High Priority Response Queue Full Interrupt Status logging. 1: Enables High Priority Response Queue Full Interrupt Status logging.	R/W
5	TABTE	High Priority Transfer Abort Enable 0: Disables High Priority Transfer Abort Interrupt Status logging. 1: Enables High Priority Transfer Abort Interrupt Status logging.	R/W
8:6	—	These bits are read as 0. The write value should be 0.	R/W
9	TEE	High Priority Transfer Error Enable 0: Disables High Priority Transfer Error interrupt Stats logging. 1: Enables High Priority Transfer Error interrupt Stats logging.	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

Note: This register supports I3C master mode and I3C secondary master mode.

TDBEE bit (High Priority Tx Data Buffer Empty Enable)

When TDBEE is 1, the operation of HTST.TDBEF is enabled.

For the setting conditions and clearing conditions of the HTST.TDBEF flag, see the details of HTST.TDBEF.

RDBFE bit (High Priority Rx Data Buffer Full Enable)

When RDBFE is 1, the operation of HTST.RDBFF is enabled.

For the setting conditions and clearing conditions of the HTST.RDBFF flag, see the details of HTST.RDBFF.

CMDQEE bit (High Priority Command Queue Empty Enable)

When CMDQEE is 1, the operation of HTST.CMDQEF is enabled.

For the setting conditions and clearing conditions of the HTST.CMDQEF flag, see the details of HTST.CMDQEF.

RSPQFE bit (High Priority Response Queue Full Enable)

When RSPQFE is 1, the operation of HTST.RSPQFF is enabled.

For the setting conditions and clearing conditions of the HTST.RSPQFF flag, see the details of HTST.RSPQFF.

TABTE bit (High Priority Transfer Abort Enable)

When TABTE is 1, the operation of HTST.TABTF is enabled.

For the setting conditions and clearing conditions of the HTST.TABTF flag, see the details of HTST.TABTF.

TEE bit (High Priority Transfer Error Enable)

When TEE is 1, the operation of HTST.TEF is enabled.

For the setting conditions and clearing conditions of the HTST.TEF flag, see the details of HTST.TEF.

40.2.56 HTIE : High Priority Transfer Interrupt Enable Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x208

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TEIE	—	—	—	TABTIE	RSPQFIE	CMDQEIE	—	RDBFIE	TDBEIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TDBEIE	High Priority Tx Data Buffer Empty Interrupt Enable 0: Disables High Priority Tx Data Buffer Empty Interrupt Signal. 1: Enables High Priority Tx Data Buffer Empty Interrupt Signal.	R/W
1	RDBFIE	High Priority Rx Data Buffer Full Interrupt Enable 0: Disables High Priority Rx Data Buffer Full Interrupt Signal. 1: Enables High Priority Rx Data Buffer Full Interrupt Signal.	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	CMDQEIE	High Priority Command Queue Empty Interrupt Enable 0: Disables High Priority Command Queue Empty Interrupt Signal. 1: Enables High Priority Command Queue Empty Interrupt Signal.	R/W

Bit	Symbol	Function	R/W
4	RSPQFIE	High Priority Response Queue Full Interrupt Enable 0: Disables High Priority Response Queue Full Interrupt Signal. 1: Enables High Priority Response Queue Full Interrupt Signal.	R/W
5	TABTIE	High Priority Transfer Abort Interrupt Enable 0: Disables High Priority Transfer Abort interrupt Signal. 1: Enables High Priority Transfer Abort interrupt Signal.	R/W
8:6	—	These bits are read as 0. The write value should be 0.	R/W
9	TEIE	High Priority Transfer Error Interrupt Enable 0: Disables High Priority Transfer Error Interrupt Signal. 1: Enables High Priority Transfer Error Interrupt Signal.	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

The high priority interrupt signal enable register enables signaling of outstanding high priority interrupts received by I3C.

TDBEIE bit (High Priority Tx Data Buffer Empty Interrupt Enable)

TDBEIE is used to enable or disable the High Priority Tx Data buffer empty interrupt (I3C_HTX) requests when the HTST.TDBEF flag is set to 1.

RDBFIE bit (High Priority Rx Data Buffer Full Interrupt Enable)

RDBFIE is used to enable or disable the High Priority Rx Data buffer full interrupt (I3C_HRX) requests when the HTST.RDBFF flag is set to 1.

CMDQEIE bit (High Priority Command Queue Empty Interrupt Enable)

CMDQEIE is used to enable or disable the High Priority Command Queue empty interrupt (I3C_HCMD) requests when the HTST.CMDQEF flag is set to 1.

RSPQFIE bit (High Priority Response Queue Full Interrupt Enable)

RSPQFIE is used to enable or disable the High Priority Response Queue full interrupt (I3C_HRESP) requests when the HTST.RSPQFF flag is set to 1.

TABTIE bit (High Priority Transfer Abort Interrupt Enable)

TABTIE is used to enable or disable the High Priority Transfer Abort interrupt (I3C_EEI) requests when the HTST.TABTF flag is set to 1.

TEIE bit (High Priority Transfer Error Interrupt Enable)

TEIE is used to enable or disable the High Priority Transfer Error interrupt (I3C_EEI) requests when the HTST.TEF flag is set to 1.

40.2.57 HTSTFC : High Priority Transfer Status Force Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x20C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TEFC	—	—	—	TABTFC	RSPQFFC	CMDQEFC	—	RDBFFC	TDBEFC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TDBEFC	High Priority Tx Data Buffer Empty Force 0: Not Force High Priority Tx Data Buffer Empty Interrupt for software testing. 1: Force High Priority Tx Data Buffer Empty Interrupt for software testing.	W
1	RDBFFC	High Priority Rx Data Buffer Full Force 0: Not Force High Priority Rx Data Buffer Full Interrupt for software testing. 1: Force High Priority Rx Data Buffer Full Interrupt for software testing.	W
2	—	This bit is read as 0.	R
3	CMDQEFC	High Priority Command Queue Empty Force 0: Not Force High Priority Command Queue Empty Interrupt for software testing. 1: Force High Priority Command Queue Empty Interrupt for software testing.	W
4	RSPQFFC	High Priority Response Queue Full Force 0: Not Force High Priority Response Queue Full Interrupt for software testing. 1: Force High Priority Response Queue Full Interrupt for software testing.	W
5	TABTFC	High Priority Transfer Abort Force 0: Not Force High Priority Transfer Abort Interrupt for software testing. 1: Force High Priority Transfer Abort Interrupt for software testing.	W
8:6	—	These bits are read as 0.	R
9	TEFC	High Priority Transfer Error Force 0: Not Force High Priority Transfer Error Interrupt for software testing. 1: Force High Priority Transfer Error Interrupt for software testing.	W
31:10	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

40.2.58 BCST : Bus Condition Status Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x210

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	BIDLF	BAVLF	BFREF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BFREF	Bus Free Detection Flag 0: Have not Detected Bus Free 1: Have Detected Bus Free	R
1	BAVLF	Bus Available Detection Flag[*1] 0: Have not Detected Bus Available 1: Have Detected Bus Available	R
2	BIDLF	Bus Idle Detection Flag[*1] 0: Have not Detected Bus Idle 1: Have Detected Bus Idle	R
31:3	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note 1. This bit supports all I3C mode.

BFREF bit (Bus Free Detection Flag)

The Bus Free Condition is a period occurring after a STOP and before a START, and with the following duration:

- For Pure Bus: A duration of at least tCAS (see section 62, Electrical Characteristics)
- For Mixed Bus (at least one Legacy I^2C is present on the I3C Bus): A duration of at least tBUF (see section 62, Electrical Characteristics)

[Setting conditions]

- After a STOP condition is detected, when the number of cycles (I3Cφ) that are set by BFRECDT.FRECYC[8:0] has passed in the state of SCL = SDA = 1.
- After setting BCTL.BUSE to 1, when the number of cycles (I3Cφ) that are set by BFRECDT.FRECYC[8:0] has passed in the state of SCL = SDA = 1.

[Clearing conditions]

- When SCL and SDA are other than high.
- When the BCTL.BUSE bit is set to 0.

BAVLF bit (Bus Available Detection Flag)

The Bus Available Condition is a period during which the Bus Free Condition is sustained continuously for a duration of at least tAVAL (see section 62, Electrical Characteristics). A Slave can only issue a START Request (for an In-Band Interrupt, or for a Master Handoff Request) after a Bus Available Condition.

[Setting conditions]

- After a STOP condition is detected, when the number of cycles (I3Cφ) that are set by BAVLCDT.AVLCYC[8:0] has passed in the state of SCL = SDA = 1.
- After setting BCTL.BUSE to 1, when the number of cycles (I3Cφ) that are set by BAVLCDT.AVLCYC[8:0] has passed in the state of SCL = SDA = 1.

[Clearing conditions]

- When SCL and SDA are other than high.
- When the BCTL.BUSE bit is set to 0.

BIDLF bit (Bus Idle Detection Flag)

The Bus Idle Condition is a period during which the Bus Available Condition is sustained continuously for a duration of at least tIDLE (see section 62, Electrical Characteristics).

[Setting conditions]

- After a STOP condition is detected, when the number of cycles (I3Cφ) that are set by BIDLCDT.IDLCYC[17:0] has passed in the state of SCL = SDA = 1.
- After setting BCTL.BUSE to 1, when the number of cycles (I3Cφ) that are set by BIDLCDT.IDLCYC[17:0] has passed in the state of SCL = SDA = 1.

[Clearing conditions]

- When SCL and SDA are other than high.
- When the BCTL.BUSE bit is set to 0.

40.2.59 SVST : Slave Status Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x214

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SVAF[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	HOAF	—	—	—	—	—	—	—	—	DVIDF	HSMCF	—	—	—	—	GCAF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	GCAF	General Call Address Detection Flag 0: General call address does not detect. 1: General call address detects.	R/W*1
4:1	—	These bits are read as 0. The write value should be 0.	R/W
5	HSMCF	Hs-mode Master Code Detection Flag 0: Hs-mode Master Code does not detect. 1: Hs-mode Master Code detects.	R/W*1
6	DVIDF	Device-ID Address Detection Flag 0: Device-ID command does not detect. 1: Device-ID command detects. ● This bit set to 1 when the first frame received immediately after a START condition is detected matches a value of (device ID (1111 100) + 0[W]).	R/W*1
14:7	—	These bits are read as 0. The write value should be 0.	R/W
15	HOAF	Host Address Detection Flag 0: Host address does not detect. 1: Host address detects. ● This bit set to 1 when the received slave address matches the host address (0001 000).	R/W*1
18:16	SVAF[2:0]	Slave Address Detection Flag n (n = 0 to 2) 0: Slave n does not detect 1: Slave n detect	R/W*1
31:19	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I^2C mode.
Note 1. Clearing (to 0) condition : Writing 0 after the 1 state is read.

GCAF flag (General Call Address Detection Flag)

I^2C Normal Wake-Up Mode1 / 2 sets GCAF to 1 when switching from asynchronous operation to synchronous unit.

[Setting conditions]

- This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.
  1. The SVCTL.GCAE bit = 1 (General call address detection is enabled).
  2. When the received slave address matches the general call address (0000 000 + 0 (write)).

[Clearing conditions]

- When 0 is written to the GCAF flag after reading GCAF flag to be 1.
- When a STOP condition is detected.
- When a Repeated START condition is detected.

HSMCF flag (Hs-mode Master Code Detection Flag)

The I^2C Normal Wake-Up Mode 1/2 sets 1 to HSMCF when switching from asynchronous operation to synchronous unit.

[Setting conditions]

- This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.
  1. The SVCTL.HSMCE bit = 1 (Hs-mode master code detection is enabled).
  2. When the first byte received immediately after a START condition is detected matches a value of Hs-mode master code (0000 1XXX) + 1 (NACK).

[Clearing conditions]

- When 0 is written to the HSMCF flag after reading HSMCF flag to be 1.
- When a STOP condition is detected.

DVIDF flag (Device-ID Address Detection Flag)

[Setting conditions]

- This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.
  1. The SVCTL.DVIDE bit = 1 (Device-ID address detection is enabled).
  2. When the first byte received immediately after a START condition or Repeated START condition is detected matches a value of Device ID (1111 100) + 0 (write).

[Clearing conditions]

- When 0 is written to the DVIDF flag after reading DVIDF flag to be 1.
- When a STOP condition is detected.
- This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the first byte when the following 1 and 2 or 1 and 3 are satisfied.
  1. The SVCTL.DVIDE bit = 1 (Device-ID address detection is enabled).
  2. When the first byte received immediately after a START condition or Repeated START condition is detected does not match a value of Device ID (1111 100).
  3. When the first byte received immediately after a START condition or Repeated START condition is detected matches a value of (device ID (1111 100) + 0 [W]) and the second byte does not match any of slave addresses 0 to 2.

HOAF flag (Host Address Detection Flag)

I^2C Normal Wake-Up Mode1 / 2 sets HOAF to 1 at the time of switching from asynchronous operation to synchronous unit.

[Setting conditions]

- This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.
  1. The SVCTL.HOAE bit = 1 (Host address detection is enabled).
  2. When the received slave address matches the host address (0001 000).

[Clearing conditions]

- When 0 is written to the HOAF flag after reading HOAF flag to be 1.
- When a STOP condition is detected.
- When a Repeated START condition is detected.

SVAF[2:0] flags (Slave Address Detection Flag n (n = 0 to 2))

I^2C Normal Wake-Up Mode1 / 2 sets 1 to SVAF[2:0] when switching from asynchronous operation to synchronous unit.

[Setting conditions]

For 7-bit address format: SVDVADn.SADLG bit = 0.

- This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.
  1. The SVCTL.SVAEn bit = 1 (Slave n enabled).
  2. When the received slave address matches the SVDVADn.SVAD[6:0] bits value.

For 10-bit address format: SVDVADn.SADLG bit = 1.

- This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the second byte when all of the followings are satisfied.
  1. The SVCTL.SVAEn bit = 1 (Slave n enabled).
  2. When the received slave address matches a value of 11110 + SVDVADn.SVAD[9:8] bits and the following address matches the SVDVADn.SVAD[7:0] value.

[Clearing conditions]

- When 0 is written to the SVAF[2:0] flag after reading SVAF[2:0] flag to be 1.
- When a STOP condition is detected.

For 7-bit address format: SVDVADn.SADLG bit = 0.

- This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.
  1. The SVCTL.SVAEn bit = 1 (Slave n enabled).
  2. When the received slave address does not match SVDVADn.SVAD[6:0] bits value.

For 10-bit address format: SVDVADn.SADLG bit = 1.

- This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the first byte when all of the followings are satisfied.
  1. The SVCTL.SVAEn bit = 1 (Slave n enabled).
  2. When the received slave address does not match a value of 11110 + SVDVADn.SVAD[9:8] bits.
- This flag is set to 0 at the rising edge of the ninth SCL clock cycle in the second byte when all of the followings are satisfied.
  1. The SVCTL.SVAEn bit = 1 (Slave n enabled).
  2. When the received slave address matches a value of 11110 + SVDVADn.SVAD[9:8] bits and the following address does not match the SVDVADn.SVAD[7:0] value.

40.2.60 WUST : Wake Up Unit Operating Status Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x218

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WUASYNF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	WUASYNF	Wake-up function asynchronous operation status flag 0: I3C synchronous circuit enable condition. 1: I3C asynchronous circuit enable condition.	R
31:1	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3

WUASYNF flag (Wake-up function asynchronous operation status flag)

This bit shows whether I3C is in the TCLK asynchronous operation (WUCTL.WUFE bit = 1).

[Setting condition]

- All of the followings are satisfied.
  1. The WUCTL.WUFE bit = 1 (Wake-up function is enabled)
  2. When the BCST.BFREF flag = 1 after 0 is written to the WUCTL.WUFSYNE bit

[Clearing condition : I^{2}C slave]

- The WUCTL.WUFE bit = 0 (Wake-up function is disabled)
- All of the followings are satisfied.
  1. The WUCTL.WUFE bit = 1 (Wake-up function is enabled)
  2. Wake-up event is detected
  3. When 1 is written to the WUCTL.WUFSYNE bit during WUASYNF flag = 1

[Clearing condition : I3C slave]

- The WUCTL.WUFE bit = 0 (Wake-up function is disabled)
- All of the followings are satisfied.
  1. The WUCTL.WUFE bit = 1 (Wake-up function is enabled)
  2. Wake-up event is detected
  3. When 1 is written to the WUCTL.WUFSYNE bit during WUASYNF flag = 1
  4. When a STOP condition is detected.

[Clearing condition : I^{2}C/I3C slave]

- All of the followings are satisfied.
  1. The WUCTL.WUFE bit = 1 (Wake-up function is enabled)
  2. Wake-Up event is not detected
  3. The WUASYNF flag = 1
  4. The WUCTL.WUFSYNE bit = 1
  5. When a STOP condition is detected.

[Clearing condition : I3C master]

- The WUCTL.WUFE bit = 0 (Wake-up function is disabled)
- All of the followings are satisfied.
  1. The WUCTL.WUFE bit = 1 (Wake-up function is enabled)
  2. Wake-up event is detected.
  3. The WUASYNF flag = 1
  4. The WUCTL.WUFSYNE bit = 1

40.2.61 MRCCPT : MSyncCNT Counter Capture Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x21C

Bit position:	31																															0
Bit field:																																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	MSyncCNT Counter Capture Used in Async Mode 1, not used in Async Mode0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

MRCCPT[31:0] Bits

- Async Mode 1 (Asynchronous Advanced Mode)
  When ATCCNTE.ATCE is enabled, it starts counting. It captures as MSyncCNT for each aME (SDA falling edge of START condition), and store it in the capture register.

40.2.62 DATBASm : Device Address Table Basic Register m (m = 0 to 7)

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x224 + 0x08 × m

Bit position:	31	30	29	28	27	26	25	24	23							16
Bit field:	DVTYP	DVNACK[1:0]		—	—	—	—	—	DVDYAD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6						0
Bit field:	DVIBITS	DVMRRJ	DVSIRRJ	DVIBIPL	—	—	—	—	—	DVSTAD[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	DVSTAD[6:0]	Device Static Address I3C/I^{2}C Static Address	R/W
11:7	—	These bits are read as 0. The write value should be 0.	R/W
12	DVIBIPL	Device IBI Payload 0: IBIs from this Device do not carry a Data Payload. 1: IBIs from this Device do carry a Data Payload.	R/W
13	DVSIRRJ	Device In-Band Slave Interrupt Request Reject 0: This Device shall ACK the SIR. 1: This Device shall NACK the SIR and send the auto-disable CCC.	R/W
14	DVMRRJ	Device In-Band Master Request Reject 0: This Device shall ACK Master Requests. 1: This Device shall NACK Master Requests and send the auto-disable command.	R/W
15	DVIBITS	Device IBI Time-stamp 0: The Master shall not time-stamp IBIs from this Device with Master Time-stamps. 1: The Master shall time-stamp IBIs for this Device with Master Time-stamps.	R/W
23:16	DVDYAD[7:0]	Device I3C Dynamic Address Bit 23 is the parity bit, per the I3C specification, computed and updated by the software driver.	R/W

Bit	Symbol	Function	R/W
28:24	—	These bits are read as 0. The write value should be 0.	R/W
30:29	DVNACK[1:0]	Device NACK Retry Count Device-specific retry count	R/W
31	DVTYP	Device Type 0: I3C Device 1: I^{2}C Device	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

DVIBIPL bit (Device IBI Payload)

Indicates whether IBIs from this Device have a Data Payload. This field reflects the IBI Payload bit in the Device’s Bus Characteristics Register (BCR).

During IBI handling for this Device, the Master shall use this field to determine whether or not to drive reception of the IBI Data Payload. Data continuation is indicated by the T-Bit.

DVSIRRJ bit (Device In-Band Slave Interrupt Request Reject)

Controls whether this Device, when operating in the Master role, will accept vs. reject Slave Interrupt Requests from other Devices.

DVMRRJ bit (Device In-Band Master Request Reject)

Controls whether this Device, when operating in the Master role, will accept vs. reject Master requests from other Devices.

This bit is only valid if I3C declares Non-Current Master Capability.

DVIBITS bit (Device IBI Time-stamp)

Enables or disables IBI time-stamping for a specific Device.

Note: The IBI Status Descriptor for each IBI event indicates whether or not the individual IBI event was actually time-stamped. Set to 0 except for Async mode 0 and Async mode 1 of timing control.

DVNACK[1:0] bits (Device NACK Retry Count)

These bits set the number of retries when a NACK response is received from the slave for the transaction set in the Command Descriptor.

Note: When ENTDAA is executed by Address Assign Command, the setting of this bit is ignored and the transaction ends when NACK is received once.

Note: I3C will retry according to the setting of DVNACK[1:0] bit, even if it receives a NACK for the broadcast address.

Note: If DVNACK[1:0] bits are 0x0, I3C will not retry even for Direct CCCs.

40.2.63 EXDATBAS : Extended Device Address Table Basic Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x2A0

Bit position:	31	30	29	28	27	26	25	24	23							16
Bit field:	EDTYP	EDNACK[1:0]		—	—	—	—	—	EDDYAD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6						0
Bit field:	—	—	—	—	—	—	—	—	—	EDSTAD[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	EDSTAD[6:0]	Extended Device Static Address I3C/I^{2}C static address	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W
23:16	EDDYAD[7:0]	Extended Device I3C Dynamic Address Bit 23 is the parity bit, per the I3C specification, computed and updated by the software driver.	R/W
28:24	—	These bits are read as 0. The write value should be 0.	R/W
30:29	EDNACK[1:0]	Extended Device NACK Retry Count Device-specific retry count	R/W
31	EDTYP	Extended Device Type 0: I3C Device 1: I^{2}C Device	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

40.2.64 SDATBASn : Slave Device Address Table Basic Register n (n = 0 to 2)

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x2B0 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22						16
Bit field:	—	—	—	—	—	—	—	—	—	SDDYAD[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9									0
Bit field:	—	—	—	SDIBIPL	—	SDADLS	SDSTAD[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	SDSTAD[9:0]	Slave Device Static Address[*2] I3C Static Address	R/W
10	SDADLS	Slave Device Address Length Selection[*3] 0: Slave device address length 7 bits selected. 1: Slave device address length 10 bits selected. (I^{2}C device only)	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	SDIBIPL[*1]	Slave Device IBI Payload[*4] This bit is the mirror bit of the SVDCT.TBCR2. 0: IBIs from this device do not carry a data payload. 1: IBIs from this device carry a data payload.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
22:16	SDDYAD[6:0][*1]	Slave Device I3C Dynamic Address[*5]	R/W
31:23	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: SW write to the SDATBAS register of the main master is prohibited.
Note 1. This bit is valid only in SDATBAS0 register.
Note 2. These bits support I^{2}C, I3C secondary master, and I3C slave mode.
Note 3. This bit supports I^{2}C mode.
Note 4. This bit supports I3C secondary master mode and I3C slave mode.
Note 5. These bits support I3C secondary master mode and I3C slave mode.

SDSTAD[9:0] bits (Slave Device Static Address)

When the 7-bit address format is selected (SDADLS bit is 0), the lower 7 bits of SDSTAD[9:0] function as the 7-bit address.

When the 10-bit address format is selected (SDADLS bit is 1), the SDSTAD[9:0] function as the 10-bit address. While the SVCTL.SVAEn bit is 0, the setting of this bit is ignored.

SDIBIPL bit (Slave Device IBI Payload)

Indicates whether IBIs from this Device have a Data Payload. This field reflects the IBI Payload bit in the Device’s Bus Characteristics Register (BCR).

During IBI handling for this Device, the Master shall use this field to determine whether or not to drive reception of the IBI Data Payload. Data continuation is indicated by the T-Bit.

SDDYAD[6:0] bits (Slave Device I3C Dynamic Address)

[Update conditions]

- When writing Dynamic Address value.
- When Slave Address value is its own Static Address in receiving SETDASA CCC (Direct), these bits are updated to Dynamic Address value.*1
- When Dynamic Address Assignment procedure that starts by receiving ENTDAA CCC (Broadcast) is established.*1
- When receiving RSTDAA CCC (Broadcast), all bits are cleared to 0. *1
- When Slave Address value is its own Dynamic Address in receiving RSTDAA CCC (Direct), all bits are cleared to 0. *1
- When Slave Address value is its own Dynamic Address in receiving SETNEWDA CCC (Direct), these bits are updated to the Dynamic Address value. *1
- When receiving SETAASA CCC (Broadcast), these bits are updated to the value of SDSTAD[6:0] bits*2.

Note 1. See the MIPI I3C Specification v1.0.
Note 2. See the MIPI I3C Basic Specification v1.0.

40.2.65 MSDCTm : Master Device Characteristic Table Register m (m = 0 to 7)

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x2D0 + 0x04 × m

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RBCR76[1:0]		RBCR5	RBCR4	RBCR3	RBCR2	RBCR1	RBCR0	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	RBCR0	Max Data Speed Limitation*1 0: No Limitation 1: Limitation	R/W
9	RBCR1	IBI Request Capable 0: Not Capable 1: Capable	R/W

Bit	Symbol	Function	R/W
10	RBCR2	IBI Payload 0: No data byte follows the accepted IBI. 1: Mandatory one or more data bytes follow the accepted IBI. Data byte continuation is indicated by T-Bit.	R/W
11	RBCR3	Offline Capable*2 0: Device will always respond to I3C bus commands. 1: Device will not always respond to I3C bus commands.	R/W
12	RBCR4	Bridge Identifier*3 0: Not a Bridge Device 1: A Bridge Device	R/W
13	RBCR5	SDR Only / SDR and HDR Capable 0: SDR only 1: HDR Capable	R/W
15:14	RBCR76[1:0]	Device Role 0 0: I3C Slave 0 1: I3C Master*4 Others: Setting prohibited	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.
Note 1. Master shall use the GETMXDS CCC to interrogate the Slave for specific limitation.
Note 2. Offline Capable Devices retain the Dynamic Address.
Note 3. Bridge Devices are required to comply with this MIPI Specification for I3C.
Note 4. For an I3C Device acting as I3C Main Master, the BCR Device Role bits will contain the value 01.

The DCT table captures the Device characteristics (PID, BCR, DCR) and assigned Dynamic Address of each Device on the I3C Bus that participates in the Dynamic Address Allocation (ENTDAA) procedure.

RBCRn bits (Received Bus Characteristic Register)

Each I3C Device that is connected to the I3C Bus shall have an associated read-only Bus Characteristics Register (BCR). This read-only register describes the I3C compliant Device’s role and capabilities for use in Dynamic Address assignment and Common Command Codes.

Note: When RBCR2 is 0 and when ACK response to Slave Interrupt Request from I3C Slave by DATBASm.DVSIRRJ = 0 (m = 0 to 7), STOP Condition is issued after ACK response. When RBCR2 is 1 and when ACK response to Slave Interrupt Request from I3C Slave by DATBASm.DVSIRRJ = 0 (m = 0 to 7), IBI Payload is received after ACK response. STOP Condition is issued after end of IBI Payload.

[Update condition]

- When receiving of Bus Characteristics Register (BCR) from Device in the Dynamic Address Assignment procedure starting by receiving ENTDAA CCC (Broadcast).*1

Note 1. See the MIPI I3C Specification v1.0

40.2.66 SVDCT : Slave Device Characteristic Table Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x320

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7							0
Bit field:	TBCR76[1:0]		TBCR5	TBCR4	TBCR3	TBCR2	TBCR1	TBCR0	TDCR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TDCR[7:0]	Transfer Device Characteristic Register 255 available codes for describing the type of sensor, or device. Examples: Accelerometer, gyroscope, composite devices Default value is 0x0: Generic Device	R/W
8	TBCR0	Max Data Speed Limitation*1 0: No Limitation 1: Limitation	R/W
9	TBCR1	IBI Request Capable 0: Not Capable 1: Capable	R/W
10	TBCR2	IBI Payload 0: No data byte follows the accepted IBI. 1: Mandatory one or more data bytes follow the accepted IBI. Data byte continuation is indicated by T-Bit.	R/W
11	TBCR3	Offline Capable*2 0: Device will always respond to I3C bus commands. 1: Device will not always respond to I3C bus commands.	R/W
12	TBCR4	Bridge Identifier*3 0: Not a Bridge Device 1: A Bridge Device	R/W
13	TBCR5	SDR Only / SDR and HDR Capable 0: SDR only 1: HDR Capable	R/W
15:14	TBCR76[1:0]	Device Role 0 0: I3C Slave 0 1: I3C Master*4 Others: Setting prohibited	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.
Note 1. Master shall use the GETMXDS CCC to interrogate the Slave for specific limitation.
Note 2. Offline Capable Devices retain the Dynamic Address.
Note 3. Bridge Devices are required to comply with this MIPI Specification for I3C.
Note 4. For an I3C Device acting as I3C Main Master, the BCR Device Role bits will contain the value 01b.

The DCT table captures the Device characteristics (PID, BCR, DCR) and assigned Dynamic Address of each device on the I3C bus that participates in the Dynamic Address Allocation (ENTDAA) procedure.

TDCR[7:0] bits (Transfer Device Characteristic Register)

Each I3C device that is connected to the I3C bus has an associated Device Characteristics Register (DCR). This register describes the I3C compliant device type such as accelerometer and gyroscope, for use in Dynamic Address assignment and Common Command Codes.

TBCRn bits (Transfer Bus Characteristic Register)

Each I3C device that is connected to the I3C bus has an associated Bus Characteristics Register (BCR). This register describes the role and capabilities of the I3C compliant device for use in Dynamic Address assignment and Common Command Codes.

When I3C Slave issues IBI by Command Descriptor, the condition of TBCRn is described as follows:

[Slave Interrupt Request : No IBI Payload follow the accepted IBI]

- TBCR1 = 1
- TBCR2 = 0

Note: Set DATA_LENGTH[15:0] of Command Descriptor to 0.

[Slave Interrupt Request : IBI Payload follow the accepted IBI]

- TBCR1 = 1
- TBCR2 = 1

Note: Set DATA_LENGTH[15:0] of Command Descriptor to any value.

[Mastership Request]

- TBCR1 = 1
- TBCR76[1:0] = 01b

When I3C Slave receives CCC from I3C Master, it performs the following operations according to the setting of TBCRn:

- When TBCR2 = 1, CMRLG.IBIPSZ[7:0] is sent as the 3rd byte data to GETMRL CCC from I3C Master
- When TBCR0 = 0, NACK responses to GETMXDS CCC from I3C Master
- When TBCR0 = 1, ACK responses to GETMXDS CCC from I3C Master and sends data from CMDSPW, CMDSPR, and CMDSPT registers

40.2.67 SDCTPIDL : Slave Device Characteristic Table Provisional ID Low Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x324

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Transfer Device Provisional ID Low Bits 31 to 16 are read as 0. Bits 15 to 0 are bits [15:0] of device's I3C PID.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

40.2.68 SDCTPIDH : Slave Device Characteristic Table Provisional ID High Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x328

Bit position:	31 … 0
Bit field:	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	Transfer Device Provisional ID High Bits [47:16] of device's I3C PID.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

40.2.69 SVDVADn : Slave Device Address Register n (n = 0 to 2)

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x330 + 0x04 × n

Bit position:	31	30	29	28	27	26	25									16
Bit field:	SDYADV	SSTADV	—	—	SADLG	—	SVAD[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0.	R
25:16	SVAD[9:0]	Slave Address*1 A slave address is set. When rewriting SVAD, change to SVAE = 0 and rewrite.	R
26	—	This bit is read as 0.	R
27	SADLG	Slave Address Length*2 0: The 7-bit address format is selected. 1: The 10-bit address format is selected.	R
29:28	—	These bits are read as 0.	R
30	SSTADV	Slave Static Address Valid*1 0: Slave address is disabled. 1: Slave address is enabled.	R
31	SDYADV*4	Slave Dynamic Address Valid*3 0: Dynamic Address is disabled. 1: Dynamic Address is enabled.	R

Note: S-TYPE3, P-TYPE3
Note 1. These bits support I²C, I3C secondary master, and I3C slave mode.
Note 2. This bit supports I²C mode.
Note 3. This bit supports I3C secondary master mode and I3C slave mode.
Note 4. This bit is valid only in SVDVAD0 register.

SVAD[9:0] bits (Slave Address)

The SVAD[9:0] bits indicate a valid slave address.

[The SVDVAD0.SDYADV bit = 1]

Note: This condition is only for SVDVAD0.SVAD[9:0].

- The SVAD[9:7] bits = 0
- The SVAD[6:0] bits = the SDATBAS0.SDDYAD[6:0] bits

[The SVDVADn.SSTADV bit = 1 and the SVDVADn.SADLG bit = 0]

- The SVAD[9:7] bits = 0
- The SVAD[6:0] bits = the SDATBASn.SDSTAD[6:0] bits

[The SVDVADn.SSTADV bit = 1 and the SVDVADn.SADLG bit = 1]

- The SVAD[9:0] bits = the SDATBASn.SDSTAD[9:0] bits

SADLG bit (Slave Address Length)

[Setting conditions]

- All of the followings are satisfied:

1. The PRTS.PRTMD bit = 1 (I^2C Protocol mode)
2. The SVCTL.SVAEn bit = 1 (Slave n is enabled)
3. The SDATBASn.SDADLS bit = 1 (The address length is 10 bits)

[Clearing condition]

- [Setting condition] is not satisfied.

SSTADV bit (Slave Static Address Valid)

[Setting conditions]

- All of the followings are satisfied:

1. The SVCTL.SVAEn bit = 1 (Slave n is enabled)
2. The SVDVAD0.SDYADV bit = 0 (Dynamic Address is disabled)

   Note: This condition is only for SVDVAD0.SSTADV.
3. If the SVDVADn.SADLG bit = 0, the SDATBASn.SDSTAD[6:0] bits are not all 0
   If the SVDVADn.SADLG bit = 1, the SDATBASn.SDSTAD[9:0] bits are not all 0

[Clearing condition]

- [Setting condition] is not satisfied.

SDYADV bit (Slave Dynamic Address Valid)

[Setting conditions]

- All of the followings are satisfied:

1. The PRTS.PRTMD bit = 0 (I3C Protocol mode)
2. The SVCTL.SVAEn bit = 1 (Slave n is enabled)
3. The SDATBAS0.SDDYAD[6:0] bits are not all 0

Note: This condition is only for SVDVAD0.SDYADV.

[Clearing condition]

- [Setting condition] is not satisfied.

40.2.70 CSECMD : CCC Slave Events Command Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x350

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSRQE	SVIRQE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SVIRQE	Slave Interrupt Requests Enable 0: DISABLED: Slave-initiated Interrupts is Disabled by the Master to control. 1: ENABLED: Slave-initiated Interrupts is Enabled by the Master to control.	R/W
1	MSRQE	Mastership Requests Enable 0: DISABLED: Mastership requests from Secondary Masters is Disabled by the Current Master to control. 1: ENABLED: Mastership requests from Secondary Masters is Enabled by the Current Master to control.	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

SVIRQE bit (Slave Interrupt Requests Enable)

This bit allows the Master to control when Slave-initiated Interrupts are allowed on the I3C Bus.

These four Direct (ENEC/DISEC Format 1) or Broadcast (ENEC/DISEC Format 2) CCCs allows the Master to control when Slave-initiated traffic is (Enable) vs. is not (Disable) allowed on the I3C Bus. This control governs a Slave's attempts to request an Interrupt (ENI), or to request Mastership (ENMR).

[Setting conditions]

- When writing 1
- When receiving ENEC CCC (Broadcast) with ENINT bit = 1.*1
- When ENINT bit = 1 with own Slave Address in receiving ENEC CCC (Direct).*1

[Clearing conditions]

- When writing 0.
- When receiving DISEC CCC (Broadcast) with DISINT bit = 1.*1
- When DISINT bit = 1 with own Slave Address in receiving DISEC CCC (Direct).*1

MSRQE bit (Mastership Requests Enable)

This bit allows the Current Master to control when Mastership requests from Secondary Masters are allowed on the I3C Bus.

[Setting conditions]

- When writing 1.
- When receiving ENEC CCC (Broadcast) with ENMR bit = 1.*1
- When ENMR bit = 1 with own Slave Address in receiving ENEC CCC (Direct).*1

[Clearing conditions]

- When writing 0.
- When receiving DISEC CCC (Broadcast) with DISMR bit = 1.[*1]
- When DISMR bit = 1 with own Slave Address in receiving DISEC CCC (Direct).[*1]

Note 1. See the MIPI I3C Specification v1.0

40.2.71 CEACTST : CCC Enter Activity State Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x354

Bit position:	31																												3			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ACTST[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	ACTST[3:0]	Activity State 0x1: ENTAS0 (1μs: Latency-free operation) 0x2: ENTAS1 (100 μs) 0x4: ENTAS2 (2 ms) 0x8: ENTAS3 (50 ms: Lowest-activity operation) Others: Setting prohibited	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

ACTST[3:0] bits (Activity State)

[Update conditions]

- When writing Activity State value.
- When receiving ENTAS0 CCC (Broadcast), these bits are updated to 0x1.[*1]
- When receiving ENTAS1 CCC (Broadcast), these bits are updated to 0x2.[*1]
- When receiving ENTAS2 CCC (Broadcast), these bits are updated to 0x4.[*1]
- When receiving ENTAS3 CCC (Broadcast), these bits are updated to 0x8.[*1]
- When Slave Address value is its own Slave Address in receiving ENTAS0 CCC (Direct), these bits are updated to 0x1.[*1]
- When Slave Address value is its own Slave Address in receiving ENTAS1 CCC (Direct), these bits are updated to 0x2.[*1]
- When Slave Address value is its own Slave Address in receiving ENTAS2 CCC (Direct), these bits are updated to 0x4.[*1]
- When Slave Address value is its own Slave Address in receiving ENTAS3 CCC (Direct), these bits are updated to 0x8.[*1]

Note 1. See the MIPI I3C Specification v1.0.

40.2.72 CMWLG : CCC Max Write Length Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x358

Bit position:	31																15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MWLG[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	MWLG[15:0]	Max Write Length	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

MWLG[15:0] bits (Max Write Length)

These bits use for the I3C Master to set or get a maximum data write length in bytes for one Slave Device.

This Max Write Length does not affect data write lengths for Broadcast CCCs. The Set/Get Max Write Length value is transmitted over two bytes, with the most significant byte (MSB) transmitted first. The minimum value that Max Write Length can be set to is 8.

[Update conditions]

- When writing Max Write Length value.
- When receiving SETMWL CCC (Broadcast), these bits are updated to MWL value.*1
- When Slave Address value is its own Slave Address in receiving SETMWL CCC (Direct), these bits are updated to MWL value.*1

Note 1. See the MIPI I3C Specification v1.0

40.2.73 CMRLG : CCC Max Read Length Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x35C

Bit position:	31							23								16	15															0
Bit field:	—	—	—	—	—	—	—	—	IBIPSZ[7:0]								MRLG[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	MRLG[15:0]	Max Read Length	R/W
23:16	IBIPSZ[7:0]	IBI Payload Size	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

MRLG[15:0] bits (Max Read Length)

These bits use for the I3C Master to set or get a maximum data read length for one Slave Device.

The Set/Get Max Read Length value is transmitted over the first two bytes, with most significant byte (MSB) transmitted first. The minimum value to which Max Read Length can be set is 16.

[Update conditions]

- When writing Max Read Length value.
- When receiving SETMRL CCC (Broadcast), these bits are updated to MRL value.*1
- When Slave Address value is its own Slave Address in receiving SETMRL CCC (Direct), these bits are updated to MRL value.*1

IBIPSZ[7:0] bits (IBI Payload Size)

These bits use for the I3C Master to set or get optionally a maximum IBI payload size.

For devices with BCR bit 2 set to 1, the Max IBI payload size value is added as a third-byte, where a value of 0 indicates an unlimited payload size. If Timing Control is used, then the minimum IBI payload size is either four bytes or five bytes. If Timing Control is not used, then the minimum IBI payload size is 1 (one byte).

This CCC is optional for the Slave, with two exceptions:

1. This CCC is required if both (a) any private Read Request Message (s) and/or any extended Read Request CCC (s) implemented by the Slave support a variable limit on the maximum number of data bytes that the Slave may return per Message, and (b) this limit is greater than 16 bytes.
2. This CCC is required if the Slave both (a) supports an IBI Payload (as indicated with BCR bit 1), and (b) will transmit more than one byte of private payload (not counting Timing Control bytes, when Timing Control used).

[Update conditions]

- When writing Max IBI payload size value.
- When receiving SETMRL CCC (Broadcast), these bits are updated to IBI payload size value.*1
- When Slave Address value is its own Slave Address in receiving SETMRL CCC (Direct), these bits are updated to IBI payload size value.*1

Note 1. See the MIPI I3C Specification v1.0.

40.2.74 CETSTMD : CCC Enter Test Mode Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x360

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSTMD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TSTMD[7:0]	Test Mode 0x00: Exit Test Mode This value removes all I3C devices from Test Mode. 0x01: Vendor Test Mode This value indicates that I3C devices shall return a random 32bit value in the provisional ID during the Dynamic Address Assignment procedure. Others: Setting prohibited	R
31:8	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

TSTMD[7:0] bits (Test Mode)

When these bits set to 0x00, all I3C Devices remove from Test Mode.

When these bits set to 0x01, I3C Devices shall return a random 32bit value in the Provisional ID during the Dynamic Address Assignment procedure.

The Broadcast CCC informs all I3C Devices that the Master is entering a specified Test Mode during manufacturing or Device test. The Enter Test Mode command Frame format includes a byte that specifies which Test Mode to enter. Supporting I3C Devices shall enter the indicated Test Mode upon receipt of the Enter Test Mode CCC.

[Update condition]

- When receiving ENTTM CCC (Broadcast), these bits are updated to Test Mode Byte value.*1

Note 1. See the MIPI I3C Specification v1.0.

40.2.75 CGDVST : CCC Get Device Status Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x364

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15							8	7	6	5	4	3			0
Bit field:	VDRSV[7:0]								ACTMD[1:0]		PRTE	—	PNDINT[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	PNDINT[3:0]	Pending Interrupt Contains the interrupt number of any pending interrupt, or 0 if no interrupts are pending. This encoding allows for up to 15 numbered interrupts. If more than one interrupt is set, then the highest priority interrupt shall be returned.	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	PRTE	Protocol Error 0: The Slave has not detected a protocol error since the last Status read. 1: The Slave has detected a protocol error since the last Status read.	R/W
7:6	ACTMD[1:0]	Slave Device’s current Activity Mode 0 0: Activity Mode 0 0 1: Activity Mode 1 1 0: Activity Mode 2 1 1: Activity Mode 3	R/W
15:8	VDRSV[7:0]	Vendor Reserved Reserved for vendor-specific meaning	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

PRTE bit (Protocol Error)

If this bit set to 1, then the Slave detects a protocol error since the last Status read.

The Slave checks for such errors. Note that this value self-clears by the hardware upon every successful completion of a Master read of the Slave’s Status.

The Direct CCC is a Get request for one I3C Slave Device to return its current Status, in the two-byte format detailed. Note that byte 0 is the LSB, and byte 1 is the MSB.

[Setting condition]

- When the Slave detected a protocol error.*1

[Clearing condition]

- When transmission by own Slave Address is completed without error after receiving GETSTATUS CCC (Direct).*1

ACTMD[1:0] bits (Slave Device's current Activity Mode)

Contains the two-bit ID of the Slave Device's current Activity Mode (readiness to support data read of sensor or related information).

Note 1. See the MIPI I3C Specification v1.0.

40.2.76 CMDSPW : CCC Max Data Speed W (Write) Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x368

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2		0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	MSWDR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	MSWDR[2:0]	Maximum Sustained Write Data Rate 0 0 0: fscl Max (default value) 0 0 1: 8 MHz 0 1 0: 6 MHz 0 1 1: 4 MHz 1 0 0: 2 MHz Others: Setting prohibited	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

40.2.77 CMDSPR : CCC Max Data Speed R (Read) Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x36C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5		3	2		0
Bit field:	—	—	—	—	—	—	—	—	—	—	CDTTIM[2:0]			MSRDR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	MSRDR[2:0]	Maximum Sustained Read Data Rate 0 0 0: fscl Max (default value) 0 0 1: 8 MHz 0 1 0: 6 MHz 0 1 1: 4 MHz 1 0 0: 2 MHz Others: Setting prohibited	R/W

Bit	Symbol	Function	R/W
5:3	CDTTIM[2:0]	Clock to Data Turnaround Time (TSCO) 0 0 0: 8 ns or less (default value) 0 0 1: 9 ns or less 0 1 0: 10 ns or less 0 1 1: 11 ns or less 1 0 0: 12 ns or less 1 1 1: TSCO is more than 12 ns, and is reported by private agreement. Others Setting prohibited	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

40.2.78 CMDSPT : CCC Max Data Speed T (Turnaround) Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x370

Bit position:	31	30	29	28	27	26	25	24	23–0
Bit field:	MRTE	—	—	—	—	—	—	—	MRTTIM[23:0]
Value after reset:	0	0	0	0	0	0	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
23:0	MRTTIM[23:0]	Maximum Read Turnaround Time 24-bit field can encode turnaround times from 0.0 seconds to 16 seconds. 0x000000: 0 μs (minimum value) 0x000001: 1 μs (resolution) ⋮ 0xF42400: 16 seconds (maximum value) Others: Setting prohibited	R/W
30:24	—	These bits are read as 0. The write value should be 0.	R/W
31	MRTE	Maximum Read Turnaround Time Enable 0: Disables transmission of the Maximum Read Turnaround Time. (GETMXDS Format 1: Without Turnaround) 1: Enables transmission of the Maximum Read Turnaround Time. (GETMXDS Format 2: With Turnaround)	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

40.2.79 CETSM : CCC Exchange Timing Support Information M (Mode) Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x374

Bit position:	31	30	29	28	27	26	25	24	23–16	15–8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	INAC[7:0]	FREQ[7:0]	—	—	—	—	—	SPTASYN1	SPTASYN0	SPTSYN
Value after reset:	0	0	0	0	0	0	0	0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPTSYN	Supports Sync Mode 0: Sync Mode is not supported. 1: Sync Mode is supported.	R/W
1	SPTASYN0	Support Async Mode 0 0: Async Mode 0 is not supported. 1: Async Mode 0 is supported.	R/W
2	SPTASYN1	Support Async Mode 1 0: Async Mode 1 is not supported. 1: Async Mode 1 is supported.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
15:8	FREQ[7:0]	Frequency Byte This byte represents the Slave's internal oscillator frequency in increments of 0.5 MHz (500 kHz), up to 127.5 MHz. 0x00: 32.0 KHz 0x01: 0.5 MHz 0x02: 1.0 MHz ⋮ 0xFD: 126.5 MHz 0xFE: 127.0 MHz 0xFF: 127.5 MHz	R/W
23:16	INAC[7:0]	Inaccuracy Byte This byte represents the maximum variation of the Slave's internal oscillator in 1/10th percent (0.1%) increments, up to 25.5%. 0x00: 0.0% 0x01: 0.1% 0x02: 0.2% ⋮ 0xFD: 25.3% 0xFE: 25.4% 0xFF: 25.5%	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

SPTSYN bit (Supports Sync Mode)

Bit mask indicating which Supports Sync Mode of Timing Control Mode (s) the target Slave supports.

If this bit set (has value 1), then that Slave supports the corresponding Supports Sync Mode of Timing Control Mode.

SPTASYN0 bit (Support Async Mode 0)

Bit mask indicating which Supports Async Mode 0 of Timing Control Mode (s) the target Slave supports.

If this bit set (has value 1), then that Slave supports the corresponding Supports Async Mode 0 of Timing Control Mode.

SPTASYN1 bit (Support Async Mode 1)

Bit mask indicating which Supports Async Mode 1 of Timing Control Mode (s) the target Slave supports.

If this bit set (has value 1), then that Slave supports the corresponding Supports Async Mode 1 of Timing Control Mode.

The Directed CCC provides the framework for the Master to query the Exchange Timing capabilities supported by the I3C Slaves. The Get Exchange Timing Support Information CCC causes the addressed Slave to return four data bytes containing key information on supported Timing Control modes, current state, and internal oscillator/clock frequency and inaccuracy.

40.2.80 CETSS : CCC Exchange Timing Support Information S (State) Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x378

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	ICOVF	—	—	—	—	ASYNE[1:0]		SYNE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SYNE	Sync Mode Enabled 0: Sync Mode Disabled 1: Sync Mode Enabled	R/W
2:1	ASYNE[1:0]	Async Mode Enabled Async Mode 3, 2 are unsupported and set to 0. 0 0: All Mode Disable 0 1: Async Mode 0 Enabled 1 0: Async Mode 1 Enabled Others: Setting prohibited	R/W
6:3	—	These bits are read as 0. The write value should be 0.	R/W
7	ICOVF	Internal Counter Overflow 0: Slave has not experienced a counter overflow since the most recent previous check. 1: Slave experienced a counter overflow since the most recent previous check.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

Bit mask indicating which Timing Control Mode (if any) is currently enabled for the target Slave, and whether any counter overflows have occurred since the most recent previous check. If a Timing Control Mode bit is set (has value 1), then that Slave has currently enabled the corresponding Timing Control Mode. If the Overflow bit is set (has value 1), then that Slave experienced a counter overflow since the most recent previous check.

ASYNE[0] Bit (Async Mode 0 Enabled)

Slave Timing Control Async Mode 0 is enabled.

[Setting condition]

- When writing 1.
- When CETSM.SPTASYN[0] bit = 1 and either of the following 1 or 2 are satisfied.

1. When receiving SETXTIME CCC (Broadcast) with Defining byte value 0xDF.
2. When Slave Address value is its own Slave Address in receiving SETXTIME CCC (Direct) with Defining byte value 0xDF.

[Clearing condition]

- When writing 0.
- When CETSM.SPTASYN[0] bit = 1 and either of the following 1 or 2 are satisfied.

1. When receiving SETXTIME CCC (Broadcast) with Defining byte value 0xEF.

2. When Slave Address value is its own Slave Address in receiving SETXTIME CCC (Direct) with Defining byte value 0xEF.

ASYNE[1] Bit (Async Mode 1 Enabled)

Slave Timing Control Async Mode 1 is enabled.

[Setting condition]

- When writing 1.
- When CETSM.SPTASYN[1] bit = 1 and either of the following 1 or 2 are satisfied.

1. When receiving SETXTIME CCC (Broadcast) with Defining byte value 0xEF.
2. When Slave Address value is its own Slave Address in receiving SETXTIME CCC (Direct) with Defining byte value 0xEF.

[Clearing condition]

- When writing 0.
- When CETSM.SPTASYN[1] bit = 1 and either of the following 1 or 2 are satisfied.

1. When receiving SETXTIME CCC (Broadcast) with Defining byte value 0xDF.
2. When Slave Address value is its own Slave Address in receiving SETXTIME CCC (Direct) with Defining byte value 0xDF.

40.2.81 CGHDRCAP : CCC Get HDR Capability Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x37C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSLEN	TSPEN	DDREN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DDREN	HDR-DDR Operation Enable 0: HDR-DDR Operation is Disabled. 1: HDR-DDR Operation is Enabled.	R/W
1	TSPEN	HDR-TSP Operation Enable 0: HDR-TSP Operation is Disabled. 1: HDR-TSP Operation is Enabled.	R/W
2	TSLEN	HDR-TSL Operation Enable 0: HDR-TSL Operation is Disabled. 1: HDR-TSL Operation is Enabled.	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports for I3C secondary master mode and I3C slave mode.

DDREN bit (HDR-DDR Operation Enable)

This bit is used to enable or disable the HDR-DDR communication when used with I3C Slave.

HDR-DDR communication is permitted only when SVDCT.TBCR5 = 1 and DDREN = 1.

[Setting condition]

- When writing 1.

[Clearing condition]

- When writing 0.

TSPEN bit (HDR-TSP Operation Enable)

This bit is used to enable or disable the HDR-TSP communication when used with I3C Slave.

HDR-TSP communication is permitted only when SVDCT.TBCR5 = 1 and TSPEN = 1.

[Setting condition]

- When writing 1.

[Clearing condition]

- When writing 0.

TSLEN bit (HDR-TSL Operation Enable)

This bit is used to enable or disable the HDR-TSL communication when used with I3C Slave.

HDR-TSL communication is permitted only when SVDCT.TBCR5 = 1 and TSLEN = 1.

[Setting condition]

- When writing 1.

[Clearing condition]

- When writing 0.

Note: Refer to the MIPI I3C Specification v1.0.
The lower 8 bits of this register are used as the response data of GETHDRCAP CCC.

40.2.82 BITCNT : Bit Count Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x380

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BCNT[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	BCNT[4:0]	Bit Counter Indicates the number of bits remaining to be transferred. For details on the values, see Table 40.7 and Table 40.8.	R
31:5	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3

BCNT[4:0] bits (Bit Counter)

These bits function as a counter that indicates the number of bits remaining to be transferred at the detection of a sampling edge on the I3C_SCL line.

Table 40.7 I²C / Legacy I²C transfer (1 of 2)

BCNT[4:0]	Master		Slave	
	Address phase	Data phase	Address phase	Data phase
0x00	2 to 1 bits	2 to 1 bits	3 to 1 bits	2 to 1 bits
0x01	3 bits	3 bits	4 bits	3 bits

Table 40.7 I^2C / Legacy I^2C transfer (2 of 2)

BCNT[4:0]	Master		Slave	
	Address phase	Data phase	Address phase	Data phase
0x02	4 bits	4 bits	5 bits	4 bits
0x03	5 bits	5 bits	6 bits	5 bits
0x04	6 bits	6 bits	7 bits	6 bits
0x05	7 bits	7 bits	8 bits	7 bits
0x06	8 bits	8 bits	9 bits	8 bits
0x07	9 bits	9 bits	—	9 bits

Table 40.8 I3C transfer

BCNT[4:0]	SDR*1		HDR-DDR		HDR-TS
	Transmission	Reception	Command/Data	CRC	
0x00	1 bit	2 to 1 bits	19, 1 bits	11, 1 bits	1 Symbol
0x01	2 bits	3 bits	20, 2 bits	12, 2 bits	2 Symbols
0x02	3 bits	4 bits	3 bits	3 bits	3 Symbols
0x03	4 bits	5 bits	4 bits	4 bits	4 Symbols
0x04	5 bits	6 bits	5 bits	5 bits	5 Symbols
0x05	6 bits	7 bits	6 bits	6 bits	6 Symbols
0x06	7 bits	8 bits	7 bits	7 bits	7 Symbols
0x07	8 bits	9 bits	8 bits	8 bits	8 Symbols
0x08	9 bits	—	9 bits	9 bits	9 Symbols
0x09	—	—	10 bits	10 bits	10 Symbols
0x0A	—	—	11 bits	—	11 Symbols
0x0B	—	—	12 bits	—	12 Symbols
0x0C	—	—	13 bits	—	—
0x0D	—	—	14 bits	—	—
0x0E	—	—	15 bits	—	—
0x0F	—	—	16 bits	—	—
0x10	—	—	17 bits	—	—
0x11	—	—	18 bits	—	—

Note 1. The address phase is the same as in Table 40.7.

40.2.83 NQSTLV : Normal Queue Status Level Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x394

Bit position:	31	30	29	28 to 24	23 to 16	15 to 8	7 to 0
Bit field:	—	—	—	IBISCNT[4:0]	IBIQLV[7:0]	RSPQLV[7:0]	CMDQFLV[7:0]
Value after reset:	0	0	0	0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 1 0 0

Bit	Symbol	Function	R/W
7:0	CMDQFLV[7:0]	Normal Command Queue Free Level*1 Number of free buffer entries currently in the Normal Command Queue. Reset value is the depth of the Normal Command Queue.	R

Bit	Symbol	Function	R/W
15:8	RSPQLV[7:0]	Normal Response Queue Level*1 Number of buffer entries currently in the Normal Response Queue.	R
23:16	IBIQLV[7:0]	Normal IBI Queue Level*1 Number of buffer entries currently in the Normal IBI Queue.	R
28:24	IBISCNT[4:0]	Normal IBI Status Count*2 Number of IBI Status entries currently in the Normal IBI Queue.	R
31:29	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note 1. These bits support all I3C mode.
Note 2. These bits support I3C master mode and I3C secondary master mode.

40.2.84 NDBSTLV0 : Normal Data Buffer Status Level Register 0

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x398

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15–8	7–0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RDBLV[7:0]	TDBFLV[7:0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1

Bit	Symbol	Function	R/W
7:0	TDBFLV[7:0]	Normal Tx Data Buffer Free Level Indicates the number of free Tx Data Buffer entries in the Tx Data Queue. Reset value is the depth of the Tx Data Queue.	R
15:8	RDBLV[7:0]	Normal Rx Data Buffer Level Indicates the number of Rx Data Buffer entries in the Rx Data Queue.	R
31:16	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports all I3C mode.

40.2.85 NRSQSTLV : Normal Receive Status Queue Status Level Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x3C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7–0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RSQLV[7:0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
7:0	RSQLV[7:0]	Normal Receive Status Queue Level	R
31:8	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

40.2.86 HQSTLV : High Priority Queue Status Level Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x3C4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RSPQLV[7:0]								CMDQLV[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
7:0	CMDQLV[7:0]	High Priority Command Queue Level Number of free buffer entries currently in the High Priority Command Queue. Reset value is the depth of the High Priority Command Queue.	R
15:8	RSPQLV[7:0]	High Priority Response Queue Level Number of buffer entries currently in the High Priority Response Queue.	R
31:16	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

40.2.87 HDBSTLV : High Priority Data Buffer Status Level Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x3C8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15							8	7							0
Bit field:	RDBLV[7:0]								TDBFLV[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
7:0	TDBFLV[7:0]	High Priority Tx Data Buffer Free Level Indicates the number of free High Priority Tx Data Buffer entries in the High Priority Tx Data Queue. Reset value is the depth of the High Priority Tx Data Queue.	R
15:8	RDBLV[7:0]	High Priority Rx Data Buffer Level Indicates the number of High Priority Rx Data Buffer entries in the High Priority Rx Data Queue.	R
31:16	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

40.2.88 PRSTDBG : Present State Debug Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x3CC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	SDOLV	SCOLV	SDILV	SCILV
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bit	Symbol	Function	R/W
0	SCILV	I3C_SCL Line Signal Level This bit is used to check the I3C_SCL Line level, in order to recover from errors and for debugging.	R
1	SDILV	I3C_SDA Line Signal Level This bit is used to check the I3C_SDA Line level, in order to recover from errors and for debugging.	R
2	SCOLV	SCL Output Level 0: I3C has driven the I3C_SCL pin low. 1: I3C has released the I3C_SCL pin.	R
3	SDOLV	SDA Output Level 0: I3C has driven the I3C_SDA pin low. 1: I3C has released the I3C_SDA pin.	R
31:4	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3

SCILV bit (I3C_SCL Line Signal Level)

This bit is used to check the I3C_SCL Line level, in order to recover from errors and for debugging.

SDILV bit (I3C_SDA Line Signal Level)

This bit is used to check the I3C_SDA Line level, in order to recover from errors and for debugging.

SCOLV bit (SCL Output Level)

This bit is used to select the output level of I3C_SCL pin.

SDOLV bit (SDA Output Level)

This bit is used to select the output level of I3C_SDA pin.

40.2.89 MSERRCNT : Master Error Counters Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x3D0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	M2ECNT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	M2ECNT[7:0]	M2 Error Counter Counts I3C Type M2 errors on the I3C Bus. Cleared upon read out.	R
31:8	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C master mode and I3C secondary master mode.

40.2.90 SC1CPT : SC1 Capture Monitor Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x3E0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SC1C[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	SC1C[15:0]	SC1 Capture	R
31:16	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

SC1C[15:0] bit (SC1 Capture)

- Async Mode 0 (Asynchronous Basic Mode)
  After enabling ATCCNTE.ATCE, SC1C[15:0] Counter counts up from SC1C[15:0] count trigger*1 to SCL rise edge next to ACK for the IBI, and capture it as SC1C[15:0].
- Async Mode 1 (Asynchronous Advanced Mode)
  After enabling ATCCNTE.ATCE, SC1C[15:0] Counter counts up from SC1C[15:0] count trigger*1 to the first aME, and capture it as SC1C[15:0].

Note: As the timing control specification, the SC1C[15:0] counter value is included in the IBI frame as IBI data and is sent to I3C Master, therefore it is not necessary for the I3C Slave to read this register. If the I3C Slave needs to read this register, read it after completing the IBI frame.

Note 1. SW or external trigger can be selected by selection bits.

40.2.91 SC2CPT : SC2 Capture Monitor Register

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x3E4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SC2C[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	SC2C[15:0]	SC2 Capture	R
31:16	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3
Note: This register supports I3C secondary master mode and I3C slave mode.

SC2C[15:0] bits (SC2 Capture)

- Async Mode 0 (Asynchronous Basic Mode)
  After enabling ATCCNTE.ATCE, SC2C[15:0] Counter counts up from SCL rise edge next to ACK for the IBI transmitted from I3C Slave to SCL rise edge next to Tbit after Mandatory Byte, and capture it as SC2C[15:0].
- Async Mode 1 (Asynchronous Advanced Mode)
  After enabling ATCCNTE.ATCE, SC2C[15:0] Counter counts up from SCL rise edge next to ACK for the IBI transmitted from I3C Slave to SCL rise edge next to Tbit after Mandatory Byte, and capture it as SC2C[15:0].

Note: As the timing control specification, the SC2C[15:0] counter value is included in the IBI frame as IBI data and is sent to I3C Master, therefore it is not necessary for the I3C Slave to read this register. If the I3C Slave needs to read this register, read it after completing the IBI frame.

40.2.92 CECTL : Clock Enable Control Resister

Base address: I3C = 0x4035_F000
I3C_NS = 0x5035_F000

Offset address: 0x010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CLKE	Clock Enable 0: Clock disable 1: Clock enable	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

CLKE bit (Clock Enable)

- This bit controls enabling / disabling of clock supply of the communication function.

40.3 Operation

40.3.1 Data Structures

40.3.1.1 Command Descriptor

The write-only Command Descriptor structure is 64 bits in length. The Command Descriptor is put to the Command Queue with writes to the Command Queue Port (High Priority or Normal).

Write to the Command Queue Port (High Priority or Normal) in the following order:

1. First write : The least significant DWORD (Command Descriptor Structure Low).
2. Second write : The most significant DWORD (Command Descriptor Structure High).

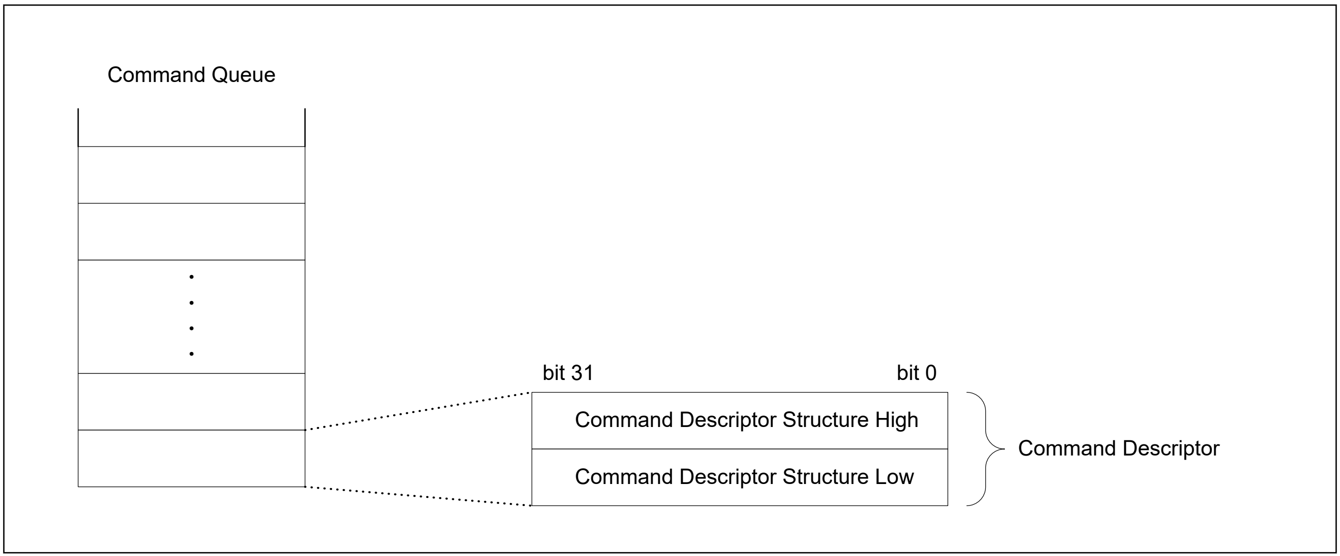


Figure 40.2 Command descriptor data structure

I3C provides a Command Descriptor structure for each command type as follows:

- Address Assign Command
- Immediate Transfer Command
- Regular Transfer Command
- Combo Transfer Command
- Internal Control Command

Details are explained in the following sections.

40.3.1.1.1 Address Assign Command

This command is used for address assignment (ENTDAA, SETDASA).

Note: When issuing SETAASA CCC, use the Immediate Transfer command.

The I3C provides an address assign command for the following mode:

- I3C Master mode

Details of the Address Assign command structure are as follows.

Bit position:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TOC	ROC	DEV_COUNT[3:0]				—	—	—	—	EXT_DEVICE	DEV_INDEX[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	CMD[7:0]								TID[3:0]				CMD_ATTR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CMD_ATTR[2:0]	Command Attributes 0x0: XFER: Regular Transfer 0x1: IMMED_DATA_XFER: Immediate Data Transfer 0x2: ADDR_ASSGN_CMD: Address Assignment Command 0x3: WWR_COMBO_XFER: Write + Write/Read Combo Transfer 0x7: INTERNAL_CONTROL: Internal Control command Others: Setting prohibited	W
6:3	TID[3:0]	Transaction ID	W
14:7	CMD[7:0]	Transfer Command CCC Value	W
15	—	The write value should be 0.	W
20:16	DEV_INDEX[4:0]	Device Index	W
21	EXT_DEVICE	Extended Device Index 0: Use the DATBASm table indicated by DEV_INDEX[4:0]. 1: Use the EXDATBAS table.	W
25:22	—	The write value should be 0.	W
29:26	DEV_COUNT[3:0]	Device Count	W
30	ROC	Response on Completion 0: NOT_REQUIRED: Response Status is not required. 1: REQUIRED: Response Status is required.	W
31	TOC	Terminate on Completion 0: RESTART: Issue Repeated START (Sr) at end of transfer 1: STOP: Issue STOP (P) at end of transfer	W
63:32	—	The write value should be 0.	W

CMD_ATTR[2:0] bits (Command Attributes)

Command Type, defining the format of the other fields.

TID[3:0] bits (Transaction ID)

Used as an identification tag for this command. This field shall be populated by the software Driver, and the same value shall be reflected in the Response Descriptor.

CMD[7:0] bits (Transfer Command CCC Value)

Specifies CCC code indicating whether Address Assignment uses ENTDAA or SETDASA commands. The field comprises the entire command code (ENTDAA or SETDASA).

DEV_INDEX[4:0] bits (Device Index)

Indicates the DATBASm table index for the Slave device being addressed with the transfer. Static and device addressing related information are stored to this index in the DATBASm.

DEV_COUNT[3:0] bits (Device Count)

Indicates the number of devices that a dynamic address is assigned to.

ROC bit (Response on Completion)

Controls whether Response Status is sent after successful completion of the Transfer command. The successful completion is read from register NRSPQP. Upon unsuccessful transfer the Response Status is sent.

TOC bit (Terminate on Completion)

Controls what bus condition to issue after the Transfer command completes.

For ENTDAA, a STOP condition is issued regardless of the setting value of TOC. It is meaningful for SETDASA transfers.

When sending SETDASA CCC by TOC = 0 (RESTART), the next command must be set to SETDASA CCC with the Address Assign Command.

When the next command is not the same SETDASA CCC flame, it must be set to TOC = 1 (STOP).

40.3.1.1.2 Immediate Transfer Command

This structure directly contains data (max 4 bytes) to be transferred, and as a result is only useful for Transfers/CCCs that write data. This structure shall not be used for Read operations (for example, to Rx data).

When transmitting data of 4 bytes or less, use this Immediate Transfer Command to communicate.

When transmitting data of 5 bytes or more, use the Regular Transfer Command to communicate.

For the Regular Transfer Command, see section 40.3.1.1.3. Regular Transfer Command.

I3C provides an Immediate Transfer Command for the following mode:

- I3C Master Mode

Details of the Immediate Transfer Command Structure of each mode are shown in this section.

Bit position:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
Bit field:	DATA_BYTE_4[7:0]								DATA_BYTE_3[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Bit field:	DATA_BYTE_2[7:0]								DATA_BYTE_1[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TOC	ROC	RNW	MODE[2:0]			BYTE_CNT[2:0]			—	EXT_DEVICE	DEV_INDEX[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CP	CMD[7:0]								TID[3:0]				CMD_ATTR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CMD_ATTR[2:0]	Immediate Data Transfer Command Attribute 0x0: XFER: Regular Transfer 0x1: IMMED_DATA_XFER: Immediate Data Transfer 0x2: ADDR_ASSGN_CMD: Address Assignment Command 0x3: WWR_COMBO_XFER: Write + Write/Read Combo Transfer 0x7: INTERNAL_CONTROL: Internal Control command Others: Setting prohibited	W
6:3	TID[3:0]	Immediate Data Transfer Transaction ID	W
14:7	CMD[7:0]	Immediate Data Transfer CCC / HDR Command Code Value For CCC: 8 bits For HDR: 7 bits	W
15	CP	Immediate Data Transfer Command Present 0: TRANSFER: This structure describes an SDR transfer, so the CMD field is not valid. 1: CCC_HDR: This structure describes a CCC or HDR transfer, so the CMD field is valid.	W
20:16	DEV_INDEX[4:0]	Immediate Data Transfer Device Index	W
21	EXT_DEVICE	Immediate Data Transfer Extended Device Index 0: Use the DATBASm table indicated by DEV_INDEX[4:0]. 1: Use the EXDATBAS table.	W
22	—	The write value should be 0.	W
25:23	BYTE_CNT[2:0]	Immediate Data Transfer Byte Count 0x0: No payload 0x1 to 0x4: N bytes are valid. Others: Setting prohibited	W
28:26	MODE[2:0]	Immediate Data Transfer Mode and Speed Values 0x0: I3C SDR0 / Data rate : STDBR (I3C mode) Legacy I^{2}C Message 0 / Data rate : STDBR (I^{2}C mode) 0x1: I3C SDR1 / Data rate : EXTBR (I3C mode) Legacy I^{2}C Message 0 / Data rate : EXTBR (I2C mode) 0x2: I3C SDR2 / Data rate : STDBR × 2 (I3C mode) Reserved (I^{2}C mode) 0x3: I3C SDR3 / Data rate : EXTBR × 2 (I3C mode) Reserved (I^{2}C mode) 0x4: I3C SDR4 / Data rate : EXTBR × 4 (I3C mode) Reserved (I^{2}C mode) 0x5: I3C HDR-TS / Data rate : STDBR (I3C mode) HDR-Ternary Mode. Selection of HDR-TSP vs. HDR-TSL depends on the value of register BCTL, field BMDS Reserved (I^{2}C mode) 0x6: I3C HDR-DDR / Data rate : STDBR (I3C mode) Reserved (I^{2}C mode) Others: Setting prohibited	W
29	RNW	Immediate Data Transfer R/W 0: WRITE: Write transfer 1: READ: Read transfer	W

Bit	Symbol	Function	R/W
30	ROC	Immediate Data Transfer Response on Completion 0: NOT_REQUIRED: Response Status is not required. 1: REQUIRED: Response Status is required.	W
31	TOC	Immediate Data Transfer Terminate on Completion 0: RESTART: Issue Repeated START (Sr) at end of data transfer 1: STOP: Issue STOP (P) at end of data transfer	W
39:32	DATA_BYTE_1[7:0]	Immediate Data Transfer Data Byte 1 Direct argument	W
47:40	DATA_BYTE_2[7:0]	Immediate Data Transfer Data Byte 2 Direct argument	W
55:48	DATA_BYTE_3[7:0]	Immediate Data Transfer Data Byte 3 Direct argument	W
63:56	DATA_BYTE_4[7:0]	Immediate Data Transfer Data Byte 4 Direct argument	W

CMD_ATTR[2:0] bits (Immediate Data Transfer Command Attribute)

Command Type, defining the format of the other fields.

TID[3:0] bits (Immediate Data Transfer Transaction ID)

Used as an identification tag for this command. This field shall be populated by the software Driver, and the same value shall be reflected in the Response Descriptor.

CP bit (Immediate Data Transfer Command Present)

Indicates whether CMD field is valid for CCC or HDR Transfer.

DEV_INDEX[4:0] bits (Immediate Data Transfer Device Index)

Indicates the DATBASm Table index for the Slave Device being addressed with the transfer. Static and Device addressing related information will be stored to this index in the DATBASm.

BYTE_CNT[2:0] bits (Immediate Data Transfer Byte Count)

Number of valid data bytes to use in this Immediate Data Transfer Descriptor.

This field must be set to non-zero value, except for CCCs that does not have payload defined.

This field must be set to even value, when specifying HDR mode (MODE[2:0] = 0x5 or 0x6).

MODE[2:0] bits (Immediate Data Transfer Mode and Speed Values)

Sets the mode and speed for the I3C or I^2C transfer.

Interpretation of this field depends on whether the Device is in I3C Mode vs. I^2C Mode (see the DEVICE field in the DATBASm Table entry indexed by field DEV_INDEX).

RNW bit (Immediate Data Transfer R/W)

Identifies direction of the transfer.

This field shall always be set to 0, because Immediate transfers are valid for Write transactions only.

ROC bit (Immediate Data Transfer Response on Completion)

Controls whether Response Status is required after successful completion of the data transfer command. The successful completion shall be read from NRSPQP register. Upon unsuccessful transfer the Response Status shall always be sent.

TOC bit (Immediate Data Transfer Terminate on Completion)

Controls what Bus condition is issued after completion of the data transfer.

When sending Direct CCC by TOC = 0 (RESTART), next command must be set to same Direct CCC.

When the next command is not the same Direct CCC, must be set to TOC = 1 (STOP).

40.3.1.1.3 Regular Transfer Command

This structure does not contain data to be transferred.

For Master Mode, the data buffer is available through Transfer Data Queue Port (Rx Data Queue Port and Tx Data Queue Port).

When transmitting data of 5 bytes or more, use this Regular Transfer Command to communicate.

When transmitting data of 4 bytes or less, use the Immediate Transfer Command to communicate.

For the Immediate Transfer Command, see section 40.3.1.1.2. Immediate Transfer Command.

For I3C Slave Mode, the IBI Payload buffer is available through IBI Queue Port.

I3C provides a Regular Transfer Command for the following modes:

- I3C Master Mode
- I3C Slave Mode

Details of the regular transfer command structure of each mode are as follows.

(1) I3C Master Mode

Bit position:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
Bit field:	DATA_LENGTH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TOC	ROC	RNW	MODE[2:0]			—	—	—	—	EXT_DEVICE	DEV_INDEX[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CP	CMD[7:0]								TID[3:0]				CMD_ATTR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CMD_ATTR[2:0]	Data Transfer Command Attribute Command Type, defining the format of the other fields. Values: 0x0: XFER: Regular Transfer 0x1: IMMED_DATA_XFER: Immediate Data Transfer 0x2: ADDR_ASSGN_CMD: Address Assignment Command 0x3: WWR_COMBO_XFER: Write + Write/Read Combo Transfer 0x7: INTERNAL_CONTROL: Internal Control command Others: Setting prohibited	W
6:3	TID[3:0]	Data Transfer Transaction ID Identification tag for this command	W
14:7	CMD[7:0]	Data Transfer CCC / HDR Command Code Value Specifies the I3C Command code For CCC: 8 bits For HDR: 7 bits	W
15	CP	Data Transfer Command Present 0: TRANSFER: This structure describes an SDR transfer, so the CMD field is not valid. 1: CCC_HDR: This structure describes a CCC or HDR transfer, so the CMD field is valid.	W

Bit	Symbol	Function	R/W
20:16	DEV_INDEX[4:0]	Data Transfer Device Index	W
21	EXT_DEVICE	Data Transfer Extended Device Index 0: Use the DATBASm Table indicated by DEV_INDEX[4:0]. 1: Use the EXDATBAS table.	W
25:22	—	The write value should be 0.	W
28:26	MODE[2:0]	Data Transfer Speed and Mode 0x0: I3C SDR0 / Data rate : STDBR (I3C mode) Legacy I^{2}C Message 0 / Data rate : STDBR (I^{2}C mode) 0x1: I3C SDR1 / Data rate : EXTBR (I3C mode) Legacy I^{2}C Message 0 / Data rate : EXTBR (I2C mode) 0x2: I3C SDR2 / Data rate : STDBR × 2 (I3C mode) Reserved (I^{2}C mode) 0x3: I3C SDR3 / Data rate : EXTBR × 2 (I3C mode) Reserved (I^{2}C mode) 0x4: I3C SDR4 / Data rate : EXTBR × 4 (I3C mode) Reserved (I^{2}C mode) 0x5: I3C HDR-TS / Data rate : STDBR × 4 (I3C mode) HDR-Ternary Mode. Selection of HDR-TSP vs. HDR-TSL depends on the value of register BCTL, field BMDS Reserved (I^{2}C mode) 0x6: I3C HDR-DDR / Data rate : STDBR (I3C mode) Reserved (I^{2}C mode) Others: Setting prohibited	W
29	RNW	Data Transfer R/W 0: WRITE: Write transfer 1: READ: Read transfer	W
30	ROC	Data Transfer Response on Completion 0: NOT_REQUIRED: Response Status is not required. 1: REQUIRED: Response Status is required.	W
31	TOC	Data Transfer Terminate on Completion 0: RESTART: Issue Repeated START (Sr) at end of transfer 1: STOP: Issue STOP (P) at end of transfer	W
47:32	—	The write value should be 0.	W
63:48	DATA_LENGTH [15:0]	Data Transfer Data Length Indicates the number of bytes to be transferred. This field must be set to non-zero value, except for CCCs that does not have payload defined.	W

CMD_ATTR[2:0] bits (Data Transfer Command Attribute)

Command Type, defining the format of the other fields.

TID[3:0] bits (Data Transfer Transaction ID)

Used as an identification tag for this command. This field shall be populated by the software Driver, and the same value shall be reflected in the Response Descriptor.

CP bit (Data Transfer Command Present)

Indicates whether he contents of the CMD field is valid for a CCC or HDR Transfer.

DEV_INDEX[4:0] bits (Data Transfer Device Index)

Indicates the DATBASm Table index for the Slave Device being addressed with the transfer. Static and Device addressing related information will be stored to this index in the DATBASm.

MODE[2:0] bits (Data Transfer Speed and Mode)

Sets the mode and speed for the I3C or I^{2}C transfer.

Interpretation of this field depends on whether the Device is in I3C Mode vs. I^{2}C Mode (see the DEVICE field in the DATBASm Table entry indexed by field DEV_INDEX).

RNW bit (Data Transfer R/W)

Identifies direction of the transfer.

ROC bit (Data Transfer Response on Completion)

Controls whether Response Status is required after successful completion of the transfer command. The successful completion shall be read from NRSPQP register. Upon unsuccessful transfer the Response Status shall always be sent.

TOC bit (Data Transfer Terminate on Completion)

Controls what Bus condition will be issued after completion of the transfer.

When sending Direct CCC by TOC = 0 (RESTART), next command must be set to same Direct CCC.

When the next command is not the same Direct CCC, must be set to TOC = 1 (STOP).

DATA_LENGTH[15:0] bits (Data Transfer Data Length)

Number of valid data bytes to use in this Regular Transfer Descriptor.

This field must be set to non-zero value, except for CCCs that does not have payload defined.

This field must be set to even value, when specifying HDR mode (MODE[2:0] = 0x5 or 0x6).

Length setting of GETMXDS command should be fixed to 5.

(2) I3C Slave Mode

Bit position:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
Bit field:	DATA_LENGTH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	ROC	RNW	—	—	—	—	—	—	ITS	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	TID[3:0]				CMD_ATTR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CMD_ATTR[2:0]	Data Transfer Command Attribute Command Type, defining the format of the other fields. Values: 0x0: XFER: Regular Transfer 0x1: IMMED_DATA_XFER: Immediate Data Transfer 0x2: ADDR_ASSGN_CMD: Address Assignment Command 0x3: WWR_COMBO_XFER: Write + Write/Read Combo Transfer 0x7: INTERNAL_CONTROL: Internal Control command Others: Setting prohibited	W
6:3	TID[3:0]	Data Transfer Transaction ID Identification tag for this command	W

Bit	Symbol	Function	R/W
21:7	—	The write value should be 0.	W
22	ITS	Include timestamp for Async Mode 0: Do not include timestamp. 1: Include timestamp.	W
28:23	—	The write value should be 0.	W
29	RNW	Data Transfer R/W 0: WRITE: Write transfer (Mastership Request) 1: READ: Read transfer (Slave Interrupt Request)	W
30	ROC	Data Transfer Response on Completion 0: NOT_REQUIRED: Response Status is not required. 1: REQUIRED: Response Status is required.	W
47:31	—	The write value should be 0.	W
63:48	DATA_LENGTH [15:0]	Data Transfer Data Length Indicates the number of bytes to be transferred. This field must be set to non-zero value, except for CCCs that does not have payload defined.	W

CMD_ATTR[2:0] bits (Data Transfer Command Attribute)

Command Type, defining the format of the other fields.

TID[3:0] bits (Data Transfer Transaction ID)

Used as an identification tag for this command. This field shall be populated by the software Driver, and the same value shall be reflected in the Response Descriptor.

RNW bit (Data Transfer R/W)

Identifies direction of the transfer.

ROC bit (Data Transfer Response on Completion)

Controls whether Response Status is required after successful completion of the transfer command. The successful completion shall be read from NRSPQP register. Upon unsuccessful transfer the Response Status shall always be sent.

40.3.1.1.4 Combo Transfer Command

This structure contains a combined Write + Read/Write operation.

The data buffer is available through Transfer Data Queue Port (Rx Data Queue Port and Tx Data Queue Port).

I3C provides a Combo Transfer Command for the following mode:

- I3C Master mode

Details of the Combo Transfer Command Structure of each mode are as follows.

Bit position:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
Bit field:	DATA_LENGTH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Bit field:	OFFSET[15:0]/SUBOFFSET[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TOC	ROC	RNW	MODE[2:0]			16_BIT_SUBOFFSET	FIRST_PHASE_MODE	DATA_LENGTH_POSITION[1:0]		EXT_DEVICE	DEV_INDEX[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CP	CMD[7:0]								TID[3:0]				CMD_ATTR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CMD_ATTR[2:0]	Combo Transfer Command Attribute Command Type, defining the format of the other fields. 0x0: XFER: Regular Transfer 0x1: IMMED_DATA_XFER: Immediate Data Transfer 0x2: ADDR_ASSGN_CMD: Address Assignment Command 0x3: WWR_COMBO_XFER: Write + Write/Read Combo Transfer 0x7: INTERNAL_CONTROL: Internal Control command Others: Setting prohibited	W
6:3	TID[3:0]	Combo Transfer Transaction ID Identification tag for the command	W
14:7	CMD[7:0]	Combo Transfer HDR Command Code Value Specifies the I3C Command code (7 bits).	W
15	CP	Combo Transfer Command Present Indicates whether the CMD field is valid for an HDR Transfer 0: TRANSFER: This structure describes an SDR transfer, so the CMD field is not valid. 1: CCC_HDR: This structure describes an HDR transfer, so the CMD field is valid.	W
20:16	DEV_INDEX[4:0]	Combo Transfer Device Index	W
21	EXT_DEVICE	Combo Transfer Extended Device Index 0: Use the DATBASm table indicated by DEV_INDEX[4:0]. 1: Use the EXDATBAS table.	W
23:22	DATA_LENGTH_POSITION[1:0]	Data Length Field Position 0 0: NO: Do not put length field. 0 1: FIRST: Put length as first field. 1 0: SECOND: Put length as second field. Others: Setting prohibited	W
24	FIRST_PHASE_MODE	Combo Transfer First Phase Mode 0: SDR: First phase is executed in SDR mode. 1: MODE: First phase is executed in the mode indicated by the MODE field.	W
25	16_BIT_SUBOFFSET	Combo Transfer Sub Offset Size 0: 8_BIT_SUBOFFSET: Sub-offset is 8-bits long. Value is encoded in Lower Byte of OFFSET/SUBOFFSET field. 1: 16_BIT_SUBOFFSET: Sub-offset is 16-bits long.	W
28:26	MODE[2:0]	Combo Transfer Speed and Mode Values for I3C Mode 0x0: I3C SDR0/Data rate: STDBR 0x1: I3C SDR1/Data rate: EXTBR 0x2: I3C SDR2/Data rate: STDBR × 2 0x3: I3C SDR3/Data rate: EXTBR × 2 0x4: I3C SDR4/Data rate: EXTBR × 4 0x5: I3C HDR-TS/Data rate: STDBR × 4 HDR-Ternary Mode. Selection of HDR-TSP vs. HDR-TSL depends on the value of register BCTL, field BMDS 0x6: I3C HDR-DDR/Data rate: STDBR Others: Setting prohibited	W

Bit	Symbol	Function	R/W
29	RNW	Combo Transfer R/W Identifies direction of the transfer 0: WRITE: Write transfer 1: READ: Read transfer	W
30	ROC	Combo Transfer Response on Completion 0: NOT_REQUIRED: Response Status is not required. 1: REQUIRED: Response Status is required.	W
31	TOC	Combo Transfer Terminate on Completion 0: RESTART: Issue Repeated START (Sr) at end of transfer 1: STOP: Issue STOP (P) at end of transfer	W
47:32	OFFSET[15:0]/ SUBOFFSET[15:0]	Combo Transfer Offset / Sub-Offset Offset of the target operation	W
63:48	DATA_LENGTH [15:0]	Combo Transfer Data Length Number of bytes to be transferred. This field must be set to non-zero value.	W

CMD_ATTR[2:0] bits (Combo Transfer Command Attribute)

Command Type, defining the format of the other fields.

TID[3:0] bits (Combo Transfer Transaction ID)

Used as an identification tag for this command. This field shall be populated by the software Driver, and the same value shall be reflected in the Response Descriptor.

CP bit (Combo Transfer Command Present)

Indicates whether the contents of the CMD field is valid for an HDR Transfer.

DEV_INDEX[4:0] bits (Combo Transfer Device Index)

Indicates the DATBASm table index for the Slave Device being addressed with the transfer. Static and Device addressing related information will be stored to this index in the DATBASm.

DATA_LENGTH_POSITION[1:0] bits (Data Length Field Position)

Indicates whether and where to put Data Length (DATA_LENGTH) in the first phase of the transfer. This field is only applicable if First phase of the transfer is executed in HDR mode.

Whether 8-bit or 16-bit of Data Length field is used is indicated with 16_BIT_SUBOFFSET field. In case of 8-bit value, it is encoded in Lower Byte of DATA_LENGTH field.

FIRST_PHASE_MODE bits (Combo Transfer First Phase Mode)

Indicates whether the first phase of the Combo Transfer is executed in SDR Mode, vs. the Mode indicated by the MODE field.

MODE[2:0] bits (Combo Transfer Speed and Mode Values for I3C Mode)

Sets the mode and speed for the I3C or I^2C transfer.

Interpretation of this field depends on whether the Device is in I3C Mode vs. I^2C Mode (see the DEVICE field in the DATBASm Table entry indexed by field DEV_INDEX).

RNW bit (Combo Transfer R/W Identifies direction of the transfer)

Identifies direction of the transfer.

ROC bit (Combo Transfer Response on Completion)

Controls whether Response Status is required after successful completion of the data transfer command. The successful completion shall be read from NRSPQP register. Upon unsuccessful transfer the Response Status shall always be sent.

TOC bit (Combo Transfer Terminate on Completion)

Controls what Bus condition is issued after completion of the data transfer.

When sending 2nd flame by TOC = 0 (RESTART) and HDR mode (MODE[2:0] = 0x5 or 0x6), next command must be set to that 1st flame will be the same as HDR mode (MODE[2:0] = 0x5 or 0x6).

When the next command is not the same HDR mode (MODE[2:0] = 0x5 or 0x6), must be set to TOC = 1 (STOP).

DATA_LENGTH[15:0] bit (Combo Transfer Data Length)

Number of valid data bytes to use in this Combo Transfer Descriptor.

This field must be set to non-zero value.

This field must be set to even value, when specifying HDR mode (MODE[2:0] = 0x5 or 0x6).

40.3.1.1.5 Internal Control Command

This structure is used for controlling I3C itself (not for transfer commands).

I3C provides an Internal Control Command for the following mode:

- I3C Master mode

Details of the Internal Control Command Structure are as follows:

Bit position:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	ON_OFF	MIPI_CMD[3:0]				—	TID[3:0]				CMD_ATTR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CMD_ATTR[2:0]	Command Attribute*2 Command Type, defining the format of the other fields. 0x0: XFER: Regular Transfer 0x1: IMMED_DATA_XFER: Immediate Data Transfer 0x2: ADDR_ASSGN_CMD: Address Assignment Command 0x3: WWR_COMBO_XFER: Write + Write/Read Combo Transfer 0x7: INTERNAL_CONTROL: Internal Control command Others: Setting prohibited	W
6:3	TID[3:0]	Transaction ID Identification tag for the command	W
7	—	The write value should be 0.	W
11:8	MIPI_CMD[3:0]	MIPI Alliance Command 0x00: NoOp, so the ON_OFF field is not valid. 0x02: Include 7E (IBA), so the ON_OFF field is valid. Others: Setting prohibited	W
12	ON_OFF	Bus Instance 7E On / Off*1 Enables or disables automatic transmission of the I3C Broadcast Header after every START condition on this I3C Bus instance. 0: IBA_INCLUDE off 1: IBA_INCLUDE on	W

Bit	Symbol	Function	R/W
63:13	—	The write value should be 0.	W

Note 1. The IBA_INCLUDE on state set by MIPI_CMD [3:0] = 0x2 and ON_OFF = 1 is cleared by setting RSTCTL.INTLRST to 1.
Note 2. The Response descriptor is not stored when the Internal Control Command is executed.

40.3.1.2 Response Descriptor

The Response Descriptor is a read-only structure describing the success or failure of a command, and the amount of data transferred.

The Response Descriptor is read from Response Queue with reads from Response Queue Port.

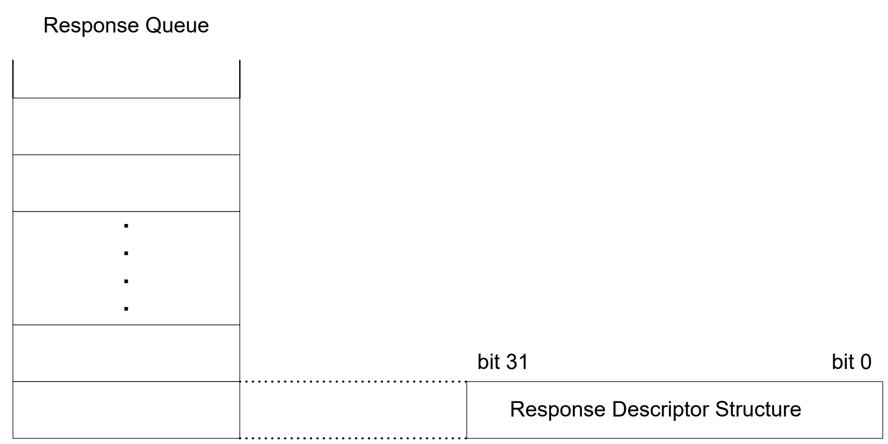


Figure 40.3 Response descriptor data structure

I3C provides a Response Descriptor for the following modes:

- I3C Master mode
- I3C Slave mode

Details of the Response Descriptor structure of each mode are shown in the following sections.

(1) I3C Master Mode

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ERR_STATUS[3:0]				TID[3:0]				—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DATA_LENGTH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DATA_LENGTH [15:0]	Data Length / Device Count The meaning of this field depends on the context: For Write Transfer: Remaining data length (in bytes) For Read Transfer: Received data length (in bytes) For Address Assignment: Remaining Device count	R
23:16	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
27:24	TID[3:0]	Command/Response Transaction ID Identification tag for the command. This value shall match one of commands sent on the Bus. 0x0-0x7: Valid Transaction IDs Others: Reserved	R
31:28	ERR_STATUS[3:0]	Response Error Status 0x0: SUCCESS: Transfer successful, no error 0x1: CRC: CRC Error 0x2: PARITY: Parity Error 0x3: FRAME: Frame Error 0x4: ADDR_HEADER: Address Header Error 0x5: NACK: Address NACKed or Dynamic Address Assignment NACKed 0x6: OVL: Receive Overflow or Transfer Underflow Error 0x8: ABORTED: Aborted 0x9: I^2C_WR_DATA_NACK: NACK received for the I^2C Write Data transfer 0xA: NOT_SUPPORTED: Command with specific parameters not supported by I3C implementation (for example, specific Internal Control codes may not be supported) Others: Reserved	R

Note: In I3C Master mode, when an abnormal command with a specific parameter that is not supported is stored in Command Descriptor, it is indicated as NOT_SUPPORTED (0xA) in ERR_STATUS [3:0].

(2) I3C Slave Mode

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ERR_STATUS[3:0]				TID[3:0]				—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DATA_LENGTH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DATA_LENGTH [15:0]	Data Length Remaining data length (in bytes) for Slave Interrupt Request	R
23:16	—	These bits are read as 0.	R
27:24	TID[3:0]	Command/Response Transaction ID Identification tag for the command. This value matches one of commands sent on the bus. 0x0-0x7: Valid Transaction IDs Others: Reserved	R
31:28	ERR_STATUS[3:0]	Response Error Status 0x0: SUCCESS: Transfer successful, no error. 0x3: FRAME: Frame Error 0x4: ADDR_HEADER: Address Header Error 0x5: NACK: Address NACK'ed or Dynamic Address Assignment NACK'ed 0x6: OVL: Receive Overflow or Transfer Underflow Error 0x8: ABORTED: Aborted 0xA: NOT_SUPPORTED: Command with specific parameters not supported by I3C implementation (for example, specific Internal Control codes may not be supported) Others: Reserved	R

Note: In I3C Slave mode, it is indicated as NOT_SUPPORTED (0xA) in ERR_STATUS[3:0] in the following cases:

- When an abnormal command with a specific parameter that is not supported is stored in the Command Descriptor.
- When the IBI to be transmitted is disabled in the CSECMD register.
- After the normal command for IBI transmission is prepared in the Command Queue, when that IBI is disabled in the CSECMD register by the DISEC CCC frame from the I3C Master.

40.3.1.3 IBI Status Descriptor

The IBI Status Descriptor is a read-only structure describing an IBI event received from a Slave device on the I3C Bus.

The IBI Status Descriptor is read from IBI Queue with reads from IBI Queue Port.

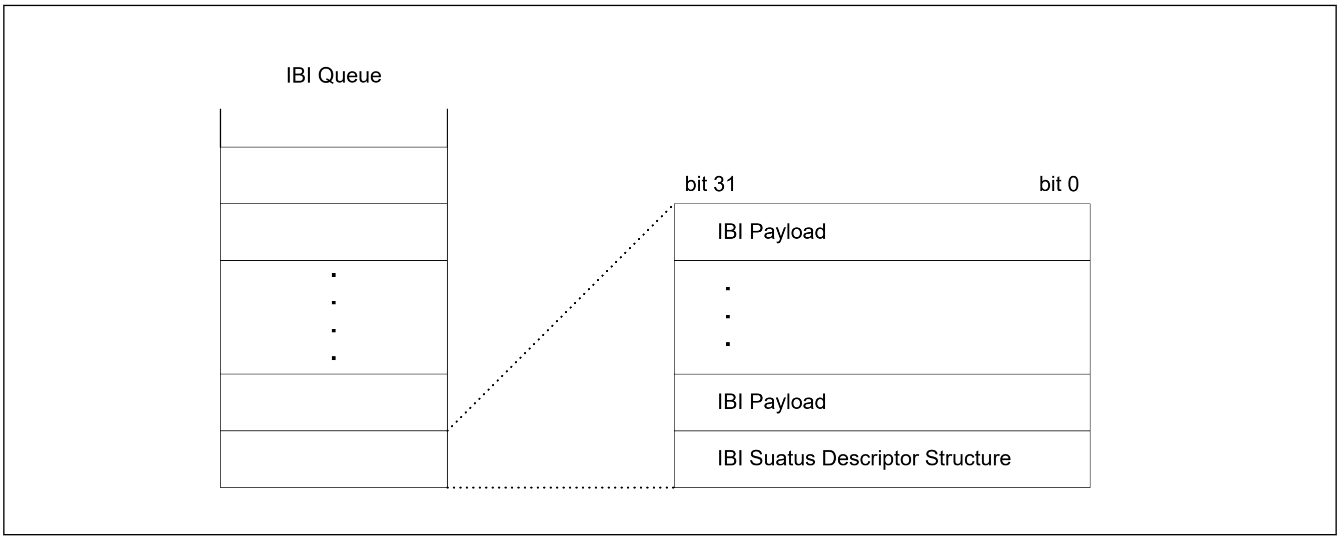


Figure 40.4 IBI status descriptor data structure

I3C provides a IBI Status Descriptor for the following mode:

- I3C Master mode

Details of the IBI Status Descriptor Structure are as follows.

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	IBI_ST	—	—	ERR_STATUS[2:0]			TS	LAST_STATUS	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	IBI_ID[7:0]								DATA_LENGTH[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	DATA_LENGTH[7:0]	IBI Data Length Number of data bytes in IBI Data.	R
15:8	IBI_ID[7:0]	IBI Received ID The meaning of this field depends on the context: For Slave Interrupt or Master Request: Bits 15:9 contain the Slave's Device Address, and bit 8 contains the R/W bit.	R
23:16	—	These bits are read as 0. The write value should be 0.	R
24	LAST_STATUS	Last IBI Status Last IBI status for the IBI transaction.	R

Bit	Symbol	Function	R/W
25	TS	IBI Time-stamp Present Indicates whether a time-stamp is available for the IBI. 0: OFF: IBI is not time-stamped. 1: ON: IBI is time-stamped.	R
28:26	ERR_STATUS[2:0]	IBI Error Status 0x0: SUCCESS 0x3: ERROR: FRAME (Frame Error) 0x4: ERROR: ADDR_HEADER (Address Header Error) 0x5: NACK: Address NACKed 0x7: ERROR: ABORT (Aborted to Master) Others: Reserved	R
30:29	—	These bits are read as 0.	R
31	IBI_ST	IBI Received Status Indicates how the received IBI was handled. 0: The IBI was handled with ACK. 1: NACK: The IBI was handled with NACK, and then Auto-Disabled.	R

LAST_STATUS bits (Last IBI Status)

Even if LAST_STATUS is set to 0, the software driver still evaluates the data payload length by examining the DATA_LENGTH field.

40.3.1.4 Receive Status Descriptor

The Receive Status Descriptor is a read-only structure describing the success or failure of read/write operation from the master, and the amount of data transferred.

The Receive Status Descriptor is read from Receive Status Queue with reads from Receive Status Queue Port.

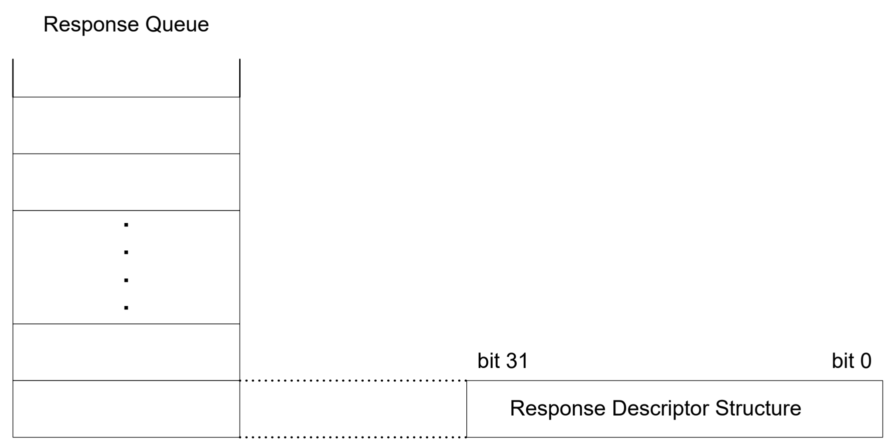


Figure 40.5 Receive status descriptor data structure

I3C provides a Receive Status Descriptor for the following mode:

- I3C Slave mode

Details of the Receive Status Descriptor structure of each mode are as follows.

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DEV_INDEX[2:0]			TRANSFER_TYPE[1:0]		ERR_STATUS[2:0]			CMD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DATA_LENGTH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DATA_LENGTH [15:0]	Data Length The meaning of this field depends on the context. For Write Transfer: Received data length (in bytes) For Read Transfer: Transmitted data length (in bytes)	R
23:16	CMD[7:0]	The contents are different depending on the operation mode. Details are follows: [SDR Private Message Mode] CMD[7]: R/W Type CMD[6:4]: Reserved CMD[3]: I3C_I^{2}C Type CMD[2:0]: Reserved [SDR CCC Mode] CCC code[7:0] [HDR-DDR mode] HDR CMD[7:0] 0x00 – 0x7F: Write Command 0x80 – 0xFF: Read Command [HDR-TSL/TSP mode] HDR CMD[7:0]	R
26:24	ERR_STATUS[2:0]	Error Status 0x0: SUCCESS 0x1: ERROR: CRC (CRC Error) 0x2: ERROR: PARITY (Parity Error) 0x3: ERROR: FRAME (Frame Error) 0x4: ERROR: ADDR_HEADER (Address Header Error) 0x5: ERROR: NACK (Slave NACKed) 0x6: ERROR: OVL (FIFO Overflow/Underflow) 0x7: ERROR: ABORT (Aborted to Master)	R
28:27	TRANSFER_TYPE [1:0]	Transfer Type 0 0: I3C SDR/I^{2}C Message 0 1: I3C CCC 1 0: I3C HDR-DDR 1 1: I3C HDR-TS	R
31:29	DEV_INDEX[2:0]	Device Index Indicates the SVDVADn index for the response with the transfer.	R

40.3.2 Details of Function

40.3.2.1 Operation Mode

The support relationship between the mode select (I3C mode / I^{2}C mode) and operation mode (Master / Slave) on the I3C bus or the I^{2}C bus is shown in Table 40.9.

Table 40.9 Support of operating mode

I3C/I^{2}C Bus	I3C mode		I^{2}C mode	
	Master	Slave	Master	Slave
I3C Bus	✓	✓	—	✓
I^{2}C Bus	—	—	✓	✓

Note: ✓: Supported

—: Unsupported

40.3.2.1.1 Master Mode Operation

(1) I^{2}C Master Operation

(a) Data Write Transfer (Single Buffer transfer)

In master transmit operation, I3C outputs the SCL clock and transmitted data signals as the master device, and the slave device returns acknowledgments. Figure 40.143 shows an example of usage of master transmission and Figure 40.6 to Figure 40.8 show the timing of operations in master transmission.

The following describes the procedure and operations for master transmission.

1. Initial settings. For details, see section 40.3.3.1. Initial Setting Flow.
2. Read the BCST.BFREF flag to check that the bus is open, and then set the CNDCTL.STCND bit to 1 (START condition issuance request). Upon receiving the request, I3C issues a START condition. At the same time, the BFREF flag bit is automatically set to 0, the BST.STCNDDF flag is automatically set to 1 and the STCND bit is automatically set to 0. At this time, if the START condition is detected and the internal levels for the SDA output state and the levels on the I3C_SDA line have matched while the STCND bit = 1, I3C recognizes that issuing of the START condition as requested by the STCND bit has been successfully completed, and bits CRMS and TRMD in the PRSST register are automatically set to 1, placing I3C in master transmit mode. The NTST.TDBEF0 flag is also automatically set to 1 in response to setting of the TRMD bit to 1.
3. Check that the NTST.TDBEF0 flag = 1, and then write the value for transmission (the slave address and the R/W# bit) to the NTDTBP0 register. Once the data for transmission are written to the NTDTBP0 register, the TDBEF0 flag is automatically set to 0, the data are transferred from the Normal Tx Data Buffer to the Shift Register, and the TDBEF0 flag is again set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRMD bit is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, I3C continues in master transmit mode. Because the BST.NACKDF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the CNDCTL.SPCND bit to issue a STOP condition. For data transmission with an address in the 10-bit format, start by writing 1111 0, the 2 higher-order bits of the slave address, and W to the NTDTBP0 register as the first address transmission. Then, as the second address transmission, write the 8 lower-order bits of the slave address to the NTDTBP0 register.
4. After confirming that the NTST.TDBEF0 flag = 1, write the data for transmission to the NTDTBP0 register. I3C automatically holds the I3C_SCL line low until the data for transmission are ready or a STOP condition is issued.
5. After all bytes of data for transmission have been written to the NTDTBP0 register, wait until the value of the BST.TENDF flag returns to 1, and then set the CNDCTL.SPCND bit to 1 (STOP condition issuance request). Upon receiving a STOP condition issuance request, I3C issues the STOP condition.
6. Upon detecting the STOP condition, I3C automatically sets bits CRMS and TRMD in the PRSST register to 00 and enters slave receive mode. Furthermore, it automatically sets the TDBEF0 and TENDF flags to 0, and sets the BST.SPCNDDF flag to 1.
7. After checking that the BST.SPCNDDF flag = 1, set the BST.NACKDF and SPCNDDF flags to 0 for the next transfer operation.

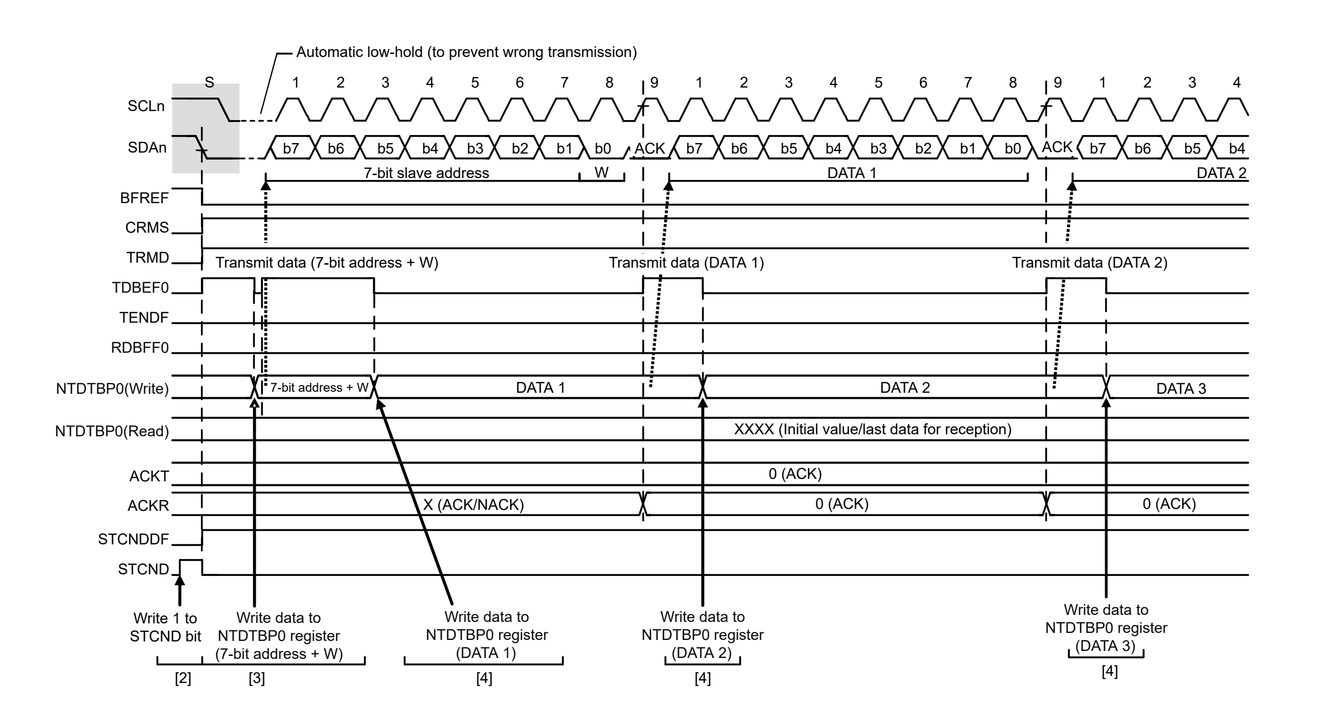


Figure 40.6 Master transmit operation timing (1) (7-bit address format)

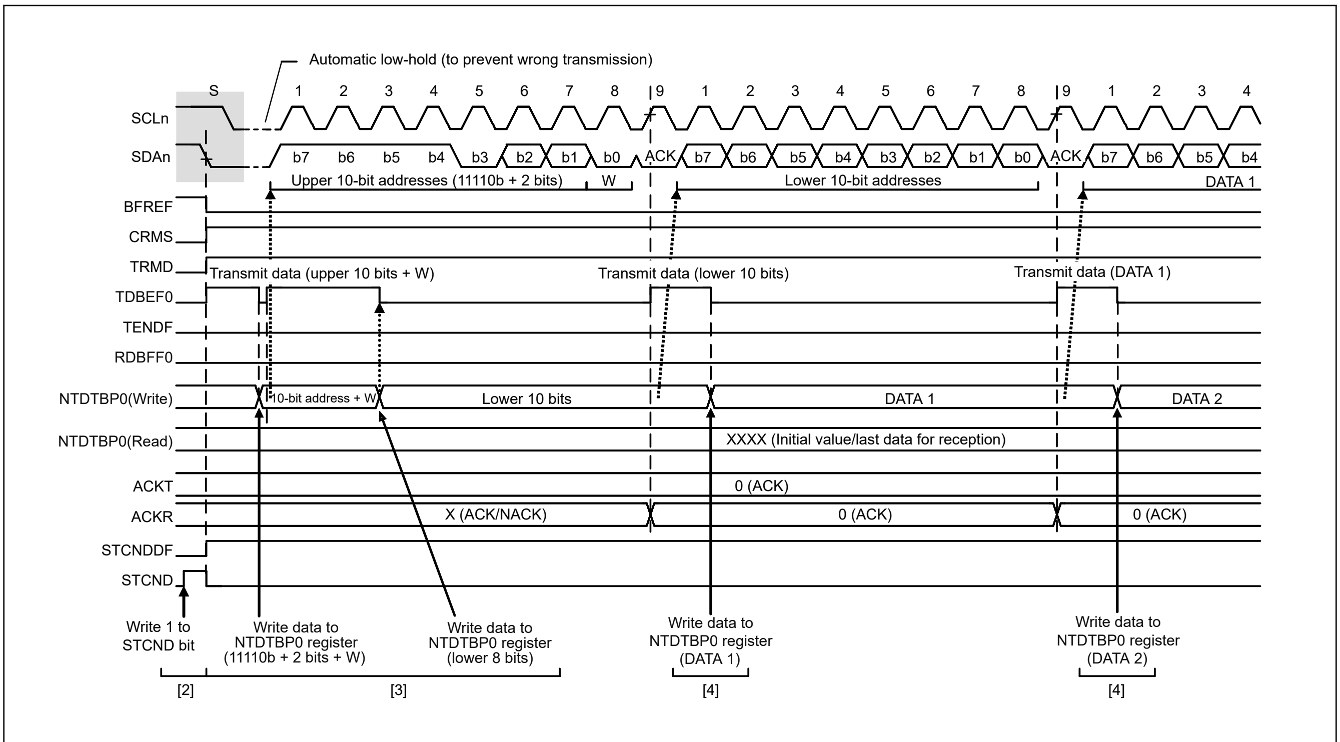


Figure 40.7 Master transmit operation timing (2) (10-bit address format)

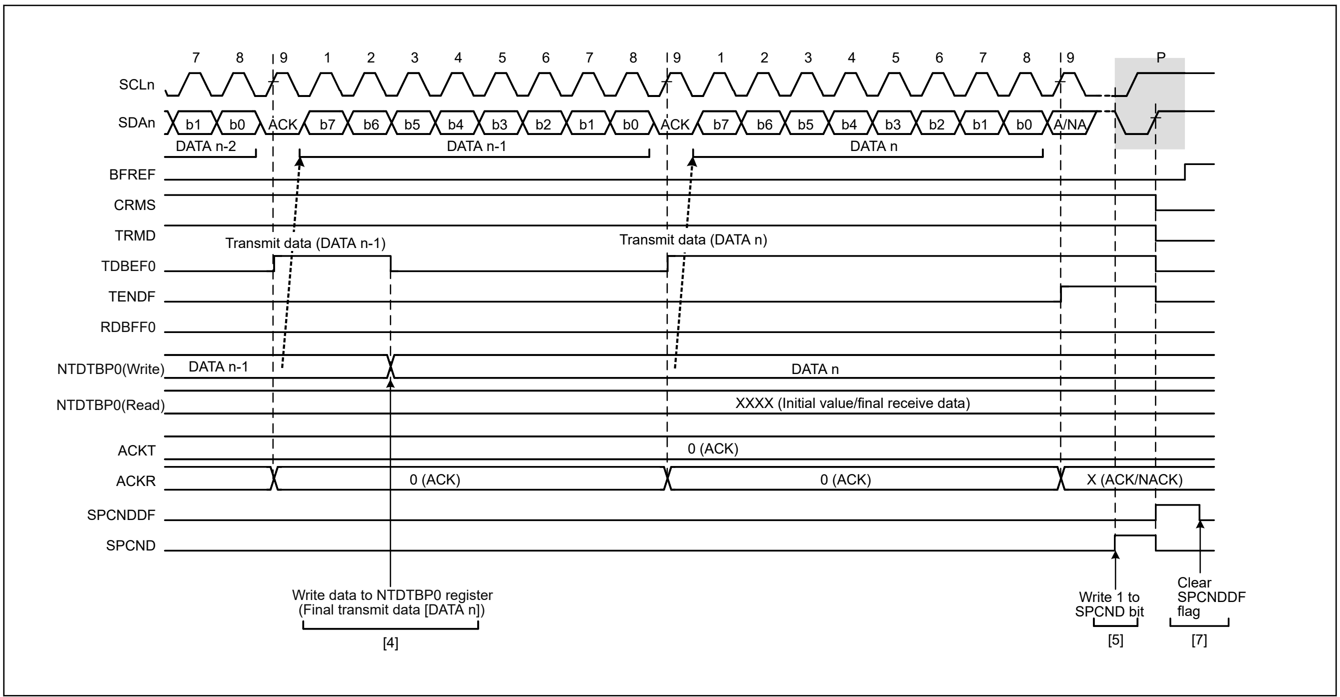


Figure 40.8 Master transmit operation timing (3)

(b) Data Read Transfer (Single Buffer transfer)

In master receive operation, I3C as a master device outputs the SCL clock, receives data from the slave device, and returns acknowledgments. Because I3C must start by sending a slave address to the corresponding slave device, this part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.

Figure 40.144 and Figure 40.145 show examples of usage of master reception (7-bit address format) and Figure 40.9 to Figure 40.11 show the timing of operations in master reception.

The following describes the procedure and operations for master reception.

1. Initial settings. For details, see section 40.3.3.1. Initial Setting Flow.
2. Read the BCST.BFREF flag to check that the bus is open, and then set the CNDCTL.STCND bit to 1 (START condition issuance request). Upon receiving the request, I3C issues a START condition. When I3C detects the START condition, the BFREF flag is automatically set to 0 and the BST.STCNDDF flag is automatically set to 1 and the STCND bit is automatically set to 0. At this time, if the START condition is detected and the levels for the SDA output and the levels on the I3C_SDA line have matched while the STCND bit = 1, I3C recognizes that issuing of the START condition as requested by the STCND bit has been successfully completed, and bits CRMS and TRMD in the PRSST register are automatically set to 1, placing I3C in master transmit mode. The NTST.TDBEF0 flag is also automatically set to 1 in response to setting of the TRMD bit to 1.
3. Check that the NTST.TDBEF0 flag = 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to the NTDTBP0 register. Once the data for transmission are written to the NTDTBP0 register, the TDBEF0 flag is automatically set to 0, the data are transferred from the Normal Tx Data Buffer to the Shift Register, and the TDBEF0 flag is again set to 1. Once the byte containing the slave address and R/W# bit has been transmitted, the value of the PRSST.TRMD bit is automatically updated to select transmit or receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 1, the TRMD bit is set to 0 on the rising edge of the ninth cycle of SCL clock, placing I3C in master receive mode. At this time, the TDBEF0 flag is set to 0. The NTST.RDBFF0 flag is automatically set to 1 when ACK response is received from the slave device. If the slave device is not recognized or a communication failure occurs, the BST.NACKDF flag will be set to 1. At this time, set 1 to the CNDCTL.SPCND bit to issue a STOP condition. For master reception from a device with a 10-bit address, start by using master transmission to issue the 10-bit address, and then issue a Repeated START condition. After that, transmitting 1111 0, the two higher-order bits of the slave address, and the R bit places I3C in master receive mode.
4. Dummy read the NTDTBP0 register after confirming that the NTST.RDBFF0 flag = 1; this makes I3C start output of the SCL clock and start data reception.
5. After 1 byte of data has been received, the NTST.RDBFF0 flag is set to 1 on the rising edge of the eighth or ninth cycle of SCL clock (the clock signal) as selected by the SCSTRCTL.ACKTWE bit. Reading the NTDTBP0 register at

this time will produce the received data, and the RDBFF0 flag is automatically set to 0 at the same time. Furthermore, the value of the acknowledgment field received during the ninth cycle of SCL clock is returned as the value set in the ACKCTL.ACKT bit. Furthermore, if the next byte to be received is the next to last byte, set the SCSTRCTL.RWE bit to 1 (for wait insertion) before reading the NTDTBP0 register (containing the second byte from last). As well as enabling NACK output even in the case of delays in processing to set the ACKCTL.ACKT bit to 1 (NACK) in step 6, due to other interrupts, etc., this fixes the I3C_SCL line to the low level on the falling edge of the ninth clock cycle in reception of the last byte, so the state is such that issuing a STOP condition is possible.

6. When the SCSTRCTL.ACKTWE bit = 0 and the slave device must be notified that it is to end transfer for data reception after transfer of the next (final) byte, set the ACKCTL.ACKT bit to 1 (NACK).
7. After reading the byte before last from the NTDTBP0 register, if the value of the NTST.RDBFF0 flag is confirmed to be 1, write 1 to the CNDCTL.SPCND bit (STOP condition issuance request) and then read the last byte from the NTDTBP0 register. When 1 is written to the CNDCTL.SPCND bit, I3C is released from the wait state and issues the STOP condition after low-level output in the ninth clock cycle is completed or the I3C_SCL line is released from the low-hold state.
8. Upon detecting the STOP condition, I3C automatically sets bits CRMS and TRMD in the PRSST register to 00 and enters slave receive mode. Furthermore, detection of the STOP condition leads to setting of the BST.SPCNDDF flag to 1.
9. After checking that the BST.SPCNDDF flag = 1, set the BST.NACKDF and SPCNDDF flags to 0 for the next transfer operation.

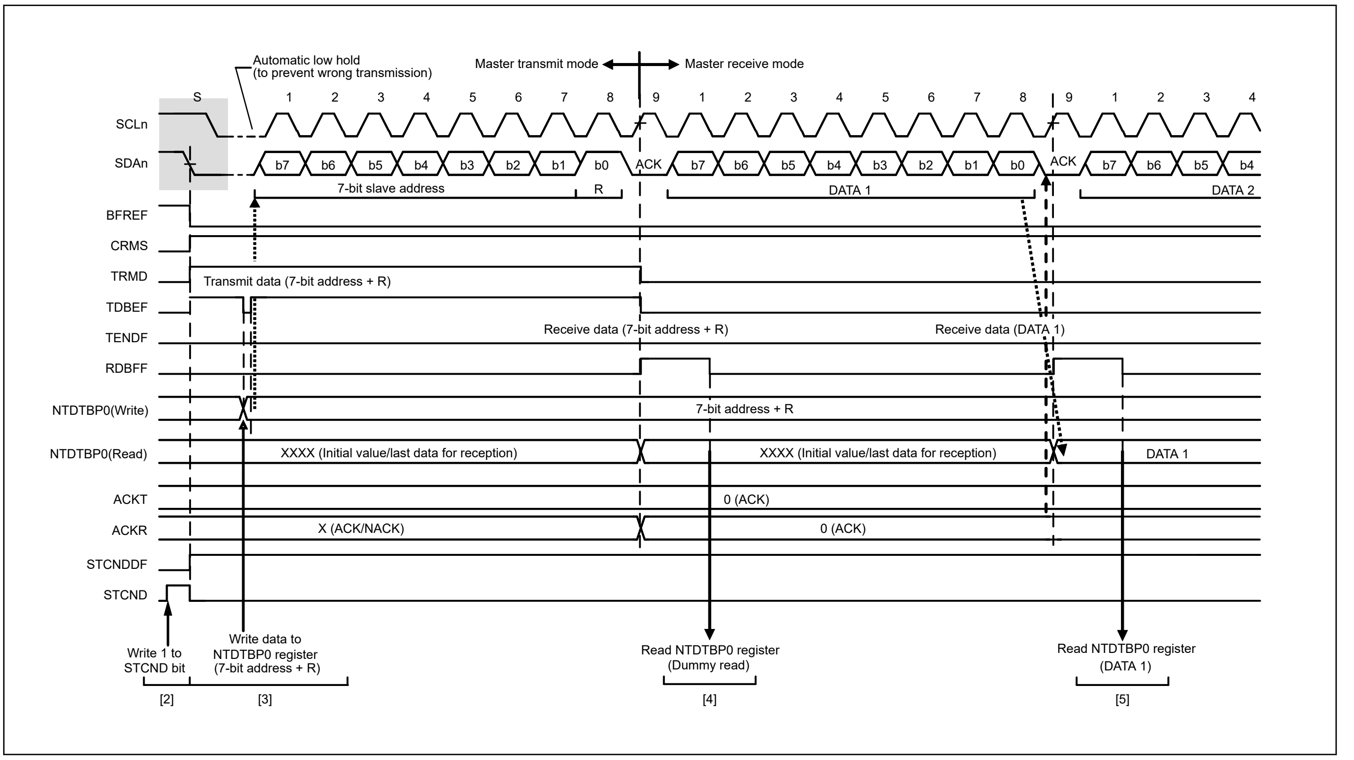


Figure 40.9 Master receive operation timing (1) (7-bit address format, when ACKTWE = 0)

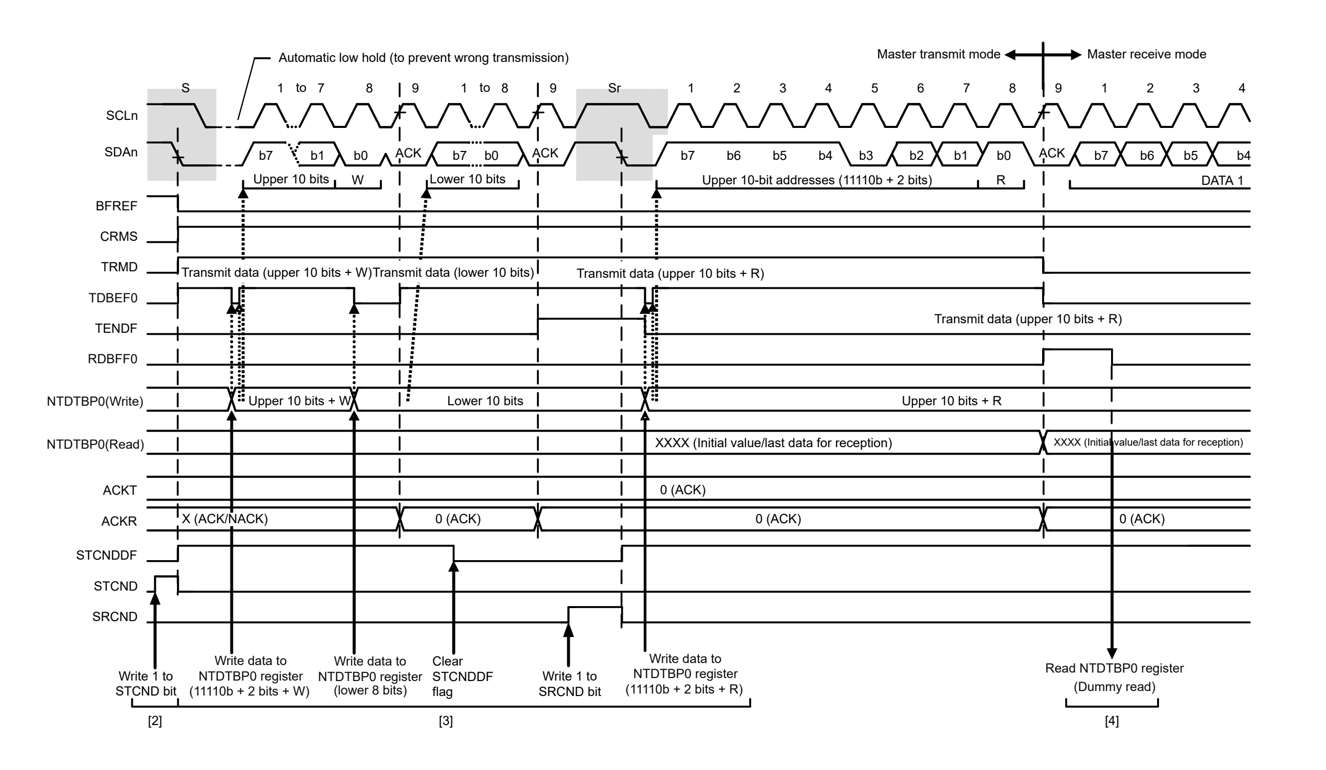


Figure 40.10 Master receive operation timing (2) (10-bit address format, when ACKTWE = 0)

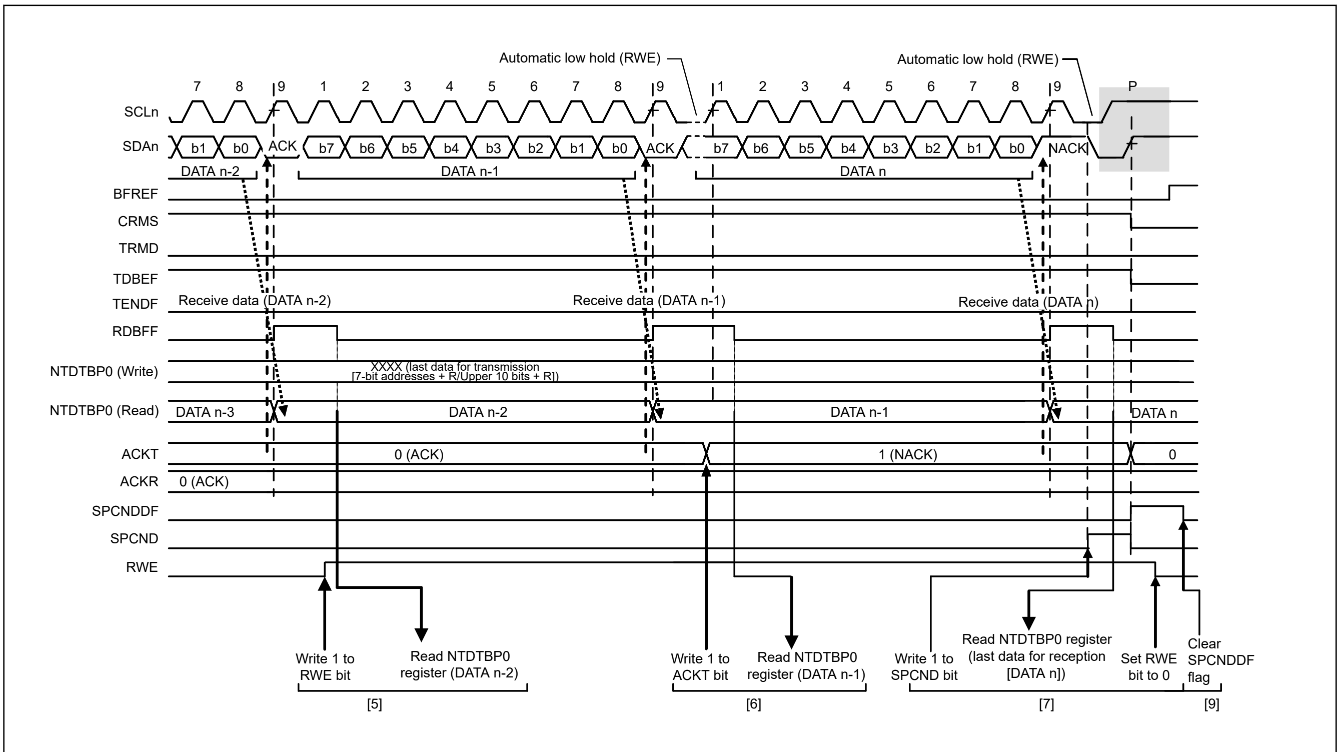


Figure 40.11 Master receive operation timing (3) (when ACKTWE = 0)

(2) I3C Master Operation

(a) Dynamic Address Assign Procedure

After initializing I3C, first execute Dynamic Address Assign Procedure for I3C Slave connected on the I3C Bus. The following describes the procedure.

1. Initial setting (see section 40.3.3.1.2. I3C Initial Setting Flow for details)
2. Execute Dynamic Address Assign with ENTDAA or SETDASA Common Command Code (CCC) for I3C Slave set in DAT (DATBASm register).
   Write Command Descriptor (Address Assign Command) to Command Buffer via the NCMDQP register.
3. When Command Descriptor is written to Command Buffer, Transaction is issued on I3C Bus.
4. When ENTDAA is specified for CMD[7:0] of Address Assign Command:
   Execute Dynamic Address Assign for I3C Slave for the number of DATs specified by DEV_COUNT[3:0] starting with DAT specified by DEV_INDEX[4:0] of Address Assign Command.
   When SETDASA is specified for CMD[7:0] of Address Assign Command:
   Execute Dynamic Address Assign for I3C Slave indicated by DAT specified by DEV_INDEX[4:0] of Address Assign Command.
5. In case of ENTDAA, the Provisional ID, BCR, DCR transmitted from I3C Slave is stored in Receive Data Buffer (BCR is also automatically stored in the MSDCTm register).
   Read the Provisional ID, BCR, and DCR from the Receive Data Buffer via the NTDTBPn register with an interrupt by RDBFF0 = 1.
6. When execution of Dynamic Address Assign is completed, issue STOP condition and store the Response Descriptor into the Response Buffer.
7. Read the Response Descriptor via the NRSPQP register and check the status.
8. Check whether the value of the DATA_LENGTH[15:0] bits of the Response Descriptor matches the value of DEV_COUNT[3:0] of the Address Assign Command.

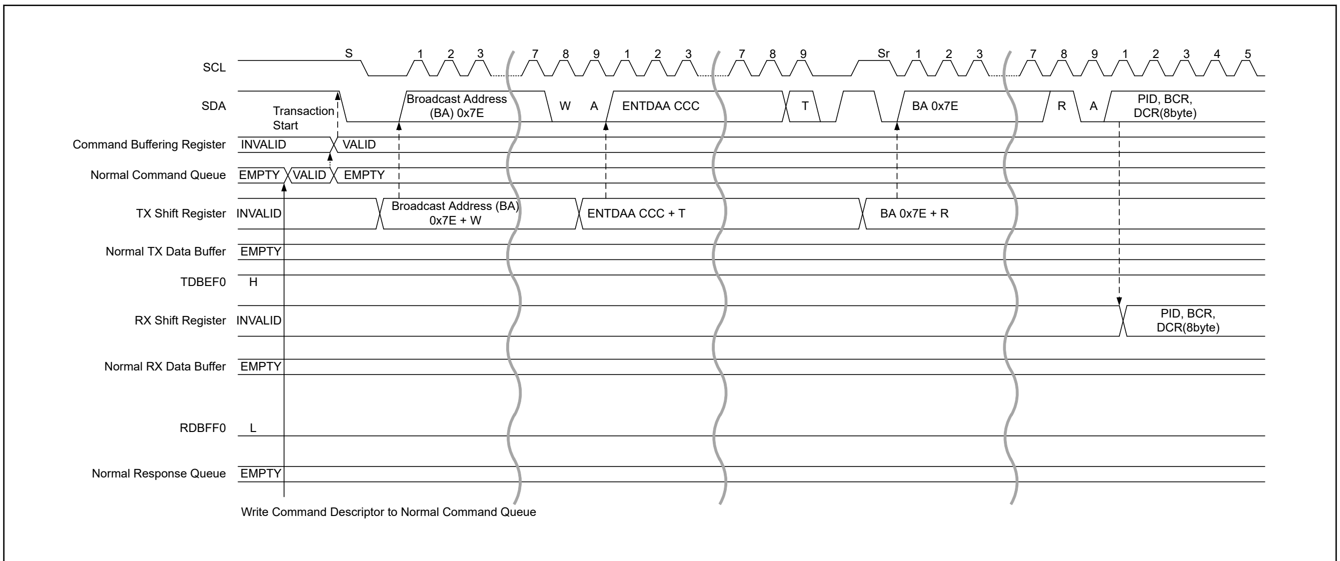


Figure 40.12 Dynamic address assign procedure (ENTDAA CCC) timing (1/3)

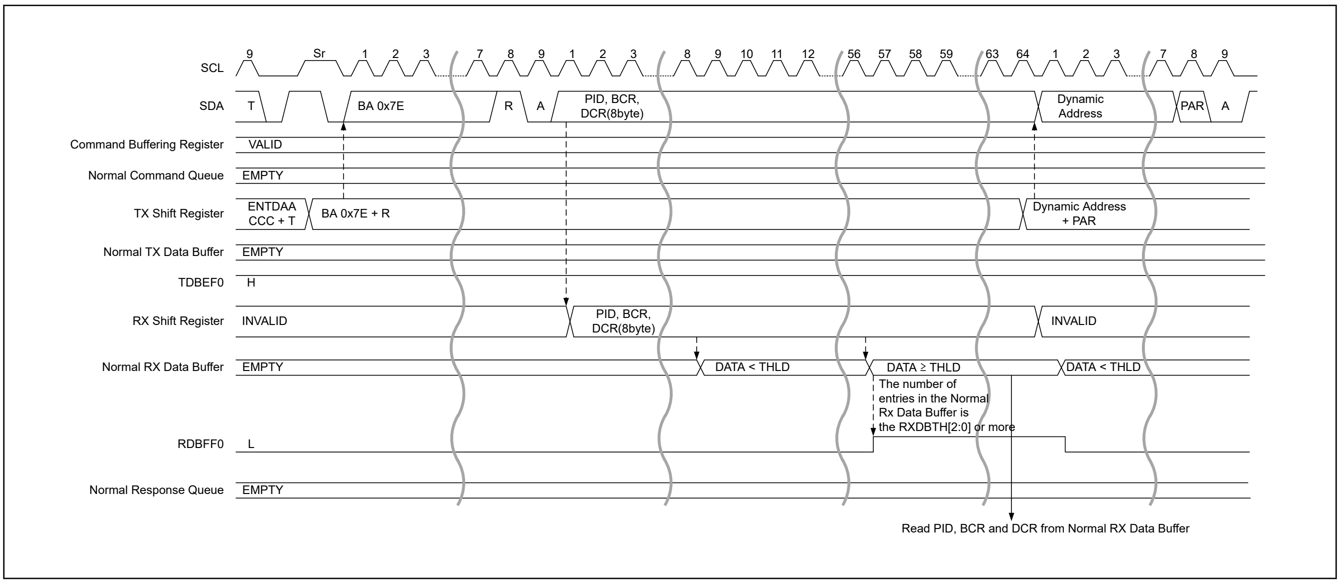


Figure 40.13 Dynamic address assign procedure (ENTDAA CCC) timing (2/3)

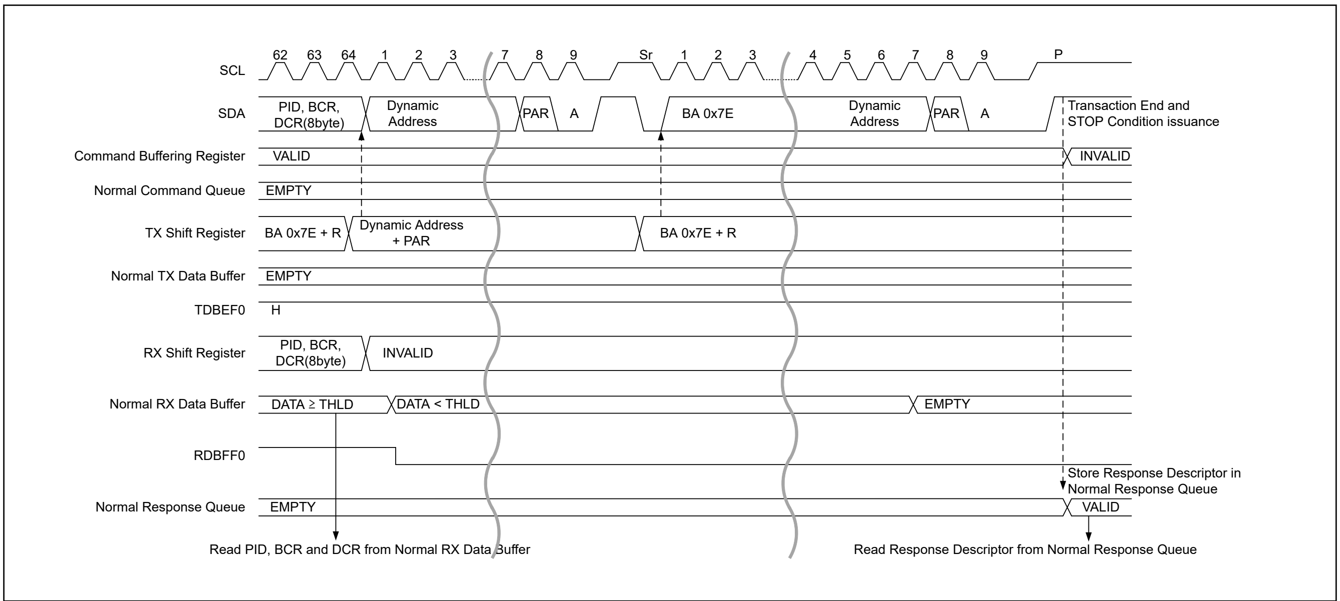


Figure 40.14 Dynamic address assign procedure (ENTDAA CCC) timing (3/3)

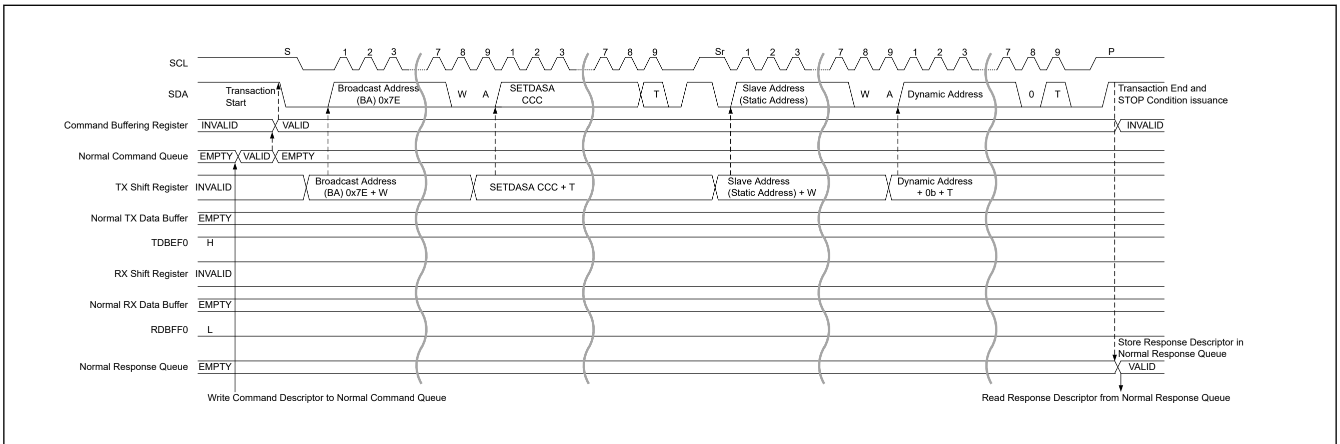


Figure 40.15 Dynamic address assign procedure (SETDASA CCC) timing

(b) SDR Data Write Transfer

1. Write data for transmission to the Tx Data Buffer via the NTDTBPn register.
2. Write a Command Descriptor (Immediate Transfer Command or Regular Transfer Command or Combo Transfer Command) for Data Transfer to the Command Buffer via the NCMDQP register.
3. When Command Descriptor is written to Command Buffer, transaction is issued on I3C Bus.
When NACK is received with the Address Header, transaction of the same command is automatically issued according to the NACK Retry Count value (DATBASm.DVNACK) of DAT.
4. If data for transmission still remain, write data for transmission by an interrupt with TDBEF0 = 1 to the Tx Data Buffer via the NTDTBPn register.
5. When data transmission for the number of Data Length specified by the DATA_LENGTH[15:0] bits of the Command Descriptor is completed, the Repeated START condition or STOP condition is issued and the Response Descriptor is stored in the Response Buffer.
6. Read the Response Descriptor via the NRSPQP register and check the status.
7. Check that the value of the DATA_LENGTH[15:0] bits of the Response Descriptor is 0.

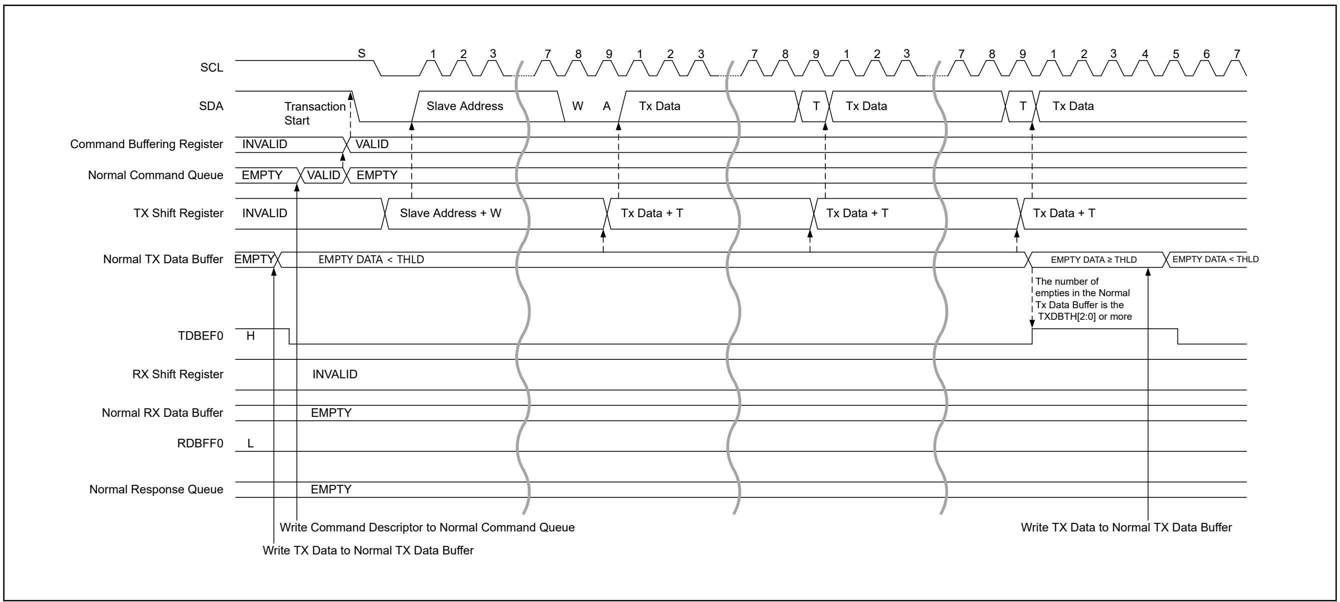


Figure 40.16 SDR data write transfer timing (1/2)

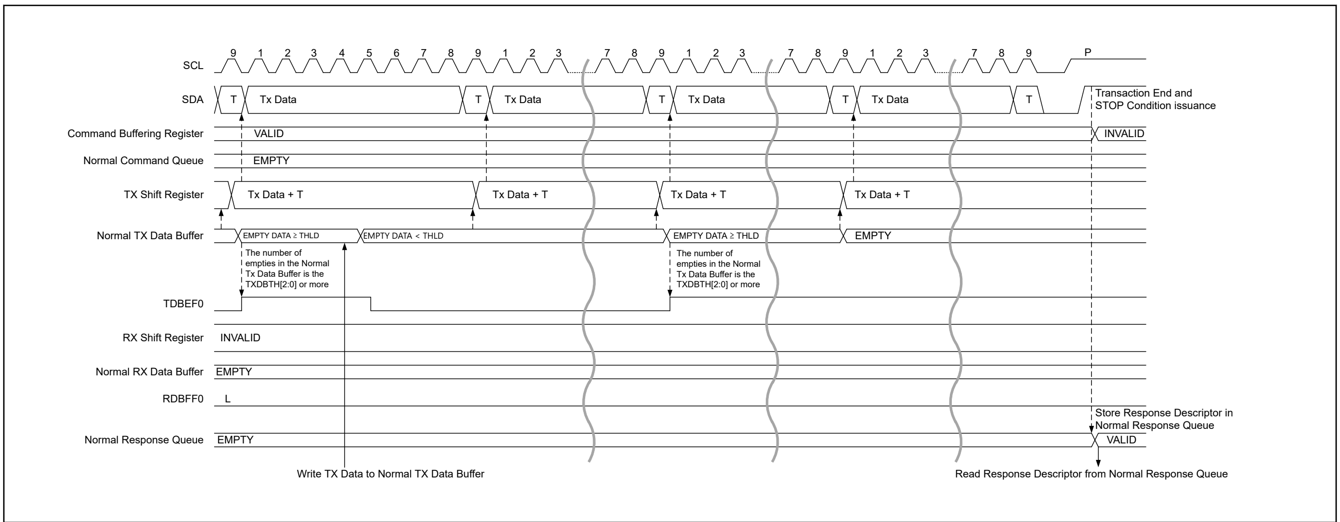


Figure 40.17 SDR data write transfer timing (2/2)

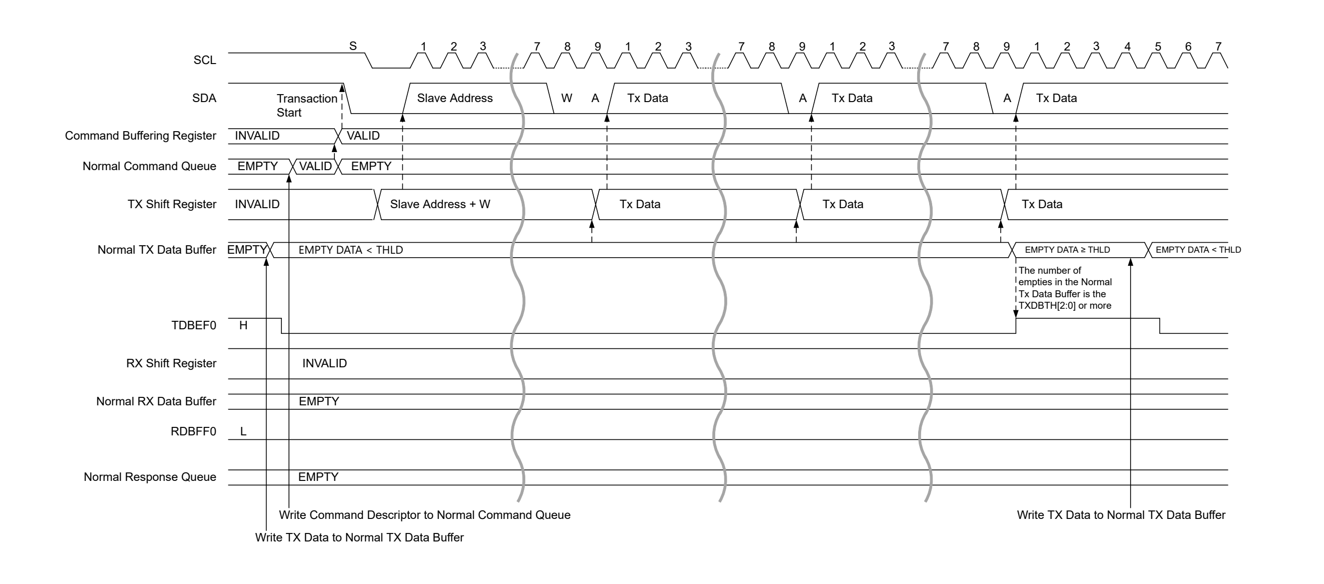


Figure 40.18 Legacy I²C message data write timing (1/2)

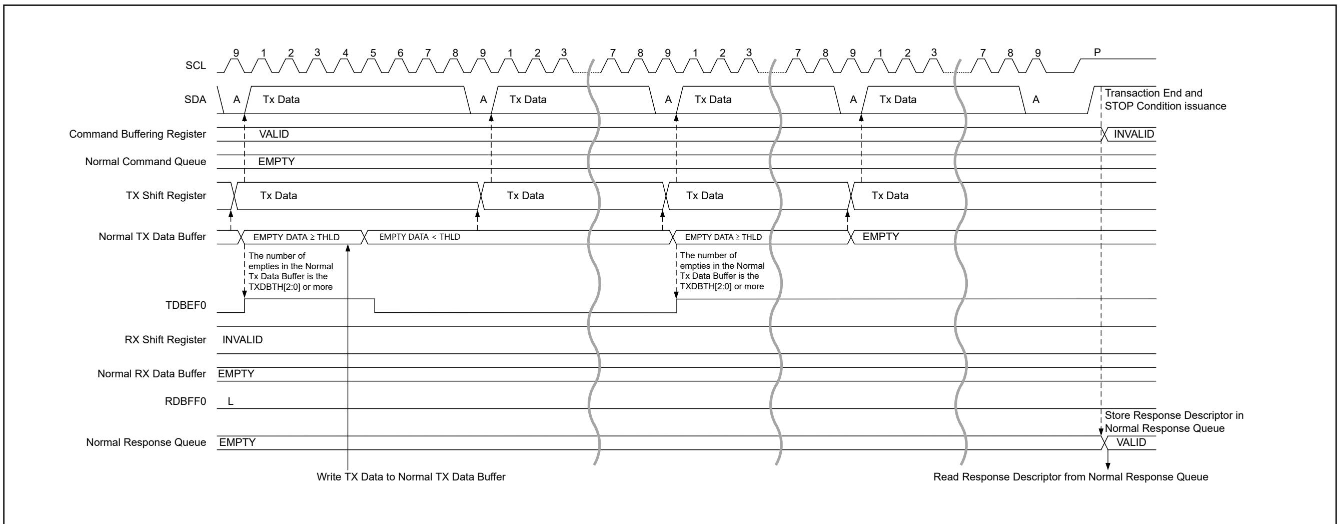


Figure 40.19 Legacy I²C message data write timing (2/2)

(c) SDR Data Read Transfer

1. Write a Command Descriptor (Immediate Transfer Command or Regular Transfer Command or Combo Transfer Command) for Data Transfer to the Command Buffer via the NCMDQP register.
2. When Command Descriptor is written to Command Buffer, transaction is issued on I3C Bus.
   When NACK is received with the Address Header, transaction of the same command is automatically issued according to the NACK Retry Count value (DATBASm.DVNACK) of DAT.
3. Data received from the I3C Slave is stored in the Receive Data Buffer.
4. With the RDBFF0 = 1 interrupt, the received data is read from the Receive Data Buffer via the NTDTBPn register.
5. SDR:
   Detecting Low in T-bit or receiving Data for the number of Data Length specified by the DATA_LENGTH[15:0] bits of Command Descriptor is completed, issue Repeated START condition or STOP condition and store the Response Descriptor into the Response Buffer.
   Legacy I²C Message:
   When data reception for the number of Data Length specified by the DATA_LENGTH[15:0] bits of Command Descriptor is completed, NACK is issued. After that, issue a Repeated START condition or STOP condition and store the Response Descriptor into the Response Buffer.
6. Read the Response Descriptor via the NRSPQP register and check the status.

7. Check whether the value of the DATA_LENGTH[15:0] bits of the Response Descriptor matches the data length setting value of the Command Descriptor.

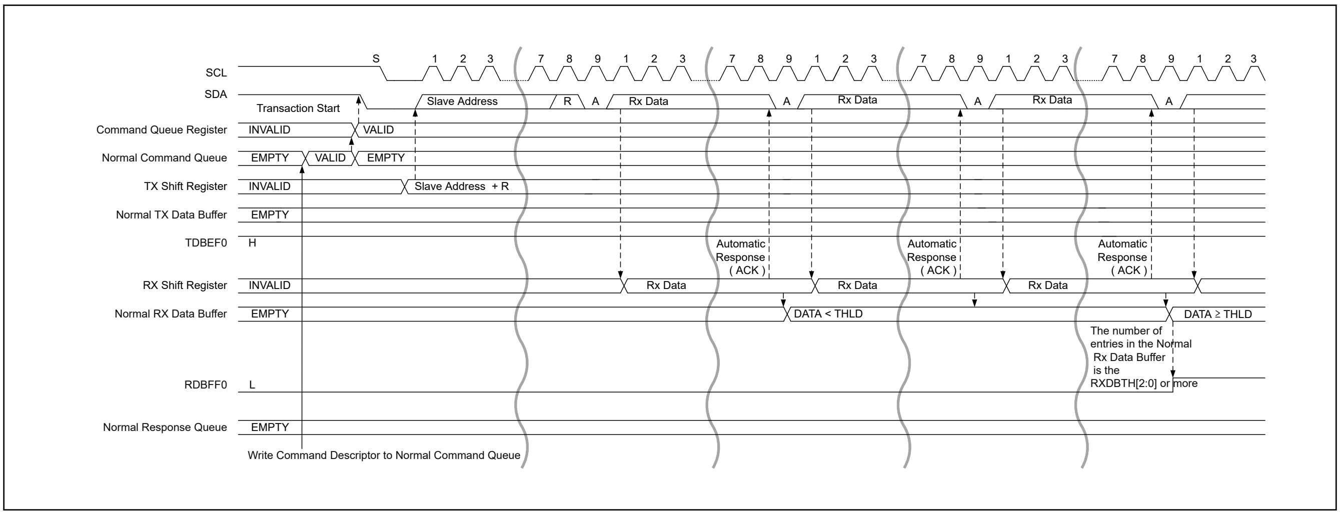


Figure 40.20 SDR data read transfer timing (1/2)

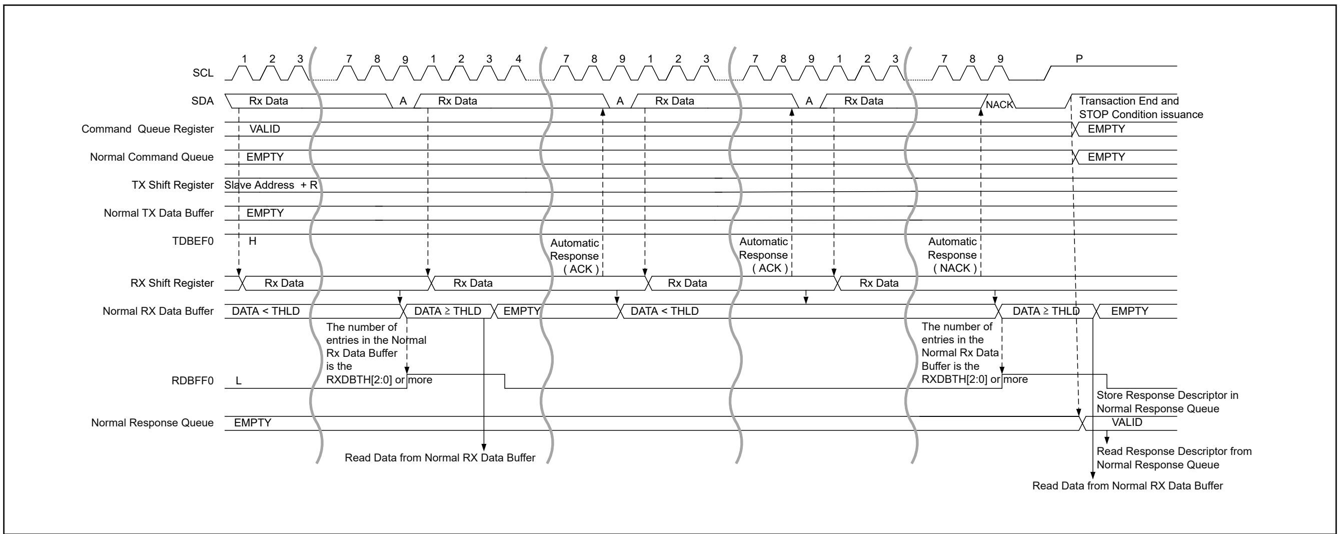


Figure 40.21 SDR data read transfer timing (2/2)

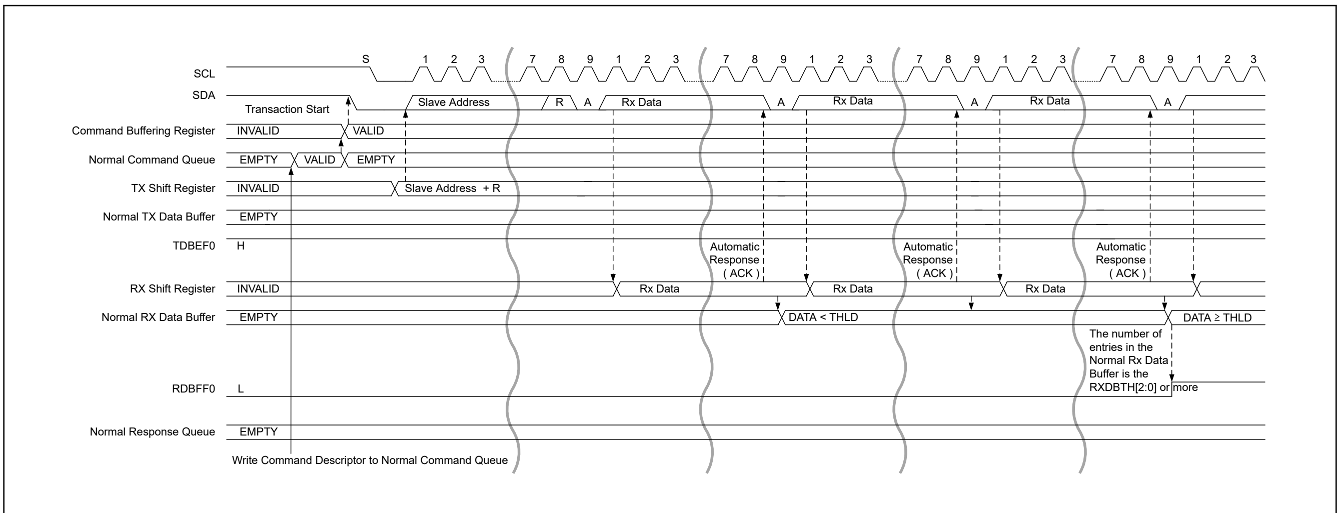


Figure 40.22 Legacy I^2C message data read transfer timing (1/2)

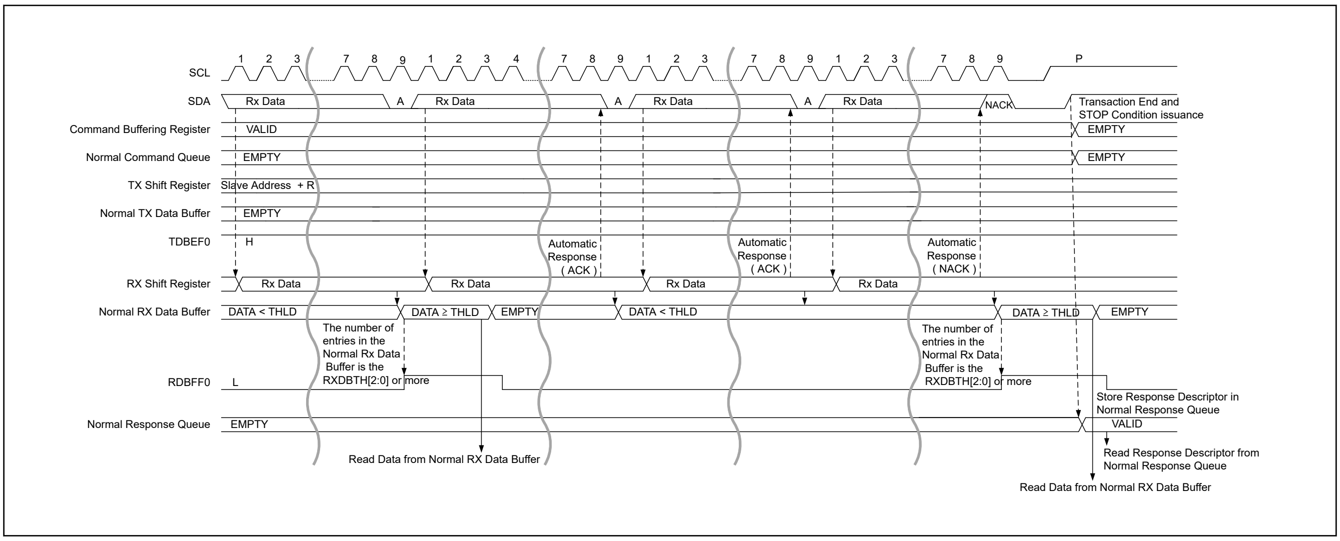


Figure 40.23 Legacy I²C message data read transfer timing (2/2)

(d) HDR Data Write Transfer

1. Write data for transmission to the Tx Data Buffer via the NTDTBPn register.
2. Write a Command Descriptor (Immediate Transfer Command or Regular Transfer Command or Combo Transfer Command) for Data Transfer to the Command Buffer via the NCMDQP register.
3. When Command Descriptor is written to Command Buffer, transaction is issued on I3C Bus.
   When NACK is received with the Address Header, transaction of the same command is automatically issued according to the NACK Retry Count value (DATBASm.DVNACK) of DAT.
4. If data for transmission still remains, write data for transmission by an interrupt with TDBEF0 = 1 to the Tx Data Buffer via the NTDTBPn register.
5. HDR-DDR:
   When the data transmission for the number of Data Length specified by the DATA_LENGTH[15:0] bits of the Command Descriptor is completed, CRC Word is transmitted. After that, issue HDR Restart Pattern or HDR Exit Pattern + STOP condition and store the Response Descriptor into Response Buffer.
   HDR-Ternary:
   When the data transmission for the number of Data Length specified by the DATA_LENGTH[15:0] bits of the Command Descriptor is completed, the HDR Restart Pattern or HDR Exit Pattern + STOP condition is issued and the Response Descriptor is stored in the Response Buffer.
6. Read the Response Descriptor via the NRSPQP register and check the status.
7. Check that the value of the DATA_LENGTH[15:0] bits of the Response Descriptor is 0.

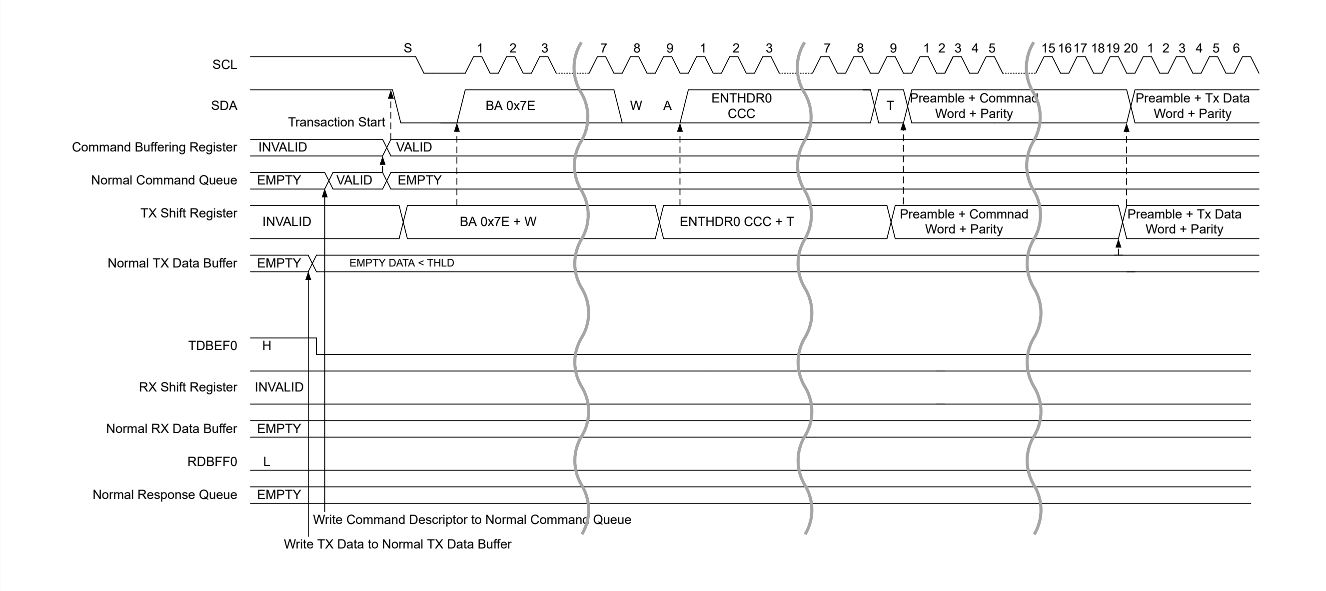


Figure 40.24 HDR data write transfer (HDR-DDR) timing (1/4)

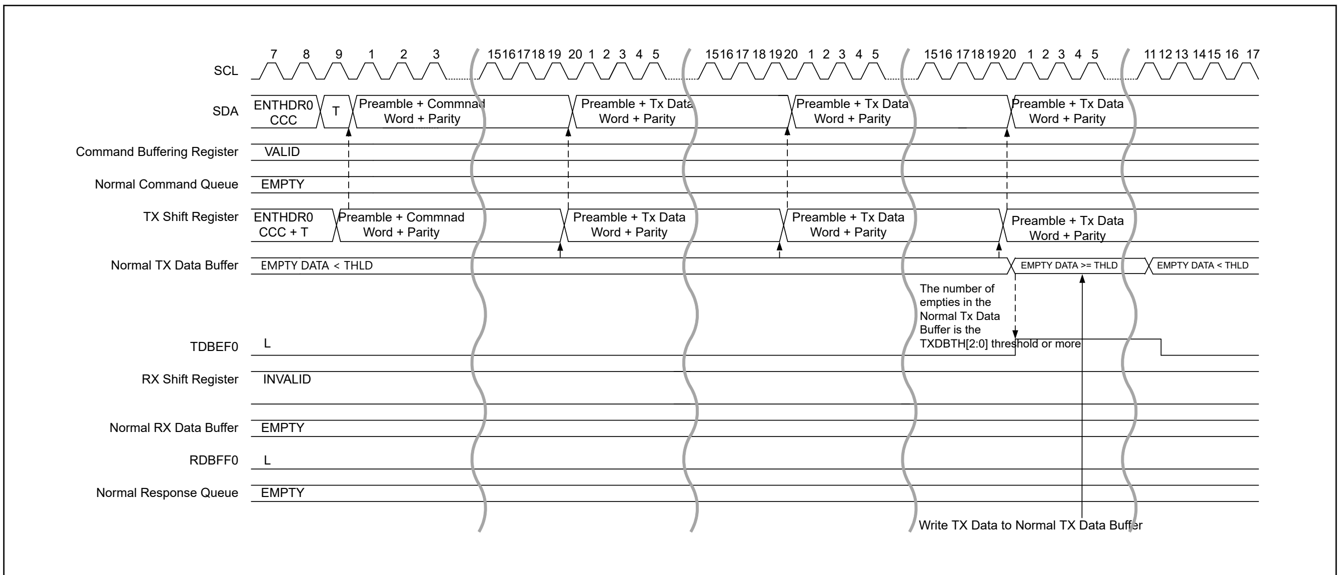


Figure 40.25 HDR data write transfer (HDR-DDR) timing (2/4)

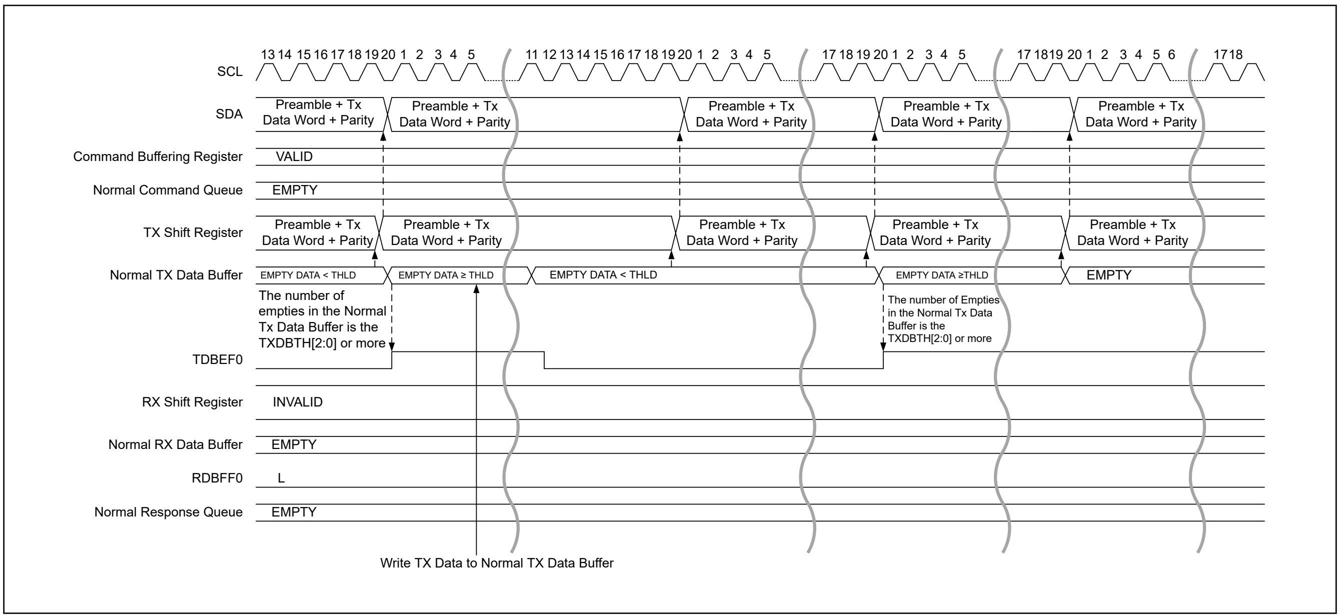


Figure 40.26 HDR data write transfer (HDR-DDR) timing (3/4)

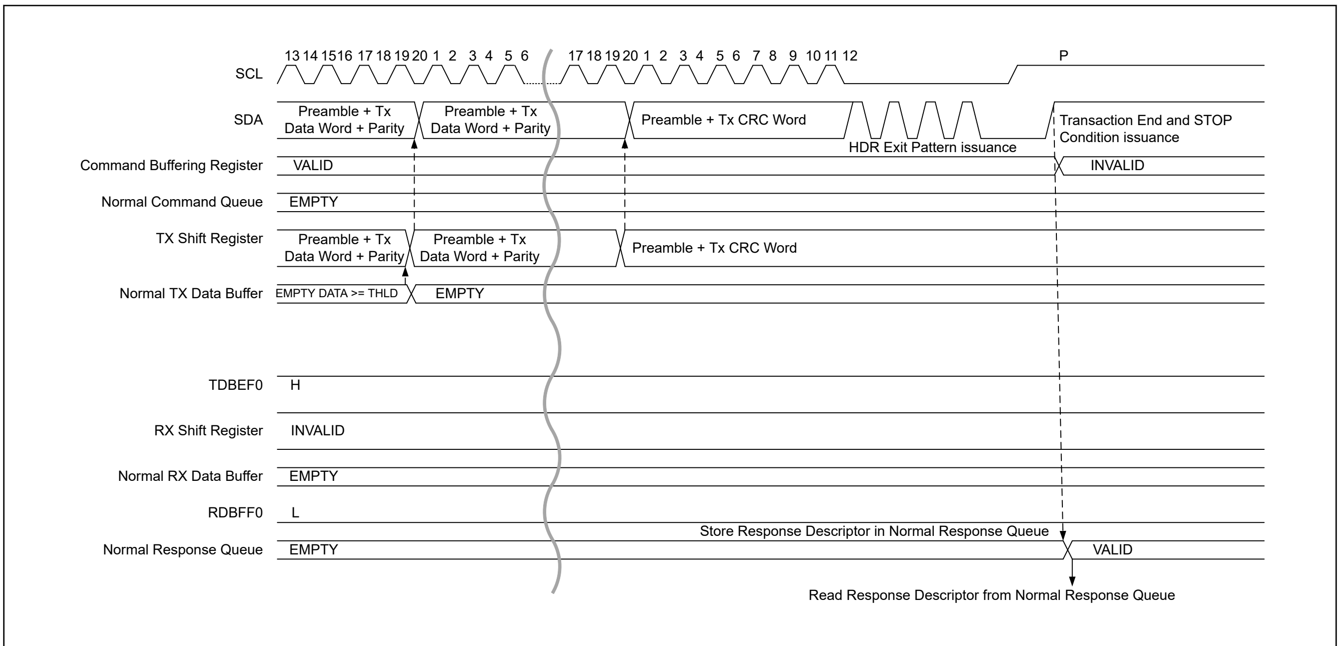


Figure 40.27 HDR data write transfer (HDR-DDR) timing (4/4)

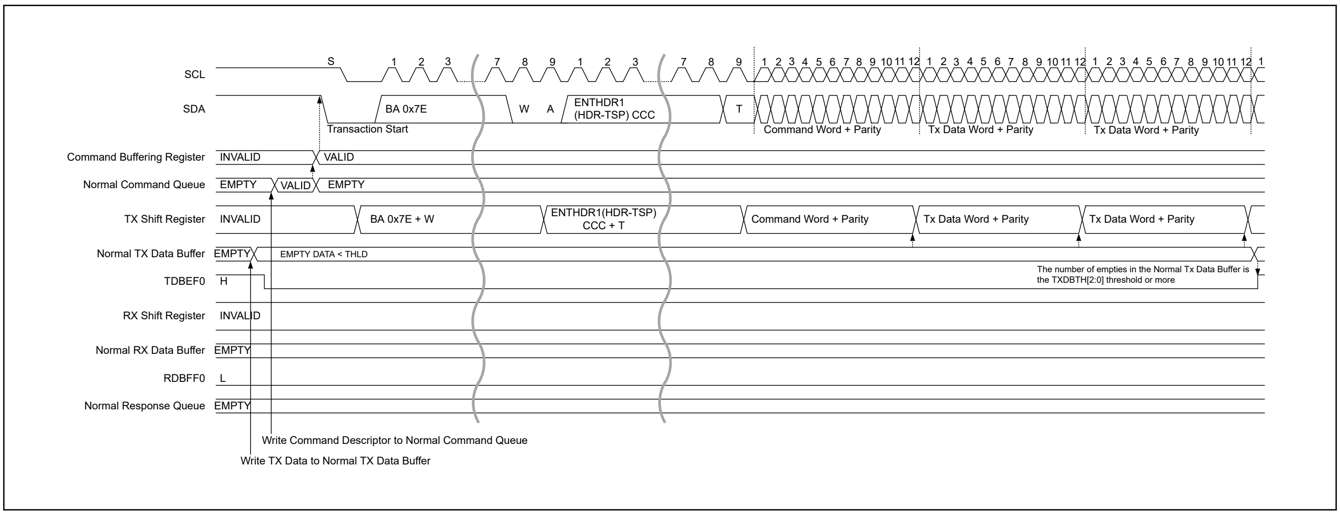


Figure 40.28 HDR data write transfer (HDR-TSP, TSL) timing (1/2)

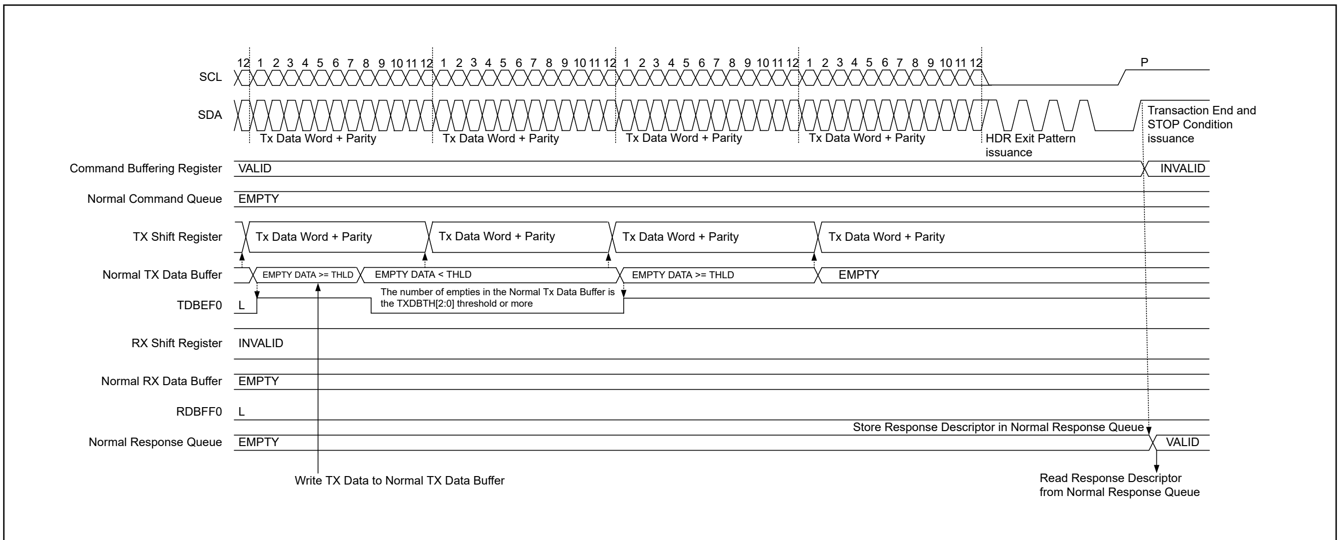


Figure 40.29 HDR data write transfer (HDR-TSP, TSL) timing (2/2)

(e) HDR Data Read Transfer

1. Write a Command Descriptor (Immediate Transfer Command or Regular Transfer Command or Combo Transfer Command) for Data Transfer to the Command Buffer via the NCMDQP register.
2. When Command Descriptor is written to Command Buffer, transaction is issued on I3C Bus.
   When NACK is received with the Address Header, transaction of the same command is automatically issued according to the NACK Retry Count value (DATBASm.DVNACK) of DAT.
3. Data received from the I3C Slave is stored in the Receive Data Buffer.
4. With the RDBFF0 = 1 interrupt, the received data is read from the Receive Data Buffer via the NTDTBPn register.
5. HDR-DDR:
   (a) When CRC Word is received, it issues HDR Restart Pattern or HDR Exit Pattern + STOP condition and stores the Response Descriptor into the Response Buffer.
   (b) When the preamble following the received data for the number of Data Length specified by the DATA_LENGTH[15:0] bits of Command Descriptor indicates Data Word (11), drive (Abort) low to the second bit of the preamble. After that, issue HDR Restart Pattern or HDR Exit Pattern + STOP condition and store the Response Descriptor into the Response Buffer.

   HDR-Ternary:
   When detecting the start of the HDR Restart Pattern or HDR Exit Pattern from I3C Slave, issue HDR Restart Pattern or HDR Exit Pattern + STOP condition and store the Response Descriptor into the Response Buffer.

6. Read the Response Descriptor via the NRSPQP register and check the status.
7. Check whether the value of the DATA_LENGTH[15:0] bits of the Response Descriptor matches the data length setting value of the Command Descriptor.

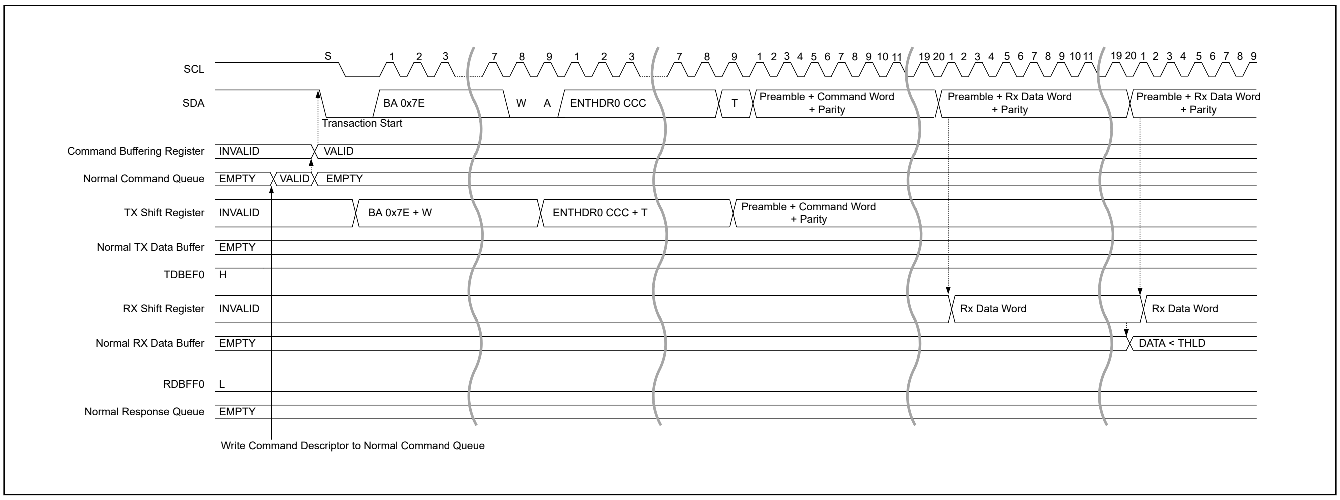


Figure 40.30 HDR data read transfer (HDR-DDR) timing (1/2)

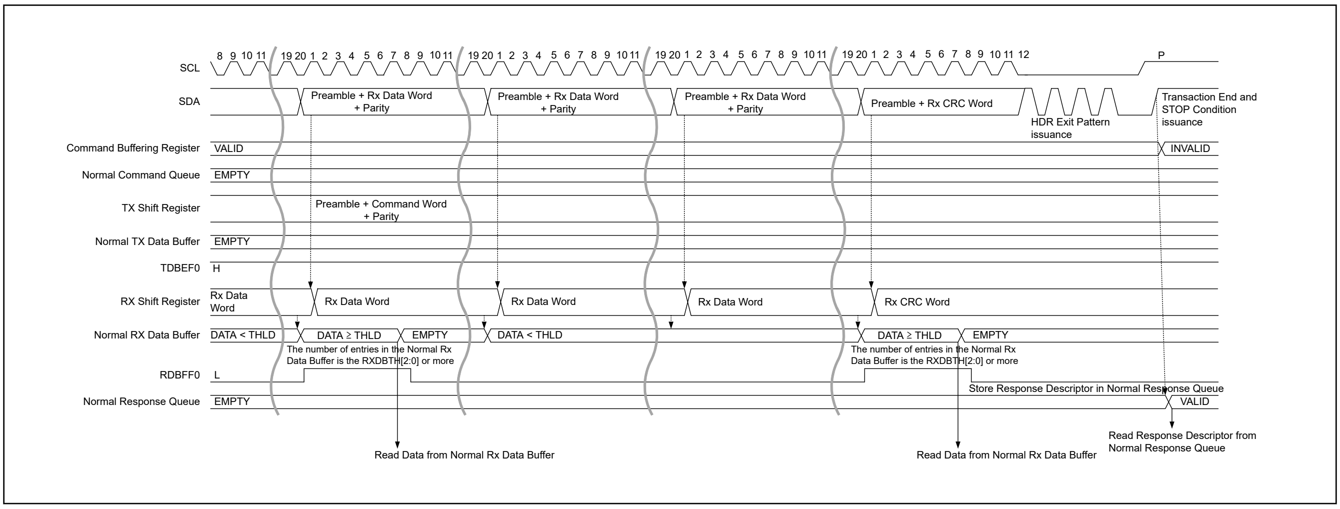


Figure 40.31 HDR data read transfer (HDR-DDR) timing (2/2)

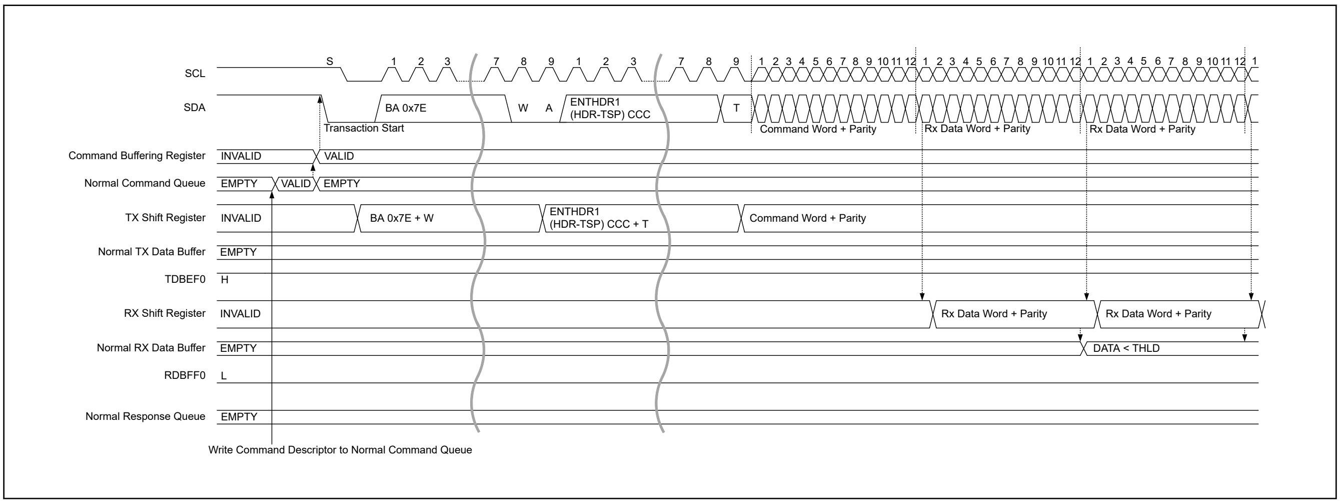


Figure 40.32 HDR data read transfer (HDR-TSP, TSL) timing (1/2)

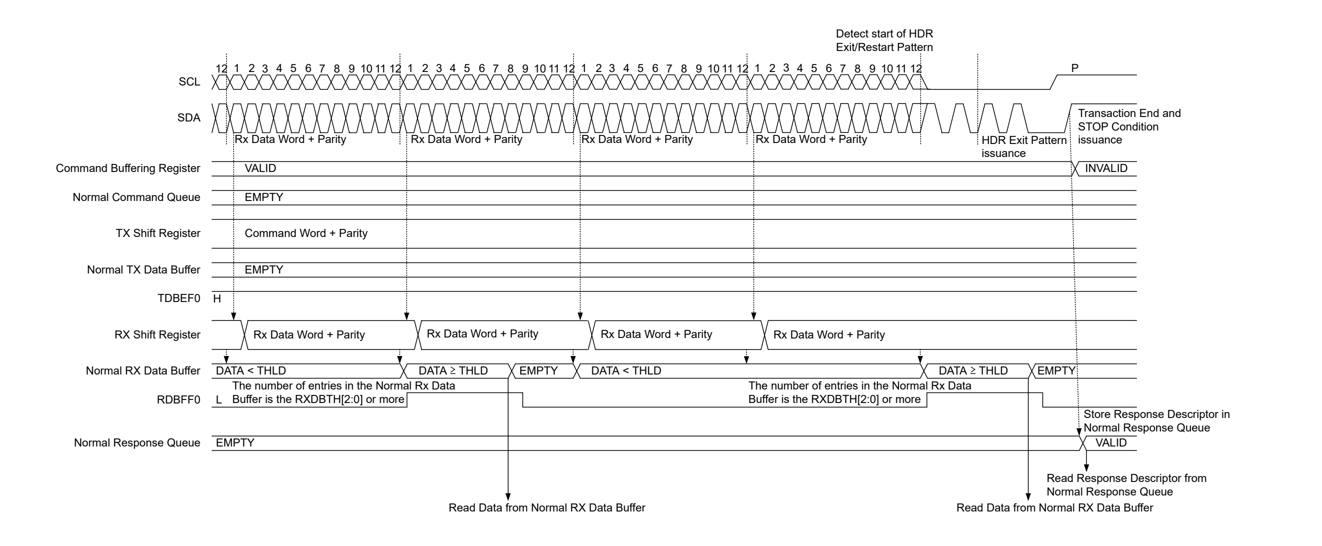


Figure 40.33 HDR data read transfer (HDR-TSP, TSL) timing (2/2)

(f) IBI Transfer

1. Write Command Descriptor to the Command Buffer and issue Transaction on I3C Bus.
   If START Request (SDA Low Drive) is issued from the slave device, I3C drives SCL to Low and completes START condition.
   Thereafter, the SCL is supplied and In-Band Interrupt Request is received.
2. In Slave Address with RnW of the Address Header, if losing Arbitration by issuing In-Band Interrupt from I3C Slave, stop issuing Transaction.
3. According to section 40.3.2.3.8. In-Band Interrupt [I3C Mode], detect In-Band Interrupt and process.
4. In the interrupt with IBIQEFF = 1, read the IBI Status Descriptor from the IBI Queue via the NIBIQP register and check the status.
   When detected a Slave Interrupt Request and responded with ACK, Read the IBI Data for the Data Length indicated by the DATA_LENGTH[15:0] bits of the IBI Status Descriptor from the IBI Data Buffer via the NIBIQP register.
5. Restart issuing Transaction of Command of Step1.

An example of the processing procedure after detection of In-Band Interrupt is shown below.

Processing procedure for detecting Mastership Request and transferring master right to Secondary Master

1. If the I3C Secondary Master wins the Arbitration, issue a DEFSLVS CCC and notify Slave information to Secondary Master.
2. Issue a GETACCMST CCC and complete CCC by a STOP condition.

The Mastership processing flow is shown in Figure 40.36

Note:
- After transferring master right to Secondary Master, to get master right again, issue a Mastership Request according to (f) IBI Transfer of (2) I3C Slave Operation.
- After Mastership Request is accepted by the Current Master, to get master right again at receiving the GETACCMST CCC and complete CCC by a STOP condition.

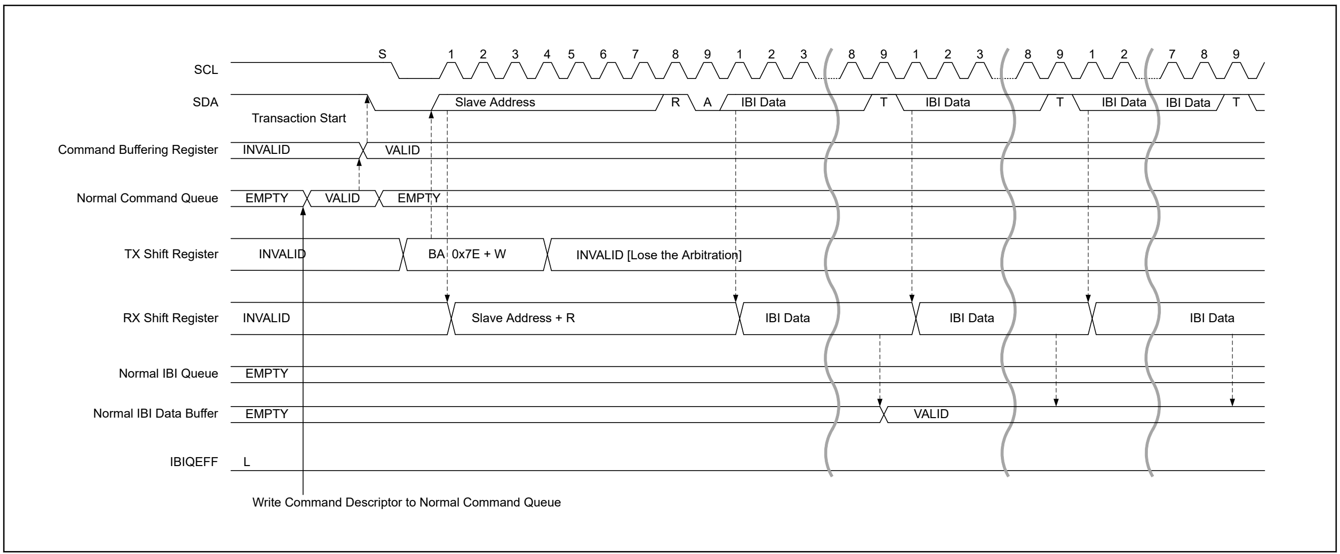


Figure 40.34 I3C master IBI transfer timing (1/2)

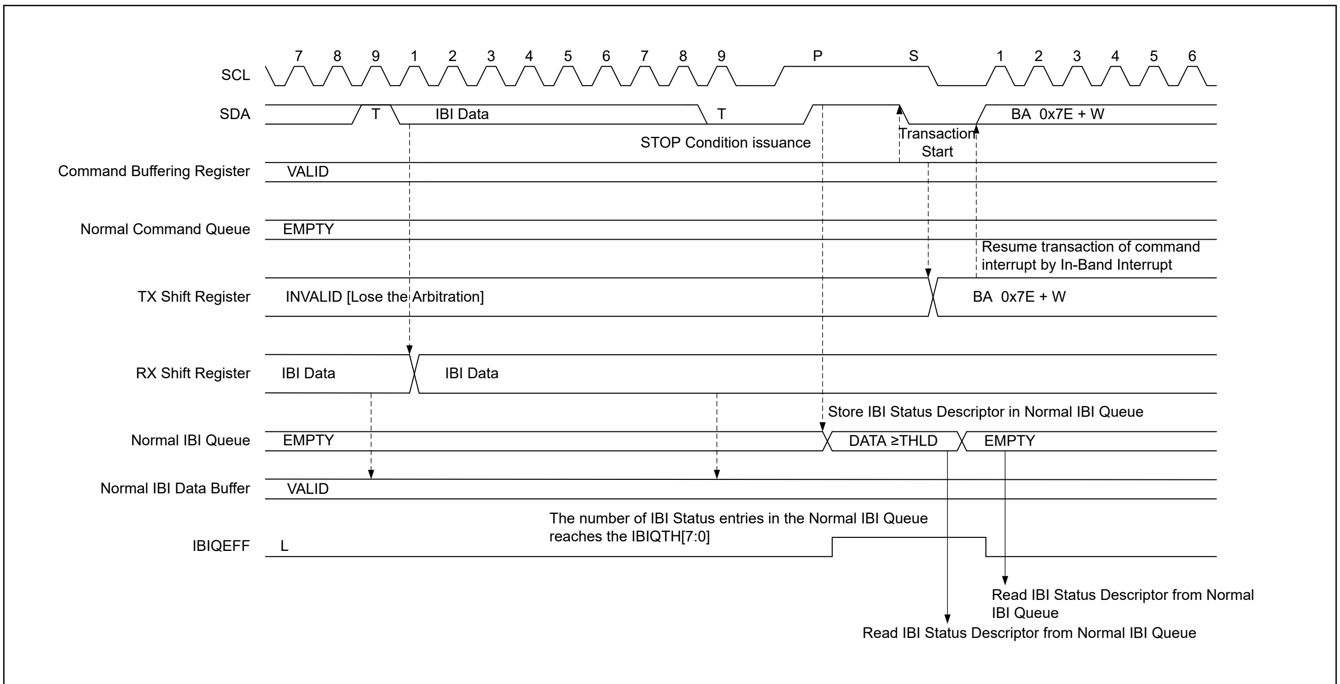


Figure 40.35 I3C master IBI transfer timing (2/2)

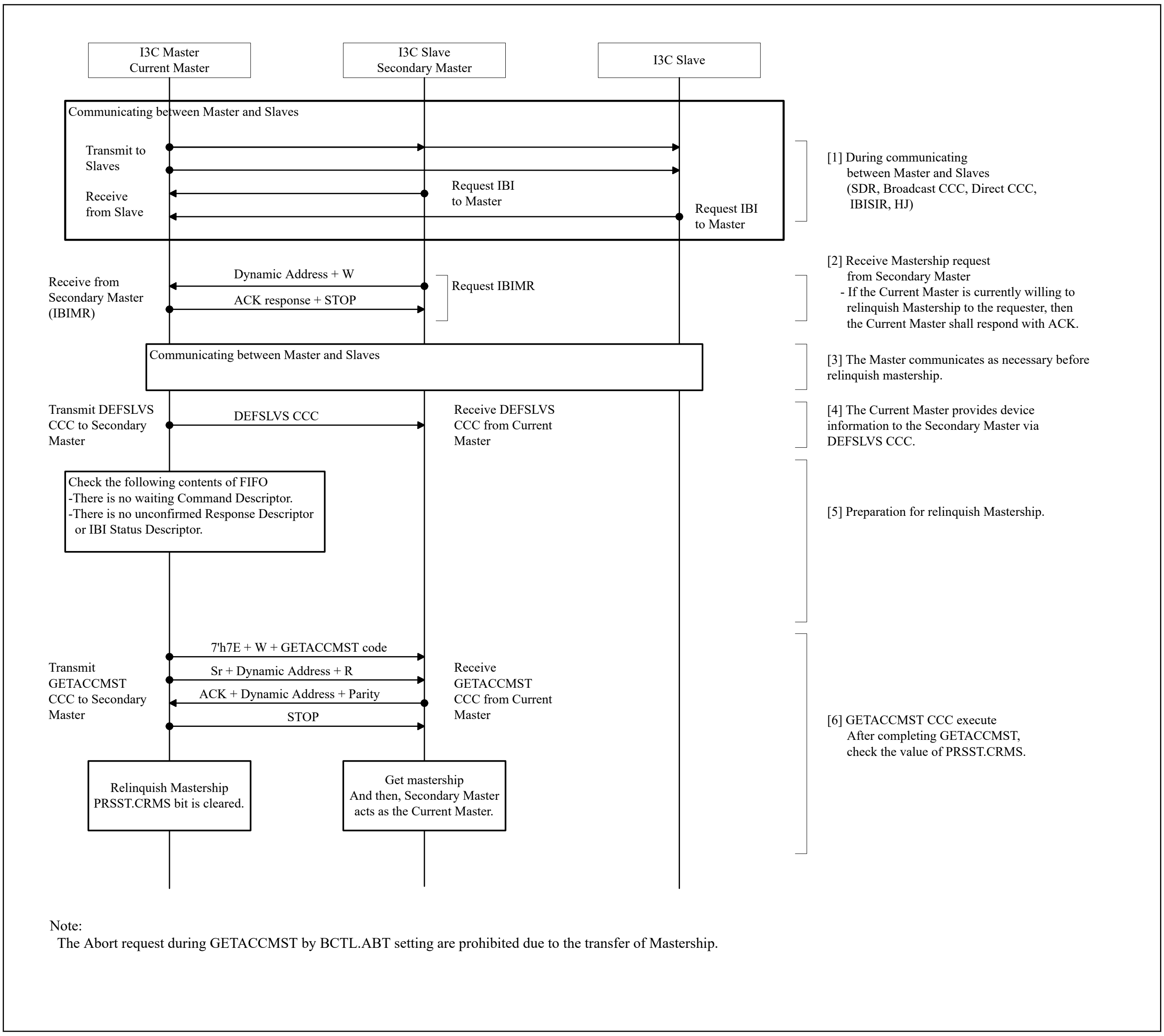


Figure 40.36 I3C Master Mastership processing flow

40.3.2.1.2 Slave Mode Operation

(1) I2C Slave Operation

(a) Data Write Transfer (Single Buffer transfer)

In slave receive operation, the master device outputs the SCL clock and transmit data, and I3C returns acknowledgments as a slave device.

Figure 40.153 shows an example of usage of slave reception and Figure 40.37 and Figure 40.38 show the timing of operations in slave reception.

The following describes the procedure and operations for slave reception.

1. Initial settings. For details, see section 40.3.3.1. Initial Setting Flow. After initial settings, I3C will stay in the standby state until it receives a slave address that it matches.
2. After receiving a matching slave address, I3C sets one of the corresponding bits SVST.HOAF, GCAF, and SVAF[n] (n = 0 to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and outputs the acknowledge bit (ACK) on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 0, I3C continues to place itself in slave receive mode and sets the NTST.RDBFF0 flag to 1.

3. After the BST.SPCNDDF flag is confirmed to be 0 and the NTST.RDBFF0 flag to be 1, dummy read the NTDTBP0 register (the dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower 8 bits when the 10-bit address format is selected).
4. When the NTDTBP0 register is read, I3C automatically sets the NTST.RDBFF0 flag to 0. If reading of the NTDTBP0 register is delayed and a next byte is received while the RDBFF0 flag is still set to 1, I3C holds the I3C_SCL line low from one SCL cycle before the timing with which RDBFF0 should be set. In this case, reading the NTDTBP0 register releases the I3C_SCL line from being held at the low level. When the BST.SPCNDDF flag = 1 and the NTST.RDBFF0 flag is also 1, read the NTDTBP0 register until all the data is completely received.
5. Upon detecting the STOP condition, I3C automatically clears bits SVST.HOAF, GCAF, and SVAF[n] (n = 0 to 2) to 0.
6. After checking that the BST.SPCNDDF flag = 1, set the BST.SPCNDDF flag to 0 for the next transfer operation.

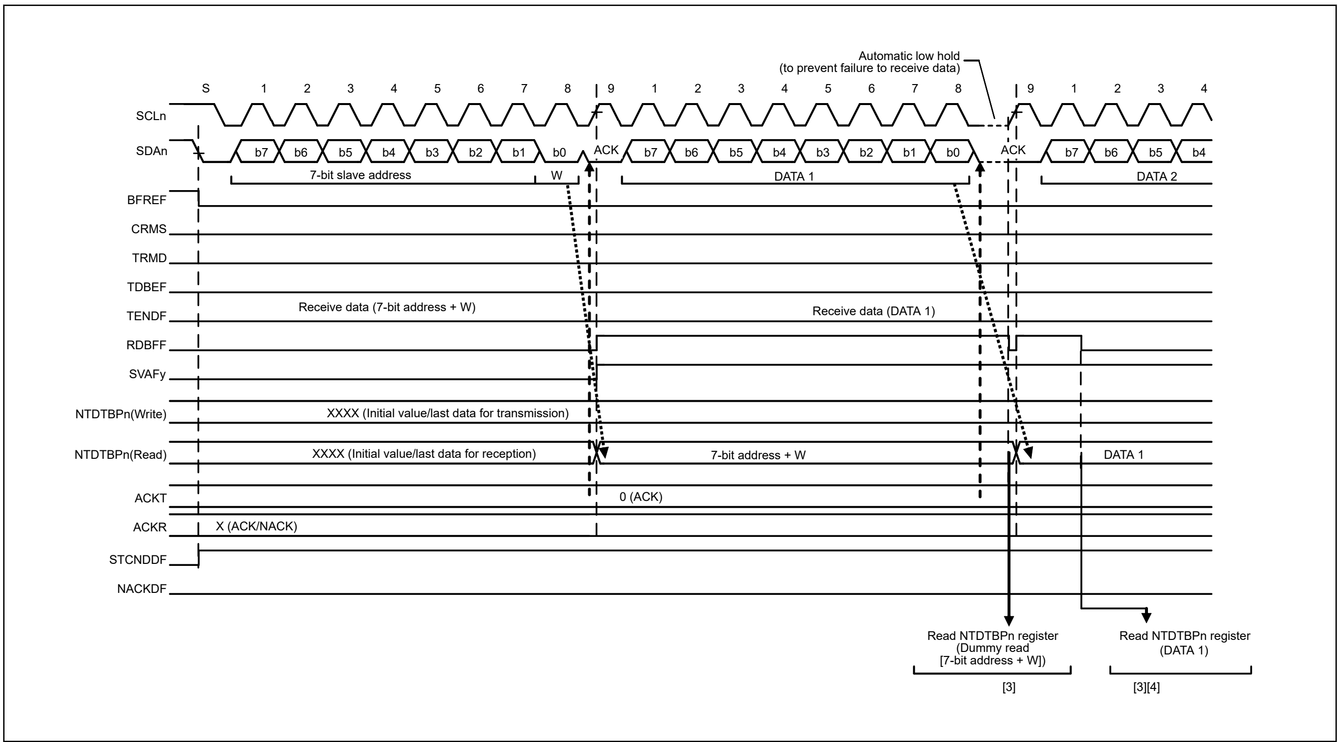


Figure 40.37 Slave receive operation timing (1) (7-bit address format, when ACKTWE = 0)

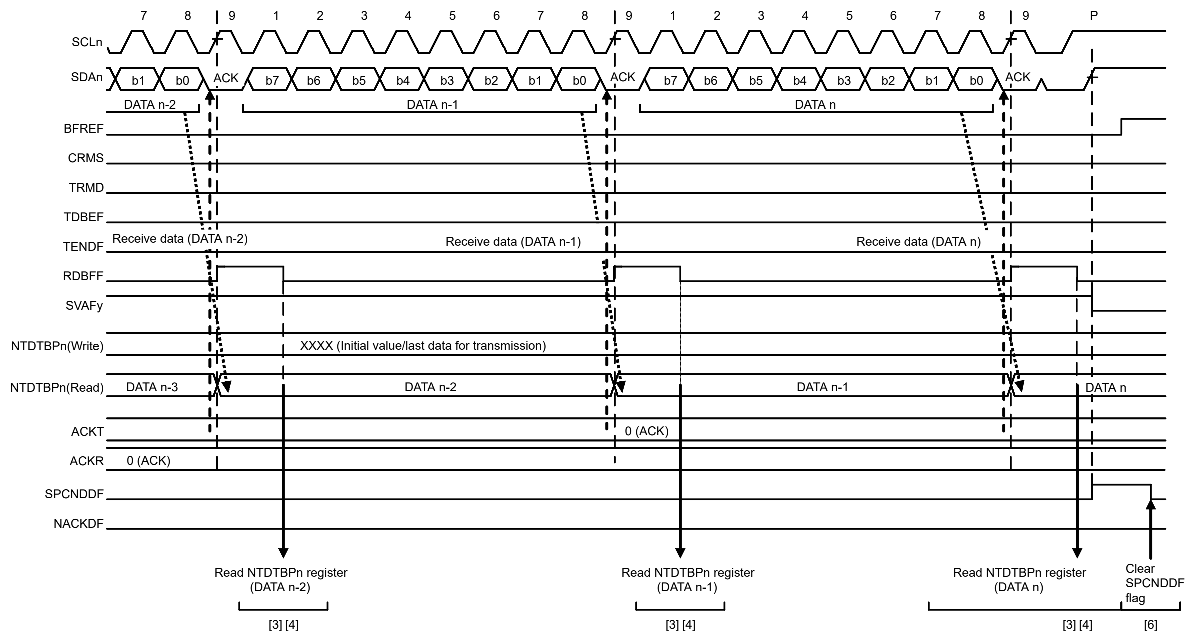


Figure 40.38 Slave receive operation timing (2) (when ACKTWE = 0)

(b) Data Read Transfer (Single Buffer transfer)

In slave transmit operation, the master device outputs the SCL clock, I3C transmits data as a slave device, and the master device returns acknowledgments.

Figure 40.152 shows an example of usage of slave transmission and Figure 40.39 and Figure 40.40 show the timing of operations in slave transmission.

The following describes the procedure and operations for slave transmission.

1. Initial settings. For details, see section 40.3.3.1. Initial Setting Flow.
   After initial settings, I3C will stay in the standby state until it receives a slave address that it matches.
2. After receiving a matching slave address, I3C sets one of the corresponding bits SVST.HOAF, GCAF, and SVAF[n] (n = 0 to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and outputs the acknowledge bit (ACK) on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 1, I3C automatically places itself in slave transmit mode by setting both the PRSST.TRMD bit and the NTST.TDBEF0 flag to 1.
3. After the NTST.TDBEF0 flag is confirmed to be 1, write the data for transmission to the NTDTBP0 register.
   At this time, if I3C does not receive acknowledge from the master device (receives an NACK signal) while the BSTE.NACKDE bit = 1, I3C aborts transfer of the next data.
4. Wait until the following (a) or (b) condition.
   (a) The BST.NACKDF flag is set to 1.
   (b) The BST.TENDF flag is set to 1 while the NTST.TDBEF0 flag = 1, after the last byte for transmission is written to the NTDTBP0 register.
5. When the BST.NACKDF flag or the BST.TENDF flag = 1, dummy read the NTDTBP0 register to complete the processing. This releases the I3C_SCL line.
6. Upon detecting the STOP condition, I3C automatically sets bits SVST.HOAF, GCAF, and SVAF[n] (n = 0 to 2), flags NTST.TDBEF0 and BST.TENDF, and the PRSST.TRMD bit to 0, and enters slave receive mode.
7. After checking that the BST.SPCNDDF flag = 1, set the BST.NACKDF and SPCNDDF flags to 0 for the next transfer operation.

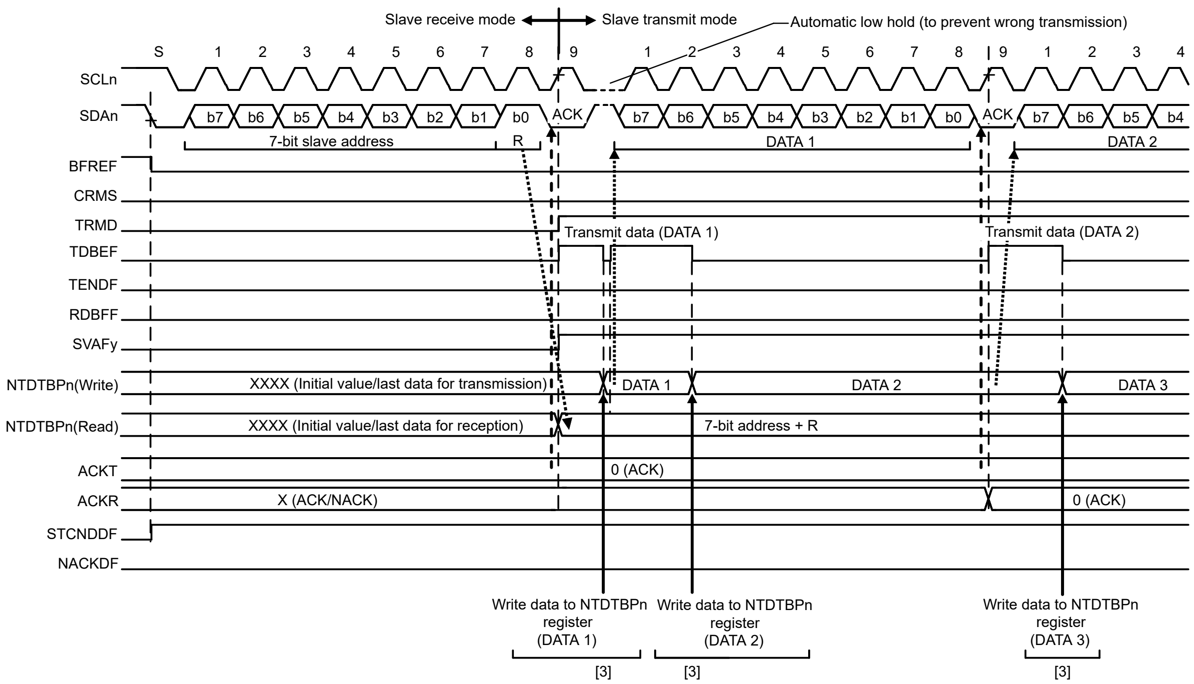


Figure 40.39 Slave transmit operation timing (1) (7-bit address format)

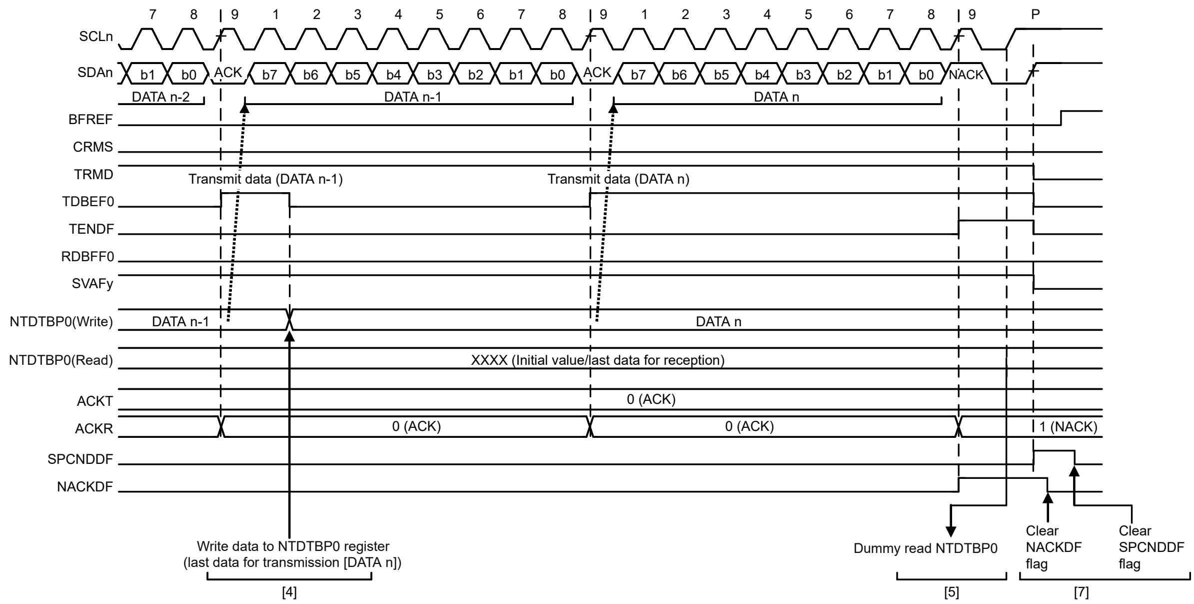


Figure 40.40 Slave transmit operation timing (2)

(2) I3C Slave Operation

(a) Dynamic Address Assign Procedure

After initializing I3C, the I3C master first performs Dynamic Address Assign Procedure.

The operation of R-I3 during the Dynamic Address Assign Procedure by ENTDAA CCC is described below.

1. Initial setting (For details, see section 40.3.3.1.2. I3C Initial Setting Flow)

2. When ENTDAA CCC is received, I3C transmits Provisional ID (SDCTPIDH[31:0], SDCTPIDL[15:0]), BCR (SVDCT.TBCRn), DCR (SVDCT.TDCR[7:0]) until a dynamic address is assigned. (For details, see "In case of Broadcast CCC (ENTDAA)" of (6) CCC detection function [I3C mode].)
3. When ENTDAA CCC is completed and a STOP condition is detected, Receive Status Descriptor is stored in Receive Status Queue.
4. Read Receive Status Descriptor via NRSQP register and check the status.
5. Read the data for the Data Length indicated by the DATA_LENGTH[15:0] bits of the Receive Status Descriptor from the Receive Data Buffer via the NTDTBP0 register.

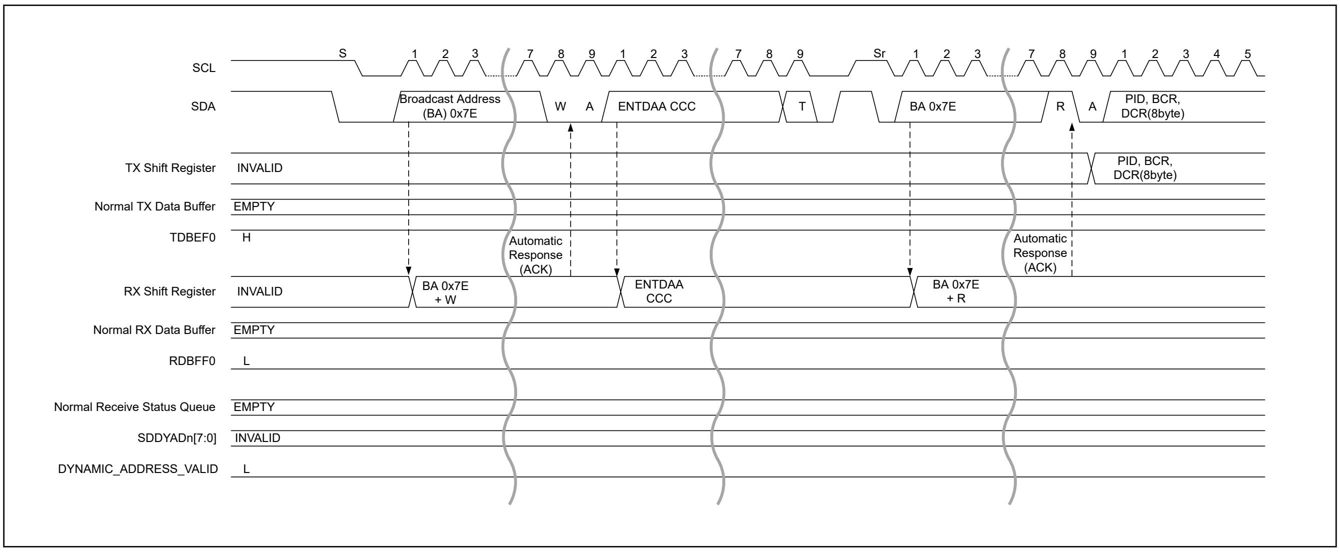


Figure 40.41 Dynamic address assign procedure (ENTDAA CCC) timing (1/3)

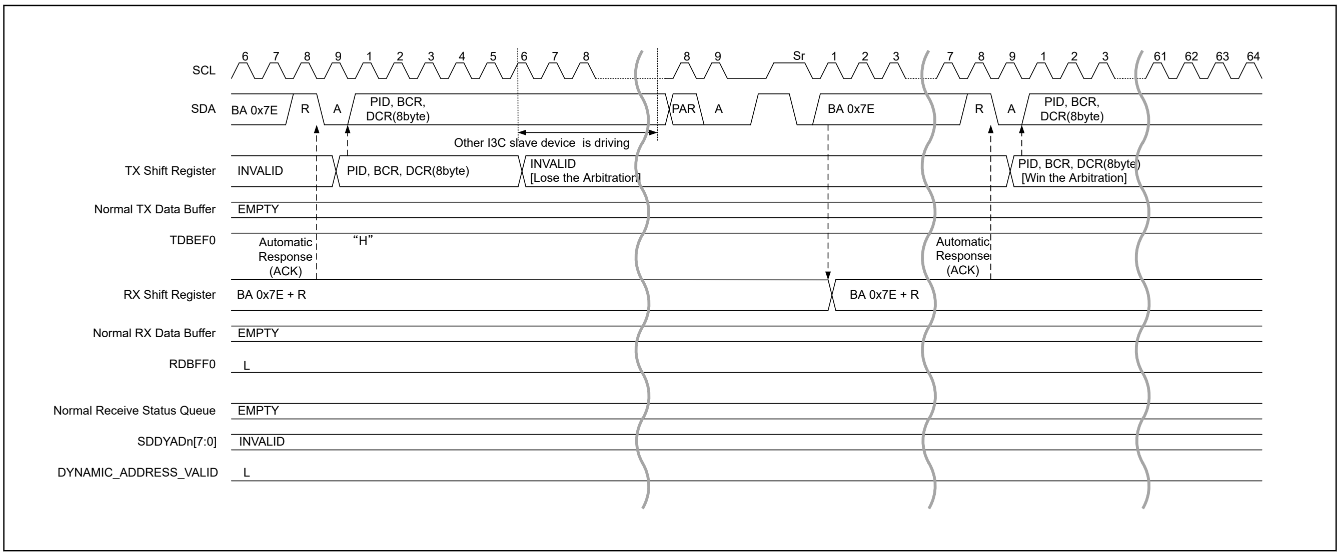


Figure 40.42 Dynamic address assign procedure (ENTDAA CCC) timing (2/3)

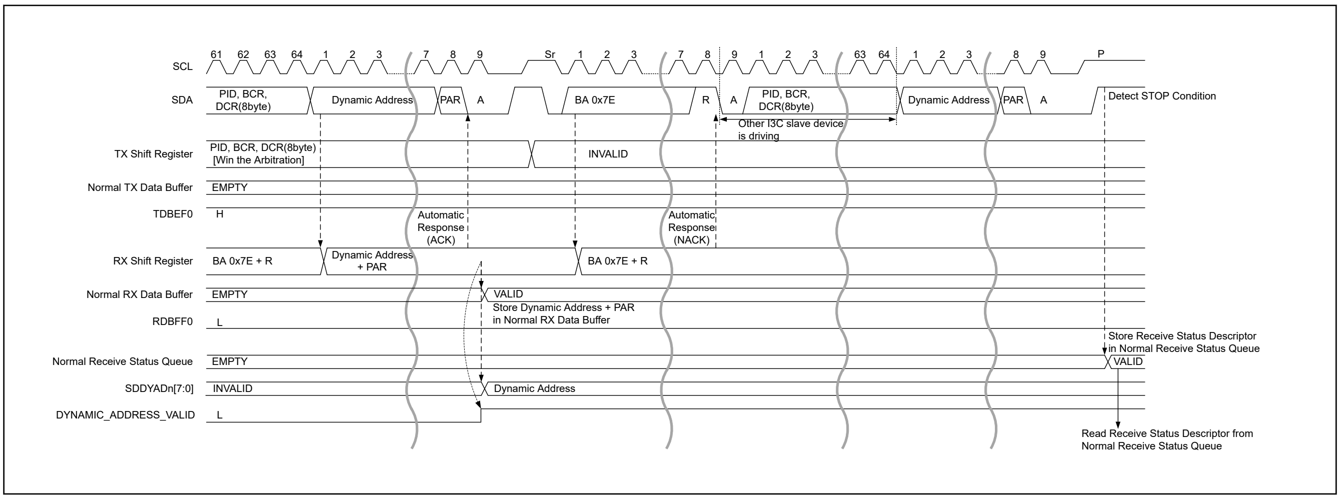


Figure 40.43 Dynamic address assign procedure (ENTDAA CCC) timing (3/3)

When communicating with a Static Address until the Dynamic Address is assigned from the I3C Master, by setting to the DVSTAD[6:0] bit of DAT (SDATBASn register), the SSTADV bit of the SVDVADn register is set to 1 and the Static Address Will be effective.

If the I3C Slave has a Static Address and the I3C Master executes the Dynamic Address Assign Procedure, it is possible to assign a Dynamic Address with SETDASA CCC.

The operation of I3C during SETDASA CCC Dynamic Address Assign Procedure is described below.

1. Initial setting (For details, see section 40.3.3.1.2. I3C Initial Setting Flow)
2. When SETDASA CCC which agrees with its own Static Address is received, the SDDYAD [7:0] bit of DAT (SDATBAS0 register) is renewed and SDYADV bit of SVDVAD0 register is set in 1. (For details, see "In case of Direct Write CCC" of (6) CCC detection function [I3C mode].)
3. When SETDASA CCC is completed and a STOP condition is detected, Receive Status Descriptor is stored in Receive Status Queue.
4. Read Receive Status Descriptor via NRSQP register and check the status.

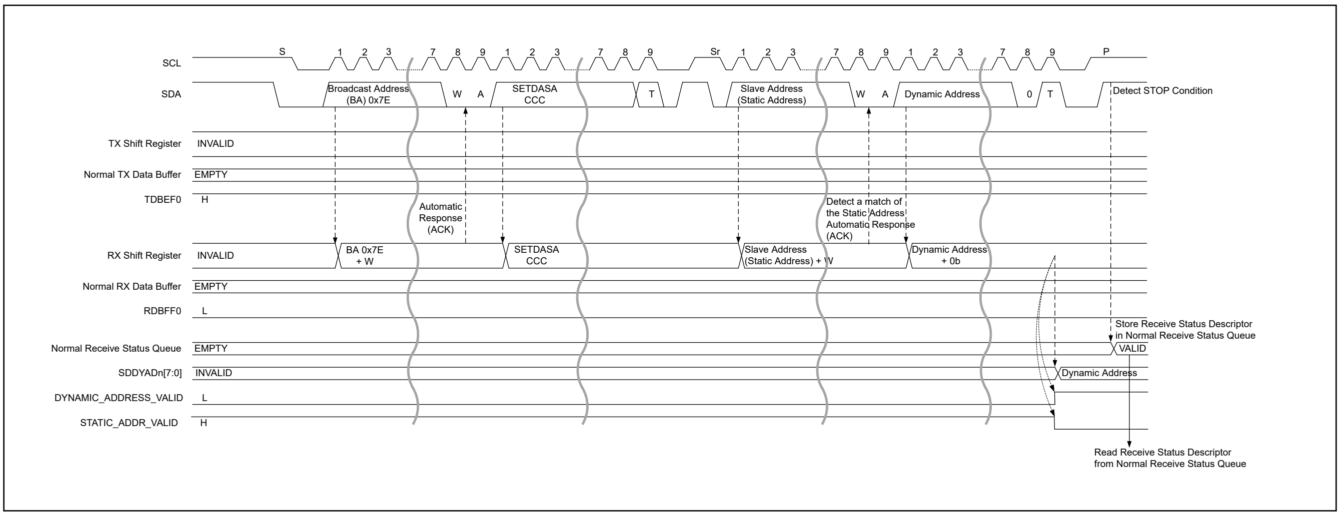


Figure 40.44 Dynamic address assign procedure (SETDASA CCC) timing

(b) SDR Data Write Transfer

1. When Transaction is issued from the I3C Master, it compares the Slave Address of Address Header with its own Slave Address, and if it matches, I3C responds with ACK.
When a Transaction is received, if the Receive Data Buffer is full, the I3C Slave will respond with NACK in the Address Header.
In preparation for retrying the I3C Master, read the data from the Receive Data Buffer via the NTDTBPn register, and empty the Receive Data Buffer.

2. Data received from I3C Master is stored in the Receive Data Buffer.
3. With the RDBFF0 = 1 interrupt, the received data is read from the Receive Data Buffer via the NTDTBPn register.
4. When Repeated START condition or STOP condition is detected, the Receive Status Descriptor is stored in the Receive Status Queue.
5. Read Receive Status Descriptor via NRSQP register and check the status.

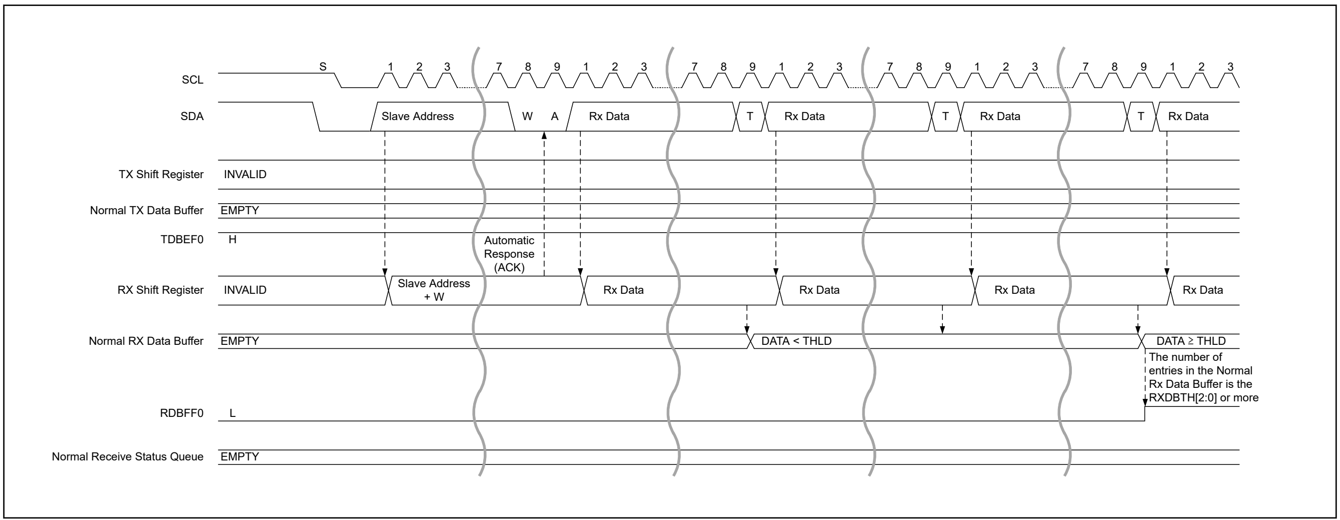


Figure 40.45 SDR data write transfer timing (1/2)

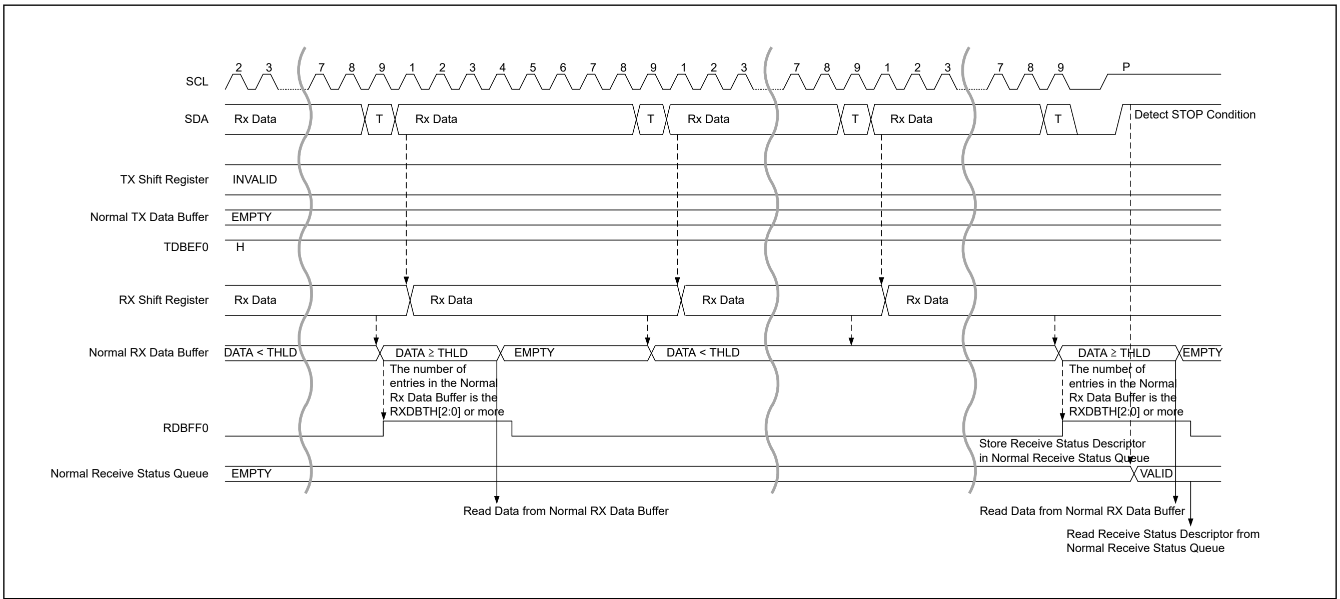


Figure 40.46 SDR data write transfer timing (2/2)

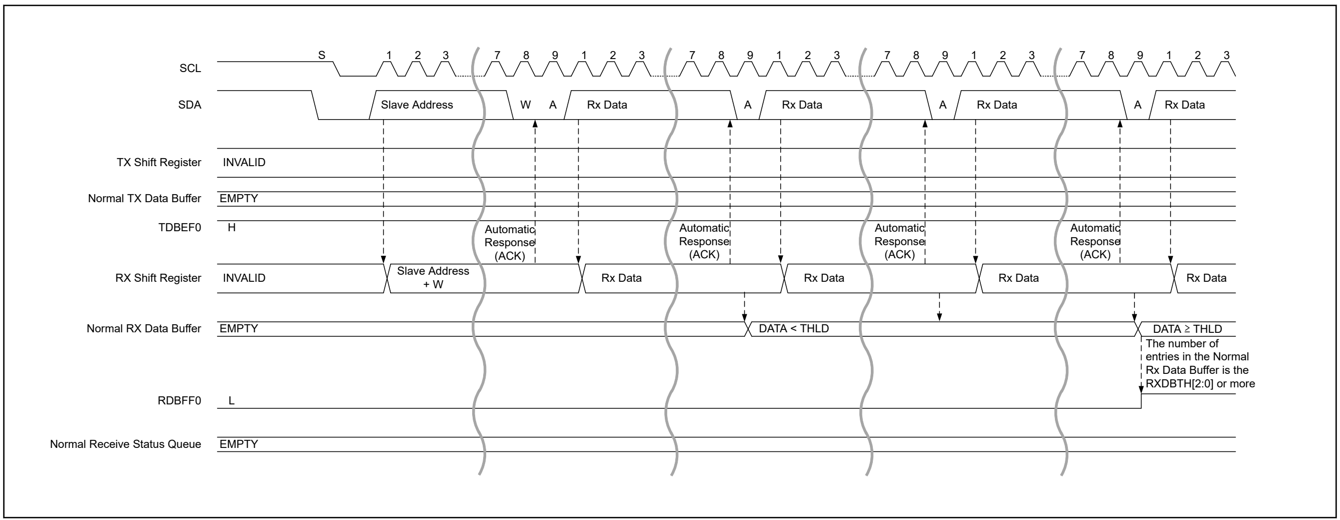


Figure 40.47 Legacy I²C message data write transfer timing (1/2)

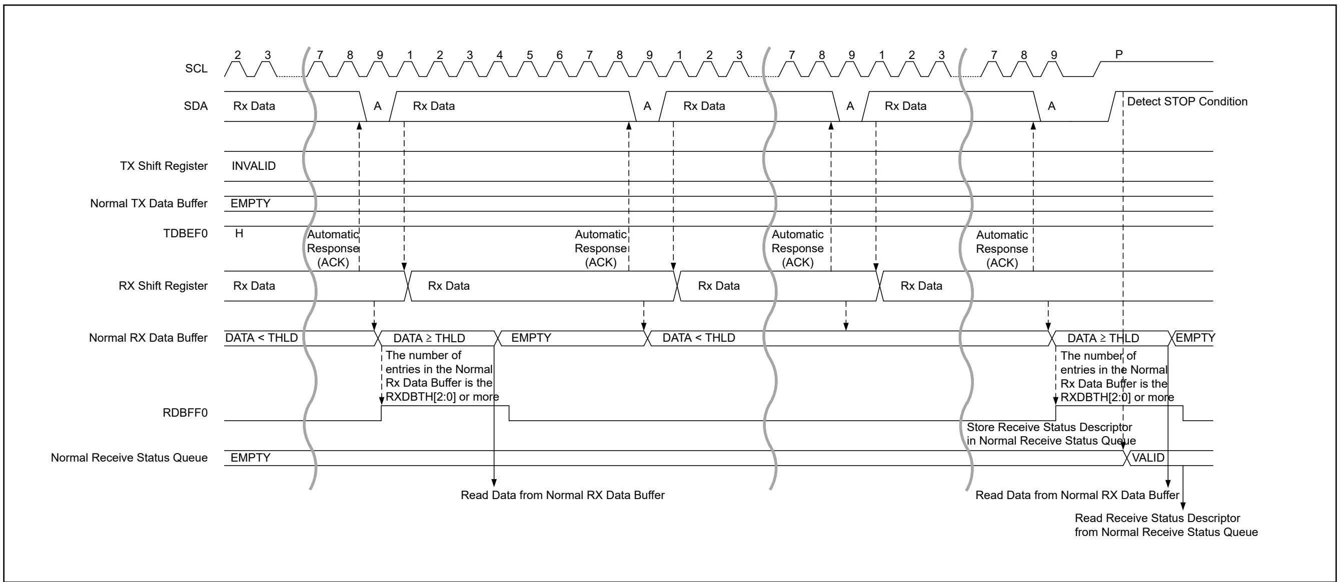


Figure 40.48 Legacy I²C message data write transfer timing (2/2)

(c) SDR Data Read Transfer

1. Write the data requested from the I3C Master to the Tx Data Buffer via the NTDTBPn register.
2. When Transaction is issued from the I3C Master, it compares the Slave Address of Address Header with its own Slave Address, and if it matches, I3C responds with ACK.
   When a Transaction is received, if the Tx Data Buffer is EMPTY, I3C Slave responds with NACK with the Address Header.
   In preparation for retrying the I3C Master, write data to the Tx Data Buffer via the NTDTBPn register.
3. Transmit the data stored in the Tx Data Buffer.
4. If data to be transmitted still remains, write the data to be transmitted with an interrupt by TDBEF0 = 1 to the Tx Data Buffer via the NTDTBPn register.
5. SDR:
   When the transmission of the data stored in the Tx Data Buffer is completed, Low is output to the T-bit following Data, and it is notified to the I3C Master that it is the final data.
   Legacy I²C Message:
   When NACK is detected, data transmission is terminated.
6. When a Repeated START condition or STOP condition is detected, the Receive Status Descriptor is stored in the Receive Status Queue.

7. Read the Receive Status Descriptor via NRSQP and check the status.

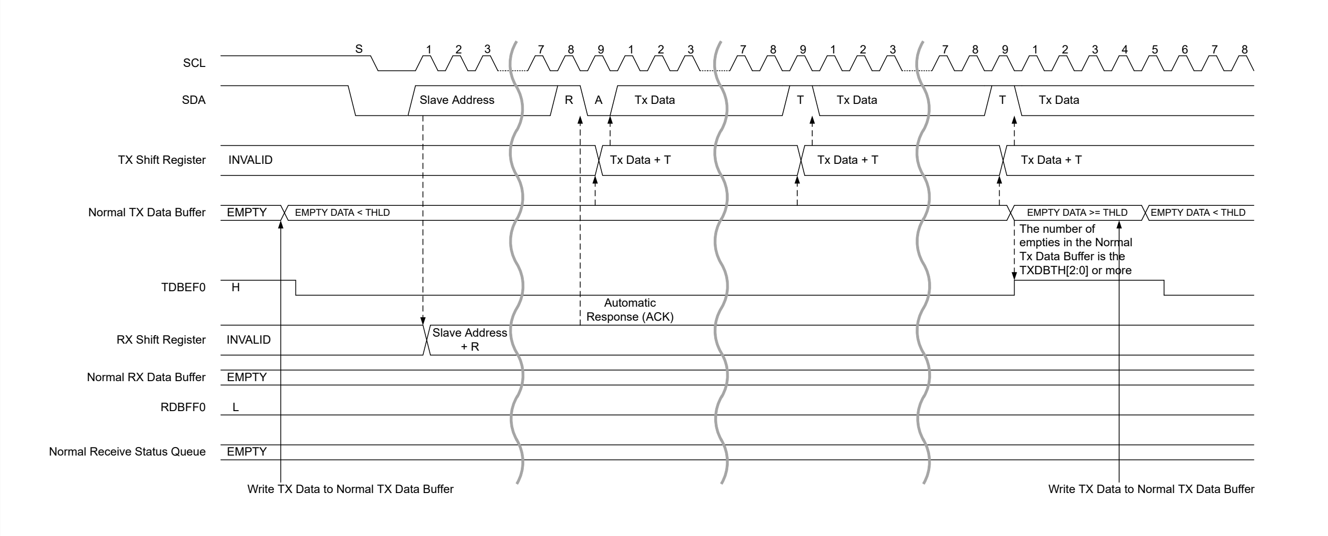


Figure 40.49 SDR data read transfer timing (1/2)

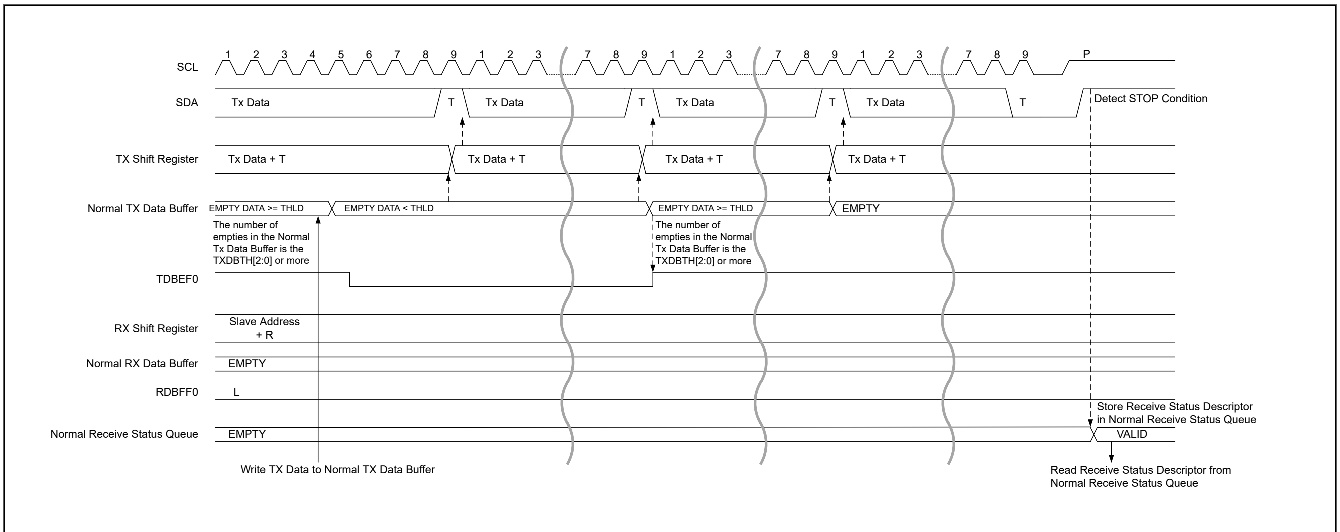


Figure 40.50 SDR data read transfer timing (2/2)

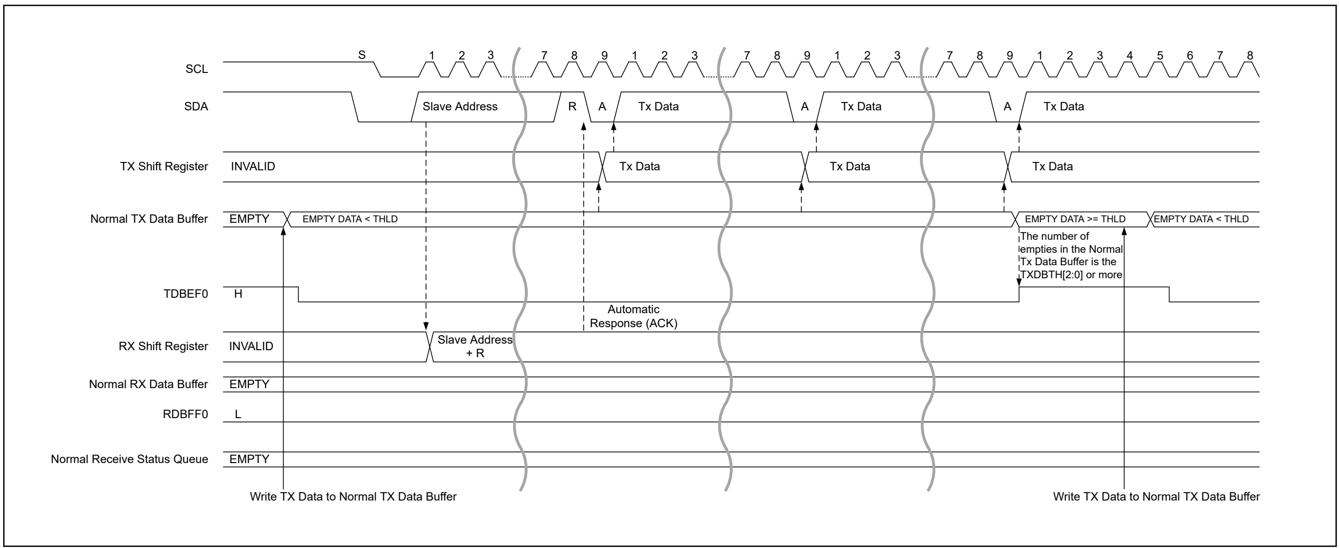


Figure 40.51 Legacy I2C message data read transfer timing (1/2)

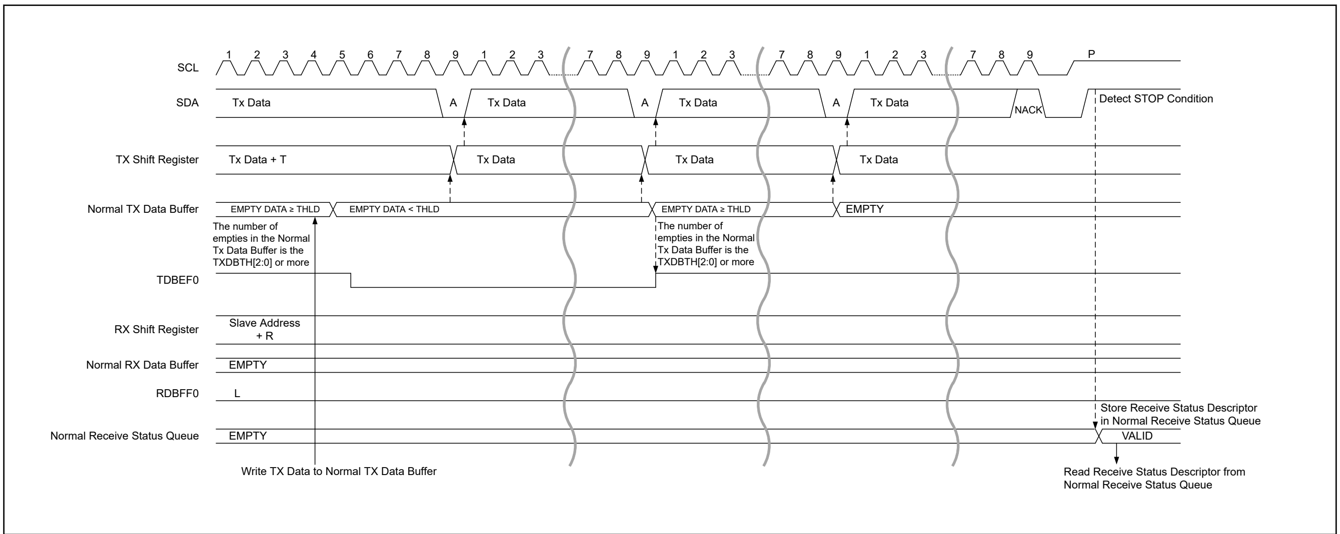


Figure 40.52 Legacy I2C message data read transfer timing (2/2)

(d) HDR Data Write Transfer

1. Upon receipt of ENTHDR * CCC from the I3C Master, it transits to HDR mode (For details, see "In case of Broadcast CCC (ENTHDR *)" of (6) CCC detection function [I3C mode]).
2. Compare the Slave Address of the HDR Command Word issued from the I3C Master with its own Slave Address, and if it matches, receive the following Data Word.
3. Data received from the I3C Master is stored in the Receive Data Buffer.
4. With the RDBFF0 = 1 interrupt, the received data is read from the Receive Data Buffer via the NTDTBPn register.
5. When detecting HDR Restart Pattern or HDR Exit Pattern + STOP condition, it stores the Receive Status Descriptor into the Receive Status Queue.
6. Read the Receive Status Descriptor via NRSQP and check the status.

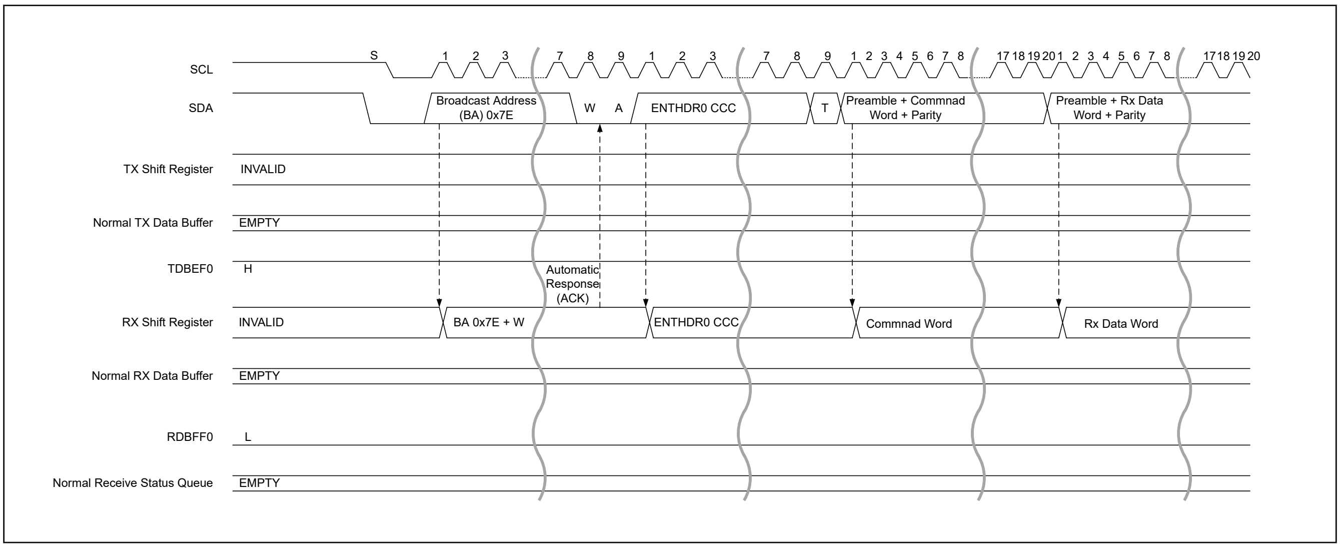


Figure 40.53 HDR data write transfer (HDR-DDR) timing (1/3)

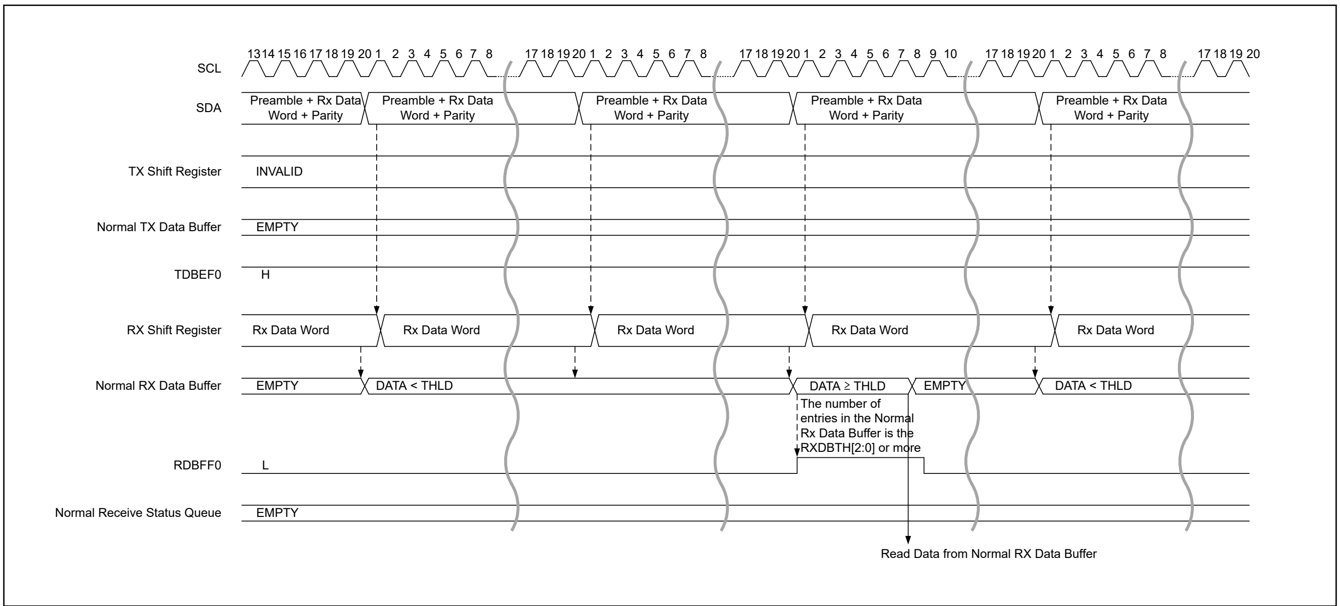


Figure 40.54 HDR data write transfer (HDR-DDR) timing (2/3)

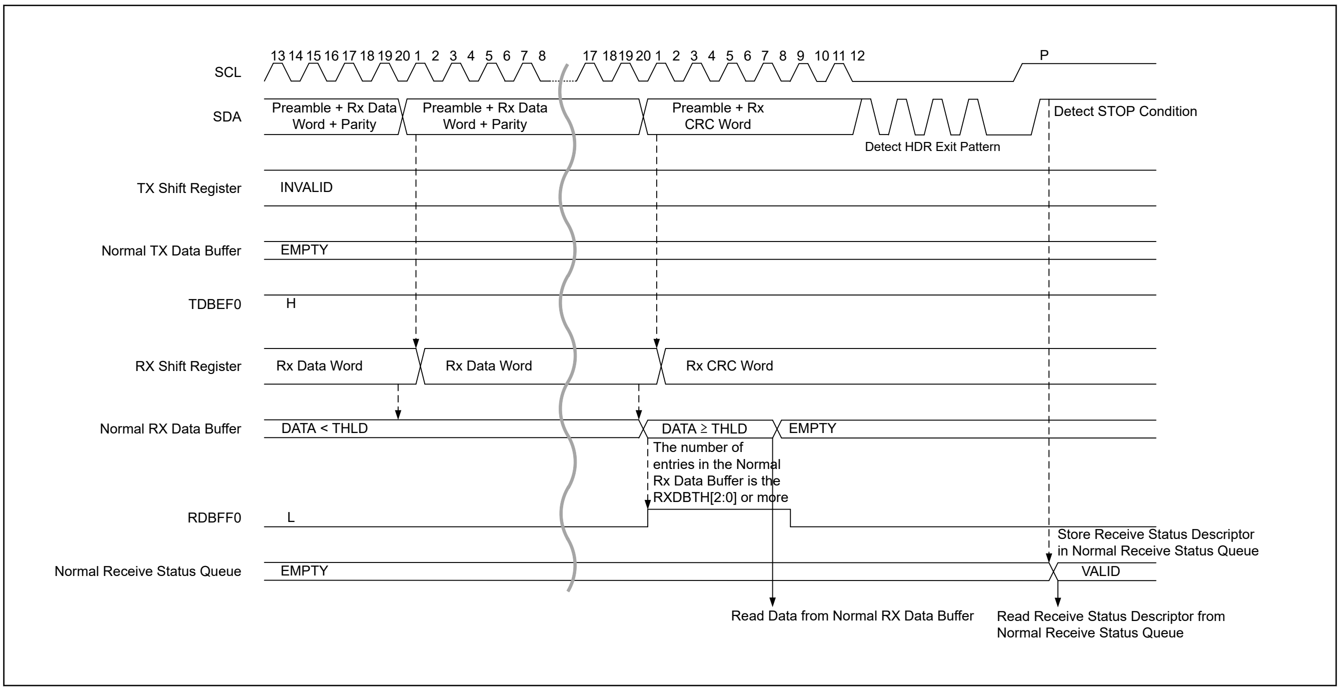


Figure 40.55 HDR data write transfer (HDR-DDR) timing (3/3)

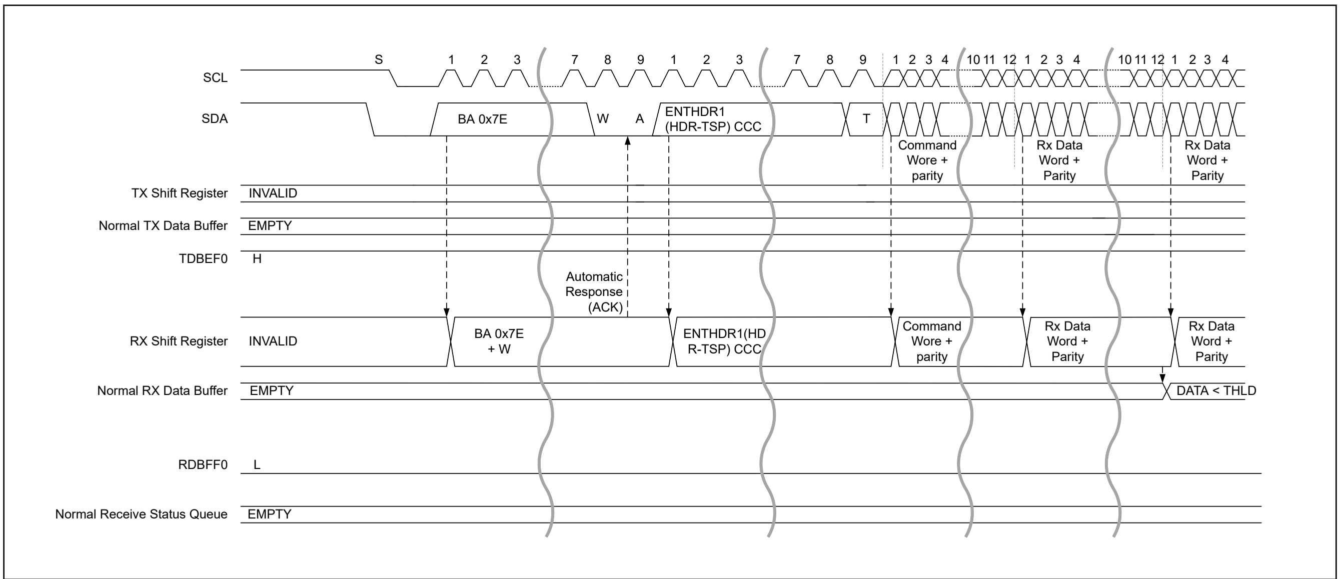


Figure 40.56 HDR data write transfer (HDR-TSP, TSL) timing (1/2)

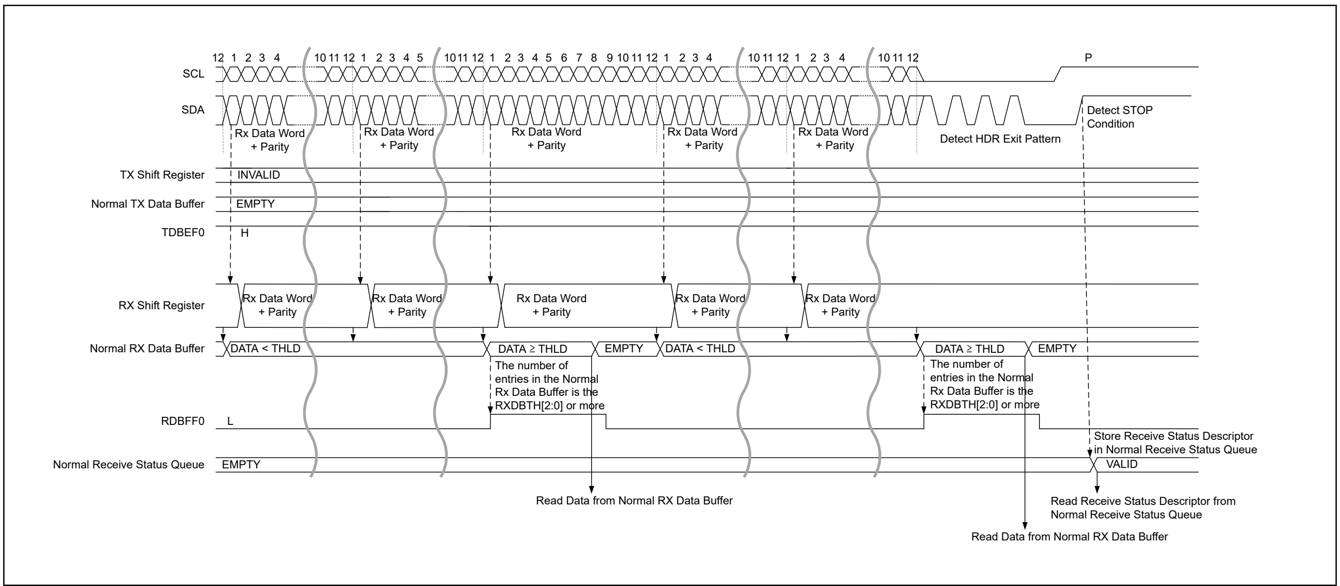


Figure 40.57 HDR data write transfer (HDR-TSP, TSL) timing (2/2)

(e) HDR Data Read Transfer

1. Write the data requested by the I3C Master to the Tx Data Buffer via the NTDTBPn register.
2. Upon receipt of ENTHDR * CCC from the I3C Master, it transits to HDR mode (For details, see "In case of Broadcast CCC (ENTHDR *)" of (6) CCC detection function [I3C mode]).
3. When comparing the Slave Address of the HDR Command Word issued from the I3C Master with its own Slave Address, if it matches, the Data stored in the Tx Data Buffer is transmitted.
   If the Tx Data Buffer is EMPTY when the Slave Address of the HDR Command Word matches its own Slave Address, a NACK response will be sent.
   NACK response :
   - HDR-DDR :
     Send 11 followed by Command Word followed by Preamble.
   - HDR-Ternary :
     Send the beginning of HDR Restart and HDR Exit Pattern (in English translation: "start of the HDR Restart Pattern and HDR Exit Pattern." In MIPI Spec)
4. If data to be transmitted still remains, Write the data to be transmitted with an interrupt by TDBEF0 = 1 to the Tx Data Buffer via the NTDTBPn register.
5. HDR-DDR:
   When the transmission of the data stored in the Tx Data Buffer is completed, the CRC Word is sent continuously to notify the I3C Master that it is the final data.
   HDR-Ternary:
   When the transmission of the data stored in the Tx Data Buffer is completed, the beginning of the HDR Restart or HDR Exit Pattern is sent continuously to notify the I3C Master that it is the final data.
6. When detecting HDR Restart Pattern and HDR Exit Pattern + STOP condition, it stores the Receive Status Descriptor into the Receive Status Queue.
7. Read the Receive Status Descriptor via NRSQP and check the status.

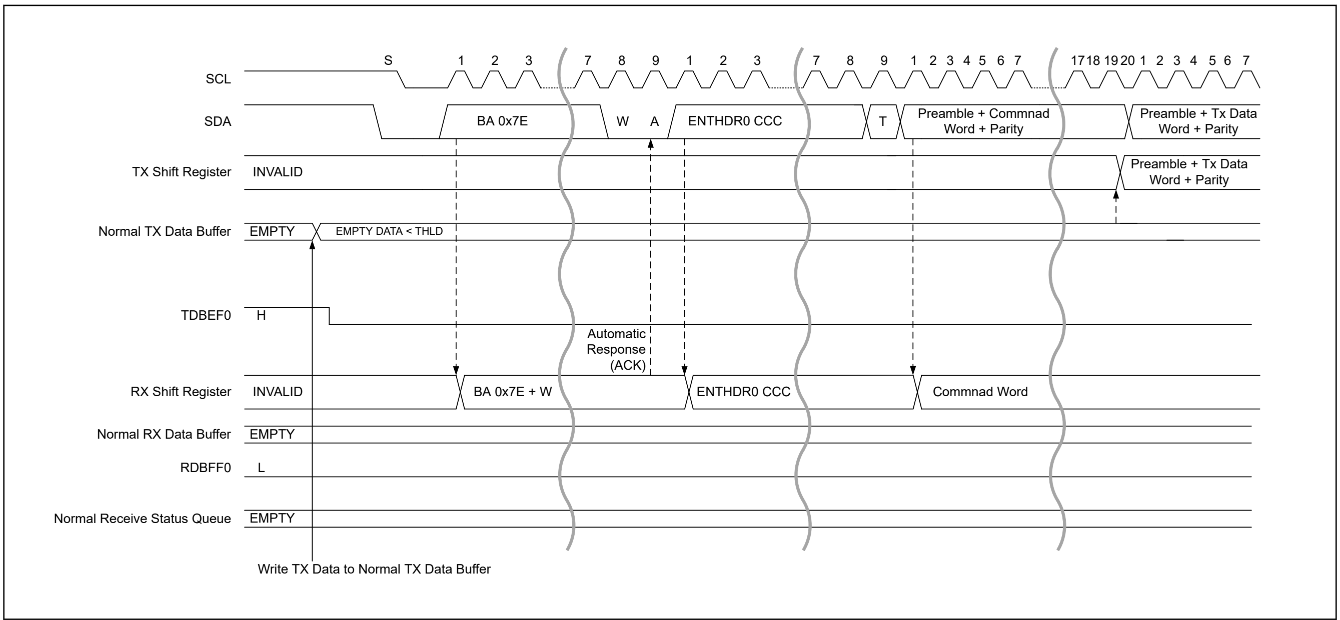


Figure 40.58 HDR data read transfer (HDR-DDR) timing (1/3)

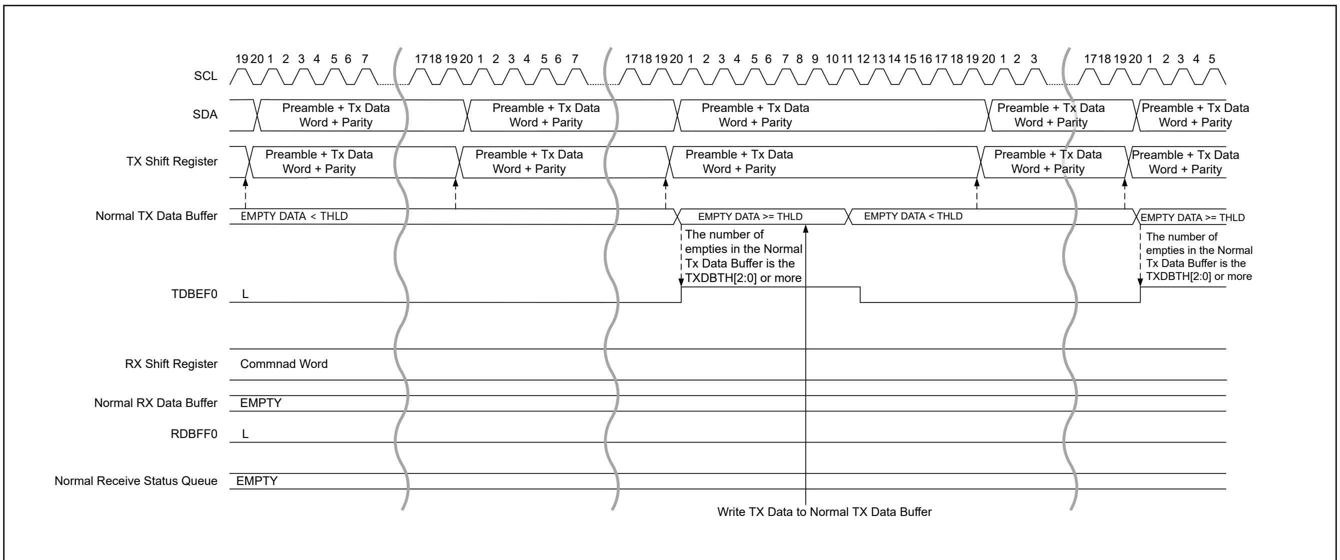


Figure 40.59 HDR data read transfer (HDR-DDR) timing (2/3)

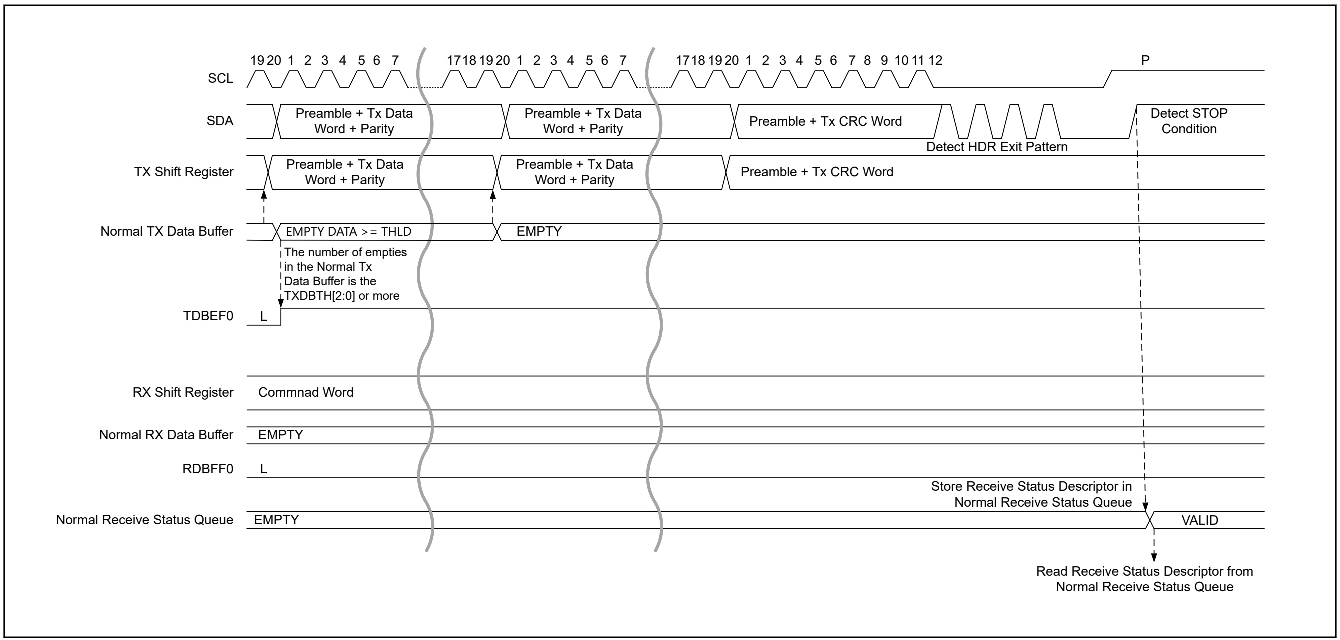


Figure 40.60 HDR data read transfer (HDR-DDR) timing (3/3)

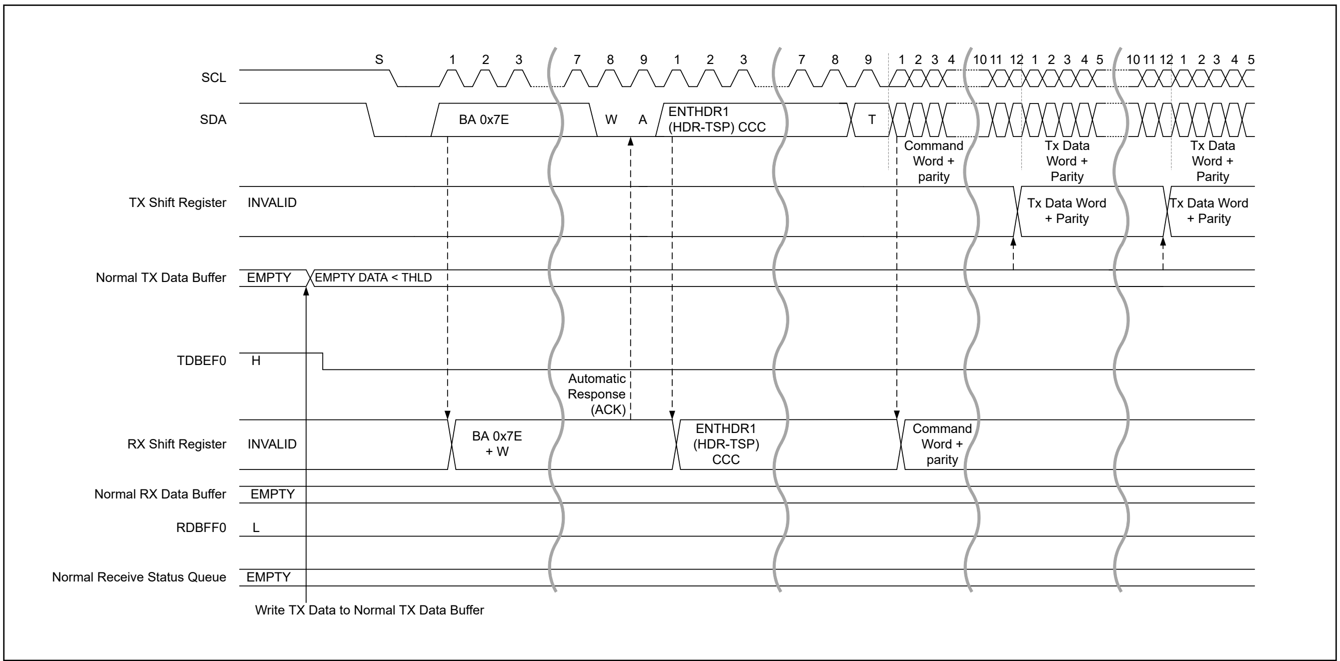


Figure 40.61 HDR data read transfer (HDR-TSP, TSL) timing (1/2)

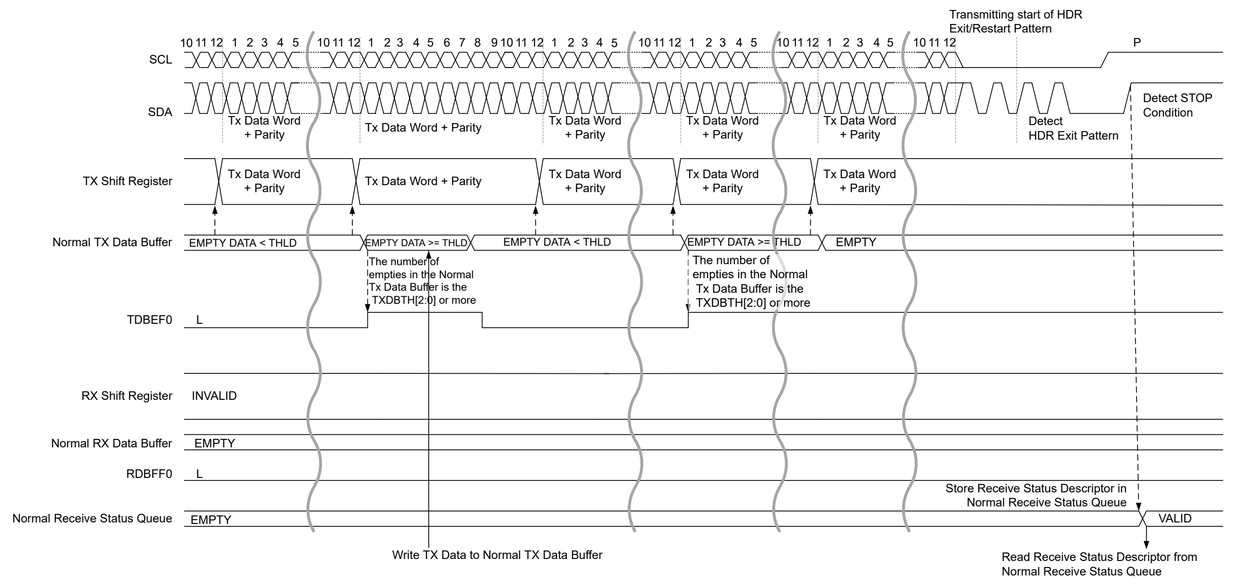


Figure 40.62 HDR data read transfer (HDR-TSP, TSL) timing (2/2)

(f) IBI Transfer

1. When sending Slave Interrupt Request.
   When transmitting IBI Data, write IBI Data to the IBI Data Buffer via the NIBIQP register.
2. Write Command Descriptor (Immediate Transfer Command or Regular Transfer Command) to the Command Buffer for IBI Transfer via the NCMDQP register.
3. When Command Descriptor is written to Command Buffer, IBI Transaction is issued under the following conditions.
   - When START condition is detected in Slave Interrupt Request or Mastership Request. (Does not apply a Repeated START condition)
   - If no START is forthcoming within the following Bus Condition, then this module issue a START Request by pulling the I3C_SDA line Low.
   - (a) Slave Interrupt Request, Mastership Request : Bus Available
4. In Slave Address with RnW of the Address Header, if losing Arbitration by issuing a Transaction from I3C Master, stop issuing Transaction.
   When detecting Repeated START condition or STOP condition, store the Response Descriptor into the Response Buffer.
5. When sending Slave Interrupt Request:
   - When IBI data for transmission still remain, write IBI data with an interrupt by IBIQEFF = 1 to the IBI Data Buffer via the NIBIQP register.
   - When the transmission of IBI Data for the number of Data Length specified by the DATA_LENGTH[15:0] bits of the Command Descriptor is completed, output Low to the T-bit following IBI Data and notify the I3C Master that it is the final IBI Data.
6. When detecting Repeated START condition or STOP condition, store the Response Descriptor into the Response Buffer.
7. Read the Response Descriptor from the Response Buffer with the NRSPQP register and check the status. If NACK is responded, repeat steps 1 to 7.
8. When sending Slave Interrupt Request:
   Check that the value of the DATA_LENGTH[15:0] bit of the Response Descriptor is 0.

The Mastership processing flow is shown in Figure 40.65.

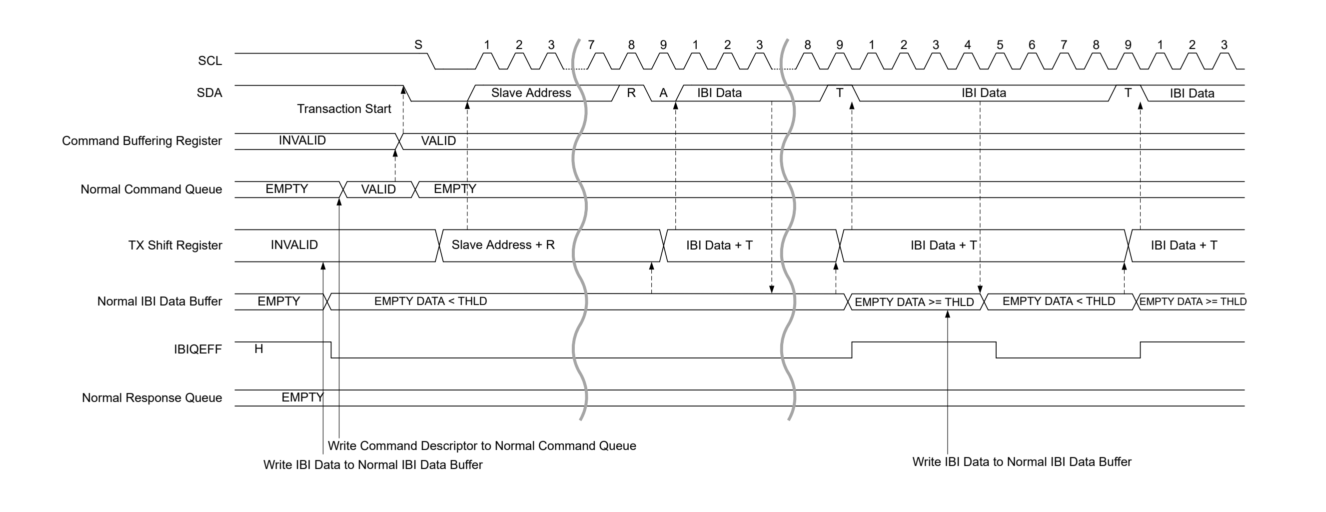


Figure 40.63 I3C slave IBI transfer timing (1/2)

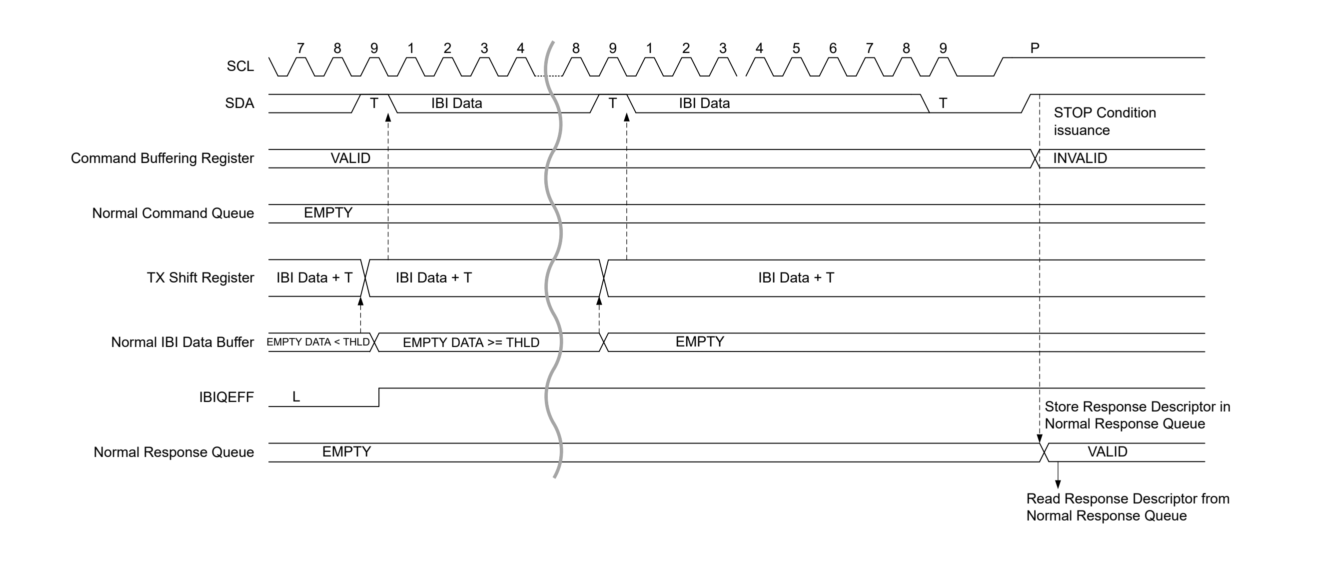


Figure 40.64 I3C slave IBI transfer timing (2/2)

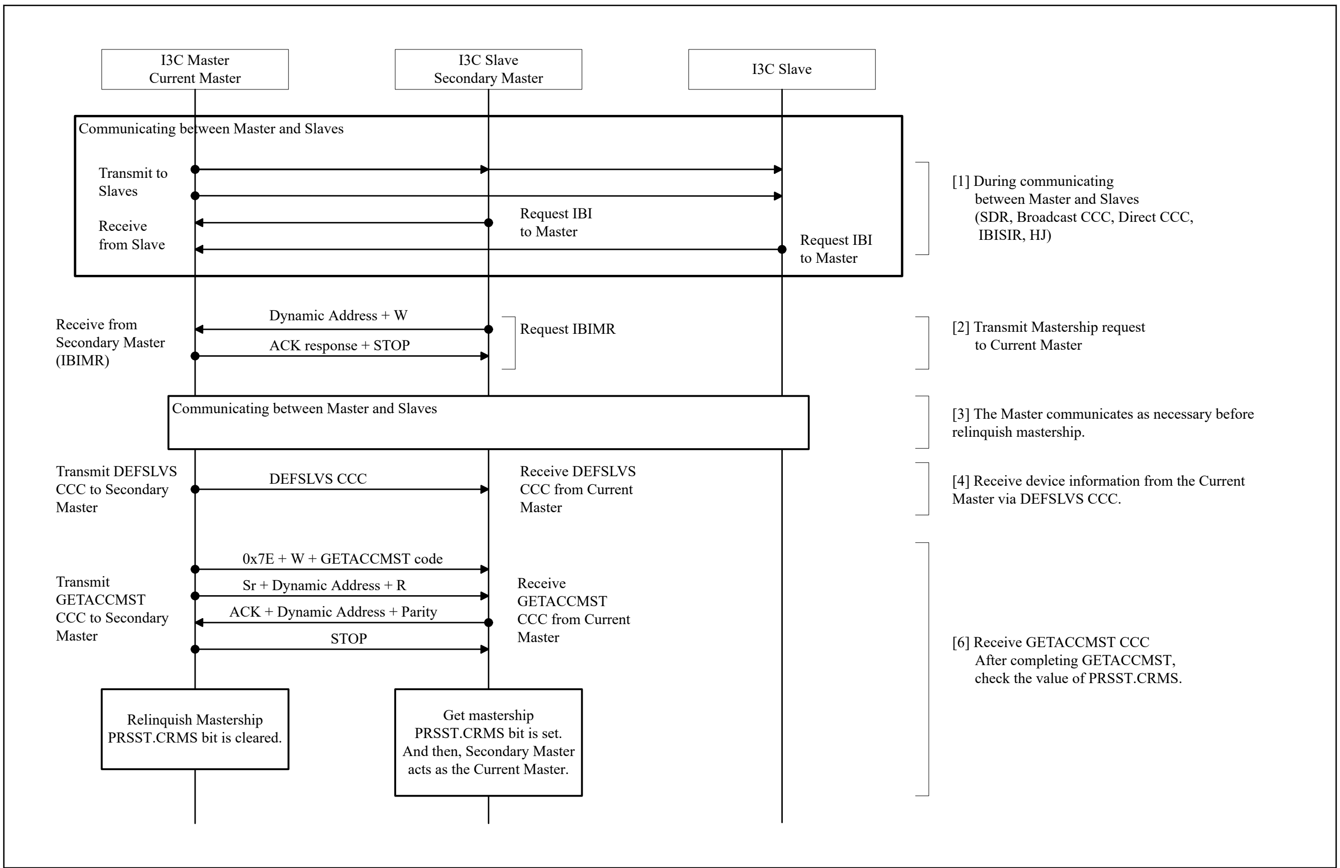


Figure 40.65 I3C Slave Mastership processing flow

40.3.2.2 Data Handler

The relationship between the transfer method and the queue is shown in Table 40.10.

Table 40.10 Transfer method and queue

Protocol	Transfer method	Queue/Buffer	size	Master	Slave	Secondary Master
I^2C Mode	Single buffer transfer	Normal Tx Data	1 byte	✓	✓	—
		Normal Rx Data	1 byte	✓	✓	—
I3C Mode	Normal FIFO buffer transfer	Normal Command	4 QUEUEs	✓	✓	✓
		Normal Response	4 QUEUEs	✓	✓	✓
		Normal Tx Data	16 DWORDs	✓	✓	✓
		Normal Rx Data	16 DWORDs	✓	✓	✓
		Normal Receive Status	2 QUEUEs	—	✓	✓
		Normal IBI Status	2 QUEUEs	✓	—	✓
		Normal IBI Data	8 DWORDs	✓	✓	✓
	High Priority FIFO buffer transfer (in Master Mode only)	High Priority Command	2 QUEUEs	✓	—	✓
		High Priority Response	2 QUEUEs	✓	—	✓
		High Priority Tx Data	2 DWORDs	✓	—	✓
		High Priority Rx Data	2 DWORDs	✓	—	✓

40.3.2.2.1 Transfer Method in I²C Mode

(1) Single Buffer transfer

Each process (condition issue, data transfer, ACK / NACK response) is controlled by software.

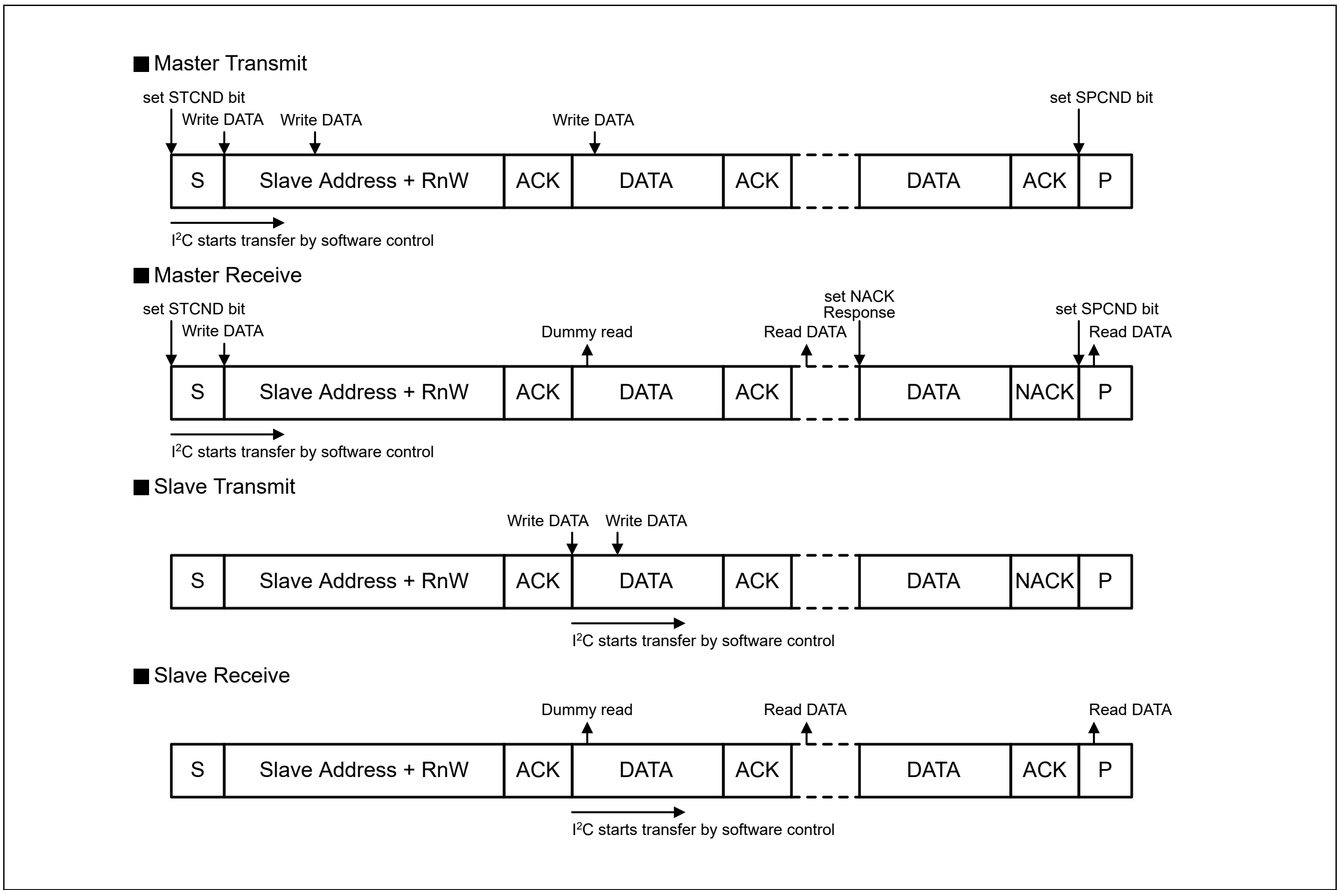


Figure 40.66 Data handler with single buffer transfer

40.3.2.2.2 Transfer Method in I3C Mode

(1) Normal FIFO Buffer Transfer

I3C autonomously starts transfer when data and command are written.

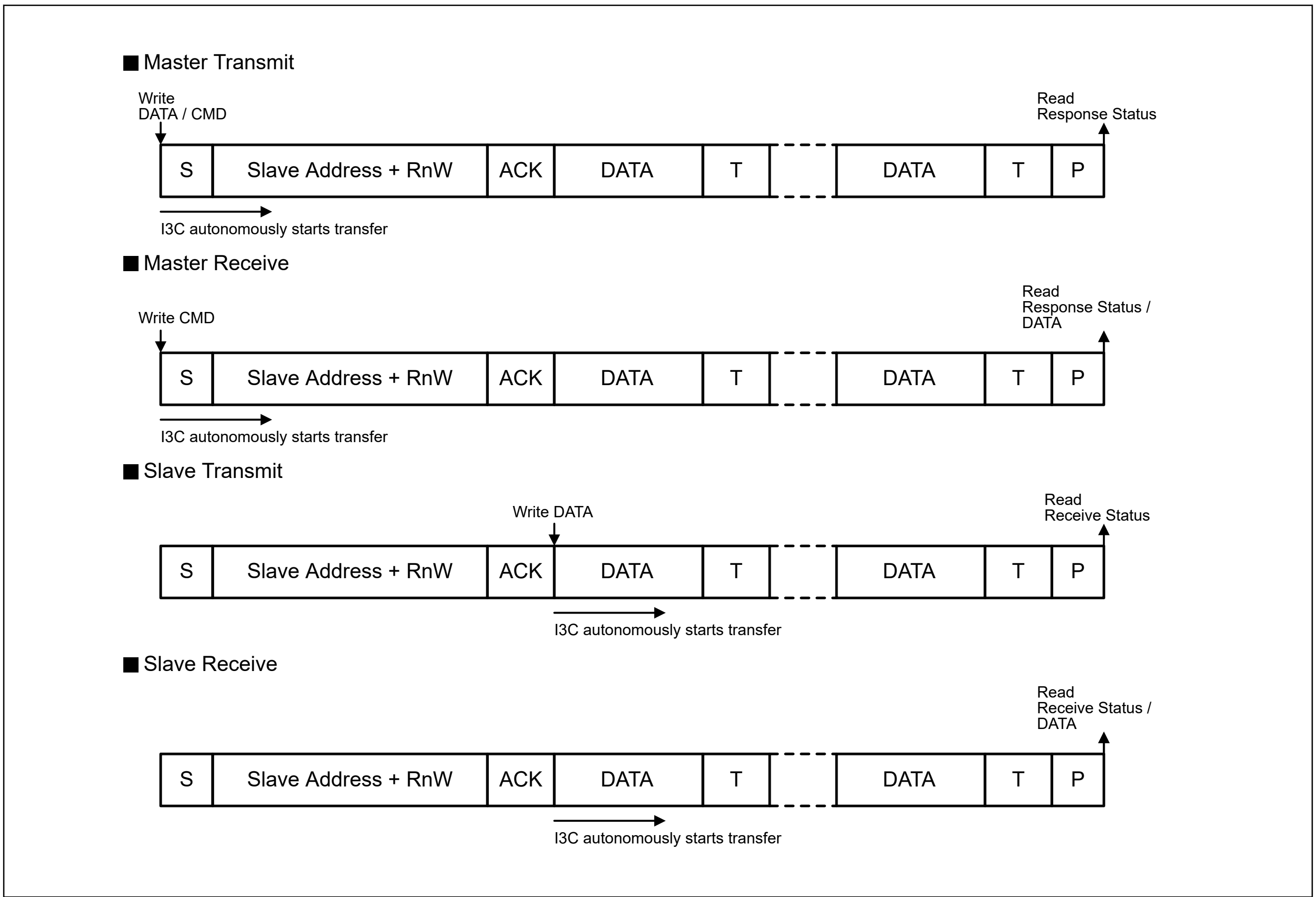


Figure 40.67 Data handler with normal FIFO buffer transfer

(2) High Priority FIFO Buffer Transfer

I3C handles the command of the High Priority FIFO buffer transfer higher priority than the command of the normal FIFO buffer transfer.

If data and commands are written to the High Priority FIFO buffer during normal FIFO buffer transfer, I3C waits for the STOP condition and then processes the command in the High Priority FIFO buffer.

After the command processing in the High Priority FIFO buffer is completed, if the command remains in the normal FIFO buffer, the I3C resumes processing the command in the normal FIFO buffer.

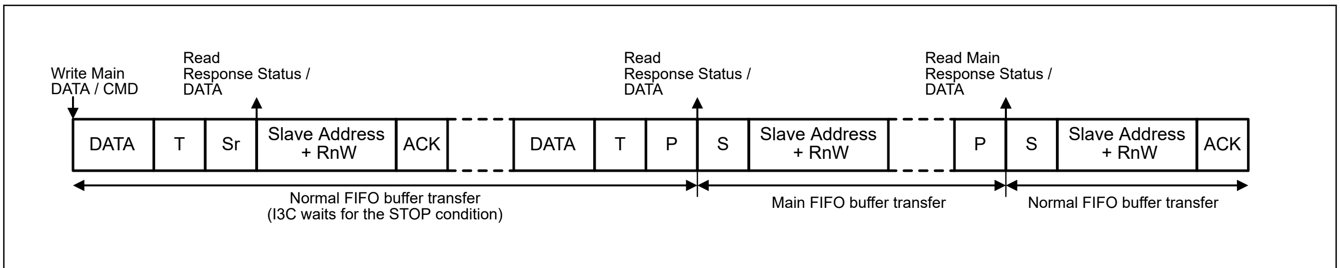


Figure 40.68 Data handler with high priority FIFO buffer transfer

40.3.2.3 I²C/I3C Protocol

40.3.2.3.1 Communication Protocol

(1) I²C Communication Data Format

The I²C bus format consists of 8-bit data and 1-bit acknowledge. The frame following a START condition or Repeated START condition is an address frame used to specify a slave device with which the master device communicates. The specified slave is valid until a new slave is specified or a STOP condition is issued.

Figure 40.69 shows the I²C bus format, and Figure 40.70 shows the I²C bus timing.

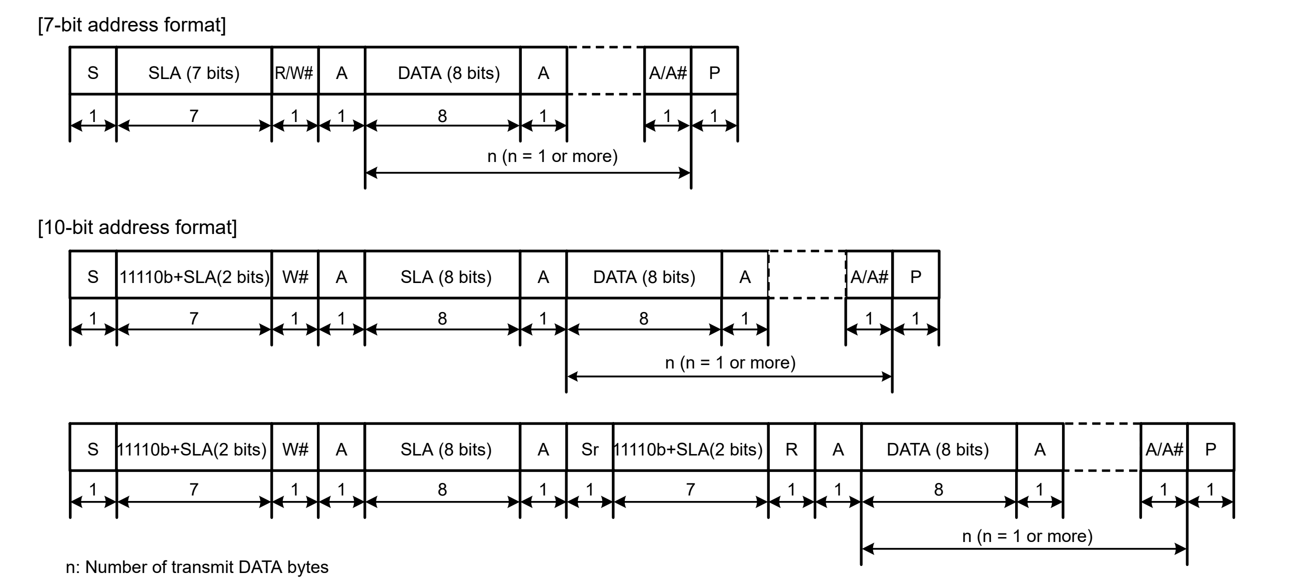


Figure 40.69 I²C bus format

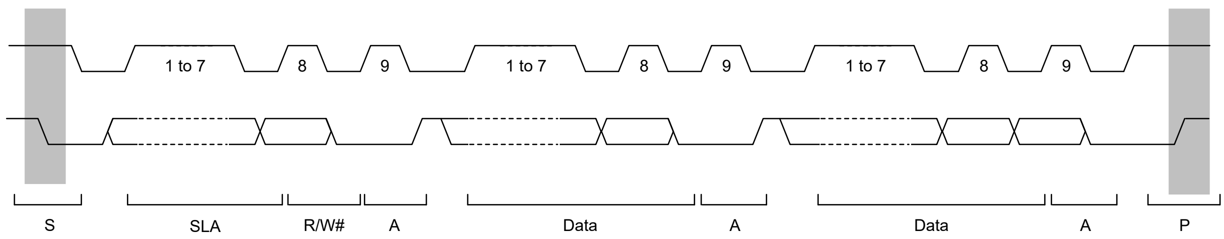


Figure 40.70 I²C bus timing (SLA = 7 bits)

S:	START condition. The master device drives the I3C_SDA line low from high level while the I3C_SCL line is at a high level.
SLA:	Slave address, by which the master device selects a slave device.
R/W#:	Indicates the direction of data transfer: from the slave device to the master device when R/W = 1, or from the master device to the slave device when R/W = 0.
A:	Acknowledge. The receive device drives the I3C_SDA line low. (In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.)
A#:	Not Acknowledge. The receive device drives the I3C_SDA line high.
Sr:	Repeated START condition. The master device drives the I3C_SDA line low from the high level after the setup time has elapsed with the I3C_SCL line at the high level.

DATA: Transmitted or received data

P: STOP condition. The master device drives the I3C_SDA line high from low level while the I3C_SCL line is at a high level.

(2) I3C Communication Data Format

Figure 40.71 through Figure 40.77Figure 40.74 illustrate a typical communication for each of the six I3C Protocols. While these diagrams do not exhaustively illustrate all possible I3C communications, they do serve as useful introductions to the signaling and transmission formatting used in each I3C Protocol.

Figure 40.71 illustrates example communication using I3C Single Data Rate (SDR) coding with Broadcast (0x7E). It shows the Master reading a byte of data from the Slave at Address 0x2B in SDR Mode. From the Bus Free Condition, the Master issues a START by driving the I3C_SDA line Low while keeping the I3C_SCL line High. It then issues the Broadcast Address (0x7E) followed by RnW (0 for Write). Then the Master turns on a pull-up resistor and goes to Open Drain.

All Slaves ACK by pulling the I3C_SDA line Low (in the Figure, pink fill means the Slave is in control of the I3C_SDA line at this time). The Master then issues a Repeated START, then the Address of the Slave (0x2B) it wants to read followed by RnW (1 for Read). The Master then turns on a pull-up resistor and goes to Open Drain, allowing the Slave to acknowledge by pulling the I3C_SDA line Low. At this point, the Master continues to toggle the I3C_SCL line and release the I3C_SDA line, allowing the Slave to drive SDA to send one byte of data (0x4A) followed by T. T = 1 informs the Master that there is additional data, whereas T = 0 signals the end. Here there is additional data, so the Slave drives SDA High until SCL goes High, at which time it releases SDA. The Master has the option of holding SDA High with a weak pullup, which signals to the Slave that the Master allows another byte to be transmitted, or to pull SDA Low (while SCL is High – hence a Repeated START), which would signal to the Slave that the Master has terminated the Read and is taking over.

SDR Mode is backwards compatible with Legacy I2C Devices, because the High time of an SCL pulse is always less than 50ns and therefore SCL will always appear to be Low because of the I2C 50ns Spike Filter.

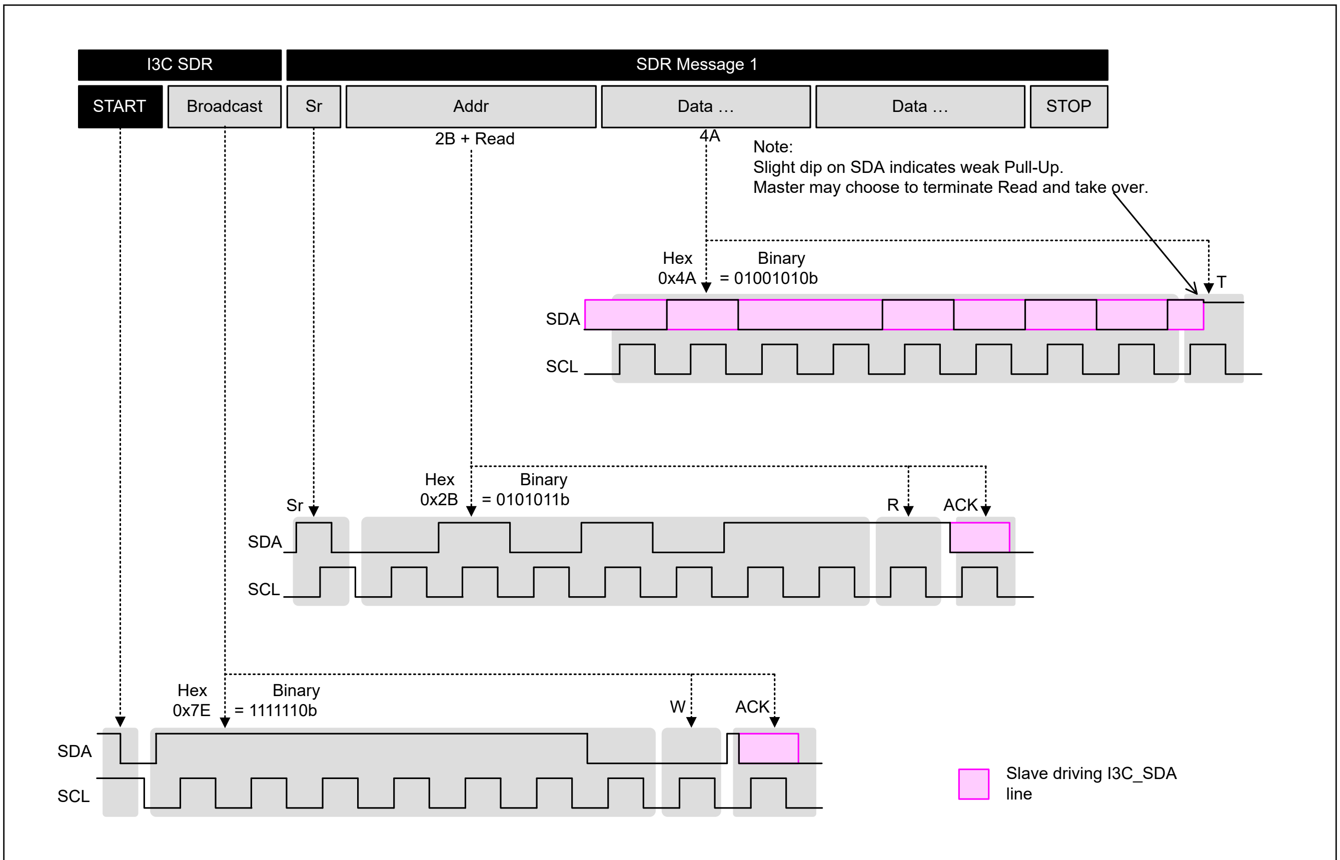


Figure 40.71 Example communication using I3C coding SDR with broadcast (0x7E)

Figure 40.72 illustrates example communication using I3C Single Data Rate (SDR) coding without Broadcast (0x7E). It shows the Master reading a byte of data from the Slave at Address 0x2B in SDR Mode. From the Bus Free Condition, The Master then issues a START, then the Address of the Slave (0x2B) it wants to read followed by RnW (1 for Read).

The Master then turns on a pull-up resistor and goes to Open Drain, allowing the Slave to acknowledge by pulling the I3C_SDA line Low. At this point, the Master continues to toggle the I3C_SCL line and release the I3C_SDA line, allowing the Slave to drive SDA to send one byte of data (0x4A) followed by T. T = 1 informs the Master that there is additional data, whereas T = 0 signals the end. Here there is additional data, so the Slave drives SDA High until SCL goes High, at which time it releases SDA. The Master has the option of holding SDA High with a weak pull-up, which signals to the Slave that the Master allows another byte to be transmitted, or to pull SDA Low (while SCL is High – hence a Repeated START), which would signal to the Slave that the Master has terminated the Read and is taking over.

SDR Mode is backwards compatible with Legacy I2C Devices, because the High time of an SCL pulse is always less than 50ns and therefore SCL will always appear to be Low because of the I2C 50ns Spike Filter.

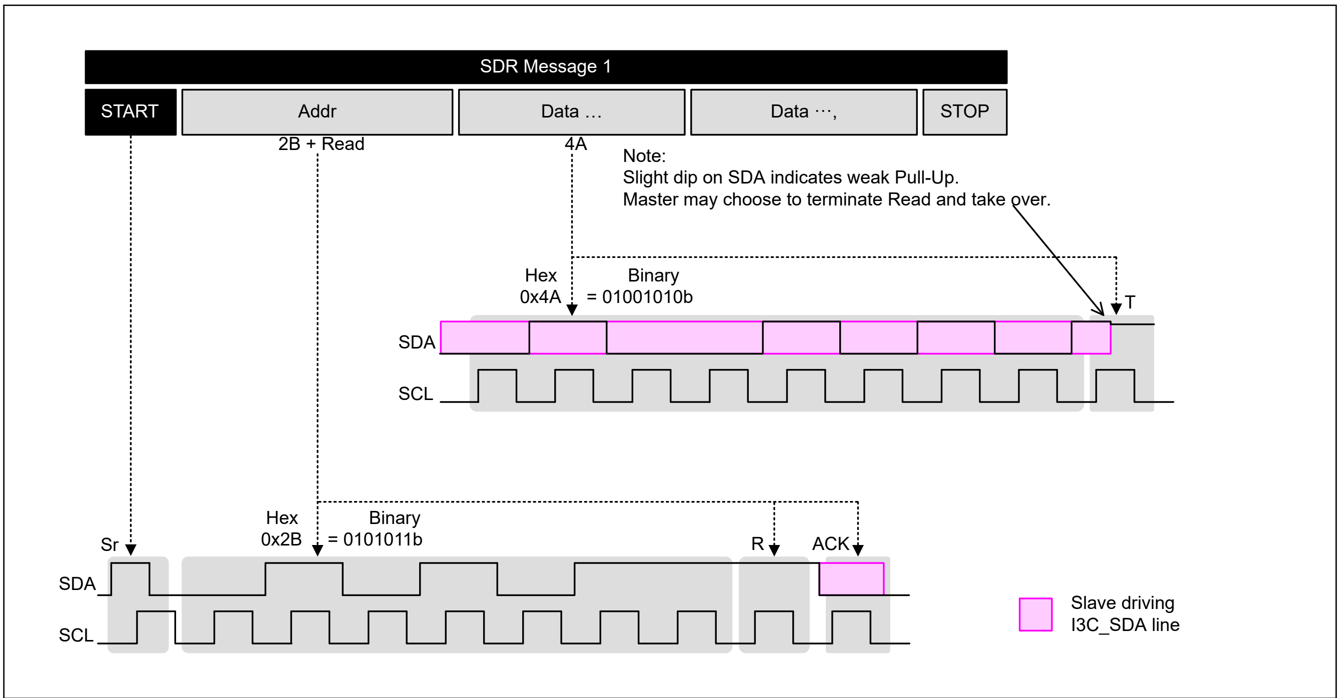


Figure 40.72 Example communication using I3C coding SDR without broadcast (0x7E)

Figure 40.73 shows the Master issuing a CCC Direct Command to a single Slave. This particular command (GETPID) reads the Provisional ID of a Slave.

From the Bus Free Condition, the Master issues a START by driving the I3C_SDA line Low while keeping the I3C_SCL line High. It then issues the Broadcast Address (0x7E) followed by RnW (0 for Write). Then the Master turns on a pull-up resistor and goes to Open Drain. All Slaves ACK by pulling SDA Low (in the Figure, pink fill means the Slaves are in control of SDA at this time). The Master then issues the Direct Common Command Code for GETPID (0x8C) followed by parity bit T (odd parity = 0 for 0x8C) then the 7-bit Dynamic Address of the Slave (chosen arbitrarily here to be 0x2B) followed by a RnW bit (1 for Read). Then the Master turns on a pull-up resistor and goes to Open Drain, allowing the Slave at Address 0x2B to ACK by pulling SDA Low, which tells the Master that the Slave Acknowledges the command and will comply. (Alternatively, the Slave may NACK by not pulling SDA Low, which would inform the Master that the Slave will not comply – in this case, that an error occurred.) Following the ACK the Slave outputs its 48-bit PID one byte at a time, and then the Master issues a Repeated START (this part of the waveform sequence is not shown in the Figure).

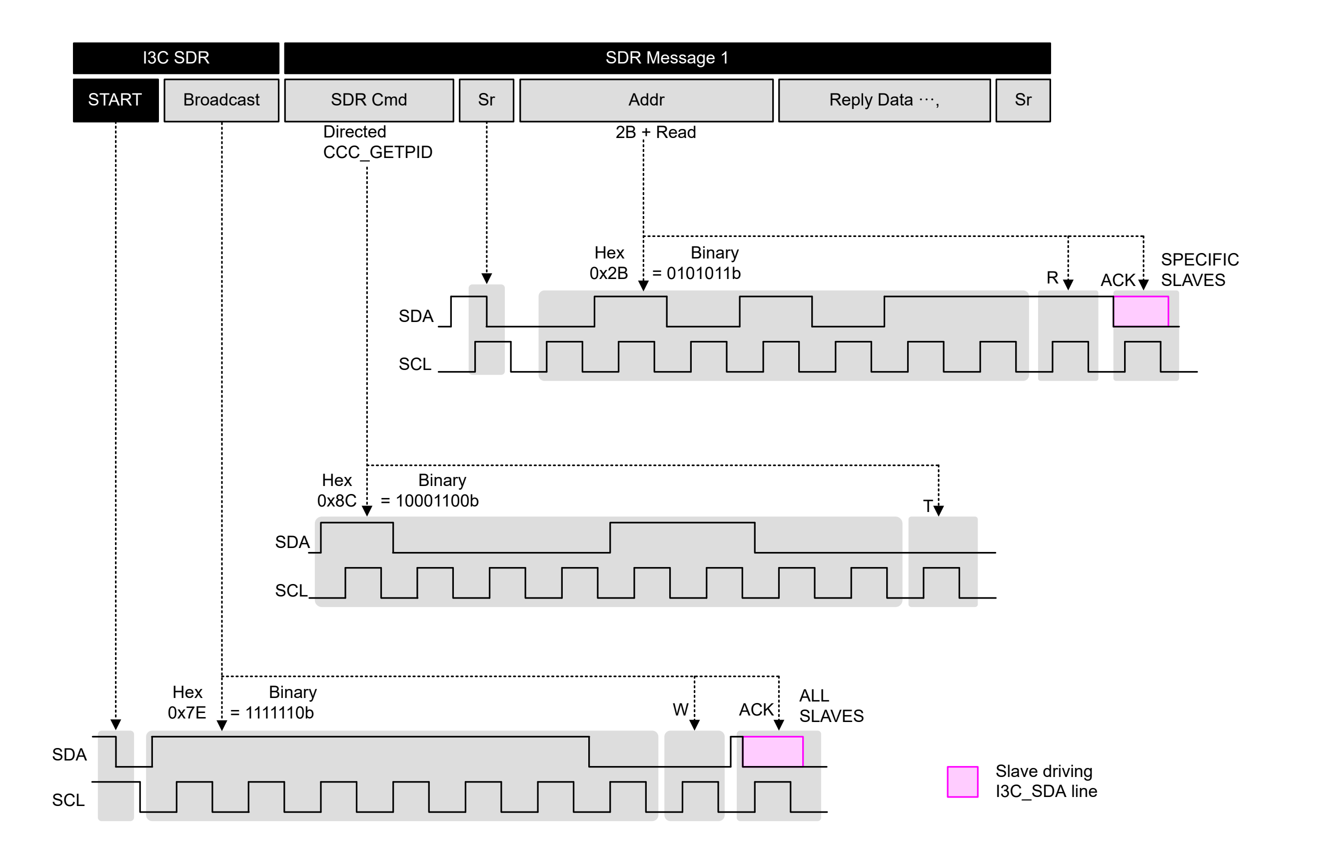


Figure 40.73 Example communication using I3C coding SDR with CCC direct addressing

Figure 40.74 illustrates example SDR communication with a CCC Broadcast command. The command used in this example sets the Maximum Read Length of all Slaves to 43 bytes (0x002B).

From the Bus Free Condition, the Master issues a START by driving the I3C_SDA line Low while keeping the I3C_SCL line High. It then issues the Broadcast Address (0x7E) followed by RnW (0 for Write). Then the Master turns on a pull-up resistor and goes to Open Drain. All Slaves ACK by pulling SDA Low (in the Figure, pink fill means the Slaves are in control of SDA at this time). The Master then issues the Broadcast Common Command Code for SETMRL (0x09) followed by parity bit T (odd parity = 1 for 0x09), and then 2 data bytes (MSB first) to define the maximum number of bytes which can be read from a Slave in a single read operation. Each data byte is followed by a T bit (parity bit – odd parity). After this the Master issues a Repeated START.

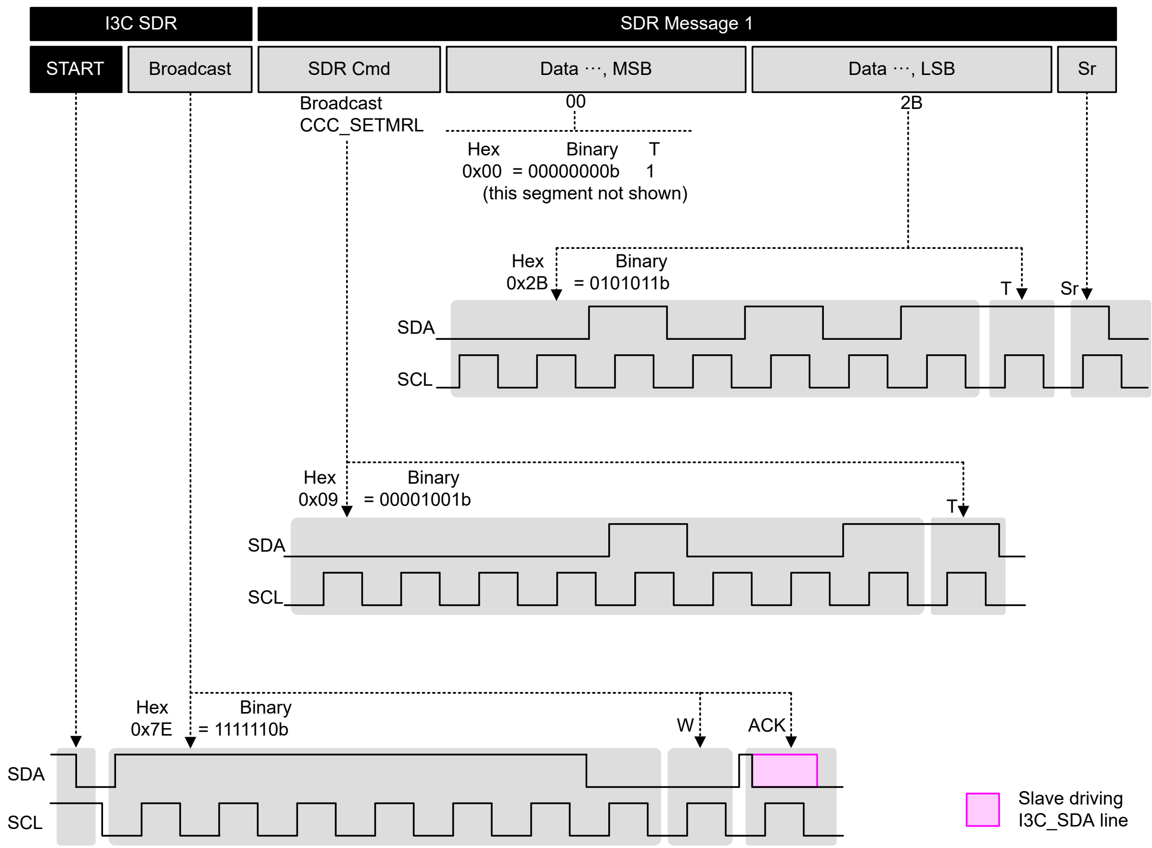


Figure 40.74 Example communication using I3C coding SDR with CCC broadcast

Figure 40.75 illustrates use of the HDR-DDR (Double Data Rate) Mode. It shows how the Master can change the Mode from SDR (Single Data Rate) Mode to HDR-DDR Mode, and a sample of the HDR-DDR data format.

From the Bus Free Condition, the Master issues a START by driving the I3C_SDA line Low while keeping the I3C_SCL line High. The Master then issues the Broadcast Address (0x7E) followed by RnW (0 for Write). Then the Master turns on a pull-up resistor and goes to Open Drain. All Slaves ‘ACK’ by pulling SDA Low (in the Figure, pink fill means the Slaves are in control of SDA at this time). The Master then issues the Broadcast Common Command Code for ENTHDR0 (0x20), followed by parity bit T (odd parity = 0 for 0x20). At this point, the Bus is in HDR-DDR Mode. In the HDR-DDR protocol, the I3C_SDA line is sampled on every SCL edge (both Low-to-High and High-to-Low transitions of SCL). The HDR-DDR Word consists of a 2-bit Preamble, followed by two bytes of data, followed by two parity bits. The waveform for the 5-bit CRC and following traffic is not shown in the Figure.

HDR-DDR Mode is backwards compatible with Legacy I^2C Devices, because the High time of an SCL pulse is always less than 50ns and as a result SCL will always appear to be Low because of the I^2C 50ns Spike Filter.

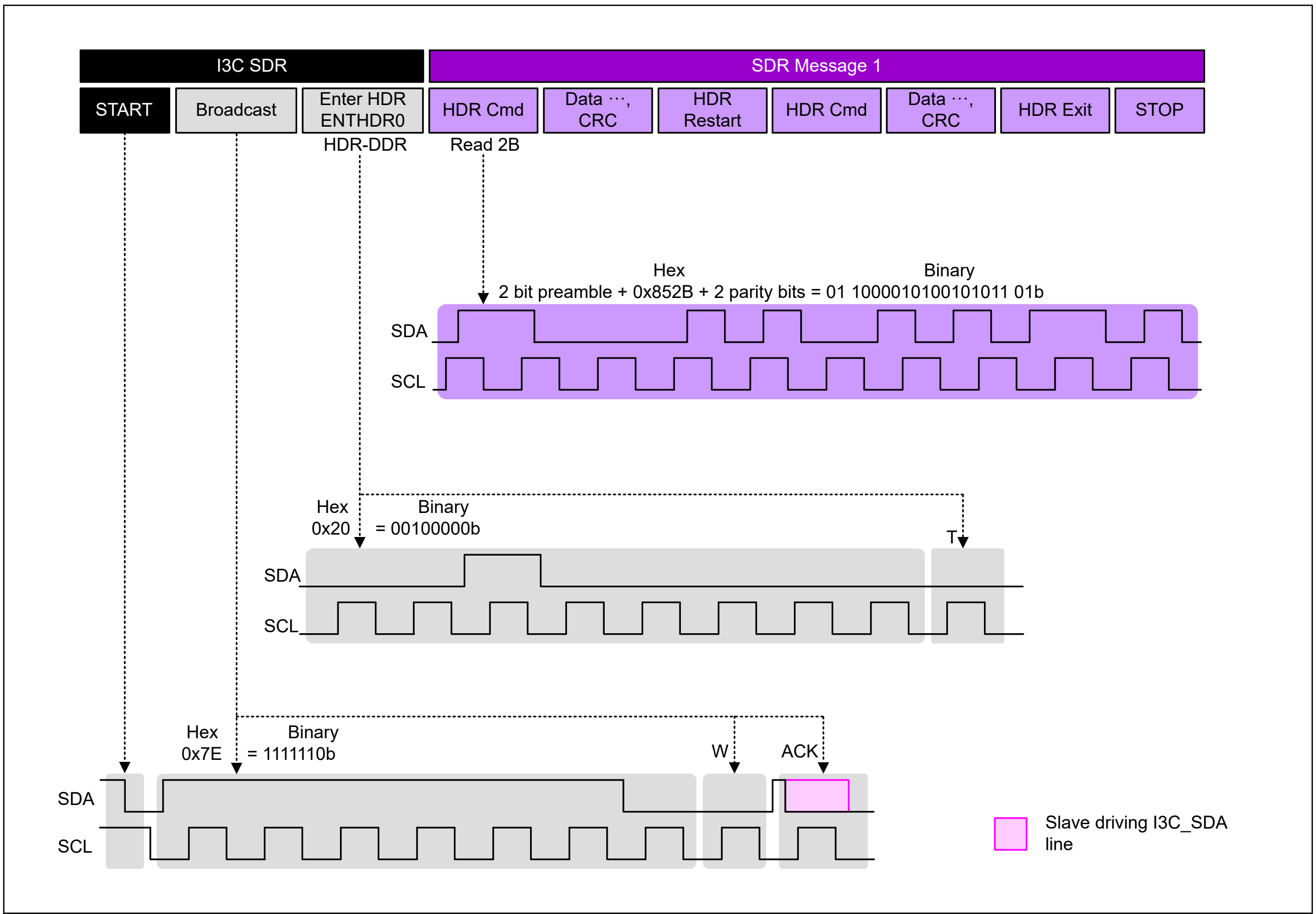


Figure 40.75 Example communication using HDR-DDR protocol

Figure 40.76 illustrates use of the HDR-TSL (Ternary Symbol Legacy) Mode, which is only used if a Legacy I²C Device is present on the Bus. It shows how the Master can change the Mode from SDR (Single Data Rate) Mode to HDR-TSL Mode, and a sample of the HDR-TSL data format.

From the Bus Free Condition, the Master issues a START by driving the I3C_SDA line Low while keeping the I3C_SCL line High. It then issues the Broadcast Address (0x7E) followed by RnW (0 for Write). Then the Master turns on a pull-up resistor and goes to Open Drain. All Slaves ACK by pulling SDA Low (in the Figure, pink fill means the Slaves are in control of SDA at this time). The Master then issues the Broadcast Common Command Code for ENTHDR2 (0x22) followed by parity bit ‘T’ (odd parity = 1 for 0x22). At this point, the Bus is in HDR-TSL Mode. In this protocol, the I3C_SCL line is more than just the clock. Both SDA transitions and SCL transitions carry data. The ternary conversion can be understood by considering that each octal digit is equivalent to two ternary Symbols. For example, for octal sequence 412255 we have: 4 = 3 + 1, 1 = 0 + 1, 2 = 0 + 2, 2 = 0 + 2, 5 = 3 + 2, 5 = 3 + 2. The clock pulse occurs when SCL transitions, when SDA transitions, or when both lines transition simultaneously.

HDR-TSL Mode is backwards compatible with Legacy I²C Devices, because the High time of an SCL pulse is always less than 50ns, and as a result SCL will always appear to be Low due to the I²C 50ns Spike Filter. Note that in the Figure, the thick blue High-to-Low transitions of SCL are dummy edges added in order to make sure that SCL pulses are always less than 50ns. High-to-Low SCL transitions that occur without a simultaneous SDA transition are always dummy transitions.

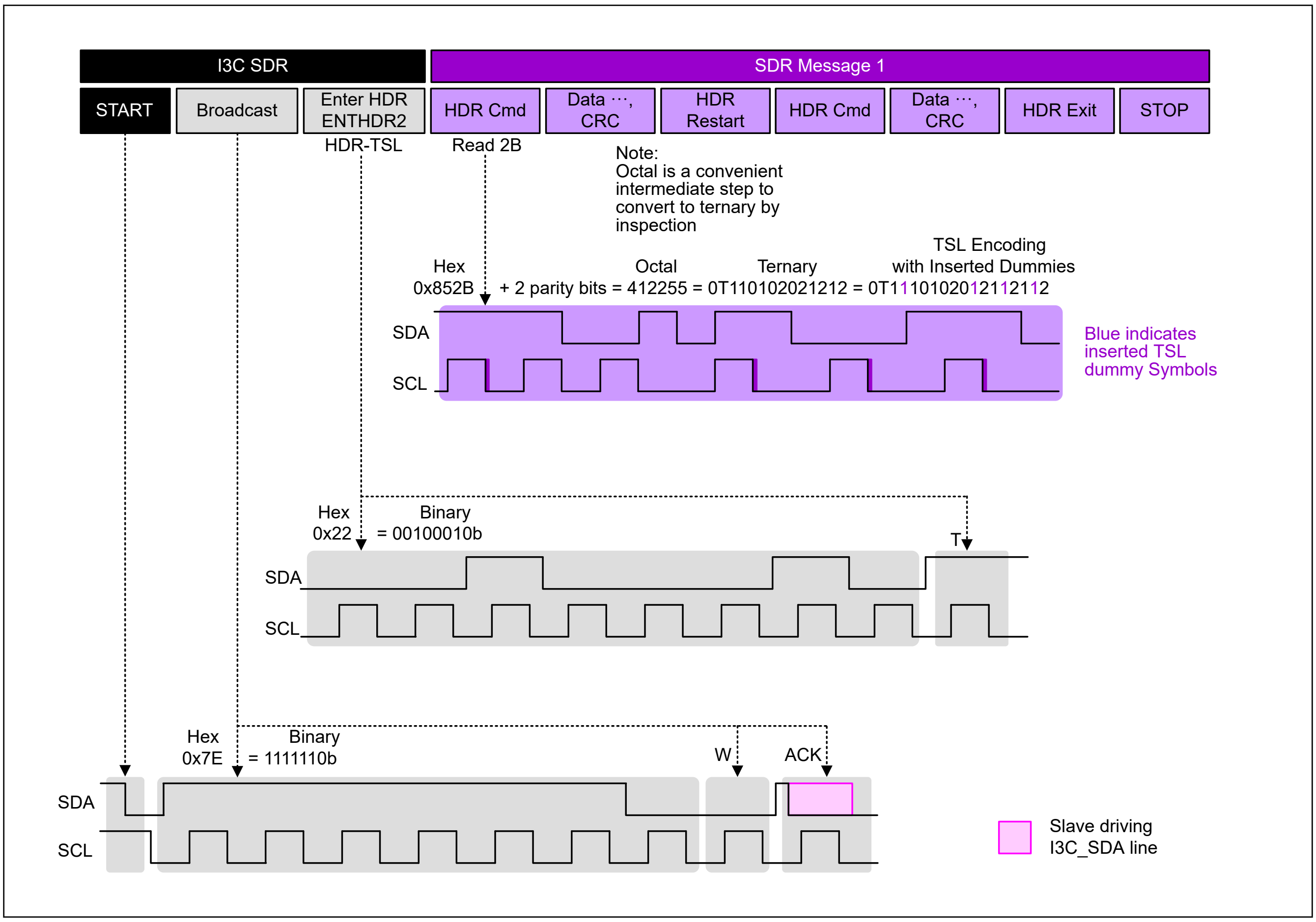


Figure 40.76 Example communication using HDR-TSL protocol

Figure 40.77 illustrates use of HDR-TSP (High Data Rate – Ternary Symbol for Pure Bus) Mode. This protocol is only used when there are no I^{2}C Devices on the Bus. The Figure shows how the Master can change the Mode from SDR (Single Data Rate) to HDR-TSP Mode, and a sample of the TSL data format.

From the Bus Free Condition, the Master issues a START by driving the I3C_SDA line Low while keeping the I3C_SCL line High. The Master then issues the Broadcast Address (0x7E) followed by RnW (0 for Write). Then the Master turns on a pull-up resistor and goes to Open Drain. All Slaves ACK by pulling SDA Low (in the Figure, pink fill means the Slaves are in control of SDA at this time). The Master then issues the Broadcast Common Command Code for ENTHDR1 (0x21) followed by parity bit 'T' (odd parity = 1 for 0x21). At this point, the Bus is in HDR-TSP Mode. In the HDRTSP protocol, the SCL is more than just the clock. Both SDA transitions and SCL transitions carry data. The ternary conversion can be understood by considering that each octal digit is equivalent to two ternary Symbols. For example, for octal sequence 412255 (4 = 3 + 1, 1 = 0 + 1, 2 = 0 + 2, 2 = 0 + 2, 5 = 3 + 2, 5 = 3 + 2). The clock pulse occurs when SCL transitions, when SDA transitions, or when both lines transition simultaneously.

HDR-TSP Mode supports the highest data-rate and uses the least energy per byte, however it is not backwards compatible with Legacy I^{2}C Devices since the High duration of an SCL pulse can exceed 50ns.

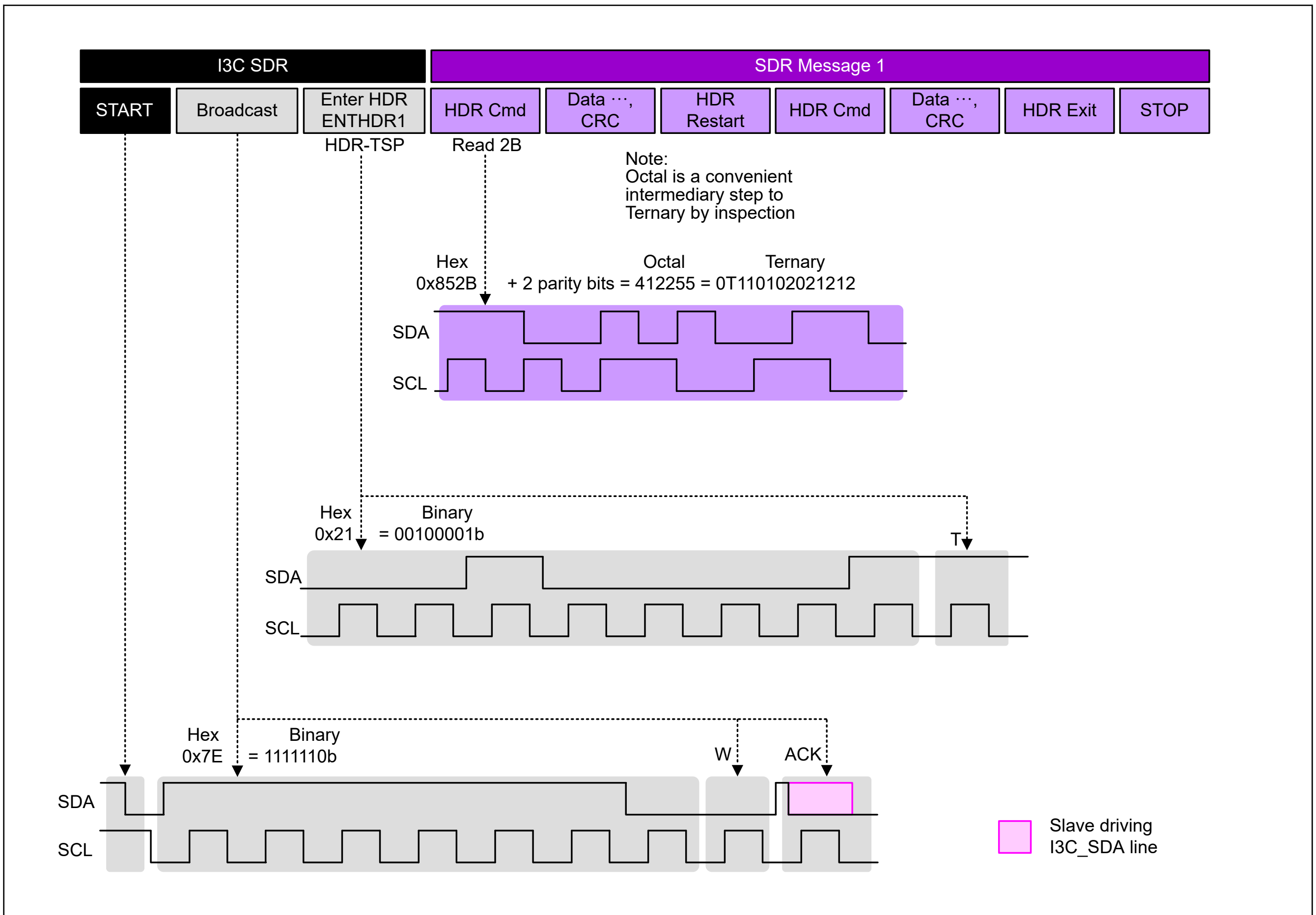


Figure 40.77 Example communication using HDR-TSP protocol

40.3.2.3.2 Bus Conditions

I3C defines three distinct conditions in which the I3C Bus shall be considered inactive: Bus Free, Bus Available, and Bus Idle (see Figure 40.78).

(1) Bus Free Condition

State on the I3C Bus where both the I3C_SCL line and the I3C_SDA line are High for at least the period set by BFRECDT.FRECYC[8:0] bit.

(2) Bus Available Condition [I3C mode]

State on the I3C Bus where both the I3C_SCL line and the I3C_SDA line are High for at least the period set by BAVLCDT.AVLCYC[8:0] bit.

A Slave may only issue a START Request (For example, for an In-Band Interrupt, or for a Master Handoff Request) after a Bus Available Condition.

(3) Bus Idle Condition [I3C mode]

State on the I3C Bus where both the I3C_SCL line and the I3C_SDA line are High for at least the period set by BIDLCDT.IDLCYC[17:0] bit.

A Slave may only issue a START Request after a Bus Idle Condition.

Specifications are as follows. IDLE needs to be the largest.

BFRECDT.FRECYC[8:0] < BAVLCDT.AVLCYC[8:0] < BIDLCDT.IDLCYC[17:0]

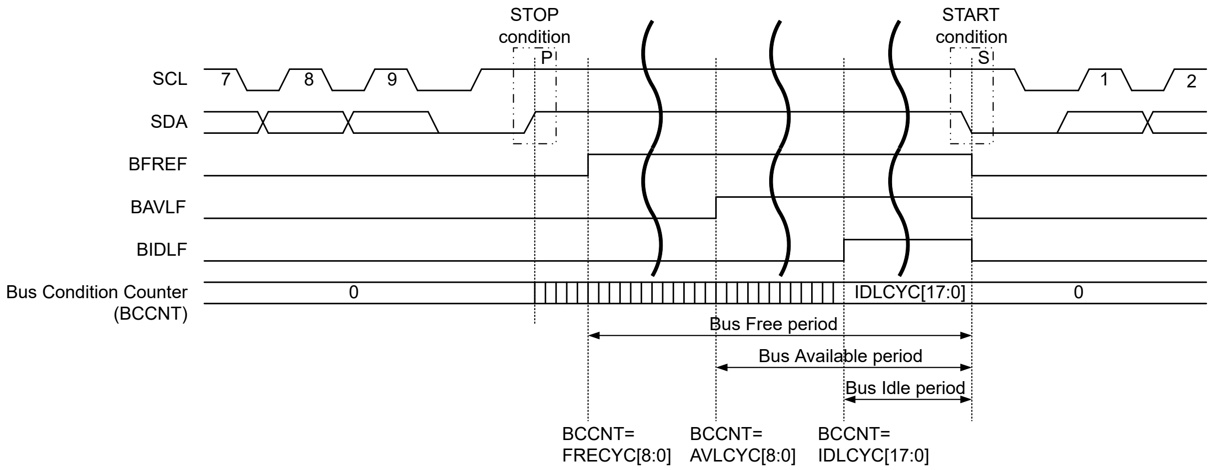


Figure 40.78 Bus conditions

40.3.2.3.3 START Condition / Repeated START Condition / STOP Condition Issuing Function

(1) Issuing a START Condition

I3C issues a START condition when the CNDCTL.STCND bit is set to 1.

Set the STCND bit to 1 (START condition issuance request) when the BCST.BFREF flag is set to 1 (bus free state).

I3C issues a START condition.

When a START condition is issued normally, I3C automatically shifts to the master transmit mode. A START condition is issued in the following sequence.

[START condition issuance]

- Drive the I3C_SDA line low (high level to low level).
- Ensure the time set in STDBR.SBRHO[7:0] and the START condition hold time.
- Drive the I3C_SCL line low (high level to low level).
- Detect low level of the I3C_SCL line and ensure the low-level period of I3C_SCL line set in STDBR.SBRLO[7:0].

(2) Issuing a Repeated START Condition

I3C issues a Repeated START condition when the CNDCTL.SRCND bit is set to 1.

When the SRCND bit is set to 1, a Repeated START condition issuance request is made and I3C issues a Repeated START condition when the BCST.BFREF flag = 0 (bus busy state) and the PRSST.CRMS bit = 1 (master mode).

A Repeated START condition is issued in the following sequence.

[Repeated START condition issuance]

- Release the I3C_SDA line.
- Ensure the low-level period of I3C_SCL line set in STDBR.SBRLO[7:0] or EXTBR.EBRLO[7:0].
- Release the I3C_SCL line (low level to high level).
- Detect a high level of the I3C_SCL line and ensure the time set in STDBR.SBRLO[7:0] or EXTBR.EBRLO[7:0] and the Repeated START condition setup time.
- Drive the I3C_SDA line low (high level to low level).
- Ensure the time set in STDBR.SBRHO[7:0] or EXTBR.EBRHO[7:0] or EXTBR.EBRHO[7:0] and the Repeated START condition hold time.
- Drive the I3C_SCL line low (high level to low level).

- Detect a low level of the I3C_SCL line and ensure the low-level period of I3C_SCL line set in STDBR.SBRLO[7:0] or EXTBR.EBRLO[7:0].

Note: When issuing Repeated START conditions request, write the slave address to NTDTBP0 after confirming CNDCTL.SRCND = 0. Data written in the period of CNDCTL.SRCND = 1 is not forwarded because retransmission condition before the occurrence.

To issue a Repeated START condition in Hs-mode, follow the steps below.

1. Wait for PRSTDBG.SCOLV=0.
2. Set EXTBR.EBRHO[7:0] to satisfy the hold time of the Repeated START condition.
3. Set the CNDCTL.SRCND bit to 1.
4. After confirming CNDCTL.SRCND=0, wait for PRSTDBG.SCOLV=0.
5. Set EXTBR.EBRHO [7: 0] according to the High period of the SCL clock in Hs-mode.
6. Write the slave address to NTDTBP0.

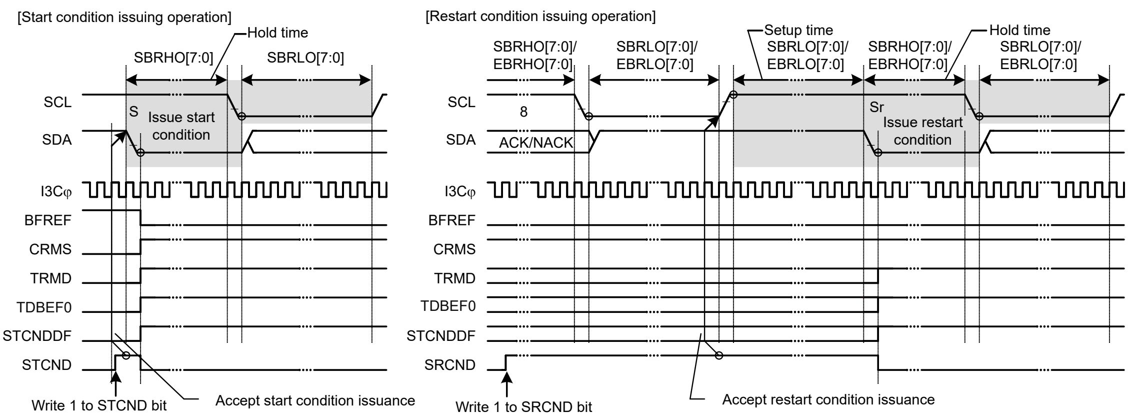


Figure 40.79 START condition / repeated START condition issue timing (STCND and SRCND bits)

Figure 40.80 shows the operation to issue a Repeated START condition after the master transmission.

[Repeated START condition issuance after the master transmission]

- Initial setting. For details, see section 40.3.3.1. Initial Setting Flow.
- Read the BFREF flag in BCST to check that the bus is open, and then set the STCND bit in CNDCTL to 1 (START condition issuance request). Upon receiving the request, I3C issues a START condition. At the same time, the BFREF flag is automatically set to 0 and the STCNDDF flag in BST is automatically set to 1 and the STCND bit is automatically set to 0. At this time, if the START condition is detected and the internal levels for the SDA output state and the levels on the I3C_SDA line have matched while the STCND bit = 1, I3C recognizes that issuing of the START condition as requested by the STCND bit has been successfully completed, and CRMS and TRMD bits in PRSST is automatically set to 1, placing I3C in master transmit mode. The NTST.TDBEF0 flag is also automatically set to 1 in response to setting of the TRMD bit to 1.
- Check that the NTST.TDBEF0 flag = 1, and then write the value for transmission (the slave address and the R/W# bit) to NTDTBP0. Once the data for transmission are written to NTDTBP0, the TDBEF0 flag is automatically set to 0, the data are transferred from NTDTBP0, and the TDBEF0 flag is again set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRMD bit is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, I3C continues in master transmit mode. Since the BST.NACKDF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to CNDCTL.SPCND bit to issue a STOP condition. For data transmission with an address in the 10-bit format, start by writing 1111 0, the 2 higher-order bits of the slave address, and W to NTDTBP0 as the first address transmission. Then, as the second address transmission, write the 8 lower-order bits of the slave address to NTDTBP0.

- After confirming that the NTST.TDBEF0 flag = 1, write the data for transmission to the NTDTBP0 register. I3C automatically holds the I3C_SCL line low until the data for transmission are ready, a Repeated START condition is issued or a STOP condition is issued.
- After all bytes of data for transmission have been written to the NTDTBP0 register, wait until the value of the BST.TENDF flag returns to 1, and then, after check that the BST.STCNDDF flag = 1, set the BST.STCNDDF flag to 0.
- Set the SRCND bit in CNDCTL to 1 (Repeated START condition issuance request). Upon receiving the request, I3C issues a Repeated START condition.
- After check that the BST.STCNDDF flag = 1, write the value for transmission (the slave address and the R/W# bit) to NTDTBP0.

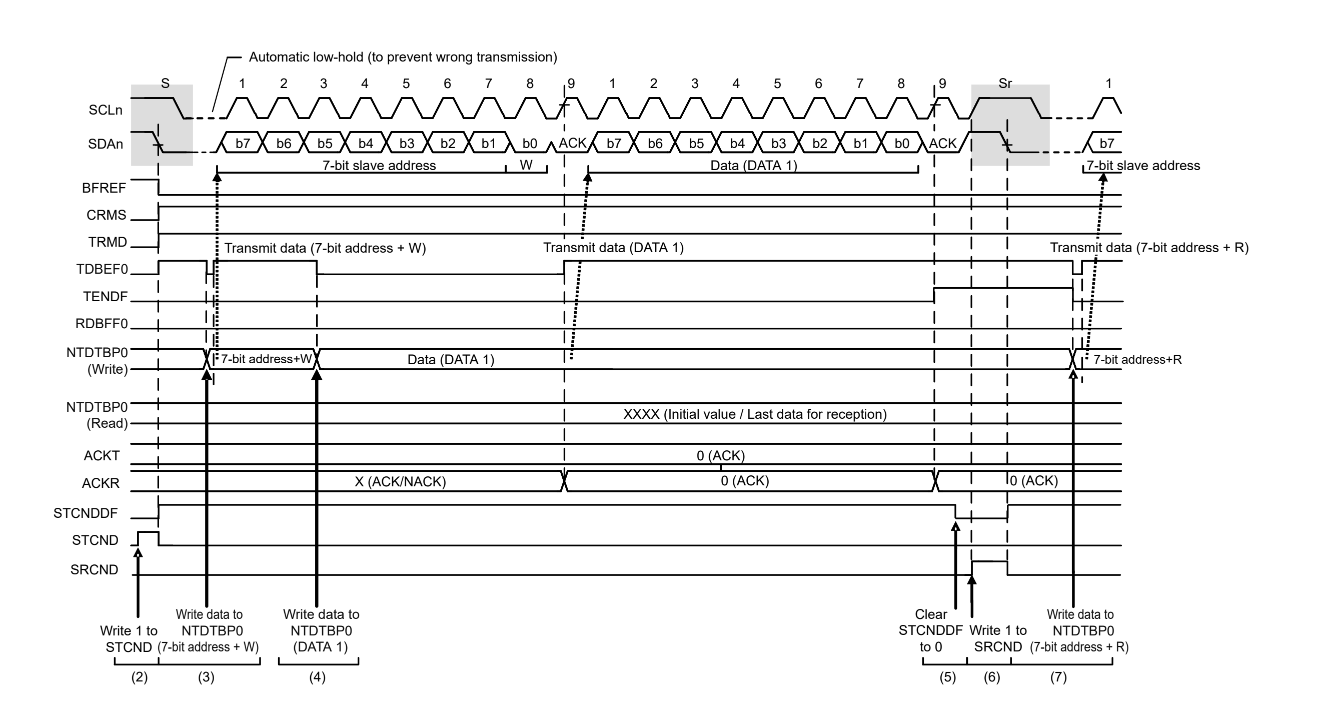


Figure 40.80 Repeated START condition issuance after the master transmission timing

(3) Issuing a STOP Condition

I3C issues a STOP condition when the SPCND bit in CNDCTL is set to 1.

When the SPCND bit is set to 1, a STOP condition issuance request is made and I3C issues a STOP condition when the BCST.BFREF flag = 0 (bus busy state) and the PRSST.MST bit = 1 (master mode).

A STOP condition is issued in the following sequence.

[STOP condition issuance]

- Drive the I3C_SDA line low (high level to low level).
- Ensure the low-level period of I3C_SCL line set in STDBR.SBRLO[7:0] or EXTBR.EBRLO[7:0].
- Release the I3C_SCL line (low level to high level).
- Detect a high level of the I3C_SCL line and ensure the time set in STDBR.SBRHO[7:0] or EXTBR.EBRHO[7:0] and the STOP condition setup time.
- Release the I3C_SDA line (low level to high level).
- Ensure the time set in BFRECDT.FRECYC[8:0] and the bus free time.
- Set the BFREF flag to 1 (to release the bus mastership).

Note: To issue a STOP condition in Hs-mode, follow the steps below.

1. Wait for PRSTDBG.SCOLV=0.

2. Set EXTBR.EBRHO[7:0] to satisfy the setup time for the STOP condition.
3. Set the CNDCTL.SPCND bit to 1.
4. Wait for CNDCTL.SPCND=0.
5. Set EXTBR.EBRHO [7:0] according to the High period of the SCL clock in Hs-mode.

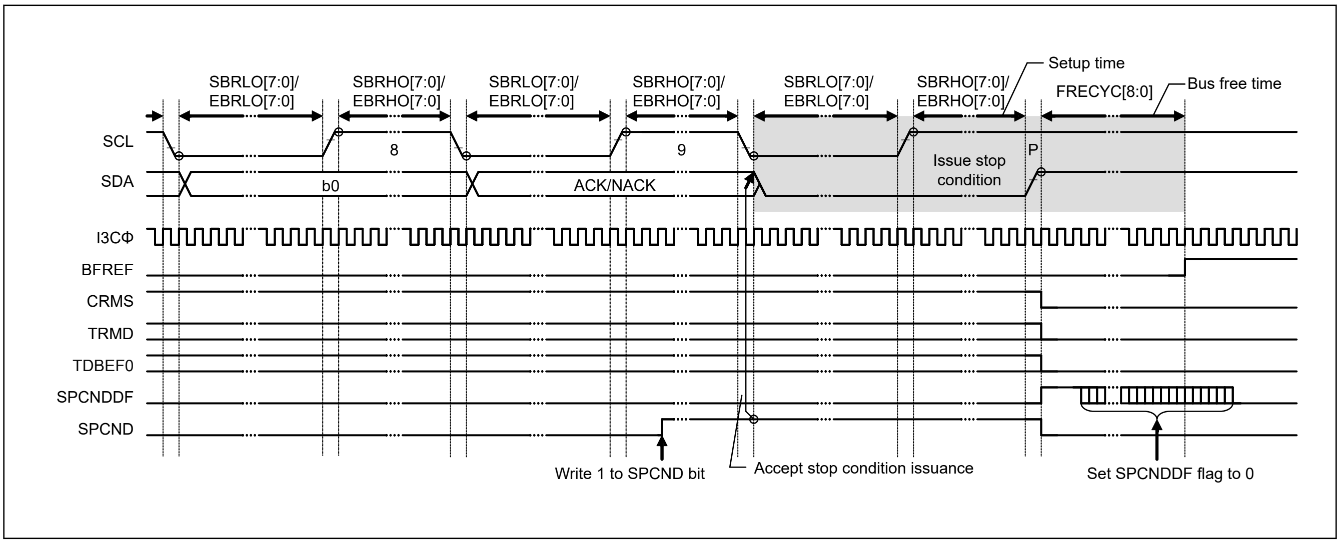


Figure 40.81 STOP condition issue timing (SPCND bit)

40.3.2.3.4 Address Match Detection

I3C can set three unique slave addresses in addition to the general call address and host address, and also can set 7- bit or 10- bit slave addresses.

(1) Slave-Address Match Detection [I^{2}C mode]

I3C can set three unique slave addresses, and has a slave address detection function for each unique slave address.

When the SVCTL.SVAEy bit (y = 0 to 2) is set to 1, the slave addresses set in the SVDVADy register (y = 0 to 2) can be detected.

When I3C detects a match of the set slave address, the corresponding SVST.SVAF[y] flag (y = 0 to 2) is set to 1 at the rising edge of the ninth SCL clock cycle, and the NTST.RDBFF0 flag or the NTST.TDBEF0 flag is set to 1 by the following R/W# bit. This causes a Normal Rx Data buffer full interrupt (I3C_RX) or Normal Tx Data buffer empty interrupt (I3C_TX) to be generated. The SVAF[y] flag is used to identify which slave address has been specified.

Figure 40.82 to Figure 40.84 show the SVAF[y] flag set timing in three cases.

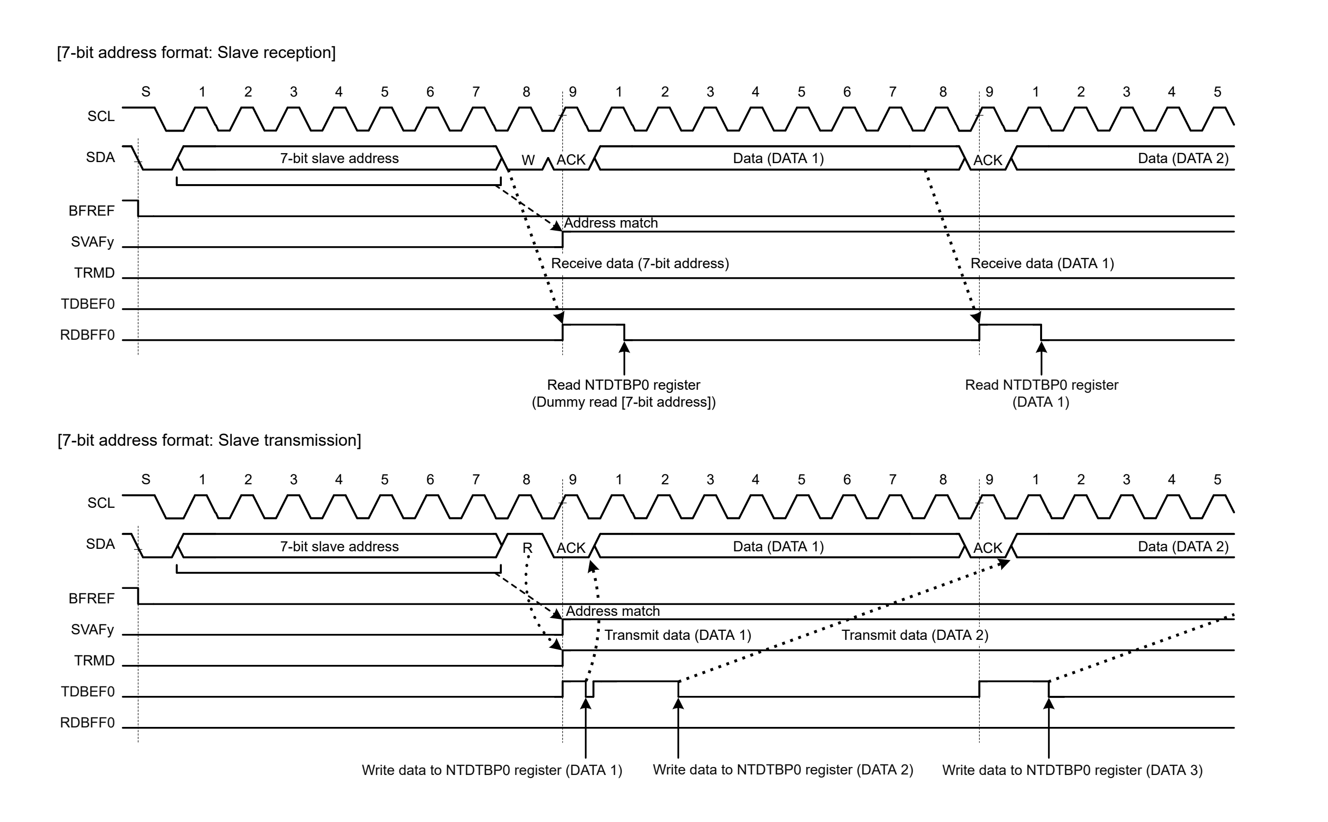


Figure 40.82 SVAFy flag set timing with 7-bit address format selected

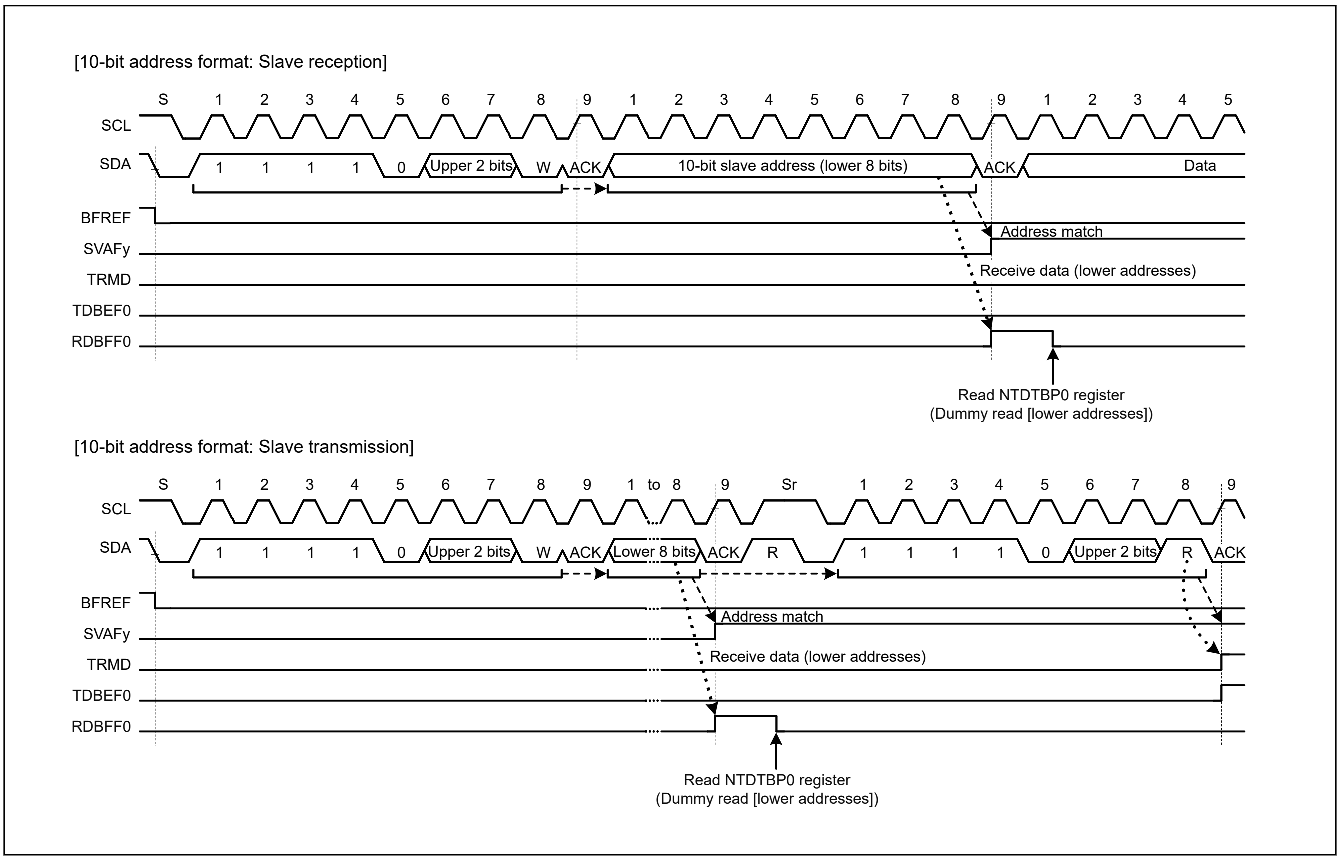


Figure 40.83 SVAFy flag set timing with 10-bit address format selected

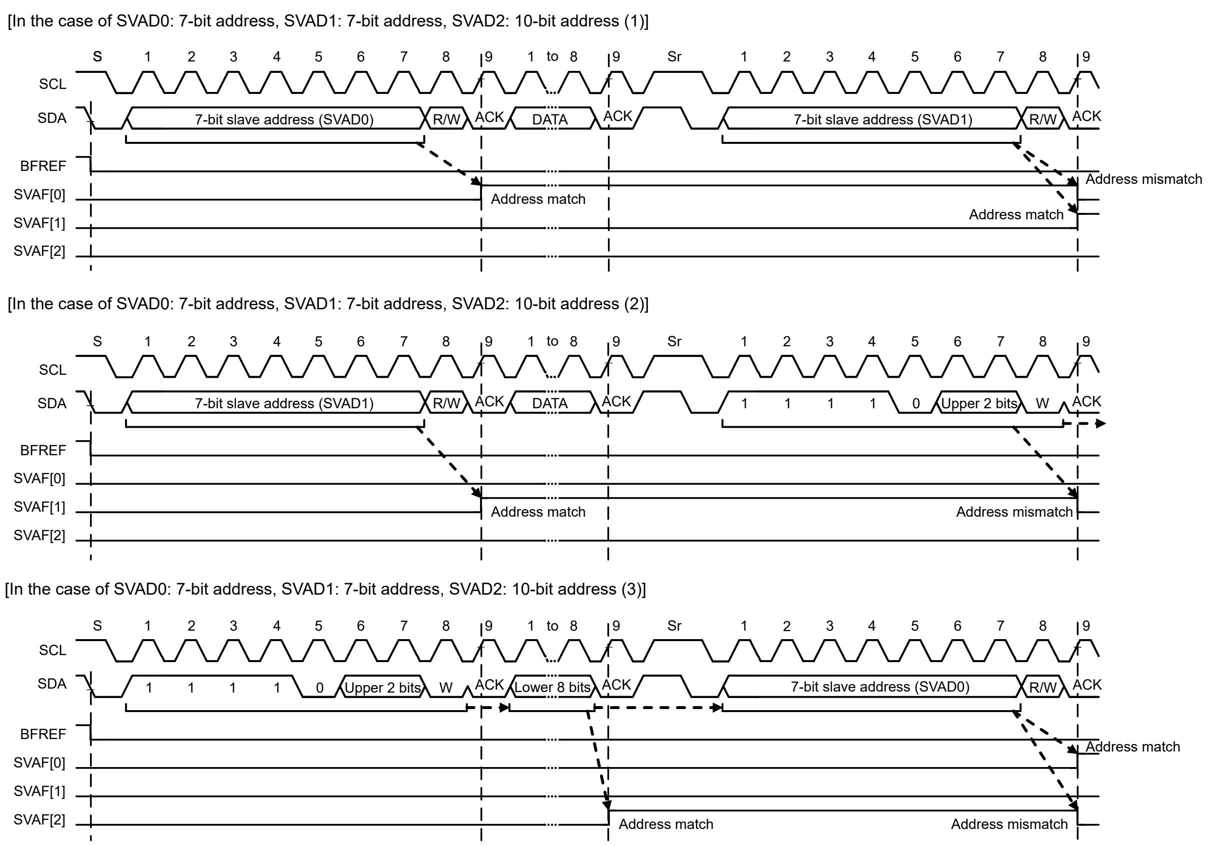


Figure 40.84 SVAFy flag set/clear timing with 7-bit/10-bit address formats mixed

(2) Detection of the General Call Address [I²C mode]

I3C has a facility for detecting the general call address (0000 000 + 0 (write)). This is enabled by setting the SVCTL.GCAE bit to 1.

If the address received after a START or Repeated START condition is issued is 0000 000 + 1 (read) (start byte), I3C recognizes this as the address of a slave device with an all-zero address but not as the general call address.

When I3C detects the general call address, both the SVST.GCAF flag and the NTST.RDBFF0 flag are set to 1 on the rising edge of the ninth cycle of SCL clock. This leads to the generation of a Normal Rx Data buffer full interrupt (I3C_RX). The value of the GCAF flag can be confirmed to recognize that the general call address has been transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

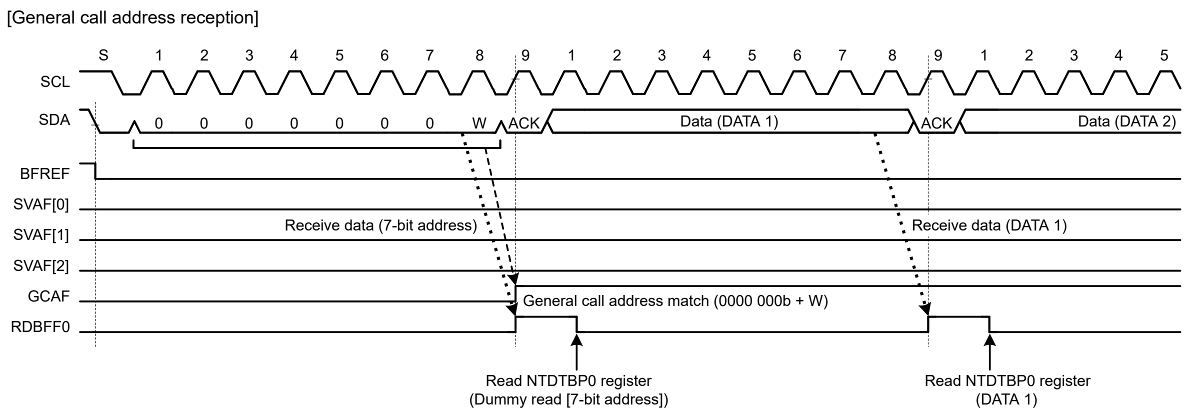


Figure 40.85 Timing of GCAF flag setting during reception of general call address

(3) Device-ID Address Detection [I²C mode]

I3C module has a facility for detecting device-ID addresses conformant with the I²C-bus specification (Rev.03). When I3C receives 1111 100 as the first byte after a START condition or Repeated START condition was issued with the SVCTL.DVIDE bit set to 1, I3C recognizes the address as a device ID, sets the SVST.DVIDF flag to 1 on the rising edge of the ninth SCL clock cycle when the following R/W# bit = 0, and then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, I3C sets the corresponding SVST.SVAF[y] flag (y = 0 to 2) to 1.

After that, when the first byte received after a START or Repeated START condition is issued matches the device ID address (1111 100) again and the following R/W# bit = 1, I3C does not compare the second and subsequent bytes and sets the NTST.TDBEF0 flag to 1.

In the device-ID address detection function, I3C sets the DVIDF flag to 0 if a match with I3C's own slave address is not obtained or a match with the device ID address is not obtained after a match with I3C's own slave address and the detection of a Repeated START condition. If the first byte after detection of a START or Repeated START condition matches the device ID address (1111 100) and the R/W# bit = 0, I3C sets the DVIDF flag to 1 and compares the second and subsequent bytes with I3C's slave address. If the R/W# bit = 1, the DVIDF flag holds the previous value and I3C does not compare the second and subsequent bytes. Therefore, the reception of a device-ID address can be checked by reading the DVIDF flag after confirming that TDBEF0 flag = 1.

Furthermore, prepare the device-ID fields (3 bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal data for transmission. For details of the information that must be included in device-ID fields, contact NXP Semiconductors.

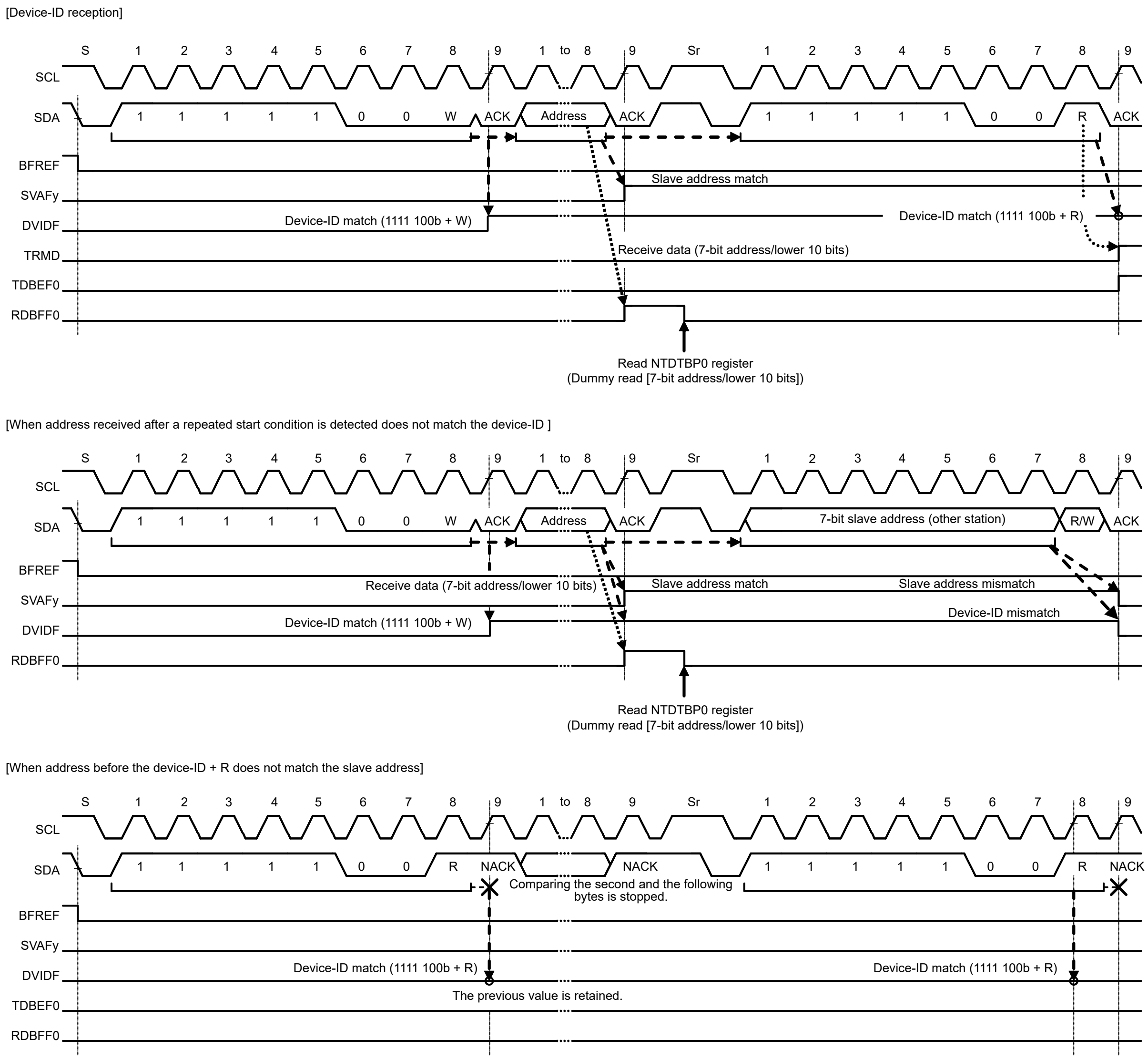


Figure 40.86 SVAFy/DVIDF flag set/clear timing during reception of device-ID

(4) Host Address Detection [I^{2}C mode]

I3C has a function to detect the host address while the SMBus is operating. When the SVCTL.HOAE bit is set to 1 while the BFCTL.SMBS bit = 1, I3C can detect the host address (0001 000) in slave receive mode (bits CRMS and TRMD in the PRSST register = 00).

When I3C detects the host address, the SVST.HOAF flag is set to 1 at the rising edge of the ninth SCL clock cycle, and at the same time, the NTST.RDBFF0 flag is set to 1 when the R/W# bit = 0 (Wr bit). This causes a Normal Rx Data buffer full interrupt (I3C_RX) to be generated. The HOAF flag is used to recognize that the host address was sent from the smart battery or other devices.

If the bit following the host address (0001 000) is an Rd bit (R/W# bit = 1), I3C can also detect the host address. After the host address is detected, I3C operates in the same manner as normal slave operation.

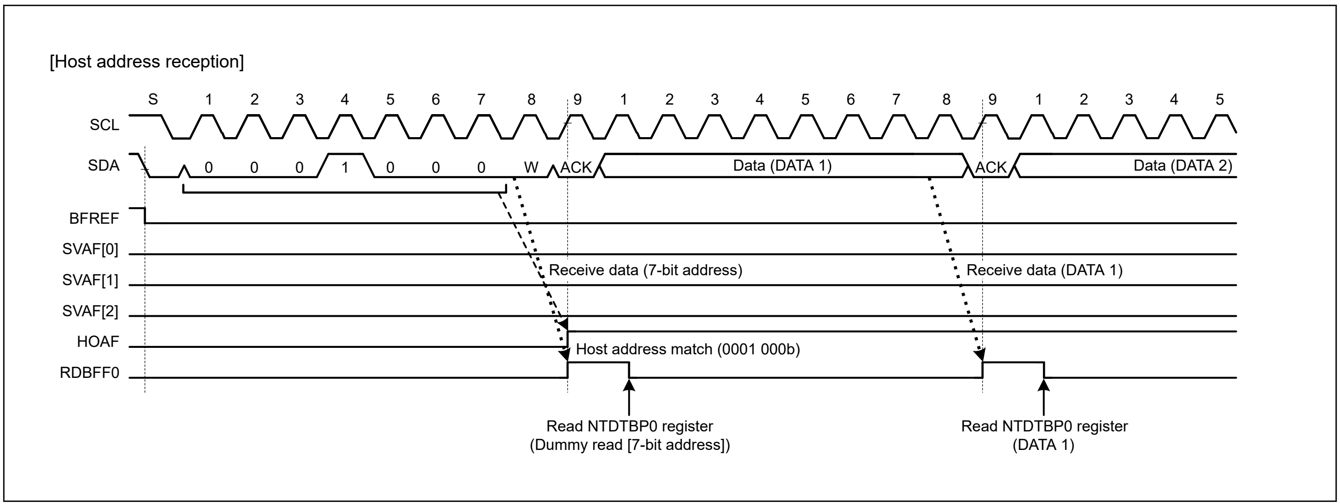


Figure 40.87 HOAF flag set timing during reception of host address

(5) Hs-mode master code Detection [I2C mode]

IIC has a facility for detecting the Hs-mode master code (0000 1XXXb). When IIC receives the Hs-mode master code (0000 1XXXb) as the first byte after a START condition was issued with the SVCTL.HSMCE bit set to 1, this module recognizes the address as the Hs-mode master code, sets the SVST.HSMCF flag to 1 on the rising edge of the ninth SCL clock cycle. The first byte after Repeated START after NACK response to Hs-mode master code is recognized as a slave address and compared with the slave address set by SVDVADy.SVAD[9:0] (y = 0 to 2). When IIC detects a match of the set slave address, the corresponding SVST.SVAF[y] flag (y = 0 to 2) is set to 1 at the rising edge of the ninth SCL clock cycle, and the NTST.RDBFF0 flag or the NTST.TDBEF0 flag is set to 1 by the following R/W# bit. This causes a Normal Rx Data buffer full interrupt (I3C_RX) or Normal Tx Data buffer empty interrupt (I3C_TX) to be generated. The SVAF[y] flag is used to identify which slave address has been specified. The SVST.HSMCF flag is cleared to 0 when the STOP condition is detected.

Note: If the Hs-mode master code (0000 1XXXb) is received with the SVCTL.HSMCE bit set to 0, other patterns are ignored until the STOP condition is detected.

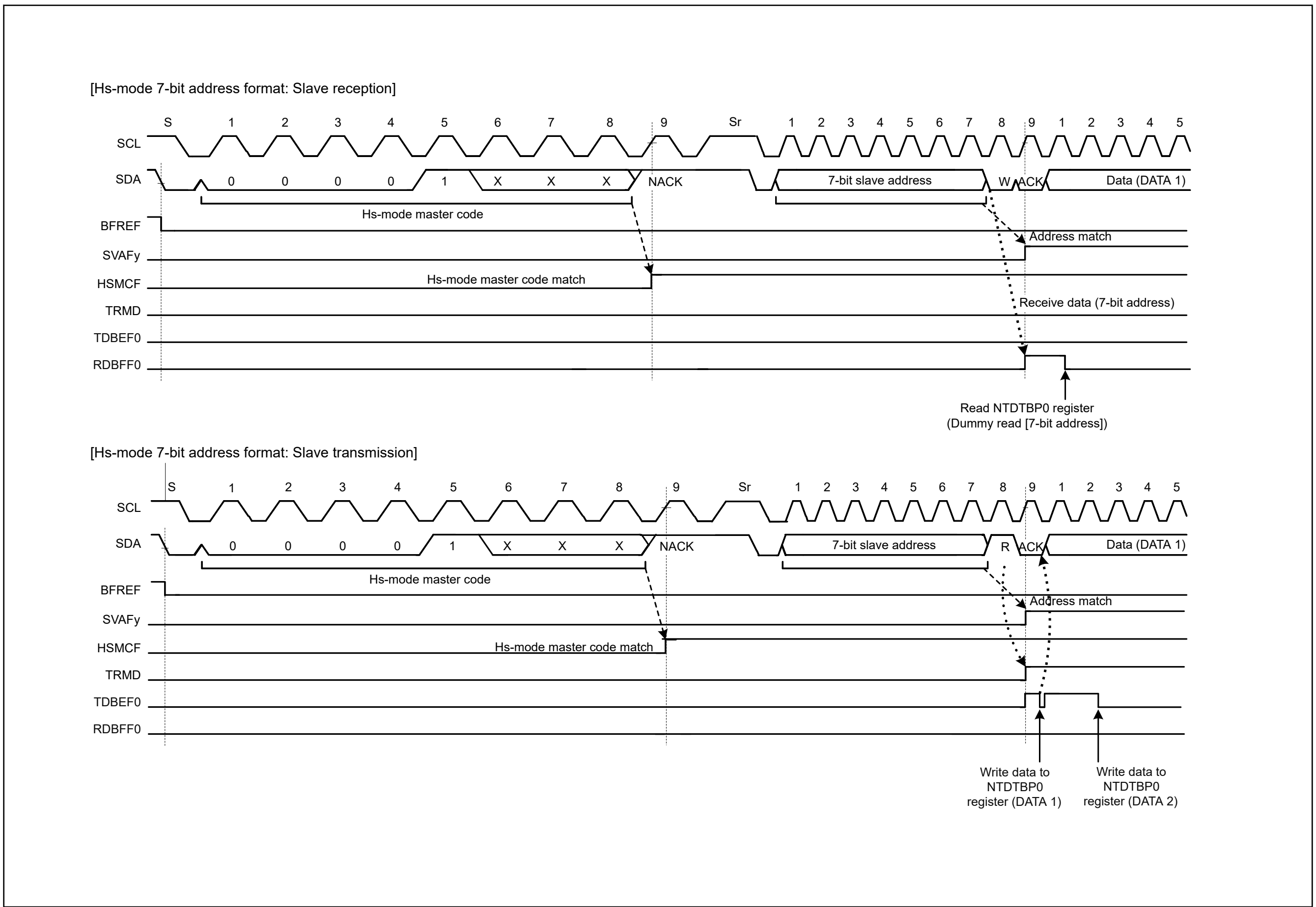


Figure 40.88 SVAFy/HSMCF Flag Set Timing during Reception of Hs-mode master code

(6) CCC detection function [I3C mode]

- In case of Broadcast CCC
  1. It receives Broadcast Address (0x7E) and R/W# = 0 after START condition or Repeated START.
  2. Respond to ACK.
  3. Receive Common Command Code (CCC).
  4. In accordance with the CCC, the following data is stored. (Storage destination: see Table 40.11)
  5. Store the Receive Status Descriptor into the Receive Status Queue.
- In case of Broadcast CCC (ENTDAA)
  1. It receives Broadcast Address (0x7E) and R/W# = 0 after START condition.
  2. Respond to ACK.
  3. Receive ENTDAA.
  4. If receives Broadcast Address (0x7E) and R/W# = 1 after Repeated START.
  5. When the Dynamic Address is not assigned, ACK response is done.
  6. This Provisional ID (SDCTPIDH[31:0], SDCTPIDL[15:0]), BCR (SVDCT.TBCRn) and DCR (SVDCT.TDCR[7:0]) are transmitted.
  7. When winning the arbitration in a transmission of the above Step 6, the dynamic address following that is received. When losing arbitration in a transmission of the above Step 6, processing of Step 6 is repeated from Step 4.
  8. When parity of the Dynamic Address is valid, ACK response is done.
  9. When parity of the Dynamic Address is invalid, NACK replies, and repeat the process from Steps 4 to 7.
  10. SDATBAS0.SDDYAD[7:0] is renewed and the SVDVAD0.SDYADV bit is set to 1.

11. Upon detecting the STOP condition, Store the Receive Status Descriptor into the Receive Status Queue.

- In case of Broadcast CCC (ENTHDR*)

1. It receives Broadcast Address (0x7E) and R/W# = 0 after START condition or Repeated START.
2. Respond to ACK.
3. Receive ENTHDR*.
4. Transit to HDR mode according to ENTHDR*.
5. Receives the HDR command word.
6. Compare the Dynamic Address in the HDR command word with the assigned Dynamic Address, and if they match, transmit / receive according to the Read / Write bit.
If they do not match, wait for HDR Restart Pattern or HDR Exit Pattern + STOP.
7. When HDR Restart Pattern or HDR Exit Pattern + STOP is received, Receive Status Descriptor is stored in Receive Status Queue.
For HDR Restart Pattern, repeat Steps 5 to 6.

- In case of Direct Write CCC

1. It receives Broadcast Address (0x7E) and R/W# = 0 after START condition or Repeated START.
2. Respond to ACK.
3. Receive Common Command Code (CCC).
4. Receive Dynamic Address and R/W# = 0 after Repeated START.
5. Compare the received Dynamic Address with the assigned Dynamic Address, and if it matches, I3C responds with ACK.
If they do not match, it responds with NACK and waits for Repeated START or STOP.
6. In accordance with the CCC, the following data is stored. (Storage destination: see Table 40.11)
7. Store the Receive Status Descriptor into the Receive Status Queue.

- In case of Direct Read CCC

1. It receives Broadcast Address (0x7E) and R/W# = 1 after START condition or Repeated START.
2. Respond to ACK.
3. Receive Common Command Code (CCC).
4. Receive Dynamic Address and R/W# = 1 after Repeated START.
5. Compare the received Dynamic Address with the assigned Dynamic Address, and if it matches, I3C responds with ACK.
If they do not match, it responds with NACK and waits for Repeated START or STOP.
6. Respond from SFR according to CCC. (Responding CCC: see Table 40.11)
7. Store the Receive Status Descriptor into the Receive Status Queue.

Table 40.11 Common command code operation (1 of 2)

Command Code	CCC Type	Command Name	With Data	Auto Response	Storage
0x00	Broadcast	ENEC	Yes	—	SFR
0x01	Broadcast	DISEC	Yes	—	SFR
0x02	Broadcast	ENTAS0	No	—	SFR
0x03	Broadcast	ENTAS1	No	—	SFR
0x04	Broadcast	ENTAS2	No	—	SFR
0x05	Broadcast	ENTAS3	No	—	SFR
0x06	Broadcast	RSTDAA	No	—	SFR
0x07	Broadcast	ENTDAA	Yes	Yes	SFR
0x08	Broadcast	DEFSLVS	Yes	—	FIFO

Table 40.11 Common command code operation (2 of 2)

Command Code	CCC Type	Command Name	With Data	Auto Response	Storage
0x09	Broadcast	SETMWL	Yes	—	SFR
0x0A	Broadcast	SETMRL	Yes	—	SFR
0x0B	Broadcast	ENTTM	Yes	—	SFR
0x20	Broadcast	ENTHDR0	No	—	n/a
0x21	Broadcast	ENTHDR1	No	—	n/a
0x22	Broadcast	ENTHDR2	No	—	n/a
0x28	Broadcast	SETXTIME	Yes	—	FIFO
0x29	Broadcast	SETAASA	No	—	SFR
0x80	Direct Write	ENEC	Yes	—	SFR
0x81	Direct Write	DISEC	Yes	—	SFR
0x82	Direct Write	ENTAS0	No	—	SFR
0x83	Direct Write	ENTAS1	No	—	SFR
0x84	Direct Write	ENTAS2	No	—	SFR
0x85	Direct Write	ENTAS3	No	—	SFR
0x86	Direct Write	RSTDAA	No	—	SFR
0x87	Direct Write	SETDASA	Yes	—	SFR
0x88	Direct Write	SETNEWDA	Yes	—	SFR
0x89	Direct Write	SETMWL	Yes	—	SFR
0x8A	Direct Write	SETMRL	Yes	—	SFR
0x8B	Direct Read	GETMWL	—	Yes	SFR
0x8C	Direct Read	GETMRL	—	Yes	SFR
0x8D	Direct Read	GETPID	—	Yes	SFR
0x8E	Direct Read	GETBCR	—	Yes	SFR
0x8F	Direct Read	GETDCR	—	Yes	SFR
0x90	Direct Read	GETSTATUS	—	Yes	SFR
0x91	Direct Read	GETACCMST	—	Yes	SFR
0x94	Direct Read	GETMXDS	—	Yes	SFR
0x95	Direct Read	GETHDRCAP	—	Yes	SFR
0x98	Direct Write	SETXTIME	Yes	—	FIFO
0x99	Direct Read	GETXTIME	—	Yes	SFR

40.3.2.3.5 Arbitration-Lost Detection [I^{2}C Mode]

In addition to the normal arbitration-lost detection function defined by the I^{2}C-bus specification, the I3C has functions to prevent double-issue of a start condition, to detect arbitration-lost during transmission of NACK, and to detect arbitration-lost in slave transmit mode.

(1) Master Arbitration-Lost Detection (MALE Bit)

The I3C drives the I3C_SDA line low to issue a start condition. However, if the I3C_SDA line has already been driven low by another master device issuing a start condition, this module causes arbitration to be lost, so priority is given to transfer by the other master device. Similarly, if the CNDCTL.STCND bit is set to 1 while the BCST.BFREF flag is 0 (bus busy state), arbitration is lost, so priority is given to transfer by the other master device. No start condition is issued in this case.

When a start condition is issued successfully, if the data for transmission including the address bits (the internal SDA output level) and the level on the I3C_SDA line do not match (the high output as the internal SDA output, that is, the I3C_SDA pin is in the high-impedance state) and the low level is detected on the I3C_SDA line, the I3C loses in arbitration.

I3C detects master arbitration-lost when the following conditions are met while the BSTE.ALE bit = 1 and the BFCTL.MALE bit = 1 (master arbitration-lost detection enabled).

If arbitration of mastership is lost, I3C immediately enters slave receive mode.

If a slave address (including the general call address) matches its own address at this time, I3C continues in slave operation.

[Conditions for master arbitration-lost]

- Non-matching of the internal level for output on SDA and the level on the I3C_SDA line after a START condition was issued by setting the CNDCTL.STCND bit to 1 while the BCST.BFREF flag was set to 1 (erroneous issuing of a START condition)
- Setting of the CNDCTL.STCND bit to 1 (START condition double-issue error) while the BFREF flag is set to 0

  Note: I3C does not issue a START condition.

- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the I3C_SDA line in master transmit mode (bits CRMS and TRMD in the PRSST register = 11b)

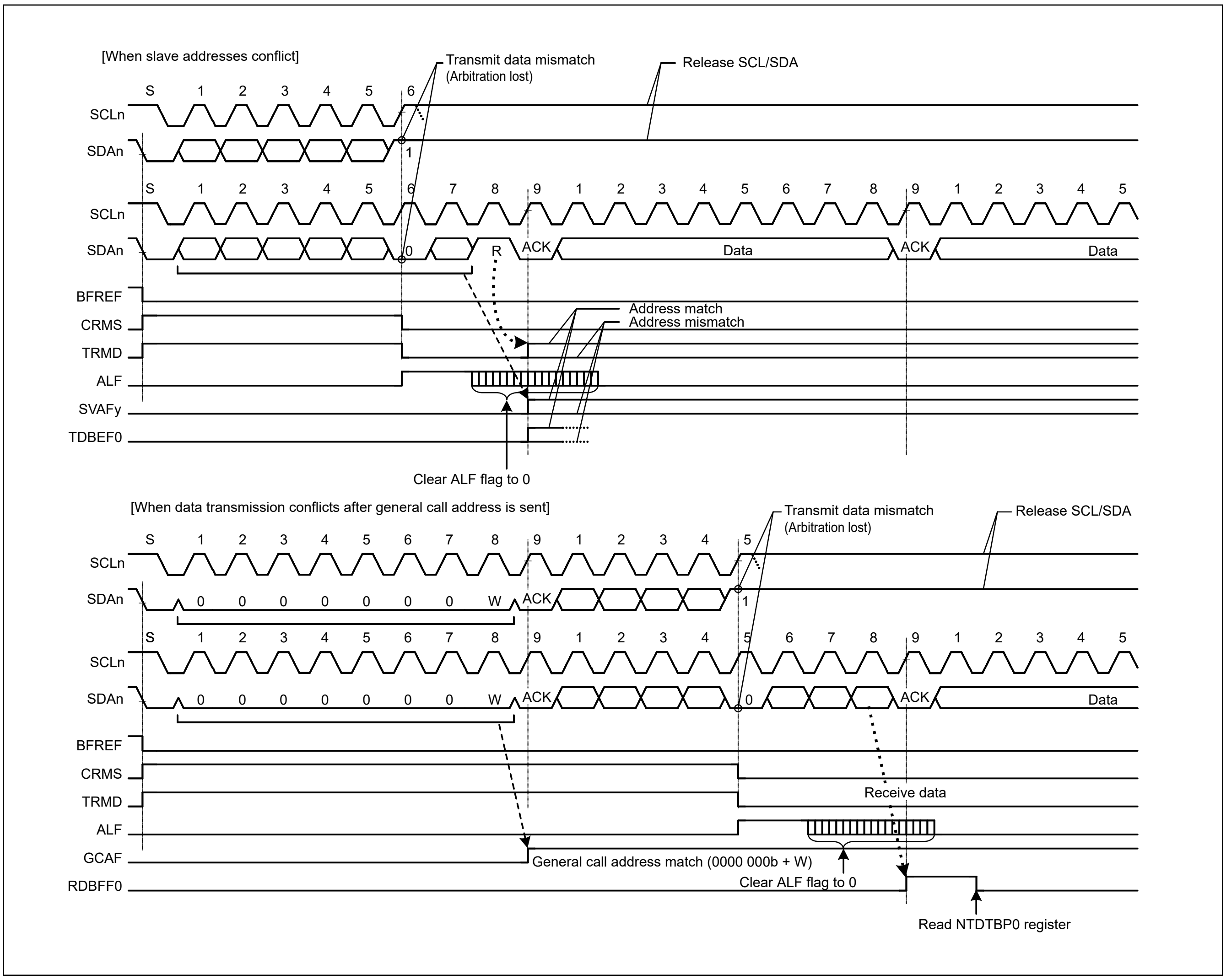


Figure 40.89 Examples of master arbitration-lost detection (MALE = 1)

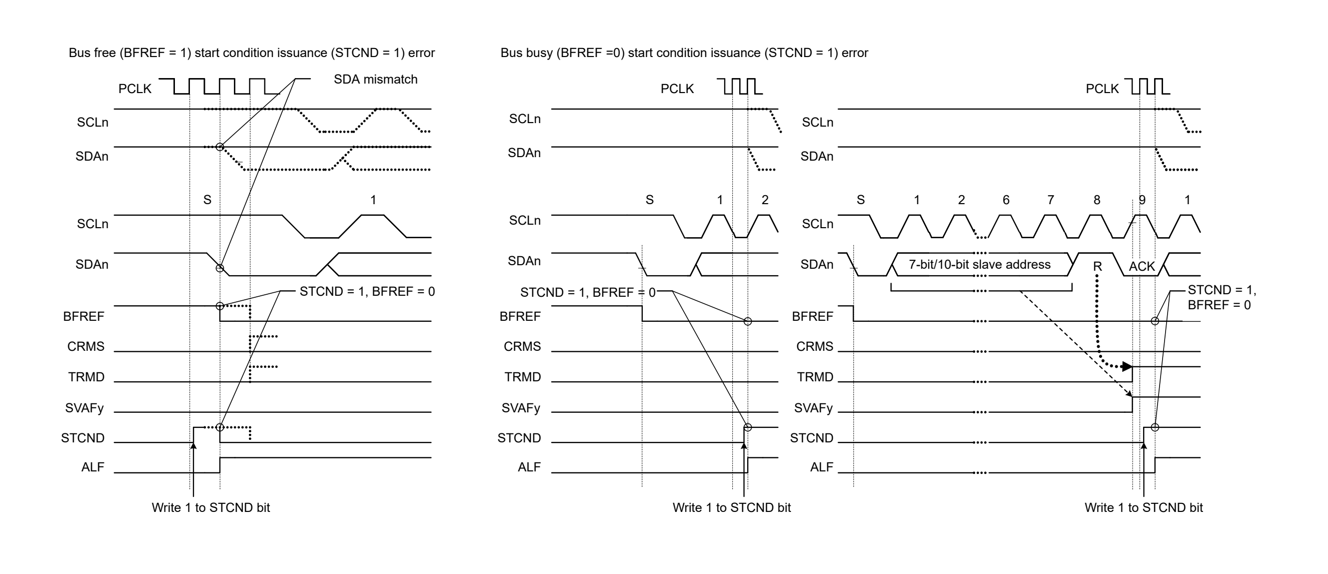


Figure 40.90 Arbitration-lost detection when a START condition is issued (MALE = 1)

(2) Arbitration-Lost Detection during NACK Transmission (NALE Bit)

The I3C has a function to cause arbitration to be lost if the internal SDA output level does not match the level on the I3C_SDA line (the high output as the internal SDA output; i.e. the I3C_SDA pin is in the high-impedance state) and the low level is detected on the I3C_SDA line during transmission of NACK in receive mode. Arbitration is lost due to a conflict of NACK transmission and ACK transmission when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such conflict occurs when multiple master devices send/receive the same information through a single slave device.

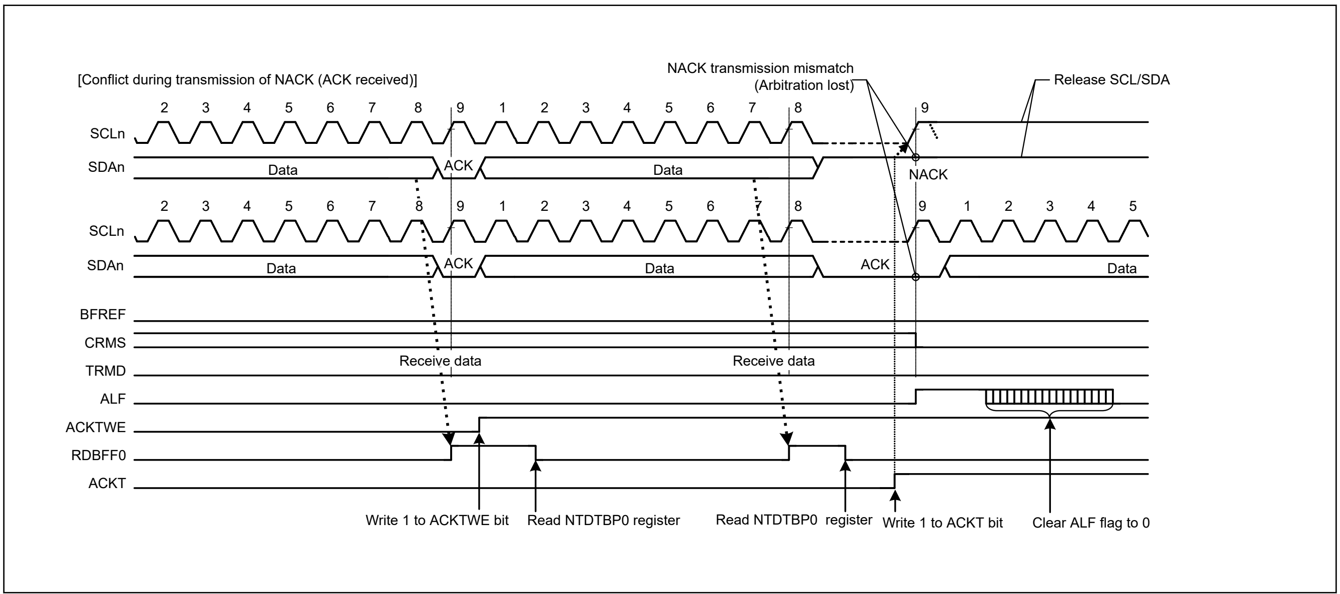


Figure 40.91 Example of arbitration-lost detection during transmission of NACK (NALE = 1)

The following section explains arbitration-lost detection using an example where two master devices (master A and master B) and a single slave device are connected through the bus. In this example, master A receives 2 bytes of data from the slave device, and master B receives 4 bytes of data from the slave device.

If master A and master B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in both master A and master B during access to the slave device. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. In this example,, master A sends NACK when it has received 2 final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received the necessary 4 bytes of data. At this time, the NACK transmission from master A and the ACK transmission from master B conflict. In general, if a conflict like this occurs, master A cannot detect ACK transmitted by master B and issues a stop condition. Therefore, the issuance of the stop condition conflicts with the SCL clock output of master B, which disturbs communication.

When this module receives ACK during transmission of NACK, it detects a defeat in conflict with other master devices and causes arbitration to be lost.

If arbitration is lost during transmission of NACK, this module is immediately released from the slave-matched state and enters slave receive mode. This prevents a stop condition from being issued, preventing a communication failure on the bus.

Similarly, in the ARP command processing of SMBus, the function to detect loss of arbitration during transmission of NACK is also available for eliminating the extra clock cycle processing (such as 0xFF transmission processing) necessary if the UDID (Unique Device Identifier) of assign address does not match in the Get UDID (general) processing after the Assign Address command.

The I3C detects arbitration-lost during transmission of NACK when the following condition is met while the BSTE.ALE bit = 1 and the BFCTL.NALE bit = 1 (arbitration-lost detection during NACK transmission enabled).

[Condition for arbitration-lost during NACK transmission]

- When the internal SDA output level does not match the I3C_SDA line (ACK is received) during transmission of NACK (ACKCTL.ACKT bit = 1)

(3) Slave Arbitration-Lost Detection (SALE Bit)

The I3C has a function to cause arbitration to be lost if the data for transmission (the internal SDA output level) and the level on the I3C_SDA line do not match (the high output as the internal SDA output, that is, the I3C_SDA pin is in the high impedance state) and the low level is detected on the I3C_SDA line in slave transmit mode. This arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identifier) over an SMBus.

If arbitration is lost during transmission of DATA, this module is immediately released from the slave-matched state and enters slave receive mode. This function can detect conflicts of data during transmission of UDIDs over an SMBus and eliminate subsequent redundant processing (processing for the transmission of 0xFF).

The I3C detects slave arbitration-lost when the following condition is met while the BSTE.ALE bit = 1 and the BFCTL.SALE bit = 1 (slave arbitration-lost detection enabled).

[Condition for slave arbitration-lost]

- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the I3C_SDA line in slave transmit mode (bits CRMS and TRMD in the PRSST register = 01b).

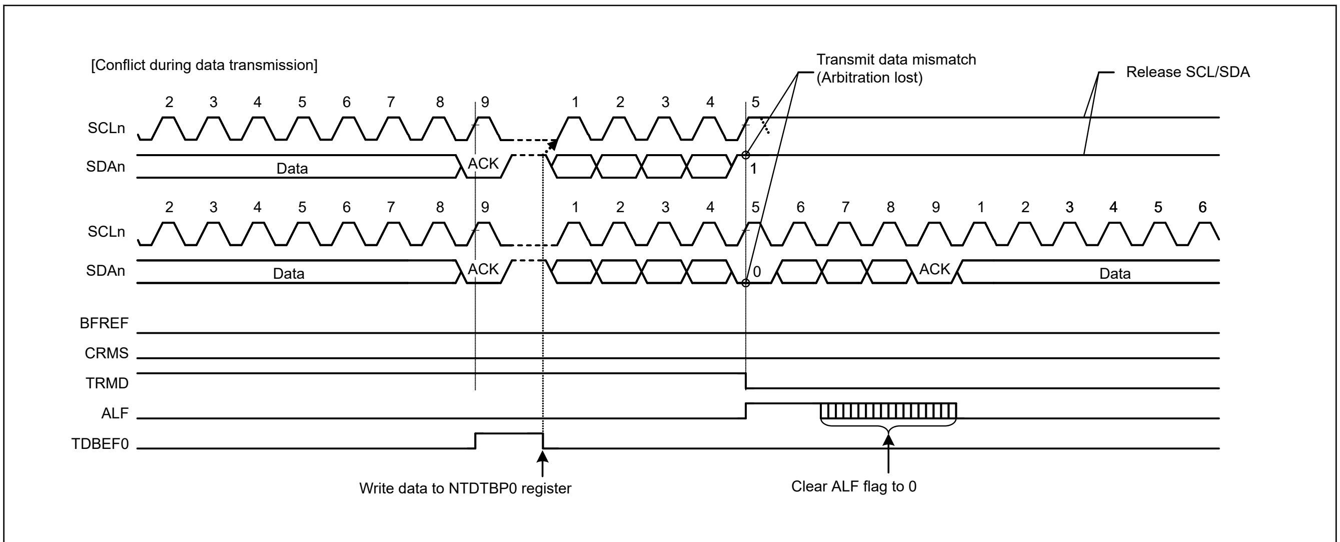


Figure 40.92 Example of slave arbitration-lost detection (SALE = 1)

40.3.2.3.6 Clock Stretching [I2C Mode]

(1) Function to Prevent Wrong Transmission of Transmit Data

When data have not been written to the Normal Transfer Data Buffer Port Register 0 (NTDTBP0) with I3C in transmission mode (PRSST.TRMD = 1), the I3C_SCL line is automatically held at the low level over the intervals shown below. This low-hold period is extended until data for transmission have been written, which prevents the unintended transmission of erroneous data.

Master transmit mode

- Low-level interval after a START condition or Repeated START condition is issued
- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next

Slave transmit mode

- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next

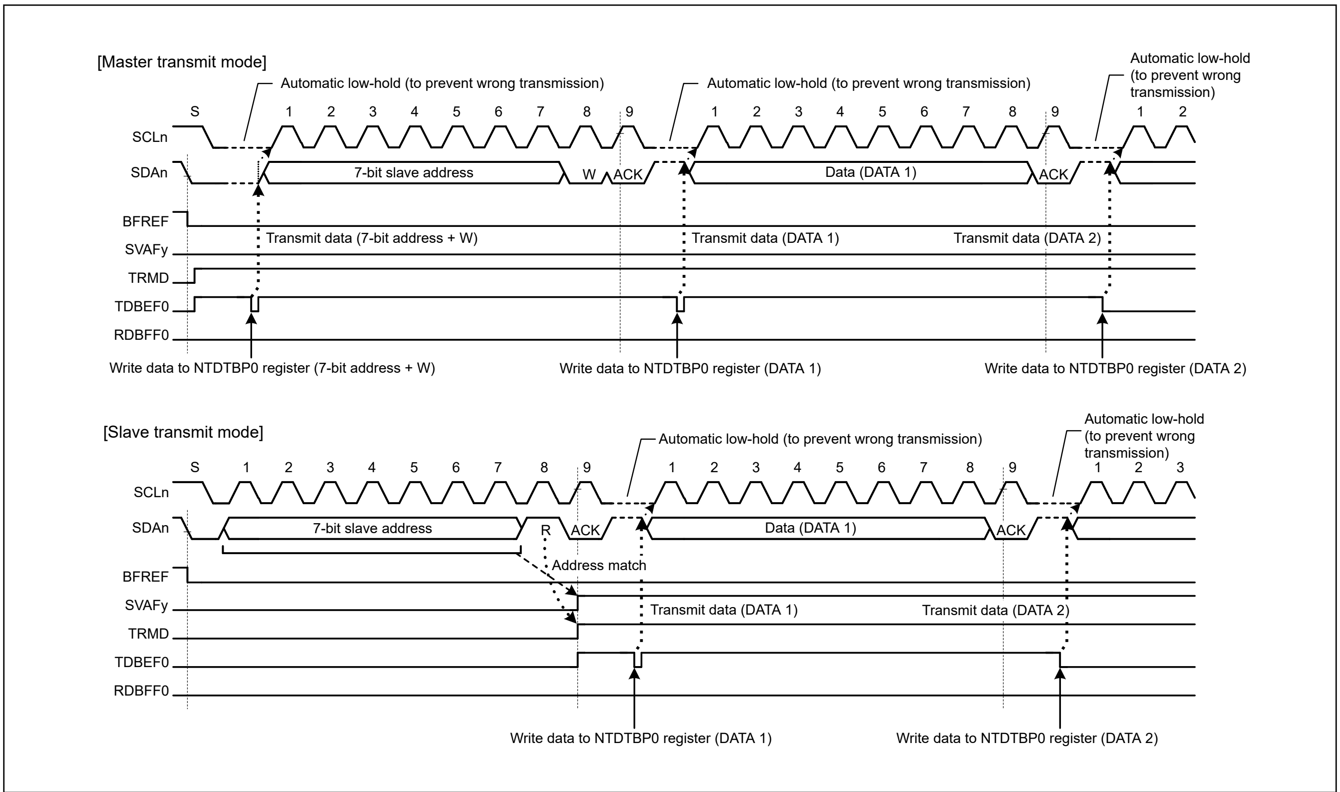


Figure 40.93 Automatic low-hold operation in transmit mode

(2) NACK Reception Transfer Abort Function

I3C has a function to abort transfer operation when NACK is received in transmit mode (PRSST.TRMD = 1). This function is enabled when the BSTE.NACKDE bit is set to 1 (transfer abort enabled). If the next transmit data has already been written (NTST.TDBEF0 = 0) when NACK is received, next data transmission at the falling edge of the ninth SCL clock cycle is automatically aborted. This prevents the I3C_SDA line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is aborted by this function (BST.NACKDF = 1), transmit operation and receive operation are discontinued. To restore transmit/receive operation, be sure to set the NACKDF flag to 0. In master transmit mode, restore operation using either of the methods below:

- After issuing a Repeated START condition, set the NACKDF flag to 0
- After issuing a STOP condition, set the NACKDF flag to 0 and then issue a START condition

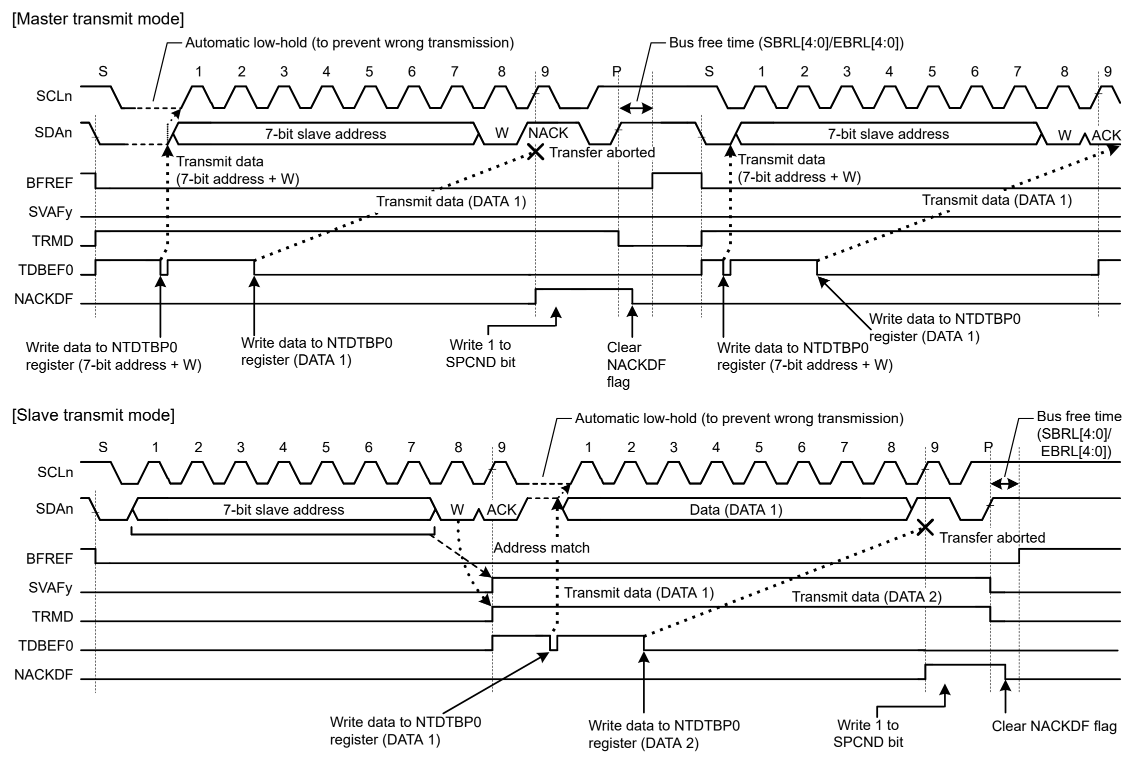


Figure 40.94 Abort of data transfer when NACK is received (NACKE = 1)

(3) Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (NTDTBP0) read is delayed for a period of one transfer frame or more with receive data full (NTST.RDBFF0 = 1) in receive mode (PRSST.TRMD = 0), I3C holds the I3C_SCL line low automatically immediately before the next data is received to prevent failure to receive data.

This function to prevent failure to receive data using the automatic low-hold function is also enabled even if the read processing of the final receive data is delayed and, in the meantime, I3C's own slave address or another slave address is received after a STOP condition is issued.

Sections in which the I3C_SCL line is held low can be selected with a combination of the RWE and ACKTWE bits in SCSTRCTL.

(a) 1-Byte Receive Operation and Automatic Low-Hold Function Using the RWE Bit

When the SCSTRCTL.RWE bit is set to 1, I3C performs 1-byte receive operation using the RWE bit function.

Furthermore, when the SCSTRCTL.ACKTWE bit = 0, I3C automatically sends the ACKCTL.ACKT bit value for the acknowledge bit in the period from the falling edge of the eighth SCL clock cycle to the falling edge of the ninth SCL clock cycle, and automatically holds the I3C_SCL line low at the falling edge of the ninth SCL clock cycle using the RWE bit function. This low-hold is released by reading data from NTDTBP0, which enables bytewise receive operation.

The RWE bit function is enabled for receive frames after a match with I3C's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

(b) 1-Byte Receive Operation (ACK/NACK Transmission Control) and Automatic Low-Hold Function Using the ACKTWE Bit

When the SCSTRCTL.ACKTWE bit is set to 1, I3C performs 1-byte receive operation using the ACKTWE bit function.

When the ACKTWE bit is set to 1, the NTST.RDBFF0 flag (Normal Rx Data buffer full) is set to 1 at the rising edge of the eighth SCL clock cycle, and the I3C_SCL line is automatically held low at the falling edge of the eighth SCL clock cycle. This low-hold is released by writing a value to the ACKCTL.ACKT bit, but cannot be released by reading data from NTDTBP0, which enables receive operation by the ACK/NACK transmission control according to the data received in byte units.

The ACKTWE bit function is enabled for receive frames after a match with I3C's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

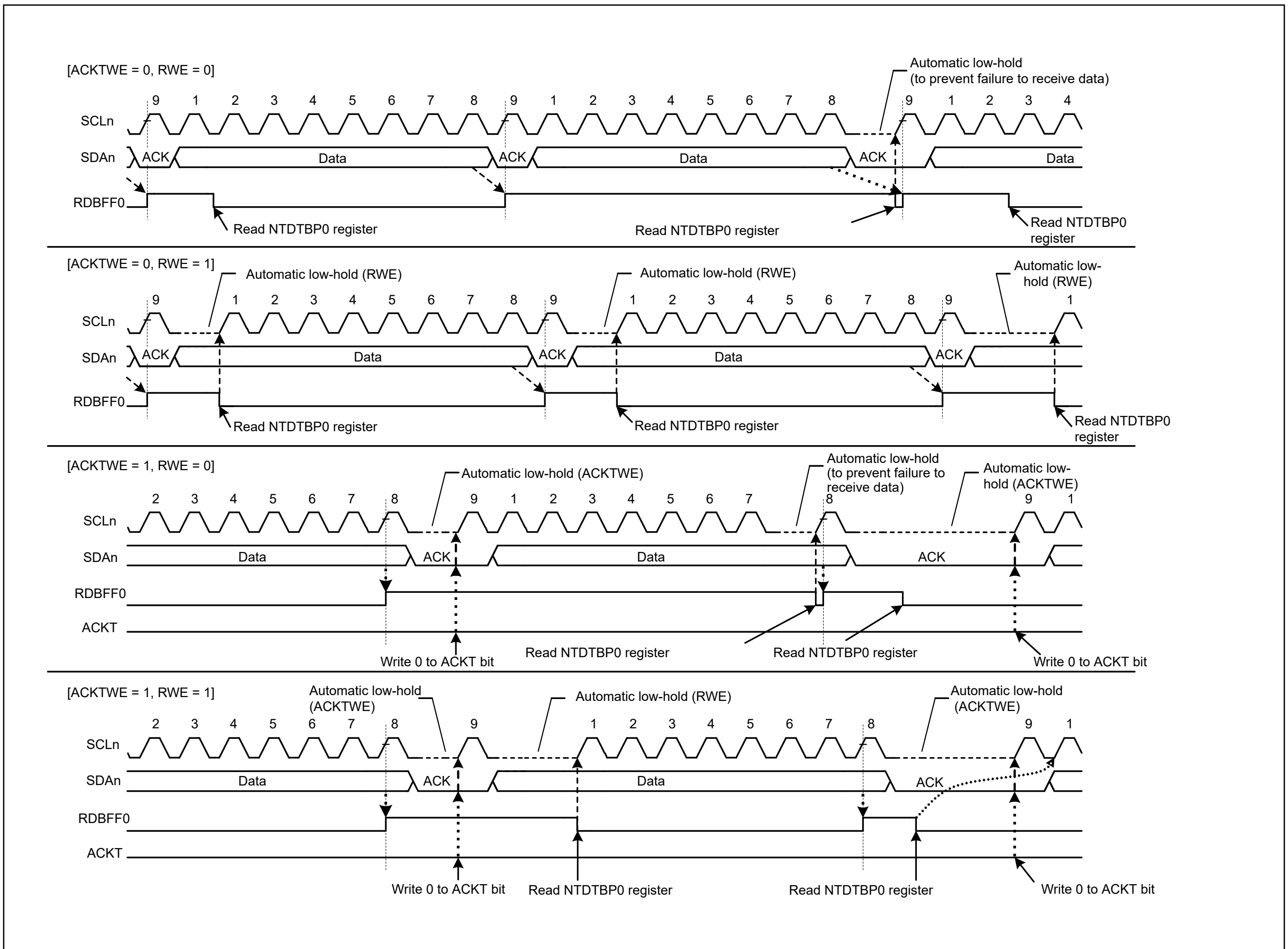


Figure 40.95 Automatic low-hold operation in receive mode (using ACKTWE and RWE bits)

40.3.2.3.7 Clock Stalling [I3C Mode]

I3C has the function of stalling the SCL during the SCL Low period.

The SCL stall control is described in the table below.

Table 40.12 I3C clock stalling (1 of 2)

Clock stalling condition	Clock stalling control	Clock stalling period
I3C Transfer, ACK/NACK Phase	SCSTLCTL.ACKPE bit setting	During the count period of SCSTLCTL.STLCYC [15:0] value
	Tx Data Buffer Empty	Until data is written to the Tx Data Buffer
	Rx Data Buffer Full	Until data is read from the Rx Data Buffer
I3C Write Data Transfer, Parity Bit	SCSTLCTL.PARPE bit setting	During the count period of SCSTLCTL.STLCYC [15:0] value
	Tx Data Buffer Empty	Until data is written to the Tx Data Buffer
I3C Read Transfer, Transition Bit	Rx Data Buffer Full	Until data is read from the Rx Data Buffer

Table 40.12 I3C clock stalling (2 of 2)

Clock stalling condition	Clock stalling control	Clock stalling period
Assigned Address Phase	SCSTLCTL.AAPE bit setting	During the count period of SCSTLCTL.STLCYC [15:0] value

The following figure shows the stalling timing of each Condition.

(1) I3C Transfer, ACK/NACK Phase

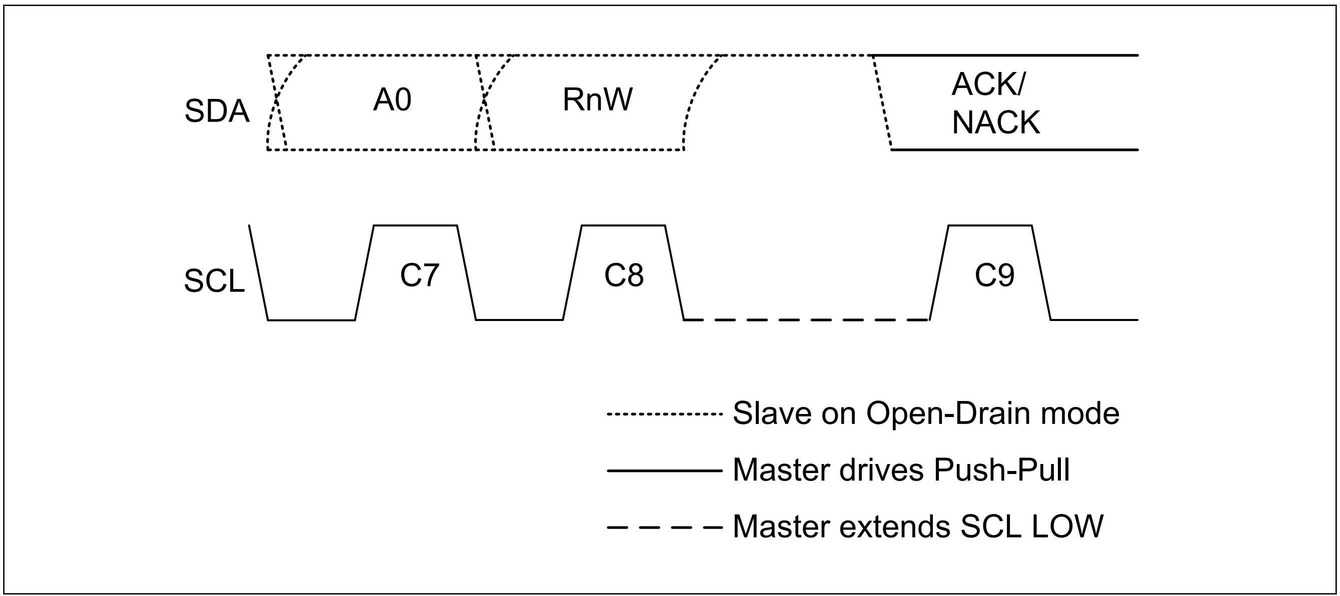


Figure 40.96 Master clock stalling in ACK phase

(2) I3C Write Data Transfer, Parity Bit

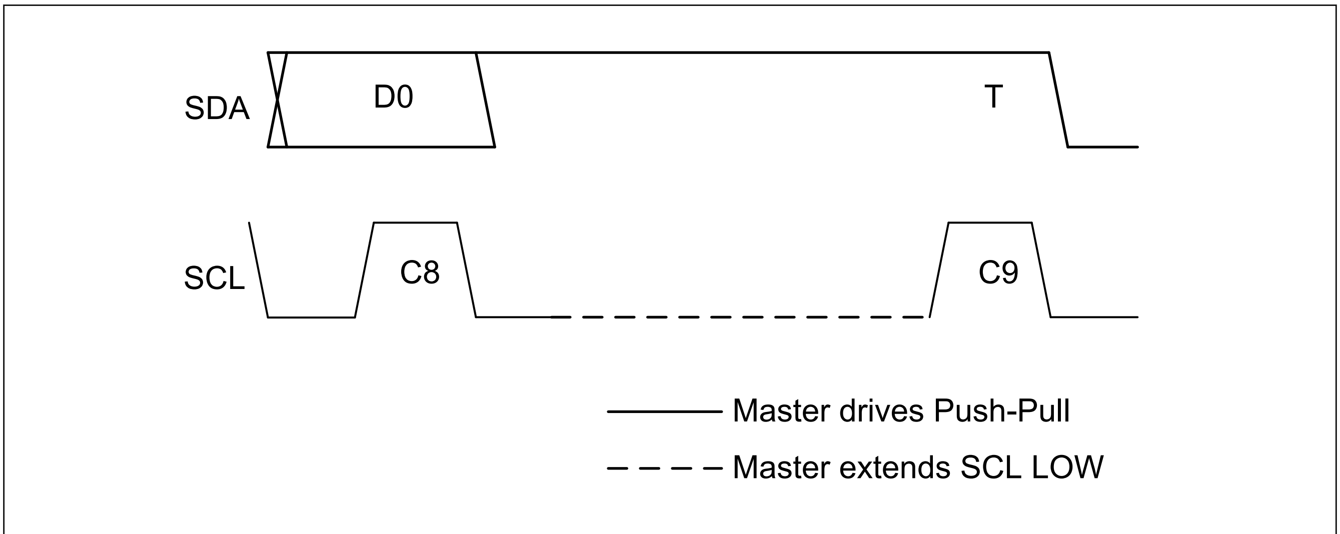


Figure 40.97 Master clock stalling in write parity bit

(3) I3C Read Transfer, Transition Bit

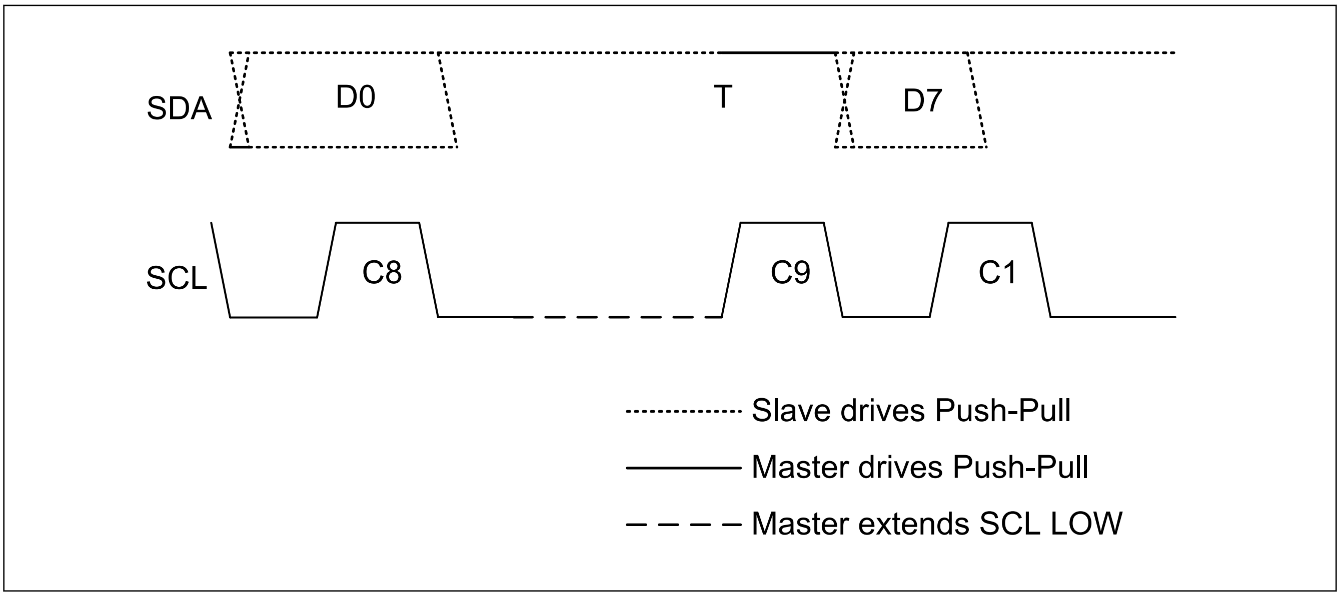


Figure 40.98 Master clock stalling in T-bit before next read data

(4) Dynamic Address Assignment, First Bit of Assigned Address

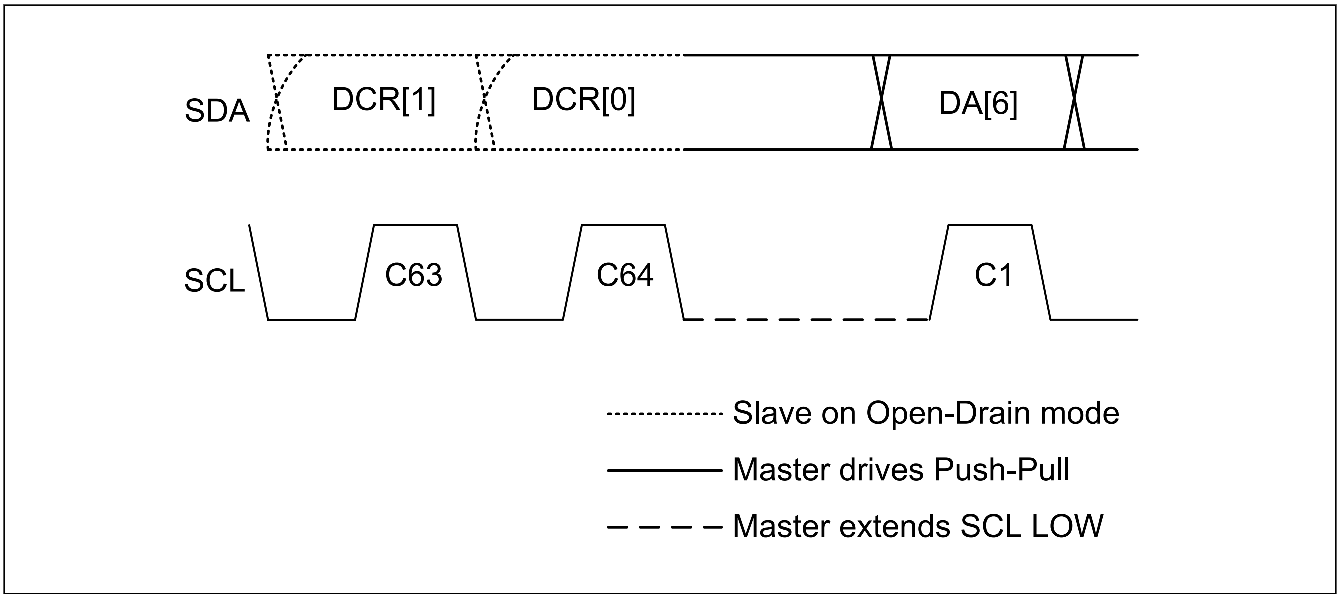


Figure 40.99 Master clock stalling in dynamic address first bit

40.3.2.3.8 In-Band Interrupt [I3C Mode]

I3C detects In-Band Interrupt in the arbitrated Address Header following a START condition (but not following a Repeated START). If START Request (SDA Low Drive) is issued from Slave Device, I3C drives SCL low and completes START condition. After that, it supplies SCL and receives In-Band Interrupt Request.

The In-Band Interrupt to be detected is classified into the following three types.

- Slave Interrupt Request
- Mastership Request

The operation when detecting each In-Band Interrupt is described below.

(1) Slave Interrupt Request

1. Detect Slave Address with RnW bit High in Address Header.
2. Compare the detected Slave Address with the DVDYAD[7:0] in each DAT (DATBASm register).
3. When it does not match DAT.DVDYAD[7:0]:
   Responds NACK, then issues the STOP condition.
   When it matches the DAT.DVDYAD[7:0] bits and the DAT.DVSIRRJ bit = 1:
   It operates in the following order:
   (a) Responds NACK.
   (b) Issues Repeated START condition, then automatically issues Direct DISEC CCC to the detected Slave.
   (c) Issues the STOP condition.

   When it matches the DAT.DVDYAD[7:0] bits and the DAT.DVSIRRJ bit = 0:
   Responds ACK.
4. When DAT.DVIBIPL = 0:
   Issues the STOP condition.
   When DAT.DVIBIPL = 1:
   Drives the SCL to receive the IBI Data from the Slave following the ACK response and receives IBI Data.
   It stores the received IBI Data into the IBI Data Queue.
   Each time IBI Data of the size set by the NQTHCTL.IBIDSSZ[7:0] bits is received, the IBI Status Descriptor is stored in the IBI Queue.
5. After detection of Low of T-bit following IBI Data, issues STOP condition.
6. After issues of STOP condition
   NACK response:
   - If IBINCTL.NRSIRCTL = 0, the IBI Status Descriptor is not stored into the IBI Queue.
   - If IBINCTL.NRSIRCTL = 1, the IBI Status Descriptor is stored into the IBI Queue.

   ACK response:
   Stores the IBI Status Descriptor into the IBI Queue.

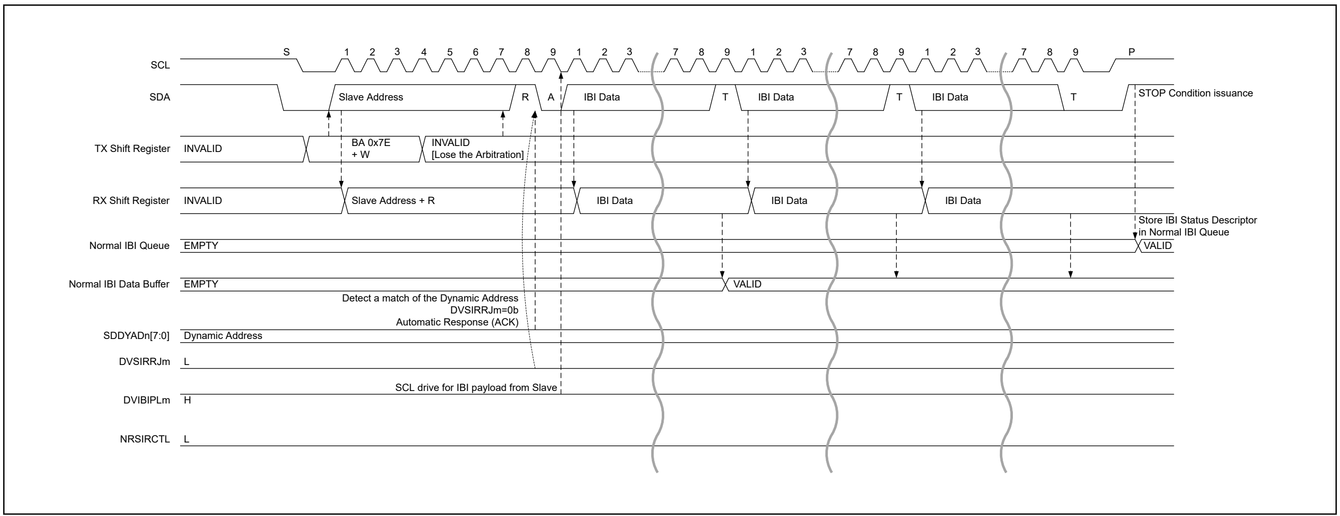


Figure 40.100 Slave interrupt request : ACK and DVIBIPL = 1

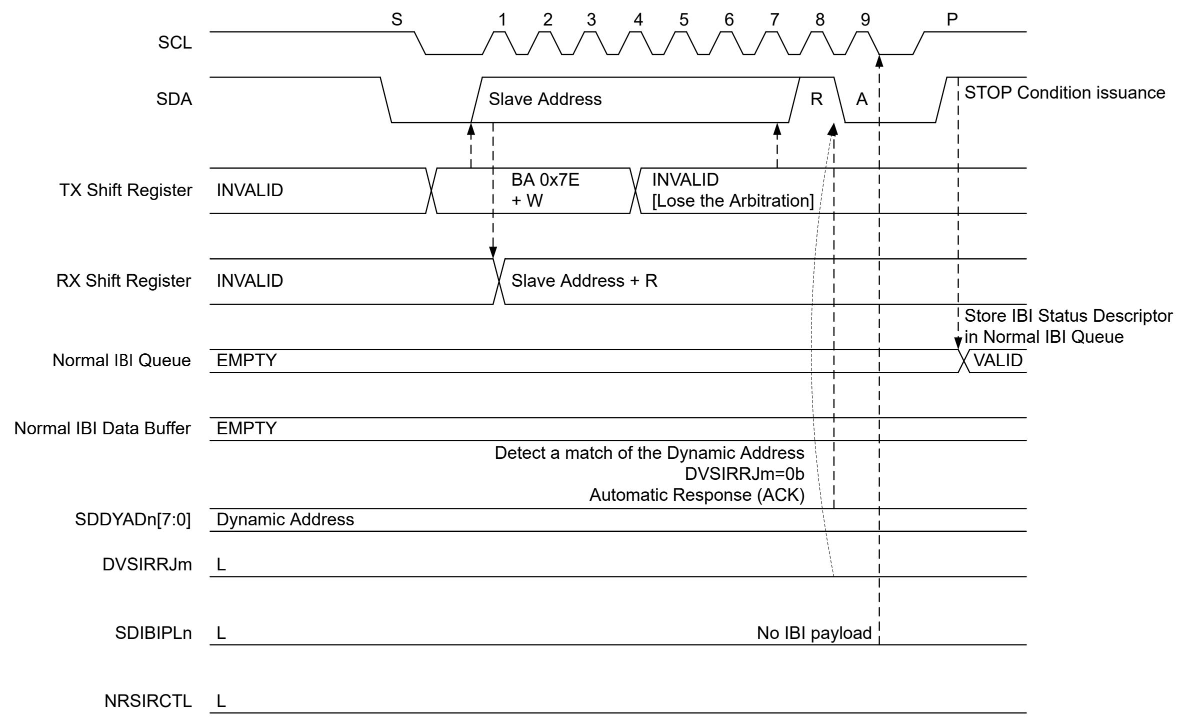


Figure 40.101 Slave interrupt request : ACK and DVIBIPL = 0

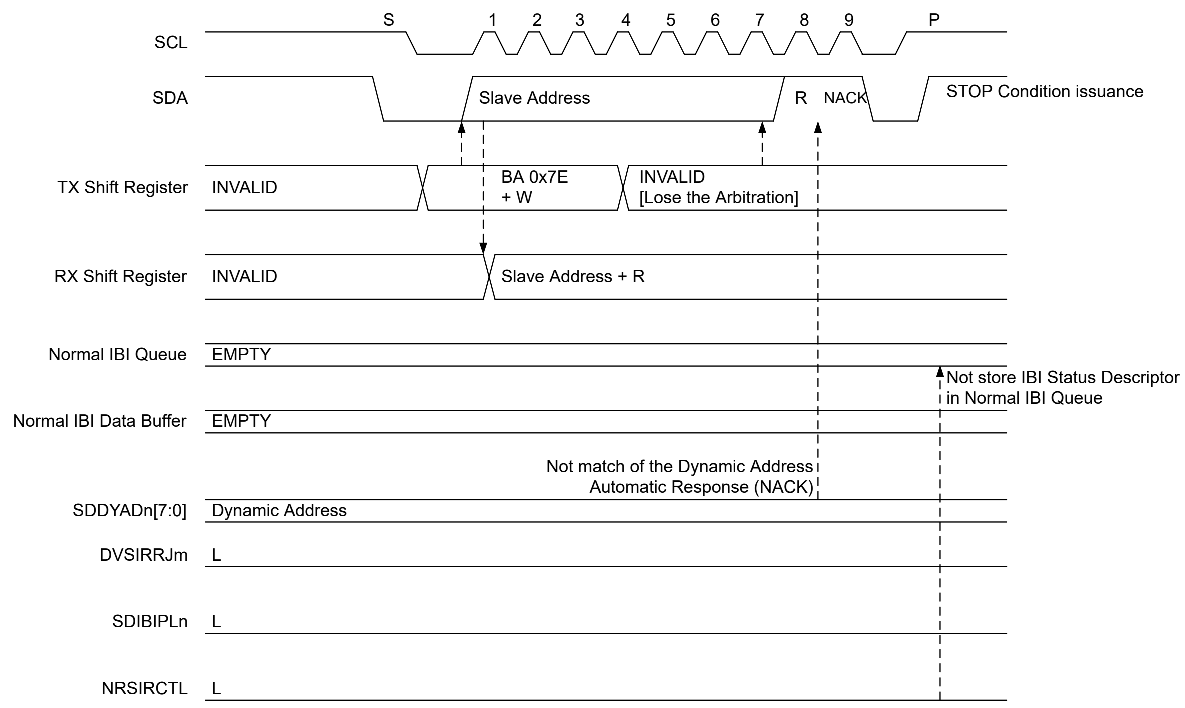


Figure 40.102 Slave interrupt request : NACK (not match the SDDYAD[7:0] of DAT) and NRSIRCTL = 0

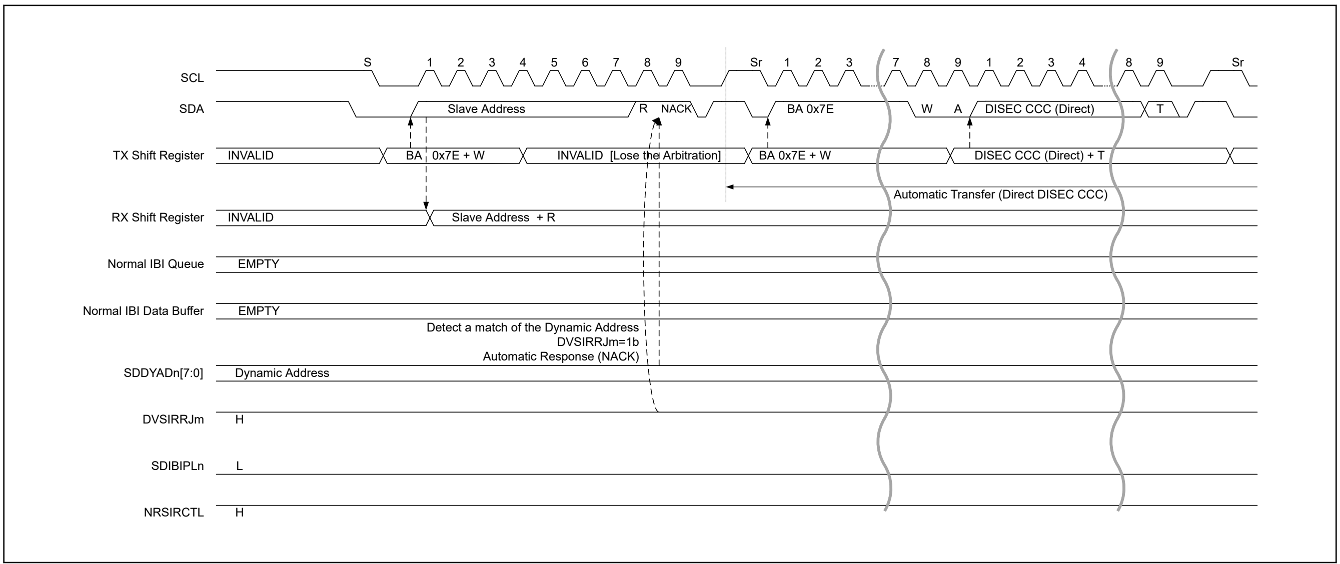


Figure 40.103 Slave interrupt request : NACK (DVSIRRJ = 1) and NRSIRCTL = 1 (1/2)

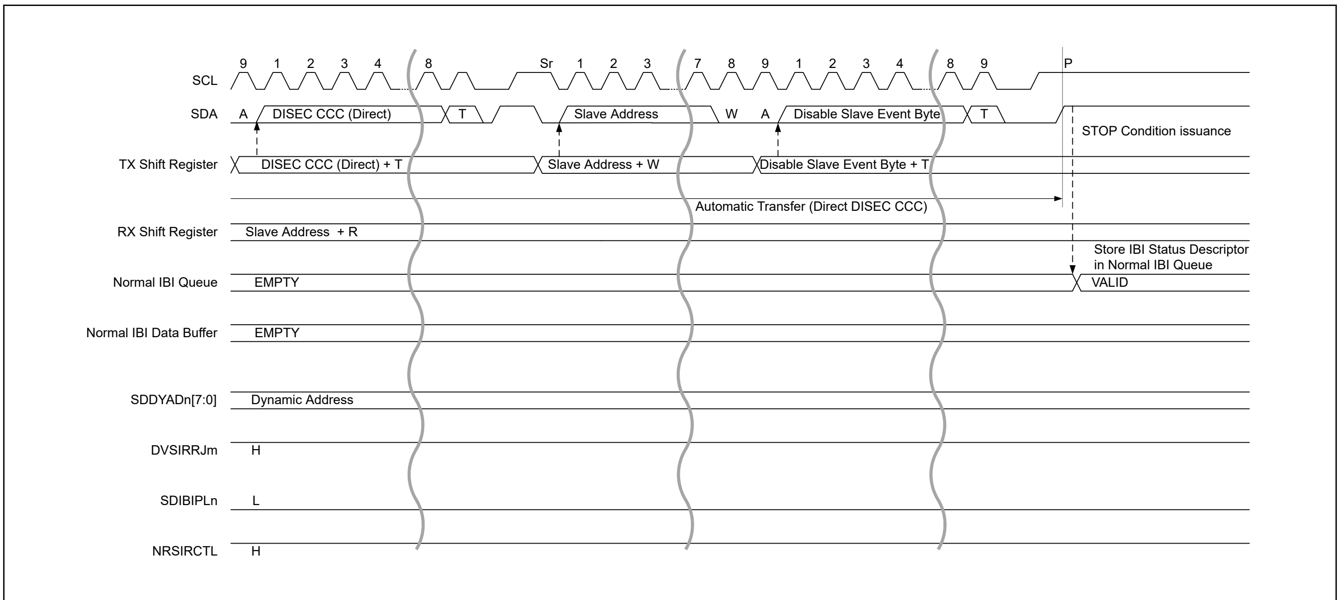


Figure 40.104 Slave interrupt request : NACK (DVSIRRJ = 1) and NRSIRCTL = 1 (2/2)

(2) Mastership Request

1. Detect Slave Address with RnW bit Low in Address Header.
2. Compare the detected Slave Address with the DVDYAD[7:0] in each DAT (DATBAS register).
3. When it does not match DAT.DVDYAD[7:0]:
   Responds NACK, then issues the STOP condition.
   When it matches the DAT.DVDYAD[7:0] bits and Device Role[1:0] in RBCR (MSDCTm) is other than I3C Master (01b):
   Responds NACK, then issues the STOP condition.
   When it matches the DAT.DVDYAD[7:0] bits and Device Role[1:0] in RBCR (MSDCTm) is I3C Master (01b):
   - When DAT.DVMRRJ = 1
     It operates in the following order.
     (a) Responds NACK.
     (b) Issued Repeated START condition and automatically issues Direct DISEC CCC to the detected Slave.
     (c) Issues the STOP condition.

- When DAT.DVMRRJ = 0
  Responds ACK, then issues STOP condition.

4. After issues of STOP condition,
   NACK response:
   - If IBINCTL.NRMRCTL = 0, the IBI Status Descriptor is not stored into the IBI Queue.
   - If IBINCTL.NRMRCTL = 1, the IBI Status Descriptor is stored into the IBI Queue.

   ACK response:
   Stores the IBI Status Descriptor into the IBI Queue.

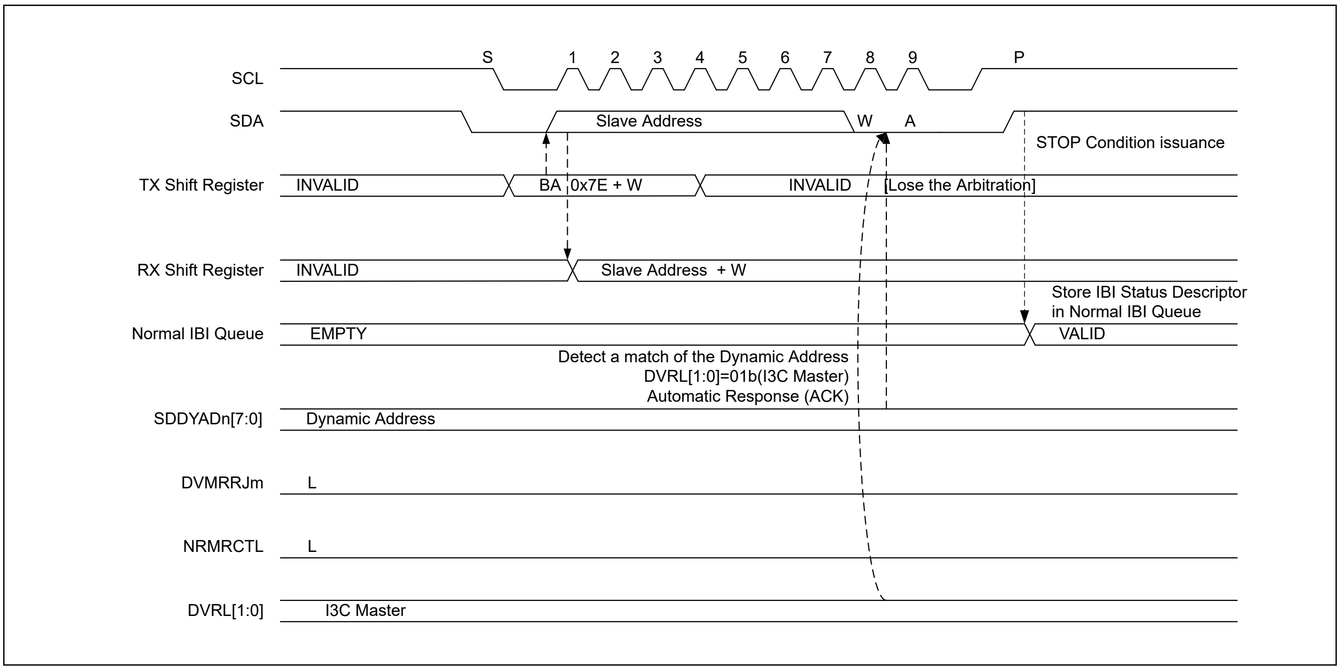


Figure 40.105 Mastership request : ACK

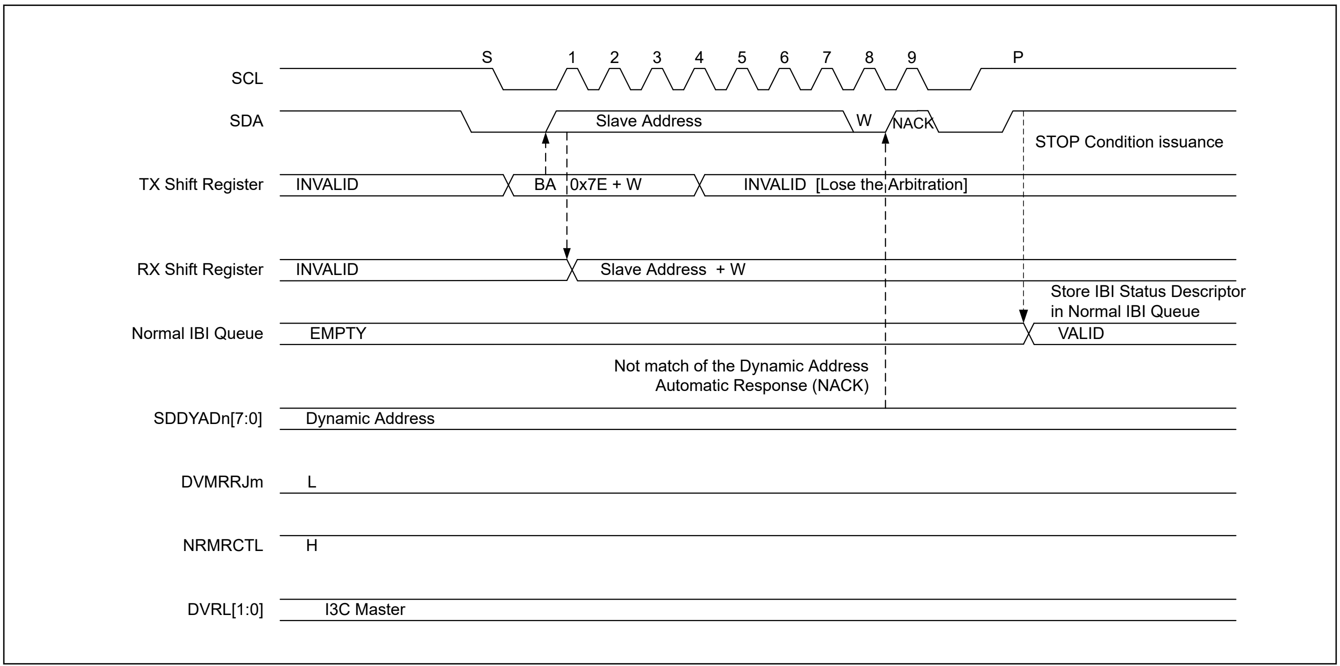


Figure 40.106 Mastership request : NACK (not match the DVDYAD[7:0] of DAT) and NRMRCTL = 1

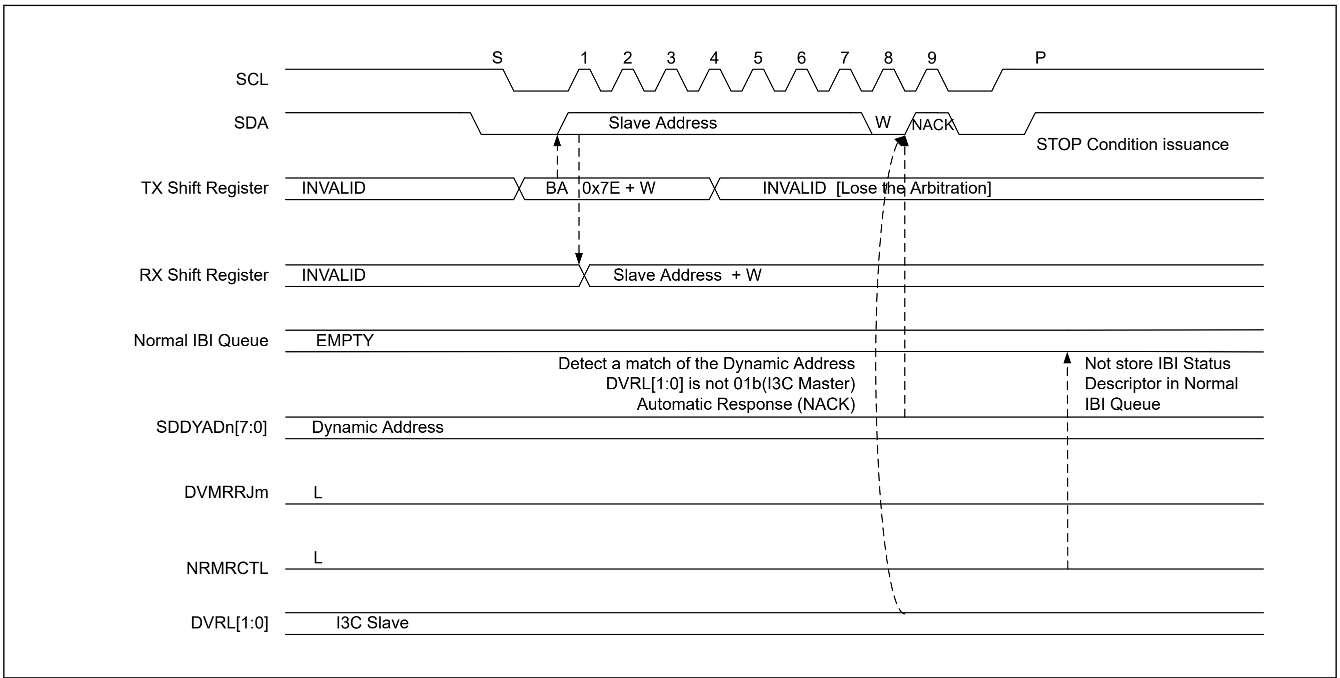


Figure 40.107 Mastership request : NACK (Device Role[1:0] is not 01 (I3C master)) and NRMRCTL = 0

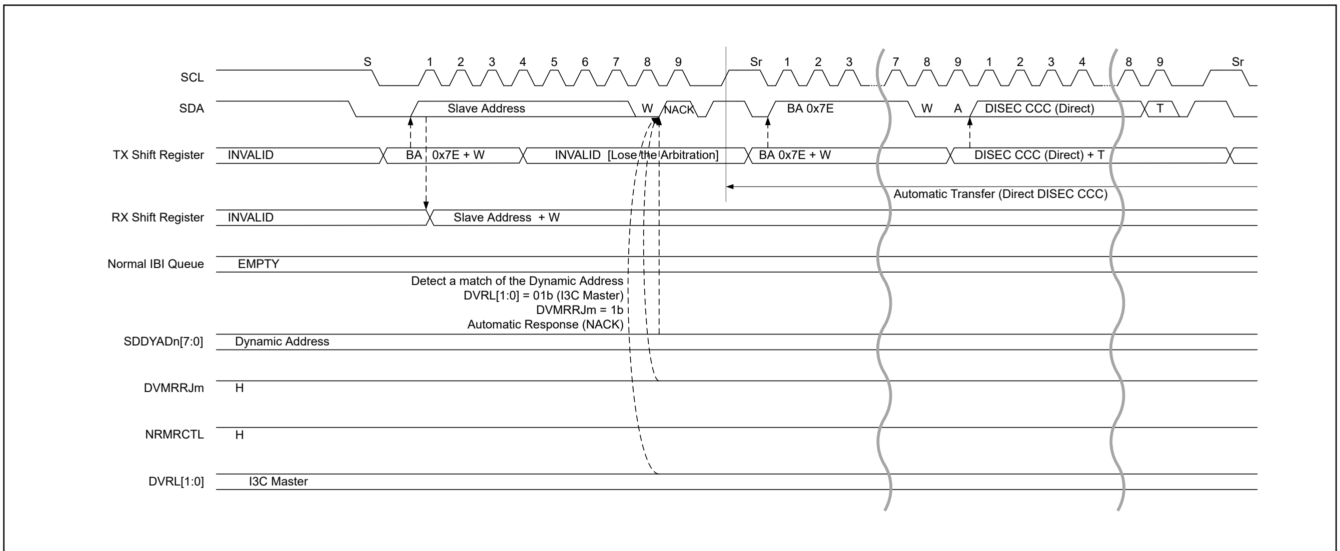


Figure 40.108 Mastership request : NACK (DVMRRJ = 1) and NRMRCTL = 1 (1/2)

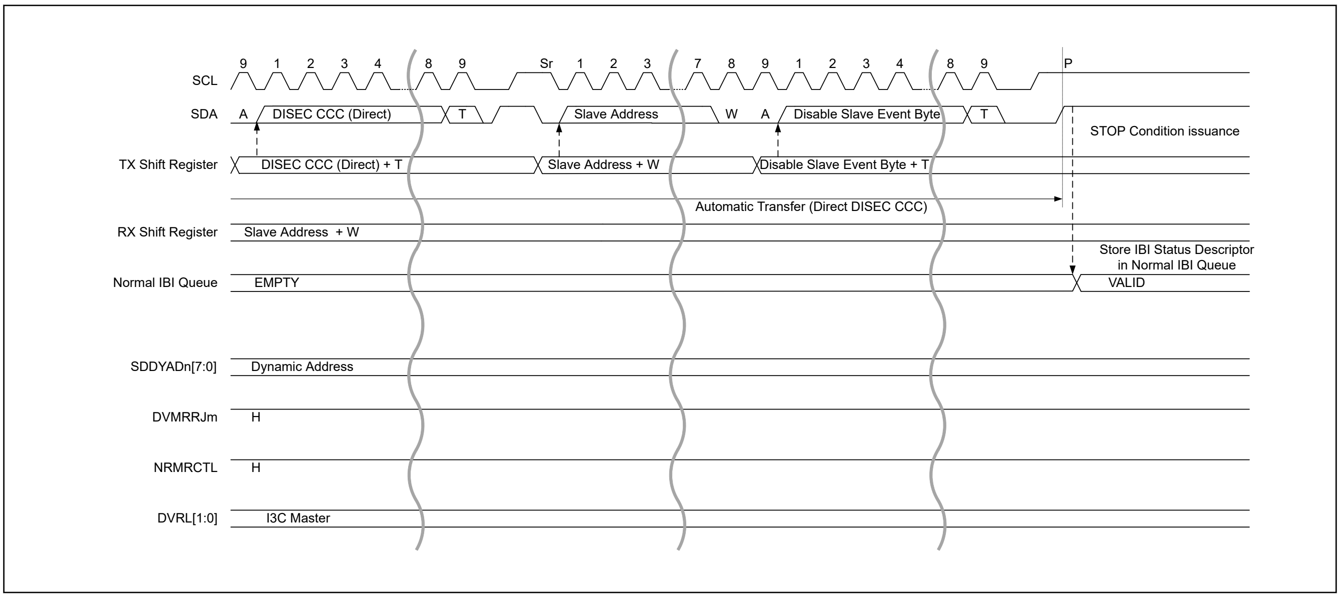


Figure 40.109 Mastership request : NACK (DVMRRJ = 1) and NRMRCTL = 1 (2/2)

40.3.2.3.9 Timing Control

Timing Control is a function that enables Master to efficiently read data from Slave by controlling and grasping the timing at which the Slave Device samples the sensor value.

For details, refer to 5.1.8 Timing Control of MIPI I3C Specification v1.0.

In I3C, timing Control supports the following three modes.

- Sync mode
- Async mode 0 (Asynchronous Basic mode)
- Async mode 1 (Asynchronous Advanced mode)

The resources for realizing Timing Control in each mode are described in the following sections.

(1) Sync Mode

1. I3C Master
   - When STCTL.STOE is set to 1, when the master sends an ST message (SETXTIME CCC with ST subcommand), there is a function to issue the synchronous timing event under the START condition of the ST message.
     While measuring the T_ph period with an external timer, the start of T_ph and the Delay Time [DT] of the ST message can be measured by capturing the count value with the synchronous timing event.
     The measured value of Delay Time is sent as a DT message (SETXTIME CCC with DT subcommand) following the ST message.
2. I3C Slave
   - When STCTL.STOE is set to 1, there is a function to issue the synchronous timing event for each START condition.
     STCTL.STOE is cleared when an ST message is received (SETXTIME CCC using the ST subcommand).
     Check the reception of the ST message with the Receive Status Descriptor, and use an external timer to correct the T_ph period based on the count value captured in the synchronous timing event and the Delay Time obtained from the DT message.
     While measuring the T_ph period with an external timer, the start of T_ph and the Delay Time [DT] of the ST message can be measured by capturing the count value with the synchronous timing event.
     The sampling timing is recalculated by the corrected T_ph.

(2) Async Mode 0 (Asynchronous Basic Mode)

For timing control in Async Mode 0, set the ATCTL register if necessary.

1. I3C Master

I3C has counters of MREFMREF(32bit) and MC2(16bit) for Async mode 0.

- MREF Counter
  When ATCCNTE.ATCE is enabled, it starts counting.
  It captures as MREF on the SCL rise edge next to ACK for the IBI transmitted from the I3C Slave.
- MC2 Counter
  After enabling ATCCNTE.ATCE, it counts up from the SCL rise edge next to ACK for the IBI transmitted from the I3C Slave to the SCL rise edge next to the Tbit after Mandatory Byte, and capture it as MC2.

The MREF and MC2 capture values are stored next to the IBI Status Descriptor when IBI is received from the I3C Slave with the DATBASm.DVIBITS bit set to 1.

The MREF counter implemented in I3C is 32-bit counter.

However, if the 32-bit counter is insufficient due to system requirements, I3C has MREF counter overflow and MREF capture event for expansion. These events are enabled by setting ATCTL.MREFOE to 1.

MREF counter overflow is output when the internal MREF counter overflows.

The MREF counter can be extended by using it as a count event for an external timer. MREF capture event is output at the same timing as the capture timing of the internal MREF counter. By using it as the capture timing of the external timer, it can be used as an MREF counter concatenated with the value stored in the IBI Data buffer.

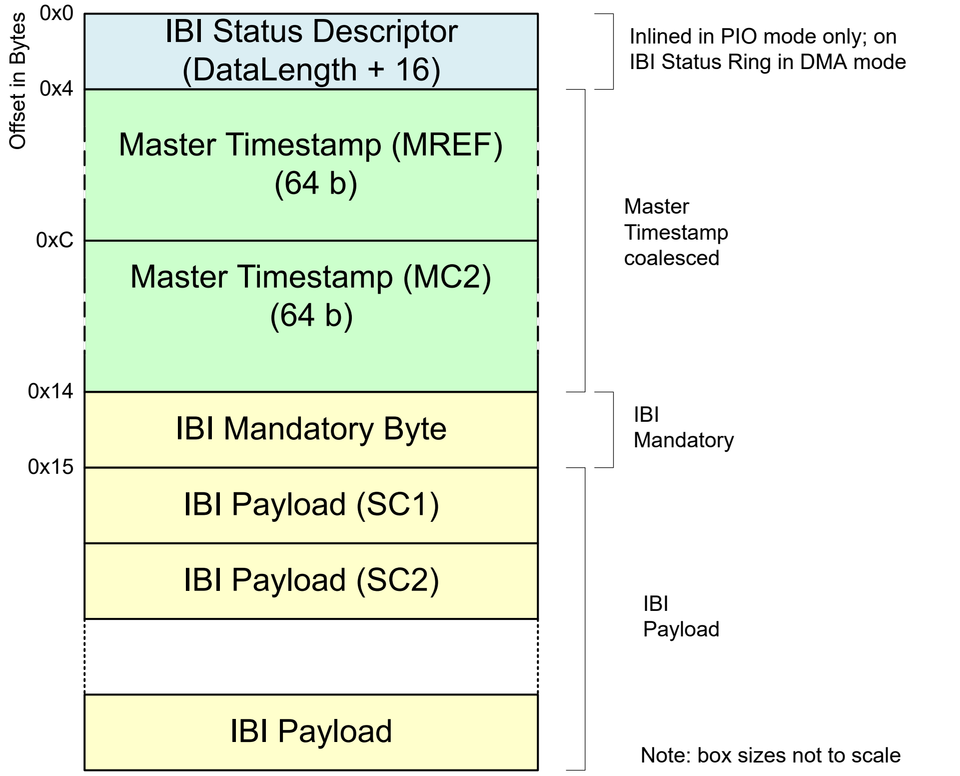


Figure 40.110 Master timestamp counters for IBI event

Note: Please evaluate the Sensor Event time of I3C Slave according to the calculation formula of the MIPI I3C specification v1.0 document.

2. I3C Slave

I3C has counters of SC1(16bit) and SC2(8bit) for Async mode 0.

- SC1 Counter

After enabling ATCCNTE.ATCE, it counts up from SC1 count trigger*1 to SCL rise edge next to ACK for the IBI, and capture it as SC1.

Note 1. SW or external trigger can be selected by selection bits.

- SC2 Counter
  After enabling ATCCNTE.ATCE, it counts up from SCL rise edge next to ACK for the IBI transmitted from I3C Slave to SCL rise edge next to Tbit after Mandatory Byte, and capture it as SC2.

When the CETSS.ASYNE [0] bit = 1 and the ITS bit in Command Descriptor for issuing IBI is 1, the SC1 and SC2 capture values are transmitted following the IBI Mandatory Byte as shown in the following figure.

If the SC1 and SC2 counters overflow, 0xFFFF and 0xFF are captured and CETSS.ICOVF is set 1.

The DATA_LENGTH[15:0] bits value of the Command Descriptor sets a value obtained by adding the number of data of SC1 and SC2 to the number of transmission data.

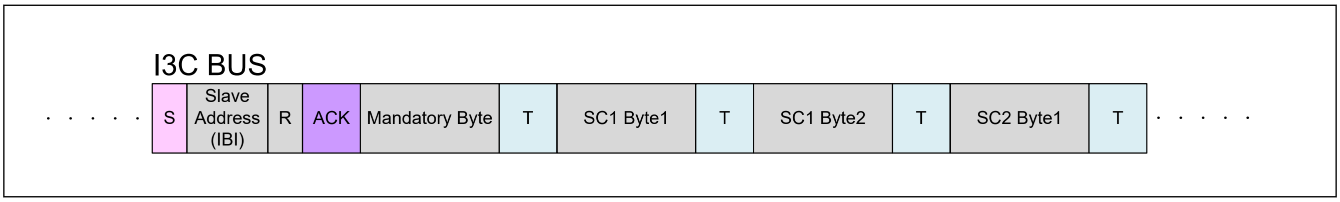


Figure 40.111 Example of asynchronous mode 0 timestamp data transfer

(3) Async Mode 1 (Asynchronous Advanced Mode)

For timing control in Async Mode 1, set the ATCTL register if necessary.

1. I3C Master
   I3C has counters of MREF(32bit), MSyncCNT(32bit) and MC2(16bit) for Async mode 1.

   - MREF Counter
     When ATCCNTE.ATCE is enabled, it starts counting.
     It captures as MREF at the SCL rise edge next to ACK for the IBI transmitted from the I3C Slave.
   - MSyncCNT Counter
     When ATCCNTE.ATCE is enabled, it starts counting.
     It captures as MSyncCNT for each aME (SDA falling edge of START condition), and store it in the capture register.
   - MC2 Counter
     After enabling ATCCNTE.ATCE, it counts up from SCL rise edge next to ACK for the IBI transmitted from I3C Slave to SCL rise edge next to Tbit after Mandatory Byte, and capture it as MC2.

   The MREF and MC2 capture values are stored next to the IBI Status Descriptor when the IBI is received from the I3C Slave with the DATBASm.DVIBITS bit set to 1 (same as Async mode 0).
   When ATCTL.AMEOE is enabled, an aME Event is issued for each aME. Use that event as a trigger to read the MSyncCNT capture value from the MRCCPT register and hold it in an external memory.

2. I3C Slave
   I3C has counters of SC1(16bit), SC2(8bit) and aME_TICK(8bit) for Async mode 1.

   - SC1 Counter
     After enabling ATCCNTE.ATCE, it counts up from SC1 count trigger *1 to the first aME, and capture it as SC1.

     Note 1. SW or external trigger can be selected by selection bits.
   - SC2 Counter
     After enabling ATCCNTE.ATCE, it counts up from SCL rise edge next to ACK for the IBI transmitted from I3C Slave to SCL rise edge next to Tbit after Mandatory Byte, and capture it as SC2.
   - aME_TICK Counter
     After enabling ATCCNTE.ATCE, it counts every aME, and capture it as aME_TICK at the SCL rise edge next to ACK for the IBI.
     The aME_TICK counter is cleared on the first aME after the SC1 count trigger.

When the CETSS.ASYNE[1] bit = 1 and the ITS bit in Command Descriptor for issuing IBI is 1, the SC1, SC2 and aME_TICK capture values are transmitted following the IBI Mandatory Byte as shown in the following figure.

If the SC1 and SC2 counters overflow, 0xFFFF and 0xFF are captured and CETSS.ICOVF is set 1.

The DATA_LENGTH[15:0] bits value of the Command Descriptor sets a value obtained by adding the number of data of SC1, SC2 and aME_TICK to the number of transmission data.

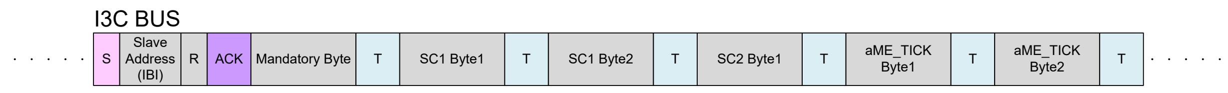


Figure 40.112 Example of asynchronous mode 1 timestamp data transfer

40.3.2.3.10 HDR-DDR CRC5 Algorithm

The CRC5 value is computed on a complete message, including the Command Word and all Data Words.

For a Command Word, the CRC5 is computed based on the value of the 16-bit payload, including:

- The Read vs. Write bit
- The Command Value
- The Slave Address, and
- The lowest-order bit (Write Reserved and Read Parity adjustment).

For Data Words, the CRC5 is computed based on all Data 16-bit values transmitted for that Command.

The CRC5 value is initialized to 0x1F. The CRC5 polynomial is below.

$CRC5 = X^5 + X^2 + X^0$

40.3.2.3.11 Port Control

(1) Extra SCL Clock Cycle Output Function

In master mode, I3C module has a facility for the output of extra SCL clock cycles to release the I3C_SDA line of the slave device from being held at the low level due to the master being out of synchronization with the slave device.

This function is mainly used in master mode to release the I3C_SDA line of the slave device from the state of being fixed to the low level by including extra cycles of SCL output from I3C with single cycles of the SCL clock as the unit in the case of a bus error where I3C cannot issue a Repeated START condition or a STOP condition because the slave device is holding the I3C_SDA line at the low level. Do not use this facility in normal situations. Using it when communications are proceeding correctly will lead to malfunctions.

When the OUTCTL.EXCYC bit is set to 1, an additional clock pulse at the frequency set by the REFCKCTL.IREFCKS[2:0] bits and the STDBR.SBRHO[7:0] and STDBR.SBRLO[7:0] registers is output from the I3C_SCL pin. After output of this clock pulse, the EXCYC bit automatically becomes 0. After confirming that the EXCYC bit is 0, wait for the setup time of the Repeated START condition or STOP condition, and then confirm the detection of the Repeated START condition or STOP condition. If the Repeated START condition or STOP condition is not detected, consecutive additional clock pulses can be output by writing 1 to the EXCYC bit again.

When I3C module is in master mode and the slave device is holding the I3C_SDA line at the low level because synchronization with the slave device has been lost due to the effects of noise, etc., the output of a Repeated START condition or a STOP condition is not possible. The facility for output of an extra cycle of the SCL clock can be used to output extra cycles of SCL one by one to make the slave device release the I3C_SDA line from being held at the low level, thus recovering the bus from an unusable state. Release of the I3C_SDA line by the slave device can be monitored by reading the SDILV bit in PRSTDBG. After the I3C_SDA line has been released by the slave device, the preset of a Repeated START condition or a STOP condition is issued.

Use this function with the BFCTL.MALE bit set to 0 (master arbitration-lost detection is disabled).

[Output conditions for using the EXCYC bit in OUTCTL]

- When the bus is free (BFREF flag in BCST = 1) or in master mode (CRMS bit = 1 in PRSST and BFREF flag = 0 in BCST)
- When the communication device does not hold the I3C_SCL line low

Figure 40.113 shows the operation timing of the extra SCL clock cycle output function (EXCYC bit).

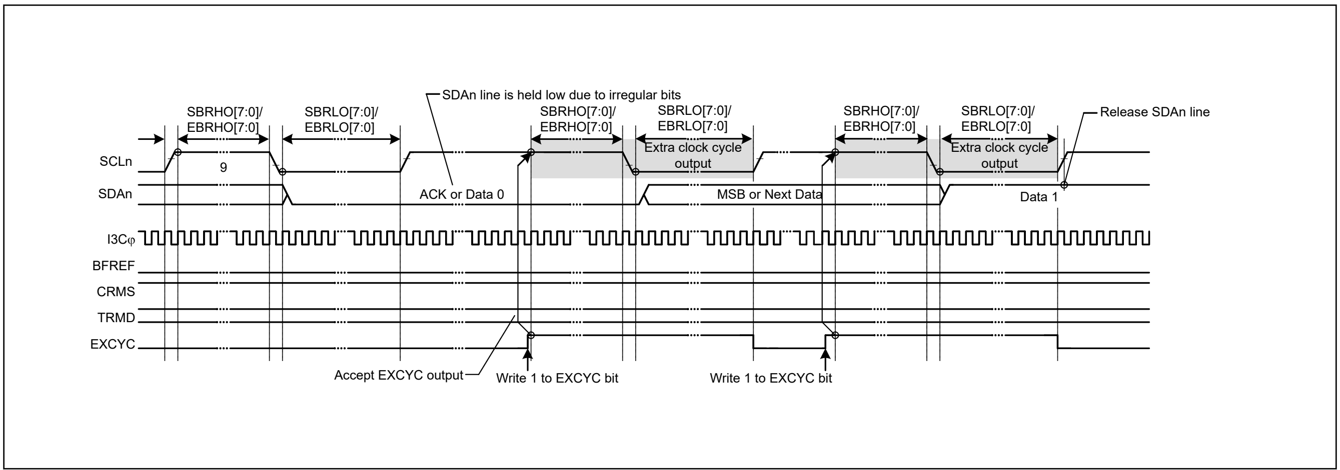


Figure 40.113 Extra SCL clock cycle output function (EXCYC bit)

40.3.2.3.12 SMBus Operation [I^2C Mode]

I3C is available for data communication conforming to the SMBus (Version 2.0). To perform SMBus communication, set the BFCTL.SMBS bit to 1. To use the transfer rate within a range of 10 kbps to 100 kbps of the SMBus specification, set the REFCKCTL.IREFCKS[2:0] bits, the STDBR.SBRHO[7:0] bits, and the STDBR.SBRLO[7:0] bits. In addition, determine the values of the OUTCTL.SDODCS bit and the OUTCTL.SDOD[2:0] bits to meet the data hold time specification of 300 ns or more. If I3C is used only as an I^2C slave device, the transfer rate setting is not necessary, whereas the STDBR.SBRLO[7:0] bits needs to be set to a value longer than the data setup time (250 ns).

For the SMBus device default address (1100 001), use one of the slave device address table basic registers 0 to 2 (SDATBASn.SDSTAD[6:0] bits) (y = 0 to 2), and set the corresponding SDATBASn.SDADLS bit (7-bit/10-bit address format select) (y = 0 to 2) to 0 (7-bit address format).

When transmitting the UDID (Unique Device Identifier), set the BFCTL.SALE bit to 1 to enable the slave arbitration lost detection function.

(1) SMBus Timeout Measurement

(a) Measuring timeout of slave device

The following period (timeout interval: $T_{LOW:SEXT}$) must be measured for slave devices in SMBus communication.

- From START condition to STOP condition

To measure timeout for slave devices, measure the period from START condition detection to STOP condition detection with the GPT timer using a START condition detection interrupt (I3C_EEI) and STOP condition detection interrupt (I3C_EEI) of I3C. The measured timeout period must be within the total clock low-level period [slave device] $T_{LOW:SEXT}$: 25 ms (max.) of the SMBus specification.

If the time measured with the GPT exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (min.) of the SMBus specification, the slave device must release the bus by writing 1 to the RSTCTL.INTLRST bit to issue an internal reset of I3C. When an internal reset is issued, I3C stops driving the bus for the I3C_SCL pin and I3C_SDA pin and make the I3C_SCL/I3C_SDA pin outputs high-impedance, which releases the bus.

(b) Measuring timeout of master device

The following periods (timeout interval: $T_{LOW:MEXT}$) must be measured for master devices in SMBus communication.

- From START condition to acknowledge bit
- Between acknowledge bits

- From acknowledge bit to STOP condition

To measure timeout for master devices, measure these periods with the GPT timer using a START condition detection interrupt (I3C_EEI), STOP condition detection interrupt (I3C_EEI), and transmit end interrupt (I3C_TEND) or Normal Rx Data buffer full interrupt (I3C_RX) of I3C. The measured timeout period must be within the total clock low-level extended period (master device) $T_{LOW:MEXT}$: 10 ms (max.) of the SMBus specification, and the total of all $T_{LOW:MEXT}$ from START condition to STOP condition must be within $T_{LOW:SEXT}$: 25 ms (max.).

For the ACK receive timing (rising edge of the ninth SCL clock cycle), monitor the BST.TENDF flag in master transmit mode (master transmitter) and the NTST.RDBFF0 flag in master receive mode (master receiver). For this reason, perform bytewise transmit operation in master transmit mode, and hold the SCSTRCTL.ACKTWE bit 0 until the byte just before reception of the final byte in master receive mode. While the ACKTWE bit = 0, the RDBFF0 flag is set to 1 at the rising edge of the ninth SCL clock cycle.

If the period measured with the GPT exceeds the total clock low-level extended period (master device) $T_{LOW:MEXT}$: 10 ms (max.) of the SMBus specification or the total of measured periods exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (min.) of the SMBus specification, the master device must stop the transaction by issuing a STOP condition. In master transmit mode, immediately stop the transmit operation (writing data to NTDTBP0).

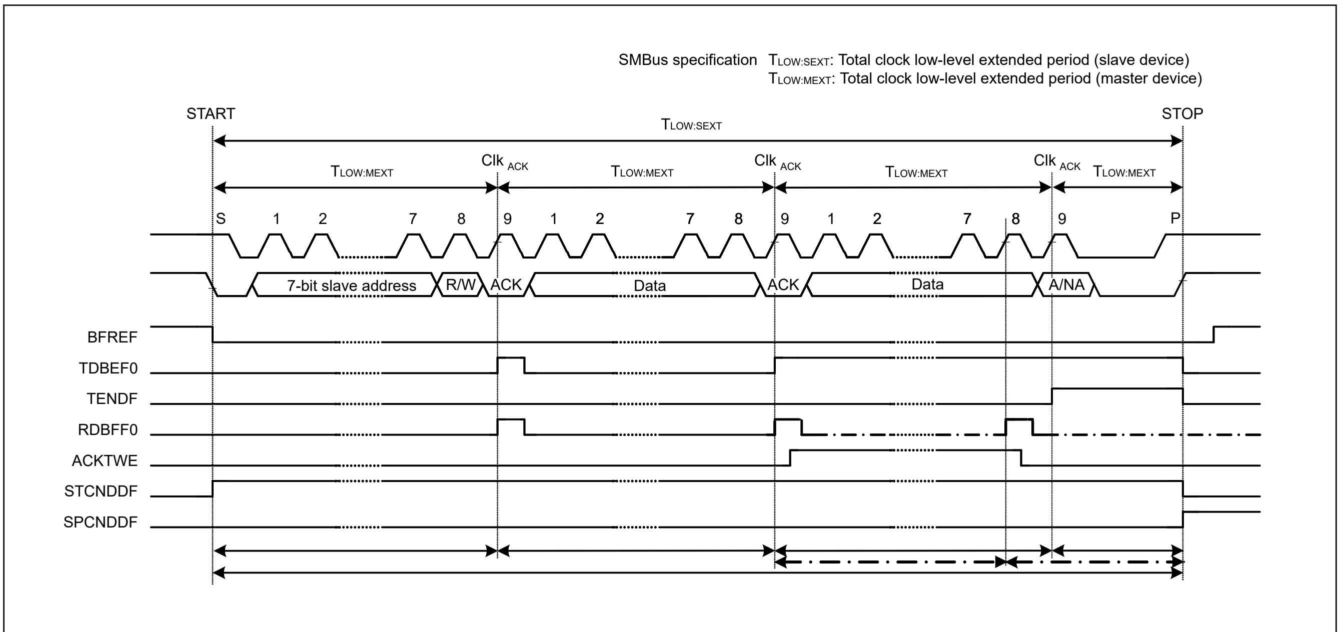


Figure 40.114 SMBus timeout measurement

(2) Packet Error Code (PEC)

This MCU incorporates a CRC calculator. The CRC calculator enables transmission of a packet error code (PEC) or checking the received data of the SMBus in data communication of I3C. For the CRC generating polynomials of the CRC calculator, see section 48, Cyclic Redundancy Check (CRC).

The PEC data in master transmit mode can be generated by writing all transmit data to the CRC data input register (CRCDIR) in the CRC calculator.

The PEC data in master receive mode can be checked by writing all receive data to CRCDIR in the CRC calculator and comparing the obtained value in the CRC data output register (CRCDOR) with the received PEC data.

To send ACK or NACK according to the match or mismatch result when the final byte is received as a result of the PEC code check, set the SCSTRCTL.ACKTWE bit to 1 before the rising edge of the eighth SCL clock cycle during reception of the final byte, and hold the I3C_SCL line low at the falling edge of the eighth clock cycle.

(3) SMBus Host Notification Protocol (Notify ARP Master Command)

In communications over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of its own slave address or to request its own slave address from the SMBus host.

For a product of this MCU to operate as an SMBus host (or ARP master), the host address (0001 000) sent from the slave device must be detected as a slave address, so I3C has a function for detecting the host address. To detect the host address as a slave address, set the BFCTL.SMBS bit and the SVCTL.HOAE bit to 1. Operation after the host address has been detected is the same as normal slave operation.

40.3.2.3.13 Common Command Codes (CCC) [I3C Mode]

For the common command code (CCC), refer to 5.1.9 Common Command Codes (CCC) in MIPI I3C Specification v1.0. I3C is based on Table 15 I3C Common Command Codes in 5.1.9.3 Common Command Definitions of MIPI I3C Specification v1.0.

The MIPI Reserved area and Vendor Extension area of Command Code are described below.

I3C Master mode :

When sending CCCs in the MIPI Reserved area and Vendor Extension area from the I3C Master, only Broadcast/Direct SET CCCs using the Immediate Transfer Command can be sent.

Sending Direct GET CCC is not supported.

I3C Slave mode :

Only Broadcast/Direct SET CCC can be received for CCC in MIPI Reserved area and Vendor Extension area.

Receiving Direct GET CCC is not supported.

40.3.2.4 Error Detection

40.3.2.4.1 SDR Error Detection and Recovery Methods for I3C Slave Devices [I3C Mode]

The seven error types summarized in Table 40.13 are supported for all I3C slave devices. Each error type is further explained below the table.

Table 40.13 SDR slave error types

Error type	Description	Error detection method	Error recovery method
S0	Broadcast address/W (= 0x7E/W) or Dynamic address/RW	Detect any of the following: 0x3E / W 0x5E / W 0x6E / W 0x76 / W 0x7A / W 0x7C / W 0x7F / W 0x7E / R	Enable HDR EXIT Detector and ignore all other patterns
S1	CCC code	Parity check, using T-Bit	Enable HDR EXIT detector and neglect other patterns
S2	Write data	Parity check, using T-Bit	Enable STOP detector and neglect other patterns
S3	Assigned address during Dynamic address arbitration	Parity check, using PAR Bit	Generate NACK (after PAR), then wait for another Repeated START and 7E/R to re-transmit the Provisional ID
S4	0x7E/R after Sr during Dynamic address arbitration	Detect any value other than 0x7E/R after Sr during Dynamic Address Arbitration	Generate NACK (after 0x7E/R), then enable STOP Detector and ignore all other patterns
S5	Transaction after detecting CCC	Detect illegally formatted CCC	Generate NACK (after Slave Address), then enable STOP Detector and ignore all other patterns
S6 (optional)	Monitoring error	Slave detects (through monitoring) that transmitted Data differs from what it intended to transmit (Does not apply during Dynamic address arbitration)	Stop the transmission, then enable STOP Detector and ignore all other patterns

40.3.2.4.2 SDR Error Detection and Recovery Methods for I3C Master Devices [I3C Mode]

The two error types summarized in Table 40.14 are supported for all I3C master devices. Each error type is further explained below the table.

Table 40.14 SDR master error types

Error type	Description	Error detection method	Error recovery method
M0	Transaction after sending CCC	Detect illegally formatted CCC	Stop the transmission, then send STOP and retry the transmission.
M1 (optional)	Monitoring error	Master detects (through monitoring) transmitted data different from what it intended to transmit (Does not apply during Dynamic address arbitration)	Stop the transmission, then send STOP and retry the transmission.
M2	No response to Broadcast address (0x7E)	Master detects NACK after Broadcast address (0x7E) transmission	Upon detection of NACK, master transmits HDR exit pattern followed by STOP

40.3.2.4.3 HDR-DDR Error Detection

Four error types are defined for HDR-DDR mode.

Table 40.15 HDR-DDR error types

Error type	Description	Error detection method	Error recovery method
Framing	The Preamble 2-bits before command and data	The Slave detects that the Preamble 2-bits before command and data is not valid values.[*1]	Master: Issue SCL clocks until 19 SCL clocks (38 bits) have been seen with SDA High. The Master shall then Park SCL Low, and emit an HDR Exit Pattern using SDA. Slave: Wait for HDR EXIT Pattern
Parity checking	Parity error	The Master and the Slave detects mismatched parity.[*2]	
CRC5 checking	CRC5 error	The Master and the Slave detects mismatched CRC.[*3]	
NACK receiving	NACK by the Slave on a Read command	The Master detects a NACK by the Slave on a Read command.[*4]	Issue SCL clocks until 19 SCL clocks (38 bits) have been seen with SDA high. The Master shall then Park SCL low, and emit either an HDR exit pattern or an HDR restart pattern using SDA.
Monitoring	Monitoring error	Both the Master and the Slave should always monitor the data that they each transmit, and they detect the monitored data differs from the data that the Master or Slave intended to send.[*5]	Both the Master and the Slave can stop the transmission and then Master and Slave should each do the following. Master: Issue SCL clocks until 19 SCL clocks (38 bits) have been seen with SDA High. The Master shall then Park SCL Low, and emit an HDR Exit Pattern using SDA. After the Master sends HDR EXIT, it can retry the transmission. Slave: Wait for HDR Exit Pattern

Note 1. This supports a positional error detection mechanism:

- A Command Word shall always follow the Enter HDR CCC and HDR Restart Pattern, and shall never appear in any other position. It is an error condition for a Command Word to appear in any other position, or to be missing where expected.
- A Data Word shall always follow either a Command Word or another Data Word, and shall never appear in any other position. It is an error condition for a Data Word to appear in any other position, or to be missing where expected.
- A single CRC Word shall always follow the last Data Word for a Command, such that it ends the Message. As a result, a CRC Word shall always be followed by either the HDR Restart Pattern or the HDR Exit Pattern. It is error condition for a CRC Word to appear in any other position, or to be missing where expected.
- The first nibble of a valid CRC shall contain the allowed value 0xC. Any other value in the first nibble should be considered a framing error.

Note 2. Parity encoding (for transmitters) parity checking (for receivers) shall be performed on all Command Words and Data Words. Mismatched parity is an error condition.
Note 3. CRC5 encoding (for transmitters) and CRC5 checking (for receivers) shall be performed on all payload bits for Command Words and Data Words. Mismatched CRC is an Error.
Note 4. NACK by the Slave on a Read command is not a normal behavior (ACK is normal). The Master may choose to treat a NACK on a Read command as a possible framing error. The Master can choose to treat the NACK of a Read command as a possible line error, and therefore use the same approach in order to ensure that the Slave is not driving the Bus.
Note 5. If the Master or Slave performs such monitoring, and if the monitored Data differs from the Data that the Master or Slave intended to send (except for Data transferred during a Dynamic Address Arbitration procedure), then this shall be considered an error.

40.3.2.4.4 HDR-TSP/TSL Error Detection

Two error types are defined for HDR-TSP/TSL Mode.

Table 40.16 HDR-TSP/TSL error types

Error type	Description	Error detection method	Error recovery method
Symbol 2 checking	Continuous error of symbol 2	More than one symbol 2 in a row. (In symbol 2 the I3C_SCL line does not change, and the I3C_SDA line does change.) *1	Master: Wait until the Slave stops transitioning the Bus for 2x the maximum edge-to-edge duration used by that Slave. Then the Master shall force out the HDR Exit Pattern. Slave: Wait for HDR Exit Pattern.
Parity checking	Parity errors	The Slave detects mismatched parity. *2	
Monitoring	Monitoring error	Both the Master and the Slave should always monitor the data that they each transmit, and they detect the monitored data differs from the data that the Master or Slave intended to send.*3	Both the Master and the Slave can stop the transmission and then Master and Slave should each do the following. Master: Wait until the Slave stops transitioning the Bus for 2x the maximum edge-to-edge duration used by that Slave. Then the Master shall force out the HDR Exit Pattern. After the Master sends HDR EXIT, it can retry the transmission. Slave: Wait for HDR EXIT Pattern.

Note 1. Exception: More than one Symbol 2 in a row is allowed for HDR Restart or HDR Exit, but only from a known starting state (SCL Low and SDA High), and only at a Data Word boundary (allowing for one Symbol to set up that state). Thus the Symbol pattern 2,2 is only allowed under the following situations:
- As part of the HDR Exit Pattern, which uses 3 or 4 such Symbol pairs and involves SCL Low
- As part of the HDR Restart pattern, which uses 2 such Symbol pairs and involves SCL Low
- As the result of one normal Symbol ending in 2, and the next Symbol starting with 2

Note 2. An error may be seen by an HDR Slave, or by an HDR Master when a Slave is driving the I3C Bus:
- A Slave that sees an error shall stop tracking Symbols and use HDR Exit and HDR Restart Pattern detection. This means to wait until safe. A Slave shall enable the HDR Exit and HDR Restart Pattern Detector upon entry to any HDR Mode, ensuring that it is always safe.
- A Master that sees an error shall wait until the Slave stops transitioning the Bus for 2x the maximum edge-to-edge duration used by that Slave. Then the Master shall force out the HDR Exit Pattern.

Note 3. If the Master or Slave performs such monitoring, and if the monitored Data differs from the Data the Master or Slave intended to send (except for data transferred during the Dynamic Address Arbitration procedure), then this shall be considered an error.

Note: It could be appropriate to wait for the Symbol 2,2, but the Master would still need to see quiescence. If the Master is seeing errors, then the results of even a Symbol 2,2 are questionable.

40.3.2.4.5 Timeout Error Detection

I3C includes a timeout function for detecting when the I3C_SCL line has been stuck longer than the predetermined time. I3C can detect an abnormal bus state by monitoring that the I3C_SCL line is stuck low or high for a predetermined time.

The timeout function monitors the I3C_SCL line state and counts the low-level period or high-level period using the internal counter. The timeout function resets the internal counter each time the I3C_SCL line changes (rising or falling), but continues to count unless the I3C_SCL line changes. If the internal counter overflows due to no I3C_SCL line change, I3C can detect the timeout and report the bus hung state.

This timeout function is enabled when BSTE.TODE = 1. It detects a hung state that the I3C_SCL line is stuck low or high during the following conditions: (When TMOCTL.TOMDS[1:0] = 00b)

- The bus is busy (BCST.BFREF = 0) in master mode (PRSST.CRMS = 1).
- I3C's own slave address is detected (SVST register is not 0x0000) and the bus is busy (BCST.BFREF = 0) in slave mode (PRSST.CRMS = 0).
- The bus is free (BCST.BFREF = 1) while generation of a START condition is requested (CNDCTL.STCND = 1).

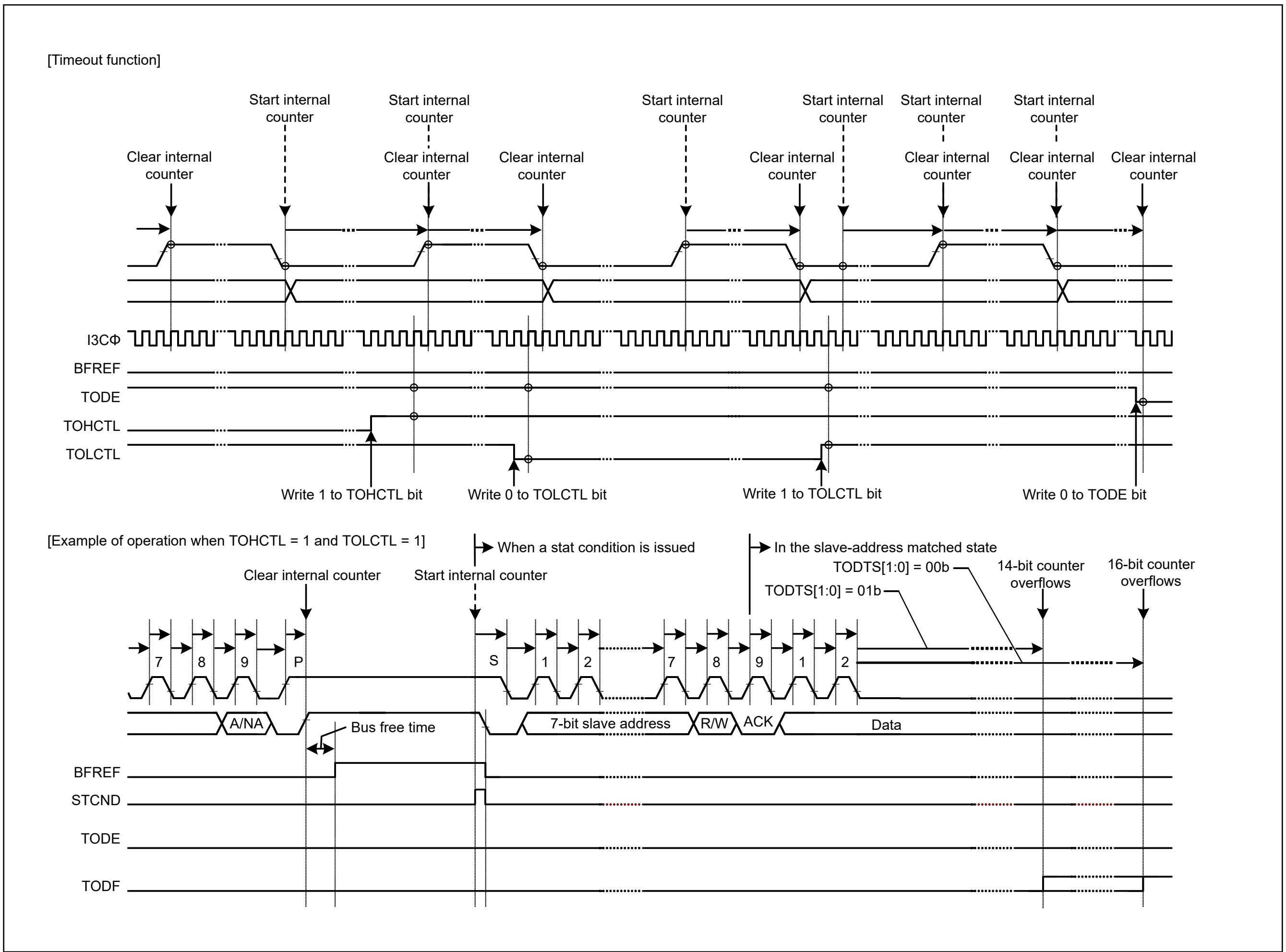


Figure 40.115 Timeout error detection (TODE, TODTS[1:0], TOHCTL, and TOLCTL bits)

40.3.2.4.6 Resume Operation [I3C Mode]

I3C enters the Halt state as a result of any type of error occurring in a transfer.

The error type is indicated by the field ERR_STATUS in Response Descriptor or Receive Status Descriptor. After I3C has entered the Halt state, the user must write the value 1 to the RSM bit to resume operation. I3C shall auto-clear the RSM bit once it has initiated the next Command transfer or detected the START condition.

40.3.2.4.7 Abort Operation [I3C Mode]

When the BCTL.ABT bit is set to 1, I3C relinquish control of the bus before completing the currently issued transfer. In response to an abort request, I3C issues the STOP condition on the bus after the complete data byte is transferred or received. After I3C has aborted, the user shall clear the BCTL.ABT bit to allow operation on the bus.

Note: For Read transaction, when BCTL.ABT is set to 1, that receive data is stored in Rx data buffer. However, for HDR-TSP / TSL, receive data received after that is not stored.

Abbreviations

Pa: Parity

Pr: Preamble

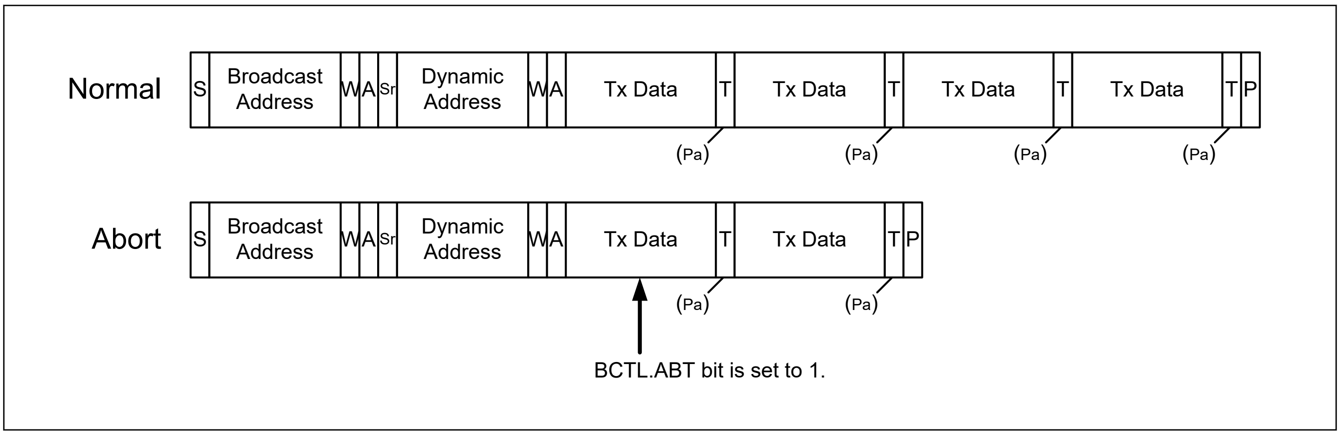


Figure 40.116 Abort operation of SDR write transfer

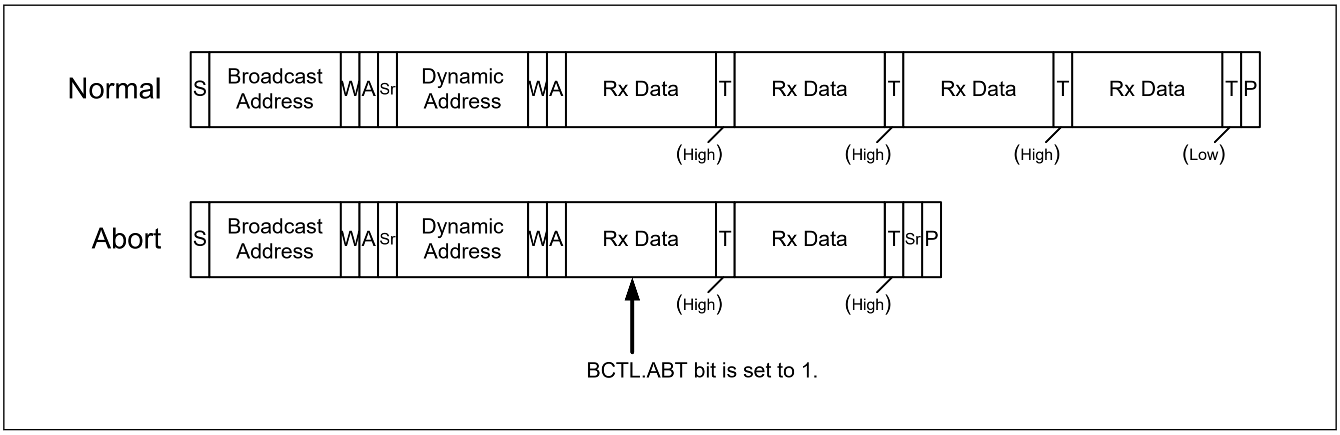


Figure 40.117 Abort operation of SDR read transfer

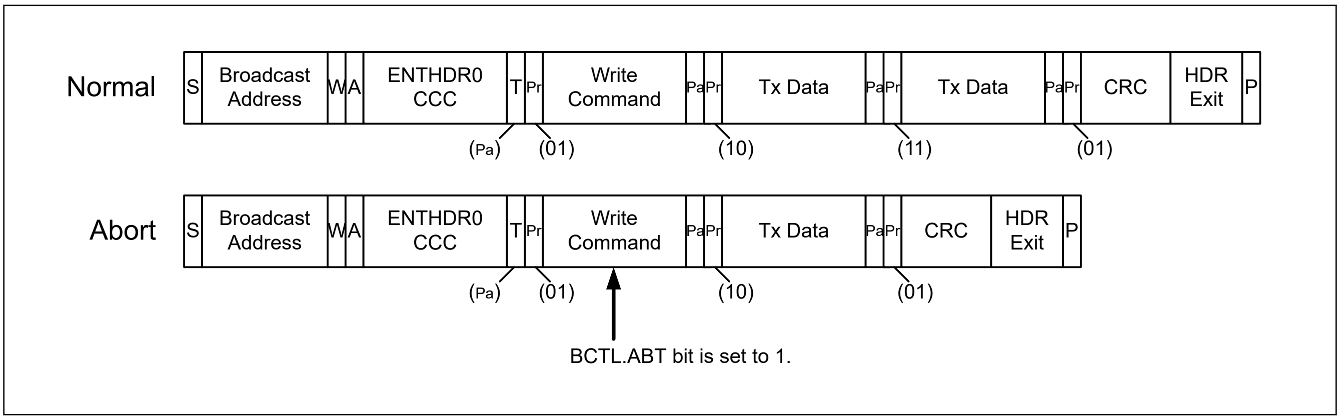


Figure 40.118 Abort operation of HDR-DDR write transfer

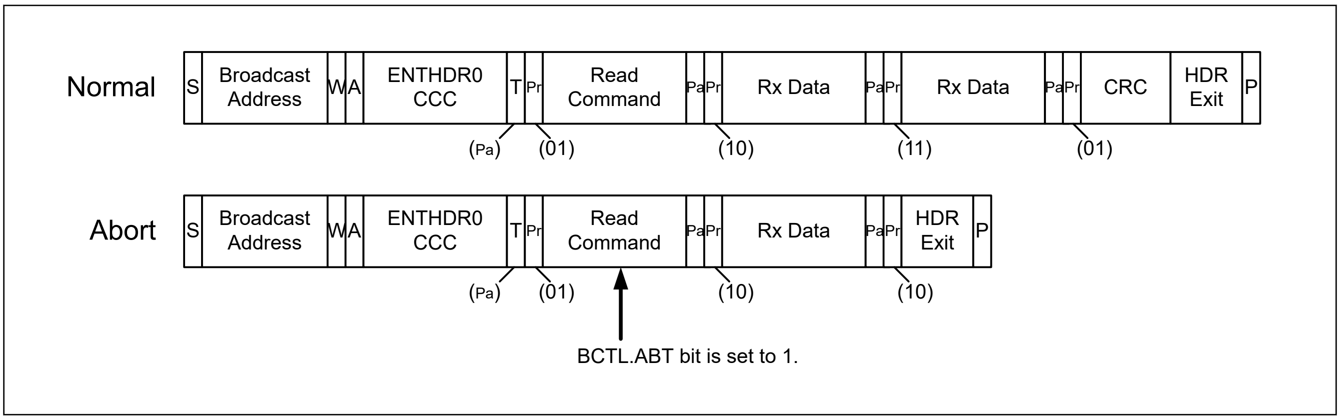


Figure 40.119 Abort operation of HDR-DDR read transfer

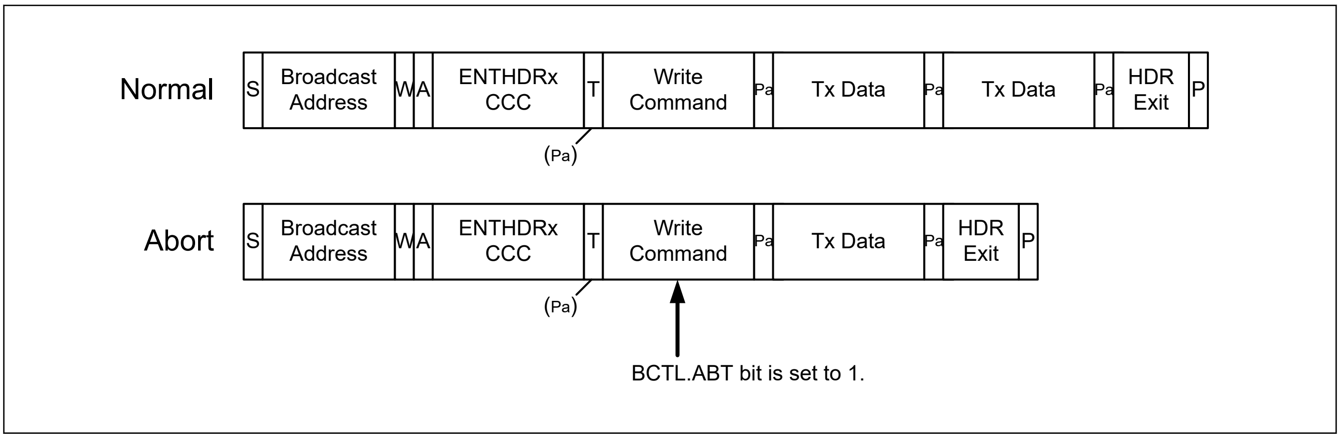


Figure 40.120 Abort operation of HDR-TSP/TSL write transfer

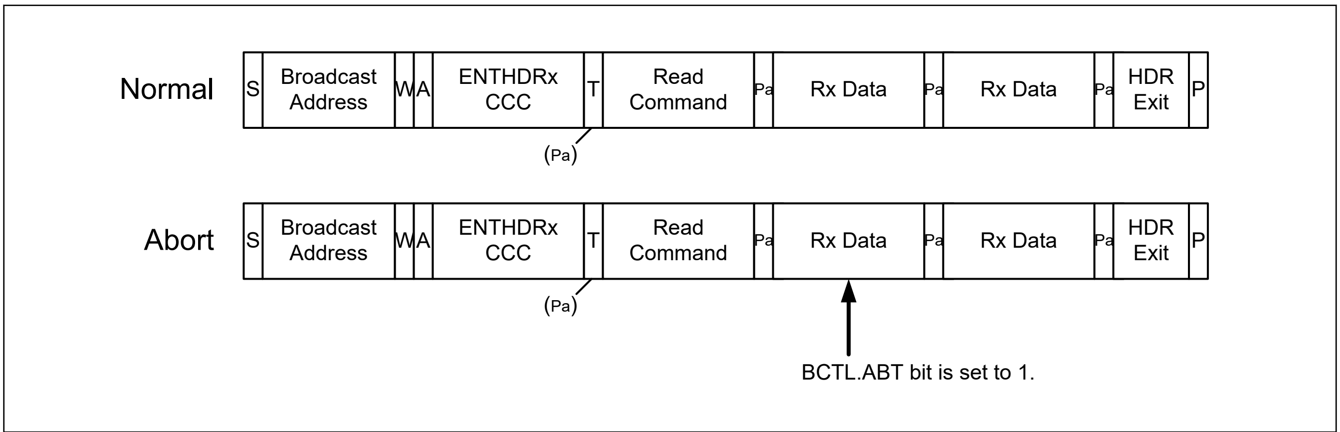


Figure 40.121 Abort operation of HDR-TSP/TSL read transfer

40.3.2.4.8 Error Recovery Operation

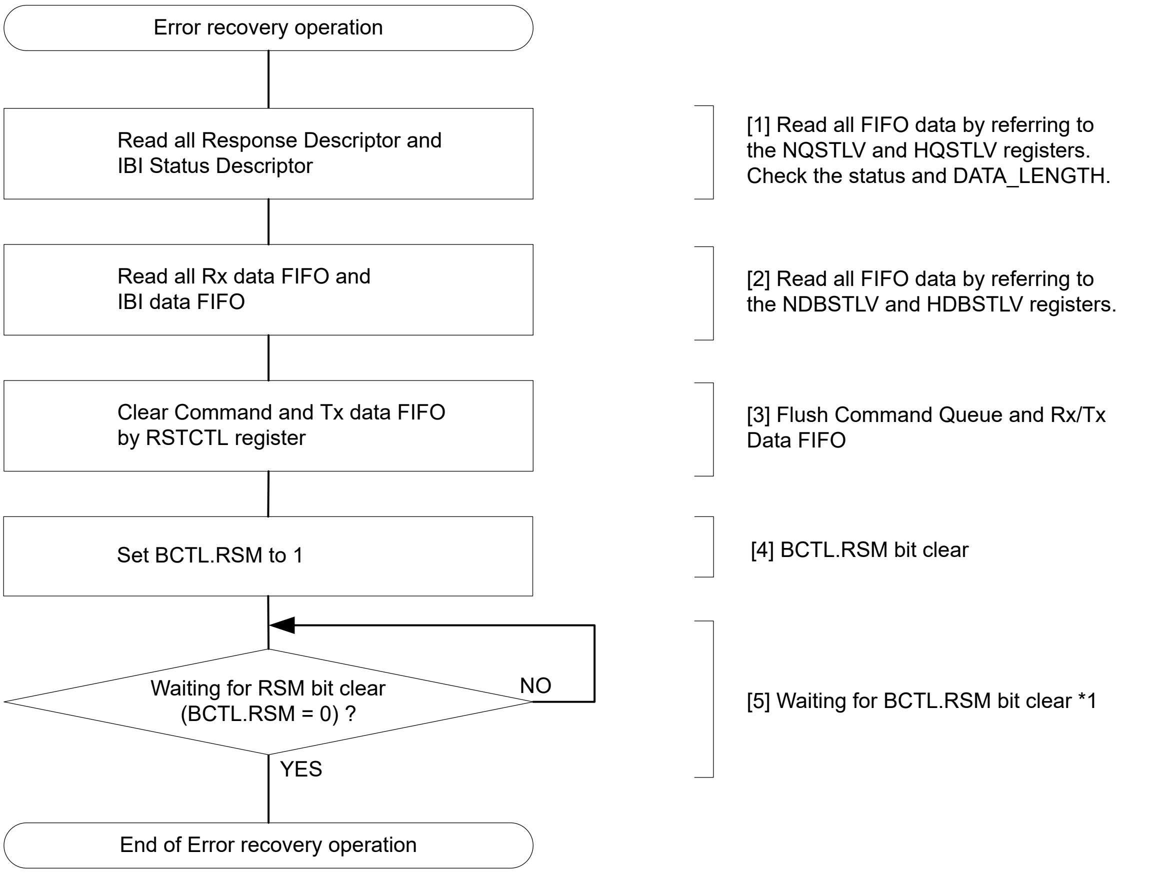
(1) Error Recovery Operation

When an error occurs, the INST.INEF, NTST.TEF, NTST.TABTF, HTST.TEF and HTST.TABTF flags are set to 1 according to the cause of the error, or the interrupts associated with each flag are asserted (when detection and interrupts are enabled.)

There is a possibility of communication error or internal module error.

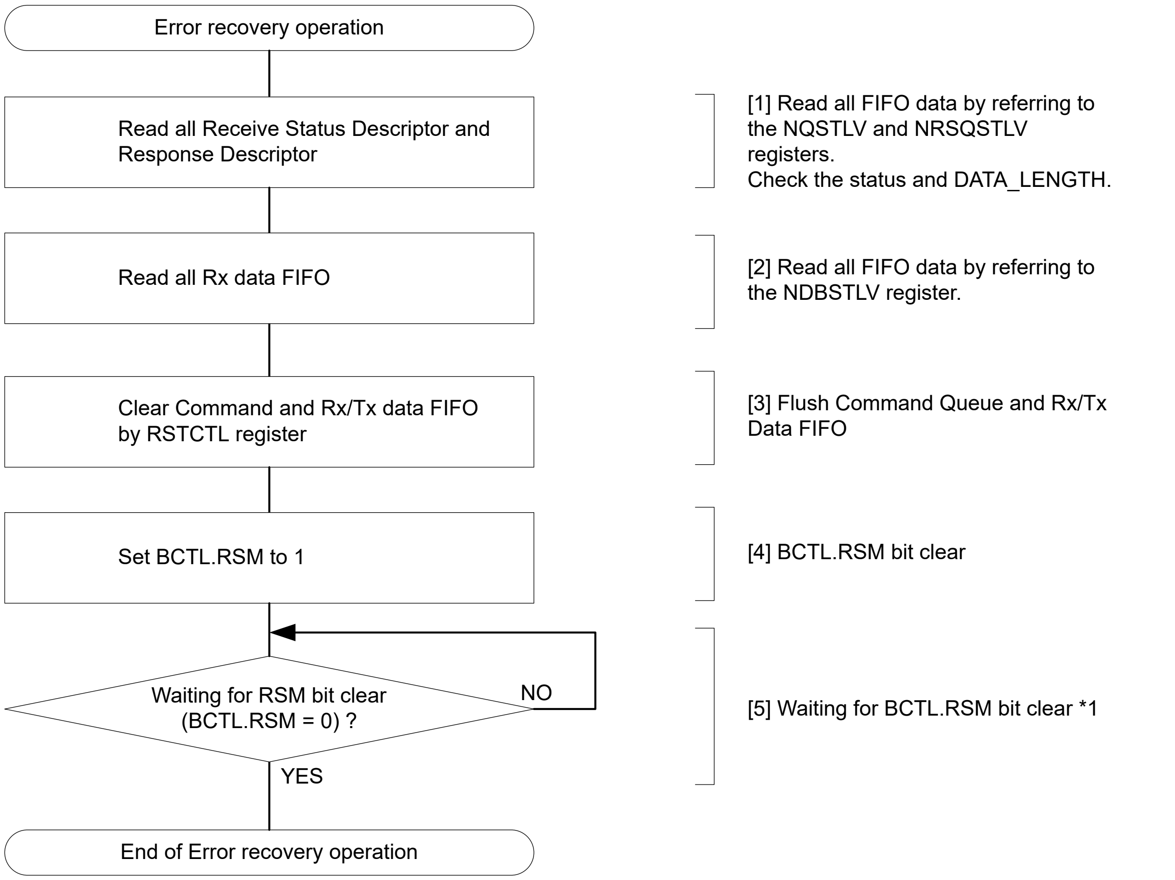
Note: If an error occurs, I3C will be suspended (BCTL.RSM becomes 1). After I3C is suspended, the application must write the value 1 to the BCTL.RSM bit to resume I3C operation and recover from the suspended state.

Figure 40.122 and Figure 40.123 show the error recovery flow.



Note 1. The Command Descriptor can be written to the Command Queue via writing to the Command Queue Port (NCMDQP or HCMDQP) before the RSM bit is cleared.

Figure 40.122　Example of error recovery operation flowchart for I3C master



Note 1. The Command Descriptor for issuing IBI can be written to the Command Queue via writing to the Command Queue Port (NCMDQP) before the RSM bit is cleared.

Figure 40.123 Example of error recovery operation flowchart for I3C slave

When I3C Slave recovers from an error according to the error recovery flow, after setting BCTL.RSM to 1, BCTL.RSM becomes 0 after detecting a state in which Bus Available period communication is not performed on I3C Bus.

If communication occurs on the I3C bus within the Bus Available period, BCTL.RSM will not be set to 0 and error recovery will not be completed, NACK response will be made to the communication.

(2) Master Error Detection and Escalation Handling

If the Master does not receive an ACK of a transmitted private Message to a Slave and Steps 1 and 2 described in Chapter 5.1.10.2.4 of MIPI I3C Spec v1.0 fail, the processing flow of Step 3 is shown in Figure 40.124 and Figure 40.125.

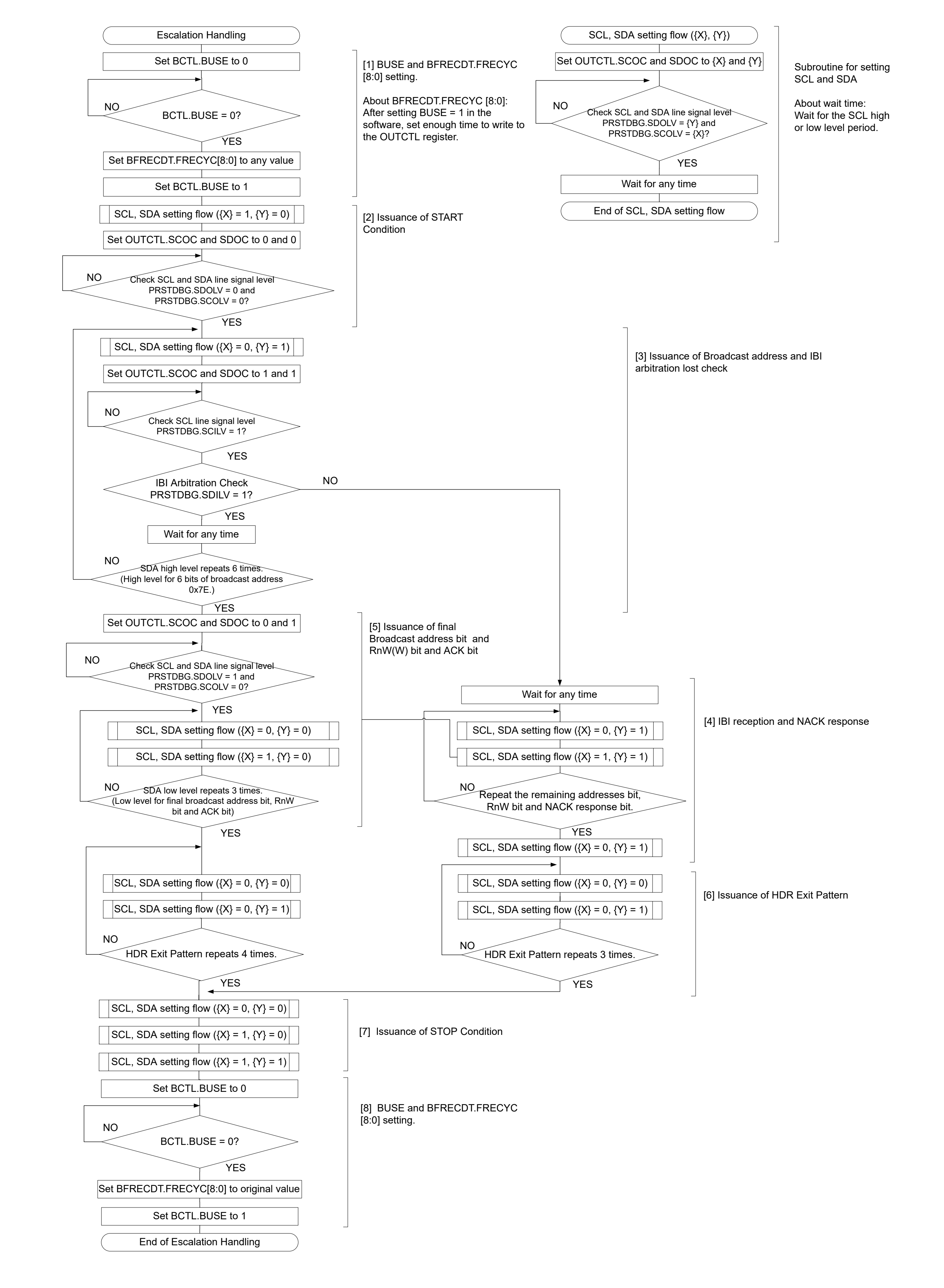


Figure 40.124 Escalation handling flowchart for I3C master

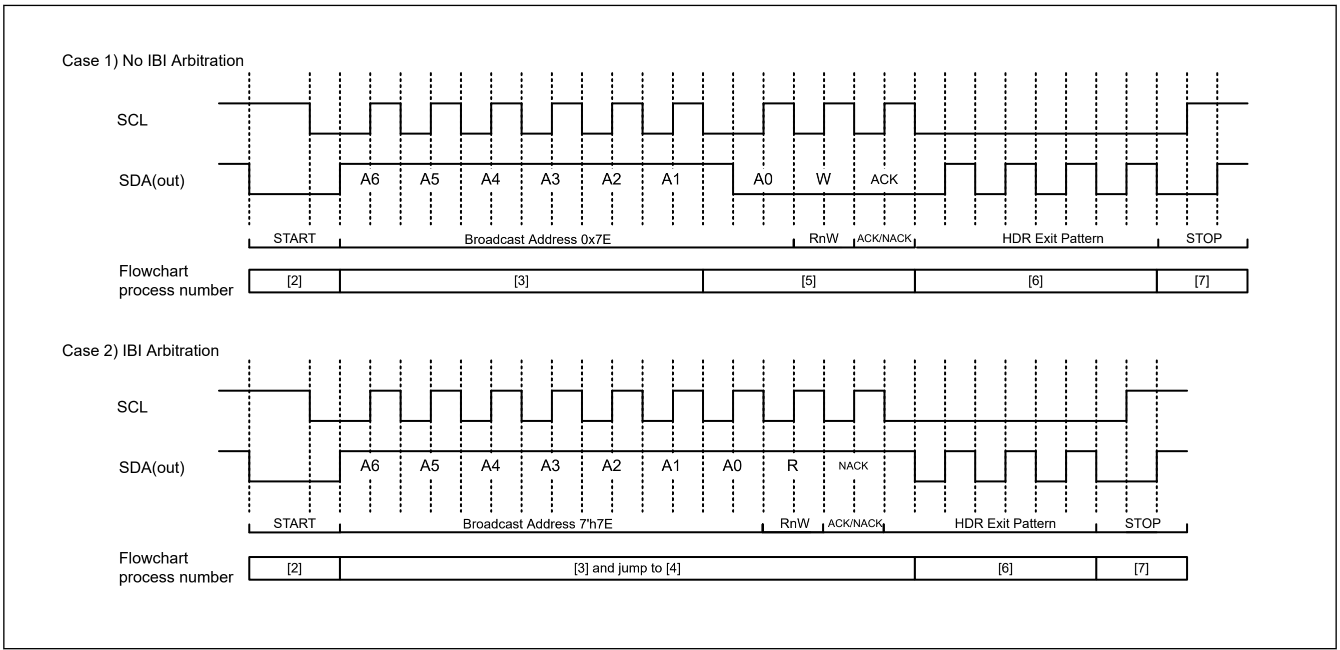


Figure 40.125 I3C Escalation Handling timing chart for I3C Master

40.3.2.5 Low Power Function

40.3.2.5.1 Wake Up Function [I^{2}C Mode]

I3C is equipped with the Wake-up function that causes the microcomputer to transition from low power consumption mode with system clock (ICLK) is stopped (software standby mode, etc.) to the normal operation. The Wake-up function is used to generate a Wake-up interrupt signal when the received data matches the address set to Wake-up interrupt factor also receives data in a state where the operating clock (PCLK/TCLK) is stopped (PCLK/TCLK asynchronous operation). This wake-up interrupt signal causes the microcomputer to transition to the normal operation. After Wake-up interrupt occurs, switch I3C to PCLK/TCLK synchronous operation, it will be able to continue the communication operation.

The Wake-up function has four wake-up operation modes (normal WU mode 1, normal WU mode 2, command recovery mode, and EEP response mode). The table below describes the behavior in these four wake-up operation modes.

Table 40.17 Wake-up operation mode

	ACK response timing	ACK Type responded before recovery to PCLK/TCLK synchronous operation	SCL state before recovery to PCLK/TCLK synchronous operation
Normal WU mode 1	Before recovery to PCLK/TCLK synchronous operation*1	ACK	Fixed to L
Normal WU mode 2	After recovery to PCLK/TCLK synchronous operation*2	Before recovery: no response (NACK level retained) After recovery: ACK response	Fixed to L
Command recovery mode	Before recovery to PCLK/TCLK synchronous operation*1	ACK	Open
EEP response mode	Before recovery to PCLK/TCLK synchronous operation*1	NACK	Open

Note 1. Switching timing from PCLK/TCLK asynchronous operation to PCLK/TCLK synchronous operation is the fall of the 9th clock of SCL.
Note 2. Switching timing from PCLK/TCLK asynchronous operation to PCLK/TCLK synchronous operation is the fall of the 8th clock of SCL.

The following is able to select as Wake-Up interrupt factor.

- Host address detection (valid when SVCTL.HOAE = 1)
- General call address detection (valid when SVCTL.GCAE = 1)
- Slave address 0*1 detection (valid when SVCTL.SVAE[0] = 1)
- Slave address 1*1 detection (valid when SVCTL.SVAE[1] = 1)

- Slave address 2[*1] detection (valid when SVCTL.SVAE[2] = 1)

Note 1. 7-bit address only can be set. Set SDADLS bit to 0 in SDATBASn.

(1) Normal Wake-Up mode 1

This section describes the behavior, the timing, and a use case of normal WU mode 1.

A wake-up interrupt triggered by the match of the slave address makes the transition to the normal operation in the manner described below. Also, the detailed timing is provided in Figure 40.128.

Before wake-up recovery:	ACK is sent in response to the data received with its own slave address.
During wake-up recovery:	ACK response is made at the 9th clock cycle of SCL, and the SCL is held low afterwards.[*1]
After wake-up recovery:	Normal operation continues.

Note 1. Between ninth clock cycle and first clock cycle during Wake-Up recovery, SCSTRCTL.RWE = 1 does not work.

If the slave address does not match, the I3C_SCL line is not held low after the fall of the 9th clock cycle of SCL, and the slave operation continues.

See Figure 40.126 below for a use case.

A wake-up interrupt is not generated at the transition to the normal operation if the transition is triggered by a cause (other recovery causes (IRQ)) other than a wake-up interrupt signal generated by a slave address match. BST.WUCNDDF is not set in this case. Carry out the following processing according to Figure 40.127.

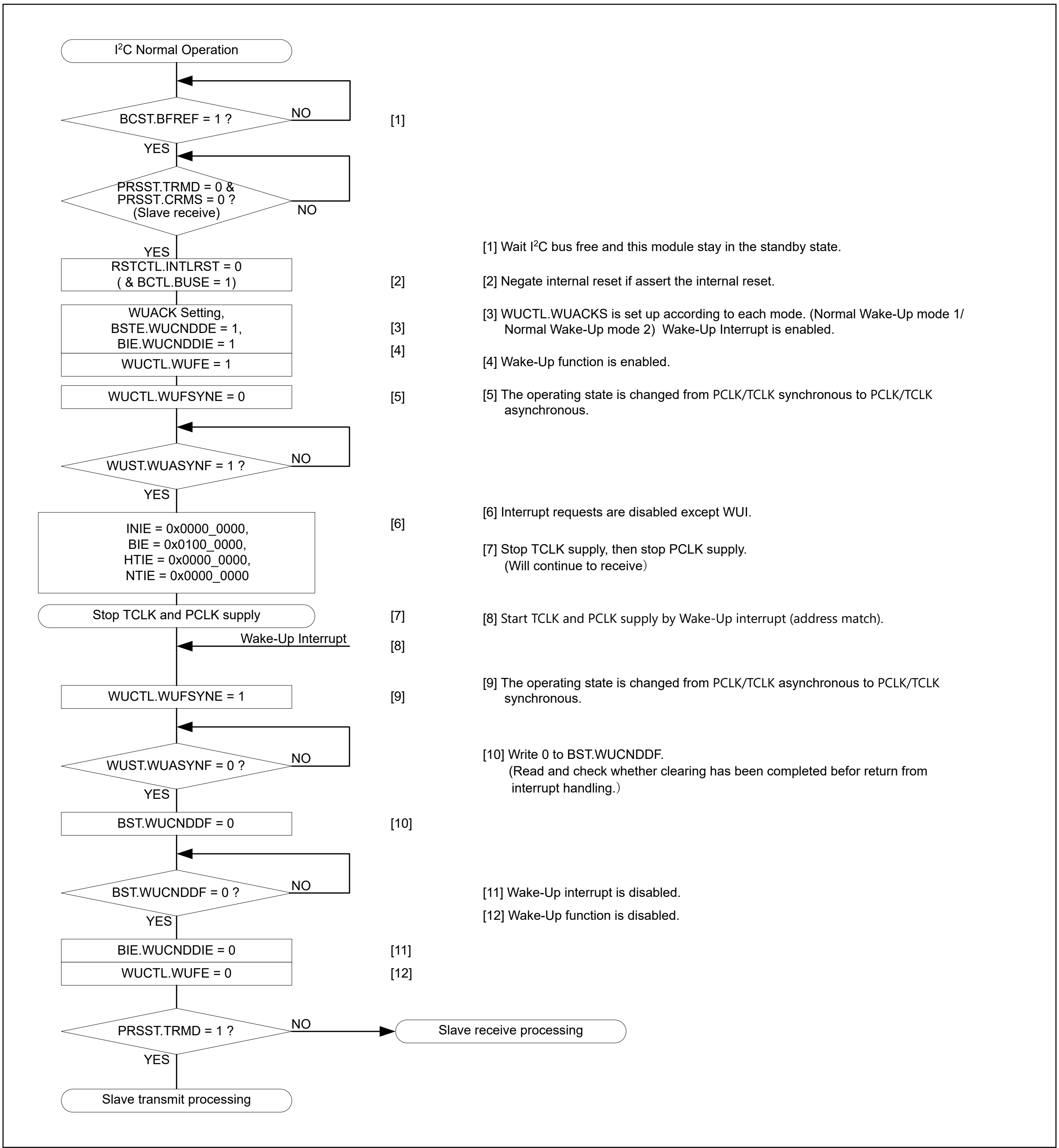


Figure 40.126 Use case of normal WU mode 1 (wake-up recovery by a wake-up interrupt triggered by the match of the slave address)

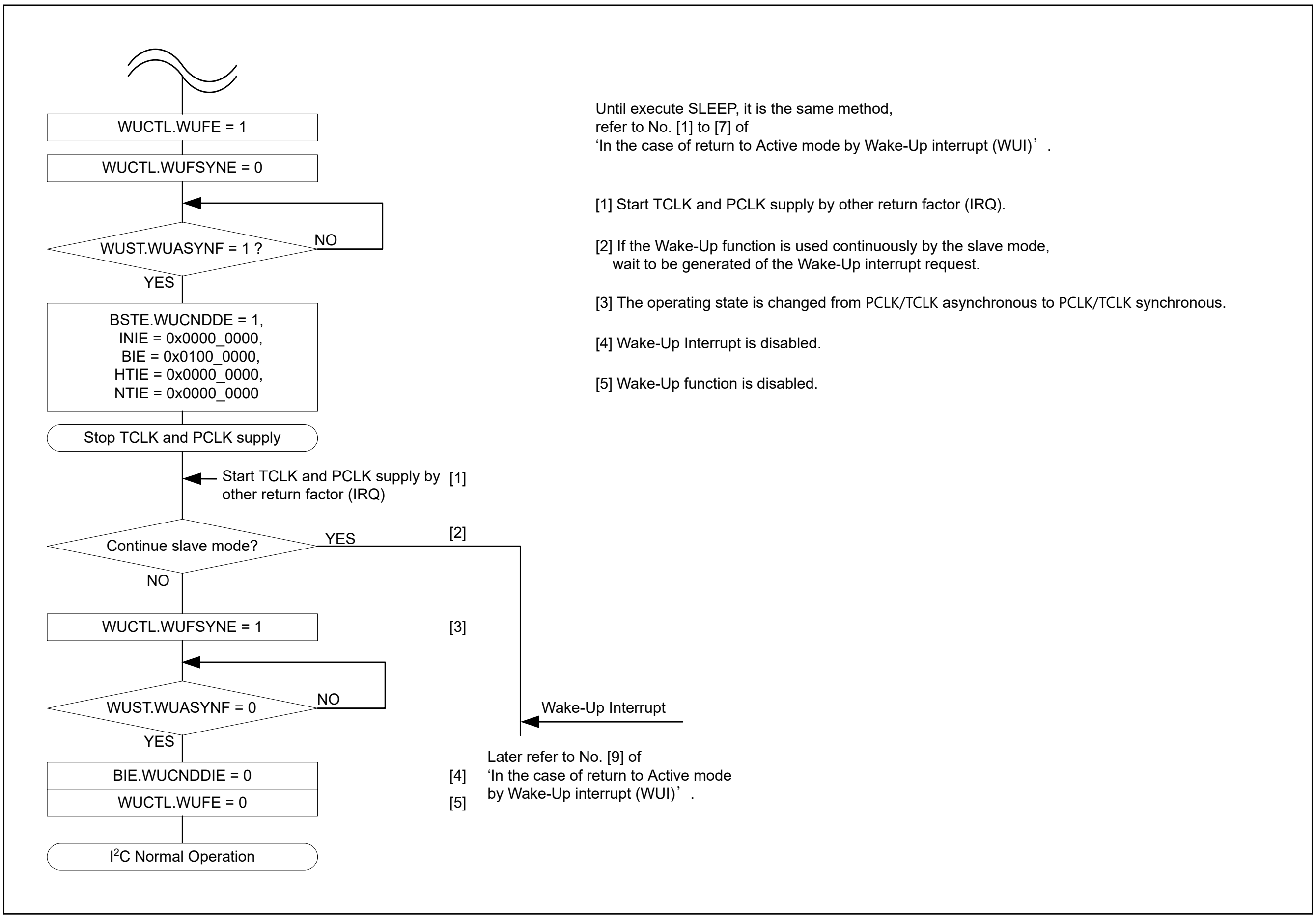


Figure 40.127 Use case of normal WU modes 1 and 2 (wake-up recovery by other recovery causes (IRQ))

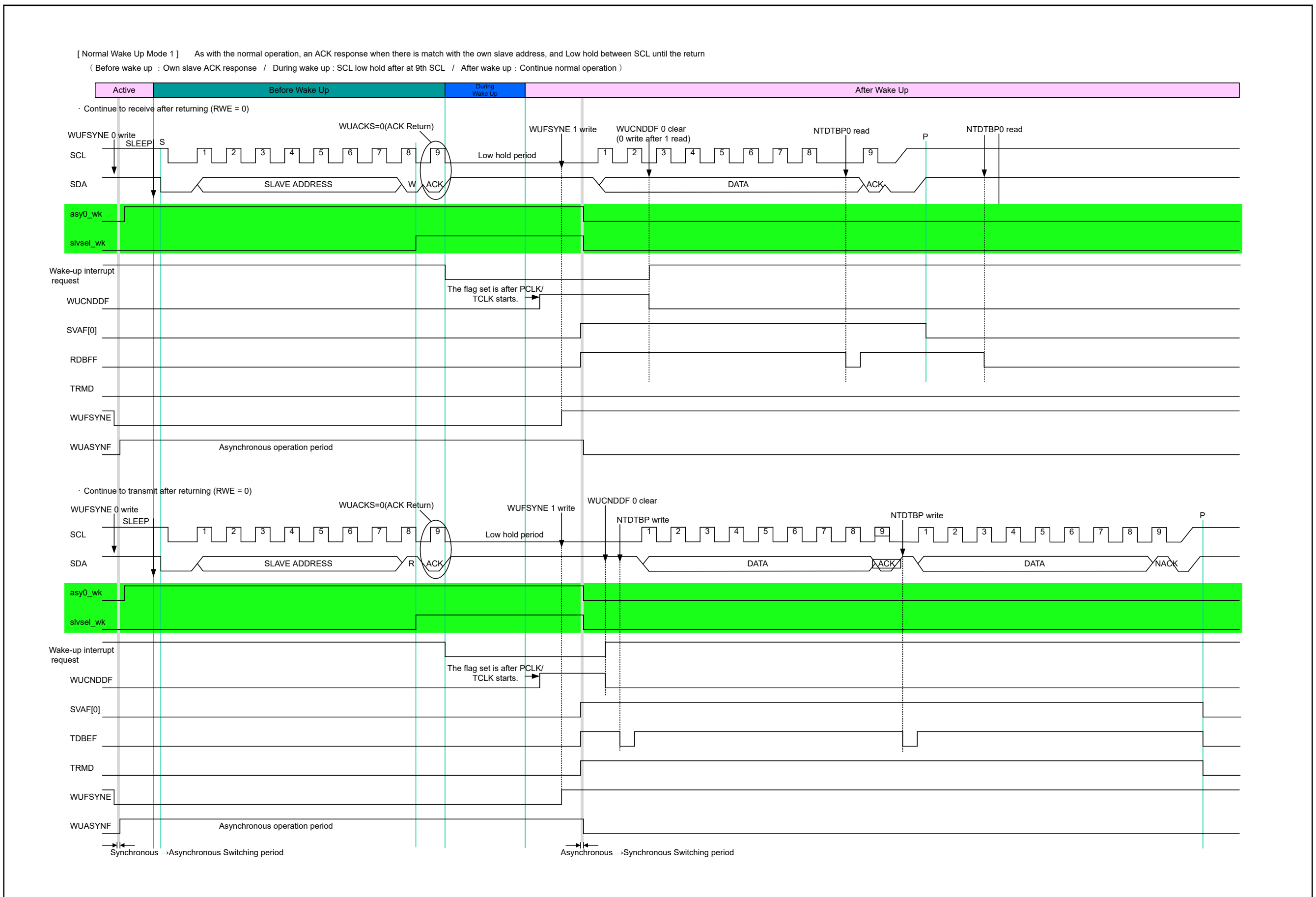


Figure 40.128 Timing of normal wake up mode 1

(2) Normal Wake Up Mode 2

This section describes the behavior, the timing, and a use case of normal WU mode 2.

A wake-up interrupt triggered by the match of the slave address makes the transition to the normal operation in the manner described below.

Also, the detailed timing is provided in Figure 40.130.

Before wake-up recovery:	No response to the data received with its own slave address (until 8th SCL cycle end)
During wake-up recovery:	Holding the I3C_SCL line low during the 8th and 9th clock cycles
After wake-up recovery:	Returning ACK at the 9th clock cycle of SCL, and continuing the normal operation

If the slave address does not match, the I3C_SCL line is not held low after the fall of the 8th SCL v clock cycle. The slave operation continues.

See Figure 40.129 below for a use case.

A wake-up interrupt is not generated at the transition to the normal operation if the transition is triggered by a cause (other recovery causes (IRQ)) other than a wake-up interrupt signal generated by a slave address match. BST.WUCNDDF is not set in this case. Carry out the following processing according to Figure 40.127.

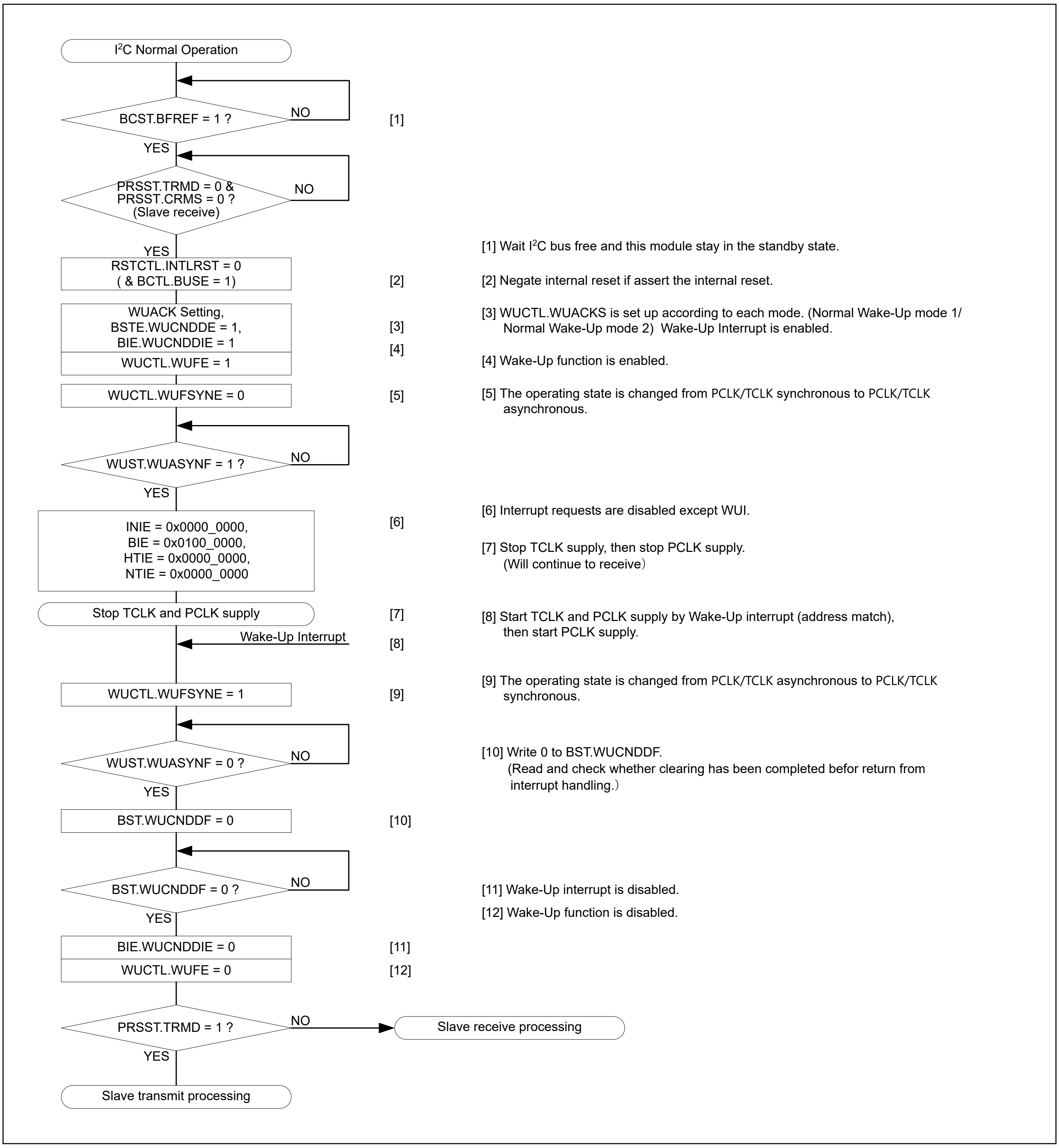


Figure 40.129 Use case of normal WU mode 2 (wake-up recovery by a wake-up interrupt triggered by the match of the slave address)

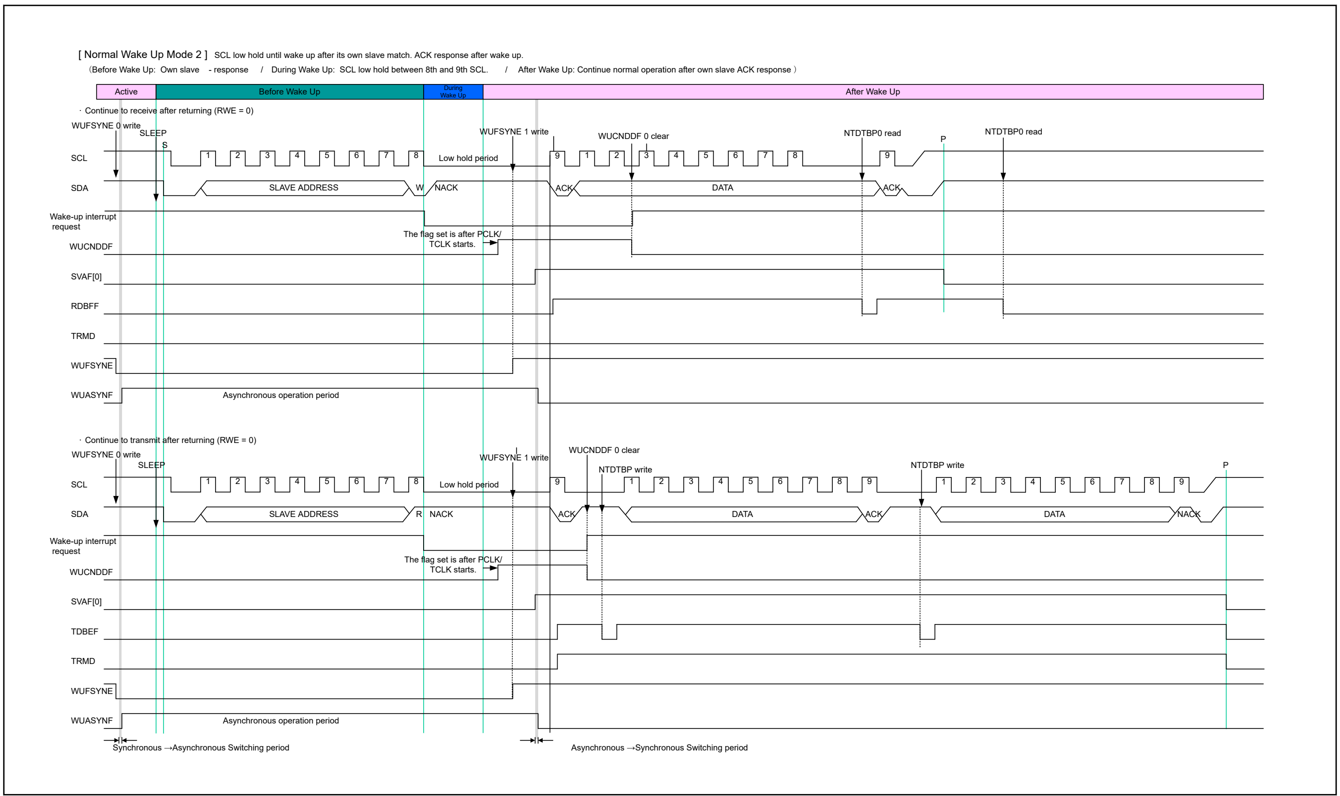


Figure 40.130 Timing of normal wake up mode 2

(3) Command recovery mode/ EEP response mode (Special Wake Up mode)

In the command recovery mode and EEP response mode, the I3C_SCL line is not held low during the wake-up recovery period (after the rise of the 9th clock cycle of SCL), so other I^{2}C/I3C devices can use the I^{2}C bus during this period. This section describes the behavior, the timing, and use cases of the command recovery mode and the EEP response mode.

A wake-up interrupt triggered by the match of the slave address makes the transition to the normal operation in the manner described below. Also, the detailed timing is provided in Figure 40.133.

Before wake-up recovery:	In response to the data received with its own slave address, ACK (command recovery mode) or NACK (EEP response mode) is returned.
During wake-up recovery:	The I3C_SCL line is not held low.
After wake-up recovery:	Normal operation continues after I3C initial setting.

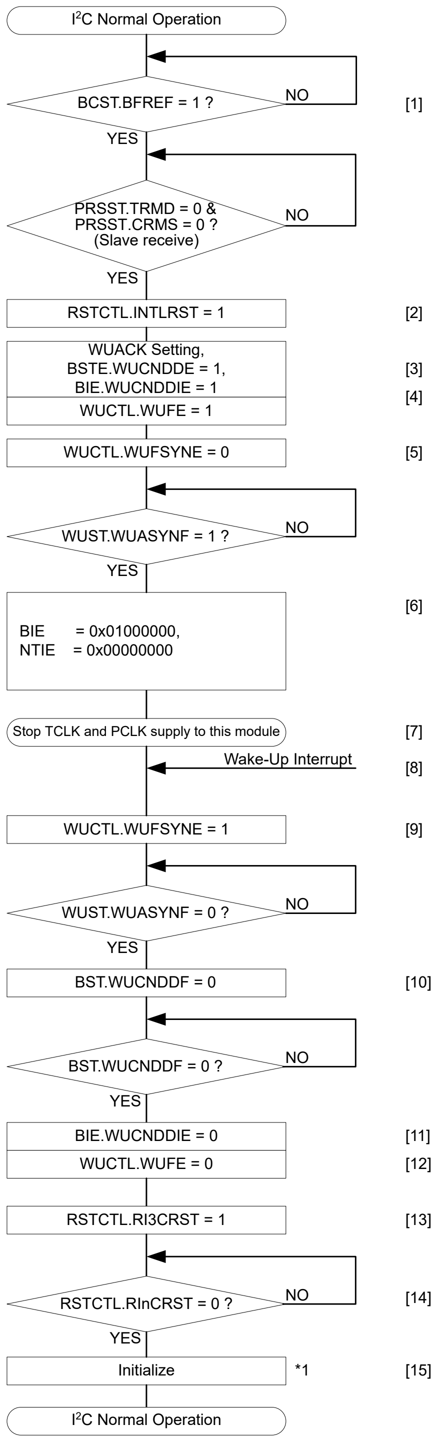
Note: Because the I3C_SCL line is not held low during wake-up recovery, the transmission/reception of the data that follows the slave address is not possible.

Note: The command recovery mode and the EEP response mode are internal reset (RSTCTL.INTLRST = 1) states. Therefore, the match of the slave address does not set the SVST flags (HOAF, GCAF, and SVAF[2:0]).

If the slave address does not match, the slave operation continues.

See Figure 40.132 below for a use case.

A wake-up interrupt is not generated at the transition to the normal operation if the transition is triggered by a cause (other recovery causes (IRQ)) other than a wake-up interrupt signal generated by a slave address match. BST.WUCNDDF is not set in this case. Carry out the following processing according to Figure 40.132.



[1] Wait I²C bus free and this module stay in the standby state.

[2] Assert the internal reset.

[3] WUCTL.WUACKS is set up according to each mode. (Command return mode/ EEP response mode) Wake-Up Interrupt is enabled.

[4] Wake-Up function is enabled.

[5] This module of operating state is changed from PCLK/TCLK synchronous to PCLK/TCLK asynchronous.

[6] Interrupt requests are disabled except WUI.

[7] Stop TCLK supply to this module, then stop PCLK supply to this module. (This module will continue to receive)

[8] Start TCLK and PCLK supply to this module by Wake-Up interrupt (address match), then start PCLK supply to this module.

[9] This module of operating state is changed from PCLK/TCLK asynchronous to PCLK/TCLK synchronous.

[10] Write 0 to BST.WUCNDDF. (Read and check whether clearing has been completed before return from interrupt handling.)

[11] Wake-Up interrupt is disabled.

[12] Wake-Up function is disabled.

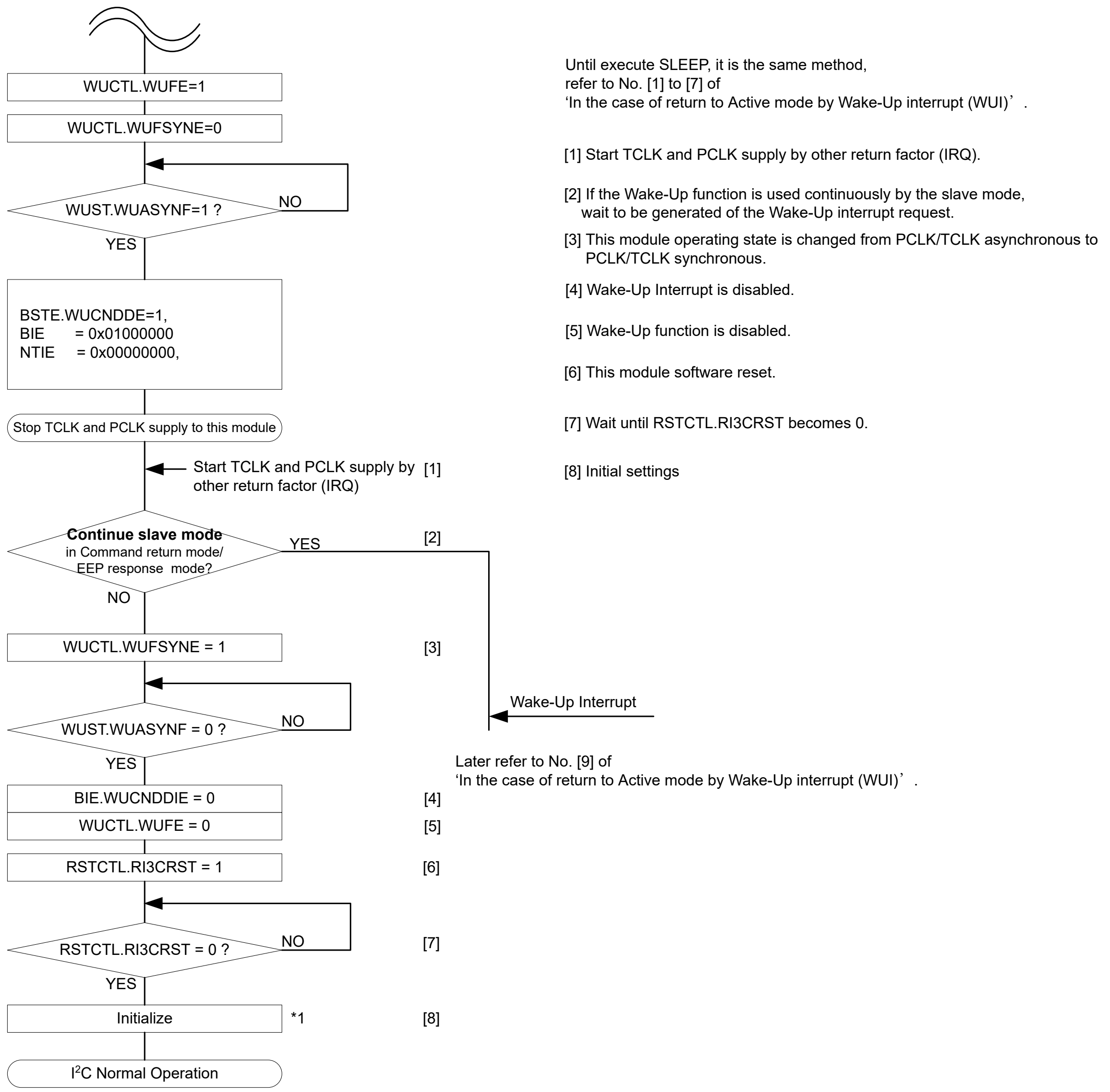
[13] Software reset.

[14] Wait until RSTCTL.RI3CRST becomes 0.

[15] Initial settings

Note 1. See section 40.3.3.1. Initial Setting Flow for details of initialization.

Figure 40.131 Use case of command recover mode and EEP response mode (wake-up recovery by a wake-up interrupt triggered by the match of the slave address)



Note 1. See section 40.3.3.1. Initial Setting Flow for details of initialization.

Figure 40.132 Use case of command recover mode and EEP response mode (wake-up recovery by other recovery causes (IRQ))

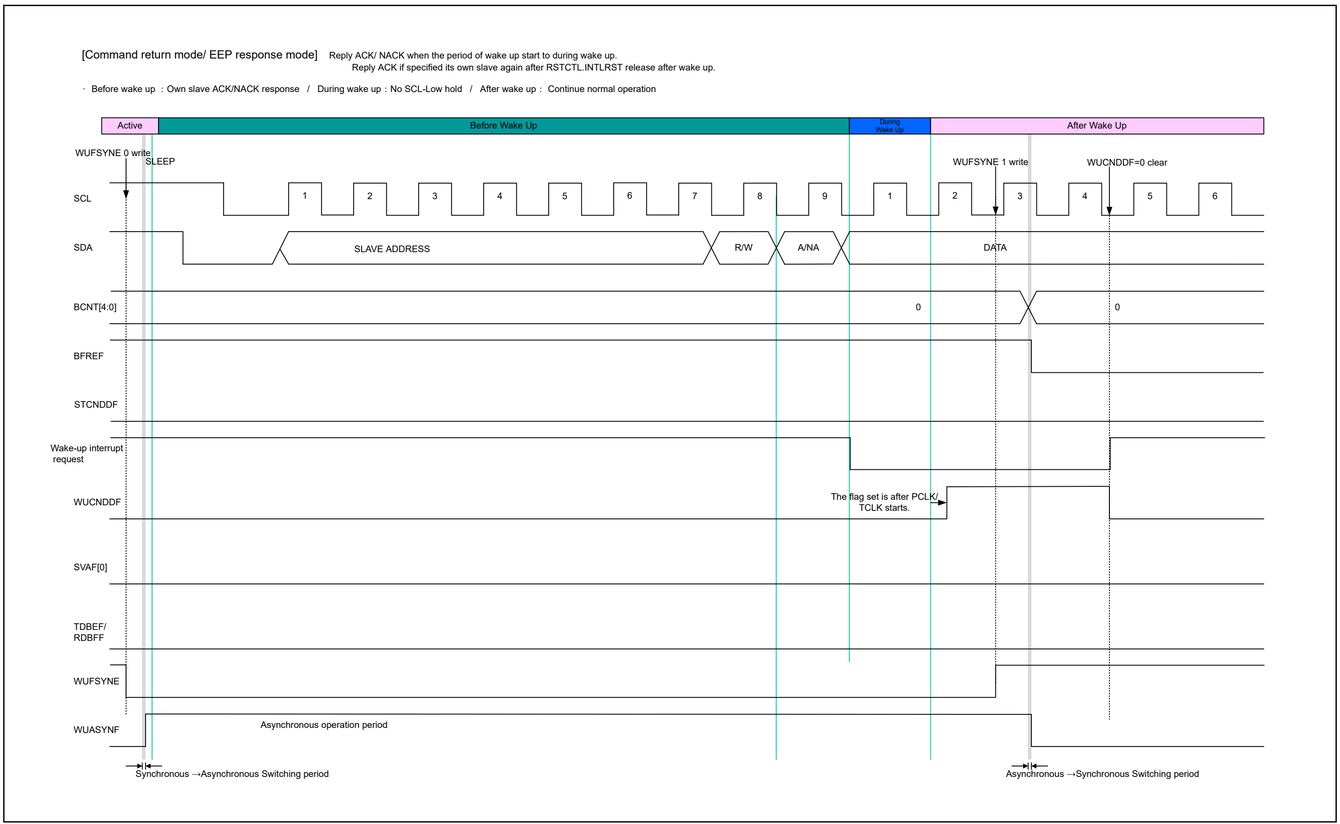


Figure 40.133 Timing of command recovery mode/EEP response mode

(4) Precautions on the use of the Wake-Up function

Precautions on the use of the Wake-up function is shown below.

- Do not change the registers in I3C except the WUCTL.WUFSYNE bit while the WUST.WUASYNF flag = 1 (while PCLK/TCLK asynchronous operation).
- Set WUCTL.WUFE = BSTE.WUCNDDE = BIE.WUCNDDIE = 1 and PRSST.CRMS = PRSST.TRMD = 0 (slave reception mode) before switching PCLK/TCLK asynchronous mode.
- Cannot select the device ID and the 10-bit slave address for wake-up interrupt factor. Set the DVIDE bit in SVCTL and SDADLS bit in SDATBASn (n = 0 to 2) to 0.
- Sets all bits in BIE (TENDIE, NACKDIE, SPCNDDIE, STCNDDIE, ALIE, TODIE) and TDBEIE0 and RDBFIE0 bits in NTIE to 0 (Interrupt disabled) before switching the asynchronous operation.
- Do not use the timeout function while the Wake-up function is enabled (WUCTL.WUFE = 1).
- Wake-up interrupt is generated while PCLK/TCLK asynchronous operation (when WUST.WUASYNF = 1). In case of detecting slave address matching, The case of detect slave address match in PCLK/TCLK synchronous mode (WUST.WUASYNF = 0), does not occur Wake-up interrupt, and BST.WUCNDDF flag will be not set also.
- If WUCTL.WUFSYNE bit to 0 write timing and START condition of detecting a conflict, I3C might start the next reception in PCLK/TCLK synchronous operation mode. In this case, WUST.WUASYNF flag becomes 1 (switch to PCLK/TCLK asynchronous mode) when data communication is finished and detected STOP condition and starts the Wake-up event detection.
- If you want to switch from PCLK/TCLK asynchronous operation to PCLK/TCLK synchronous operation without address match detection, it will switch in the STOP condition detection. When the WUCTL.WUFSYNE bit was set to 1 in a bus free state, it is continued PCLK/TCLK asynchronous operation (Reception operation: waiting communication frame). WUST.WUASYNF flag becomes to 0 when I3C detect the STOP condition of the next communication frame, and I3C switches to PCLK/TCLK synchronous operation.
- After writing 0 to WUFSYNE bit in WUCTL, do not change I3C operation mode setting register (BFCTL, SCSTRCTL, ACKCTL, INCTL, SVCTL, SDATBASn (n = 0 to 2)) until switched to the PCLK/TCLK asynchronous operation from

PCLK/TCLK synchronous operation (while WUST.WUASYNF flag = 1). If register value changes by the interrupt processing etc. in this period, I3C might malfunction without succeeding to the setting to the asynchronous operation.

- During PCLK/TCLK asynchronous operation (WUST.WUASYNF = 1), do not refer to each flag of SVST, BST, NTST, HTST register and BCST.BFREF flag.
- Do not set ACKCTL.ACKT = 1 in order to make an ACK response in the synchronization unit when Wake-up is performed by slave address match in Normal wake-up mode 2.

40.3.2.5.2 Wake Up Function

(1) I3C Master Wake-Up

Wake-up interrupt causes of I3C master are shown below.

- SDA low detection (IBI request from I3C slave)

The operation when transitioning to active mode (normal operation) by Wake-Up interrupt of SDA Low detection is shown below.

Before wake-up recovery:	SDA Low Drive is detected and the I3C_WU interrupt is asserted.
During wake-up recovery:	Keep I3C_SCL Line High.
After wake-up recovery:	Drive SCL Low and complete START condition. SCL is supplied on the I3C bus and IBI from I3C slave is received.

If transition to active mode (normal operation) due to other factors, disable the Wake-up function as necessary.

After confirming PRSTDBG.SDILV = 1, set WUCTL.WUFE = 0.

Do not use the timeout function while the Wake-up function is enabled (WUCTL.WUFE = 1).

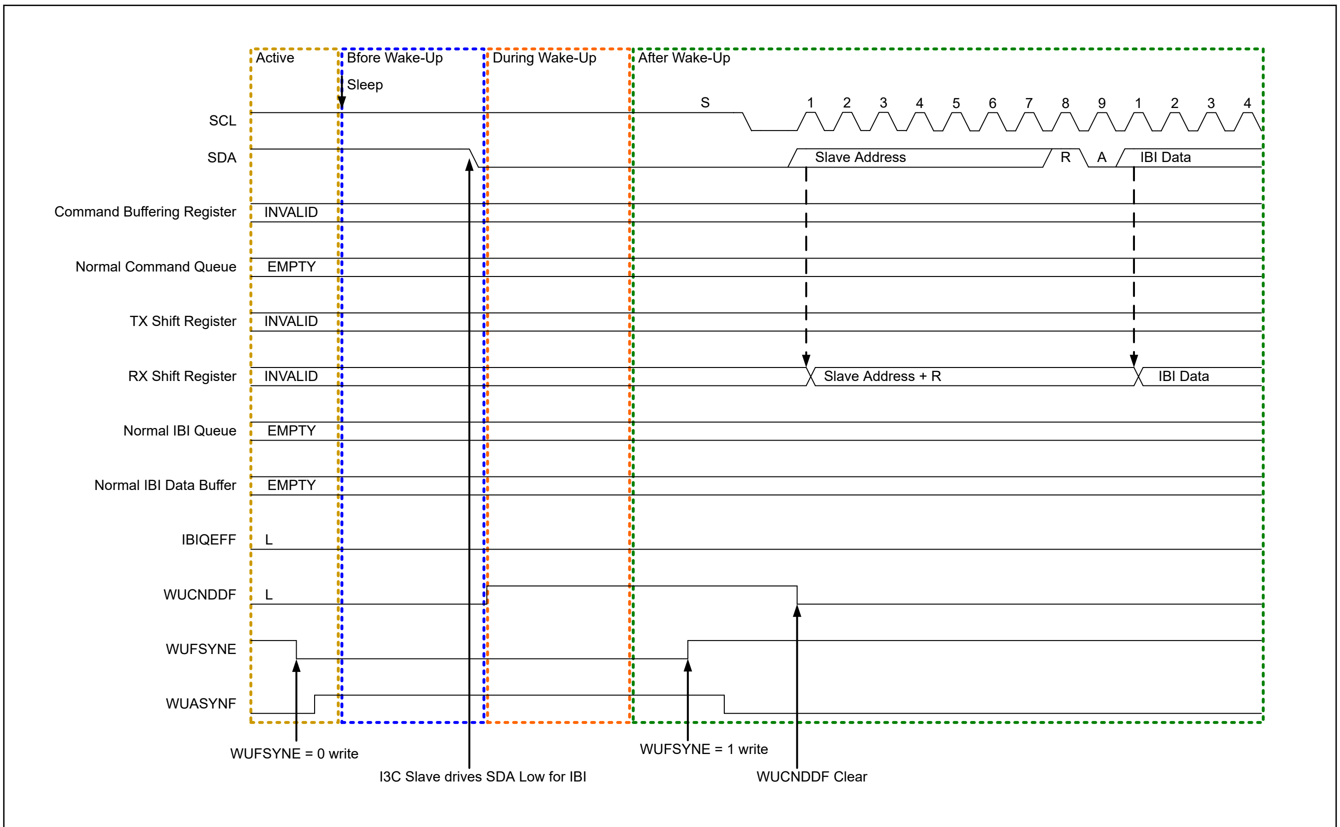


Figure 40.134 I3C master wake-up operation

(2) I3C Slave Wake-Up

Wake-up interrupt causes of I3C slave are shown below.

- Broadcast address (0x7E) and detects its own Slave address match

The operation when the Broadcast address (0x7E) and transition to active mode (normal operation) by Wake-up interrupt by detecting its own Slave address match is shown.

Before wake-up recovery:	1. If I3C detects BA (0x7E/W) following a START (or Repeated START) Condition, then I3C shall generate ACK (after 0x7E/W). 2. If I3C detects its own Dynamic address after a Repeated START condition following Step1, then I3C shall generate NACK (after its own Dynamic address) and then issues a I3C_WU interrupt.
During wake-up recovery:	I3C always generates NACK.
After wake-up recovery:	Normal operation continues.

If transition to active mode (normal operation) due to other factors, disable the Wake-up function as necessary. Do not use the timeout function while the Wake-up function is enabled (WUCTL.WUFE = 1).

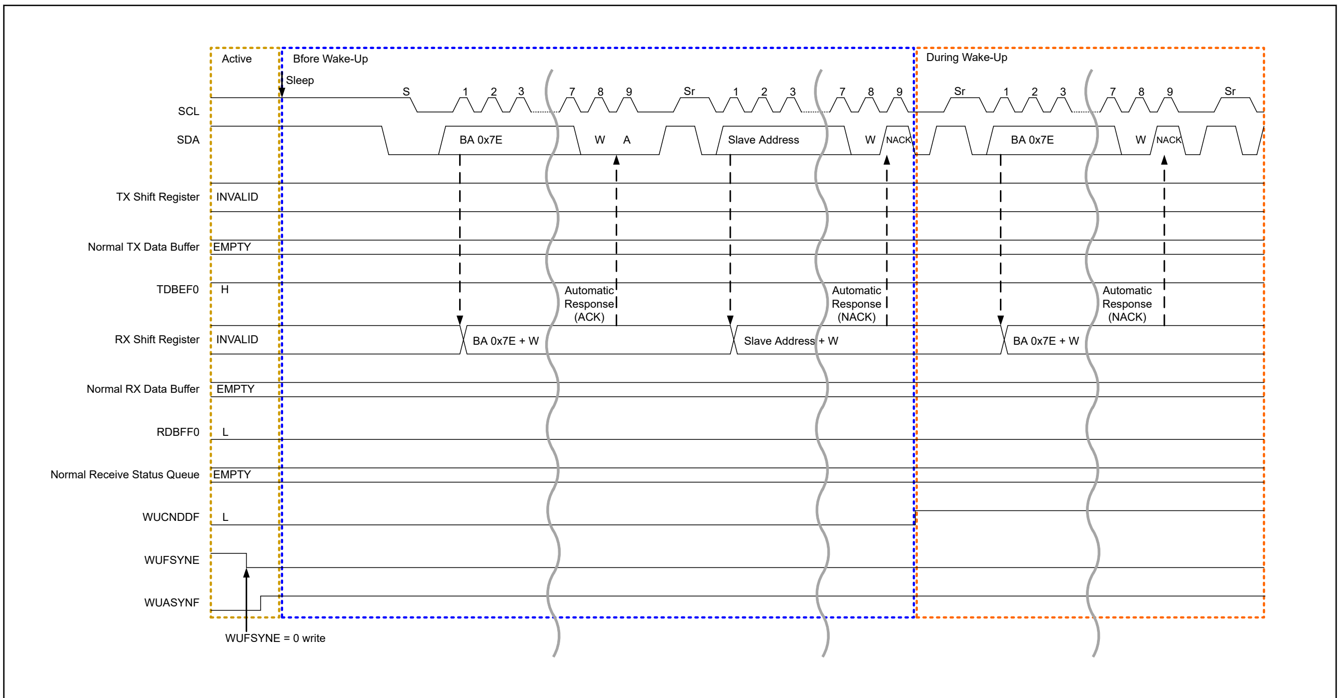


Figure 40.135 I3C slave wake-up operation (1/2)

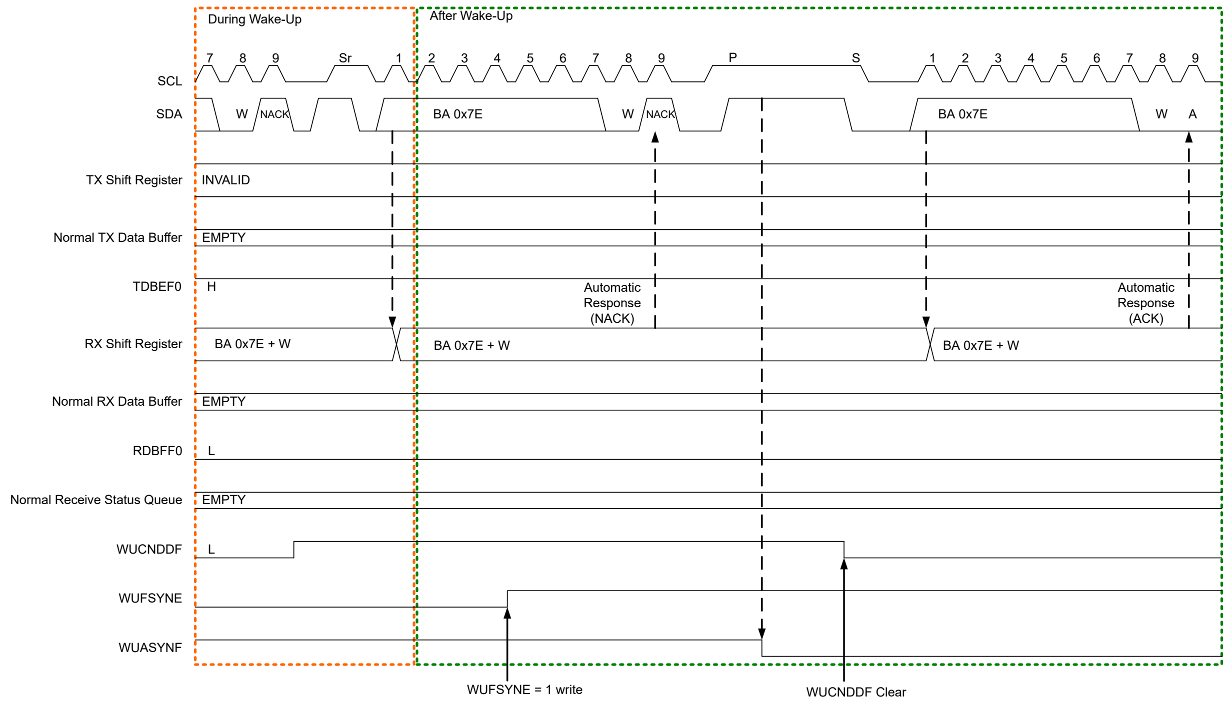


Figure 40.136 I3C slave wake-up operation (2/2)

40.3.2.6 Other

40.3.2.6.1 SCL Synchronization Circuit [I²C Mode]

This function is enabled while the PRTS.PRTMD bit is set to 1.

In generation of the SCL clock, I3C starts counting out the value for width at high level specified in STDBR.SBRHO[7:0] when it detects a rising edge on the I3C_SCL line and drives the I3C_SCL line low once counting of the width at high level is complete.

When I3C detects the falling edge of the I3C_SCL line, it starts counting out the width at low level period specified in STDBR.SBRLO[7:0], and then stops driving the I3C_SCL line (releases the line) once counting of the width at low level is complete. The SCL clock is thus generated.

If multiple master devices are connected to the I²C bus, a collision of SCL signals may arise due to contention with another master device. In such cases, the master devices have to synchronize their SCL signals. Since this synchronization of SCL signals must be bit by bit, I3C is equipped with a facility (the SCL synchronization circuit) to obtain bit-by-bit synchronization of the SCL clock signals by monitoring the I3C_SCL line while in master mode.

When I3C has detected a rising edge on the I3C_SCL line and thus started counting out the width at high level specified in STDBR.SBRHO[7:0], and the level on the I3C_SCL line falls because an SCL signal is being generated by another master device, I3C stops counting when it detects the falling edge, drives the level on the I3C_SCL line low, and starts counting out the width at low level specified in STDBR.SBRLO[7:0]. When I3C finishes counting out the width at low level, it stops driving the I3C_SCL line to the low level (releases the line). At this time, if the width at low level of the SCL clock signal from the other master device is longer than the width at low level set in this module, the width at low level of the SCL signal will be extended. Once the width at low level for the other master device has ended, the SCL signal rises because the I3C_SCL line has been released. When I3C finishes outputting the low-level period of the SCL clock, the I3C_SCL line is released and the SCL clock rises. That is, in cases of contention of SCL signals from more than one master, the width at high level of the SCL signal is synchronized with that of the clock having the narrower width, and the width at low level of the SCL signal is synchronized with that of the clock having the broader width. However, such synchronization of the SCL signal is only enabled when the SCSYNE bit in BFCTL is set to 1.

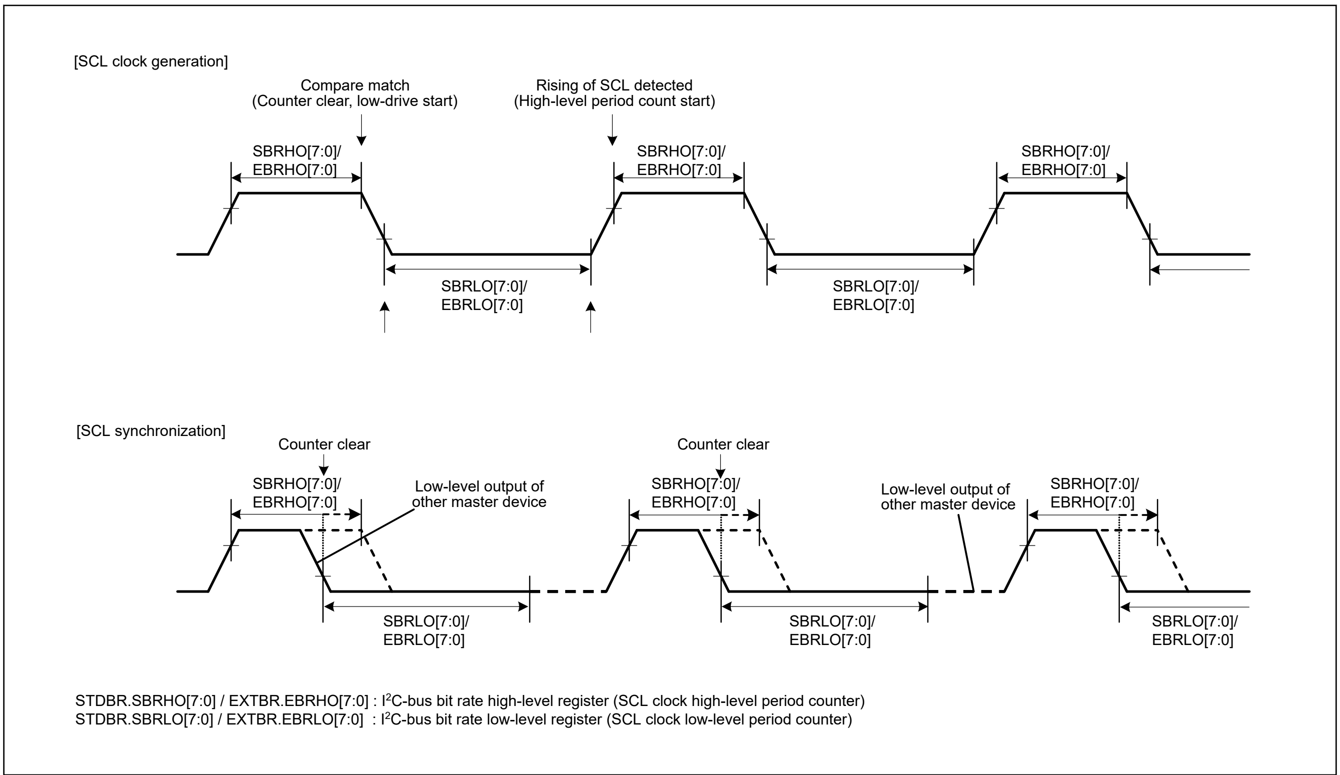


Figure 40.137 Generation and synchronization of the SCL signal

40.3.2.6.2 Facility for Delaying SDA Output [I²C Mode]

I3C module incorporates a facility for delaying output on the I3C_SDA line. The delay can be applied to all output (issuing of the START, Repeated START, and STOP conditions, data, and the ACK and NACK signals) on the I3C_SDA line.

With the SDA output delay facility, SDA output is delayed from detection of a falling edge of the SCL signal to ensure that the SDA signal is output within the interval over which the SCL clock is at the low level. Doing this leads to usage with the aim of preventing erroneous operation of communications devices, with the aim of satisfying the 300 ns (minimum) data-hold time requirement of the SMBus specification.

The output delay facility is enabled by setting the SDOD[2:0] bits in OUTCTL to any value other than 000b, and disabled by setting the same bits to 000b.

While the SDA output delay facility is enabled (while the SDOD[2:0] bits in OUTCTL are set to any value other than 000b), the SDODCS bit in OUTCTL selects the clock source for counting by the SDA output delay counter as the internal base clock (I3Cφ) for I3C module or as a clock signal derived by dividing the frequency of the internal base clock by two (I3Cφ/2). The counter counts the number of cycles set in the SDOD[2:0] bits in OUTCTL. After counting of the set number of cycles of delay is completed, I3C module places the required output (START, Repeated START, or STOP condition, data, or an ACK or NACK signal) on the I3C_SDA line.

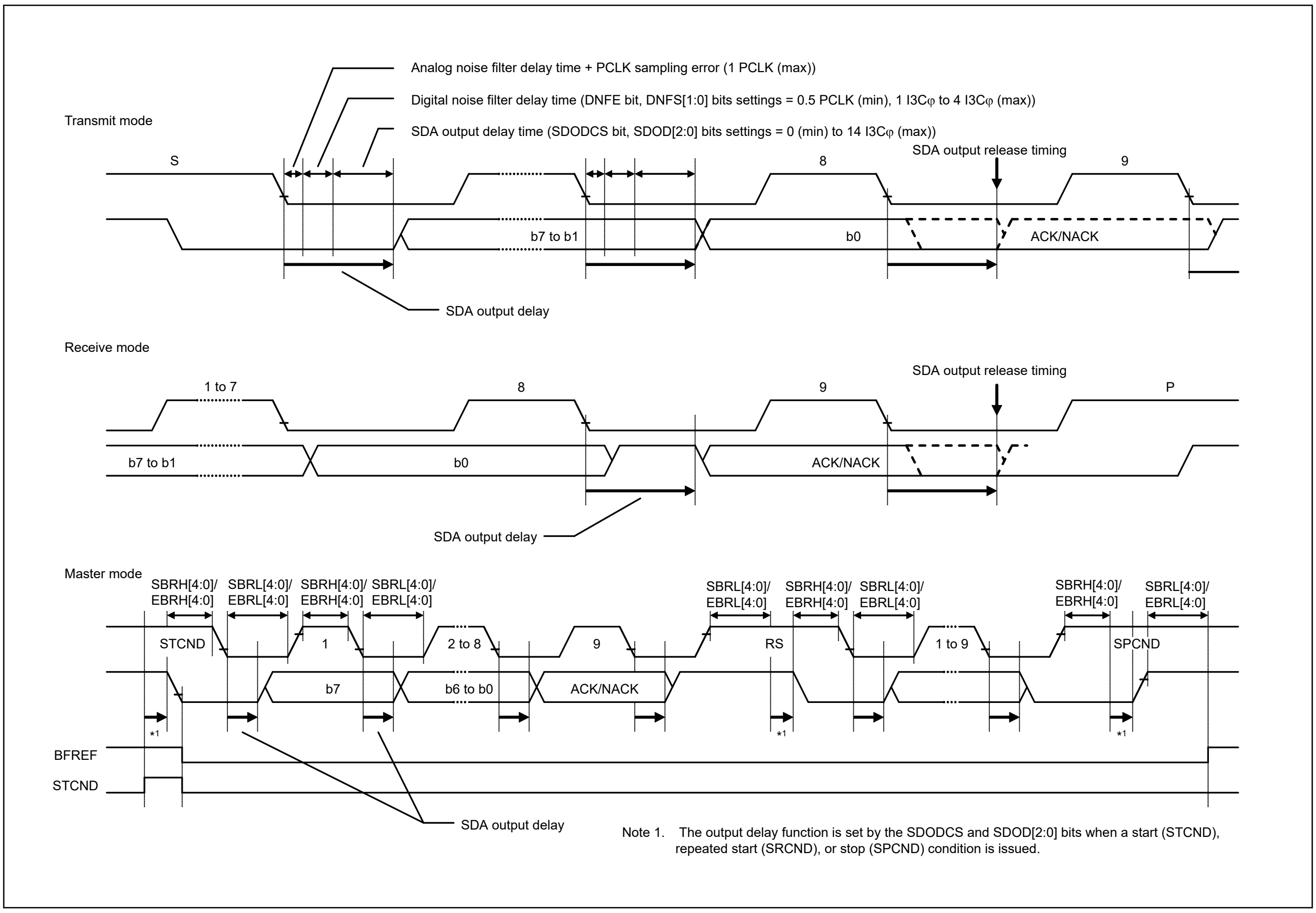


Figure 40.138 SDA output delay facility

40.3.2.6.3 Digital Noise-Filter Circuits [I²C Mode]

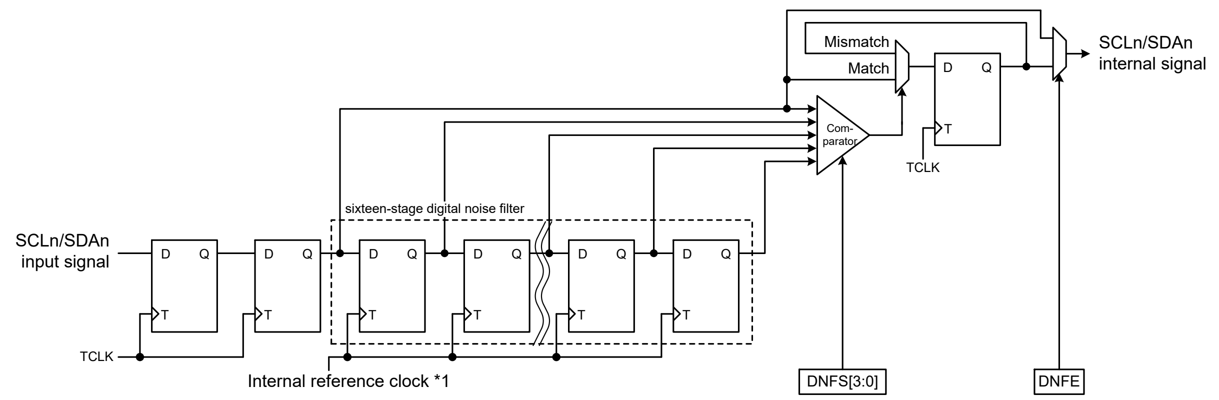
The states of the I3C_SCL and I3C_SDA pins are conveyed to the internal circuitry through digital noise-filter circuits. Figure 40.139 is a block diagram of the digital noise-filter circuit.

The on-chip digital noise-filter circuit of I3C consists of 16 flip-flop circuit stages connected in series and a match detection circuit. When HS mode is selected, only the first four flip-flop circuits stages are enabled.

The number of effective stages in the digital noise filter is selected by the INCTL.DNFS[3:0] bits. The selected number of effective stages determines the noise-filtering capability as a period from one to sixteen I3Cφ cycles.

The input signal to the I3C_SCL pin (or I3C_SDA pin) is sampled on rising edges of the I3Cφ signal. When the input signal level matches the output level of the number of effective flip-flop circuit stages as selected by the INCTL.DNFS[3:0] bits, the signal level is conveyed to the subsequent stage. If the signal levels do not match, the previous value is retained.

If the ratio between the frequency of the internal operating clock (TCLK) and the transfer rate is small (For example, data transfer at 400 kbps with TCLK = 4 MHz), the characteristics of the digital noise filter may lead to the elimination of needed signals as noise.



DNFE: Digital noise filter circuit enable bit
DNFS[3:0]: Noise filter stage select bits

Note 1. Internal reference clock = I3Cφ

Figure 40.139 Block diagram of digital noise filter circuit

40.3.3 Operation

40.3.3.1 Initial Setting Flow

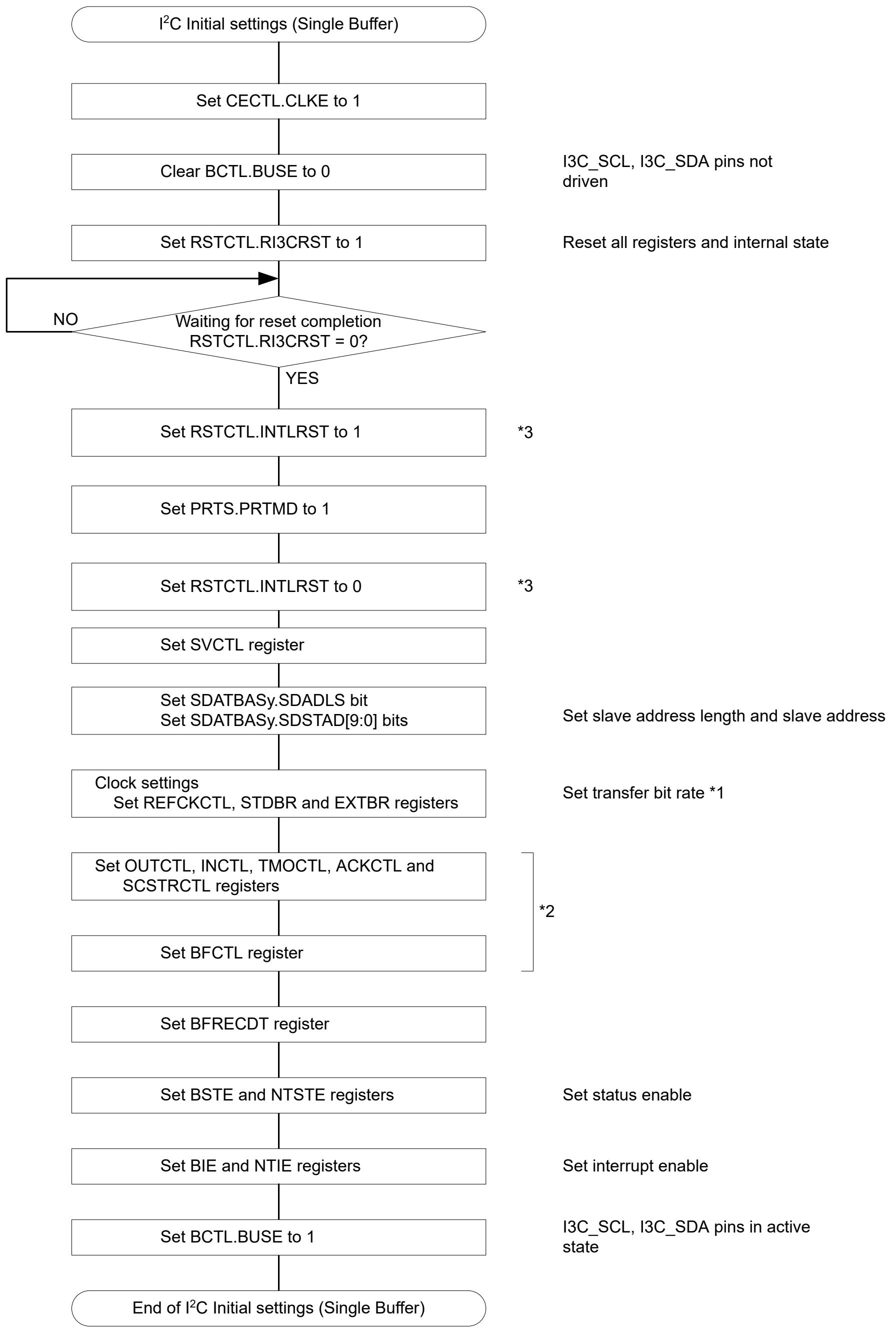
40.3.3.1.1 I²C Initial Setting Flow (Single Buffer Transfer)

Before starting data transmission and reception in I²C protocol mode, initialize I3C module.

First, set the CECTL.CLKE to 1, and set the BCTL.BUSE bit to 0 (I3C_SCL, I3C_SDA pins not driven).

Next, Set the RSTCTL.RI3CRST bit to 1. This initializes the registers and internal state of I3C module. Then, it waits for RI3CRST to become "0". For flags and registers to be initialized by this operation, see section 40.6. Reset Descriptions.

After that, set each register required for operation. For detail procedure, see Figure 40.140.



Note: y = 0 to 2
Note 1. When the I3C is used only in slave mode, set the STDBR register to a value longer than the data setup time.
Note 2. Set these registers as necessary.
Note 3. This step can be omitted if certain conditions are met. For details, see section 40.7.4. Notes on Rewriting PRTS.PRTMD Bit.

Figure 40.140 Example of I²C initialization flowchart (single buffer transfer)

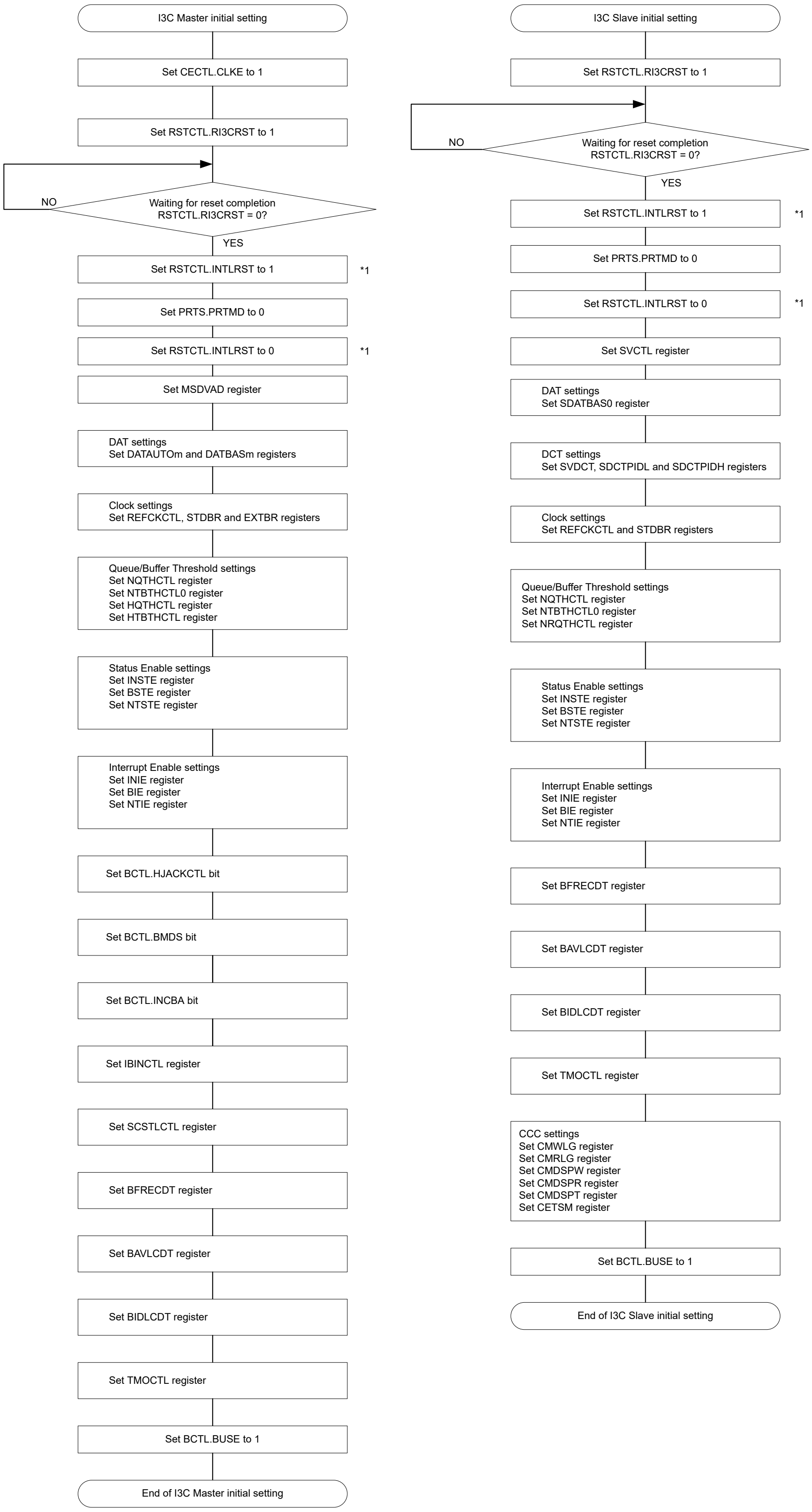
40.3.3.1.2 I3C Initial Setting Flow

Before starting data transmission and reception in I3C protocol mode, initialize I3C module.

First, set the CECTL.CLKE to 1.

Next, Set the RSTCTL.RI3CRST bit to 1. This initializes the registers and internal state of I3C module. Then, it waits for RI3CRST to become "0". For flags and registers to be initialized by this operation, see ssection 40.6. Reset Descriptions.

After that, set each register required for operation. For detail procedure on each of the master mode and slave mode, see Figure 40.141.



Note: y = 0 to 2

Note 1. This step can be omitted if certain conditions are met. For details, see section 40.7.4. Notes on Rewriting PRTS.PRTMD Bit.

Figure 40.141 Example of each initialization flowchart for I3C master mode and slave mode

40.3.3.2 I3C Communication Flow

Figure 40.142 illustrates how I3C communication is initiated:

- All I3C communication occurs within a frame. The frame begins with a START, followed by one or more transfers, and a STOP.
- For the HDR modes:
  - First the dedicated Broadcast I3C address (0x7E) is issued to all Slaves on the I3C bus.
  - Then one of the EnterHDR CCCs is issued, indicating that the Master is entering an HDR mode. Each HDR mode has its own EnterHDR CCC.
  - This is followed by one or more HDR transfers.
  - HDR mode is ended by using the HDR exit pattern protocol.

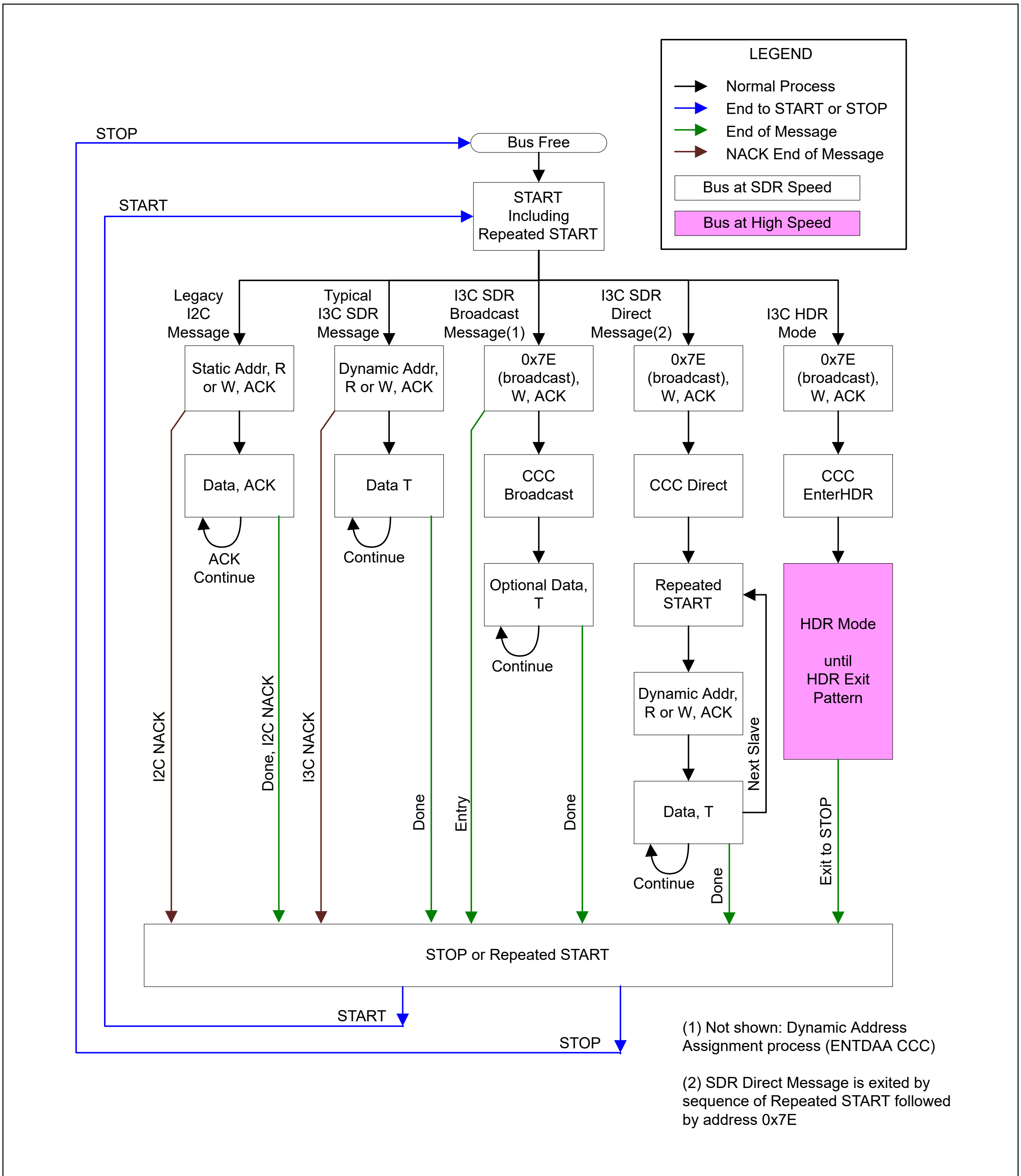


Figure 40.142 I3C communication flow

I3C is based on a frame encapsulation approach. A frame includes a data payload. The transfer protocol for the data payload is either SDR or HDR. Frames are bordered by I^2C-like bus management.

The I3C frame always includes at least the START, the Header, the Data, and the STOP. The Header following a START allows for Bus Arbitration. The Master uses the Header to address Slave device (s). Slave devices (s) may use the Header Arbitration for multiple purposes: for In-Band Interrupt, and for Secondary Master functionality.

Common Command Codes (CCCs) are used to enter the High Data Rate (HDR) modes. It is important to understand that I3C bus activity for the HDR Message does not follow the Legacy I^2C format.

I3C allows only one Master to have control of the I3C bus at a time. Mechanisms for handoff of the Master role from one device to another device are provided.

40.3.3.3 Master Mode Communication Flow

40.3.3.3.1 I2C Master Transmission Flow (Single Buffer Transfer)

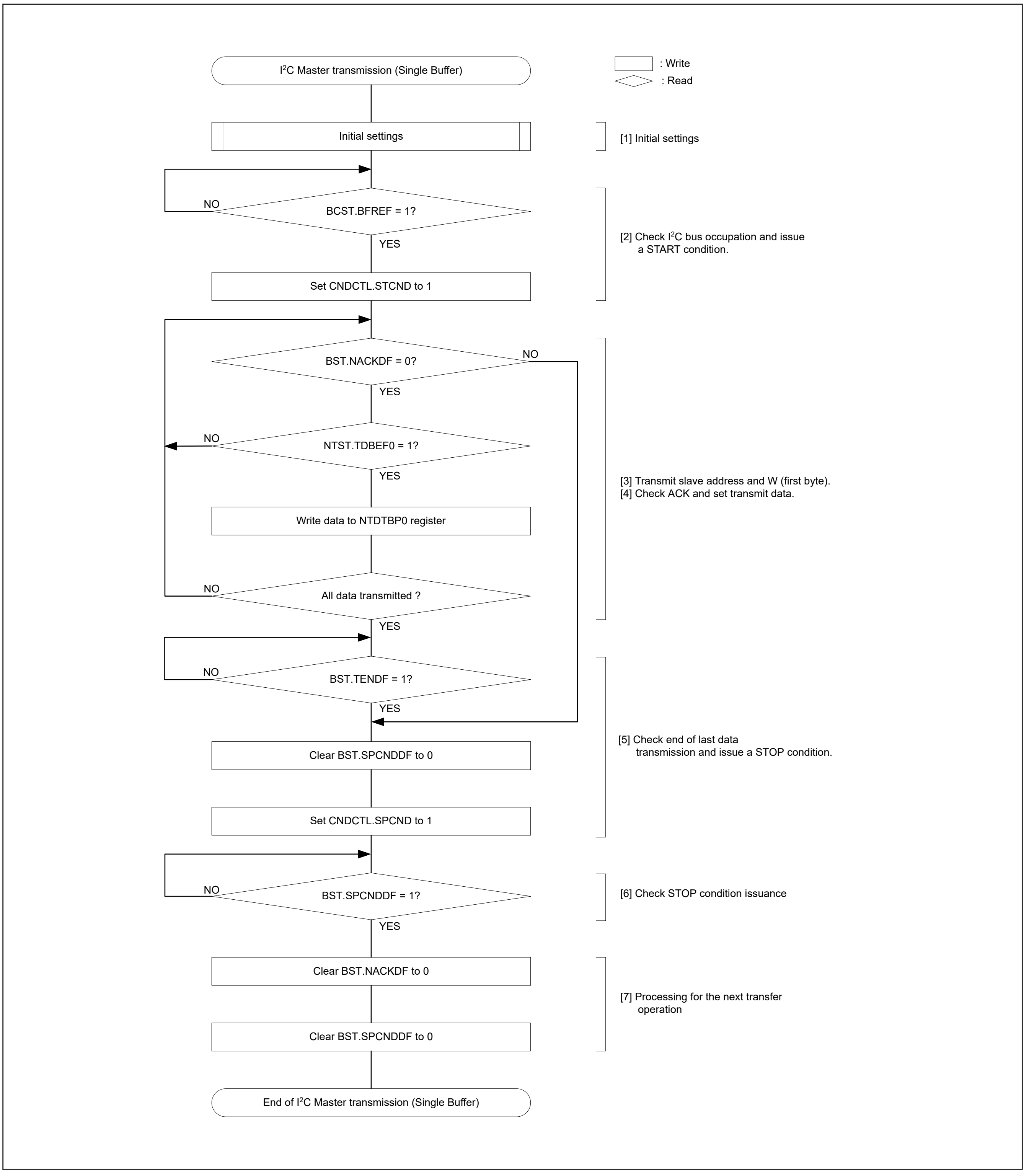


Figure 40.143 Example of I2C master transmission flowchart (single buffer transfer)

40.3.3.3.2 I²C Master Reception Flow (Single Buffer Transfer)

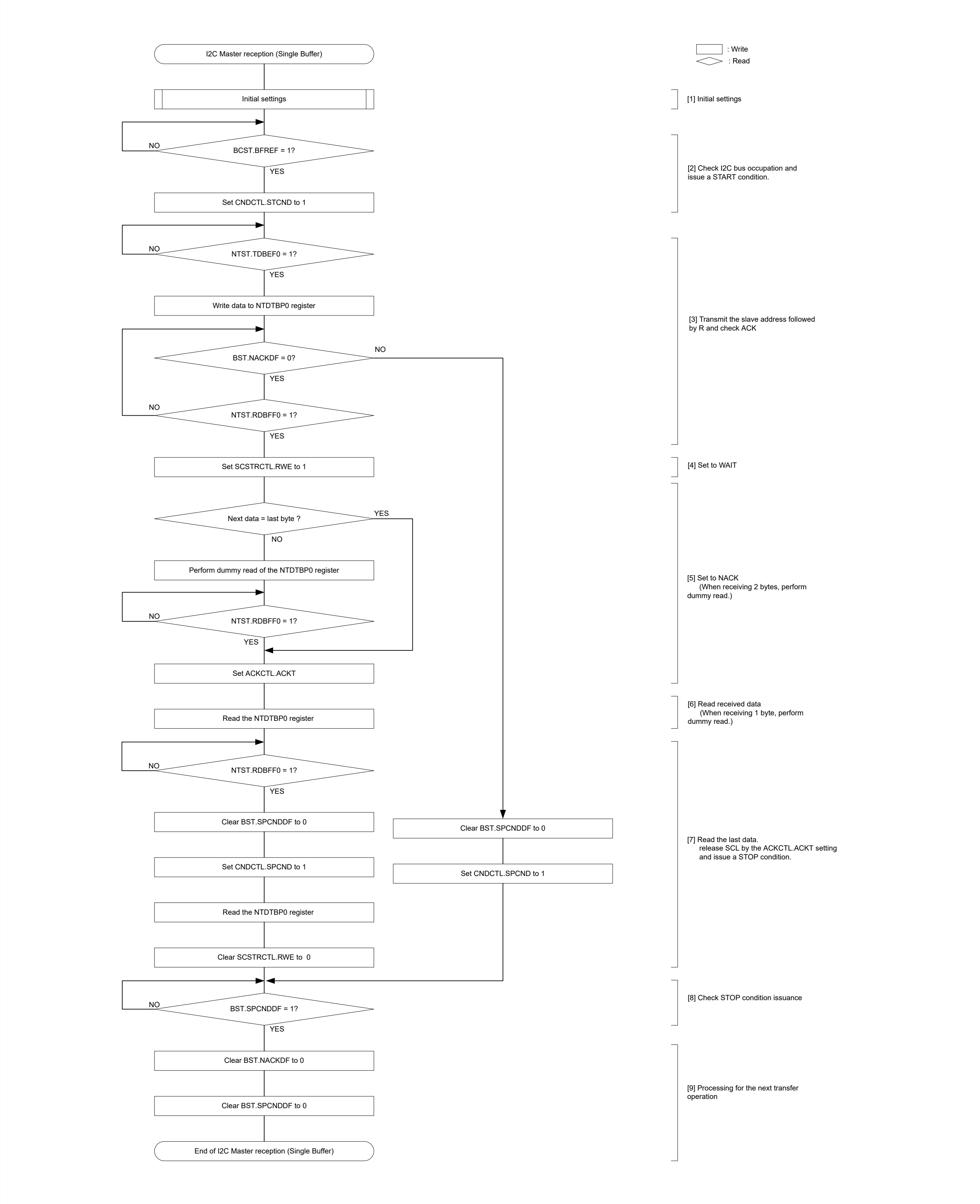


Figure 40.144 Example of I²C master reception flowchart (7-bit address format, 1 or 2 bytes)

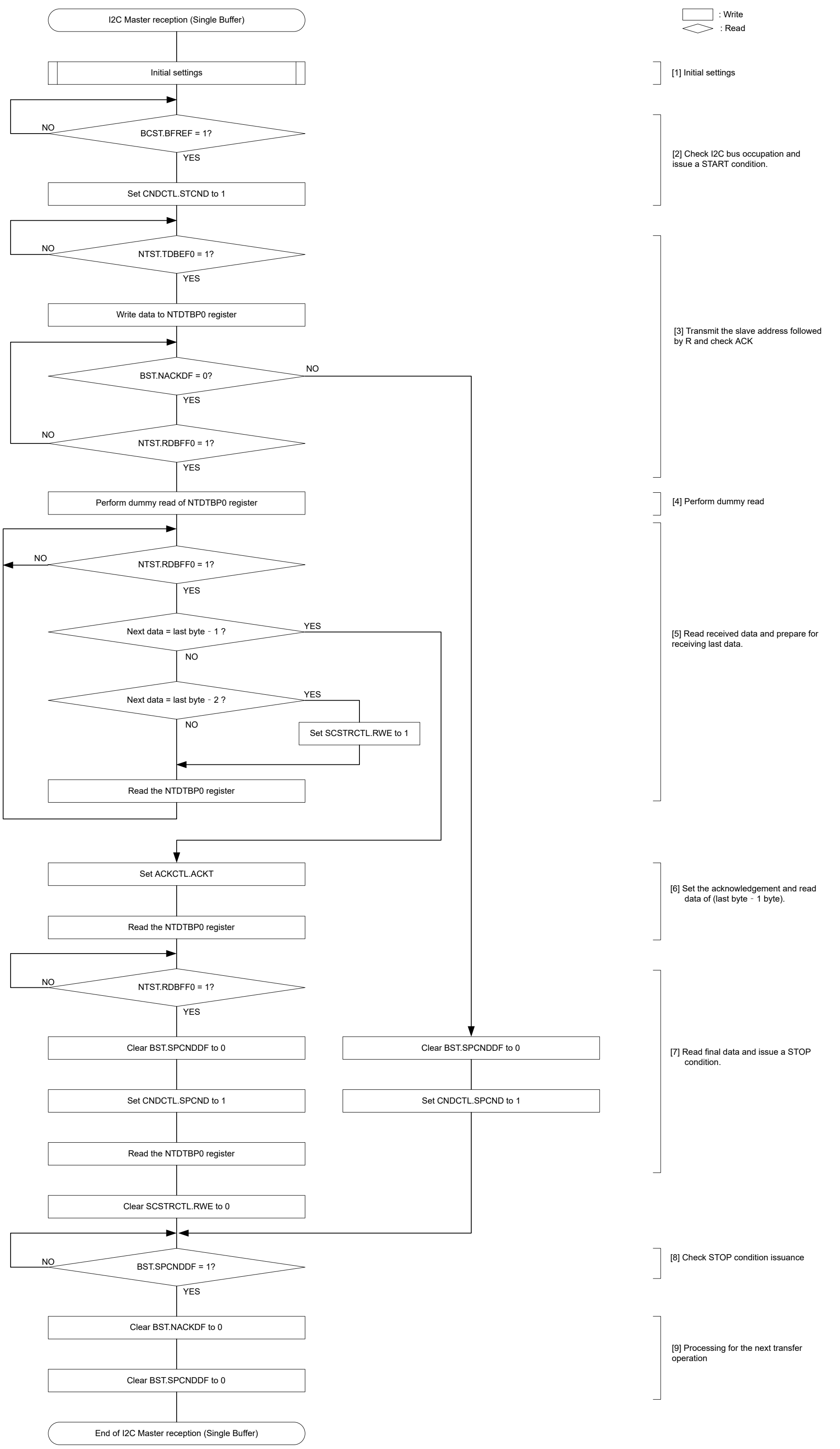
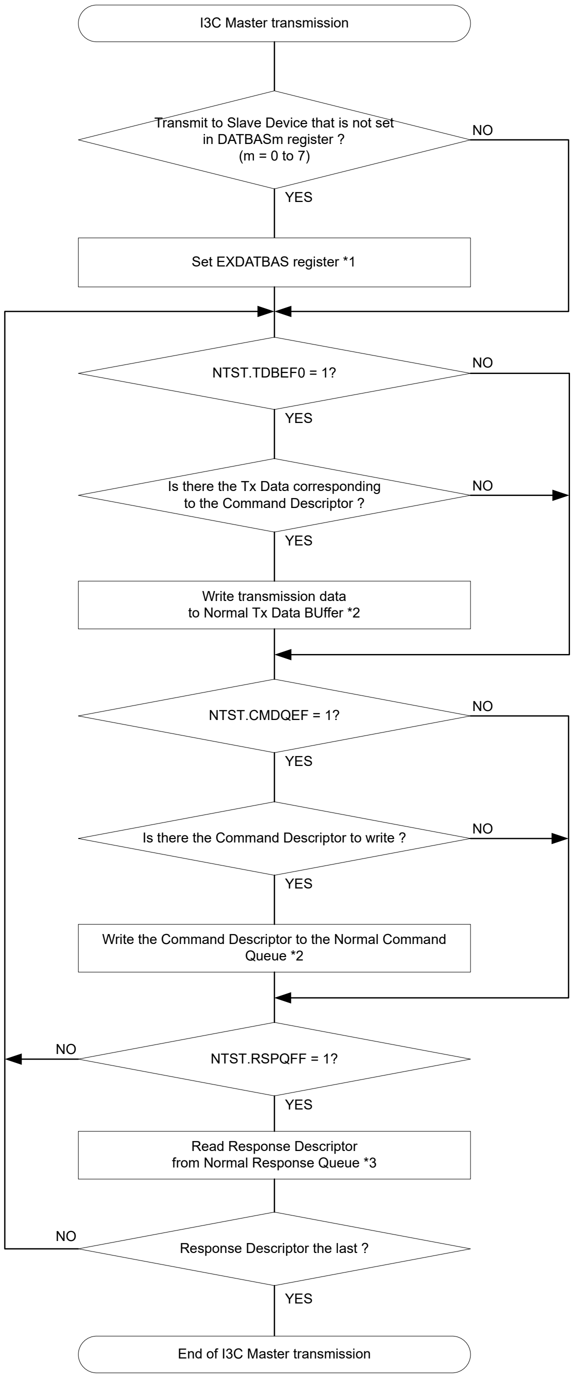


Figure 40.145 Example of I²C master reception flowchart (7-bit address format, 3 bytes or more)

40.3.3.3.3 I3C Master Transmission Flow (Normal FIFO Buffer Transfer)

Master transmission flow in I3C normal FIFO buffer transfer is common to Legacy I^2C and SDR (Private Transfer, Broadcast CCC, Direct CCC), HDR-DDR, and HDR-TSP/TSL.

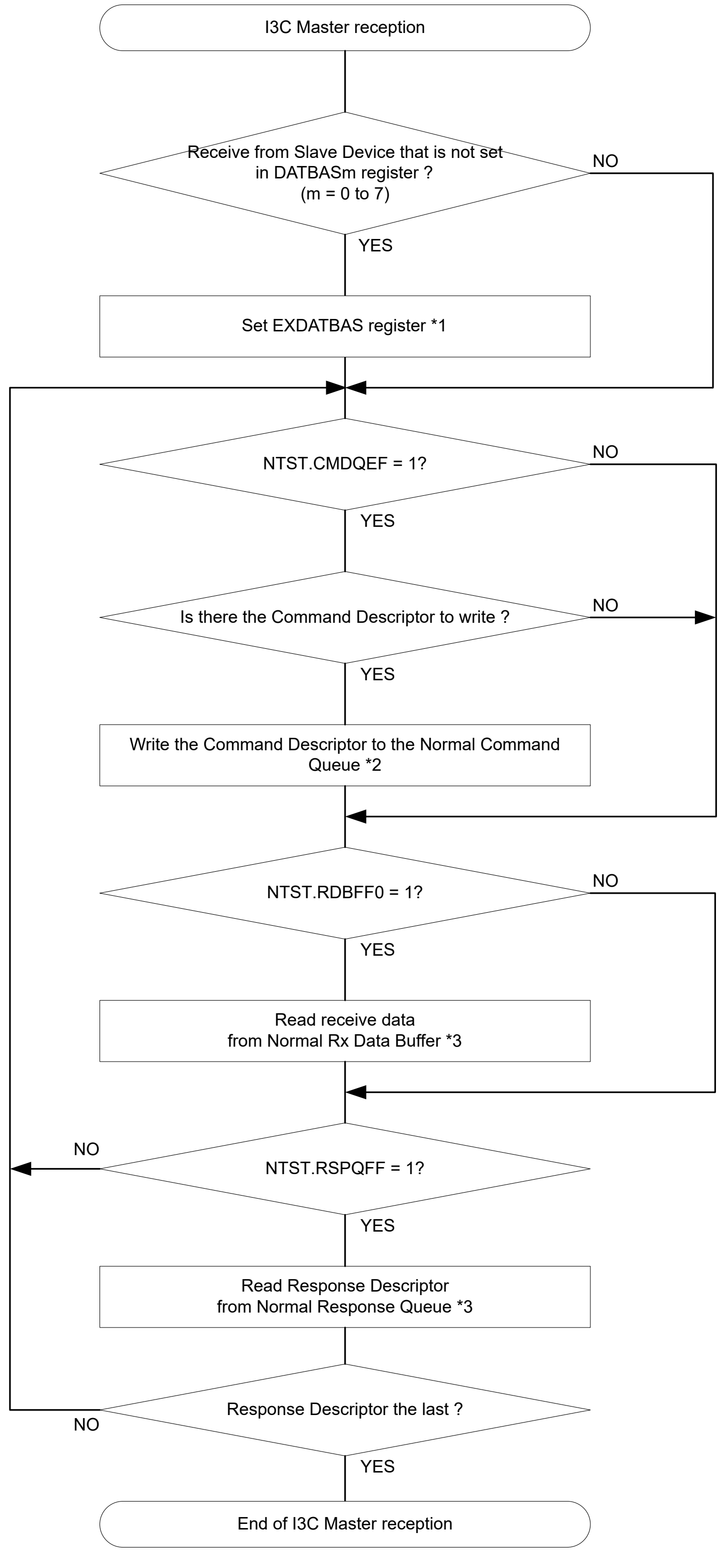


Note 1. Changing the EXDATBAS register is prohibited until all response descriptors of command for the Slave device set in the EXDATBAS register are stored.
Note 2. In the case the register is write by a way other than DMAC/DTC, clear corresponding flags after write access.
Note 3. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 40.146 Example of I3C master transmission flowchart (normal FIFO buffer transfer)

40.3.3.3.4 I3C Master Reception Flow (Normal FIFO Buffer Transfer)

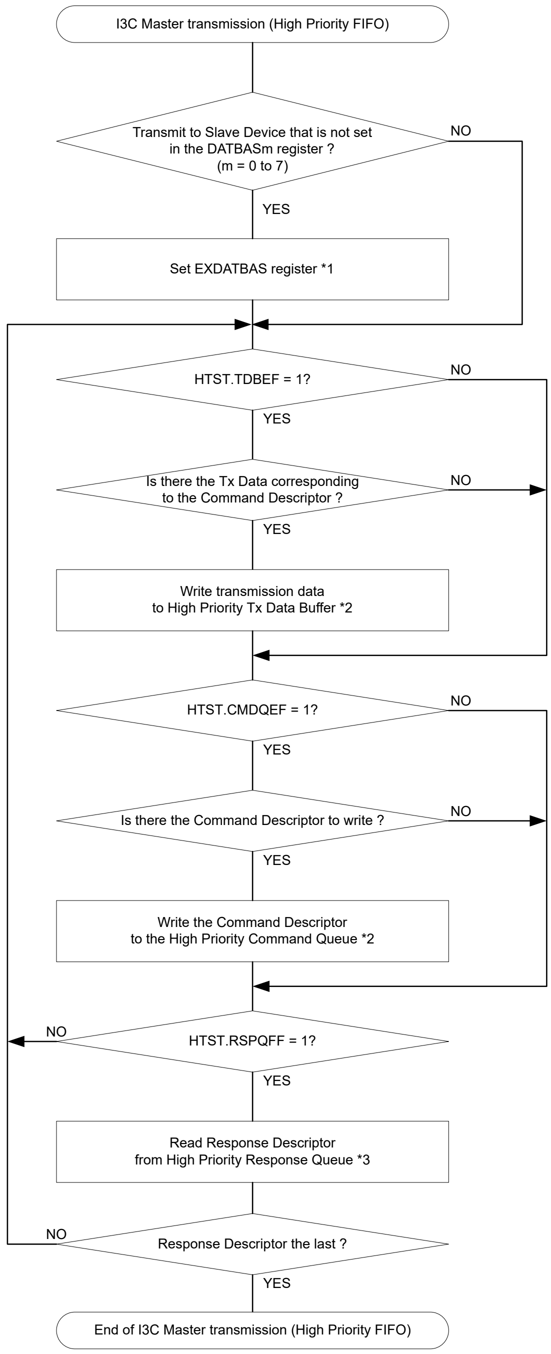
Master reception flow in I3C normal FIFO buffer transfer is common to Legacy I^2C and SDR (Private Transfer, Broadcast CCC, Direct CCC), HDR-DDR, and HDR-TSP/TSL.



Note 1. Changing the EXDATBAS register is prohibited until all response descriptors of command for the Slave device set in the EXDATBAS register are stored.
Note 2. In the case the register is write by a way other than DMAC/DTC, clear corresponding flags after write access.
Note 3. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 40.147 Example of I3C master reception flowchart (normal FIFO buffer transfer)

40.3.3.3.5 I3C Master Transmission Flow (High Priority FIFO Buffer Transfer)



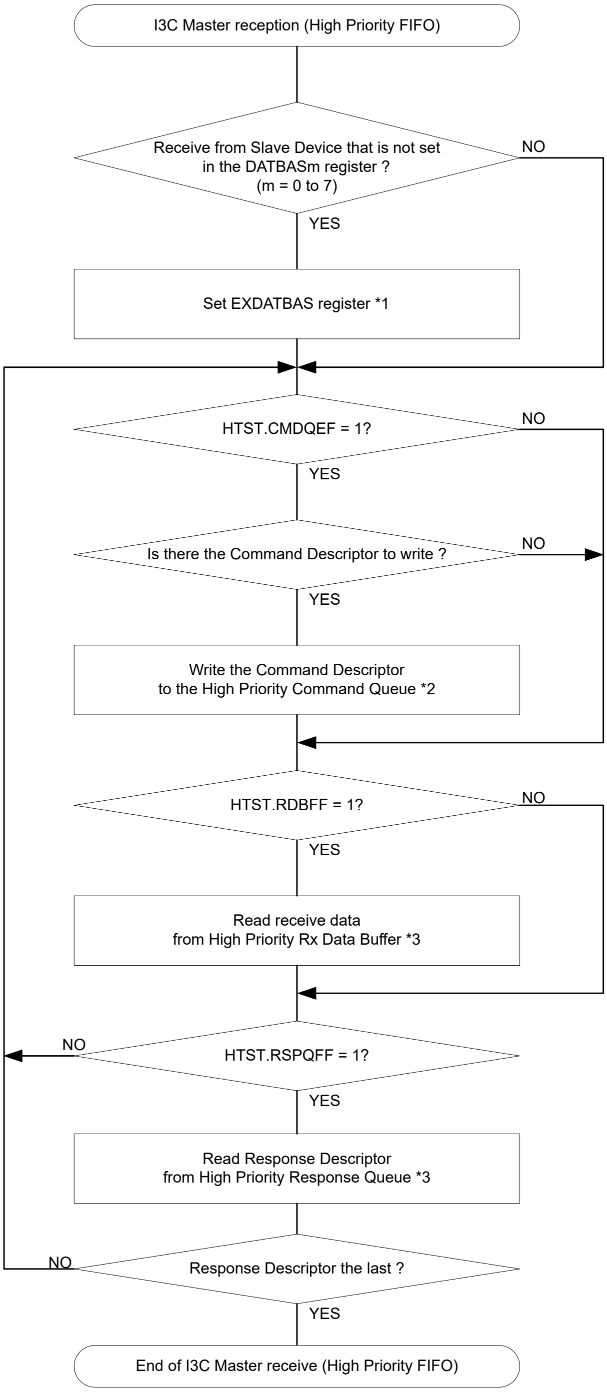
Note 1. Changing the EXDATBAS register is prohibited until all response descriptors of command for the Slave device set in the EXDATBAS register are stored.

Note 2. In the case the register is write by a way other than DMAC/DTC, clear corresponding flags after write access.

Note 3. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 40.148 Example of I3C master transmission flowchart (high priority FIFO buffer transfer)

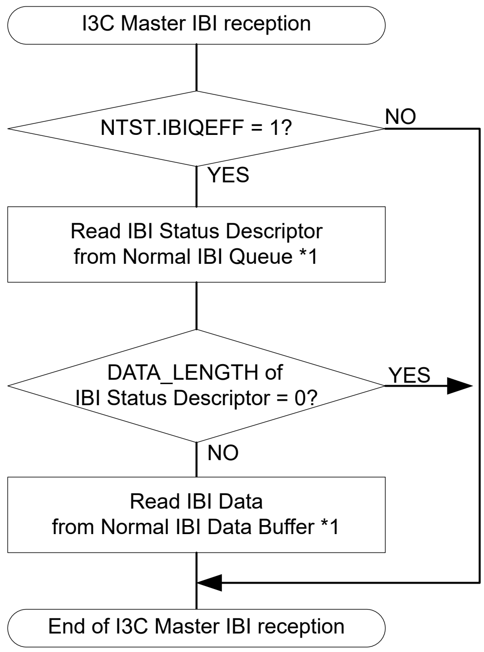
40.3.3.3.6 I3C Master Reception Flow (High Priority FIFO Buffer Transfer)



Note 1. Changing the EXDATBAS register is prohibited until all response descriptors of command for the Slave device set in the EXDATBAS register are stored.
Note 2. In the case the register is write by a way other than DMAC/DTC, clear corresponding flags after write access.
Note 3. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 40.149 Example of I3C master reception flowchart (high priority FIFO buffer transfer)

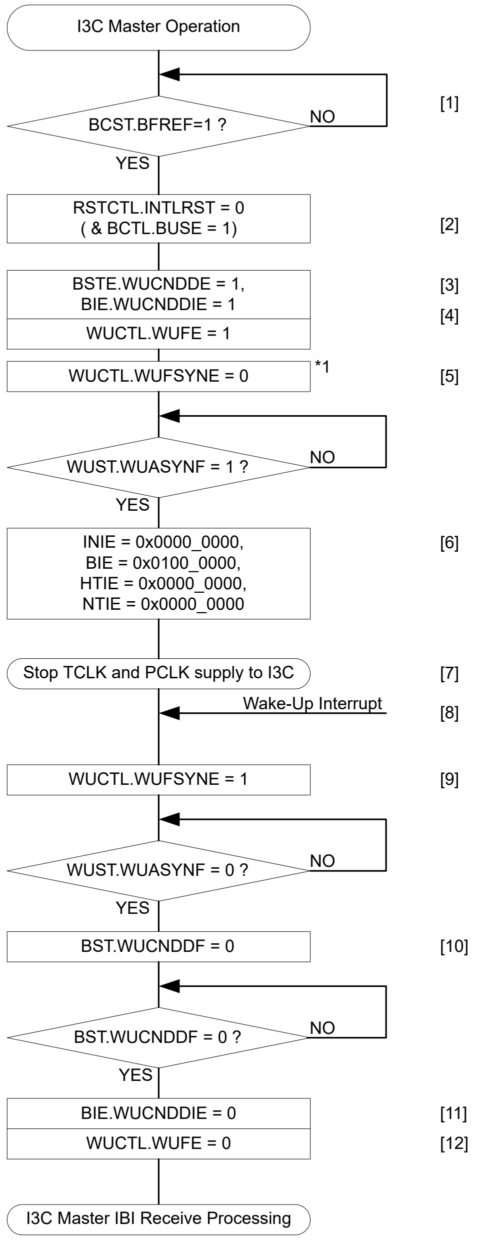
40.3.3.3.7 I3C Master IBI Reception Flow



Note 1. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 40.150 Example of I3C master IBI reception flowchart

40.3.3.3.8 I3C Master Wake-Up Flow



[1] Wait I3C bus free and I3C stay in the standby state.

[2] Negate internal reset if assert the internal reset.

[3] Wake-Up Interrupt is enabled.

[4] Wake-Up function is enabled.

[5] I3C of operating state is changed from PCLK/TCLK synchronous to PCLK/TCLK asynchronous.

[6] Interrupt requests are disabled except WUI.

[7] Stop TCLK supply to I3C, then stop PCLK supply to I3C.
(I3C will continue to receive)

[8] Start TCLK and PCLK supply to I3C by Wake-Up interrupt, then start PCLK supply to I3C.

[9] I3C of operating state is changed from PCLK/TCLK asynchronous to PCLK/TCLK synchronous.

[10] Write 0 to BST.WUCNDDF.
(Read and check whether clearing has been completed befor return from interrupt handling.)

[11] Wake-Up interrupt is disabled.

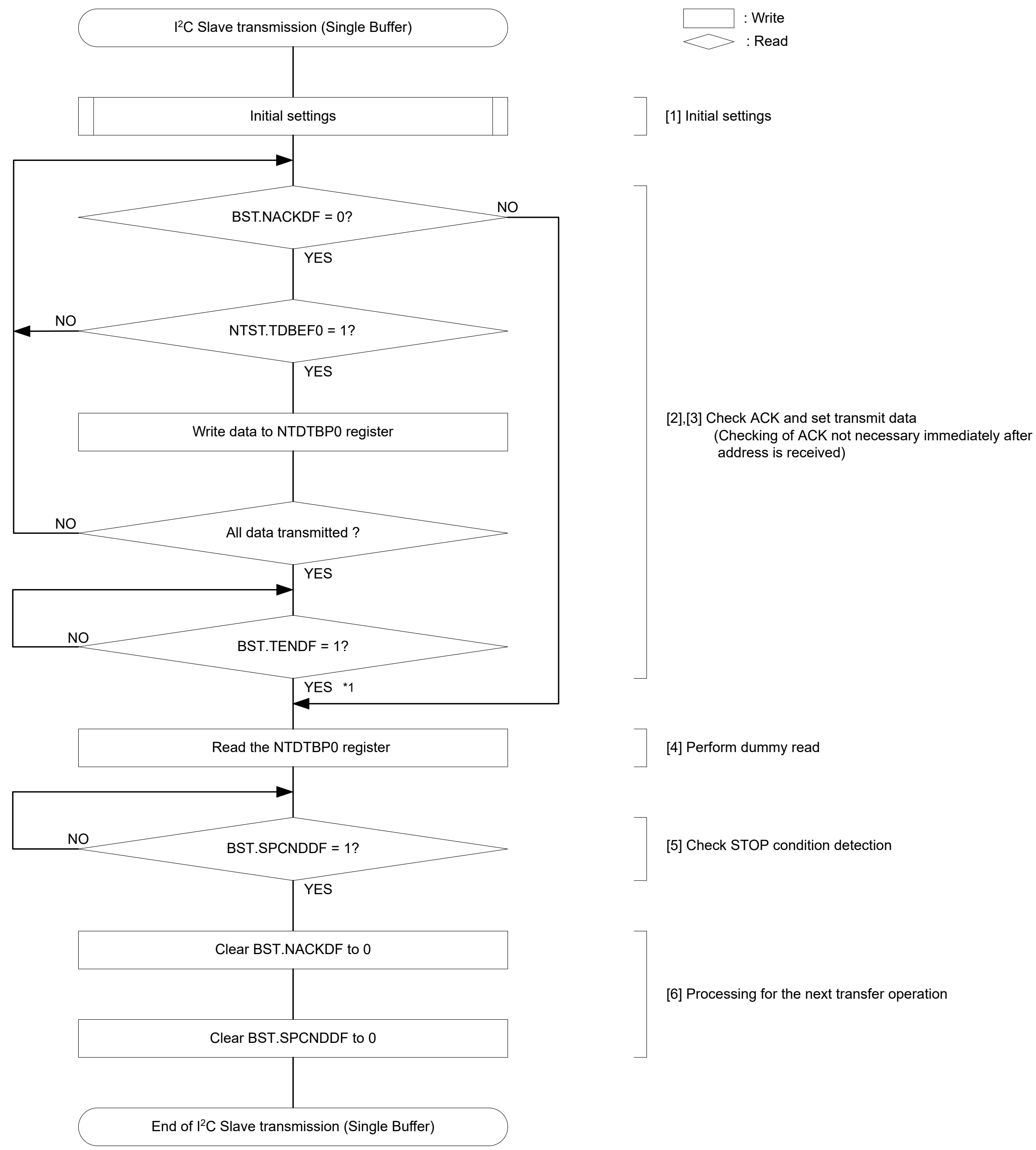
[12] Wake-Up function is disabled.

Note 1. After switching the I3C_SDA pin to the general I/O port (Input) in the I/O port function setting, set WUCTL.WUFSYNE to 1. Then wait until WUST.WUASYNF = 1, and switch the I3C_SDA pin to the peripheral function in the I/O port function setting.

Figure 40.151 Use case of I3C master wake-up

40.3.3.4 Slave Mode Communication Flow

40.3.3.4.1 I2C Slave Transmission Flow (Single Buffer Transfer)



Note 1. Normally, Master responds to the last byte with NACK, so when BST.TENDF=1, BST.NACKDF will also 1.
If BST.NACKDF is 0 after BST.TENDF=1, the master may hang-up, so reset I3C with RSTCTL.INTLRST.

Figure 40.152 Example of I2C slave transmission flowchart (single buffer transfer)

40.3.3.4.2 I²C Slave Reception Flow (Single Buffer Transfer)

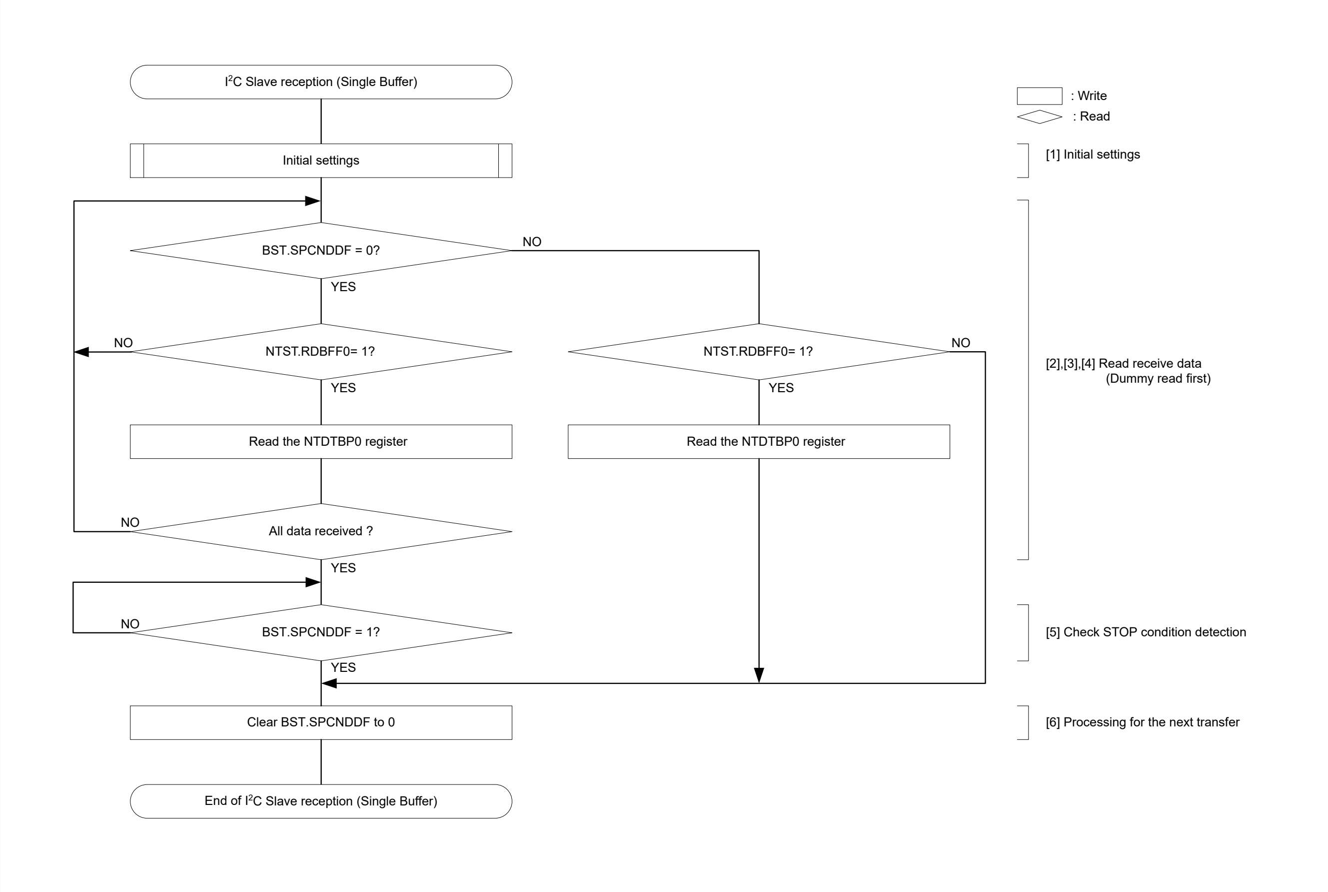
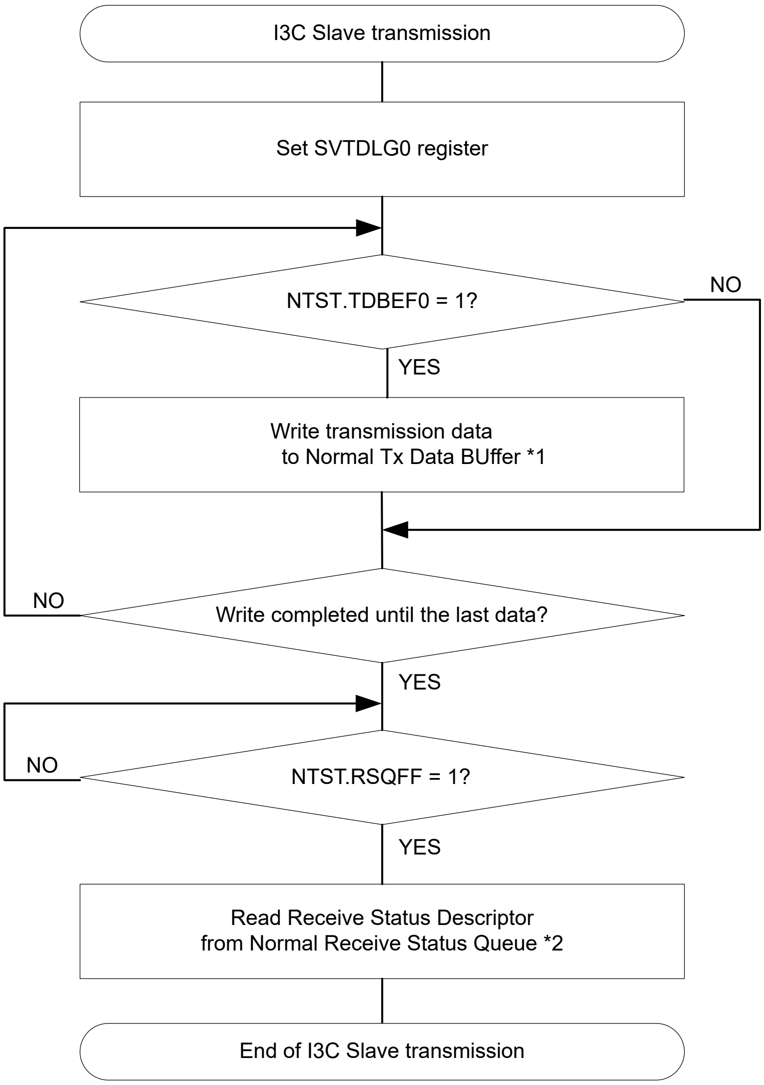


Figure 40.153 Example of I²C slave reception flowchart (single buffer transfer)

40.3.3.4.3 I3C Slave Transmission Flow (Normal FIFO Buffer Transfer)

Slave Transmission Flow in I3C normal FIFO buffer transfer is common to Legacy I²C, SDR (Private Transfer, Broadcast CCC, Direct CCC), HDR-DDR, and HDR-TSP/TSL.

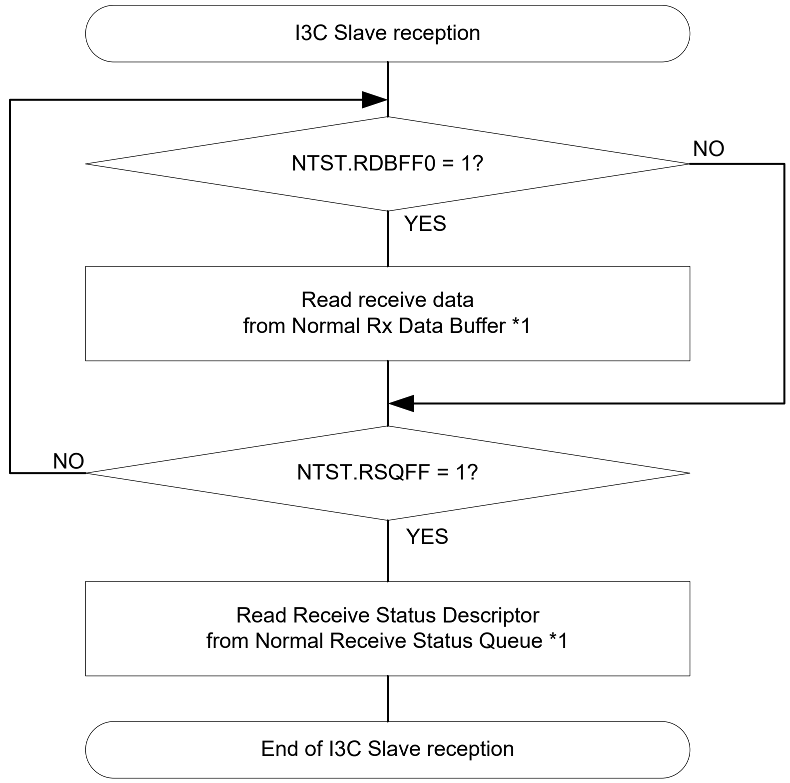


Note 1. In the case the register is write by a way other than DMAC/DTC, clear corresponding flags after write access.
Note 2. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 40.154 Example of I3C slave transmission flowchart (normal FIFO buffer transfer)

40.3.3.4.4 I3C Slave Reception Flow (Normal FIFO Buffer Transfer)

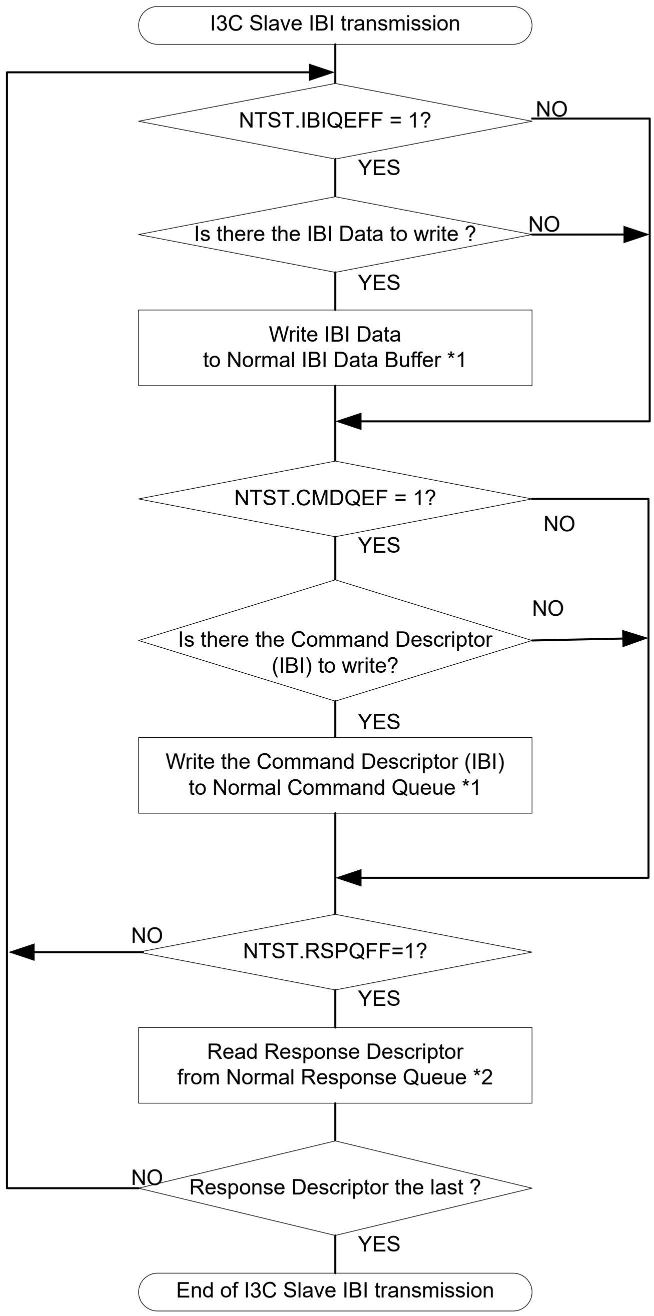
Slave Reception Flow in I3C normal FIFO buffer transfer is common to Legacy I^2C, SDR (Private Transfer, Broadcast CCC, Direct CCC), HDR-DDR, and HDR-TSP/TSL.



Note 1. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 40.155 Example of I3C slave reception flowchart (normal FIFO buffer transfer)

40.3.3.4.5 I3C Slave IBI Transmission Flow



Note 1. In the case the register is write by a way other than DMAC/DTC, clear corresponding flags after write access.
Note 2. In the case the register is read by a way other than DMAC/DTC, clear corresponding flags after read access.

Figure 40.156 Example of I3C slave IBI transmission flowchart

40.3.3.4.6 I3C Slave Wake-Up Flow

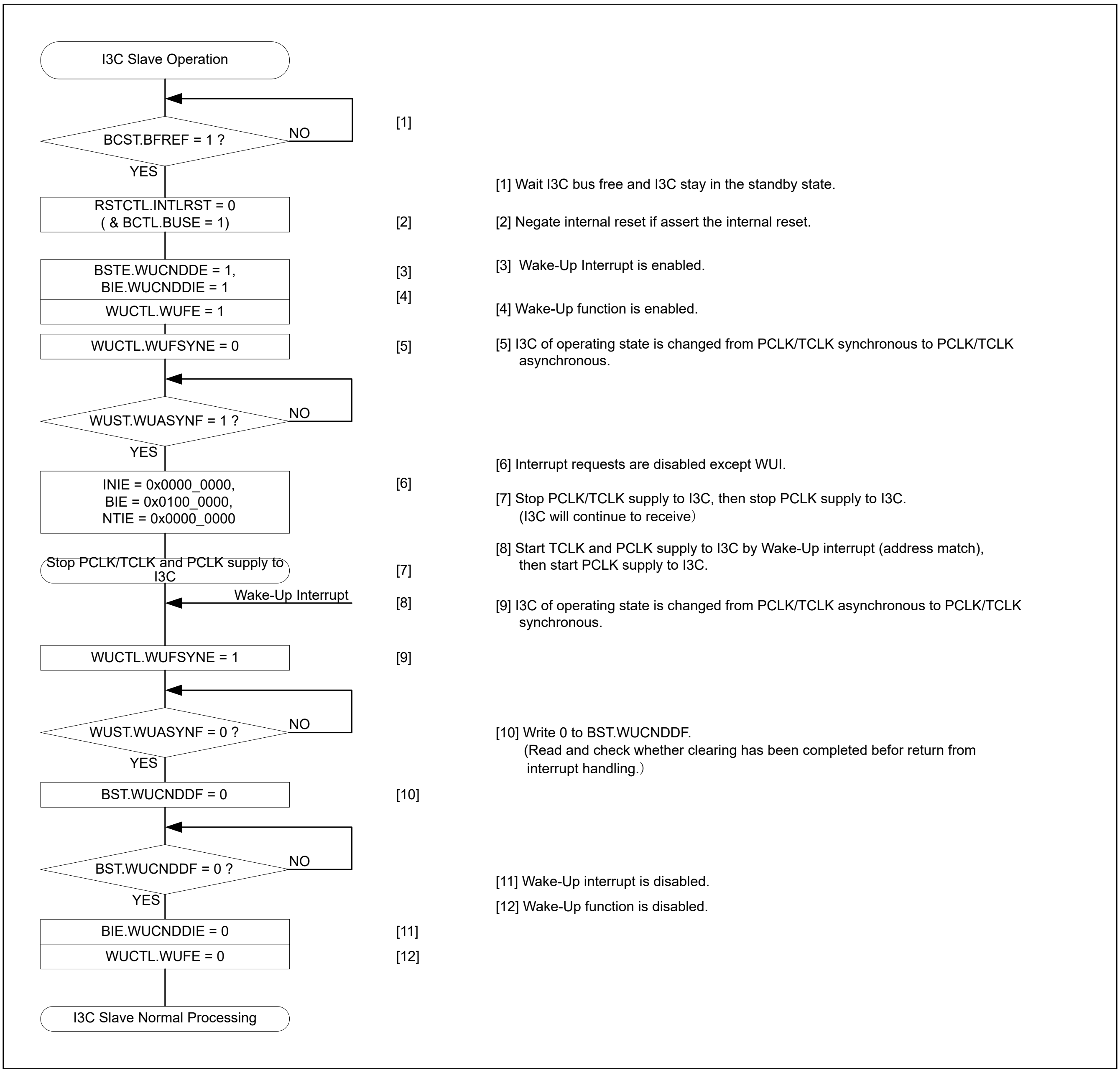


Figure 40.157 Use case of I3C slave wake-up (wake-up recovery by a wake-up interrupt triggered by the match of the slave address)

40.4 Interrupt Sources

I3C can generate the following interrupt requests:

40.4.1 Overview

The I3C has the interrupt factors shown in Table 40.18.

Table 40.18 Interrupt Generation (1 of 2)

Symbol	Interrupt source	Support			
		I2C	I3CM	I3C2M	I3CS
I3C_RESP	Normal Response Queue full	—	✓	✓	✓

Table 40.18 Interrupt Generation (2 of 2)

Symbol	Interrupt source	Support			
		I2C	I3CM	I3C2M	I3CS
I3C_CMD	Normal Command queue empty	—	✓	✓	✓
I3C_IBI	Normal IBI Queue Empty/Full	—	✓	✓	✓
I3C_RX	Normal Rx Data buffer full	✓	✓	✓	✓
I3C_TX	Normal Tx Data buffer empty	✓	✓	✓	✓
I3C_RCV	Normal Receive Status queue full	—	—	✓	✓
I3C_HRESP	High Priority Response queue full	—	✓	✓	—
I3C_HCMD	High Priority Command queue empty	—	✓	✓	—
I3C_HRX	High Priority Rx Data buffer full	—	✓	✓	—
I3C_HTX	High Priority Tx Data buffer empty	—	✓	✓	—
I3C_TEND	Transmit end	✓	—	—	—
I3C_EEI	Non-recoverable internal error	—	✓	✓	✓
	Normal Transfer Error	—	✓	✓	✓
	Normal Transfer Abort	—	✓	✓	✓
	High Priority Transfer Error	—	✓	✓	—
	High Priority Transfer Abort	—	✓	✓	—
	START condition detection	✓	✓	✓	✓
	STOP condition detection	✓	✓	✓	✓
	HDR Exit Pattern detection	—	✓	✓	✓
	NACK detection	✓	—	—	—
	Arbitration lost	✓	—	—	—
	Timeout detection	✓	✓	✓	✓
I3C_STEV	Synchronous Timing	—	✓	✓	✓
I3C_MREFOVF	MREF Counter Overflow	—	✓	✓	—
I3C_MREFCPT	MREF Capture	—	✓	✓	—
I3C_AMEV	Additional Master-initiated bus Event	—	✓	✓	—
I3C_WU	Wake-up condition detection	✓	✓	✓	✓

Note: ✓ : Support
— : Not support

Note: I^2C: I^2C Master/Slave (Single Buffer)
I3CM: I3C Master
I3C2M: I3C Secondary Master
I3CS: I3C Slave

Note: There is a delay time between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. Thus, when an interrupt flag has been cleared, read the relevant flag again to check whether clearing has been completed, and then return from interrupt handling. Returning from interrupt handling without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.

For I2C Protocol mode:

- Since I3C_TX is an edge-detected interrupted, it does not require clearing. Furthermore, the NTST.TDBEF0 flag (a condition for I3C_TX) is automatically set to 0 when data for transmission are written to NTDTBP0 or a STOP condition is detected (SPCNDDF flag = 1 in BST).

- Since I3C_RX is an edge-detected interrupted, it does not require clearing. Furthermore, the NTST.RDBFF0 flag (a condition for I3C_RX) is automatically set to 0 when data are read from NTDTBP0.

For I3C Protocol mode:

For details, refer to the detailed explanation of each flag bit.

- The I3C_CMD, I3C_TX, I3C_HCMD, I3C_HTX and I3C_IBI (I3C Slave) interrupts are cleared under the following conditions.
  On completion of the last write access by DMAC/DTC.
  Write 0 to this bit after 1 state is read by CPU.
- The I3C_RESP, I3C_IBI (I3C Master), I3C_RX, I3C_RCV, I3C_HRESP and I3C_HRX interrupts are cleared under the following conditions.
  On completion of the last read access by DMAC/DTC.
  Write 0 to this bit after 1 state is read by CPU.

40.4.2 Buffer Operation for Buffer Full/Empty Interrupts

If the conditions for generating each buffer full/empty interrupts are satisfied while the corresponding IR flag is 1, the interrupt request is not output for the ICU but retained internally (the capacity for internal retention is one request per source).

An interrupt request that was being retained within the ICU is output when the value of the IELSRn.IR flag becomes 0. Internally retained interrupt requests are automatically cleared under normal conditions of usage. Internally retained interrupt requests can also be cleared by writing 0 to the interrupt enable bit within the given peripheral module.

40.5 Event Link Output

I3C handles event output for the event link controller (ELC) corresponding to the following sources.

(1) Communication event

When a Communication event (arbitration-lost detection, detection of NACK, detection of timeout, detection of a START condition, or detection of a STOP condition) occurs, the corresponding event signal can be output for another module via the ELC.

(2) Rx Data buffer full

When a receive data register becomes full, the corresponding event signal can be output for another module via the ELC.

(3) Tx Data buffer empty

When a transmit data register becomes empty, the corresponding event signal can be output for another module via the ELC.

(4) Transmit end

On completion of transfer, the corresponding event signal can be output for another module via the ELC.

Other events are also available. For details, see section 19, Event Link Controller (ELC)18.2.3 ELSRn : Event Link Setting Register n.

40.5.1 Interrupt Handling and Event Linking

I3C module produces four kinds of interrupt: communication event (arbitration-lost detection, detection of NACK, detection of timeout, detection of a START condition, or detection of a STOP condition), Rx Data buffer full, Tx Data buffer empty, and transmit end interrupts. Each of these has an enable bit to control enabling and disabling of the interrupt signal. An interrupt request signal is output for the CPU when an interrupt source condition is satisfied while the setting of the corresponding enable bit is enabled.

The corresponding event link output signals are sent to other modules as event signals via the ELC when the interrupt source conditions are satisfied, regardless of the settings of the interrupt enable bits. For details on interrupt sources, see section 40.4.1. Overview.

40.6 Reset Descriptions

Table 40.19 Register states when issuing each condition (1) (1 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
PRTS	PRTMD	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BCTL	BUSE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSM	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ABT	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	BMDS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	INCBA	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
MSDVAD	MDYADV	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	MDYAD[6:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
RSTCTL	INTLRST	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HRDBRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HTDBRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HRSPQRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HCMDQRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSQRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	IBIQRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RDBRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TDBRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQRST	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RI3CRST	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
PRSST	PRSSTWP	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TRMD	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CRMS[*1]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
INST	INEF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Table 40.19 Register states when issuing each condition (1) (2 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
INSTE	INEE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
INIE	INEIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
INSTFC	INEFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
DVCT	IDX[4:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
IBINCTL	NRSIRCTL	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	NRMRCTL	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BFCTL	HSME	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	FMPE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SMBS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SCSYNE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SALE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	NALE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	MALE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (The FIFO corresponding to this register is cleared)
Note 1. In I3C mode, CRMS is not reset by INTLRST. In I2C mode, CRMS is reset by INTLRST

Table 40.20 Register states when issuing each condition (2) (1 of 3)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
SVCTL	SVAE[2]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SVAE[1]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SVAE[0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HOAE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVIDE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HSMCE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	GCAE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
REFCKCTL	IREFCKS [2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
STDBR	DSBRPO	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SBRHP[5:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SBRLP[5:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SBRHO[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SBRLO[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
EXTBR	EBRHP[5:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	EBRLP[5:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	EBRHO[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	EBRLO[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BFRECDT	FRECYC [8:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BAVLCDT	AVLCYC[8:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BIDLCDT	IDLCYC [17:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Table 40.20 Register states when issuing each condition (2) (2 of 3)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
OUTCTL	SDODCS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SDOD[2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	EXCYC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SOCWP	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SCOC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SDOC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
INCTL	SDID[1:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DNFE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DNFS[3:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
TMOCTL	TOMDS[1:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TOHCTL	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TOLCTL	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TODTS[1:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
WUCTL	WUFE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	WUFSYNE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	WUANFS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	WUACKS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
ACKCTL	ACKTWP	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ACKT	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ACKR	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
SCSTRCTL	RWE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ACKTWE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Table 40.20 Register states when issuing each condition (2) (3 of 3)

Register symbol	Register bit name	System reset	RSTCTL Register RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
SCSTLCTL	ACKPE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	PARPE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	AAPE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	STLCYC [15:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
SVTDLG0	STDLG[15:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (The FIFO corresponding to this register is cleared)

Table 40.21 Register states when issuing each condition (3) (1 of 3)

Register symbol	Register bit name	System reset	RSTCTL Register RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
STCTL	STOE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
ATCTL	CDIV[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	AMEOE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	MREFOE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ATTRGS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
ATTRG	ATSTRG	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
ATCCNTE	ATCE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CNDCTL	SPCND	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SRCND	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	STCND	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
NCMDQP	NCMDQP [31:0]	In reset	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
NRSPQP	NRSPQP [31:0]	In reset	In reset	In reset	Saved	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
NTDTBP0	NTDTBP0 [31:0]	In reset	In reset	In reset	Saved	Saved	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved
NIBIQP	NIBIQP[31:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	In reset	Saved	Saved	Saved	Saved	Saved

Table 40.21 Register states when issuing each condition (3) (2 of 3)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
NRSQP	NRSQP [31:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	In reset	Saved	Saved	Saved	Saved
HCMDQP	HCMDQP [31:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	In reset	Saved	Saved	Saved
HRSPQP	HRSPQP [31:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	In reset	Saved	Saved
HTDTBP	HTDTBP [31:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	In reset	In reset
NQTHCTL	IBIQTH[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	IBIDSSZ[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQTH [7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQTH [7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
NTBTHCTL0	RXSTTH [2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TXSTTH[2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RXDBTH [2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TXDBTH [2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
NRQTHCTL	RSQTH[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
HQTHCTL	RSPQTH [7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQTH [7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
HTBTHCTL	RXSTTH [2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TXSTTH[2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RXDBTH [2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TXDBTH [2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Table 40.21 Register states when issuing each condition (3) (3 of 3)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
BST	WUCNDDF	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TODF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ALF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TENDF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	NACKDF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HDREXDF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SPCNDDF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	STCNDDF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BSTE	WUCNDDE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TODE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ALE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TENDE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	NACKDE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HDREXDE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SPCNDDE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	STCNDDE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (The FIFO corresponding to this register is cleared)

Table 40.22 Register states when issuing each condition (4) (1 of 3)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
BIE	WUCNDDIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TODIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ALIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TENDIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	NACKDIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HDREXDIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SPCNDDIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	STCNDDIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BSTFC	WUCNDDFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TODFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ALFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TENDFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	NACKDFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HDREXDFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SPCNDDFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	STCNDDFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Table 40.22 Register states when issuing each condition (4) (2 of 3)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
NTST	RSQFF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	In reset	Saved	Saved	Saved	Saved
	TEF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TABTF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQFF	In reset	In reset	In reset	Saved	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQEF	In reset	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	IBIQEFF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	In reset	Saved	Saved	Saved	Saved	Saved
	RDBFF0	In reset	In reset	In reset	Saved	Saved	Saved	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	TDBEF0	In reset	In reset	In reset	Saved	Saved	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
NTSTE	RSQFE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TEE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TABTE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQFE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQEE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	IBIQEFE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RDBFE0	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TDBEE0	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Table 40.22 Register states when issuing each condition (4) (3 of 3)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
NTIE	RSQFIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TEIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TABTIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQFIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQEIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	IBIQEFIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RDBFIE0	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TDBEIE0	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (The FIFO corresponding to this register is cleared)

Table 40.23 Register states when issuing each condition (5) (1 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
NTSTFC	RSQFFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TEFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TABTFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQFFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQEFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	IBIQEFFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RDBFFC0	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TDBEFC0	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Table 40.23 Register states when issuing each condition (5) (2 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
HTST	TEF	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TABTF	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQFF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	In reset	Saved	Saved
	CMDQEF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	In reset	Saved	Saved	Saved
	RDBFF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	In reset
	TDBEF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	In reset	Saved
HTSTE	TEE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TABTE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQFE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQEE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RDBFE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TDBEE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
HTIE	TEIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TABTIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQFIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQEIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RDBFIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TDBEIE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (The FIFO corresponding to this register is cleared)

Table 40.24 Register states when issuing each condition (6) (1 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
HTSTFC	TEFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TABTFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RSPQFFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQEFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	RDBFFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TDBEFC	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BCST	BIDLF	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	BAVLF	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	BFREF	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
SVST	SVAF[2]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SVAF[1]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SVAF[0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HOAF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVIDF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	HSMCF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	GCAF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
WUST	WUASYNF	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
MRCCPT	MRCCPT [31:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Table 40.24 Register states when issuing each condition (6) (2 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
DATBASm (m = 0 to 7)	DVTYP	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVNACK [1:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVDYAD [7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVIBITS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVMRRJ	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVSIRRJ	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVIBIPL	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVADLS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DVSTAD [9:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (The FIFO corresponding to this register is cleared)

Table 40.25 Register states when issuing each condition (7) (1 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
EXDATBAS	EDTYP	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	EDNACK [1:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	EDDYAD [7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	EDADLS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	EDSTAD [9:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
SDATBASn (n = 0 to 2)	SDDYAD [6:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SDIBIPL	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SDADLS	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SDSTAD [9:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
MSDCTm (m = 0 to 7)	RBCRn	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Table 40.25 Register states when issuing each condition (7) (2 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
SVDCT	TBCRn	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TDCR[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (The FIFO corresponding to this register is cleared)

Table 40.26 Register states when issuing each condition (8) (1 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
SDCTPIDL	SDCTPIDL [31:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
SDCTPIDH	SDCTPIDH [31:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
SVDVADn (n = 0 to 2)	SDYADV	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SSTADV	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SADLG	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SVAD[9:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CSECMD	MSRQE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SVIRQE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CEACTST	ACTST[3:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CMWLG	MWLG[15:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CMRLG	IBIPSZ[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	MRLG[15:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CETSTMD	TSTMD[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CGDVST	VDRSV[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ACTMD[1:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	PRTE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	PNDINT[3:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Table 40.26 Register states when issuing each condition (8) (2 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
CMDSPW	MSWDR[2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CMDSPR	CDTTIM[2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	MSRDR[2:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CMDSPT	MRTE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	MRTTIM [23:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (The FIFO corresponding to this register is cleared)

Table 40.27 Register states when issuing each condition (9) (1 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
CETSM	INAC[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	FREQ[7:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SPTASYN [3:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SPTSYN	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CGHDRCAP	TSLEN	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	TSPEN	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	DDREN	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CETSS	ICOVF	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	ASYNE[3:0]	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SYNE	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
BITCNT	BCNT[4:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Table 40.27 Register states when issuing each condition (9) (2 of 2)

Register symbol	Register bit name	System reset	RSTCTL Register											
			RI3CRST	INTLRST	CMDQRST	RSPQRST	TDBRST	RDBRST	IBIQRST	RSQRST	HCMDQRST	HRSPQRST	HTDBRST	HRDBRST
NQSTLV	IBISCNT[4:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	In reset	Saved	Saved	Saved	Saved	Saved
	IBIQLV[7:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	In reset	Saved	Saved	Saved	Saved	Saved
	RSPQLV [7:0]	In reset	In reset	In reset	Saved	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	CMDQFLV [7:0]	In reset	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
NDBSTLV0	RDBLV[7:0]	In reset	In reset	In reset	Saved	Saved	Saved	In reset	Saved	Saved	Saved	Saved	Saved	Saved
	TDBFLV[7:0]	In reset	In reset	In reset	Saved	Saved	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved
NRSQSTLV	RSQLV[7:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	In reset	Saved	Saved	Saved	Saved
HQSTLV	RSPQLV [7:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	In reset	Saved	Saved
	CMDQLV [7:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	In reset	Saved	Saved	Saved
HDBSTLV	RDBLV[7:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	In reset
	TDBFLV[7:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	In reset	Saved
PRSTDBG	SDOLV	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SCOLV	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SDILV	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
	SCILV	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
MSERRCNT	M2ECNT [7:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
SC1CPT	SC1C[15:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
SC2CPT	SC2C[15:0]	In reset	In reset	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved
CECTL	CLKE	In reset	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved	Saved

Note: In reset: To be reset (The FIFO corresponding to this register is cleared)

40.7 Usage Notes

40.7.1 Settings for the Operating Clock

The following relation is required between the frequencies of the bus clock (PCLK) and transfer clock(TCLK).

$TCLK/2 \leq PCLK \leq TCLK$

40.7.2 Settings for the SCL Frequency Clock

As described in Chapter 5.1.2.4 of MIPI I3C Specification V1.0, the SCL frequency setting is required depending on the bus configuration.

In Mixed Bus with both I3C and legacy I2C devices are present on the bus, If the Legacy I2C device has a 50ns Spike Filter, I3C Master can change the frequency by setting the SCL High period to less than 50ns (tDIG_H_MIXED).

40.7.3 Module-stop Function

I3C operation can be disabled or enabled using Module Stop Control Register B (MSTPCRB). The I3C module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Mode.

40.7.4 Notes on Rewriting PRTS.PRTMD Bit

When the PRTS.PRTMD bit is rewritten, the SCL terminal (and SDA terminal) may output low. In this case, the bus line is unintentionally put into a busy state, and communication cannot be started even if the communication start setting is made after that.

Set the RSTCTL.INTRST bit to 1 and issue an internal software reset before rewriting the PRTMD bit. After rewriting, set the RSTCTL.INTRST bit to 0 to release the internal software reset.

However, no countermeasure is necessary if the following conditions are applicable.

(1) When the PRTS.PRTMD bit is not changed from its initial value of 1 (I2C protocol mode) to 0 (I3C protocol mode).

(2) When the clock source selected by SCKSCR.CKSEL[2:0] and I3CCKCR.I3CCKSEL[3:0] are the same.

(3) In master mode, when the timeout error detection function is enabled under the following conditions and a software reset is set in the I3C error interruption routine to release unintended low output.

1. Set the BSTE.TODE bit to 1 to enable the timeout error detection function.
2. Select TMOCTL.TOMDS[1:0] = 00b or 01b as the timeout operation mode and set the TMOCTL.TOLCTL bit = 1 and TMOCTL.TOHCTL bit = 1 to treat the SCL line fixed at low and fixed at high as errors.
3. In the I3C error interruption routine, set the RSTCTL.RI3CRST bit to 1 to perform an I3C software reset, or set the RSTCTL.INTLRST bit to 1 to perform an internal software reset.

(4) In slave mode, when the timeout error detection function is enabled under the following conditions and a software and TMOCTL.TOHCTL bit = 1 to treat the SCL line fixed at low and fixed at high as errors.

1. Set the BSTE.TODE bit to 1 to enable the timeout error detection function.
2. Select TMOCTL.TOMDS[1:0] = 01b as the timeout operation mode and set TMOCTL.TOLCTL = 1 and TMOCTL.TOHCTL = 1 to treat the SCL line fixed at low and fixed at high as errors.
3. In the I3C error interruption routine, set the RSTCTL.RI3CRST bit to 1 to perform an I3C software reset, or set the RSTCTL.INTLRST bit to 1 to perform an internal software reset.

41. CAN with Flexible Data-rate (CANFD)

41.1 Overview

The CAN with Flexible Data-rate (CANFD) supports the following functions:

- CAN with Flexible Data-rate.*1

Note 1. This feature is not available in the classical CAN function.

The CANFD module has a flexible message buffer and FIFO structure that meet the requirements of various applications. It also provides test modes to achieve high testability of the module that can be useful for power-on testing.

This specification describes of the CANFD module.

The CANFD mode is only available in certain products that support it.

41.1.1 CANFD Module

Table 41.1 CANFD module specifications (1 of 2)

Parameter		Specifications
Communication		CAN functionality conforms to CANFD ISO 11898-1 (2015)
Protocol engine version		RS-CANFD_PE V3.0
Data transfer rate	CANFD*1	Up to 1 Mbps for arbitration phase and up to 8 Mbps for data phase, individually for CAN channel
	Classical CAN	Up to 1 Mbps
Operation frequency of Peripheral module clock		100 MHz (PCLKA) RAM clock: 200 MHz (PCLKE)
Data Link Layer (DLL) clock		8 MHz to 80MHz The following clocks can be selected. ● CANMCLK : External Oscillator Clock ● CANFDCLK : CANFD Core Clock
Input/Output pins		CRXn/CTXn (n = 0, 1)
CAN channels		2 channels
Selectable ID type		11-bit Standard ID
		11-bit Standard ID + 18-bit Extended ID
Selectable frame type		Data frame (RTR = 0) (CAN and CANFD frames)
		Remote frame (RTR = 1) (only CAN frames)
Variable data byte count for data frames		DLC range: 0 to F
Message buffer		Up to 16 reception message buffers
		4 transmit message buffers per channel
		1 transmission queue per channel Automatic message transfer into transmission queues supported
FIFO number		2 reception FIFO buffers 1 COMMON FIFO individually configurable as: ● Reception FIFO ● Transmission FIFO
Automatic delay interval timer for transmission		The delay timer can be applied to: ● Transmission FIFO

Table 41.1 CANFD module specifications (2 of 2)

Parameter	Specifications
Enhanced reception filtering	Support of 11 bits and 29 bits CAN identifier
	Programmable 29 bits CAN identifier acceptance filter mask for each entry
	Programmable routing capability for each FIFO and reception message buffers (up to 2 routing destinations)
	RTR and IDE masking
	Data Length Code (DLC) filter
	Message buffer payload overload protection
	Payload filter
	Updating Acceptance Filter List (AFL) entry during communication
General software support	Automatic label information added to receive message (for upper software layer support)
Timer	TX and RX Time Stamp function
Power down function	Module start stop function for each CAN node (Channel and Global Sleep mode)
RAM	RAM ECC protected (2 bits error detection, 1-bit error correction)
Module-stop function	Module-stop state can be set for each channels to reduce power consumption.
TrustZone Filter	Security and Privilege attribution can be set for each channels.

Note 1. The CANFD mode is available only for CANFD supported product.

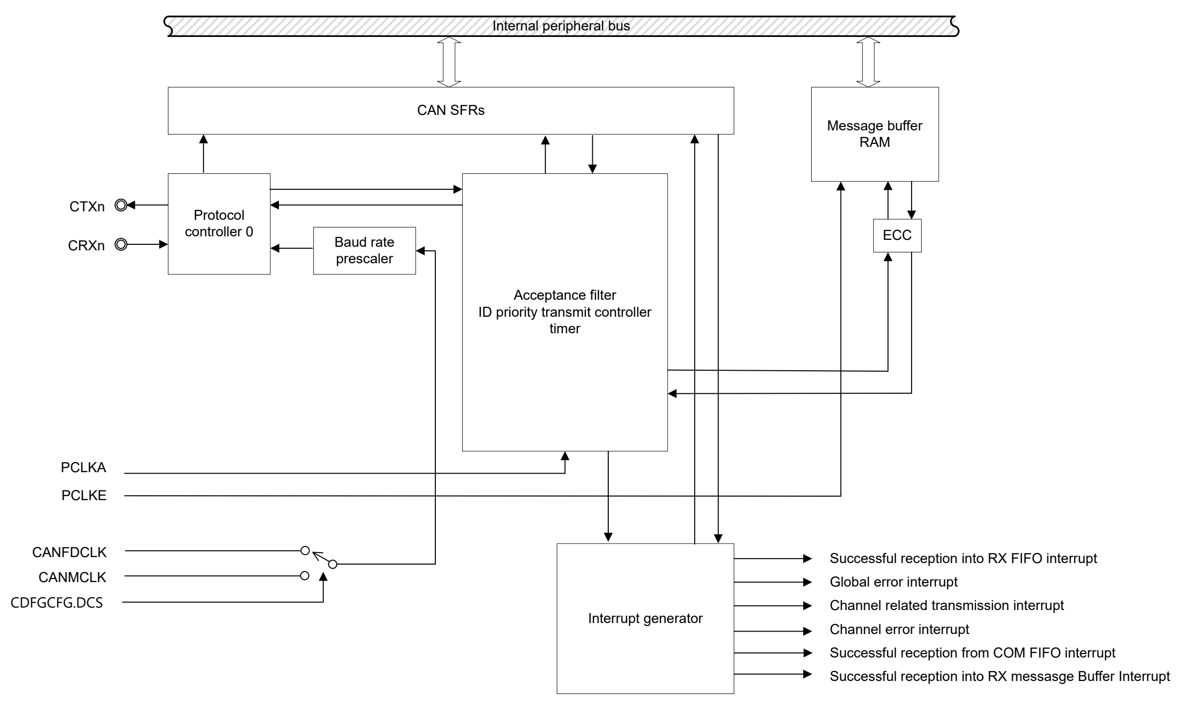


Figure 41.1 Overview of the CANFD module

- CRXn/CTXn:
  Input/Output pins of the CANFD module
- Protocol controller:
  Handles CAN protocol processing such as bus arbitration, bit timing at transmission and reception, stuffing, error handling

- Message buffer RAM (MBRAM):
  This RAM is used to store messages after reception or for transmission using a normal message buffer or a FIFO. Each message entry has an individual ID, data length code, data field, message pointer for upper layer application usage and a time stamp.
  This RAM is used to store the message acceptance filtering entries. Each acceptance filter entry has an individual ID, data length code, data field, message pointer for upper layer application usage and message direction pointer.
- Acceptance filter list RAM (AFLRAM):
  Performs filtering of received messages. The entries in the Acceptance filter list RAM are used for the filtering process.
- Two timers:
  - Reception Timestamp function
  - Transmission separation time for FIFO buffers
- Interrupt generator:
  Generates several types of global and channel interrupts
- CAN Special Function Registers (SFRs):
  Registers associated with CAN. See section 41.2. Register Descriptions.

41.1.2 Clock Restriction

For the CAN communication the following restriction for the clocks should be satisfied:

- PCLKE / 2 = PCLKA ≥ CANFDCLK
- PCLKE / 2 = PCLKA ≥ CANMCLK

To avoid missing events the CAN engine clock (CANFDCLK or CANMCLK) frequency must be less than the PCLKA clock frequency.

To avoid loss of CAN message, the PCLKA should be set to a clock with a frequency depend on the CAN communication baud rate. The constraint of a baud rate and a PCLKA clock is shown in Table 41.2.

Table 41.2 Clock restriction

	Baud rate	PCLKA
CANFD	1 Mbps Nominal 8 Mbps Data	PCLKA ≥ 40 MHz
	500 Kbps Nominal 5 Mbps Data	PCLKA ≥ 32 MHz
Classical CAN	1 Mbps Data	PCLKA ≥ 32 MHz

The frequency of CANFDCLK and CANMCLK depend on the required baud rate. For information how to configure the baud rate, see section 41.4.1.3. Baud Rate.

41.2 Register Descriptions

41.2.1 Register Table

The reset value shown for the RAM area, consisting of CFDGAFLIDr, CFDGAFLMr, CFDGAFLP0r, CFDGAFLP1r, CFDRMBCPb, CFDRFMBCPb, CFDCFMBCP0, CFDTMBCPb, CFDTHLACC0, CFDTHLACC1 and CFDRPGACCk is valid after initialization of a hardware reset. See section 41.4.2. CAN Module Configuration after Hardware Reset for details of the initialization process.

If a write access with a size of 8 or 16 bits is performed for the RAM area, then the CANFD module does a read-modify write-access to the RAM location, because the RAM requires a 32-bit access through the ECC module.

For single bit error, the correct data is written back. For multiple bit errors, unknown data is written back.

Do not access the space where the register is not assigned.

The read data from the space where the register is not assigned is unknown.

41.2.2 CFDC0NCFG : Nominal Bitrate Configuration Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0000

Bit position:	31–25	24–17	16–10	9–0
Bit field:	NTSEG2[6:0]	NTSEG1[7:0]	NSJW[6:0]	NBRP[9:0]
Value after reset:	0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
9:0	NBRP[9:0]	Channel Nominal Baud Rate Prescaler Nominal baud rate prescaler division ratio	R/W
16:10	NSJW[6:0]	Resynchronization Jump Width 0x00: 1 Tq 0x01: 2 Tq ⋮ 0x7E: 127 Tq 0x7F: 128 Tq	R/W
24:17	NTSEG1[7:0]	Timing Segment 1 0x00: Reserved 0x01: 2 Tq 0x02: 3 Tq 0x03: 4 Tq ⋮ 0xFE: 255 Tq 0xFF: 256 Tq	R/W
31:25	NTSEG2[6:0]	Timing Segment 2 0x00: Reserved 0x01: 2 Tq ⋮ 0x7E: 127 Tq 0x7F: 128 Tq	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Tq means time quantum.

This register configures the transmission/reception nominal baud rate parameters of the channels.

NBRP[9:0] bits (Channel Nominal Baud Rate Prescaler)

The NBRP[9:0] bits are used to define the peripheral bus clock periods contained in a time quantum.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the CANFD channel is in CH_RESET or CH_HALT mode.

NSJW[6:0] bits (Resynchronization Jump Width)

The NSJW[6:0] bits set the synchronization jump width. A value from 1 to 128 time quanta can be set.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the CANFD channel is in CH_RESET or CH_HALT mode.

NTSEG1[7:0] bits (Timing Segment 1)

The NTSEG1[7:0] bits set the segment TSEG1 to compensate for edges on the CAN bus with a positive phase error. These bits contain the propagation segment.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the CANFD channel is in CH_RESET or CH_HALT mode.

Additionally, configure a Tq value only between 2 and 256, inclusive. See section 41.4.1.2. CAN Bit Timing for more details.

NTSEG2[6:0] bits (Timing Segment 2)

The NTSEG2[6:0] bits set the segment TSEG2 to compensate for edges on the CAN bus with a negative phase error.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the CANFD channel is in CH_RESET or CH_HALT mode.

Additionally, configure a Tq value only between 2 and 128, inclusive.

41.2.3 CFDC0CTR : Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ROM	BFT	—	—	—	CTMS[1:0]		CTME	ERRD	BOM[1:0]		—	TDCVFIE	SOCOIE	EOCOIE	TAIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ALIE	BLIE	OLIE	BORIE	BOEIE	EPIE	EWIE	BEIE	—	—	—	—	RTBO	CSLPR	CHMDC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Function	R/W
1:0	CHMDC[1:0]	Channel Mode Control 0 0: Channel operation mode request 0 1: Channel reset request 1 0: Channel halt request 1 1: Keep current value	R/W
2	CSLPR	Channel Sleep Request 0: Channel sleep request disabled 1: Channel sleep request enabled	R/W
3	RTBO	Return from Bus-Off 0: Channel is not forced to return from bus-off 1: Channel is forced to return from bus-off	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	BEIE	Bus Error Interrupt Enable 0: Bus error interrupt disabled 1: Bus error interrupt enabled	R/W
9	EWIE	Error Warning Interrupt Enable 0: Error warning interrupt disabled 1: Error warning interrupt enabled	R/W
10	EPIE	Error Passive Interrupt Enable 0: Error passive interrupt disabled 1: Error passive interrupt enabled	R/W
11	BOEIE	Bus-Off Entry Interrupt Enable 0: Bus-off entry interrupt disabled 1: Bus-off entry interrupt enabled	R/W
12	BORIE	Bus-Off Recovery Interrupt Enable 0: Bus-off recovery interrupt disabled 1: Bus-off recovery interrupt enabled	R/W
13	OLIE	Overload Interrupt Enable 0: Overload interrupt disabled 1: Overload interrupt enabled	R/W
14	BLIE	Bus Lock Interrupt Enable 0: Bus lock interrupt disabled 1: Bus lock interrupt enabled	R/W

Bit	Symbol	Function	R/W
15	ALIE	Arbitration Lost Interrupt Enable 0: Arbitration lost interrupt disabled 1: Arbitration lost interrupt enabled	R/W
16	TAIE	Transmission Abort Interrupt Enable 0: TX abort interrupt disabled 1: TX abort interrupt enabled	R/W
17	EOCOIE	Error Occurrence Counter Overflow Interrupt Enable 0: Error occurrence counter overflow interrupt disabled 1: Error occurrence counter overflow interrupt enabled	R/W
18	SOCOIE	Successful Occurrence Counter Overflow Interrupt Enable 0: Successful occurrence counter overflow interrupt disabled 1: Successful occurrence counter overflow interrupt enabled	R/W
19	TDCVFIE*1	Transceiver Delay Compensation Violation Interrupt Enable 0: Transceiver delay compensation violation interrupt disabled 1: Transceiver delay compensation violation interrupt enabled	R/W
20	—	This bit is read as 0. The write value should be 0.	R/W
22:21	BOM[1:0]	Channel Bus-Off Mode 0 0: Normal mode (comply with ISO 11898-1) 0 1: Entry to Halt mode automatically at bus-off start 1 0: Entry to Halt mode automatically at bus-off end 1 1: Entry to Halt mode (during bus-off recovery period) by software	R/W
23	ERRD	Channel Error Display 0: Only the first set of error codes displayed 1: Accumulated error codes displayed	R/W
24	CTME	Channel Test Mode Enable 0: Channel test mode disabled 1: Channel test mode enabled	R/W
26:25	CTMS[1:0]	Channel Test Mode Select 0 0: Basic test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (External loopback mode) 1 1: Self-test mode 1 (Internal loopback mode)	R/W
29:27	—	These bits are read as 0. The write value should be 0.	R/W
30	BFT	Bit Flip Test 0: First data bit of reception stream not inverted 1: First data bit of reception stream inverted	R/W
31	ROM*1	Restricted Operation Mode 0: Restricted operation mode disabled 1: Restricted operation mode enabled	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. These bits are not available in the classical CAN function.

Channel Control register controls the modes of the related channel. It is used to enable generation of interrupts if errors are detected on the CAN bus connected to this channel. It is also used to configure the channel in test mode.

CHMDC[1:0] bits (Channel Mode Control)

The CHMDC[1:0] bits can be used to configure modes of the CAN channel.

CAN mode transitions are described in more details in section 41.3.3. Channel Modes.

Setting CHMDC[1:0] bits to 11b has no effect. When the CANFD module is in GL_HALT mode, these bits can only be set to 10b or 01b. These bits cannot be set in CH_SLEEP mode.

These bits can change automatically when transitioning to Halt mode by the CFDC0CTR.BOM settings.

If CPU write access to CFDC0CTR.CHMDC occurs at the same time when the CAN channel enters Halt mode (at the start of bus-off when CFDC0CTR.BOM = 01b, or at the end of bus-off when CFDC0CTR.BOM = 10b), then the CPU write access has the highest priority.

The CAN channel changes the value of CFDC0CTR.CHMDC within the Channel Control Registers for the specified cases only if the CFDC0CTR.CHMDC value is 00b (Operation mode).

CSLPR bit (Channel Sleep Request)

When the CSLPR bit is 1, a Sleep mode request is generated for the corresponding CAN channel

When this bit is 0, a request to exit Sleep mode is generated for the related CANFD channel.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_SLEEP mode.

RTBO bit (Return from Bus-Off)

When the protocol controller of the CAN channel enters bus-off state, you can force a recovery from bus-off state by setting the RTBO bit in the Channel Control Register to 1.

The error state changes from bus-off state to integrating with a maximum delay of 1 CAN bit time.

When the RTBO bit is set to 1, the REC and TEC registers are initialized and the Bus-Off Status bit (Channel Bus-off Status, CFDC0STS.BOSTS) is set to 0.

Registers other than the REC and TEC registers are not initialized by this command. Even if CFDC0CTR.BORIE is set, a bus-off recovery interrupt is not generated by this recovery from the bus-off state.

The RTBO bit cannot be set in CH_SLEEP mode. Setting this bit in any state other than bus-off state has no effect and the bit is cleared immediately. The read value is always 0.

Return from the Bus-Off command should be used only when CFDC0CTR.BOM is set to 00b.

Only write to this bit when the related CANFD channel is in CH_OPERATION mode. This bit is automatically cleared when set by software.

BEIE bit (Bus Error Interrupt Enable)

When the BEIE and the CFDC0ERFL.BEF bits are both 1, an error interrupt request is generated.

This bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

EWIE bit (Error Warning Interrupt Enable)

When the EWIE and the CFDC0ERFL.EWF bits are both 1, an error interrupt request is generated.

The EWIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

EPIE bit (Error Passive Interrupt Enable)

An error interrupt request is generated when the EPIE bit and the CFDC0ERFL.EPF are both 1.

The EPIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

BOEIE bit (Bus-Off Entry Interrupt Enable)

When the BOEIE and the CFDC0ERFL.BOEF bits are both 1, an error interrupt request is generated.

The BOEIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

BORIE bit (Bus-Off Recovery Interrupt Enable)

When the BORIE and the CFDC0ERFL.BORF bits are both 1, an error interrupt request is generated.

The BORIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

OLIE bit (Overload Interrupt Enable)

When the OLIE and the CFDC0ERFL.OVLF bits are both 1, an error interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

BLIE bit (Bus Lock Interrupt Enable)

When the BLIE and the CFDC0ERFL.BLF bits are both 1, an error interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

ALIE bit (Arbitration Lost Interrupt Enable)

When the ALIE and the CFDC0ERFL.ALF bits are both 1, an error interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

TAIE bit (Transmission Abort Interrupt Enable)

When the TAIE bit is 1 and a transmission is successfully aborted from a TX MB belonging to the corresponding CAN channel, an interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

EOCOIE bit (Error Occurrence Counter Overflow Interrupt Enable)

When the EOCOIE bit is 1 and the CFDC0FDSTS.EOCO bit belonging to the corresponding CAN channel is 1, an error interrupt request is generated.

The EOCOIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

SOCOIE bit (Successful Occurrence Counter Overflow Interrupt Enable)

When the SOCOIE bit is 1 and the CFDC0FDSTS.SOCO bit belonging to the corresponding CAN channel is 1, an error interrupt request is generated.

The SOCOIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET mode.

TDCVFIE bit (Transceiver Delay Compensation Violation Interrupt Enable)

When the TDCVFIE bit is 1 and the CFDC0FDSTS.TDCVF bit belonging to the corresponding CAN channel is 1, an error interrupt request is generated.

The TDCVFIE bit cannot be set in CH_SLEEP mode.

Only write to this bit when the related CANFD channel is in CH_RESET mode. Do not set this bit when in Classical CAN mode.

Note: This bit is not available in the classical CAN function.

BOM[1:0] bits (Channel Bus-Off Mode)

The BOM[1:0]bits control the timing of the recovery from Bus-Off mode of the CANFD Channel.

Do not write to these bits in CH_SLEEP mode. Only write to these bits when the related CANFD channel is in CH_RESET mode.

Only write to these bits when the related CANFD channel is in CH_RESET mode.

ERRD bit (Channel Error Display)

The ERRD bit controls the display mode of the error flag bits [14:8] in the Channel Error Flag Register (CFDC0ERFL).

If the ERRD bit is 0 and more than one error occur at the same time, the error flag bits are set for all the errors that occurred at the same time. No further errors are flagged until CFDC0ERFL[14:8] is cleared.

Do not write to the ERRD bit in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode.

CTME bit (Channel Test Mode Enable)

The CTME bit enables the channel test modes.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_HALT mode.

CTMS[1:0] bits (Channel Test Mode Select)

The CTMS[1:0] bits are used to select the required test mode.

Do not write to these bits in CH_SLEEP or CH_RESET mode. Only write to these bits when the related CANFD channel is in CH_HALT mode.

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

BFT bit (Bit Flip Test)

The BFT bit checks the internal CRC generator logic of the protocol controller.

It inverts the first bit (ID bit) of the CAN message data stream being received, so that the internal generated CRC result will not match the received CRC value of the frame. Refer to the bit stuffing rule, when using this feature, as there is the possibility of receiving a stuff error (due to the inversion) rather than a CRC error.

The internal generated CRC value is always observed in the following registers:

- CFDC0ERFL.CRCREG (Classical CAN frames)
- CFDC0FDCRC.CRCREG (CANFD frames).*1

Note 1. This feature is not available in the classical CAN function.

Some restrictions exist when using this bit:

Other CAN node will send a reference message and the receiver node(s) can invert one bit of incoming bit stream.

Note: The transmitter and receiver modes share the same CRC generator, therefore it is not necessary to consider the modes separately when testing.

The Bit Flip test mode is enabled if the BFT (new control signal that inverts the first bit of the bit stream) and CTME bits are both 1 and CFDC0CTR.CTMS is 0x00.

If this function is used by a transmitting node, a bit error or an arbitration lost will occur.

Do not write to the BFT bit in CH_SLEEP mode. Users should not use this function when the Self test mode 1 (Internal Loop back mode). Only write to this bit when the related CANFD channel is in CH_HALT mode.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

ROM bit (Restricted Operation Mode)

When the ROM and CTME bits are both 1, the restricted operation mode is enabled. This mode should only be used in basic test mode (CFDC0CTR.CTMS[1:0] = 00b).

The ROM bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CANFD channel is in CH_HALT mode.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. Do not set this bit when in Classical CAN mode.

Note: This bit is not available in the classical CAN function.

41.2.4 CFDC0STS : Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TEC[7:0]								REC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ESIF	COMSTS	RECSTS	TRMSTS	BOSTS	EPSTS	CSLPSTS	CHLTSTS	CRSTSTS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Function	R/W
0	CRSTSTS	Channel Reset Status 0: Channel not in Reset mode 1: Channel in Reset mode	R
1	CHLTSTS	Channel Halt Status 0: Channel not in Halt mode 1: Channel in Halt mode	R
2	CSLPSTS	Channel Sleep Status 0: Channel not in Sleep mode 1: Channel in Sleep mode	R
3	EPSTS	Channel Error Passive Status 0: Channel not in error passive state 1: Channel in error passive state	R
4	BOSTS	Channel Bus-Off Status 0: Channel not in bus-off state 1: Channel in bus-off state	R
5	TRMSTS	Channel Transmit Status 0: Channel is not transmitting 1: Channel is transmitting	R
6	RECSTS	Channel Receive Status 0: Channel is not receiving 1: Channel is receiving	R
7	COMSTS	Channel Communication Status 0: Channel is not ready for communication 1: Channel is ready for communication	R
8	ESIF*1	Error State Indication Flag 0: No CANFD message has been received when the ESI flag was set 1: At least one CANFD message was received when the ESI flag was set	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
23:16	REC[7:0]	Reception Error Count These bits increment or decrement the counter value according to error status of the CAN channel during reception.	R
31:24	TEC[7:0]	Transmission Error Count These bits increment or decrement the counter value according to error status of the CAN channel during transmission.	R

Note: S-TYPE-3, P-TYPE-3

Note 1. This bit is not available in the classical CAN function.

Channel Status Register shows the mode, error and transmission or reception status of the related channel together with its reception and transmission error count values.

CRSTSTS bit (Channel Reset Status)

The CRSTSTS bit indicates whether the related CAN channel is in Reset mode.

This bit is set automatically when the related CAN channel enters Channel Reset mode. When the mode is changed from Reset mode to Sleep mode, the CRSTSTS bit remains 1.

This bit is cleared automatically when the related CAN channel exits the Channel Reset mode, except when changing to Sleep mode.

CHLTSTS bit (Channel Halt Status)

The CHLTSTS bit indicates whether the related CAN channel is in Halt mode.

This bit is set automatically when the related CAN module enters Halt mode, and is cleared automatically when the related CAN module exits Halt mode.

CSLPSTS bit (Channel Sleep Status)

The CSLPSTS bit indicates whether the related CAN channel is in Sleep mode.

This bit is set automatically when the related CANFD channel enters Sleep mode, and is cleared automatically when the related CANFD channel exits Sleep mode.

EPSTS bit (Channel Error Passive Status)

The EPSTS bit indicates whether the related CANFD channel has entered the error passive state.

This bit is set automatically when the value of the CAN Transmission or Reception Counter Register exceeds the value of 0x7F.

This bit is cleared automatically when the related CANFD channel exits the error passive state or enters Reset mode.

BOSTS bit (Channel Bus-Off Status)

The BOSTS bit indicates whether the related CANFD channel has entered the error bus-off state.

This bit is set automatically when the value of the related CAN Transmission Error Count Register exceeds 0xFF and the related CANFD channel is in the bus-off state (CAN Transmission Error Count Register > 0xFF).

This bit is cleared automatically when the related CANFD channel exits bus-off state.

TRMSTS bit (Channel Transmit Status)

The TRMSTS bit indicates whether the related CANFD channel is transmitting a message.

This bit is set automatically when the related CANFD channel is operating as a transmitter node or is in the bus-off state.

This bit is cleared automatically when the related CANFD channel is in the bus-idle state or starts operating as a receiver node.

RECSTS bit (Channel Receive Status)

The RECSTS bit indicates whether the related CANFD channel is receiving a message.

This bit is set automatically when the related CANFD channel is operating as a receiver node.

This bit is cleared automatically when the related CANFD channel is in the bus-idle state or starts operating as a transmitter node.

COMSTS bit (Channel Communication Status)

The COMSTS bit indicates whether the related CANFD channel is ready for communication.

This bit is set automatically when the related CANFD channel is ready to perform communication following the detection of 11 consecutive recessive bits after exiting the Reset or Halt mode.

This bit is cleared automatically when the related CANFD channel is in CH_RESET or CD_HALT mode.

Note: This bit is 1 during bus-off state.

ESIF bit (Error State Indication Flag)

The ESIF bit is set when the ESI bit is sampled recessively for a reception CAN message without any error. When in Loopback or Mirror mode, the self-transmitted messages are considered reception messages.

If a set from the CANFD channel occurs simultaneously with a clear by a write access, then the bit is set.

This bit is cleared by writing 0 to it. This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

Note: This bit is not available in the classical CAN function.

REC[7:0] bits (Reception Error Count)

The REC[7:0] bits increment or decrement the counter value according to error status of the CANFD channel during reception, and display the value of the REC error counter.

The value in bus-off state is indeterminate.

These bits are cleared automatically when the CANFD module enters GL_RESET or the CANFD channel is in CH_RESET mode.

TEC[7:0] bits (Transmission Error Count)

The TEC[7:0] bits increment or decrement the counter value according to error status of the CANFD channel during transmission, and display the value of the TEC error counter.

Only write to these bits when in test mode and CANFD channel is in CH_HALT mode.

These bits are cleared automatically when CANFD module is in GL_RESET or CANFD channel is in CH_RESET mode.

41.2.5 CFDC0ERFL : Error Flag Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x000C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	CRCREG[14:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	ADERR	B0ERR	B1ERR	CERR	AERR	FERR	SERR	ALF	BLF	OVLF	BORF	BOEF	EPF	EWF	BEF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BEF	Bus Error Flag 0: Channel bus error not detected 1: Channel bus error detected	R/W
1	EWF	Error Warning Flag 0: Channel error warning not detected 1: Channel error warning detected	R/W
2	EPF	Error Passive Flag 0: Channel error passive not detected 1: Channel error passive detected	R/W
3	BOEF	Bus-Off Entry Flag 0: Channel bus-off entry not detected 1: Channel bus-off entry detected	R/W
4	BORF	Bus-Off Recovery Flag 0: Channel bus-off recovery not detected 1: Channel bus-off recovery detected	R/W

Bit	Symbol	Function	R/W
5	OVLF	Overload Flag 0: Channel overload not detected 1: Channel overload detected	R/W
6	BLF	Bus Lock Flag 0: Channel bus lock not detected 1: Channel bus lock detected	R/W
7	ALF	Arbitration Lost Flag 0: Channel arbitration lost not detected 1: Channel arbitration lost detected	R/W
8	SERR	Stuff Error 0: Channel stuff error not detected 1: Channel stuff error detected	R/W
9	FERR	Form Error 0: Channel form error not detected 1: Channel form error detected	R/W
10	AERR	Acknowledge Error 0: Channel acknowledge error not detected 1: Channel acknowledge error detected	R/W
11	CERR	CRC Error 0: Channel CRC error not detected 1: Channel CRC error detected	R/W
12	B1ERR	Bit 1 Error 0: Channel bit 1 error not detected 1: Channel bit 1 error detected	R/W
13	B0ERR	Bit 0 Error 0: Channel bit 0 error not detected 1: Channel bit 0 error detected	R/W
14	ADERR	Acknowledge Delimiter Error 0: Channel acknowledge delimiter error not detected 1: Channel acknowledge delimiter error detected	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
30:16	CRCREG[14:0]	CRC Register value These bits show the CRC value calculated for the CAN2.0 CAN frame.	R
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Channel Error Flag register shows the status of various error conditions detectable regardless of the setting of the related CAN Channel Error Interrupt Enable Register. It also shows the status of the various bus errors detectable by the CAN channel. Refer to the CAN specification (ISO 11898-1) to check when each error condition occurs.

For this register, only a single bit can be cleared by software. Do not use the bit clear instruction to clear the bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

Example in assembler language to clear the CFDC0ERFL.BEF bit:

mov.b #0x0FE, CFDC0ERFL ;

BEF bit (Bus Error Flag)

The BEF bit indicates a detection of a CAN channel bus error state, flagged by bits [14:8] in this register.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when a bus error is detected, and is cleared automatically when the related CANFD channel is in CH_RESET mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

EWF bit (Error Warning Flag)

The EWF bit indicates whether an error warning condition has been detected for the CAN channel.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when either TEC or REC exceeds 0x5F.

The setting of this bit only occurs when the TEC or REC initially exceeds 0x5F. Therefore, if the TEC or REC remains > 0x5F and the EWF bit is cleared by software, it is not set again until both the TEC and REC go below 0x60 and either TEC or REC crosses over again from a value 0x5F to a value > 0x5F.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

EPF bit (Error Passive Flag)

The EPF bit indicates a detection of a CAN channel error passive state.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when the CAN error state becomes error passive state.

The setting of this bit only occurs when the TEC or REC initially exceeds 0x7F. Therefore, if the TEC or REC remains > 0x7F and the bit is cleared by software, it is not set again until both the TEC and REC go below 0x80 and either TEC or REC crosses over again from a value ≤ 0x7F to a value > 0x7F.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

BOEF bit (Bus-Off Entry Flag)

The BOEF bit indicates a detection of a CAN channel bus-off entry state.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when the CAN error state enters the bus-off state.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If a set condition occurs simultaneously with a clear condition, then the bit is set.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

BORF bit (Bus-Off Recovery Flag)

The BORF bit indicates a detection of a CAN channel bus-off recovery state.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically if CAN channel recovers from bus-off state in the following conditions:

- When CFDC0CTR.BOM is 00b and normal recovery (11 consecutive recessive bits x 128 times detected) occurs
- When CFDC0CTR.BOM is 10b and normal recovery (11 consecutive recessive bits x 128 times detected) occurs
- When CFDC0CTR.BOM is 11b and normal recovery (11 consecutive recessive bits x 128 times detected) occurs.

The bit is not set if CAN channel recovers from bus-off state in the following conditions:

- When CAN Reset mode is requested
- When CFDC0CTR.RTBO is set to 1 (the CAN channel returns to error active)
- When CFDC0CTR.BOM is 01b
- When CFDC0CTR.BOM is 11b and a halt request is asserted before the CAN channel reaches the end of the bus-off state.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If a set condition occurs simultaneously with a clear condition, the flag is set.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

OVLF bit (Overload Flag)

The OVLF flag indicates a detection of a CAN channel overload state.

The OVLF bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when an overload condition is detected. If a set condition occurs simultaneously with a clear condition, then the bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

BLF bit (Bus Lock Flag)

The BLF bit indicates a detection of a CAN channel bus lock condition.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when 32 consecutive dominant bits are detected on the CAN bus while the CAN channel is in Operation mode.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

It is cleared automatically when the related CANFD channel is in CH_RESET mode.

ALF bit (Arbitration Lost Flag)

The ALF bit indicates a detection of a CAN channel bus arbitration lost condition.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

The bit is set automatically when an arbitration lost condition is detected on the CAN bus while the CAN channel is in Operation mode.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

SERR bit (Stuff Error)

The SERR bit indicates a detection of a CAN stuff error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a stuff error is detected. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

FERR bit (Form Error)

The FERR bit indicates a detection of a CAN form error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.

3. If yes, continue, else go back to step 1.

This bit is set automatically when a form error is detected. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

AERR bit (Acknowledge Error)

The AERR bit indicates a detection of a CAN acknowledge error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when an acknowledge error is detected. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION.

CERR bit (CRC Error)

The CERR bit indicates a detection of a CAN CRC error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a CRC error is detected. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

B1ERR bit (Bit 1 Error)

The B1ERR bit indicates a detection of a recessive bit error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a recessive bit error (expected recessive bit, sampled as dominant bit) is detected. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

B0ERR bit (Bit 0 Error)

The B0ERR bit indicates a detection of a dominant bit error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a dominant bit error (expected dominant bit, sampled as recessive bit) is detected. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

ADERR bit (Acknowledge Delimiter Error)

The ADERR bit indicates a detection of an acknowledge delimiter bit error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a form error is detected during the acknowledge delimiter state of frame transmission. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

CRCREG[14:0] bits (CRC Register value)

The CRCREG[14:0] bits read the calculated CRC value when CFDC0CTR.CTME bit is 1 for the channel.

If CFDC0CTR.CTME bit is 0, then these bits are always read as 0.

These bits show the CAN2.0 CRC value calculated by the CANFD channel logic when the CTME bit is enabled.

The CFDC0ERFL.CRCREG value is updated in the first bit of the CRC field of the CAN frame (reception and transmission).

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

41.2.6 CFDC0DCFG : Data Bitrate Configuration Register

This register is not available in the classical CAN function.

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0100

Bit position:	31	30	29	28	27–24	23	22	21	20	19–16	15	14	13	12–8	7–0
Bit field:	—	—	—	—	DSJW[3:0]	—	—	—	—	DTSEG2[3:0]	—	—	—	DTSEG1[4:0]	DBRP[7:0]
Value after reset:	0	0	0	0	0 0 0 0	0	0	0	0	0 0 0 0	0	0	0	0 0 0 0 0	0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
7:0	DBRP[7:0]	Channel Data Baud Rate Prescaler Data Baud Rate Prescaler division ratio	R/W
12:8	DTSEG1[4:0]	Timing Segment 1 0x00: Reserved 0x01: 2 Tq 0x02: 3 Tq 0x03: 4 Tq ⋮ 0x1E: 31 Tq 0x1F: 32 Tq	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
19:16	DTSEG2[3:0]	Timing Segment 2 0x0: Reserved 0x1: 2 Tq ⋮ 0xE: 15 Tq 0xF: 16 Tq	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
27:24	DSJW[3:0]	Resynchronization Jump Width 0x0: 1 Tq 0x1: 2 Tq ⋮ 0xF: 16 Tq	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Tq means time quantum.

The Data Bitrate Configuration Register configures the transmission/reception data baud rate parameters of the channels.

The channel of Classical CAN mode does not perform configuration of this register.

DBRP[7:0] bits (Channel Data Baud Rate Prescaler)

The DBRP[7:0] bits define the peripheral bus clock periods contained in a time quantum.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

DTSEG1[4:0] bits (Timing Segment 1)

The DTSEG1[4:0] bits set the segment TSEG1 to compensate for edges on the CAN bus with a positive phase error. A value from 2 to 32 time quanta can be set.

The DTSEG1[4:0] bits are also used to set the propagation segment.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode. Do not write any other value to these bits. See section 41.4.1.2. CAN Bit Timing for more details.

DTSEG2[3:0] bits (Timing Segment 2)

The DTSEG2[3:0] bits set the segment TSEG2 to compensate for edges on the CAN bus with a negative phase error. A value from 2 to 16 time quanta can be set.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode. Do not write any other value to these bits.

DSJW[3:0] bits (Resynchronization Jump Width)

The DSJW[3:0] bits set the synchronization jump width. A value from 1 to 16 time quanta can be set.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

41.2.7 CFDC0FDCFG : CANFD Configuration Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0104

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	CLOE	REFE	FDOE	—	—	—	—	TDCO[7:0]							
Value after reset:	0	0/1*1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	ESIC	TDCE	TDCOC	—	—	—	—	—	EOCCFG[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	EOCCFG[2:0]	Error Occurrence Counter Configuration 0 0 0: All transmitter or receiver CAN frames 0 0 1: All transmitter CAN frames 0 1 0: All receiver CAN frames 0 1 1: Reserved 1 0 0: Only transmitter or receiver CANFD data-phase (fast bits) 1 0 1: Only transmitter CANFD data-phase (fast bits) 1 1 0: Only receiver CANFD data-phase (fast bits) 1 1 1: Reserved	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	TDCOC*2	Transceiver Delay Compensation Offset Configuration 0: Measured + offset 1: Offset-only	R/W
9	TDCE*2	Transceiver Delay Compensation Enable 0: Transceiver delay compensation disabled 1: Transceiver delay compensation enabled	R/W
10	ESIC*2	Error State Indication Configuration 0: The ESI bit in the frame represents the error state of the node itself 1: The ESI bit in the frame represents the error state of the message buffer if the node itself is not in error passive. If the node is in error passive, then the ESI bit is driven by the node itself.	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
23:16	TDCO[7:0]*2	Transceiver Delay Compensation Offset	R/W
27:24	—	These bits are read as 0. The write value should be 0.	R/W
28	FDOE*2	FD-Only Enable 0: FD-only mode disabled 1: FD-only mode enabled	R/W

Bit	Symbol	Function	R/W
29	REFE	RX Edge Filter Enable 0: RX edge filter disabled 1: RX edge filter enabled	R/W
30	CLOE[*2] [*3]	Classical CAN Enable 0: Classical CAN mode disabled 1: Classical CAN mode enabled	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. The value after reset is 0 for products that support the CAN-FD protocol, and 1 for products that support only classical CAN protocol.

Note 2. These bits are not available in the classical CAN function.

Note 3. This bit can only be written for products that support the CAN-FD protocol. For products that support only classical CAN protocol, this bit is reserved and fixed to 1.

The CANFD Configuration Register configures which communication direction (transmitter/receiver) errors are counted.

EOCCFG[2:0] bits (Error Occurrence Counter Configuration)

The EOCCFG[2:0] bits select which type of CAN frame configuration and direction, including protocol errors are counted.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

TDCOC bit (Transceiver Delay Compensation Offset Configuration)[*1]

The TDCOC bit selects which offset is used when defining the position of the secondary sample point (SSP) for the CANFD channel. If the bit is set to 0, the position of the SSP is the measured transceiver delay plus the fixed offset. If the bit is 1, the position of the SSP is defined only by the offset.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode. Do not set this bit when in Classical CAN mode.

TDCE bit (Transceiver Delay Compensation Enable)[*1]

The TDCE bit enables the transceiver delay compensation for the CANFD channel.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode. Do not set this bit when in Classical CAN mode.

ESIC bit (Error State Indication Configuration)[*1]

The ESIC bit controls the transmission of either the ESI flag information or the message of ESI flag information (CFDCFFDCSTS.CFESI or CFDTMFDCTRb.TMESI).

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode. Do not set this bit when in Classical CAN mode.

TDCO[7:0] bits (Transceiver Delay Compensation Offset)[*1]

The TDCO[7:0] bits set the secondary sample point offset. How this value is used, depends on the CFDC0FDCFG.TDCOC setting.

If CFDC0FDCFG.TDCOC = 0, the transceiver delay compensation result is equal to the Trv_Delay (measured delay) + the value in CFDC0FDCFG.TDCO, rounded down to the nearest integer number of time quanta. Otherwise, the result is equal to the value in CFDC0FDCFG.TDCO. See section 41.4.1.5. Transmitter Delay Compensation for details on how CFDC0FDCFG.TDCO is used.

The actual offset value is interpreted as TDCO + 1. For example, if 4 is set in TDCO, the offset is 5 clock cycles. Clock cycle is 1 cycle of CAN channel DLL clock.

Do not write to the TDCO[7:0] bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode. Do not set this bit when in Classical CAN mode.

FDOE bit (FD-Only Enable)*1

The FDOE bit enables the reception and transmission of CANFD-only frames. If enabled, communication in Classical CAN frame format is disabled. Transmission of Classical CAN frames is not possible because the FDF bit of the message buffer is a don't care (CFDCFFDCSTS.CFFDF/CFDTMFDCTRb.TMFDF).

If messages with Classical CAN frame format are received, the protocol controller treats them as invalid frames and response with error frames. When a Classical CAN frame is configured for transmitting, the FDF bit is sent as recessive, therefore an FD frame is sent. If the data length code (DLC) is configured of greater than 8 bytes, the remaining data bytes are padded with 0xCC.

The FDOE bit cannot be written in CH_OPERATION, CH_HALT or CH_SLEEP mode.

Do not set CFDC0FDCFG.FDOE and CFDC0FDCFG.CLOE simultaneously.

REFE bit (RX Edge Filter Enable)

The REFE bit enables the RX edge filter during the IDLE detection (bus integration). When the bit is enabled, two consecutive dominant time quanta are required to detect a synchronization edge.

The REFE bit cannot be written in CH_OPERATION, CH_HALT and CH_SLEEP mode. Do not set this bit when in Classical CAN mode.

CLOE bit (Classical CAN Enable)*1

The CLOE bit enables the Classical CAN mode. If this bit is 1, the protocol controller can only send classical frames and response with a form or CRC error on FD frames.

Do not set CFDC0FDCFG.CLOE and CFDC0FDCFG.FDOE simultaneously.

CFDC0FDCFG.CLOE	CFDC0FDCFG.FDOE	Channel mode
0	0	CANFD mode
0	1	FD-only mode
1	0	Classical CAN mode
1	1	Reserved

The CANFD mode is available only for CANFD supported product.

Do not write to this bit in CH_OPERATION, CH_HALT or CH_SLEEP mode.

Only write to these bits when the CANFD channel is in CH_RESET mode.

Note 1. These bits are not available in the classical CAN function.

41.2.8 CFDC0FDCTR : CANFD Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0108

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOCCLR	EOCCLR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EOCCLR	Error Occurrence Counter Clear 0: No error occurrence counter clear 1: Clear error occurrence counter	R/W
1	SOCCLR	Successful Occurrence Counter Clear 0: No successful occurrence counter clear 1: Clear successful occurrence counter	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The CANFD Control Register controls the error and successful occurrence counters.

EOCCLR bit (Error Occurrence Counter Clear)

The EOCCLR bit is used to clear the error occurrence counter.

Do not write to this bit in CH_SLEEP or CH_RESET mode. The read value is always 0.

This bit is cleared automatically by the CANFD module logic and when the related CANFD channel is in CH_RESET mode.

SOCCLR bit (Successful Occurrence Counter Clear)

The SOCCLR bit is used to clear the successful occurrence counter.

Do not write to this bit in CH_SLEEP or CH_RESET mode. The read value is always 0.

This bit is cleared automatically by the CANFD module logic and when the related CANFD channel is in CH_RESET mode.

41.2.9 CFDC0FDSTS : CANFD Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x010C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SOC[7:0]								EOC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TDCVF	—	—	—	—	—	SOCO	EOCO	TDCR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TDCR[7:0]*1	Transceiver Delay Compensation Result	R
8	EOCO	Error Occurrence Counter Overflow 0: Error occurrence counter has not overflowed 1: Error occurrence counter has overflowed	R/W
9	SOCO	Successful Occurrence Counter Overflow 0: Successful occurrence counter has not overflowed 1: Successful occurrence counter has overflowed	R/W
14:10	—	These bits are read as 0. The write value should be 0.	R/W
15	TDCVF*1	Transceiver Delay Compensation Violation Flag 0: Transceiver delay compensation violation has not occurred 1: Transceiver delay compensation violation has occurred	R/W
23:16	EOC[7:0]	Error Occurrence Counter These bits show the error occurrence counter value.	R

Bit	Symbol	Function	R/W
31:24	SOC[7:0]	Successful occurrence counter These bits show the successful occurrence counter value.	R

Note: S-TYPE-3, P-TYPE-3

Note 1. These bits are not available in the classical CAN function.

The CANFD Status Register indicates the transceiver compensation delay result and its related FIFO message lost status.

TDCR[7:0] bits (Transceiver Delay Compensation Result)

The TDCR[7:0] bits are set when the transceiver delay has been measured.

The measured delay is a multiple of the CAN channel DLL clock. The result depends on the CFDC0FDCFG.TDCOC configuration and the offset value in CFDC0FDCFG.TDCO. See section 41.4.1.5. Transmitter Delay Compensation for details on how this value is derived.

The TDCR[7:0] bits are updated at the falling edge between FDF and the RES bit when CFDC0FDCFG.TDCOC = 0 and the transceiver delay compensation is enabled (CFDC0FDCFG.TDCE = 1).

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

Note: These bits are not available in the classical CAN function.

EOCO bit (Error Occurrence Counter Overflow)

The EOCO bit indicates whether the related CAN channel error occurrence counter has overflowed. This bit is cleared by writing 0 to it. Writing 1 has no effect.

This bit is set automatically when CFDC0FDSTS.EOC is 0xFF and a CAN bus error is detected based on the configuration defined in CFDC0FDCFG.EOCCFG.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

SOCO bit (Successful Occurrence Counter Overflow)

The SOCO bit indicates whether the related CAN channel successful occurrence counter has overflowed. This bit is cleared by writing 0 to it. Writing 1 has no effect.

This bit is set automatically when CFDC0FDSTS.SOC is 0xFF and a successful message reception or successful message transmission occurs.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

Write to this bit only when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

TDCVF bit (Transceiver Delay Compensation Violation Flag)

The CANFD module captures internally the transmitted data bit-by-bit. This data is then compared against the received CAN bus level which is delayed by the transceiver loop delay.

The transceiver delay has some variations depending on the physical parameters such as temperature. The result bit CFDC0FDSTS.TDCR is updated by each message. However, temporary maximum delay violation could be missed. Therefore, the TDCVF bit captures this violation.

This bit is cleared by writing 0 to it. Writing 1 has no effect.

This bit is set automatically when the transceiver delay compensation is greater than the maximum delay compensation (6 data bit times - 2 clk_dlc) and the internal bit is overrun.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

Note: This bit is not available in the classical CAN function.

EOC[7:0] bits (Error Occurrence Counter)

The EOC[7:0] bits are used together with the SOC[7:0] bits to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing the reduced payload bit length have significant higher error rates compared to other messages.

This higher error rate can be detected depending on the configuration of the CFDC0FDCFG.EOCCFG bits.

The EOC[7:0] bits are set only by CANFD module logic. These bits are cleared by writing 1 to CFDC0FDCTR.EOCCLR. Writing any other value has no effect.

These bits are updated when an error occurs, according to the configuration of the CFDC0FDCFG.EOCCFG bits. When the counter reaches the value of 0xFF, the update stops.

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

SOC[7:0] bits (Successful occurrence counter)

The SOC[7:0] bits are used together with the EOC[7:0] bits to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing the reduced payload bit length have significant higher error rates compared to other messages.

The SOC[7:0] bits are set only by CANFD module logic. Writing any other value has no effect.

These bits are updated when the occurrence of any error-free messages on the bus is detected through reception or transmission. When the counter reaches the value of 0xFF, the update stops.

Note: In Loopback mode, the counter is incremented twice.

These bits are cleared by writing 1 to CFDC0FDCTR.SOCCLR.

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

41.2.10 CFDC0FDCRC : CANFD CRC Register

This register is not available in the classical CAN function.

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0110

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	SCNT[3:0]				—	—	—	CRCREG[20:0]																				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
20:0	CRCREG[20:0]	CRC Register value These bits show the CRC value calculated for the CANFD frame.	R
23:21	—	These bits are read as 0.	R
27:24	SCNT[3:0]	Stuff bit count These bits shows the stuff bit count (mod 8) for the CANFD frame.	R
31:28	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The CANFD CRC Register holds the CRC value calculated for the CANFD frame.

CRCREG[20:0] bits (CRC Register value)

The CRCREG[20:0] bits contain the CRC value calculated by the CANFD channel logic when the CFDC0CTR.CTME bit is enabled.

The CFDC0FDCRC.CRCREG value is updated in the first bit of the CRC field of the CANFD frame (reception and transmission).

When the CFDC0CTR.CTME bit is 0, the CRCREG[20:0] bits are always read as 0.

When bit 17th of the CRC field is used, CRCREG[20:17] are always read as 0.

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

SCNT[3:0] bits (Stuff bit count)

The SCNT[3:0] bits contain the stuff count value of the CANFD frame. These bits indicate the number of inserted stuff bits (modulo 8, Gray coded) for a CANFD frame when the CFDC0CTR.CTME bit is enabled in CFDC0FDCRC.SCNT[3:1]. SCNT[0] is the parity bit.

When the CFDC0CTR.CTME bit is 0, the SCNT[3:0] bits are always read as 0.

The SCNT value is updated in the first bit of CRC field of the CANFD frame (reception and transmission).

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

41.2.11 CFDGCFG : Global Configuration Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ITRCP[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TSSS	TSP[3:0]				—	—	CMPOC	DCS	MME	DRE	DCE	TPRI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TPRI	Transmission Priority 0: ID priority 1: Message buffer number priority	R/W
1	DCE	DLC Check Enable 0: DLC check disabled 1: DLC check enabled	R/W
2	DRE	DLC Replacement Enable 0: DLC replacement disabled 1: DLC replacement enabled	R/W
3	MME	Mirror Mode Enable 0: Mirror mode disabled 1: Mirror mode enabled	R/W
4	DCS	Data Link Controller Clock Select 0: CANFD core clock (CANFDCLK) 1: External oscillator clock (CANMCLK)	R/W
5	CMPOC*1	CANFD Message Payload Overflow Configuration 0: Message is rejected 1: Message payload is cut to fit to configured message size	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
11:8	TSP[3:0]	Timestamp Prescaler 0x0: Timestamp prescaler = 1 0x1: Timestamp prescaler = 2 0x2: Timestamp prescaler = 4 0x3: Timestamp prescaler = 8 ⋮ 0xD: Timestamp prescaler = 8192 0xE: Timestamp prescaler = 16384 0xF: Timestamp prescaler = 32768	R/W
12	TSSS	Timestamp Source Select 0: Source clock for timestamp counter is peripheral clock 1: Source clock for timestamp counter is bit time clock	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
31:16	ITRCP[15:0]	Interval Timer Reference Clock Prescaler FIFO interval timer prescaler value	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. This bit is not available in the classical CAN function.

The Global Configuration Register is used to select the transmission priority to be used for all the TX message buffers and the clock source for the CAN protocol engine of CAN channel. The CFDGCFG register is also used to select the source for the timestamp clock and to configure the frequency for the timestamp clock and interval timer reference clock.

TPRI bit (Transmission Priority)

The TPRI bit selects the transmission priority for CAN channel.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CANFD module is in GL_RESET mode.

Message buffer number priority should not be used together with TX queue transmission.

DCE bit (DLC Check Enable)

The DCE bit enables data length code (DLC) check for CAN channel.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CANFD module is in GL_RESET mode.

DRE bit (DLC Replacement Enable)

When the DRE bit is 1 and the DCE is 1, the CANFD stores the configured value (CFDGAFLP0r.GAFLDLC) of the DLC in the destination RX message buffer or FIFO buffer if the DLC check passes. Otherwise, the DLC value in the destination RX message buffer or FIFO buffer is unchanged.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CANFD module is in GL_RESET mode.

MME bit (Mirror Mode Enable)

The MME bit enables the Mirror mode for CAN channel.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CANFD module is in GL_RESET mode.

DCS bit (Data Link Controller Clock Select)

The DCS bit selects CANFDCLK or CANMCLK as the clock source for CAN communication.

Do not write to this bit in GL_SLEEP or GL_OPERATION mode. Only write to this bit when CANFD module is in GL_RESET mode.

CMPOC bit (CANFD Message Payload Overflow Configuration)

The CMPOC bit controls the message payload acceptance mechanism when the received payload is higher than the message buffer payload size CFDRMNB.RMPLS, CFDRFCCa.RFPLS, and CFDCFCC.CFPLS. The received message payload is always compared with the available message payload size in the message buffer.

Do not write to this bit in GL_SLEEP or GL_OPERATION mode. Only write to this bit when CANFD module is in GL_RESET mode.

When this bit is set and payload overflow occurs, the DLC value is stored in the RX message buffer or FIFO buffer unchanged.

Note: This bit is not available in the classical CAN function.

TSP[3:0] bits (Timestamp Prescaler)

The value configured in the TSP[3:0] bits defines the period of the clock source used for the timestamp counter.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CANFD module is in GL_RESET mode.

TSSS bit (Timestamp Source Select)

The TSSS bit allows the selection of the clock source for the timestamp counter.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CANFD module is in GL_RESET mode. Additionally, do not set this bit to 1 when CANFD communication is used.*1

Note: The bit time clock varies depending on the nominal and data rate bit configuration.

Note 1. This feature is not available in the classical CAN function.

ITRCP[15:0] bits (Interval Timer Reference Clock Prescaler)

The ITRCP[15:0] bits allow the definition of a reference clock for the FIFO interval timer source clock.

When these bits are 0x0000, the timer is disabled.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CANFD module is in GL_RESET mode.

41.2.12 CFDGCTR : Global Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0018

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	CMPOFIE	THLEIE	MEIE	DEIE	—	—	—	—	—	GSLPR	GMDC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Function	R/W
1:0	GMDC[1:0]	Global Mode Control 0 0: Global operation mode request 0 1: Global reset mode request 1 0: Global halt mode request 1 1: Keep current value	R/W
2	GSLPR	Global Sleep Request 0: Global sleep request disabled 1: Global sleep request enabled	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	DEIE	DLC Check Interrupt Enable 0: DLC check interrupt disabled 1: DLC check interrupt enabled	R/W
9	MEIE	Message Lost Error Interrupt Enable 0: Message lost error interrupt disabled 1: Message lost error interrupt enabled	R/W
10	THLEIE	TX History List Entry Lost Interrupt Enable 0: TX history list entry lost interrupt disabled 1: TX history list entry lost interrupt enabled	R/W

Bit	Symbol	Function	R/W
11	CMPOFIE*1	CANFD Message Payload Overflow Flag Interrupt Enable 0: CANFD message payload overflow flag interrupt disabled 1: CANFD message payload overflow flag interrupt enabled	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
16	TSRST	Timestamp Reset 0: Timestamp not reset 1: Timestamp reset	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. This bit is not available in the classical CAN function.

The Global Control Register controls the global mode of the CANFD module and the timestamp function. The register also enables and disables the global error interrupts.

GMDC bits (Global Mode Control)

The GMDC bits can be used to configure the modes for the CANFD module. Additionally, if CFDGCTR.GSLPR bit is 1 when the CANFD module is in Reset mode, the CANFD module enters Global Sleep mode.

Setting the GMDC bits to 11b has no effect. Mode transition is described in detail in section 41.3.2. Global Modes.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

GSLPR bit (Global Sleep Request)

The GSLPR bit globally selects the sleep request for CANFD module including CAN channels. Channel sleep request is set automatically for channels.

Only write to this bit when the CANFD module is in GL_RESET or GL_SLEEP mode.

DEIE bit (DLC Check Interrupt Enable)

When the DEIE bit is 1, an interrupt is generated if a DLC error is detected in the received frames.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

MEIE bit (Message Lost Error Interrupt Enable)

When the MEIE bit is 1, an interrupt is generated if a message lost condition occurs.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

THLEIE bit (TX History List Entry Lost Interrupt Enable)

When the THLEIE bit is 1, an interrupt is generated if a TX history list entry lost condition occurs.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

CMPOFIE bit (CANFD Message Payload Overflow Flag Interrupt Enable)

When the CMPOFIE bit is 1, an interrupt is generated when a CANFD message payload overflow condition occurs.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

Note: This bit is not available in the classical CAN function

TSRST bit (Timestamp Reset)

When the TSRST bit is 1, the Global Timestamp Register is reset to 0x0000.

Do not write to this bit when the CANFD module is in GL_SLEEP or GL_RESET mode.

Read value is always 0.

This bit is cleared automatically by the CANFD module logic.

41.2.13 CFDGSTS : Global Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x001C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	GRAMINIT	GSLPSTS	GHLTSTS	GRSTSTS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

Bit	Symbol	Function	R/W
0	GRSTSTS	Global Reset Status 0: Not in Reset mode 1: In Reset mode	R
1	GHLTSTS	Global Halt Status 0: Not in Halt mode 1: In Halt mode	R
2	GSLPSTS	Global Sleep Status 0: Not in Sleep mode 1: In Sleep mode	R
3	GRAMINIT	Global RAM Initialization 0: RAM initialization is complete 1: RAM initialization is ongoing	R
31:4	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The Global Status Register indicates the global status of the CANFD module.

GRSTSTS bit (Global Reset Status)

The GRSTSTS bit indicates the status of Global CANFD module Reset mode.

This bit is set automatically when the CANFD module enters GL_RESET mode. When the mode changes from GL_RESET mode to GL_SLEEP mode, this bit remains set.

This bit is cleared automatically when the CANFD module exits the GL_RESET mode.

GHLTSTS bit (Global Halt Status)

The GHLTSTS bit indicates the status of Global CANFD module Halt mode.

This bit is set automatically when the CANFD module enters GL_HALT mode.

This bit is cleared automatically when the CANFD module exits the GL_HALT mode.

GSLPSTS bit (Global Sleep Status)

The GSLPSTS bit indicates the status of Global CANFD module Sleep mode.

This bit is set automatically when the CANFD module enters GL_SLEEP mode.

This bit is cleared automatically when the CANFD module exits the GL_SLEEP mode.

GRAMINIT bit (Global RAM Initialization)

The GRAMINIT bit indicates the status of Global CANFD module RAM initialization.

This bit is set automatically when the CANFD module enters GL_SLEEP mode after a hardware reset.

This bit is cleared automatically when the CANFD module completed RAM initialization.

This bit is cleared when the test_mode input port is set to 1.

41.2.14 CFDGERFL : Global Error Flag Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0020

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EEF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	CMPOF	THLES	MES	DEF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DEF	DLC Error Flag 0: DLC error not detected 1: DLC error detected	R/W
1	MES	Message Lost Error Status 0: Message lost error not detected 1: Message lost error detected	R
2	THLES	TX History List Entry Lost Error Status 0: TX history list entry lost error not detected 1: TX history list entry lost error detected	R
3	CMPOF*1	CANFD Message Payload Overflow Flag 0: CANFD message payload overflow not detected 1: CANFD message payload overflow detected	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W
16	EEF0	ECC Error Flag 0: ECC error not detected during TX-SCAN 1: ECC error detected during TX-SCAN	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. This bit is not available in the classical CAN function.

The Global Error Flag register indicates the detection of global errors.

DEF bit (DLC Error Flag)

The DEF bit indicates the error status of the DLC.

Do not write to this bit when the CANFD module is in GL_SLEEP or GL_RESET mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when a DLC error is detected in a received frame.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set

The bit is cleared by writing 0 to it.

This bit is cleared automatically in GL_RESET mode.

MES bit (Message Lost Error Status)

The MES bit indicates status of the message lost error.

This bit is set automatically when a FIFO message lost error is detected.

This bit is cleared automatically when:

- All FIFO message lost flags are cleared
- The CANFD module is in GL_RESET mode.

THLES bit (TX History List Entry Lost Error Status)

The THLES bit indicates status of the TX history list entry lost error.

This bit is set automatically when a TX history list entry lost error is detected.

This bit is cleared automatically when:

- All TX history list entry lost flags are cleared
- The CANFD module is in GL_RESET mode.

CMPOF bit (CANFD Message Payload Overflow Flag)

The CMPOF bit is set automatically when a CANFD message payload overflow is detected on at least one channel.

Do not write to this bit when the CANFD module is in GL_SLEEP or GL_RESET mode.

This bit is cleared by writing 0 to it. Writing 1 to this bit has no effect.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is cleared automatically in GL_RESET mode.

Note: This bit is not available in the classical CAN function

EEF0 bit (ECC Error Flag)

The EEF0 bit specifies whether an ECC error has occurred.

Do not write to this bit when the CANFD module is in GL_SLEEP or GL_RESET mode. Writing 1 to this bit has no effect.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

The bit is cleared by writing 0 to it. This bit is cleared automatically in GL_RESET mode.

41.2.15 CFDGTINTSTS : Global TX Interrupt Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00A4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	THIF0	CFTIF0	TQIF0	TAI0	TSIF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TSIF0	TX Successful Interrupt Flag 0: Channel n TX Successful Interrupt flag not set 1: Channel n TX Successful Interrupt flag set	R
1	TAI0	TX Abort Interrupt Flag 0: Channel n TX Abort Interrupt flag not set 1: Channel n TX Abort Interrupt flag set	R

Bit	Symbol	Function	R/W
2	TQIF0	TX Queue Interrupt Flag 0: Channel n TX Queue Interrupt flag not set 1: Channel n TX Queue Interrupt flag set	R
3	CFTIF0	COM FIFO TX Mode Interrupt Flag 0: Channel n COM FIFO TX Mode Interrupt flag not set 1: Channel n COM FIFO TX Mode Interrupt flag set	R
4	THIF0	TX History List Interrupt 0: Channel n TX History List Interrupt flag not set 1: Channel n TX History List Interrupt flag set	R
31:5	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The Global TX Interrupt Status register indicates the detection of transmit specific interrupts.

TSIF0 bit (TX Successful Interrupt Flag)

The TSIF0 bit is set to 1 when the TX Successful Interrupt flag of the related channel is set (when the interrupt is enabled). This bit is cleared automatically:

- When the related TX MB Result Status bits are cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

TAI0 bit (TX Abort Interrupt Flag)

The TAIF0 bit is set to 1 when the TX Abort Interrupt flag of the related channel is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX MB Result Status bits are cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

TQIF0 bit (TX Queue Interrupt Flag)

The TQIF0 bit is set to 1 when the TX Queue Interrupt flag of the related channel is set (when the interrupt is enabled). This bit is cleared automatically:

- When the related TX Queue Interrupt flag is cleared (when the interrupt is enable disabled)
- When in GL_RESET or CH_RESET mode.

CFTIF0 bit (COM FIFO TX Mode Interrupt Flag)

The CFTIF0 bit is set to 1 when the related COM TX FIFO Mode Interrupt flag (CFDCFSTS.CFTXIF) is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related COM TX FIFO Mode Interrupt flag (CFDCFSTS.CFTXIF) is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

THIF0 bit (TX History List Interrupt)

The THIF0 bit is set to 1 when the related TX History List Interrupt flag (CFDTHLSTS.THLIF) is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX History List Interrupt flag (CFDTHLSTS.THLIF) is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

41.2.16 CFDGTSC : Global Timestamp Counter Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0024

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TS[15:0]	Timestamp value	R
31:16	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The Global Timestamp Counter register stores the timestamp based on the selected configuration.

TS[15:0] bits (Timestamp value)

The Timestamp value is stored in the Global Timestamp Counter register based on the configuration of TSSS, TSBTCS and TSP. The accuracy of the timestamp counter cannot be guaranteed when transitioning to halt state.

The Timestamp value is stored in this register based on the configuration of TSSS, TSBTCS and TSP.

Do not write to bits TS[15:0] when the CANFD module is in GL_RESET or GL_SLEEP mode.

The TS[15:0] bits are cleared automatically in GL_RESET mode.

41.2.17 CFDGAFLECTR : Global Acceptance Filter List Entry Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0028

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	AFLDAE	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	AFLDAE	Acceptance Filter List Data Access Enable 0: Acceptance Filter List data access disabled 1: Acceptance Filter List data access enabled	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The Global Acceptance Filter List Entry Control Register is used to select the Global Acceptance Filter List page for reading or writing entries into the Global Acceptance Filter List.

AFLDAE bit (Acceptance Filter List Data Access Enable)

The AFLDAE bit prevents write access to the Acceptance Filter List when cleared after configuration of the Acceptance Filter List.

Data can be read from the Acceptance Filter List independent of the status of this bit.

Do not write to this bit when the CANFD module is in GL_SLEEP mode. Set this bit to enable write access for the Acceptance Filter List.

41.2.18 CFDGAFLCFG : Global Acceptance Filter List Configuration Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x002C

Bit position:	31											20				16																0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	RNC0[4:0]					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
20:16	RNC0[4:0]	Rule Number Number of rules dedicated	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The Global Acceptance Filter List Configuration Register is used to define the number of rules for entries in the Acceptance Filter List.

The total number of available entries in the Acceptance Filter List is 16.

RNC0[4:0] bits (Rule Number)

The RNC0[4:0] bits define the number of rules in the Acceptance Filter List.

Only write to these bits when the CANFD module is in GL_RESET mode. These bits can set to 5 bits for 16 rules.

41.2.19 CFDGAFLIDr : Global Acceptance Filter List ID Registers (r = 1 to 16)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0120 + 0x0010 × (r - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	GAFLIDE	GAFLRTR	GAFLLB	GAFLID[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	GAFLID[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
28:0	GAFLID[28:0]	Global Acceptance Filter List Entry ID Field ID part of the Global Acceptance Filter List entry	R/W
29	GAFLLB	Global Acceptance Filter List Entry Loopback Configuration 0: Global Acceptance Filter List entry ID for acceptance filtering with attribute RX 1: Global Acceptance Filter List entry ID for acceptance filtering with attribute TX	R/W
30	GAFLRTR	Global Acceptance Filter List Entry RTR Field 0: Data frame 1: Remote frame	R/W

Bit	Symbol	Function	R/W
31	GAFLIDE	Global Acceptance Filter List Entry IDE Field 0: Standard identifier of rule entry ID is valid for acceptance filtering 1: Extended identifier of rule entry ID is valid for acceptance filtering	R/W

Note: S-TYPE-3, P-TYPE-3

The Global Acceptance Filter List ID Registers are used to configure the ID field for the rules of entries in the Global Acceptance Filter List.

GAFLID[28:0] bits (Global Acceptance Filter List Entry ID Field)

The GAFLID[28:0] bits represent the CAN identifier (ID) field of each entry in the Global Acceptance Filter List.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

GAFLLB bit (Global Acceptance Filter List Entry Loopback Configuration)

The GAFLLB bit selects whether entry in the Global Acceptance Filter List gets the attribute RX or TX.

This attribute determines the validity of the entry in Mirror mode, Loopback test mode, and during standard (non-loopback) reception. See section 41.5.5. Loopback Modes for detailed description of the validity of the Global Acceptance Filter List entry depending on transmitter/receiver case, the type of loopback mode, and RX/TX attribute.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode.

GAFLRTR bit (Global Acceptance Filter List Entry RTR Field)

The GAFLRTR bit allows the configuration of the specified frame format (data frame or remote frame) for each entry of the Global Acceptance Filter List. For each rule entry in a CAN channel, the acceptance filter process compares this bit against the RTR bit of the received CAN message.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode.

GAFLIDE bit (Global Acceptance Filter List Entry IDE Field)

The GAFLIDE bit allows the configuration of the ID format (standard ID or extended ID) for each entry in the Global Acceptance Filter List. For each rule entry in a CAN channel, the acceptance filter process compares this bit against the IDE bit of the received CAN message.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode.

41.2.20 CFDGAFLMr : Global Acceptance Filter List Mask Registers (r = 1 to 16)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0124 + 0x0010 × (r - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	GAFLIDEM	GAFLRTRM	GAFLIFL1	GAFLIDM[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	GAFLIDM[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
28:0	GAFLIDM[28:0]	Global Acceptance Filter List ID Mask Field Global Acceptance Filter List Mask field bits for ID field	R/W
29	GAFLIFL1	Global Acceptance Filter List Information Label 1 Global Acceptance Filter List information label bit 1	R/W
30	GAFLRTRM	Global Acceptance Filter List Entry RTR Mask 0: RTR bit is not used for ID matching 1: RTR bit is used for ID matching	R/W
31	GAFLIDEM	Global Acceptance Filter List IDE Mask 0: IDE bit is not used for ID matching 1: IDE bit is used for ID matching	R/W

Note: S-TYPE-3, P-TYPE-3

The Global Acceptance Filter List Mask Registers are used to configure the Mask field of each rule for entries in the Global Acceptance Filter List.

GAFLIDM[28:0] bits (Global Acceptance Filter List ID Mask Field)

GAFLIDM[28:0] bits are the filter mask bits for the related bits in the CAN Identifier field of each Global Acceptance Filter List entry.

0	Corresponding STD-ID/EXT-ID bit is not used for ID matching
1	Corresponding STD-ID/EXT-ID bit is used for ID matching

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

GAFLIFL1 bit (Global Acceptance Filter List Information Label 1)

The GAFLIFL1 bit allows the configuration of a 2-bit information label to be attached to a received message accepted by the associated entry in the Global Acceptance Filter List. This bit is an MSB bit of an information label.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode.

This bit is stored in the Information Label Field [1] (CFDRMFDSTSb.RMIFL [1], CFDRFFDSTSb.RFIFL [1], CFDCFFDCSTS.CFIFL [1]) of the storage location of an incoming message.

GAFLRTRM bit (Global Acceptance Filter List Entry RTR Mask)

The GAFLRTRM bit allows the configuration of the RTR mask bit for each entry in the Global Acceptance Filter List.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode.

GAFLIDEM bit (Global Acceptance Filter List IDE Mask)

The GAFLIDEM bit allows the configuration of the IDE mask bit for each entry in the Global Acceptance Filter List.

When the IDE mask bit is 0, the ID comparison depends on the received IDE bit.

If the received IDE bit is 0, the STD-ID comparison takes place.

If the received IDE bit is 1, the EXT-ID comparison takes place.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CANFD channel is in CH_RESET or CH_HALT mode.

41.2.21 CFDGAFLP0r : Global Acceptance Filter List Pointer 0 Registers (r = 1 to 16)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0128 + 0x0010 × (r - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	GAFLPTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	GAFLRMV	—	—	GAFLRMDP[4:0]					GAFLIFL0	—	—	—	GAFLDLC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	GAFLDLC[3:0]	Global Acceptance Filter List DLC Field Minimum number of data bytes in a data frame required for acceptance	R/W
6:4	—	These bits are read as 0. The write value should be 0.	R/W
7	GAFLIFL0	Global Acceptance Filter List Information Label 0	R/W
12:8	GAFLRMDP[4:0]	Global Acceptance Filter List RX Message Buffer Direction Pointer RX message buffer number for storage of received messages	R/W
14:13	—	These bits are read as 0. The write value should be 0.	R/W
15	GAFLRMV	Global Acceptance Filter List RX Message Buffer Valid 0: Single message buffer direction pointer is invalid 1: Single message buffer direction pointer is valid	R/W
31:16	GAFLPTR[15:0]	Global Acceptance Filter List Pointer	R/W

Note: S-TYPE-3, P-TYPE-3

The Global Acceptance Filter List Pointer 0 Registers are used to configure the data length code (DLC), software pointer, single message buffer select, and message buffer direction pointer for each rule entry in the Global Acceptance Filter List.

GAFLDLC[3:0] bits (Global Acceptance Filter List DLC Field)

The GAFLDLC[3:0] bits allow the configuration of a minimum data length code (DLC) value for a message to be accepted by the associated entry in the Global Acceptance Filter List (automatic DLC filter function).

DLC filter process is only passed if the DLC value of the message accepted by an entry in the Global Acceptance Filter List is equal to or higher than the DLC value configured for this associated Global Acceptance Filter List entry. Automatic DLC filter function is disabled for the corresponding rule entry when this field is set to 0.

Table 41.3 shows DLC value that can be configured.

Table 41.3 Configuration of DLC value (1 of 2)

Format	DLC[3]	DLC[2]	DLC[1]	DLC[0]	Description
CAN and CANFD	0	0	0	0	DLC of received message = 0 or more (DLC filter check is disabled)
CAN and CANFD	0	0	0	1	DLC of received message = 1 or more
CAN and CANFD	0	0	1	0	DLC of received message = 2 or more
CAN and CANFD	0	0	1	1	DLC of received message = 3 or more
CAN and CANFD	0	1	0	0	DLC of received message = 4 or more
CAN and CANFD	0	1	0	1	DLC of received message = 5 or more

Table 41.3 Configuration of DLC value (2 of 2)

Format	DLC[3]	DLC[2]	DLC[1]	DLC[0]	Description
CAN and CANFD	0	1	1	0	DLC of received message = 6 or more
CAN and CANFD	0	1	1	1	DLC of received message = 7 or more
CAN	1	x	x	x	DLC of received message = 8 or more
CANFD	1	0	0	0	DLC of received message = 8 or more*1
CANFD	1	0	0	1	DLC of received message = 12 or more*1
CANFD	1	0	1	0	DLC of received message = 16 or more*1
CANFD	1	0	1	1	DLC of received message = 20 or more*1
CANFD	1	1	0	0	DLC of received message = 24 or more*1
CANFD	1	1	0	1	DLC of received message = 32 or more*1
CANFD	1	1	1	0	DLC of received message = 48 or more*1
CANFD	1	1	1	1	DLC of received message = 64*1

Note 1. This setting is not available in the classical CAN function.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

GAFLIFL0 bit (Global Acceptance Filter List Information Label 0)

The GAFLIFL0 bit allows the configuration of a 2-bit information label that can be attached to a received message accepted by the related Global Acceptance Filter List entry. This bit is an LSB bit of an information label.

You cannot write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to the bit when the related CANFD channel is in CH_RESET or CH_HALT mode.

This bit is stored in Information Label Field[0] (CFDRMFDSTSb.RMIFL[0], CFDRFFDSTSb.RFIFL[0], CFDCFFDCSTS.CFIFL[0]) of the storage location of an incoming message.

GAFLRMDP[4:0] bits (Global Acceptance Filter List RX Message Buffer Direction Pointer)

The GAFLRMDP[4:0] bits allow the configuration of a single reception message buffer as the destination target for a received message that passes the acceptance check of the related Global Acceptance Filter List entry. The value entered is the single destination message buffer number.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

CFDRMNB.NRXMB[4:0] is the value entered in the RX Message Buffer Number Register to configure the number of RX message buffers. The value to be entered in CFDGAFLP0r.GAFLRMDP[4:0] bits should only be between 0x00 and CFDMNB.NMB[4:0] to 1 less.

If CFDRMNB.NRXMB[4:0] = 0x00, the GAFLRMV bit should be configured as 0.

GAFLRMV bit (Global Acceptance Filter List RX Message Buffer Valid)

The GAFLRMV bit allows the enabling or disabling of a single reception message buffer as the target for a received message that passes the acceptance check of the related Global Acceptance Filter List entry.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

GAFLPTR[15:0] bits (Global Acceptance Filter List Pointer)

The GAFLPTR[15:0] bits allow the configuration of a 16-bit pointer to be attached to a received message accepted by the related Global Acceptance Filter List entry. The pointer is added during message storage in the Message Buffer area and can be used by the application as a support function. The pointer information can be used for example, to support PDU Identifier allocation for the received message in AUTOSAR systems.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

41.2.22 CFDGAFLP1r : Global Acceptance Filter List Pointer 1 Registers (r = 1 to 16)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x012C + 0x0010 × (r - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	GAFL FDP8	—	—	—	—	—	—	GAFL FDP1	GAFL FDP0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	GAFLFDP0	Global Acceptance Filter List FIFO Direction Pointer FIFO direction pointer bits for received message storage 0: Disable RX FIFO 0 as target for reception 1: Enable RX FIFO 0 as target for reception	R/W
1	GAFLFDP1	Global Acceptance Filter List FIFO Direction Pointer FIFO direction pointer bits for received message storage 0: Disable RX FIFO 1 as target for reception 1: Enable RX FIFO 1 as target for reception	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	GAFLFDP8	Global Acceptance Filter List FIFO Direction Pointer FIFO direction pointer bits for received message storage 0: Disable Common FIFO as target for reception 1: Enable Common FIFO as target for reception	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The Global Acceptance Filter List Pointer 1 registers are used to configure the FIFO direction pointer fields in each Rule Entry of the Global Acceptance Filter List.

GAFLFDP8, GAFLFDP1, GAFLFDP0 bits (Global Acceptance Filter List FIFO Direction Pointer)

These bits allow the configuration of FIFO Buffers as the target for a received message passing the acceptance check of the related Global Acceptance Filter List entry. Each bit of the GAFLFDP8, GAFLFDP1, GAFLFDP0 is configuring a dedicated FIFO.

Users cannot write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

For storage in Common FIFO, target for reception can only be those Common FIFO Buffers that are configured as RX FIFO.

Only write to these bits when the related CANFD channel is in CH_RESET or CH_HALT mode.

Users should only configure up to 2 destination FIFO Buffers or 1 destination FIFO Buffers plus one RX Message Buffer.

41.2.23 CFDRMNB : RX Message Buffer Number Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0030

Bit position:	31 … 11	10 … 8	7 … 5	4 … 0
Bit field:	— —	RMPLS[2:0]	— — —	NRXMB[4:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0	0 0 0	0 0 0 0 0

Bit	Symbol	Function	R/W
4:0	NRXMB[4:0]	Number of RX Message Buffers	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
10:8	RMPLS[2:0]	Reception Message Buffer Payload Data Size 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The RX Message Buffer Number register is used to configure the total number of RX message buffers allocated to channels.

NRXMB[4:0] bits (Number of RX Message Buffers)

The NRXMB[4:0] bits are used to configure the number of RX message buffers.

Only write to these bits when the CANFD module is in GL_RESET mode.

Enter only values between 0 and 16 inclusive, with 0x00 indicating that no RX message buffer is allocated.

RMPLS[2:0] bits (Reception Message Buffer Payload Data Size)

The RMPLS[2:0] bits are used to configure the message buffer payload data size.

Only write to these bits when the CANFD module is in GL_RESET mode.

41.2.24 CFDRMND : RX Message Buffer New Data Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0034

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RMNS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RMNS[15:0]	RX Message Buffer New Data Status 0: New data not stored in corresponding RX message buffer 1: New data stored in corresponding RX message buffer	R/W

Note: S-TYPE-3, P-TYPE-3

The RX Message Buffer New Data Status Register specifies the new data storage status of the RX message buffers.

RMNS[15:0] bits (RX Message Buffer New Data Status)

The RMNS[15:0] bits indicate the status of new data for the corresponding RX message buffer. RMNS bit [0] corresponds to RX message buffer [0] and so on.

The bit position of CFDRMND corresponds to the buffer number of RXMB.

Do not write to these bits when the CANFD module is in GL_RESET or GL_SLEEP mode. Writing 1 has no effect.

These bits cannot be cleared when message storage in the corresponding RX message buffer is in progress.
Do not use the bit clear instruction to clear these bits. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

These bits are set automatically when storage of new messages are in the corresponding RX message buffer. These bits are cleared by writing 0. These bits are cleared automatically when the CANFD module is in GL_RESET mode.

When CFDRMNB.RMPLS = 000b (maximum 8 bytes payload), the duration of message storage is 6 PCLKA cycles.

When CFDRMNB.RMPLS > 000b, the duration of message storage is 6 PCLKA cycles + 1 for each 4 bytes (maximum of 20 PCLKA cycles for 64 bytes).

Note: This feature is not available in the classical CAN function.

41.2.25 CFDRFCCa : RX FIFO Configuration/Control Registers a (a = 0 to 1)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x003C + 0x04 × a

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RFIGCV[2:0]			RFIM	—	RFDC[2:0]			—	RFPLS[2:0]			—	—	RFIE	RFE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFE	RX FIFO Enable 0: FIFO disabled 1: FIFO enabled	R/W
1	RFIE	RX FIFO Interrupt Enable 0: FIFO interrupt generation disabled 1: FIFO interrupt generation enabled	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
6:4	RFPLS[2:0]*1	Rx FIFO Payload Data Size Configuration 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
10:8	RFDC[2:0]	RX FIFO Depth Configuration 0 0 0: FIFO Depth = 0 message 0 0 1: FIFO Depth = 4 messages 0 1 0: FIFO Depth = 8 messages 0 1 1: FIFO Depth = 16 messages 1 0 0: FIFO Depth = 32 messages 1 0 1: FIFO Depth = 48 messages 1 1 0: Reserved 1 1 1: Reserved	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	RFIM	RX FIFO Interrupt Mode 0: Interrupt generated when RX FIFO counter reaches RFIGCV value from values smaller than RFIGCV 1: Interrupt generated at the end of every received message storage	R/W
15:13	RFIGCV[2:0]	RX FIFO Interrupt Generation Counter Value 0 0 0: Interrupt generated when FIFO is 1/8th full 0 0 1: Interrupt generated when FIFO is 1/4th full 0 1 0: Interrupt generated when FIFO is 3/8th full 0 1 1: Interrupt generated when FIFO is 1/2 full 1 0 0: Interrupt generated when FIFO is 5/8th full 1 0 1: Interrupt generated when FIFO is 3/4th full 1 1 0: Interrupt generated when FIFO is 7/8th full 1 1 1: Interrupt generated when FIFO is full	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. These bits are not available in the classical CAN function.

The RX FIFO Configuration/Control Registers are used to configure and control the two RX FIFOs.

RFE bit (RX FIFO Enable)

The RFE bit enables the FIFO. When this bit is set to 0, the RX FIFO is cleared to empty.

Only write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode.

This bit can only be set if the configured FIFO depth is greater than 0x000 (CFDRFCCa.RFDC > 0x000) and less than 0x110.

Set the RFE bit with a separate write access to the CFDRFCCa register, after all the other bits in the CFDRFCCa register are set.

This bit is cleared automatically when the CANFD module is in GL_RESET mode.

RFIE bit (RX FIFO Interrupt Enable)

The RFIE bit enables generation of the FIFO interrupt.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

RFPLS[2:0] bits (Rx FIFO Payload Data Size Configuration)

The RFPLS[2:0] bits define the message data payload allocation in the RAM.

This is the maximum number of bytes which can be received by this FIFO.

Only write to these bits when the CANFD module is in GL_RESET mode.

Note: These bits are not available in the classical CAN function.

RFDC[2:0] bits (RX FIFO Depth Configuration)

The RFDC[2:0] bits select the depth of the FIFO in terms of the number of messages. If the FIFO depth is configured to 0 messages, the FIFO cannot be used.

Only write to these bits when the CANFD module is in GL_RESET mode.

RFIM bit (RX FIFO Interrupt Mode)

The RFIM bit selects the interrupt generation condition for the FIFO.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

Only write to this bit when the CANFD module is in GL_RESET mode.

RFIGCV[2:0] bits (RX FIFO Interrupt Generation Counter Value)

The RFIGCV[2:0] bits select the counter value of the FIFO for generation of FIFO interrupts. These values represent fractions of the FIFO depth for which an interrupt is generated.

Do not write to these bits when the CANFD module is in GL_SLEEP mode.

The setting of the RFIGCV[2:0] bits should be synchronized with the RFDC[2:0] bits.

Only write to these bits when the CANFD module is in GL_RESET mode.

41.2.26 CFDRFSTSa : RX FIFO Status Registers a (a = 0 to 1)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0044 + 0x04 × a

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	RFMC[5:0]						—	—	—	—	RFIF	RFMLT	RFFLL	RFEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	RFEMP	RX FIFO Empty 0: FIFO not empty 1: FIFO empty	R
1	RFFLL	RX FIFO Full 0: FIFO not full 1: FIFO full	R
2	RFMLT	RX FIFO Message Lost 0: No message lost in FIFO 1: FIFO message lost	R/W
3	RFIF	RX FIFO Interrupt Flag 0: FIFO interrupt condition not satisfied 1: FIFO interrupt condition satisfied	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
13:8	RFMC[5:0]	RX FIFO Message Count Number of messages stored in FIFO	R
31:14	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The RX FIFO Status Registers show the status of messages stored in the corresponding FIFO buffers.

RFEMP bit (RX FIFO Empty)

The RFEMP bit is set automatically when:

- The RFMC bit is 0
- RX FIFO is disabled by setting the CFDRFCCa.RFE bit to 0
- The CANFD module is in GL_RESET mode.

The RFEMP bit is cleared automatically when the first message is stored in the RX FIFO buffer.

RFFLL bit (RX FIFO Full)

The RFFLL bit is set automatically when the number of CAN messages stored in the FIFO buffer matches the configured FIFO depth.

The RFFLL is cleared automatically when:

- The number of CAN messages stored in the FIFO buffer is less than the configured FIFO depth
- RX FIFO is disabled by setting the CFDRFCCa.RFE bit to 0
- The CANFD module is in GL_RESET mode.

RFMLT bit (RX FIFO Message Lost)

Only write to the RFMLT bit when CANFD module is in GL_HALT or GL_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically whenever a message is lost due to attempted storage when the FIFO buffer is already full. If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

The bit is cleared:

- By writing 0 to it
- When the CANFD module is in GL_RESET mode.

RFIF bit (RX FIFO Interrupt Flag)

The RFIF bit is set automatically when the configured interrupt condition is satisfied. This bit is not automatically cleared when the RX FIFO buffer is disabled.

Only write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then this bit is set.

The bit is cleared by writing 0 to it. The bit is also cleared when CANFD module is in GL_RESET mode.

RFMC[5:0] bits (RX FIFO Message Count)

The RFMC[5:0] bits indicate the number of CAN messages stored in the RX FIFO buffer that can be read by the CPU.

These bits are cleared automatically when the FIFO is disabled and when the CANFD module is in GL_RESET mode.

41.2.27 CFDRFPCTRa : RX FIFO Pointer Control Registers a (a = 0 to 1)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x004C + 0x04 × a

Bit position:	31																								7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFPC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RFPC[7:0]	RX FIFO Pointer Control Increments read pointer of the corresponding RX FIFO buffers	W
31:8	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The RX FIFO Pointer Control Registers can be used to increment the read pointer of the corresponding RX FIFO buffers.

RFPC bits (RX FIFO Pointer Control)

When the value 0xFF is written to the RFPC bits, the pointer of the corresponding RX FIFO buffer is moved to the next FIFO entry. Only write 0xFF to these registers when the corresponding RX FIFO buffer is enabled and not empty.

The read value from these bits is always 0x00.

Only write to these bits when the CANFD module is in GL_HALT or GL_OPERATION mode.

Do not write to the RX FIFO Pointer Control registers when DMA is enabled.

41.2.28 CFDCFCC : Common FIFO Configuration/Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0054

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CFITT[7:0]								CFDC[2:0]			—	—	—	CFTML[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CFIGCV[2:0]			CFIM	CFITR	CFITSS	—	CFM	—	CFPLS[2:0]			—	CFTXIE	CFRXIE	CFE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CFE	Common FIFO Enable 0: FIFO disabled 1: FIFO enabled	R/W
1	CFRXIE	Common FIFO RX Interrupt Enable 0: FIFO interrupt generation disabled for Frame RX 1: FIFO interrupt generation enabled for Frame RX	R/W
2	CFTXIE	Common FIFO TX Interrupt Enable 0: FIFO interrupt generation disabled for Frame TX 1: FIFO interrupt generation enabled for Frame TX	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	CFPLS[2:0]*1	Common FIFO Payload Data Size Configuration 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	CFM	Common FIFO Mode 0: RX FIFO mode 1: TX FIFO mode	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
10	CFITSS	Common FIFO Interval Timer Source Select 0: Reference clock (× 1 / × 10 period) 1: Bit time clock of related channel (FIFO is linked to fixed channel)	R/W
11	CFITR	Common FIFO Interval Timer Resolution 0: Reference clock period × 1 1: Reference clock period × 10	R/W

Bit	Symbol	Function	R/W
12	CFIM	Common FIFO Interrupt Mode 0: RX FIFO mode: RX interrupt generated when Common FIFO counter reaches CFIGCV value from a lower value TX FIFO mode: TX interrupt generated when Common FIFO transmits the last message successfully 1: RX FIFO mode: RX interrupt generated at the end of every received message storage TX FIFO mode: interrupt generated for every successfully transmitted message	R/W
15:13	CFIGCV[2:0]	Common FIFO Interrupt Generation Counter Value 0 0 0: Interrupt generated when FIFO is 1/8th full 0 0 1: Interrupt generated when FIFO is 1/4th full 0 1 0: Interrupt generated when FIFO is 3/8th full 0 1 1: Interrupt generated when FIFO is 1/2 full 1 0 0: Interrupt generated when FIFO is 5/8th full 1 0 1: Interrupt generated when FIFO is 3/4th full 1 1 0: Interrupt generated when FIFO is 7/8th full 1 1 1: Interrupt generated when FIFO is full	R/W
17:16	CFTML[1:0]	Common FIFO TX Message Buffer Link Transmission scan link position of the corresponding channel	R/W
20:18	—	These bits are read as 0. The write value should be 0.	R/W
23:21	CFDC[2:0]	Common FIFO Depth Configuration 0 0 0: FIFO Depth = 0 message 0 0 1: FIFO Depth = 4 messages 0 1 0: FIFO Depth = 8 messages 0 1 1: FIFO Depth = 16 messages 1 0 0: FIFO Depth = 32 messages 1 0 1: FIFO Depth = 48 messages 1 1 0: FIFO Depth = Reserved 1 1 1: FIFO Depth = Reserved	R/W
31:24	CFITT[7:0]	Common FIFO Interval Transmission Time Delay the start of transmission from the FIFO if configured in TX mode, delay is a multiple of basic Interval Timer Clock Source unit	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. These bits are not available in the classical CAN function.

CFE bit (Common FIFO Enable)

The CFE bit enables the FIFO when set. FIFO is disabled when this bit is cleared.

This bit can also be used, by clearing it, to abort transmission from Common FIFO when configured in TX mode, or to stop reception into the Common FIFO in RX mode.

Only write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode and the related CANFD channel is not in CH_RESET mode for FIFOs configured as TX FIFO.

This bit can only be set if the configured FIFO depth is greater than 0x000 (CFDCFCC.CFDC > 0x000) and less than 0x110 (0x110 > CFDCFCC.CFDC > 0x000).

Set the CFE bit with a separate write access to the CFDCFCC register, after all the other bits in this register are set.

This bit is cleared automatically when the CANFD module is in GL_RESET mode.

This bit is also cleared automatically when the related channel is in CH_RESET mode if the FIFO is configured in TX mode.

CFRXIE bit (Common FIFO RX Interrupt Enable)

The CFRXIE bit enables generation of FIFO interrupts when the interrupt flag is set after reception of a frame in the corresponding FIFO buffer.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

CFTXIE bit (Common FIFO TX Interrupt Enable)

The CFTXIE bit enables generation of common FIFO interrupts when the interrupt flag is set after transmission of a frame from the corresponding FIFO buffer.

Do not write to this bit when the CANFD module is in GL_SLEEP mode.

CFPLS[2:0] bits (Common FIFO Payload Data Size Configuration)

The CFPLS[2:0] bits define the message data payload allocation in the RAM. This is the maximum number of bytes which can be received or transmitted by the FIFO buffer.

For details, see section 41.6. FIFO Buffers and Normal Message Buffer Configuration.

Only write to this bit when the CANFD module is in GL_RESET mode.

Note: These bits are not available in the classical CAN function.

CFM bit (Common FIFO Mode)

The CFM bit selects the mode of the FIFO. When a hardware reset is applied, all the Common FIFO buffers are configured in RX FIFO mode.

Do not write to these bits in GL_OPERATION or GL_SLEEP mode.

Only write to these bits when the CANFD module is in GL_RESET mode.

CFITSS bit (Common FIFO Interval Timer Source Select)

The CFITSS bit selects the basic clock source for the Interval Transmission Timer.

Do not write to this bit when the CANFD module is in GL_SLEEP mode. In addition, do not write to this bit when the CFE bit is set to 1.

Do not write 1 to this bit when CANFD communication is used.*1

Note: The bit time clock can vary depending on the nominal and data rate bit configuration.

Note 1. This feature is not available in the classical CAN function.

CFITR bit (Common FIFO Interval Timer Resolution)

The CFITR bit selects the resolution of the reference clock for the Interval Transmission Timer (peripheral clock is the source for the reference clock).

Do not write to this bit when the CANFD module is in GL_SLEEP mode. Also, do not write to this bit when the CFE bit is set to 1.

CFIM bit (Common FIFO Interrupt Mode)

The CFIM bit selects the interrupt generation condition for the FIFO buffer.

Do not write to this bit in GL_SLEEP mode.

Only write to this bit when the CANFD module is in GL_RESET mode.

CFIGCV[2:0] bits (Common FIFO Interrupt Generation Counter Value)

The CFIGCV[2:0] bits select the message counter value for the generation of FIFO interrupts. These values represent fractions of the FIFO depth at which the interrupt is to be generated.

Do not write to these bits when the CANFD module is in GL_SLEEP mode.

The setting of these bits should be synchronized with the CFDC[2:0] bits.

Only write to these bits when the CANFD module is in GL_RESET mode.

CFTML[1:0] bits (Common FIFO TX Message Buffer Link)

The CFTML[1:0] bits select the normal transmit message buffer position where the TX FIFO is linked to, for transmission scanning.

Do not write to these bits in GL_OPERATION or GL_SLEEP mode.

Only write to this bit when the CANFD module is in GL_RESET mode.

CFDC[2:0] bits (Common FIFO Depth Configuration)

The CFDC[2:0] bits select the depth of the common FIFO in terms of the number of messages. If the FIFO depth is configured to 0 message, the FIFO cannot be used.

Only write to these bits when the CANFD module is in GL_RESET mode.

CFITT[7:0] bits (Common FIFO Interval Transmission Time)

The CFITT[7:0] bits select the delay in the start of transmission for all messages transmitted from this FIFO buffer when configured in TX mode. The delay is a multiple of the basic interval timer clock source period (reference clock × 1, reference clock × 10, or bit time clock of the related CAN channel).

Do not write to these bits when the CANFD module is in GL_SLEEP mode.

Do not write to these bits when the CFE bit is set to 1.

When CFDGCFG.ITRCP[15:0] = 0x0000, set the CFITT[7:0] bits to 0x0000.

41.2.29 CFDCFSTS : Common FIFO Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0058

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	CFMC[5:0]						—	—	—	CFTXIF	CFRXIF	CFMLT	CFFLL	CFEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	CFEMP	Common FIFO Empty 0: FIFO not empty 1: FIFO empty	R
1	CFFLL	Common FIFO Full 0: FIFO not full 1: FIFO full	R
2	CFMLT	Common FIFO Message Lost 0: Number of message lost in FIFO 1: FIFO message lost	R/W
3	CFRXIF	Common RX FIFO Interrupt Flag 0: FIFO interrupt condition not satisfied after frame reception 1: FIFO interrupt condition satisfied after frame reception	R/W
4	CFTXIF	Common TX FIFO Interrupt Flag 0: FIFO interrupt condition not satisfied after frame transmission 1: FIFO Interrupt condition satisfied after frame transmission	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
13:8	CFMC[5:0]	Common FIFO Message Count Number of messages stored in FIFO	R
31:14	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CFEMP bit (Common FIFO Empty)

The CFEMP bit is set automatically when:

- The CPU has read all messages from the FIFO configured in RX mode
- All messages have been transmitted from the FIFO configured in TX mode
- The FIFO is disabled by setting the CFE bit to 0
- The CANFD module is in GL_RESET mode

- The related CANFD channel is in CH_RESET when FIFO configured in TX mode.

The CFEMP bit is cleared automatically when:

- The first reception message is stored in the FIFO buffer when configured in RX mode
- The first message to be transmitted is stored in the FIFO buffer when configured in TX mode.

CFFLL bit (Common FIFO Full)

The CFFLL bit is set automatically when the number of CAN messages stored in the FIFO matches the configured FIFO depth.

The CFFLL bit is cleared automatically when:

- The number of CAN messages stored in the FIFO is less than the configured FIFO depth
- The FIFO is disabled by setting the CFE bit to 0
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET mode when FIFO buffer is configured in TX mode.

CFMLT bit (Common FIFO Message Lost)

The CFMLT bit is set automatically whenever a message is lost due to attempted storage of a new message when FIFO is already full in RX mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then this bit is set.

Only write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode and the related CANFD channel is not in CH_RESET mode for FIFO configured as TX FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

The CFMLT bit is cleared:

- By writing 0 to it
- When the CANFD module is in GL_RESET mode
- When the related CANFD channel is in CH_RESET mode if the FIFO buffer is configured in TX mode.

CFRXIF bit (Common RX FIFO Interrupt Flag)

The CFRXIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode and the related CANFD channel is not in CH_RESET mode for FIFO configured as TX FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the configured interrupt condition is satisfied for Common FIFO buffers when configured in RX mode.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

The CFRXIF bit is cleared:

- By writing 0 to it
- When the CANFD module is in GL_RESET mode

CFTXIF bit (Common TX FIFO Interrupt Flag)

The CFTXIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode and the related CANFD channel is not in CH_RESET mode for FIFO buffer configured as TX FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the configured interrupt condition is satisfied for Common FIFO buffers configured in TX mode.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

The CFTXIF bit is cleared:

- By writing 0 to it
- When the CANFD module is in GL_RESET mode
- When the related CANFD channel is in CH_RESET mode if the FIFO buffer is configured in TX mode.

CFMC[5:0] bits (Common FIFO Message Count)

The CFMC[5:0] bits indicate the following:

- Number of CAN messages stored by the CPU in the FIFO buffer configured in TX mode pending for transmission
- Number of CAN messages stored in the FIFO buffer configured in RX mode by CANFD module to be read by the CPU

The CFMC[5:0] bits are cleared automatically when:

- The FIFO is disabled
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET mode if the FIFO buffer is configured in TX mode.

41.2.30 CFDCFPCTR : Common FIFO Pointer Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x005C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFPC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CFPC[7:0]	Common FIFO Pointer Control Increments read or write pointer of the corresponding Common FIFO buffers depending on the mode configuration.	W
31:8	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The Common FIFO Pointer Control Registers can be used to increment the read or write pointer of the corresponding Common FIFO buffer.

CFPC[7:0] bits (Common FIFO Pointer Control)

When the value 0xFF is written into the CFPC[7:0] bits, the read pointer of the corresponding Common FIFO buffer (when configured in RX mode), or the write pointer of the corresponding Common FIFO buffer (when configured in TX mode) moves to the next FIFO entry.

The read value from these bits is always 0x00.

Only write to these bits when the CANFD module is in GL_HALT or GL_OPERATION mode.

Only write 0xFF to this register when:

- The Common FIFO buffer is enabled and is not empty if configured in RX mode
- The Common FIFO buffer is enabled and is not full if configured in TX mode

Do not write to the Common FIFO Pointer Control registers when DMA is enabled.

41.2.31 CFDFESTS : FIFO Empty Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0060

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CFEMP	—	—	—	—	—	—	RFXEMP[1:0]	
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
1:0	RFXEMP[1:0]	RX FIFO Empty Status 0: Corresponding FIFO not empty 1: Corresponding FIFO empty	R
7:2	—	These bits are read as 0.	R
8	CFEMP	Common FIFO Empty Status 0: Corresponding FIFO not empty 1: Corresponding FIFO empty	R
31:9	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The FIFO Empty Status register shows status of the empty bits of the FIFO buffers.

RFXEMP[1:0] bits (RX FIFO Empty Status)

The RFXEMP[1:0] bits are set when the CANFD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

CFEMP bit (Common FIFO Empty Status)

The CFEMP bits are set when the CANFD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

41.2.32 CFDFFSTS : FIFO Full Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0064

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CFFLL	—	—	—	—	—	—	RFXFLL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	RFXFLL[1:0]	RX FIFO Full Status 0: Corresponding FIFO not full 1: Corresponding FIFO full	R
7:2	—	These bits are read as 0.	R
8	CFFLL	Common FIFO Full Status 0: Corresponding FIFO not full 1: Corresponding FIFO full	R
31:9	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The FIFO Full Status Register shows status of the full bits of the FIFO buffers.

RFXFLL[1:0] bits (RX FIFO Full Status)

The RFXFLL[1:0] bits are cleared when CANFD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

CFFLL bits (Common FIFO Full Status)

The CFFLL bits are cleared when the CANFD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

41.2.33 CFDFMSTS : FIFO Message Lost Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0068

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CFMLT	—	—	—	—	—	—	RFXMLT[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	RFXMLT[1:0]	RX FIFO Message Lost Status 0: Corresponding FIFO Message Lost flag not set 1: Corresponding FIFO Message Lost flag set	R
7:2	—	These bits are read as 0.	R
8	CFMLT	Common FIFO Message Lost Status 0: Corresponding FIFO Message Lost flag not set 1: Corresponding FIFO Message Lost flag set	R
31:9	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The FIFO Message Lost Status Register shows status of the Msg Lost bits of the FIFO buffers.

RFXMLT[1:0] bits (RX FIFO Message Lost Status)

The RFXMLT[1:0] bits are cleared when the CANFD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

CFMLT bits (Common FIFO Message Lost Status)

The CFMLT bits are cleared when the CANFD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

41.2.34 CFDRFISTS : RX FIFO Interrupt Flag Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x006C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFXIF[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	RFXIF[1:0]	RX FIFO[x] Interrupt Flag Status 0: Corresponding RX FIFO Interrupt flag not set 1: Corresponding RX FIFO Interrupt flag set	R
31:2	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The FIFO Interrupt Flag Status Register shows status of the interrupt flag bits of the RX FIFO buffers.

RFXIF[1:0] bits (RX FIFO[x] Interrupt Flag Status)

Each bit is set automatically when the corresponding interrupt flag bit is set in the RX FIFO Status Registers.

The RFXIF[1:0] bits are cleared when the CANFD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding interrupt flag bit is cleared in the RX FIFO Status Registers.

41.2.35 CFDCDTCT : DMA Transfer Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00C8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CFDMAE	—	—	—	—	—	—	RFDMAE1	RFDMAE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFDMAE0	DMA Transfer Enable for RXFIFO 0 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
1	RFDMAE1	DMA Transfer Enable for RXFIFO 1 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	CFDMAE	DMA Transfer Enable for Common FIFO 0 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The DMA Transfer Control Register controls the start and stop of DMA transfer operation.

RFDMAEe (e = 0 to 1) bit (DMA Transfer Enable for RXFIFO e)

The RFDMAEe bit cannot be set in GL_SLEEP or GL_RESET mode.

This bit is cleared when the CANFD module is in GL_RESET mode.

CFDMAE bit (DMA Transfer Enable for Common FIFO)

The CFDMAE bit enables or disables DMA transfer request for common FIFO

The CFDMAE bit cannot be set in GL_SLEEP or GL_RESET mode.

Do not enable a DMA transfer for a Common FIFO that is configured as TX FIFO.

This bit is cleared when the CANFD module is in GL_RESET mode.

41.2.36 CFDCDTSTS : DMA Transfer Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00CC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CFDMASTS	—	—	—	—	—	—	RFDMASTS1	RFDMASTS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFDMASTS0	DMA Transfer Status for RX FIFO 0 0: DMA transfer stopped 1: DMA transfer on going	R
1	RFDMASTS1	DMA Transfer Status for RX FIFO 1 0: DMA transfer stopped 1: DMA transfer on going	R
7:2	—	These bits are read as 0.	R
8	CFDMASTS	DMA Transfer Status only for Common FIFO 0: DMA transfer stopped 1: DMA transfer on going	R
31:9	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The DMA Transfer Status Register shows the status of the DMA transfer.

RFDMASTSe (e = 0 to 1) bit (DMA Transfer Status for RX FIFO e)

Each bit is set automatically when the corresponding DMA enable bit is set and the corresponding DMA FIFO is not empty.

Each bit is cleared automatically when the DMA transfer stops either because the DMA is disabled or the DMA FIFO is empty.

When CFDCDTCT.RFDMAEe (see CFDCDTCT.RFDMAEe bit in section 41.2.35. CFDCDTCT : DMA Transfer Control Register) is set to 0 while DMA transfer for the corresponding FIFO is on going, the RFDMASTSe bit becomes 0 when the DMA transfer is complete.

This bit is cleared when the CANFD module is in GL_RESET mode.

CFDMASTS bit (DMA Transfer Status only for Common FIFO)

Each bit is set automatically when the corresponding DMA enable bit is set and the corresponding DMA FIFO is not empty.

Each bit is cleared automatically when the DMA transfer stops either because the DMA is disabled or the DMA FIFO is empty.

When CFDCDTCT.CFDMAE (see CFDCDTCT.CFDMAE bit in section 41.2.35. CFDCDTCT : DMA Transfer Control Register) is set to 0 while DMA transfer for the corresponding FIFO is on going, the CFDMASTS bit becomes 0 when the DMA transfer is complete.

This bit is cleared when the CANFD module is in GL_RESET mode.

41.2.37 CFDTMCi : TX Message Buffer Control Registers i (i = 0 to 3)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0070 + 0x01 × i

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TMOM	TMTAR	TMTR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMTR	TX Message Buffer Transmission Request 0: TX Message buffer transmission not requested 1: TX message buffer transmission requested	R/W
1	TMTAR	TX Message Buffer Transmission Abort Request 0: TX message buffer transmission request abort not requested 1: TX message buffer transmission request abort requested	R/W
2	TMOM	TX Message Buffer One-shot Mode 0: TX message buffer not configured in one-shot mode 1: TX message buffer configured in one-shot mode	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The TX Message Buffer Control Registers configure the TX message buffer functions.

TMTR bit (TX Message Buffer Transmission Request)

When the TMTR bit is set, the CANFD module logic tries to transmit the message stored in the corresponding message buffer.

Only write to this bit when the related CANFD module is in CH_HALT or CH_OPERATION mode.

Do not set this bit if the corresponding TX message buffer is linked to a COM FIFO in TX mode or is a part of TX Queue.

This bit cannot be directly cleared by a CPU write access.

This bit can only be set when the Transmission Result flag bits (CFDTMSTSj.TMTRF) in the CFDTMSTSj register corresponding to the message buffer are cleared to 00b.

The TMTR bit is automatically cleared by the:

- CANFD module logic at the end of a successful transmission
- CANFD module logic at the end of a transmission abort, requested by the corresponding CFDTMCi.TMTAR bit
- CANFD module logic when there is a detection of a CAN bus error or arbitration loss if CFDTMCi.TMOM bit is set for the message buffer
- CANFD module logic when the CANFD module is in GL_RESET mode or the related channel is in CH_RESET mode.

TMTAR bit (TX Message Buffer Transmission Abort Request)

When the TMTAR bit is set, the CANFD module logic tries to abort the transmission of the frame stored in the corresponding message buffer.

In most cases, transmission cannot be aborted if the internal scan for transmission is complete and the message buffer has already been selected for transmission. In this case, frame may be transmitted successfully from the message buffer. The message buffer selection is released by entering CH_HALT mode.

However, message buffer selected for transmission can be aborted by an abort request when the CAN node detects a new message on the bus (RX pin) before it starts transmission from the selected message buffer.

Only write to the TMTAR bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode. This bit can only be set when the related transmit request TMTR bit is set.

The TMTAR bit cannot be cleared by a CPU write access. Clearing of this bit by CANFD has priority over setting by a CPU write access.

The TMTAR bit is automatically cleared by:

- The CANFD module logic at the end of a successful transmission
- The CANFD module logic at the end of a transmission abort
- The CANFD module logic when there is detection of a CAN bus error or arbitration loss
- The CANFD module logic when the CANFD module is in GL_RESET mode or the related channel enters CH_RESET mode.

TMOM bit (TX Message Buffer One-shot Mode)

When the TMOM bit is set, the CANFD module logic tries to transmit the message only once.

If the transmission is successful, the CFDTMSTSj.TMTRF bits are set to 10b or 11b. Otherwise, the transmission is automatically aborted and CFDTMSTSj.TMTRF bits are set to 01b due to a bus error or a bus arbitration lost.

The TMOM bit remains set if the transmission has completed successfully or aborted due to an error or a loss of arbitration.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

Set this bit at the same time as the TMTR bit. Clear this bit with a write access.

If a message has already been requested for transmission, do not write to this bit until the message has been successfully transmitted or transmission has been aborted.

The TMOM bit is automatically cleared by the CANFD module logic when the CANFD module is in GL_RESET mode or the related channel is in CH_RESET mode.

41.2.38 CFDTMSTSj : TX Message Buffer Status Registers j (j = 0 to 3)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0074 + 0x01 × j

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TMTARM	TMTRM	TMTRF[1:0]		TMTSTS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMTSTS	TX Message Buffer Transmission Status 0: No ongoing transmission 1: Ongoing transmission	R
2:1	TMTRF[1:0]	TX Message Buffer Transmission Result Flag 0 0: No result 0 1: Transmission aborted from the TX message buffer 1 0: Transmission successful from the TX message buffer and transmission abort was not requested 1 1: Transmission successful from the TX message buffer and transmission abort was requested	R/W
3	TMTRM	TX Message Buffer Transmission Request Mirrored 0: TX message buffer transmission not requested 1: TX message buffer transmission requested	R
4	TMTARM	TX Message Buffer Transmission Abort Request Mirrored 0: TX message buffer transmission request abort not requested 1: TX message buffer transmission request abort requested	R
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The TX Message Buffer Status Registers show status of the transmission and transmission abort for the corresponding message buffers.

TMTSTS bit (TX Message Buffer Transmission Status)

The TMTSTS bit is set automatically at the start of the transmission from the corresponding TX message buffer.

This bit is cleared automatically when:

- Transmission stops
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET mode.

TMTRF[1:0] bits (TX Message Buffer Transmission Result Flag)

The TMTRF[1:0] bits show the result for the corresponding TX message buffer. The status is as follows:

- 00: Transmission in progress or has not been requested
- 01: Transmission has been aborted from the corresponding TX message buffer
- 10: Transmission was successful from the corresponding TX message buffer and the CFDTMCi.TMTAR bit was not set for this TX message buffer
- 11: Transmission was successful from the corresponding TX message buffer, but the CFDTMCi.TMTAR bit was set for this TX message buffer.

Only write to these bits when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

The TMTRF[1:0] bits are cleared automatically when the CANFD module is in GL_RESET mode or the related channel is in CH_RESET mode.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

TMTRM bit (TX Message Buffer Transmission Request Mirrored)

The TMTRM bit is set when the CFDTMCi.TMTR bit in the corresponding CFDTMCi register is set.

This bit is cleared when the CFDTMCi.TMTR bit in the corresponding CFDTMCi register is cleared.

TMTARM bit (TX Message Buffer Transmission Abort Request Mirrored)

The TMTARM bit is set when the CFDTMCi.TMTAR bit in the corresponding CFDTMCi register is set.

This bit is cleared when the CFDTMCi.TMTAR bit in the corresponding CFDTMCi register is cleared.

41.2.39 CFDTMTRSTS : TX Message Buffer Transmission Request Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0078

Bit position:	31																												3			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFDTMTRSTS [3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CFDTMTRSTS[3:0]	TX Message Buffer Transmission Request Status 0: Transmission not requested for corresponding TX message buffer 1: Transmission requested for corresponding TX message buffer	R
31:4	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

These bits show the TX Message Buffer Transmission Request Status for the corresponding TX Message Buffer. The bit 0 of a CFDTMTRSTS register corresponds to the TX message buffer 0.

The bit position of CFDTMTRSTS corresponds to the buffer number of TX message buffer.

CFDTMTRSTS[3:0] bits (TX Message Buffer Transmission Request Status)

The CFDTMTRSTS[3:0] bits show status of the CFDTMCi.TMTR bits of the TX Message Buffer Control Registers.

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Control Registers (CFDTMCi), and only when the message buffer does not belong to a TX Queue.

Each bit is cleared automatically when:

- The corresponding bit is cleared in the TX Message Buffer Control Registers
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET mode.

41.2.40 CFDTMTARSTS : TX Message Buffer Transmission Abort Request Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x007C

Bit position:	31																												3			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFDTMTARSTS [3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CFDTMTARSTS[3:0]	TX Message Buffer Transmission Abort Request Status 0: Transmission abort not requested for corresponding TX message buffer 1: Transmission abort requested for corresponding TX message buffer	R
31:4	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

These bits show the TX Message Buffer Transmission Abort Request Status for the corresponding TX Message Buffer. The bit 0 of a CFDTMTARSTS register corresponds to the TX message buffer 0.

The bit position of CFDTMTARSTS corresponds to the buffer number of TX message buffer.

CFDTMTARSTS[3:0] bits (TX Message Buffer Transmission Abort Request Status)

The CFDTMTARSTS[3:0] bits show status of the CFDTMCi.TMTAR bits of the TX Message Buffer Control Registers.

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Control Registers, and when the message buffer belongs to a TX Queue.

Each bit is cleared automatically when:

- The corresponding bit is cleared in the TX Message Buffer Control Registers
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET mode.

41.2.41 CFDTMTCSTS : TX Message Buffer Transmission Completion Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0080

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFDTMTCSTS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CFDTMTCSTS[3:0]	TX Message Buffer Transmission Completion Status 0: Transmission not complete for corresponding TX message buffer 1: Transmission completed for corresponding TX message buffer	R
31:4	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

These bits show the TX Message Buffer Transmission Completion Status for the corresponding TX Message Buffer. The bit 0 of a CFDTMTCSTS register corresponds to the TX message buffer 0.

The bit position of CFDTMTCSTS corresponds to the buffer number of TX message buffer.

CFDTMTCSTS[3:0] bits (TX Message Buffer Transmission Completion Status)

The CFDTMTCSTS[3:0] bits show status of successful completion of the TX Message Buffer Status Registers.

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Status Registers.

Each bit is cleared automatically when:

- The corresponding bit is cleared in the TX Message Buffer Status Registers
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET mode.

If a CAN channel enters CH_RESET mode, then the bits related to that channel are cleared.

41.2.42 CFDTMTASTS : TX Message Buffer Transmission Abort Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0084

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFDTMTASTS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CFDTMTASTS[3:0]	TX Message Buffer Transmission Abort Status 0: Transmission not aborted for corresponding TX message buffer 1: Transmission aborted for corresponding TX message buffer	R
31:4	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

These bits show the TX Message Buffer Transmission abort Status for the corresponding TX Message Buffer. The bit 0 of a CFDTMTASTS register corresponds to the TX message buffer 0.

The bit position of CFDTMTASTS corresponds to the buffer number of TX message buffer.

CFDTMTASTS[3:0] bits (TX Message Buffer Transmission Abort Status)

The CFDTMTASTS[3:0] bits show status of the successful transmission abort of the corresponding TX message buffer.

Each bit is set automatically when the CFDTMSTSj.TMTRF bits are set to 01b in the corresponding TX Message Buffer Status Register.

Each bit is cleared automatically when:

- The CFDTMSTSj.TMTRF bits are cleared in the corresponding TX Message Buffer Status Register
- The CANFD module is in GL_RESET mode
- The related CANFD channel is in CH_RESET mode.

41.2.43 CFDTMIEC : TX Message Buffer Interrupt Enable Configuration Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0088

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TMIEg[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	TMIEg[3:0]	TX Message Buffer Interrupt Enable 0: TX message buffer interrupt disabled for corresponding TX message buffer 1: TX message buffer interrupt enabled for corresponding TX message buffer	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

These bits show the TX Message Buffer Interrupt Enable for the corresponding TX Message Buffer.

The bit 0 of a CFDTMIEC register corresponds to the TX message buffer 0.

The bit position of CFDTMIEC corresponds to the buffer number of TX message buffer.

g = [0…3]

TMIEg[3:0] bits (TX Message Buffer Interrupt Enable)

If the TMIEg[3:0] bits are set, an interrupt is generated at the end of a successful transmission from the corresponding message buffer.

See section 41.7. Interrupts and DMA for TX Message Buffer Interrupt specification.

Do not write to the TMIEg[3:0] bits when:

- The CANFD module is in GL_SLEEP mode
- The related CANFD channel is in CH_SLEEP mode
- The corresponding TX message buffer is part of a TX Queue
- The corresponding TX message buffer is linked to a Common FIFO with the CFDCFCC.CFTML bits.

41.2.44 CFDTXQCC : TX Queue Configuration/Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x008C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TXQDC[1:0]		TXQIM	—	TXQTXIE	—	—	—	—	TXQE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TXQE	TX Queue Enable 0: TX Queue disabled 1: TX Queue enabled	R/W
4:1	—	These bits are read as 0. The write value should be 0.	R/W
5	TXQTXIE	TX Queue TX Interrupt Enable 0: TX Queue TX interrupt disabled 1: TX Queue TX interrupt enabled	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	TXQIM	TX Queue Interrupt Mode 0: When the last message is successfully transmitted 1: At every successful transmission	R/W
9:8	TXQDC[1:0]	TX Queue Depth Configuration 00: 0 messages 01: Reserved 10: 3 messages 11: 4 messages	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The TX Queue Configuration/Control Registers are used to configure the TX Queue transmission.

TXQ is composed of TXMB0 to TXMB3 (at the maximum) when TXQE is enabled.

TXQE bit (TX Queue Enable)

The TXQE bit cannot be set if the configured TX Queue depth is 0x00 (CFDTXQCC.TXQDC == 0x00).

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in CH_RESET or CH_SLEEP mode.

The TXQE bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

TXQTXIE bit (TX Queue TX Interrupt Enable)

When the TXQTXIE bit is set, an interrupt is generated based on the setting of the TXQIM bit.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in CH_SLEEP mode.

TXQIM bit (TX Queue Interrupt Mode)

The TXQIM bit selects the interrupt generation condition for the TX Queue.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQDC[1:0] bits (TX Queue Depth Configuration)

The TXQDC[1:0] bits select the depth of the transmission queue. The message buffer selection starts from MB[0] up to MB[3] depending on the configured depth.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the related CANFD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

41.2.45 CFDTXQSTS : TX Queue Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0090

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TXQMC[2:0]			—	—	—	—	—	TXQTXIF	TXQFLL	TXQEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	TXQEMP	TX Queue Empty 0: TX Queue not empty 1: TX Queue empty	R
1	TXQFLL	TX Queue Full 0: TX Queue not full 1: TX Queue full	R
2	TXQTXIF	TX Queue TX Interrupt Flag 0: TX Queue interrupt condition not satisfied after a frame TX 1: TX Queue interrupt condition satisfied after a frame TX	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
10:8	TXQMC[2:0]	TX Queue Message Count Number of messages in the TX Queue.	R
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The TX Queue Status Registers show the status of the TX Queue of corresponding CAN channel.

TXQEMP bit (TX Queue Empty)

The TXQEMP bit is set automatically when the TX Queue is disabled or no messages are stored in the TX Queue.

This bit is set automatically when:

- The last message is transmitted from the TX Queue
- The related CANFD channel is in CH_RESET mode.

The bit is cleared automatically when the first message to be transmitted is stored in the TX Queue.

TXQFLL bit (TX Queue Full)

The TXQFLL bit is set automatically when the number of CAN messages stored in the TX Queue matches the configured TX Queue depth.

This bit is cleared automatically when:

- The number of CAN messages stored in the TX Queue is less than the configured TX Queue depth
- The related CANFD channel is in CH_RESET mode.

TXQTXIF bit (TX Queue TX Interrupt Flag)

The TXQTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

This bit is set automatically when the configured interrupt condition is satisfied for the TX Queue.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

You cannot write to this bit when the related CANFD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CANFD channel is in CH_RESET mode.

TXQMC[2:0][13:8] bits (TX Queue Message Count)

The TXQMC[2:0] bits show the number of CAN messages in the TX Queue.

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

41.2.46 CFDTXQPCTR : TX Queue Pointer Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0094

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7–0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQPC[7:0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
7:0	TXQPC[7:0]	TX Queue Pointer Control Increments the write pointer to the TX Queue buffer in the corresponding channel	W
31:8	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The TX Queue Pointer Control Registers are used to confirm storage of a full message in the corresponding TX Queue buffers.

TXQPC[7:0] bits (TX Queue Pointer Control)

When the value 0xFF is written to the TXQPC[7:0] bits, the write pointer of the corresponding TX Queue buffer is updated and a transmit request is initiated for this message.

The read value from these bits is always 0x00. Do not write to the FIFO control registers when DMA is enabled.

You cannot write to these bits when the related CANFD channel is in CH_SLEEP or CH_RESET mode.

Only write 0xFF to this register when:

- The corresponding TX Queue is enabled and not full

- The Common FIFO is enabled.

41.2.47 CFDTHLCC : TX History List Configuration/Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0098

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	THLDTE	THLIM	THLIE	—	—	—	—	—	—	—	THLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	THLE	TX History List Enable 0: TX History List disabled 1: TX History List enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	THLIE	TX History List Interrupt Enable 0: TX History List Interrupt disabled 1: TX History List Interrupt enabled	R/W
9	THLIM	TX History List Interrupt Mode 0: Interrupt generated if TX History List level reaches ¾ of the TX History List depth 1: Interrupt generated for every successfully stored entry	R/W
10	THLDTE	TX History List Dedicated TX Enable 0: TX FIFO + TX Queue 1: Flat TX MB + TX FIFO + TX Queue	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The TX History List Configuration/Control Register configures the TX History List functions.

THLE bit (TX History List Enable)

The THLE bit enables the TX History List buffer when it is set.

You cannot write to this bit when the related CANFD channel is in CH_RESET or CH_SLEEP mode.

This bit is cleared automatically when the related CANFD channel is in CH_RESET mode.

THLIE bit (TX History List Interrupt Enable)

The THLIE bit enables the generation of the TX History List interrupt when it is set.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

THLIM bit (TX History List Interrupt Mode)

The THLIM bit selects the interrupt generation condition for the FIFO.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode.

THLDTE bit (TX History List Dedicated TX Enable)

The THLDTE bit selects the condition for storing an entry in the TX History List after successful transmission.

You cannot write to this bit when the CANFD module is in GL_SLEEP mode.

Do not write to this bit when the CANFD module is in GL_HALT or GL_OPERATION mode.

41.2.48 CFDTHLSTS : TX History List Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x009C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	THLMC[3:0]				—	—	—	—	THLIF	THLELT	THLFLL	THLEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	THLEMP	TX History List Empty 0: TX History List not empty 1: TX History List empty	R
1	THLFLL	TX History List Full 0: TX History List not full 1: TX History List full	R
2	THLELT	TX History List Entry Lost 0: No entry lost in TX History List 1: TX History List entry Lost	R/W
3	THLIF	TX History List Interrupt Flag 0: TX History List interrupt condition not satisfied 1: TX History List interrupt condition satisfied	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
11:8	THLMC[3:0]	TX History List Message Count Number of messages stored in TX History List	R
31:12	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The TX History List Status register shows the status of data stored in the TX History List buffer.

THLEMP bit (TX History List Empty)

The THLEMP bit is set automatically when the CPU has read all the entries from the TX History List buffer.

This bit is cleared automatically when the first entry is stored to the TX History List.

This bit is set automatically when:

- TX History List is disabled
- The related CANFD channel is in CH_RESET mode.

THLFLL bit (TX History List Full)

The THLFLL bit is set automatically when the number of entries in the TX History List buffer matches the TX History List depth.

Each TX History List can store up to 8 entries.

This bit is cleared automatically when:

- The number of entries in the TX History List buffer is less than the TX History List depth
- The TX History List is disabled
- The related CANFD channel is in CH_RESET mode.

THLELT bit (TX History List Entry Lost)

The THLELT bit is set when a new entry cannot be stored because the related TX History List buffer is already full.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

This bit is cleared:

- By writing 0 to it
- When the related CANFD channel is in CH_RESET mode.

THLIF bit (TX History List Interrupt Flag)

The THLIF bit is set when the configured interrupt condition is satisfied.

Only write to this bit when the related CANFD channel is in CH_HALT or CH_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

This bit is cleared:

- By writing 0 to it
- When the related CANFD channel is in CH_RESET mode.

The bit is cleared by writing 0 to it.

This bit is automatically cleared in CH_RESET mode.

THLMC[3:0] bits (TX History List Message Count)

The THLMC[3:0] bits show the number of transmitted messages stored in the TX History List.

These bits are cleared automatically when the related CANFD channel is in CH_RESET mode.

41.2.49 CFDTHLACC0 : TX History List Access Register 0

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0740

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TMTS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	BN[1:0]		BT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	BT[2:0]	Buffer Type 0 0 1: Flat TX message buffer 0 1 0: TX FIFO message buffer number 1 0 0: TX Queue message buffer number	R
4:3	BN[1:0]	Buffer Number Number of the message buffer	R

Bit	Symbol	Function	R/W
15:5	—	These bits are read as 0.	R
31:16	TMTS[15:0]	Transmit Timestamp Transmit timestamp value for software drivers	R

Note: S-TYPE-3, P-TYPE-3

The TX History List Access Registers 0 provide access to the entry in the TX History List based on the read timestamp value.

BT[2:0] bits (Buffer Type)

The BT[2:0] bits indicate whether data has been stored following a transmission from a FIFO buffer, a TX Queue or a TX message buffer.

BN[1:0] bits (Buffer Number)

The BN[1:0] bits show the message buffer from which transmission was successfully completed. If a message from a Common FIFO is transmitted, then these bits show the message buffer that is linked to the Common FIFO for transmission.

TMTS[15:0] bits (Transmit Timestamp)

The TMTS[15:0] bits indicate the timestamp for use by software drivers.

41.2.50 CFDTHLACC1 : TX History List Access Register 1

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0744

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TIFL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TID[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TID[15:0]	Transmit ID These bits indicate that message buffer reference ID, TX FIFO reference ID, or AFL pointer field is stored for software drivers.	R
17:16	TIFL[1:0]	Transmit Information Label These bits indicate that message buffer information label, TX FIFO information label, or AFL information label is stored for software drivers.	R
31:18	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The TX History List Access Registers 1 provide access to entry in the TX History List based on the read pointer value.

TID[15:0] bits (Transmit ID)

The TID[15:0] bits indicate whether the message buffer reference ID (CFDTMFDCTRb.TMPTR) or the TX FIFO reference ID (CFDCFFDCSTS.CFPTR) is for use by software drivers.

TIFL[1:0] bits (Transmit Information Label)

The TIFL[1:0] bits indicate whether the message buffer information label (CFDTMFDCTRb.TMIFL) or the TX FIFO information label (CFDCFFDCSTS.CFIFL) is for use by software drivers.

41.2.51 CFDTHLPCTR : TX History List Pointer Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00A0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	THLPC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	THLPC[7:0]	TX History List Pointer Control Increments the write pointer to the TX History List in the corresponding channel	W
31:8	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The TX History List Pointer Control Registers are used to increment the read pointer of the TX History List.

THLPC[7:0] bits (TX History List Pointer Control)

When 0xFF is written to the THLPC[7:0] bits, the read pointer of the TX History List is moved to the next TX History List entry address.

The read value from these bits is always 0x00. Only write to these bits when the related CANFD channel is in CH_HALT or CH_OPERATION mode.

Only write 0xFF to these registers when the corresponding TX History List is enabled and not empty.

41.2.52 CFDGRSTC : Global SW reset Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00D8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	SRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SRST	SW Reset 0: Normal state 1: SW reset state	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits control the validity of rewriting of a SRST bit.	W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

SRST bit (SW Reset)

When the SRST bit is set, the CANFD module is in the same state as hardware reset. When a reset is required, write 1 then write 0 to this bit.

This bit is cleared when the CANFD module is in GL_SLEEP mode.

When this bit is cleared, the RAM initialization sequence does not operate. The configuration of RAM is performed by software.

The RAM is not initialized when software reset is performed during the initialization of RAM. Software must perform the initialization of RAM.

KEY[7:0] bits (Key Code)

When 0xC4 is written in the KEY[15:8] bits, a write to the SRST bit is valid.

The read value from these bits is always 0x00.

CFDGRSTC.SRST bit and the CFDGRSTC.KEY bit should be written simultaneously.

41.2.53 CFDGTSTCFG : Global Test Configuration Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00A8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	RTMPS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
19:16	RTMPS[3:0]	RAM Test Mode Page Select Select a RAM test mode page	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The Global Test Configuration Register is used to configure the RAM test mode page.

RTMPS[3:0] bits (RAM Test Mode Page Select)

The RTMPS[3:0] bits select the RAM page mode for CPU read/write access when the CANFD module is configured in RAM test mode.

See section 41.9.2.1. RAM Test Mode for the RAM test mode specification.

Do not write to these bits when the CANFD module is in GL_RESET or GL_SLEEP mode.

Only enter values from 0 to 8 (0x008) for the message buffer RAM.

Only write to these bits when the CANFD module is in GL_HALT mode.

These bits are cleared automatically when the related CANFD channel is in GL_RESET mode.

41.2.54 CFDGTSTCTR : Global Test Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00AC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	RTME	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
2	RTME	RAM Test Mode Enable 0: RAM test mode disabled 1: RAM test mode enabled	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The Global Test Control register is used to control the global test modes of the CANFD module.

RTME bit (RAM Test Mode Enable)

When the RTME bit is set, the CANFD module is configured in RAM test mode. See section 41.9.2.1. RAM Test Mode for RAM test mode specification.

Only write to this bit when the CANFD module is in GL_HALT mode.

Clear this bit when the CANFD module is in GL_HALT mode.

This bit is cleared automatically when the CANFD module is in GL_RESET mode.

41.2.55 CFDGFDCFG : Global FD Configuration Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00B0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TSCCFG[1:0]		—	—	—	—	—	—	—	RPED
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RPED	RES Bit Protocol Exception Disable 0: Protocol exception event detection enabled 1: Protocol exception event detection disabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
9:8	TSCCFG[1:0]	Timestamp Capture Configuration 0 0: Timestamp capture at the sample point of SOF (start of frame) 0 1: Timestamp capture at frame valid indication 1 0: Timestamp capture at the sample point of RES bit 1 1: Reserved	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

RPED bit (RES Bit Protocol Exception Disable)

The RPED bit configures the protocol exception event handling according to ISO 11898-1.

When this bit is enabled, the protocol exception event detection is disabled, and the protocol controller transmits an error frame when the protocol exception event is detected (RES bit is sampled recessive).

Only write to this bit when the CANFD module is in GL_RESET mode.

TSCCFG[1:0] bits (Timestamp Capture Configuration)

The TSCCFG[1:0] bits configure the different capture points of the timestamp for transmission and reception.

When CFDGFDCFG.TSCCFG[1:0] = 10b, the timestamp capture is performed for CANFD frames at RES bit and for Classical frames at the start of frame.

Only write to these bits when the CANFD module is in GL_RESET mode.

41.2.56 CFDGLOCKK : Global Lock Key Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00B8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15–0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LOCK[15:0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	LOCK[15:0]	Lock Key Key bits for unlocking the protection of test modes	W
31:16	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The Global Lock Key register is a write-only register that is used to unlock the protection for special test bits.

See section 41.9.2. Global Test Modes for Lock key specification.

LOCK[15:0] bits (Lock Key)

The unlock key sequence must be written in the LOCK[15:0] bits to configure the CANFD module in FIFO OTB disable and RAM test modes.

The read value from these bits is always 0x0000.

You cannot write to these bits when the CANFD module is in GL_SLEEP or GL_RESET mode.

Do not write to these bits when the CANFD module is in GL_OPERATION mode.

41.2.57 CFDRPGACCk : RAM Test Page Access Registers k (k = 0 to 63)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0280 + 0x0004 × k

Bit position:	31–0
Bit field:	RDTA[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	RDTA[31:0]	RAM Data Test Access RAM data bytes	R/W

Note: S-TYPE-3, P-TYPE-3

RDTA[31:0] bits (RAM Data Test Access)

Data can be read from or written into the RDTA[31:0] bits when the CANFD module is configured in RAM test mode.

Only write to this bit when the CANFD module is in GL_HALT mode and RAM test mode is enabled.

Software data should be read/written in the RAM Test Page Access registers during RAM test mode.

41.2.58 CFDGAFLIGNENT : Global AFL Ignore Entry Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00C0

Bit position:	31–4	3–0
Bit field:	— —	IRN[3:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0

Bit	Symbol	Function	R/W
3:0	IRN[3:0]	Ignore Rule Number Define rule number which ignores an AFL entry.	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

IRN[3:0] bits (Ignore Rule Number)

The IRN[3:0] bits define the rule number which updates an AFL entry.

Enter only values between 0 and 15 (0x0F) inclusive.

Only write to these bits when the CFDGAFLIGNCTR.IREN bit is 0.

You cannot write to these bits when the CANFD module is in GL_SLEEP mode.

41.2.59 CFDGAFLIGNCTR : Global AFL Ignore Control Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00C4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	IREN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IREN	Ignore Rule Enable 0: AFL entry number is not ignored 1: AFL entry number is ignored	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits control the validity of rewriting the IREN bit.	W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

IREN bit (Ignore Rule Enable)

When the IREN bit is set, the entry number (selected by CFDGAFLIGNENT register) is ignored.

This bit is cleared automatically when the CANFD module is in GL_RESET mode.

KEY[7:0] bits (Key Code)

When 0xC4 is written in the KEY[7:0] bits, a write to the IREN bit is valid.

The read value from these bits is always 0x00.

CFDGAFLIGNCTR.IREN bit and the CFDGAFLIGNCTR.KEY bit should be written simultaneously

41.2.60 CFDRMIEC : RX Message Buffer Interrupt Enable Configuration Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0038

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RMIE[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RMIE[15:0]	RX Message Buffer Interrupt Enable 0: RX Message Buffer Interrupt disabled for corresponding RX message buffer 1: RX Message Buffer Interrupt enabled for corresponding RX message buffer	R/W

Note: S-TYPE-3, P-TYPE-3

These bits show the RX Message Buffer Interrupt Enable for the corresponding RX Message Buffer. CFDRMIEC bit 0 corresponds to RX Message Buffer 0 and so on.

The bit position of CFDRMIEC corresponds to the buffer number of RXMB.

RMIE[15:0] bit (RX Message Buffer Interrupt Enable)

If this bit is set, then an interrupt will be generated at the end of a successful reception from the corresponding Message Buffer.

For details, see section 41.7.1. Interrupts.

Users cannot write to this bit when the CANFD module is in GL_SLEEP mode.

41.2.61 Message Buffer Component Structure

41.2.61.1 Start Addresses

The start address for each of the Message Buffer component is calculated using the number of related Message Buffer components.

The start addresses for each register in the Message Buffer component are depicted in Table 41.4.

Table 41.4 Message Buffer Component Register Start Addresses

b = Message buffer component index	MBCP	p	Register	Start Address
[0…15] b = [0…7]	RMBCPb[0]	x	RMID	0x0920 + b × 0x004C
		x	RMPTR	0x0924 + b × 0x004C
		x	RMFDSTSb	0x0928 + b × 0x004C
		[1…15]	RMDFbp	0x092C + b × 0x004C + p × 0x0004
[0…15] b = [8…15]	RMBCPb[0]	x	RMIDb	0x0D20 + (b-8) × 0x004C
		x	RMPTRb	0x0D24 + (b-8) × 0x004C
		x	RMFDSTSb	0x0D28 + (b-8) × 0x004C
		[1…15]	RMDFbp	0x0D2C + (b-8) × 0x004C + p × 0x0004
[0…1]	RFMBCPb[0]	x	RFIDb	0x0520 + b × 0x004C
		x	RFPTRb	0x0524 + b × 0x004C
		x	RFFDSTSb	0x0528 + b × 0x004C
		[1…15]	RFDFbp	0x052C + b × 0x004C + p × 0x0004
[0]	CFMBCPb[0]	x	CFID	0x05B8
		x	CFPTR0	0x05BC
		x	CFFDCSTS0	0x05C0
		[1…15]	CFDFp0	0x05C4 + p × 0x0004
[0…3]	TMBCPb[0]	x	TMIDb	0x0604 + b × 0x004C
		x	TMPTRb	0x0608 + b × 0x004C
		x	TMFDCTRb	0x060C + b × 0x004C
		[1…15]	TMDFbp	0x0610 + b × 0x004C + p × 0x0004

The message buffer configuration consists of four types of Message Buffer components:

- RX Message Buffer Component (CFDRMBCPb[0])
- RX FIFO Access Message Buffer Component (CFDRFMBCPb[0])
- Common FIFO Access Message Buffer Component (CFDCFMBCP0[0])
- TX Message Buffer Component (CFDTMBCPb[0]).

Where b = the Message Buffer component index that has a range that varies based on the type of Message Buffer component.

For a summary of this configuration, see Figure 41.28. For a detailed description of the number of and the different types of message buffers, see section 41.6. FIFO Buffers and Normal Message Buffer Configuration.

As described in section 41.2. Register Descriptions, each Message Buffer component consists of the following registers:

- Identifier (ID)
- Pointer (PTR)
- Data Field (DFp).

Where p = the Data Field register index that has a range that varies based on the type of message buffer component.

Rc is the Message Buffer Component register where c = Message Buffer Component register index that has a range that varies based on the type of Message Buffer component.

A description of the registers, their associated bits and their accessibility are shown below the summary and detailed figures of each component.

In each of the figures, a cell that contains '-' means reserved and has the same behavior as reserved bits for registers in section 41.2.61. Message Buffer Component Structure.

41.2.61.2 CFDRMBCPb[0] : RX Message Buffer Component b (b = 0 to 15)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: See Table 41.4

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	Rc[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	Rc[15:0]	RX Message Buffer Component c Refer to Table 41.5, Table 41.6 and the descriptions that follow for a detailed description of each register and its related bits, contained within this message buffer component.	R/W

Note: S-TYPE-3, P-TYPE-3

Where the total number of CFDRMBCPb = 16 as shown in Figure 41.28 (c = RX Message Buffer Component Register index = [0…18])

Rc[15:0] bit (RX Message Buffer Component c)

The RX Message Buffer Component is made up of the following registers: CFDRMIDb, CFDRMPTRb, CFDRMFDSTSb, and CFDRMDFbp. Refer to Table 41.6 for details of how to interpret the structure of this buffer component and how to access the respective registers.

Table 41.5 RX Message Buffer Component Summary (1 of 2)

RX Message Buffer Component (RMBCP)	
Rc	**CANFD mode (CAN_FD_MODE = 1'b1)**
R0	RX Message Buffer (b) ID Registers
R1	RX Message Buffer (b) Pointer Registers
R2	RX Message Buffer (b) CANFD Status Registers
R3	RX Message Buffer (b) Data Field 0 Registers
R4	RX Message Buffer (b) Data Field 1 Registers
R5	RX Message Buffer (b) Data Field 2 Registers
R6	RX Message Buffer (b) Data Field 3 Registers

Table 41.5 RX Message Buffer Component Summary (2 of 2)

RX Message Buffer Component (RMBCP)	
Rc	**CANFD mode (CAN_FD_MODE = 1'b1)**
R7	RX Message Buffer (b) Data Field 4 Registers
R8	RX Message Buffer (b) Data Field 5 Registers
R9	RX Message Buffer (b) Data Field 6 Registers
R10	RX Message Buffer (b) Data Field 7 Registers
R11	RX Message Buffer (b) Data Field 8 Registers
R12	RX Message Buffer (b) Data Field 9 Registers
R13	RX Message Buffer (b) Data Field 10 Registers
R14	RX Message Buffer (b) Data Field 11 Registers
R15	RX Message Buffer (b) Data Field 12 Registers
R16	RX Message Buffer (b) Data Field 13 Registers
R17	RX Message Buffer (b) Data Field 14 Registers
R18	RX Message Buffer (b) Data Field 15 Registers
R[19…31]	—

Table 41.6 RX Message Buffer Component (RMBCP) Detailed

<table>
<tr><th>Rc</th><th>p</th><th>Symbol</th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>R0</td><td>x</td><td>CFDRMIDb</td><td>RMIDE</td><td>RMRTR</td><td>—</td><td colspan="29">RMID</td></tr>
<tr><td>R1</td><td>x</td><td>CFDRMPTRb</td><td colspan="4">RMDLC</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td colspan="16">RMTS</td></tr>
<tr><td>R2</td><td>x</td><td>CFDRMFDSTSb</td><td colspan="16">RMPTR</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td colspan="2">RMIFL</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>RMFDF</td><td>RMBRS</td><td>RMESI</td></tr>
<tr><td>R3</td><td>0</td><td>CFDRMDFbp</td><td colspan="8">RMDB_HH</td><td colspan="8">RMDB_HL</td><td colspan="8">RMDB_LH</td><td colspan="8">RMDB_LL</td></tr>
<tr><td>R[4…18]</td><td>[1…15]</td><td>CFDRMDFbp</td><td colspan="8">RMDB_HH</td><td colspan="8">RMDB_HL</td><td colspan="8">RMDB_LH</td><td colspan="8">RMDB_LL</td></tr>
</table>

41.2.61.3 CFDRMIDb : RX Message Buffer ID Registers (b = 0 to 15)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0920 + 0x004C × b (b = 0 to 7)
0x0D20 + 0x004C × (b - 8) (b = 8 to 15)

<table>
<tr><td>Bit position:</td><td>31</td><td>30</td><td>29</td><td colspan="29">0</td></tr>
<tr><td>Bit field:</td><td>RMIDE</td><td>RMRTR</td><td>—</td><td colspan="29">RMID[28:0]</td></tr>
<tr><td>Value after reset:</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

Bit	Symbol	Function	R/W
28:0	RMID[28:0]	RX Message Buffer ID Field STD-ID/EXT-ID fields	R
29	—	This bit is read as 0.	R

Bit	Symbol	Function	R/W
30	RMRTR	RX Message Buffer RTR Bit 0: Data frame 1: Remote frame	R
31	RMIDE	RX Message Buffer IDE Bit 0: STD-ID is stored 1: EXT-ID is stored	R

Note: S-TYPE-3, P-TYPE-3

The RX Message Buffer ID Register b (b = 0 to 15) store the ID field, IDE bit, and RTR bit of the received message.

RMID[28:0] bits (RX Message Buffer ID Field)

The RMID[28:0] are the bits of the STD-ID/EXT-ID fields of the message stored in the RX message buffer.

See section 41.2.61.1. Start Addresses for details on how to interpret the structure of this buffer component.

RMRTR bit (RX Message Buffer RTR Bit)

The RMRTR bit shows whether a data frame or a remote frame was stored in the RX message buffer.

Note: There are no remote frames in CANFD format. When a CANFD frame is received, the register reflects the state of the received value (the RRS bit in FD frame format).

RMIDE bit (RX Message Buffer IDE Bit)

The RMIDE bit shows whether message with Standard Identifier or Extended Identifier was stored in the RX message buffer.

41.2.61.4 CFDRMPTRb : RX Message Buffer Pointer Registers (b = 0 to 15)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0924 + 0x004C × b (b = 0 to 7)
0x0D24 + 0x004C × (b - 8) (b = 8 to 15)

Bit position:	31–28	27	26	25	24	23	22	21	20	19	18	17	16	15–0
Bit field:	RMDLC[3:0]	—	—	—	—	—	—	—	—	—	—	—	—	RMTS[15:0]
Value after reset:	0 0 0 0	0	0	0	0	0	0	0	0	0	0	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	RMTS[15:0]	RX Message Buffer Timestamp Field Timestamp value stored for the message in the RX message buffer	R
27:16	—	These bits are read as 0.	R
31:28	RMDLC[3:0]	RX Message Buffer DLC Field Number of data bytes received in a CAN frame.	R

Note: S-TYPE-3, P-TYPE-3

The RX Message Buffer Pointer Register b (b = 0 to 15) store the DLC and Timestamp fields for the received message.

RMTS[15:0] bits (RX Message Buffer Timestamp Field)

The RMTS[15:0] bits store the timestamp value taken at the capture point as configured by CFDGFDCFG.TSCCFG of the received message.

RMDLC[3:0] bits (RX Message Buffer DLC Field)

The RMDLC[3:0] bits store the number of data bytes that were received in the RX message buffer.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes that were received.

Note: The maximum capacity of the buffer belongs to CFDRMNB.RMPLS and this is not available in the classical CAN function.

41.2.61.5 CFDRMFDSTSb : RX Message Buffer CANFD Status Registers (b = 0 to 15)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0928 + 0x004C × b (b = 0 to 7)
0x0D28 + 0x004C × (b - 8) (b = 8 to 15)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RMPTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	RMIFL[1:0]		—	—	—	—	—	RMFDF	RMBRS	RMESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RMESI*1	Error State Indicator bit 0: CANFD frame received from error active node 1: CANFD frame received from error passive node	R
1	RMBRS*1	Bit Rate Switch bit 0: CANFD frame received with no bit rate switch 1: CANFD frame received with bit rate switch	R
2	RMFDF*1	CAN FD Format bit 0: Non CANFD frame received 1: CANFD frame received	R
7:3	—	These bits are read as 0.	R
9:8	RMIFL[1:0]	RX Message Buffer Information Label Field	R
15:10	—	These bits are read as 0.	R
31:16	RMPTR[15:0]	RX Message Buffer Pointer Field	R

Note: S-TYPE-3, P-TYPE-3
Note 1. This bit is not available in the classical CAN function.

The RX Message Buffer CANFD Status Register b (b = 0 to 15) show the status of the FDF, BRS and ESI bits, and pointer of the received CANFD frame.

RMESI bit (Error State Indicator bit)

The RMESI bit has the same value as the ESI bit of the received CANFD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

RMBRS bit (Bit Rate Switch bit)

The RMBRS bit has the same value as the BRS bit of the received CANFD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

RMFDF bit (CAN FD Format bit)

The RMFDF bit has the same value as the FDF bit of the received CANFD frame.

Note: This bit is not available in the classical CAN function.

RMIFL[1:0] bits (RX Message Buffer Information Label Field)

The RMIFL[1:0] bits store the information label value from the related Global Acceptance Filter List entry.

RMPTR[15:0] bits (RX Message Buffer Pointer Field)

The RMPTR[15:0] bits store the pointer value from the related Global Acceptance Filter List entry.

41.2.61.6 CFDRMDFb_p : RX Message Buffer Data Field p Registers (p = 0 to 15, b = 0 to 15)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x092C + 0x004C × b + 0x0004 × p (b = 0 to 7, p = 0 to 15)
0x0D2C + 0x004C × (b - 8) + 0x0004 × p (b = 8 to 15, p = 0 to 15)

Bit position:	31–24	23–16	15–8	7–0
Bit field:	RMDB_HH[7:0]	RMDB_HL[7:0]	RMDB_LH[7:0]	RMDB_LL[7:0]
Value after reset:	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
7:0	RMDB_LL[7:0]	RX Message Buffer Data Byte (p × 4)	R
15:8	RMDB_LH[7:0]	RX Message Buffer Data Byte ((p × 4) + 1)	R
23:16	RMDB_HL[7:0]	RX Message Buffer Data Byte ((p × 4) + 2)	R
31:24	RMDB_HH[7:0]*1	RX Message Buffer Data Byte ((p × 4) + 3)	R

Note: S-TYPE-3, P-TYPE-3
Note 1. These bits are not available in the classical CAN function.

The RX Message Buffer Data Field p Register b (p = 0 to 15, b = 0 to 15) store the data bytes (p × 4) to data bytes ((p × 4) + 3) of the received message.

RMDB_LL[7:0] bits (RX Message Buffer Data Byte (p × 4))

The RMDB_LL[7:0] bits store data bytes (p × 4) of the message in the RX message buffer.

Unused data bytes are filled with 0x00.

RMDB_LH[7:0] bits (RX Message Buffer Data Byte ((p × 4) + 1))

The RMDB_LH[7:0] bits store data bytes ((p × 4) + 1) of the message in the RX message buffer.

Unused Data Bytes will be filled with 0x00.

RMDB_HL[7:0] bits (RX Message Buffer Data Byte ((p × 4) + 2))

The RMDB_HL[7:0] bits store data bytes ((p × 4) + 2) of the message in the RX message buffer.

Unused data bytes are filled with 0x00.

RMDB_HH[7:0] bits (RX Message Buffer Data Byte ((p × 4) + 3))

The RMDB_HH[7:0] bits store data bytes ((p × 4) + 3) of the message in the RX message buffer.

Unused data bytes are filled with 0x00.

41.2.61.7 CFDRFMBCPb[0] : RX FIFO Access Message Buffer Component b (b = 0 to 1)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: see Table 41.4

Bit position:	31–0
Bit field:	Rc[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	Rc[31:0]	RX FIFO Access Message Buffer Component c See Table 41.7, Table 41.8 and the descriptions that follow for a detailed description of each register and its related bits, contained within this message buffer component.	R

Note: S-TYPE-3, P-TYPE-3

Where the total number of CFDRFMBCPb = 2 as shown in Figure 41.28 (c = RX FIFO Access Message Buffer Component Register index = [0…18])

Rc[31:0] bits (RX FIFO Access Message Buffer Component c)

The RX FIFO Access Message Buffer component comprises the following registers:

- CFDRFIDb
- CFDRFPTRb
- CFDRFFDSTSb
- CFDRFDFbp

See Table 41.8 for details on how to interpret the structure of this buffer component and how to access the respective registers.

Table 41.7 RX FIFO Access Message Buffer component summary

Rc	
R0	RX FIFO Access ID Registers
R1	RX FIFO Access Pointer Register
R2	RX FIFO Access CANFD Status Registers
R3	RX FIFO Access Data Field 0 Registers
R4	RX FIFO Access Data Field 1 Registers
R5	RX FIFO Access Data Field 2 Registers
R6	RX FIFO Access Data Field 3 Registers
R7	RX FIFO Access Data Field 4 Registers
R8	RX FIFO Access Data Field 5 Registers
R9	RX FIFO Access Data Field 6 Registers
R10	RX FIFO Access Data Field 7 Registers
R11	RX FIFO Access Data Field 8 Registers
R12	RX FIFO Access Data Field 9 Registers
R13	RX FIFO Access Data Field 10 Registers
R14	RX FIFO Access Data Field 11 Registers
R15	RX FIFO Access Data Field 12 Registers
R16	RX FIFO Access Data Field 13 Registers
R17	RX FIFO Access Data Field 14 Registers
R18	RX FIFO Access Data Field 15 Registers
R[19...31]	—

Table 41.8 RX Message Buffer Component (RMBCP) Detailed (1 of 2)

<table>
<tr><th>Rc</th><th>p</th><th>Symbol</th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>R0</td><td>x</td><td>CFDRMIDb</td><td>RMIDE</td><td>RMRTR</td><td>—</td><td colspan="29">RMID</td></tr>
</table>

Table 41.8 RX Message Buffer Component (RMBCP) Detailed (2 of 2)

Rc	p	Symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R1	x	CFDRM PTRb	RMDLC				—	—	—	—	—	—	—	—	—	—	—	—	RMTS															
R2	x	CFDRM FDSTSb	RMPTR																—	—	—	—	—	—	RMIFL		—	—	—	—	—	RMFDF	RMBRS	RMESI
R3	0	CFDRM DFbp	RMDB_HH								RMDB_HL								RMDB_LH								RMDB_LL							
R[4… 18]	[1… 15]	CFDRM DFbp	RMDB_HH								RMDB_HL								RMDB_LH								RMDB_LL							

41.2.61.8 CFDRFIDb : RX FIFO Access ID Register b (b = 0 to 1)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0520 + 0x004C × b

Bit position:	31	30	29	28–0
Bit field:	RFIDE	RFRTR	—	RFID[28:0]
Value after reset:	0	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
28:0	RFID[28:0]	RX FIFO Buffer ID Field STD-ID/EXT-ID fields	R
29	—	This bit is read as 0.	R
30	RFRTR	RX FIFO Buffer RTR bit 0: Data frame 1: Remote frame	R
31	RFIDE	RX FIFO Buffer IDE bit 0: STD-ID has been received 1: EXT-ID has been received	R

Note: S-TYPE-3, P-TYPE-3

The RX FIFO Access ID Registers b (b = 0 to 1) store the ID field, IDE bit and RTR bit of the message.

RFID[28:0] bits (RX FIFO Buffer ID Field)

The RFID[28:0] bits are the bits of the STD-ID/EXT-ID fields of the message in the FIFO buffer.

For alignment of these bits in standard and extended frame format, see Identifier Bits Alignment.

RFRTR bit (RX FIFO Buffer RTR bit)

The RFRTR bit shows whether a data frame or a remote frame was stored in the FIFO buffer.

Note: There are no remote frames in CANFD format. When a CANFD frame was received, the register reflects the state of the received value (RRS bit in FD frame format).

RFIDE bit (RX FIFO Buffer IDE bit)

The RFIDE bit shows whether message with the Standard Identifier or Extended Identifier was received in the FIFO buffer.

41.2.61.9 CFDRFPTRb : RX FIFO Access Pointer Register b (b = 0 to 1)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0524 + 0x004C × b

Bit position:	31–28	27	26	25	24	23	22	21	20	19	18	17	16	15–0
Bit field:	RFDLC[3:0]	—	—	—	—	—	—	—	—	—	—	—	—	RFTS[15:0]
Value after reset:	0 0 0 0	0	0	0	0	0	0	0	0	0	0	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	RFTS[15:0]	RX FIFO Timestamp Value Timestamp value of the received CAN frame	R
27:16	—	These bits are read as 0.	R
31:28	RFDLC[3:0]	RX FIFO Buffer DLC Field Number of data bytes received in a CAN frame	R

Note: S-TYPE-3, P-TYPE-3

The FIFO Access Pointer Registers b (b = 0 to 1) store the DLC and Timestamp fields for the received message.

RFTS[15:0] bits (RX FIFO Timestamp Value)

The RFTS[15:0] bits store the timestamp value taken at the capture point as configured by the CFDGFDCFG.TSCCFG bit of the received message.

RFDLC[3:0] bits (RX FIFO Buffer DLC Field)

The RFDLC[3:0] bits store the number of data bytes that were received in the RX FIFO buffer.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes that were received.

41.2.61.10 CFDRFFDSTSb : RX FIFO Access CANFD Status Register b (b = 0 to 1)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0528 + 0x004C × b

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CFDRFPTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	RFIFL[1:0]		—	—	—	—	—	RFFDF	RFBRS	RFESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFESI*1	Error State Indicator bit 0: CANFD frame received from error active node 1: CANFD frame received from error passive node	R
1	RFBRS*1	Bit Rate Switch bit 0: CANFD frame received with no bit rate switch 1: CANFD frame received with bit rate switch	R
2	RFFDF*1	CAN FD Format bit 0: Non CANFD frame received 1: CANFD frame received	R
7:3	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
9:8	RFIFL[1:0]	RX FIFO Buffer Information Label Field	R
15:10	—	These bits are read as 0.	R
31:16	CFDRFPTR[15:0]	RX FIFO Buffer Pointer Field	R

Note: S-TYPE-3, P-TYPE-3
Note 1. This bit is not available in the classical CAN function.

The RX FIFO Access CANFD Status Registers b (b = 0 to 1) show the status of the FDF, BRS, and ESI bits, including the pointer of the received CANFD frame.

RFESI bit (Error State Indicator bit)

The RFESI bit has the same value as the ESI bit of the received CANFD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

RFBRS bit (Bit Rate Switch bit)

The RFBRS bit has the same value as the BRS bit of the received CANFD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

RFFDF bit (CAN FD Format bit)

The RFFDF bit has the same value as the FDF bit of the received CANFD frame.

Note: This bit is not available in the classical CAN function.

RFIFL[1:0] bits (RX FIFO Buffer Information Label Field)

The RFIFL[1:0] bits store the information label value from the related Global Acceptance Filter List entry.

CFDRFPTR[15:0] bits (RX FIFO Buffer Pointer Field)

The CFDRFPTR[15:0] bits store the pointer value from the related Global Acceptance Filter List entry.

41.2.61.11 CFDRFDFb_p : RX FIFO Access Data Field p Register b (p = 0 to 15, b = 0 to 1)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x052C + 0x004 × p + 0x04C × b

Bit position:	31–24	23–16	15–8	7–0
Bit field:	RFDB_HH[7:0]	RFDB_HL[7:0]	RFDB_LH[7:0]	RFDB_LL[7:0]
Value after reset:	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
7:0	RFDB_LL[7:0]	RX FIFO Buffer Data Byte (p × 4)	R
15:8	RFDB_LH[7:0]	RX FIFO Buffer Data Byte ((p × 4) + 1)	R
23:16	RFDB_HL[7:0]	RX FIFO Buffer Data Byte ((p × 4) + 2)	R
31:24	RFDB_HH[7:0]	RX FIFO Buffer Data Byte ((p × 4) + 3)	R

Note: S-TYPE-3, P-TYPE-3

The RX FIFO Access Data Field p Registers b (p = 0 to 15, b = 0 to 1) store data bytes (p × 4) to data byte ((p × 4) + 3) of the received message.

RFDB_LL[7:0] bits (RX FIFO Buffer Data Byte (p × 4))

The RFDB_LL[7:0] bits store data bytes (p × 4) of the message present in the FIFO buffer.

Unused data bytes are filled with 0x00 according to the configured data payload size CFDRFCCa.RFPLS.

RFDB_LH[7:0] bits (RX FIFO Buffer Data Byte ((p × 4) + 1))

The RFDB_LH[7:0] bits store data bytes ((p × 4) + 1) of the message present in the FIFO buffer.

Unused data bytes are filled with 0x00.

RFDB_HL[7:0] bits (RX FIFO Buffer Data Byte ((p × 4) + 2))

The RFDB_HL[7:0] bits store data bytes ((p × 4) + 2) of the message present in the FIFO buffer.

Unused data bytes are filled with 0x00.

RFDB_HH[7:0] bits (RX FIFO Buffer Data Byte ((p × 4) + 3))

The RFDB_HH[7:0] bits store data bytes ((p × 4) + 3) of the message present in the FIFO buffer.

Unused data bytes are filled with 0x00.

41.2.61.12 CFDCFMBCP0[0] : Common FIFO Access Message Buffer Component

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: See Table 41.4

Bit position:	31 … 0
Bit field:	Rc[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	Rc[31:0]	Common FIFO Access Message Buffer Component c Refer to Table 41.9, Table 41.10 and the descriptions that follow for a detailed description of each register and its related bits, contained within this message buffer component.	R

Note: S-TYPE-3, P-TYPE-3

Where the total number of CFDCFMBCP0 = 1 as shown in Figure 41.28 (c = Common FIFO Message Buffer Component Register index = [0…18])

Rc[31:0] bit (Common FIFO Access Message Buffer Component c)

The Common FIFO Access Message Buffer Component is made up of the following registers: CFDCFID, CFDCFPTR, CFFDSTS0, and CFDCFDFp. Refer to Table 41.10 for details of how to interpret the structure of this buffer component and how to access the respective registers.

Table 41.9 Common FIFO Access Message Buffer Component Summary (1 of 2)

Common FIFO Access Message Buffer Component (CFMBCP)	
Rc	**CANFD mode (CAN_FD_MODE = 1)**
R0	Common FIFO Access ID Registers
R1	Common FIFO Access Pointer Register
R2	Common FIFO Access CANFD Status Registers
R3	Common FIFO Access Data Field 0 Registers
R4	Common FIFO Access Data Field 1 Registers
R5	Common FIFO Access Data Field 2 Registers
R6	Common FIFO Access Data Field 3 Registers
R7	Common FIFO Access Data Field 4 Registers

Table 41.9 Common FIFO Access Message Buffer Component Summary (2 of 2)

Common FIFO Access Message Buffer Component (CFMBCP)	
Rc	**CANFD mode (CAN_FD_MODE = 1)**
R8	Common FIFO Access Data Field 5 Registers
R9	Common FIFO Access Data Field 6 Registers
R10	Common FIFO Access Data Field 7 Registers
R11	Common FIFO Access Data Field 8 Registers
R12	Common FIFO Access Data Field 9 Registers
R13	Common FIFO Access Data Field 10 Registers
R14	Common FIFO Access Data Field 11 Registers
R15	Common FIFO Access Data Field 12 Registers
R16	Common FIFO Access Data Field 13 Registers
R17	Common FIFO Access Data Field 14 Registers
R18	Common FIFO Access Data Field 15 Registers
R[19…31]	—

Table 41.10 Common FIFO Access Message Buffer Component (CFMBCP) Detailed

Rc	p	Symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R0	x	CFDCFID	CFIDE	CFRTR	THLEN	CFID																												
R1	x	CFDCFPTR	CFDLC				—	—	—	—	—	—	—	—	—	—	—	—	CFTS															
R2	x	CFDCFFDCSTS	CFPTR																—	—	—	—	—	—	CFIFL		—	—	—	—	—	CFFDF	CFBRS	CFESI
R3	0	CFDCFDFp	CFDB_HH								CFDB_HL								CFDB_LH								CFDB_LL							
R[4…18]	[1…15]	CFDCFDFp	CFDB_HH								CFDB_HL								CFDB_LH								CFDB_LL							
R[19…31]	x	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

41.2.61.13 CFDCFID : Common FIFO Access ID Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x05B8

Bit position:	31	30	29	28 … 0
Bit field:	CFIDE	CFRTR	THLEN	CFID[28:0]

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
28:0	CFID[28:0]	Common FIFO Buffer ID Field STD-ID / EXT-ID fields	R/W

Bit	Symbol	Function	R/W
29	THLEN	THL Entry enable TX FIFO Mode: 0: Entry will not be stored in THL after successful TX. 1: Entry will be stored in THL after successful TX. RX FIFO Mode: Reserved, this bit is read as 0	R/W
30	CFRTR	Common FIFO Buffer RTR Bit 0: Data Frame 1: Remote Frame	R/W
31	CFIDE	Common FIFO Buffer IDE Bit 0: STD-ID will be transmitted or has been received 1: EXT-ID will be transmitted or has been received	R/W

Note: S-TYPE-3, P-TYPE-3

The Common FIFO Access ID registers store the ID field, IDE bit and RTR bit of the message.

In TX mode, users can read data from the FIFO, only for the current entry based on the write pointer value, not for the other entries.

CFID[28:0] bit (Common FIFO Buffer ID Field)

These are the bits of the STD-ID / EXT-ID fields of the message in the FIFO Buffer.

In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers.

THLEN bit (THL Entry enable)

This bit controls the storage of an entry corresponding to the transmitted message in the TX History list at the end of a successful transmission.

In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers.

CFRTR bit (Common FIFO Buffer RTR Bit)

This bit selects whether a Data Frame or a Remote Frame will be transmitted from or was received in the FIFO Buffer.

Note: There are no remote frames in CANFD format. In case a CANFD frame was received (RX mode) the register reflects the state of the received value (RRS bit in FD frame format). In case of CANFD transmission (TX mode CFDCFID.CFFDF =1) the bit is always transmitted dominant (Data Frame).

In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers.

CFIDE bit (Common FIFO Buffer IDE Bit)

This bit selects whether a message with EXT-ID or STD-ID will be transmitted from or was received in the FIFO Buffer.

In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers.

41.2.61.14 CFDCFPTR : Common FIFO Access Pointer Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x05BC

Bit position:	31 28													15 0
Bit field:	CFDLC[3:0]	—	—	—	—	—	—	—	—	—	—	—	—	CFTS[15:0]
Value after reset:	0 0 0 0	0	0	0	0	0	0	0	0	0	0	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	CFTS[15:0]	Common FIFO Timestamp Value Timestamp value of the received CAN frame (FIFO in RX mode).	R/W
27:16	—	These bits are read as 0. The write value should be 0.	R/W
31:28	CFDLC[3:0]	Common FIFO Buffer DLC Field Number of data bytes received in a CAN frame, or to be transmitted in a CAN frame.	R/W

Note: S-TYPE-3, P-TYPE-3

The Common FIFO Access Pointer Registers store the DLC and Timestamp fields.

In TX mode, you can read data from the FIFO buffer, only for the current entry based on the write pointer value, and not for the other entries.

CFTS[15:0] bits (Common FIFO Timestamp Value)

The CFTS[15:0] bits store the timestamp value taken at the capture point as configured by the CFDGFDCFG.TSCCFG bit of the received message (if FIFO is configured in RX mode).

In TX mode, you can read and write from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

CFDLC[3:0] bits (Common FIFO Buffer DLC Field)

The CFDLC[3:0] bits store the number of data bytes that were received in the FIFO buffer or are to be transmitted.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes.

In TX mode, you can read and write from the FIFO buffers. Do not read data for the other entries in the FIFO when configured in TX mode.

In RX mode, you can only read data from the FIFO buffers.

41.2.61.15 CFDCFFDCSTS : Common FIFO Access CANFD Control/Status Register

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x05C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CFPTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CFIFL[1:0]		—	—	—	—	—	CFFDF	CFBRS	CFESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CFESI*1	Error State Indicator bit 0: CANFD frame received or to transmit by error active node 1: CANFD frame received or to transmit by error passive node	R/W
1	CFBRS*1	Bit Rate Switch bit 0: CANFD frame received or to transmit with no bit rate switch 1: CANFD frame received or to transmit with bit rate switch	R/W
2	CFFDF*1	CAN FD Format bit 0: Non CANFD frame received or to transmit 1: CANFD frame received or to transmit	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
9:8	CFIFL[1:0]	COMMON FIFO Buffer Information Label Field	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
31:16	CFPTR[15:0]	Common FIFO Buffer Pointer Field	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. This bit is not available in the classical CAN function.

The Common FIFO Access CANFD Control/Status Registers show the status of the FDF, BRS and ESI bits, including the pointer of the received CANFD frame or the CANFD frame to transmit.

In TX mode, you can read data from the FIFO, only for the current entry based on the write pointer value, and not for the other entries.

CFESI bit (Error State Indicator bit)

In TX mode, you can read and write from FIFO buffers. In this mode, when the CANFD module is not in error passive, the CFESI bit equals the write value. Otherwise, it is a don’t care, and the bit is transmitted as 1 on the CAN bus, indicating this is an error passive node.

In RX mode, you can only read data from FIFO buffers.

In RX mode, the CFESI bit is updated with the ESI bit value of the CANFD frame when it has been received, indicating the error state of the transmitting node. In RX mode, 0 is stored to this bit when the received FDF bit is 0, this means a CAN 2.0 frame is received.

Note: This bit is not available in the classical CAN function.

CFBRS bit (Bit Rate Switch bit)

In TX mode, you can read and write from FIFO buffers. In this mode, the CANFD module either transmits a 0 to indicate no bit rate switch in the frame to be transmitted or a 1 to indicate a bit rate switch in the frame to be transmitted.

In RX mode, you can only read data from FIFO buffers.

In RX mode, the CFBRS bit is updated with the BRS bit value of the CANFD frame when it has been received, indicating whether there is a bit rate switch (1) or (0) on the CANFD frame.

In RX mode, 0 is stored to the CFBRS bit when the received FDF bit is 0, this means a CAN 2.0 frame is received.

Note: This bit is not available in the classical CAN function.

CFFDF bit (CAN FD Format bit)

In TX mode, you can read and write from FIFO buffers. In this mode, the CANFD module either transmits a 0 to indicate a CAN 2.0 frame is to be transmitted or a 1 to indicate a CANFD frame is to be transmitted.

In RX mode, you can only read data from FIFO buffers.

In RX mode, the CFFDF bit is updated with the FDF bit value of the CAN frame when it has been received, indicating whether it is a CAN 2.0 frame (0) or a CANFD frame (1).

Note: This bit is not available in the classical CAN function.

CFIFL[1:0] bits (COMMON FIFO Buffer Information Label Field)

If the Common FIFO is configured in TX mode, the value programmed in CFDCFFDCSTS.CFIFL[1:0] is stored together with additional message information, to the TX History List after successful transmission of the message.

The information label value from the related Global Acceptance Filter List entry is stored in these bits (if FIFO is configured in either RX mode).

In TX mode, you can read and write from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

CFPTR[15:0] bits (Common FIFO Buffer Pointer Field)

If the Common FIFO is configured in TX mode, the value programmed in CFDCFFDCSTS.CFPTR[15:0] is stored together with additional message information, to the TX History List after successful transmission of the message.

The pointer value from the related Global Acceptance Filter List entry is stored in these bits (if FIFO is configured in either RX mode).

In TX mode, you can read and write from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

41.2.61.16 CFDCFDFp : Common FIFO Access Data Field p Registers (p = 0 to 15)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x05C4 + 0x004 × p

Bit position:	31–24	23–16	15–8	7–0
Bit field:	CFDB_HH[7:0]	CFDB_HL[7:0]	CFDB_LH[7:0]	CFDB_LL[7:0]
Value after reset:	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
7:0	CFDB_LL[7:0]	Common FIFO Buffer Data Bytes (p × 4)	R/W
15:8	CFDB_LH[7:0]	Common FIFO Buffer Data Bytes ((p × 4) + 1)	R/W
23:16	CFDB_HL[7:0]	Common FIFO Buffer Data Bytes ((p × 4) + 2)	R/W
31:24	CFDB_HH[7:0]	Common FIFO Buffer Data Bytes ((p × 4) + 3)	R/W

Note: S-TYPE-3, P-TYPE-3

The FIFO Access Data Field p Registers (p = 0 to 15) store data bytes (p × 4) to data bytes ((p × 4) + 3) of the message.

In TX mode, you can read data from the FIFO, only for the current entry based on the write pointer value, and not for the other entries.

CFDB_LL[7:0] bits (Common FIFO Buffer Data Bytes (p × 4))

The CFDB_LL[7:0] bits store data bytes (p × 4) of the message present in the FIFO buffer.

In TX mode, you can read and write from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In RX mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCC.CFPLS.[*1]

CFDB_LH[7:0] bits (Common FIFO Buffer Data Bytes ((p × 4) + 1))

The CFDB_LH[7:0] bits store data bytes ((p × 4) + 1) of the message present in the FIFO buffer.

In TX mode, you can read and write from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In RX mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCC.CFPLS.[*1]

CFDB_HL[7:0] bits (Common FIFO Buffer Data Bytes ((p × 4) + 2))

The CFDB_HL[7:0] bits store data bytes ((p × 4) + 2) of the message present in the FIFO buffer.

In TX mode, you can read and write from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In RX mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCC.CFPLS.[*1]

CFDB_HH[7:0] bits (Common FIFO Buffer Data Bytes ((p × 4) + 3))

The CFDB_HH[7:0] bits store data bytes ((p × 4) + 3) of the message present in the FIFO buffer.

In TX mode, you can read and write from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In RX mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCC.CFPLS.[*1]

Note 1. In RX mode, unused data bytes are filled with 0x00 according to the configured data payload size CFDCFCC.CFPLS, which is a CANFD feature not found in classical CAN.

41.2.61.17 CFDTMBCPb[0] : TX Message Buffer Component b (b = 0 to 3)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: See Table 41.4

Bit position:	31 … 0
Bit field:	Rc[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	Rc[31:0]	TX Message Buffer Component c Refer to Table 41.11, Table 41.12 and the descriptions that follow for a detailed description of each register and its related bits, contained within this message buffer component.	R

Note: S-TYPE-3, P-TYPE-3

Where the total number of CFDTMBCPn = 4 as shown in Figure 41.28 (c = TX Message Buffer Component Register index = [0…18])

Rc[31:0] bit (TX Message Buffer Component c)

TX Message Buffer Component c

The TX Message Buffer Component is made up of the following registers: CFDTMIDb, CFDTMPTRb, CFDTMFDCTRb, and CFDTMDFbp. Refer to Table 41.12 for details of how to interpret the structure of this buffer component and how to access the respective registers.

Table 41.11 TX Message Buffer Component Summary

TX Message Buffer Component (TMBCP)	
Rc	**CANFD mode (CAN_FD_MODE = 1b)**
R0	TX Message Buffer (b) ID Registers
R1	TX Message Buffer (b) Pointer Registers
R2	TX Message Buffer (b) CANFD Status Registers
R3	TX Message Buffer (b) Data Field 0 Registers
R4	TX Message Buffer (b) Data Field 1 Registers
R5	TX Message Buffer (b) Data Field 2 Registers
R6	TX Message Buffer (b) Data Field 3 Registers
R7	TX Message Buffer (b) Data Field 4 Registers
R8	TX Message Buffer (b) Data Field 5 Registers
R9	TX Message Buffer (b) Data Field 6 Registers
R10	TX Message Buffer (b) Data Field 7 Registers
R11	TX Message Buffer (b) Data Field 8 Registers
R12	TX Message Buffer (b) Data Field 9 Registers
R13	TX Message Buffer (b) Data Field 10 Registers
R14	TX Message Buffer (b) Data Field 11 Registers
R15	TX Message Buffer (b) Data Field 12 Registers
R16	TX Message Buffer (b) Data Field 13 Registers
R17	TX Message Buffer (b) Data Field 14 Registers
R18	TX Message Buffer (b) Data Field 15 Registers
R[19…31]	—

Table 41.12 TX Message Buffer Component (TMBCP) Detailed

Rc	p	Symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R0	x	CFDTMIDb	TMIDE	TMRTR	THLEN	TMID																												
R1	x	CFDTMPTRb	TMDLC				—	—	—	—	—	—	—	—	—	—	—	—	CFTS															
R2	x	CFDTMFDCTRb	TMPTR																—	—	—	—	—	—	TMIFL		—	—	—	—	—	TMFDF	TMBRS	TMESI
R3	0	CFDTMDFbp	TMDB_HH								TMDB_HL								TMDB_LH								TMDB_LL							
R[4…18]	[1…15]	CFDTMDFbp	TMDB_HH								TMDB_HL								TMDB_LH								TMDB_LL							

41.2.61.18 CFDTMIDb : TX Message Buffer ID Registers (b = 0 to 3)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0604 + 0x004C × b

Bit position:	31	30	29	28 to 0
Bit field:	TMIDE	TMRTR	THLEN	TMID[28:0]
Value after reset:	0	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
28:0	TMID[28:0]	TX Message Buffer ID Field STD-ID/EXT-ID fields	R/W
29	THLEN	Tx History List Entry 0: Entry not stored in THL after successful TX 1: Entry stored in THL after successful TX	R/W
30	TMRTR	TX Message Buffer RTR bit 0: Data frame 1: Remote frame	R/W
31	TMIDE	TX Message Buffer IDE bit 0: STD-ID is transmitted 1: EXT-ID is transmitted	R/W

Note: S-TYPE-3, P-TYPE-3

Each TX Message Buffer ID Register b (b = 0 to 3) are used to store the ID, IDE, RTR fields and history configuration of the message to be transmitted from the associated buffer.

TMID[28:0] bits (TX Message Buffer ID Field)

The TMID[28:0] bits are bits of the STD-ID/EXT-ID fields of the message stored in this TX message buffer.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

THLEN bit (Tx History List Entry)

The THLEN bit controls the storage of an entry corresponding to the transmitted message in the TX History list at the end of a successful transmission.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

TMRTR bit (TX Message Buffer RTR bit)

The TMRTR bit selects whether a data frame or remote frame is to be transmitted from this TX message buffer.

Note: There are no remote frames in CANFD format. For a CANFD transmission (CFDTMFDCTRb.CFFDF = 1), this bit is always transmitted dominant (data frame).

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

TMIDE bit (TX Message Buffer IDE bit)

The TMIDE bit selects whether a message with EXT-ID or STD-ID is to be transmitted from this TX message buffer.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

41.2.61.19 CFDTMPTRb : TX Message Buffer Pointer Register (b = 0 to 3)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0608 + 0x004C × b

Bit position:	31–28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TMDLC[3:0]	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0 0 0 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
27:0	—	The read values are undefined. The write value should be 0.	R/W
31:28	TMDLC[3:0]	TX Message Buffer DLC Field Number of data bytes to be transmitted in a CAN frame.	R/W

Note: S-TYPE-3, P-TYPE-3

Each TX Message Buffer Pointer Register b (b = 0 to 3) is used to store the DLC fields of the message to transmit from the associated buffer.

TMDLC[3:0] bits (TX Message Buffer DLC Field)

The TMDLC[3:0] bits select the number of data bytes to be transmitted from this TX message buffer when the corresponding TMRTR bit is configured as 0.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes to be transmitted.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

41.2.61.20 CFDTMFDCTRb : TX Message Buffer CANFD Control Register (b = 0 to 3)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x060C + 0x004C × b

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TMPTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TMIFL[1:0]		—	—	—	—	—	TMFDF	TMBRS	TMESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMESI*1	Error State Indicator bit 0: CANFD frame to transmit by error active node 1: CANFD frame to transmit by error passive node	R/W
1	TMBRS*1	Bit Rate Switch bit 0: CANFD frame to transmit with no bit rate switch 1: CANFD frame to transmit with bit rate switch	R/W
2	TMFDF*1	CAN FD Format bit 0: Non CANFD frame to transmit 1: CANFD frame to transmit	R/W
7:3	—	The read values are undefined. The write value should be 0.	R/W
9:8	TMIFL[1:0]	TX Message Buffer Information Label Field	R/W
15:10	—	The read values are undefined. The write value should be 0.	R/W
31:16	TMPTR[15:0]	TX Message Buffer Pointer Field	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. This bit is not available in the classical CAN function.

The TX Message Buffer CANFD Control Registers b (b = 0 to 3) show the status of the FDF, BRS and ESI bits, including the pointer fields of the CANFD frame to be transmitted.

TMESI bit (Error State Indicator bit)

If the channel is not in error passive, then the TMESI bit equals the write value, otherwise it is a don't care, and the bit is transmitted as 1 on the CAN bus, indicating this is an error passive node.

Do not write to the TMESI bit when the related CANFD channel is in CH_SLEEP mode.

Note: This bit is not available in the classical CAN function.

TMBRS bit (Bit Rate Switch bit)

Do not write to the TMBRS bit when the related CANFD channel is in CH_SLEEP mode.

Note: This bit is not available in the classical CAN function.

TMFDF bit (CAN FD Format bit)

Do not write to the TMFDF bit when the related CANFD channel is in CH_SLEEP mode.

Note: This bit is not available in the classical CAN function.

TMIFL[1:0] bits (TX Message Buffer Information Label Field)

The TMIFL[1:0] bits store the information label value to be copied, together with additional message information, in the TX History List after successful transmission of the message.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

TMPTR[15:0] bits (TX Message Buffer Pointer Field)

The TMPTR[15:0] bits store the pointer value to be copied, together with additional message information in the TX History List after successful transmission of the message.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

41.2.61.21 CFDTMDFb_p : TX Message Buffer Data Field Register (p = 0 to 15, b = 0 to 3)

Base address: CANFDn = 0x4038_0000 + 0x2000 × n (n = 0, 1)
CANFDn_NS = 0x5038_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0610 + 0x004 × p + 0x004C × b

Bit position:	31–24	23–16	15–8	7–0
Bit field:	TMDB_HH[7:0]	TMDB_HL[7:0]	TMDB_LH[7:0]	TMDB_LL[7:0]
Value after reset:	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
7:0	TMDB_LL[7:0]	TX Message Buffer Data Byte (p × 4)	R/W
15:8	TMDB_LH[7:0]	TX Message Buffer Data Byte ((p × 4) + 1)	R/W
23:16	TMDB_HL[7:0]	TX Message Buffer Data Byte ((p × 4) + 2)	R/W
31:24	TMDB_HH[7:0]	TX Message Buffer Data Byte ((p × 4) + 3)	R/W

Note: S-TYPE-3, P-TYPE-3

Each TX Message Buffer Data Field p Register b (p = 0 to 15, b = 0 to 3) is used to store data bytes (p × 4) to data bytes ((p × 4) + 3) of the message to transmit from the associated buffer.

TMDB_LL[7:0] bits (TX Message Buffer Data Byte (p × 4))

TMDB_LL[7:0] bits store data bytes (p × 4) of the message in the TX message buffer.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

TMDB_LH[7:0] bits (TX Message Buffer Data Byte ((p × 4) + 1))

TMDB_LH[7:0] bits store data bytes ((p × 4) + 1) of the message in the TX message buffer.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

TMDB_HL[7:0] bits (TX Message Buffer Data Byte ((p × 4) + 2))

TMDB_HL[7:0] bits store data bytes ((p × 4) + 2) of the message in the TX message buffer.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

TMDB_HH[7:0] bits (TX Message Buffer Data Byte ((p × 4) + 3))

TMDB_HH[7:0] bits store data bytes ((p × 4) + 3) of the message in the TX message buffer.

Do not write to these bits when the related CANFD channel is in CH_SLEEP mode.

41.3 Modes of Operation

41.3.1 Overview

The modes of the CANFD module can be classified into 2 groups:

- Global modes
- Channel modes

41.3.2 Global Modes

These modes are applicable for the complete CANFD module and therefore are called Global modes. The global modes of the CANFD module are:

- Global Sleep
- Global Reset
- Global Halt

- Global Operation.

Figure 41.2 shows the possible transitions between the Global modes.

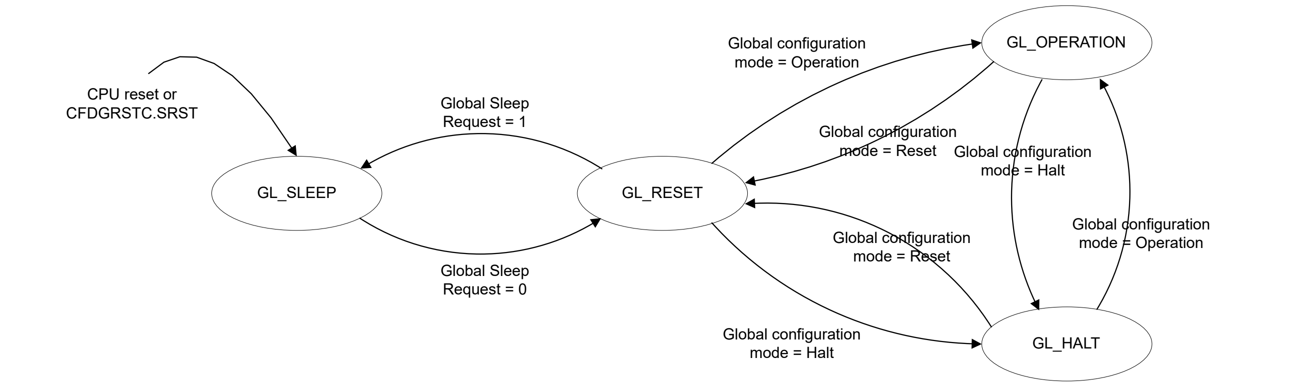


Figure 41.2 Transition between CANFD Global modes

Change in the Global mode can affect the Channel mode. Table 41.13 shows the effect of a Global mode transition on a Channel mode.

Table 41.13 Possible CANFD Channel modes and Global modes

	Target Global mode			
Current Global mode	**Sleep (GL_SLEEP)**	**Reset (GL_RESET)**	**Halt (GL_HALT)**	**Operation (GL_OPERATION)**
Sleep (GL_SLEEP)		Ch-Sleep: Keep Ch-Reset: N/A Ch-Halt: N/A Ch-Oper: N/A		
Reset (GL_RESET)	Ch-Sleep: Keep Ch-Reset: → Ch-Sleep Ch-Halt: N/A Ch-Oper: N/A		Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: N/A Ch-Oper: N/A	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: N/A Ch-Oper: N/A
Halt (GL_HALT)		Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: → Ch-Reset Ch-Oper: N/A		Ch-Sleep:Keep Ch-Reset: Keep Ch-Halt: Keep Ch-Oper: N/A
Operation (GL_OPERATION)		Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: → Ch-Reset Ch-Oper: → Ch-Reset	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: Keep Ch-Oper: → Ch-Halt	

41.3.2.1 Global Sleep Mode

After the release of a hardware reset or after setting and clearing a CFDGRSTC.SRST bit, the CANFD module automatically enters Global Sleep mode.

The CANFD module also enters the Global Sleep mode when the Global Sleep Request bit is set while it is in Global Reset mode. This control bit cannot be set in Global Halt mode or Global Operation mode.

Setting the Global Sleep Request bit sets Channel Sleep Request bit and forces the channel into the Channel Sleep mode.

Sleep mode is used for power saving purpose. When CANFD module is in Global Sleep mode, only the clock for CPU write access to the Global Sleep Mode Request bit is active. All other clocks are stopped and all other functions of the CANFD module are suspended.

Read access from all registers is still possible and all register values are preserved.

After setting the Global Sleep Request bit, it is necessary to confirm that the Global Sleep status has been updated, indicating successful transition to Global Sleep mode before the Global Sleep Request bit can be cleared again.

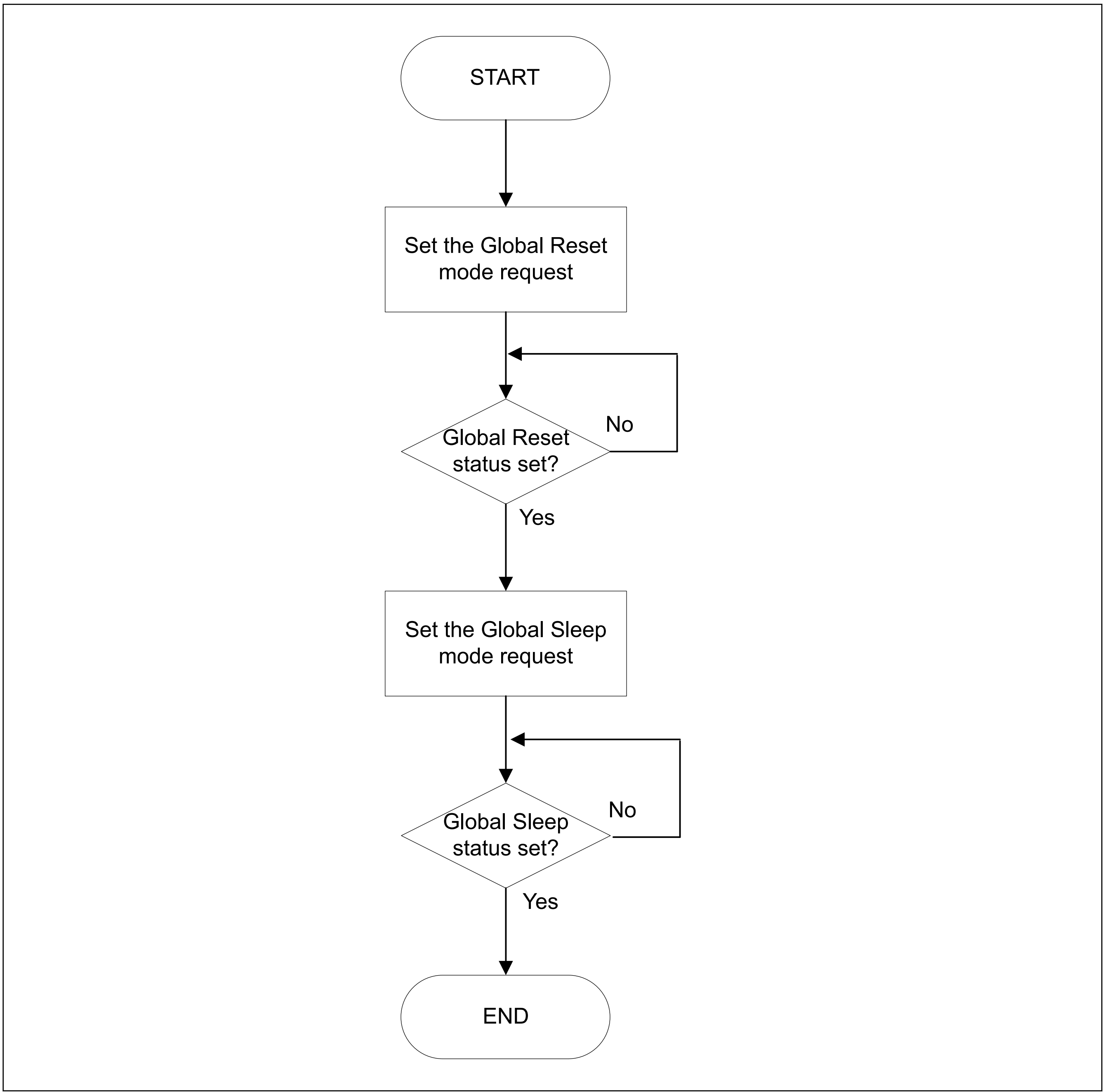


Figure 41.3 Procedure for entering Global Sleep mode

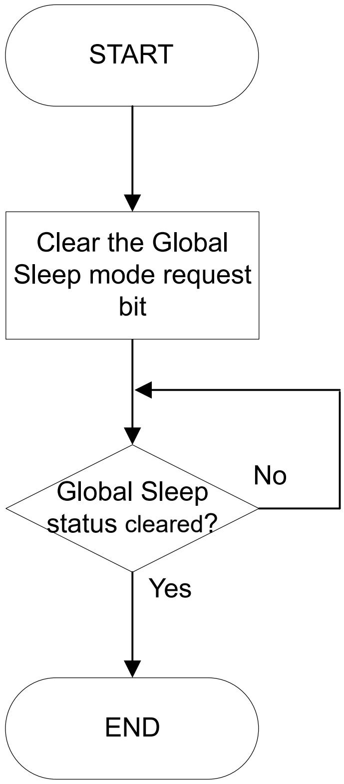


Figure 41.4 Procedure for exiting Global Sleep mode

41.3.2.2 Global Reset Mode

The CANFD module enters this mode in the following ways:

- Global Mode Control bit CFDGCTR.GMDC in the Global Control Register is configured for Global Reset mode while the CANFD module is in Global Halt or Global Operation mode
- Global Sleep Mode Request bit is cleared while CANFD module is in Global Sleep mode.

In Global Reset mode, all CANFD module functions are suspended and all status and flag registers are initialized.

Additionally, all FIFOs and TX Queues are disabled and transmission control bits are cleared.

Configuration registers (except the test mode registers) are not initialized in this mode to their MCU reset values and the CANFD module can be configured.

See section 41.3.4. Global Mode and Channel Mode Transition Interactions for a detailed description of the behavior of all registers when transition to Global Reset mode is performed.

Setting the Global mode to Reset by setting the Global Mode Control bits CFDGCTR.GMDC in the Global Control Register to 01b sets Channel Mode Control bits CFDC0CTR.CHMDC in the Channel Control Registers to 01b and forces the channel into the Channel Reset mode.

For channels that are already in Channel Reset mode or Channel Sleep mode, this automatic transition is not performed (CFDC0CTR.CHMDC of related channel already set to 01b).

After setting Global Mode Control bit CFDGCTR.GMDC to Reset mode, it is necessary to confirm that the Reset Mode Status bit CFDGSTS.GRSTSTS in the Global Status Register has been updated, indicating successful transition to Global Reset mode before CFDGCTR.GMDC can be changed again.

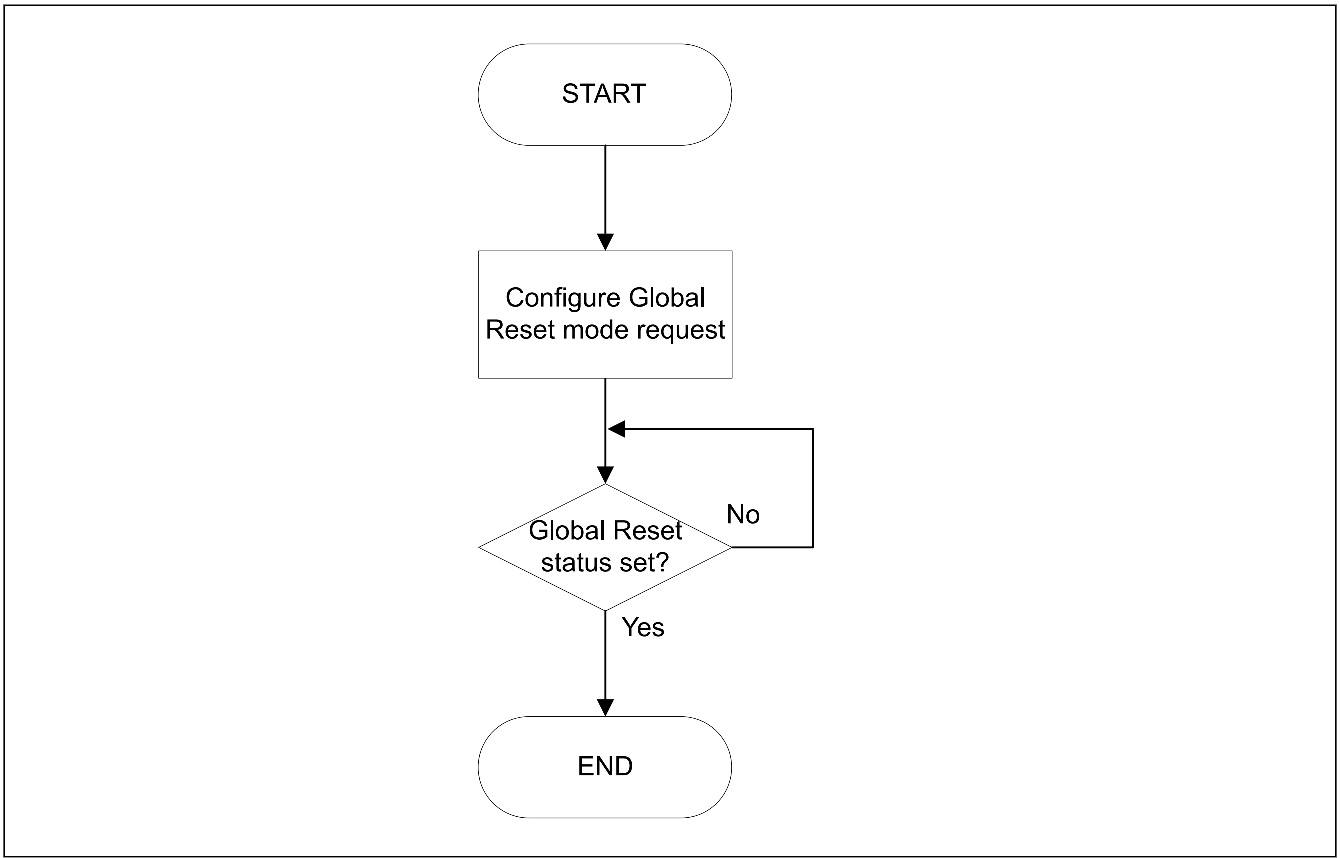


Figure 41.5 Procedure for entering Global Reset mode

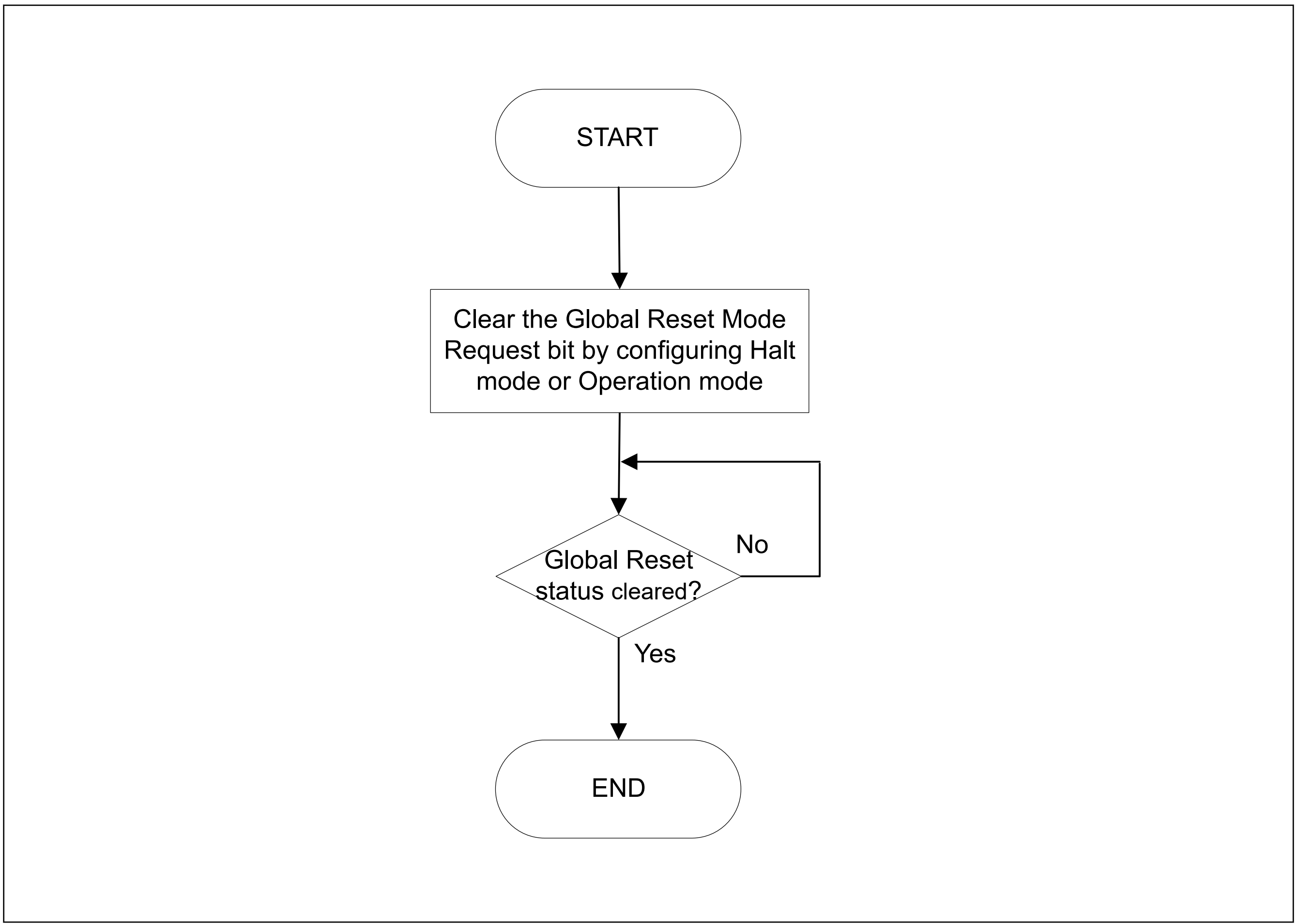


Figure 41.6 Procedure for exiting Global Reset mode

41.3.2.3 Global Halt Mode

The CANFD module enters this mode in the following ways:

- Global Mode Control bit CFDGCTR.GMDC in the Global Control Register is configured for Global Halt mode while the CANFD module is in Global Reset mode:
  - The channel in either Channel Reset or Channel Sleep mode remains in this mode
- Global Mode Control bit CFDGCTR.GMDC in the Global Control Register is configured for Global Halt mode while the CANFD module is in Global Operation mode:
  - The channel in Channel Reset, Channel Halt, or Channel Sleep mode remains in this mode
  - The channel in Channel Operation mode transitions to Channel Halt mode
  - Global Halt Mode Status bit is set when the channel has left Channel Operation mode.

If a transmission or reception is ongoing for a channel, the transition to Channel Halt mode is delayed until completion of the communication.

Similarly, if a channel is in bus-off, the full bus-off recovery sequence may be delayed depending on the channel configuration.

In Global Halt mode, all communications are suspended and CANFD logic does not cause any change to the Status and Flag registers (only when a channel is in the bus-off that its REC and TEC values are cleared). Additionally, the test mode configuration and control registers are not initialized in this mode.

The Global Halt mode should be used to configure global module test modes.

See section 41.3.4. Global Mode and Channel Mode Transition Interactions for a detailed description of the behavior of all registers when transition to Global Halt mode is performed.

Setting the Global mode to Halt by setting the Global Mode Control bit CFDGCTR.GMDC in the Global Control Register to 10b sets Channel Mode Control bits CFDC0CTR.CHMDC in the Channel Control Registers to 10b for the channel that are in Channel Operation mode and forces these channels into the Channel Halt mode.

For the channel that are already in Channel Reset, Channel Halt, or Channel Sleep mode, this automatic transition is not performed.

Therefore, the Global Halt mode request can be used to shut down all CANFD channel communications without loss of messages and disruption on the related CAN bus (no interruption of reception/transmission processes on the channels).

After setting the Global Mode Control bit CFDGCTR.GMDC to Halt mode, it is necessary to confirm that the Halt Mode Status bit CFDGSTS.GHLTSTS in the Global Status Register has been updated to indicate a successful transition to Global Halt mode. Do not specify any other SFR setting until confirming CFDGSTS.GHLTSTS is set.

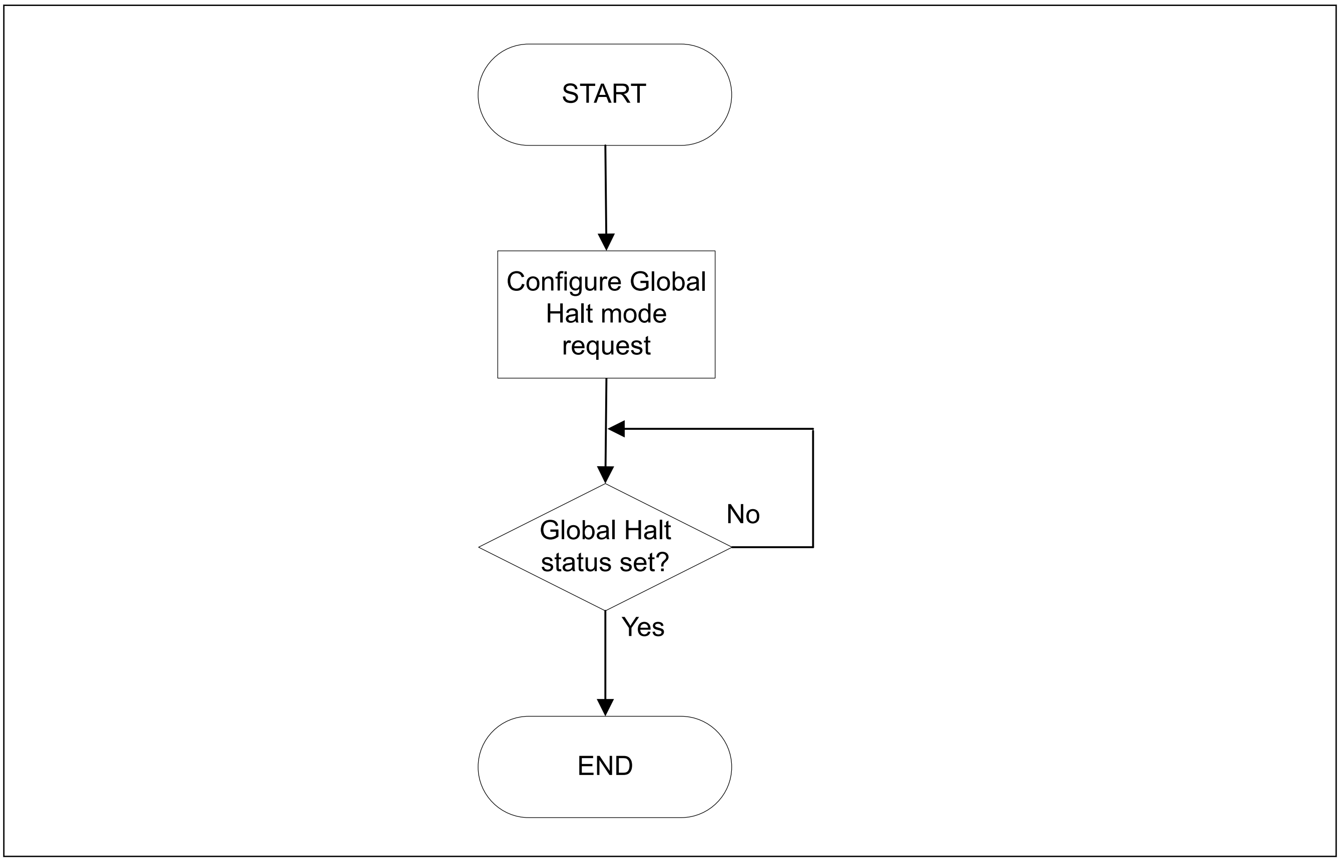


Figure 41.7 Procedure for entering Global Halt mode

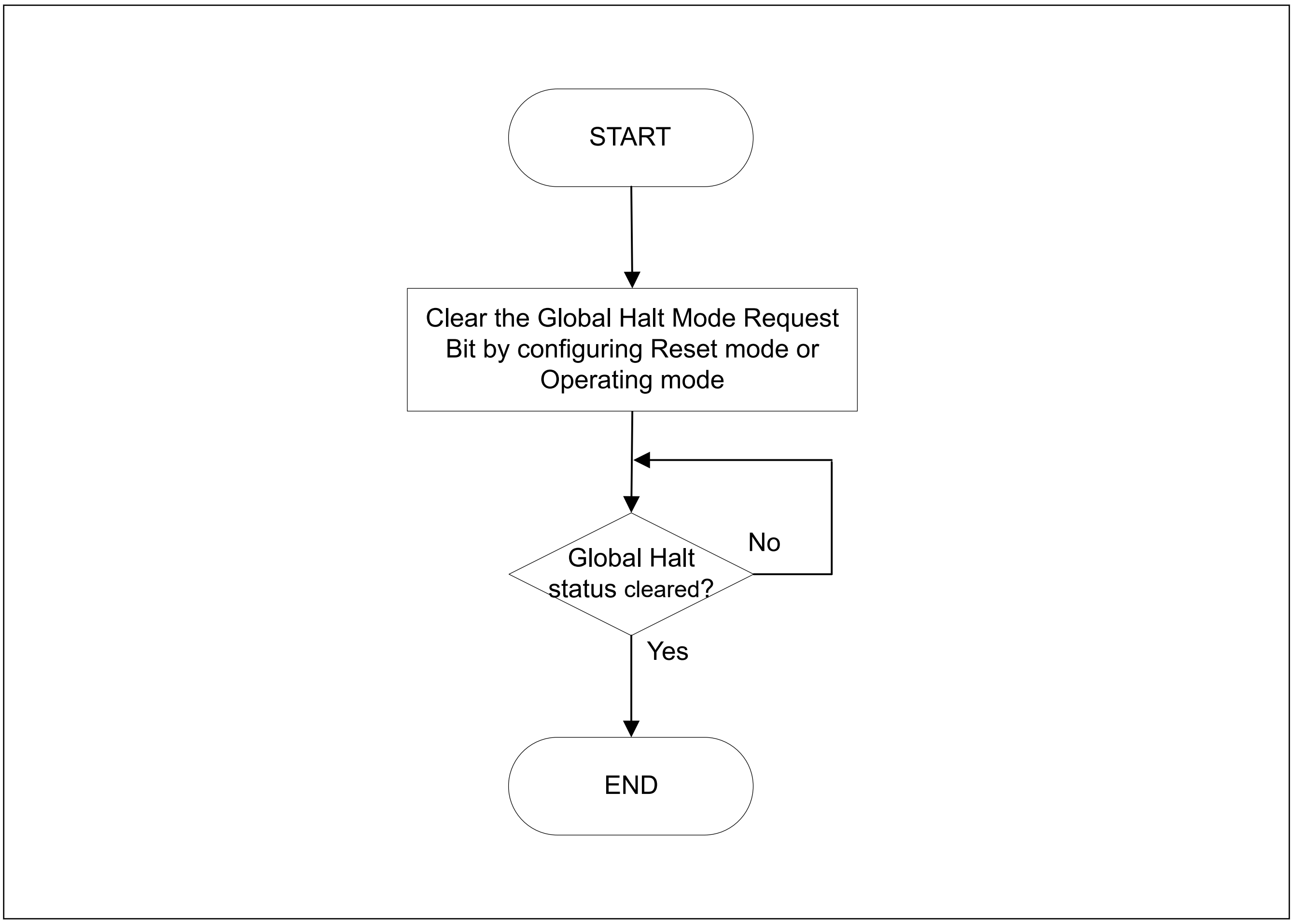


Figure 41.8 Procedure for exiting Global Halt mode

41.3.2.4 Global Operation Mode

The CANFD module enters this mode when the Global Mode Configuration bits are set to Global Operation mode.

The CANFD channel can only be set to Channel Operation mode and start CAN communication when CANFD is in Global Operation mode.

After setting the Global Mode Control bit CFDGCTR.GMDC to Global Operation mode, it is necessary to confirm that the Global Reset Mode Status bit CFDGSTS.GRSTSTS and the Global Halt Mode Status bit CFDGSTS.GHLTSTS in the Global Status Register have been cleared to indicate a successful transition to Global Operation mode before CFDGCTR.GMDC can be modified again.

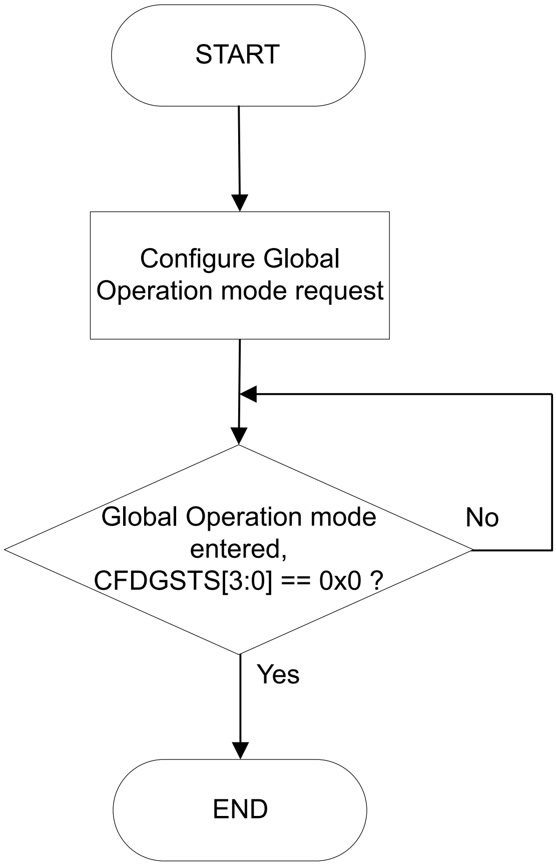


Figure 41.9 Procedure for entering Global Operation mode

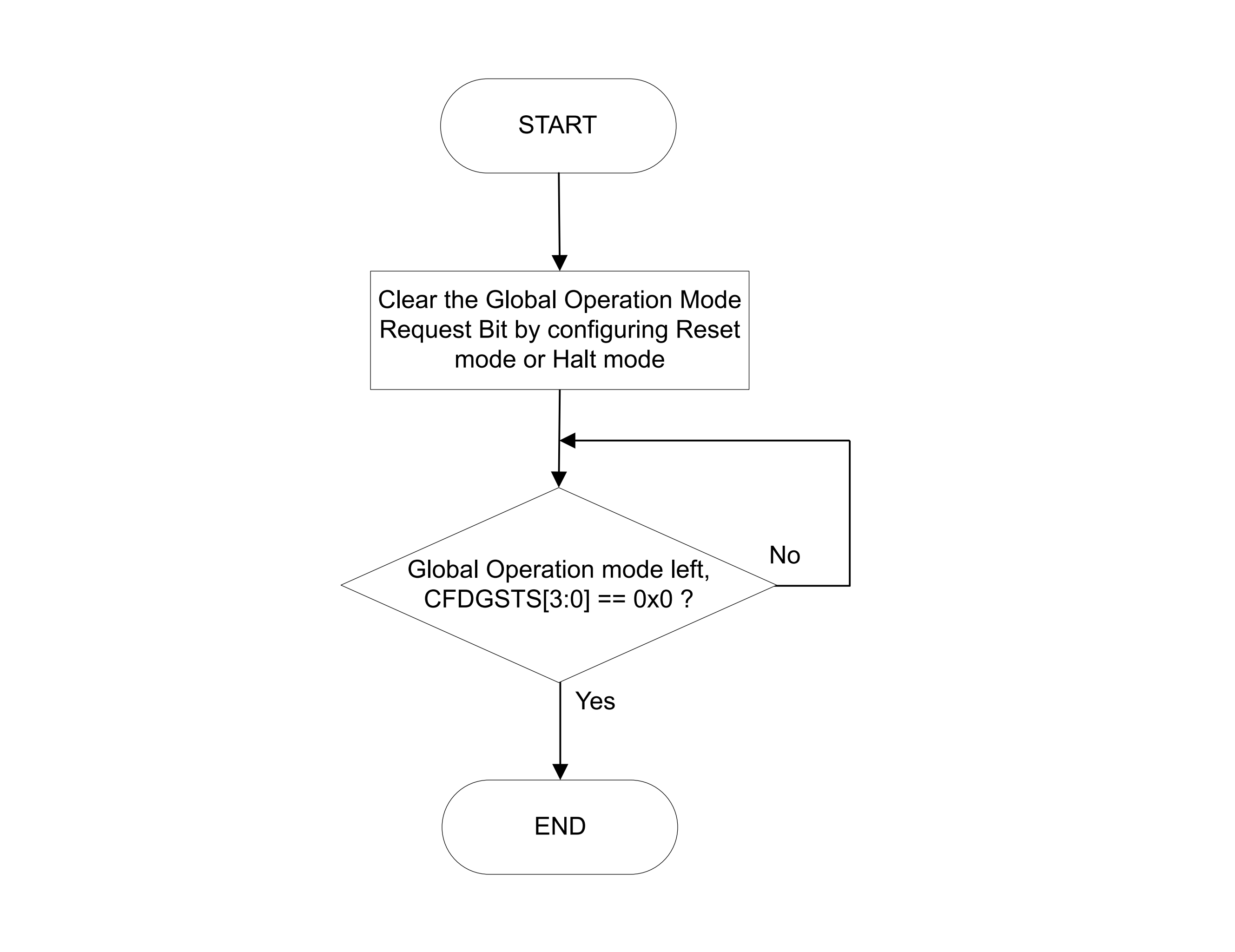


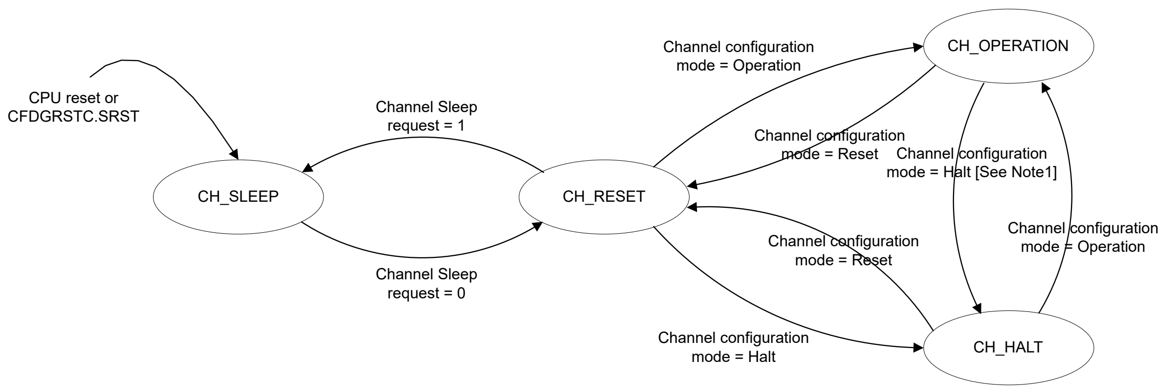
Figure 41.10 Procedure for exiting Global Operation mode

41.3.3 Channel Modes

A CAN channel can be in one of the following four channel modes:

- Reset
- Halt
- Operation
- Sleep.

Figure 41.11 shows the possible transitions between the channel modes.



Note: The transition timing depends on BOM setting in the Channel Configuration Register:

- when BOM = 01b, the state transition timing is immediate after entering bus-off state
- when BOM = 10b, the state transition timing is at the end of bus-off state
- when BOM = 11b, the state transition timing is matching with the setting of CHMDC to Halt mode.

Figure 41.11 Transition between CAN channel modes

41.3.3.1 CAN Channel Sleep Mode

After the release of a hardware reset or after setting and clearing the CFDGRSTC.SRST bit, a CAN channel of the CANFD module automatically enters Channel Sleep mode.

A CAN channel also enters Channel Sleep mode when the related Channel Sleep Mode Request bit is set while the CAN channel is in Channel Reset mode. Do not set this control bit in Channel Halt mode or Channel Operation mode.

Entering the CAN Channel Sleep mode instantly stops the clock supplied to the CAN channel unit and therefore reduces power consumption.

After setting the Channel Sleep Mode Request bit, it is necessary to confirm that the Channel Sleep mode status has been updated to indicate a successful transition to Channel Sleep mode before the Channel Sleep Mode Request bit can be cleared again.

During Channel Sleep mode, do not write to channel related registers. Read operation is still possible.

41.3.3.2 CAN Channel Reset Mode

A CANFD CAN channel enters this mode in the following ways:

- Channel Mode Control bit CFDC0CTR.CHMDC in the Channel Control Registers is configured for Channel Reset mode while the related CAN channel is in Channel Halt mode or Channel Operation mode
- Channel Sleep Mode Request bit is cleared while the related CAN channel is in Channel Sleep mode
- Global Mode Control bit CFDGCTR.GMDC is set to Global Reset mode and CAN channel is not in Channel Sleep mode or Channel Reset mode.

In Channel Reset mode, all CAN channel status and flag registers are initialized.

Additionally, all channel related transmission control bits are cleared and the channel related TX Queue is disabled.

Configuration registers (except the Channel Test Mode registers) are not initialized in this mode and the CAN channel can be configured for communication.

See section 41.3.4. Global Mode and Channel Mode Transition Interactions for a detailed description of the behavior of all registers when transition to Channel Reset mode is performed.

After setting the Channel Mode Control bit CFDC0CTR.CHMDC to Channel Reset mode, it is necessary to confirm that the Reset Mode Status bit CFDC0STS.CRSTSTS in the related Channel Status Registers has been updated to indicate a successful transition to Channel Reset mode before the related CFDC0CTR.CHMDC bit can be modified again.

See Table 41.14 for the behavior of transitioning to Channel Reset mode while CAN communication is ongoing.

41.3.3.3 CAN Channel Halt Mode

A CANFD CAN channel enters this mode in the following ways:

- Channel Mode Control bit CFDC0CTR.CHMDC in the Channel Control Registers is configured for Channel Halt mode while the related CAN channel is in Channel Reset mode or Channel Operation mode
- Global Mode Control bit CFDGCTR.GMDC is set to Global Halt mode and CAN channel is in Channel Operation mode.

In Channel Halt mode, all channel CAN communication is suspended but all status and flag registers remain unchanged during Channel Halt mode entry (except for the bus-off case where REC and TEC values are cleared for this channel).

In addition, the Channel Test Mode Configuration and Control registers are not initialized in this mode.

The Channel Halt mode should be used to configure channel test modes.

See section 41.3.4. Global Mode and Channel Mode Transition Interactions for a detailed description of the behavior of all registers when transition to Channel Halt mode is performed.

After setting the Channel Mode Control bit CFDC0CTR.CHMDC to Channel Halt mode, it is necessary to confirm that the Halt Mode Status bit CFDC0STS.CHLTSTS in the related Channel Status Register has been updated to indicate a successful transition to Channel Halt mode before the related CFDC0CTR.CHMDC can be modified again.

See Table 41.14 for the transition behavior to Channel Halt mode while CAN communication is ongoing.

Table 41.14 Transition behavior in CAN Reset mode and Halt mode

Mode	State		
	Receiver	**Transmitter**	**Bus-Off**
CAN Channel Reset mode (CFDC0CTR.CHMDC = 01b)	The CAN channel enters Channel Reset mode without waiting for the completion of the ongoing reception.*1	The CAN channel enters Channel Reset mode without waiting for the completion of the ongoing transmission.*1	The CAN channel enters Channel Reset mode without waiting for the completion of the bus-off recovery.
CAN Channel Halt mode (CFDC0CTR.CHMDC = 10b)	CAN channel enters Channel Halt mode at the end of the ongoing reception or error.*2	CAN channel enters Channel Halt mode after completion of the ongoing transmission.	When CFDC0CTR.BOM is set to 00b, a Channel Halt mode request is accepted only after the completion of the full bus-off recovery sequence. When CFDC0CTR.BOM is set to 10b, the CAN channel transits automatically to Channel Halt mode after waiting for the completion of the bus-off recovery. When CFDC0CTR.BOM is set to 01b, the CAN channel transits automatically to Channel Halt mode without waiting for the completion of the bus-off recovery. When CFDC0CTR.BOM is set to 11b, the CAN channel enters Channel Halt mode as soon as Channel Halt mode is requested (without waiting for the completion of the bus-off recovery).

Note 1. If the entry to Channel Reset mode is required only at the end of an ongoing communication, then Channel Halt mode can be requested first to prevent interruption of CAN communication by direct transition to Channel Reset mode. After the CAN channel enters Channel Halt mode, the Channel Reset mode can be requested.

Note 2. If CAN communication is locked at dominant level after an error flag, software can detect this situation by monitoring the channel related BusLock flag and resolve lock condition by setting the CAN channel to Channel Reset mode.

41.3.3.4 CAN Channel Operation Mode

The Channel Operation mode is activated by setting the CFDC0CTR.CHMDC bits to 00b. If 11 consecutive recessive bits are detected after entering the CAN Operation mode, the CFDC0STS.COMSTS bit is set and the CAN channel:

- Enables the functions of the channel communication by allowing the channel to become an active node on the CAN network
- Releases the internal fault confinement logic including receive and transmit error counters

At this point, the CAN channel can start transmission and reception of CAN messages.

Within the CAN Channel Operation mode, the channel may be in four different sub-modes, depending on which type of communication functions are performed (see Figure 41.12):

- Channel idle: The CAN channel is neither receiving nor transmitting
- Channel receives: The channel is receiving a CAN message sent by another CAN node
- Channel transmits: The channel is transmitting a CAN message

  Note: The channel may receive its own message simultaneously when Self-test mode is enabled.
- Channel is in bus-off state: The CAN channel is cut off from CAN bus communication.

After setting the Channel Mode Control bit CFDC0CTR.CHMDC to Channel Operation mode, it is necessary to confirm that the Channel Reset Mode Status bit CFDC0STS.CRSTSTS and the Channel Halt Mode Status bit CFDC0STS.CHLTSTS in the Channel Status Register have been updated to indicate a successful transition to Channel Operation mode before the related CFDC0CTR.CHMDC bit can be changed again.

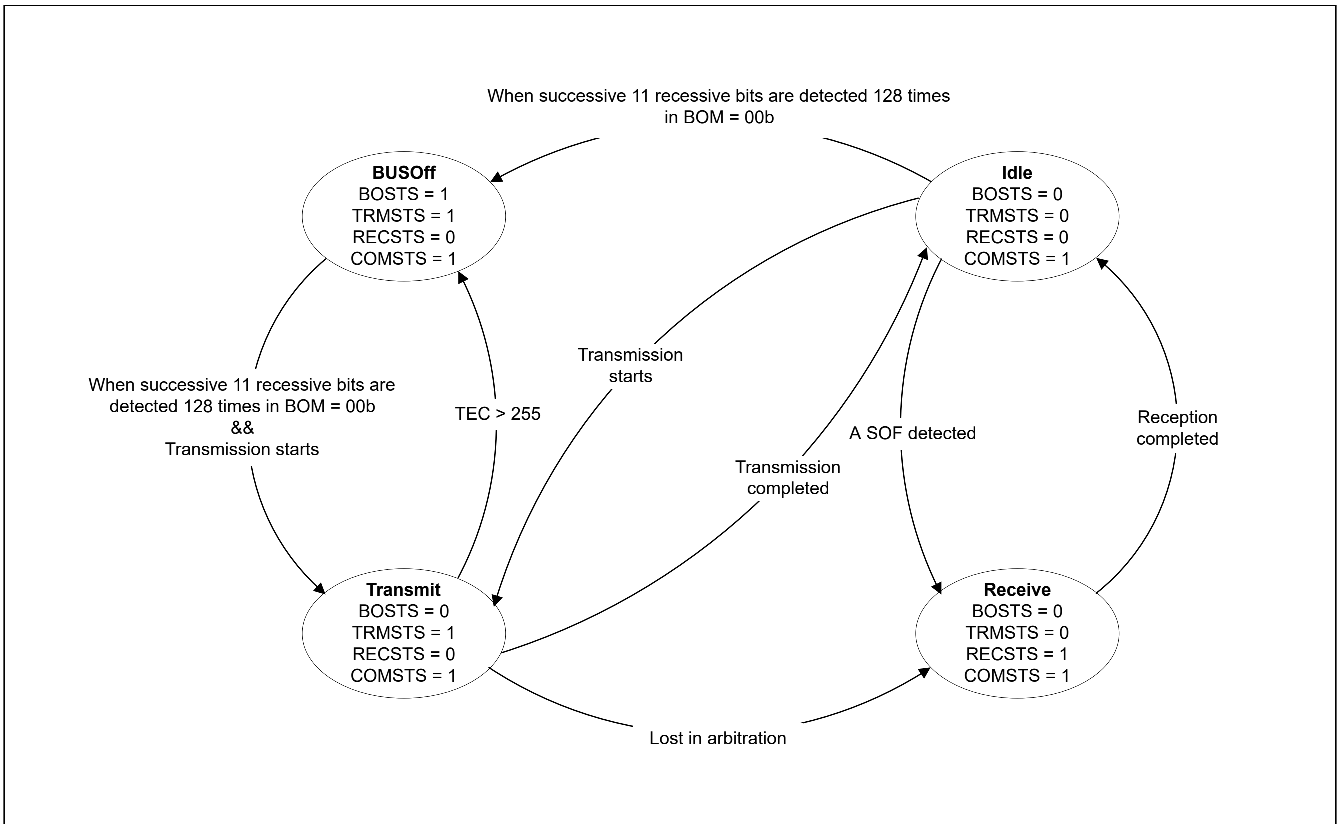


Figure 41.12 Sub-modes of CAN Channel Operation mode (only when BOM = 00b)

41.3.3.5 CAN Channel Bus-Off State

The CAN channel bus-off state is entered according to the fault confinement rules of the CAN specification. The following modes can be configured for returning to the CAN Channel Operation mode from the bus-off state:

- CFDC0CTR.BOM = 00b:
  Bus-Off recovery is compliant to ISO 11898-1, namely the CAN channel re-enters CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. TEC and REC counters are initialized to 0. The Bus-Off Recovery Flag CFDC0ERFL.BORF is set in this case.
- CFDC0CTR.BOM = 01b:

The CAN channel changes the value of the CFDC0CTR.CHMDC bits within the CAN Channel Control Register to 10b and switches immediately to Channel Halt mode automatically after entering bus-off state. TEC and REC counters are initialized to 0 and the Bus-Off Recovery Flag CFDC0ERFL.BORF is not set in this case.

- CFDC0CTR.BOM = 10b:
  The CAN channel changes the value of the CFDC0CTR.CHMDC bits within the CAN Channel Control Register to 10b as soon as it reaches bus-off state and enters Channel Halt mode automatically after the CAN channel has completed the bus-off recovery sequence (after 11 consecutive recessive bits are detected 128 times). TEC and REC counters are initialized to 0 and the Bus-Off Recovery Flag CFDC0ERFL.BORF is set in this case.
- CFDC0CTR.BOM = 11b:
  Bus-off recovery is initiated but CAN channel can immediately enter Channel Halt mode when still in bus-off state if a request is made to enter Channel Halt mode.
  TEC and REC counters are initialized to 0 and the Bus-Off Recovery Flag CFDC0ERFL.BORF is not set.
  Without setting CFDC0CTR.CHMDC [1:0] = 10b and when 11 recessive bits are detected 128 times continuously, transition conditions become the same as CFDC0CTR.BOM = 00b.

Note: If the recovery from bus-off occurs normally in this mode (after waiting for 128 sequences of 11 consecutive recessive bits), and no halt request has been generated during this period, then the Bus-Off Recovery flag CFDC0ERFL.BORF is set.

When software writes to the CFDC0CTR.CHMDC bit at the same time as the CAN channel enters Halt mode (at the start of bus-off when CFDC0CTR.BOM = 01b, or at the end of bus-off when CFDC0CTR.BOM = 10b), the software request has the highest priority.

Note: In the above case, the automatic setting of the CFDC0CTR.CHMDC bit to Channel Halt mode request is performed when the CFDC0CTR.CHMDC bit value is previously 00b (Channel Operation mode).

Additionally, it is possible to force the CAN channel to recover from the bus-off state by setting CFDC0CTR.RTBO to 1. The error state changes from bus-off state to integrating state with a maximum delay of 1 CAN bit time, and the CAN communication becomes possible again after 11 consecutive recessive bits are detected. The Bus-Off Recovery Flag is not set in this case, and the TEC and REC counters are initialized to 0.

Before setting CFDC0CTR.RTBO to 1, all pending transmissions from the TX message buffers, TX Queues and/or Common FIFO in TX mode should be disabled.

The disable of the pending transmission message buffer, TX Queue or FIFO must be confirmed by the corresponding acknowledge flags.

For the TX message buffer, the acknowledge flags are the Transmission Result Flags (CFDTMSTSj.TMTRF). For the TX Queue, it is the TX Queue Empty flag (CFDTXQSTS.TXQEMP). For the FIFO, it is the FIFO Empty flag (CFDCFSTS.CFEMP).

The CFDC0CTR.RTBO bit should be used for bus-off recovery only when CFDC0CTR.BOM is set to 00b.

Setting this bit in any state other than bus-off has no effect and the bit is cleared immediately.

Table 41.15 shows the settings for the Bus-Off Entry flag CFDC0ERFL.BOEF and the Bus-Off Recovery flag CFDC0ERFL.BORF for the different configurations of CFDC0CTR.BOM.

Table 41.15 Behavior of Bus-off Entry and Recovery flags

BOM	BOEF bit set	BORF bit set
00b	Always (on entry to bus-off)	Always (on exit from bus-off)
00b CFDC0CTR.RTBO set to 1	Always (on entry to bus-off)	Only if normal bus-off recovery occurs before software sets CFDC0CTR.RTBO to 1'
01b	Always (on entry to bus-off)	Never
10b	Always (on entry to bus-off)	Always (on exit from bus-off)
11b	Always (on entry to bus-off)	Only if normal bus-off recovery occurs before software issues a Halt request

For an efficient software procedure, it is not necessary to wait for the bus-off recovery sequence to end.

It is possible to perform a transmission re-initialization during bus-off recovery. To do this, follow the recommended software flow in Figure 41.13.

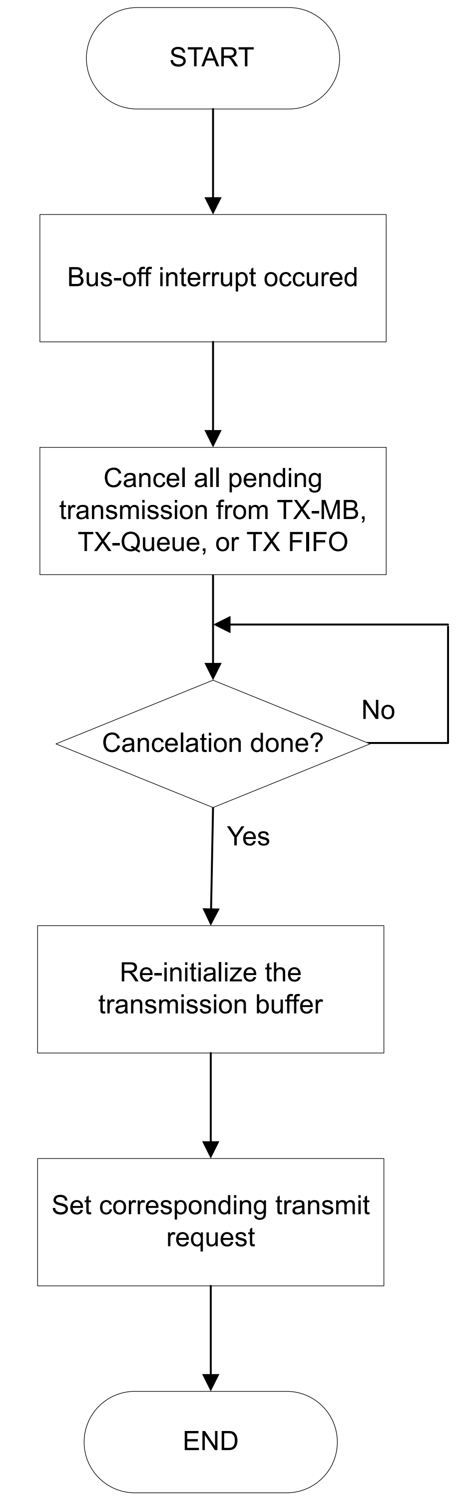


Figure 41.13 Transmission re-initialization during bus-off

41.3.4 Global Mode and Channel Mode Transition Interactions

The interaction between Global mode setting and Channel mode setting is as follows:

- Changing the Channel Mode Control bit CFDC0CTR.CHMDC in the Channel Control Registers does not affect the Global Mode Control bit CFDGCTR.GMDC.
- Changing the Global Mode Control bit CFDGCTR.GMDC affects the channel mode control as described in Table 41.16.

Table 41.16 Interaction between Global and Channel mode transition

Global mode change	Channel mode	Channel mode transition action
Sleep → Reset	Sleep	Channel remains in Sleep mode
Sleep → Halt	— (Global mode change not possible)	
Sleep → Operation	— (Global mode change not possible)	
Reset → Sleep	Sleep	Channel remains in Sleep mode
	Reset	Channel Sleep request bit is set automatically, channel enters Sleep Mode
Reset → Halt	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
Reset → Operation	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
Halt → Sleep	— (Global mode change not possible)	
Halt → Reset	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel mode control is set to Reset mode, channel enters Reset mode
Halt → Operation	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel remains in Halt mode
Operation → Sleep	— (Global mode change not possible)	
Operation → Reset	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel mode control is set to Reset mode, channel enters Reset mode
	Operation	Channel mode control is set to Reset mode, channel enters Reset mode
Operation → Halt	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel remains in Halt mode
	Operation	Channel mode control is set to Halt mode, channel enters Halt mode after communication finished

41.3.4.1 Timing of Global Mode Change

The transition time for the Global mode changes is shown in the following table.

Table 41.17 Maximum transition time for the global mode (1 of 2)

From	To	Maximum transition time
GL_SLEEP	GL_RESET	3 peripheral clock cycles*2
GL_RESET	GL_SLEEP	3 peripheral clock cycles
GL_RESET	GL_HALT	10 peripheral clock cycles
GL_RESET	GL_OPERATION	10 peripheral clock cycles

Table 41.17 Maximum transition time for the global mode (2 of 2)

From	To	Maximum transition time
GL_HALT	GL_RESET	2 CAN bit times
GL_HALT	GL_OPERATION	3 peripheral clock cycles
GL_OPERATION	GL_RESET	2 CAN bit times
GL_OPERATION	GL_HALT	3 CAN frames*1 *3

Note 1. The given transition time is the time without any errors on the bus. In case of an error condition, the transition time can lengthen to an uncalculated result. The transition time can also come to a stuck condition for locked RX lines or continued error conditions.
Note 2. Exit GL_SLEEP mode only when CFDGSTS.GRAMINIT is cleared.
Note 3. TQ, CAN frame and CAN bits are related to the individual channels. For the maximum transition time, the channel with the lowest baud rate must be used.

41.3.4.2 Timing of Channel Mode Change

The transition time for the Channel mode changes is shown in the following table.

Table 41.18 Maximum transition time for the channel mode

From	To	max. transition time
CH_SLEEP	CH_RESET	3 peripheral clock cycles
CH_RESET	CH_SLEEP	3 peripheral clock cycles
CH_RESET	CH_HALT	3 CAN bit times
CH_RESET	CH_OPERATION	4 CAN bit times
CH_HALT	CH_RESET	2 CAN bit times
CH_HALT	CH_OPERATION	4 CAN bit times*3
CH_OPERATION	CH_RESET	2 CAN bit times
CH_OPERATION	CH_HALT	2 CAN frames*1 *2

Note 1. The time specified for this transition does not include the case where channel enters bus-off state. For bus-off, the timing depends on the configuration of the CFDC0CTR.BOM[1:0] bits.
Note 2. The given transition time is the time without any errors on the bus. In case of an error condition, the transition time can lengthen to an uncalculated result. The transition time can also come to a stuck condition for locked RX lines or continued error conditions.
Note 3. In general, if the baud rate prescaler value CFDC0NCFG.NBRP is changed in CH_HALT mode, the transition time can deviate. The internal prescaler is a free running down counter that creates the TQ clock, and new BRP value is captured when the counter reaches the value 0.

41.4 Initialization

Before joining CAN communications, configure the following settings:

- Clock setting
- Bit timing setting (nominal and data rate)
- Baud Rate setting (nominal and data rate)
- CANFD setting
- Acceptance Filter setting (configuration of Global Acceptance Filter List)
- Reception, Transmission and GW-FIFO setting
- CAN Operation mode setting

41.4.1 Initialization of CAN Clock, Bit Timing and Baud Rate

41.4.1.1 Bit Timing Conditions

The following lines describe the composition of each segment and the restriction that apply to the segment setting.

1. Each segment setting
   SS = Fixed to 1 TQ

TSEG1 = See to (CFDC0NCFG) and (CFDC0DCFG)*1
TSEG2 = See to (CFDC0NCFG) and (CFDC0DCFG)*1
SJW = See to (CFDC0NCFG) and (CFDC0DCFG)*1
SS + TSEG1 + TSEG2 = 5 to 49 TQs for Data Bit Rate and 8 to 385 for Nominal Bit Rate

2. Restriction on TSEG1, TSEG2 and SJW
TSEG1(N) > TSEG2(N) ≥ SJW(N)
TSEG1(D) ≥ TSEG2(D) ≥ SJW(D)*1
When only classical frames are used, configure the bit fields TSEG1 and TSEG2 of CFDC0DCFG to valid values.

Note 1. This feature is not available in the classical CAN function.

Table 41.19 shows an example of how to set the bit timing to achieve the required Sample Point settings.

Table 41.19 Bit timing examples

1 bit	Set value (TQ)				Sample point*1(%)
	SS	TSEG1	TSEG2	SJW	
5TQ	1	2	2	1	60.00
8TQ	1	4	3	1	62.50
	1	5	2	1	75.00
10TQ	1	6	3	1	70.00
	1	7	2	1	80.00
12TQ	1	8	3	1	75.00
	1	9	2	1	83.33
15TQ	1	10	4	1	73.33
	1	11	3	1	80.00
16TQ	1	10	5	1	68.75
	1	11	4	1	75.00
20TQ	1	12	7	1	65.00
	1	13	6	1	70.00
24TQ	1	15	8	1	66.66
	1	16	7	1	70.83
50TQ	1	39	10	4	80.00

Note 1. Sample point (in case of 75%)

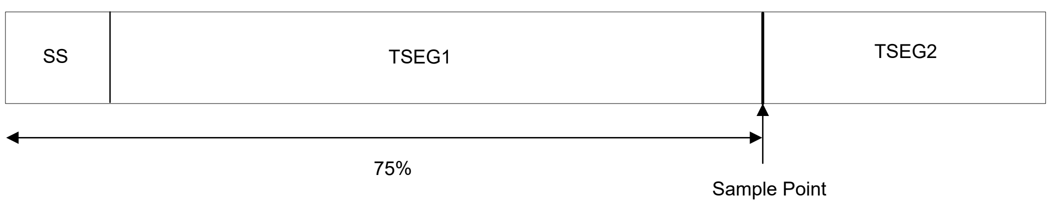


Figure 41.14 Sample point (in case of 75%)

41.4.1.2 CAN Bit Timing

In the CAN protocol, each bit in a communication frame is composed of three segments that can be configured individually for channel using the related CFDC0NCFG and CFDC0DCFG*1 registers.

Note 1. This register is not available in the classical CAN function.

Figure 41.15 shows the segment composition of a bit and the sample point in it.

Of these segments, the Time Segment 1 (TSEG1) and Time Segment 2 (TSEG2) are used to specify the position of the sample point, so that the timing at which each bit on the CAN bus is sampled can be altered by changing the values of these segments.

The minimum resolution for this timing is referred to as Time Quantum (TQ), which is determined by the clock frequency supplied to the CAN channel and the divide-by-N value of the baud rate prescaler (nominal and data rate).

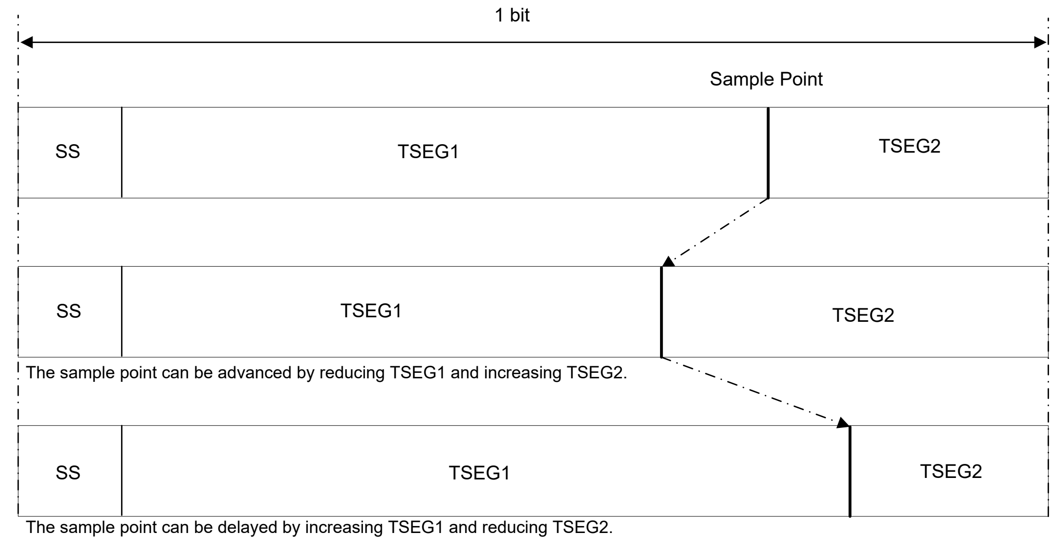


Figure 41.15 Segment composition of a bit and the sample point

1. SS: Synchronization Segment
   This segment is used to synchronize bits by monitoring a recessive-to-dominant edge during the interframe space. This consists of intermission, suspend transmission, bus idle, during bus idle, and all nodes that can start transmission.
2. TSEG1: Time Segment 1
   This segment absorbs physical delays on the CAN network. A physical delay on the network is two times the total sum of a bus delay, input comparator delay, and output driver delay. It can be lengthened by SJW.
3. TSEG2: Time Segment 2
   This segment is used to correct a phase error by performing resynchronization. It can be shortened by SJW. While sending or receiving a message, communication frames between some nodes may get out of sync due to a drift in the oscillator frequency or a delay in the transmission path. This is referred to as a phase error.
4. SJW: Resynchronization Jump Width
   This is the maximum width by which bits that have become out of sync due to a phase error may be corrected.

Figure 41.15 shows only one symbolic sample point.

41.4.1.3 Baud Rate

Either the CANFD core clock (CANFDCLK) or the external oscillator clock (CANMCLK) can be selected globally as CAN communication clock.

The transfer speed is determined by the DLL clock, the divide-by-N value of the baud rate prescaler, and the number of TQs in one bit.

$$\text{baud rate} = \frac{\text{DLL\_Clock}}{(\text{number\_of\_time\_quanta\_per\_bit}) \times (\text{BRP} + 1)}$$

Figure 41.16 shows a block diagram of the circuit that generates the CAN channel communication clock and Table 41.20 shows a baud rate examples.

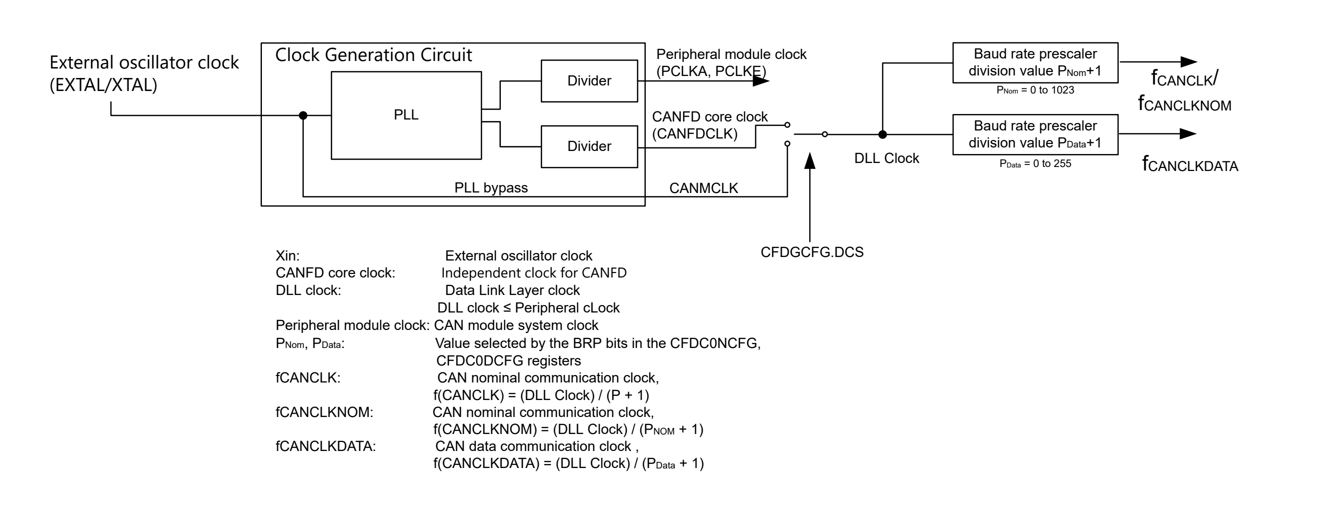


Figure 41.16 Block diagram of the circuit that generates the CAN channel communication clock

Table 41.20 Nominal baud rate calculation formula and example CAN communication configurations

Baud rate calculation formula	(DLL clock) (baud rate prescaler divide-by-N value*1) × (number of TQs in one bit)						
	40 MHz	32 MHz	30 MHz	24 MHz	20 MHz	16 MHz	10 MHz	8 MHz*2
1 Mbps	8TQ (5) 20TQ (2)	8TQ (4) 16TQ (2)	10TQ (3) 15TQ (2)	8TQ (3) 12TQ (2) 24TQ (1)	10TQ (2) 20TQ (1)	8TQ (2) 16TQ (1)	10TQ (1)	8TQ (1)
500 Kbps	8TQ (10) 20TQ (4)	8TQ (8) 16TQ (4)	10TQ (6) 15TQ (4) 20TQ (3)	8TQ (6) 12TQ (4) 24TQ (2)	10TQ (4) 20TQ (2)	8TQ (4) 16TQ (2)	10TQ (2) 20TQ (1)	8TQ (2) 16TQ (1)
250 Kbps	8TQ (20) 20TQ (8)	8TQ (16) 16TQ (8)	10TQ (12) 15TQ (8) 20TQ (6)	8TQ (12) 12TQ (8) 24TQ (4)	10TQ (8) 20TQ (4)	8TQ (8) 16TQ (4)	10TQ (4) 20TQ (2)	8TQ (4) 16TQ (2)
125 Kbps	8TQ (40) 20TQ (16)	8TQ (32) 16TQ (16)	10TQ (24) 15TQ (16) 20TQ (12)	8TQ (24) 12TQ (16) 24TQ (8)	10TQ (16) 20TQ (8)	8TQ (16) 16TQ (8)	10TQ (8) 20TQ (4)	8TQ (8) 16TQ (4)
83.3 Kbps	8TQ (60) 12TQ (40) 16TQ (30) 24TQ (20)	8TQ (48) 12TQ (32) 16TQ (24) 24TQ (16)	8TQ (45) 10TQ (36) 12TQ (30) 15TQ (24) 20TQ (18) 24TQ (15)	8TQ (36) 12TQ (24) 16TQ (18) 24TQ (12)	8TQ (30) 10TQ (24) 12TQ (20) 15TQ (16) 16TQ (15) 20TQ (12) 24TQ (10)	8TQ (24) 12TQ (16) 16TQ (12) 24TQ (8)	8TQ (15) 10TQ (12) 12TQ (10) 15TQ (8) 20TQ (6) 24TQ (5)	8TQ (12)
33.3 Kbps	8TQ (150) 12TQ (100) 16TQ (75) 20TQ (60) 24TQ (50)	8TQ (120) 10TQ (96) 12TQ (80) 15TQ (64) 16TQ (60) 20TQ (48) 24TQ (40)	10TQ (90) 12TQ (75) 15TQ (60) 20TQ (45)	8TQ (90) 10TQ (72) 12TQ (60) 15TQ (48) 16TQ (45) 20TQ (36) 24TQ (30)	8TQ (75) 10TQ (60) 12TQ (50) 15TQ (40) 20TQ (30) 24TQ (25)	8TQ (60) 10TQ (48) 12TQ (40) 15TQ (32) 16TQ (30) 20TQ (24) 24TQ (20)	10TQ (30) 12TQ (25) 15TQ (20) 20TQ (15)	8TQ (30)

Note: Shown in ( ) are the baud rate prescaler divide-by-N value.
Note 1. Baud rate prescaler divide-by-N value = P + 1 (P = 0 - 1023) P: value selected by the BRP bits in the Channel Configuration Registers.
Note 2. Minimum frequency to achieve maximum nominal baud rate of 1 Mbps.

Table 41.21 Baud rate calculation example for nominal and data bit rate CAN communication configurations

Baud rate calculation formula	(DLL clock) (baud rate prescaler divide-by-N value*1) × (number of TQs in one bit) 40 MHz	20 MHz
Nominal 1 Mbps Data 5 Mbps	40TQ (1)	20TQ (1)
	8TQ (1)	Not possible
Nominal 500 Kbps Data 2 Mbps	80TQ (1)	40TQ (1)
	20TQ (1)	10TQ (1)

Note: Shown in ( ) are the baud rate prescaler divide-by-N values and this table is not available in the classical CAN function.
Note 1. Baud rate prescaler divide-by-N value = P + 1 (P = 0 - 1023) P: value selected by the BRP bits in the Channel Configuration Registers.

For optimum clock tolerance in networks using the FD frame format, the length of the time quantum should be the same in nominal bit time and in data bit time. This means CFDC0NCFG.NBRP = CFDC0DCFG.DBRP.

Additionally, if transceiver delay compensation is used, do not program the CFDC0DCFG.DBRP bit to be greater than 1, as 1 means divide by 2.

41.4.1.4 Setting of CAN Clock, Bit Timing and Baud Rate

Figure 41.17 shows the procedure for setting the CAN clock and the baud rate for a channel.

These settings should be performed during Channel Reset mode (Configuration mode) for the CAN channels.

Before going to channel communication state, the baud rate must be configured, otherwise the mode does not switch correctly.

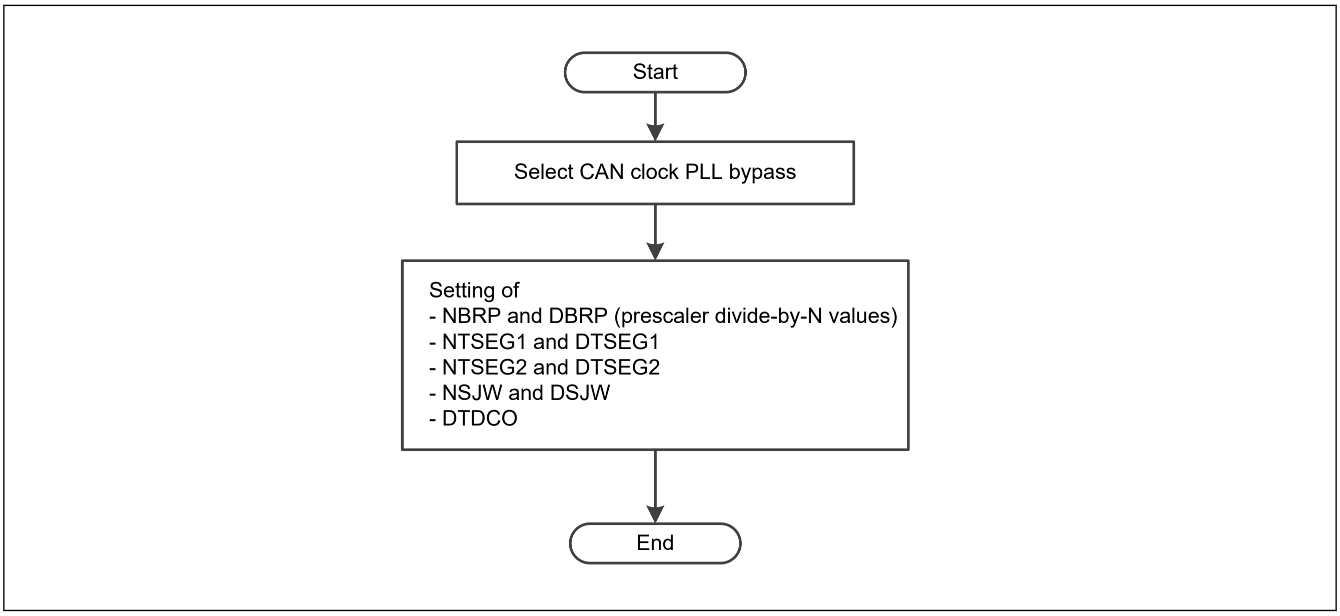


Figure 41.17 Procedure for setting the CAN bit timing and baud rate

41.4.1.5 Transmitter Delay Compensation

This chapter is not valid for classical CAN.

When a high baud rate is used such as 5 to 8 Mbps for the data phase, the transmitter delay can become greater than TSEG1. In this case, the transmitter always detects a bit-error in the data phase of the CANFD frame. The TDC compensates for the inability of the transmitter to receive its own transmitted bit at the sample point of that bit.

There is another symbolic sample point known as the Secondary Sample Point (SSP) that is used only during the data phase of CANFD frames. This is derived from the Transceiver Delay Compensation Result bit (CFDC0FDSTS.TDCR) as shown in Figure 41.18.

The resolution of the configuration, measured and offset values is based on the CAN channel DLL clock.

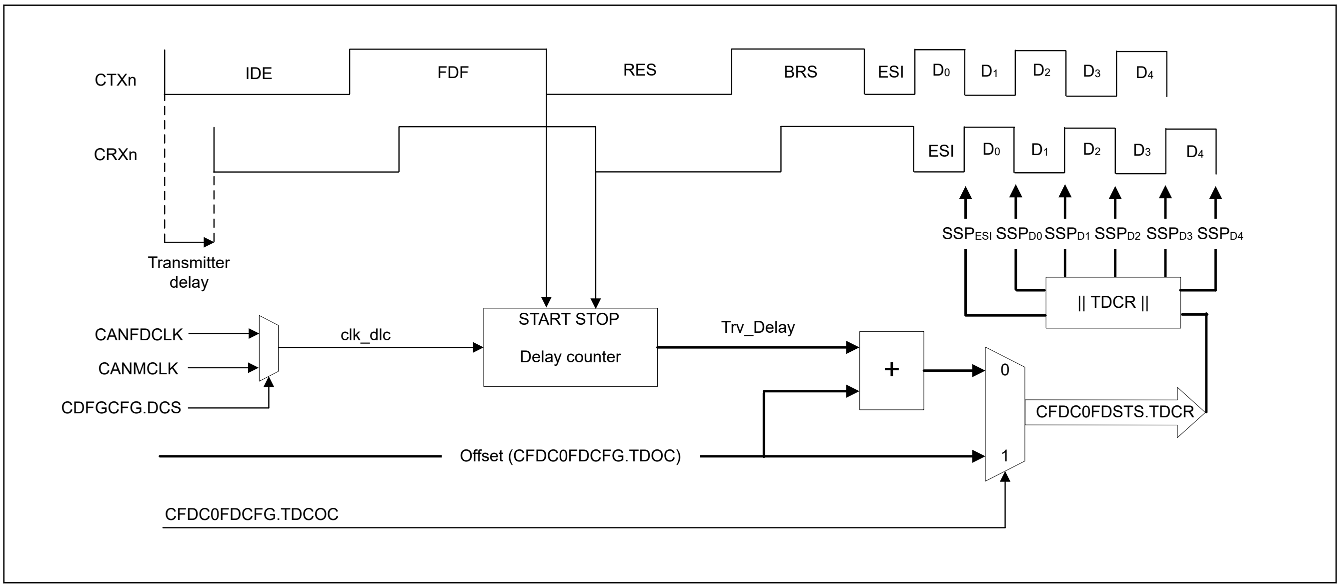


Figure 41.18 Transmitter delay compensation

The measured Trv_Delay is based on the number of clk_dlc clock cycles. The delay is counted up by one for each started clock until the dominant value is seen on CRXn. Figure 41.19 shows the measured result. Trv_Delay counted to maximum 127 with a clk_dlc clock.

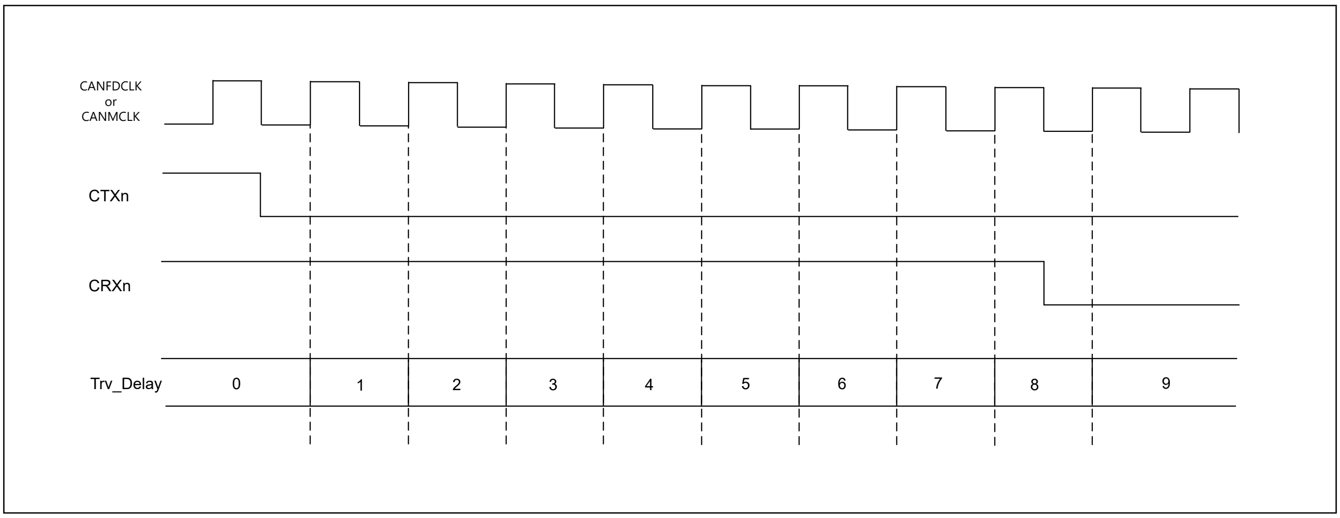


Figure 41.19 Trv_Delay measurement example

The SSP is calculated by taking the result from CFDC0FDSTS.TDCR and rounding the value down to the nearest integer number of data time quanta.

Figure 41.20 shows the positioning of the secondary sample point. When CFDC0FDCFG.TDCOC is equal to 0, the SSP is equal to the Trv_Delay (measured delay) + CFDC0FDCFG.TDCO, rounded down to the nearest integer number of time quanta. Usually, the TDCO value should have the size of (SyncSegment data + TSEG1 data) to position the SSP to a theoretical location of the sample point.

If the CFDC0FDCFG.TDCOC is equal to 1, the SSP is defined by CFDC0FDCFG.TDCO. If CFDC0DCFG.DBRP is greater than 0, the value is also rounded down to the nearest integer number of time quanta.

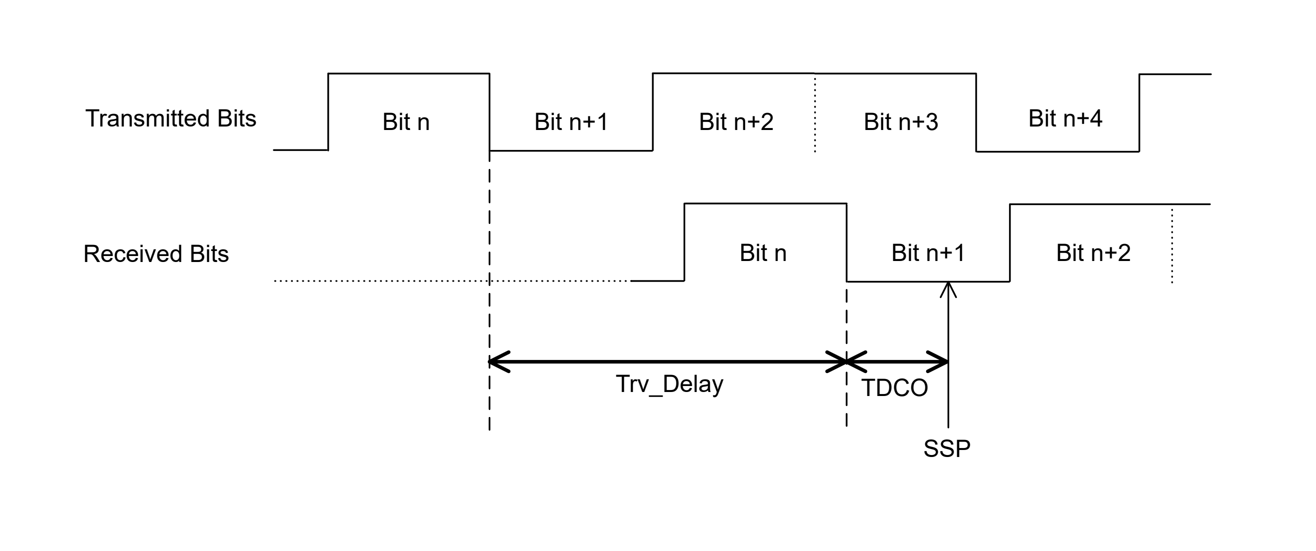


Figure 41.20 Position of the secondary sample point

The maximum delay (Trv_Delay + TDCO) which can be compensated by the CANFD module is (6 data bits - 2clk_dlc).

The ISO 11898-1 allows you to set different values for BRP_data and BRP_nom.

If different values are used for CFDC0NCFG.NBRP and CFDC0DCFG.DBRP, then two CAN nodes may be out of synchronization at the point when the bit rate changes from nominal bit rate to data bit rate after sample point of the BRS bit. This condition is shown in Figure 41.21.

The length of the time quantum should be the same in the nominal bit time and in the data bit time. This means CFDC0NCFG.NBRP = CFDC0DCFG.DBRP.

Different bit rates can be achieved by selecting different configuration values for the Time Segments. The nominal bit rate can be configured from 8 to 385 TQs and the data bit rate from 5 to 49 TQs.

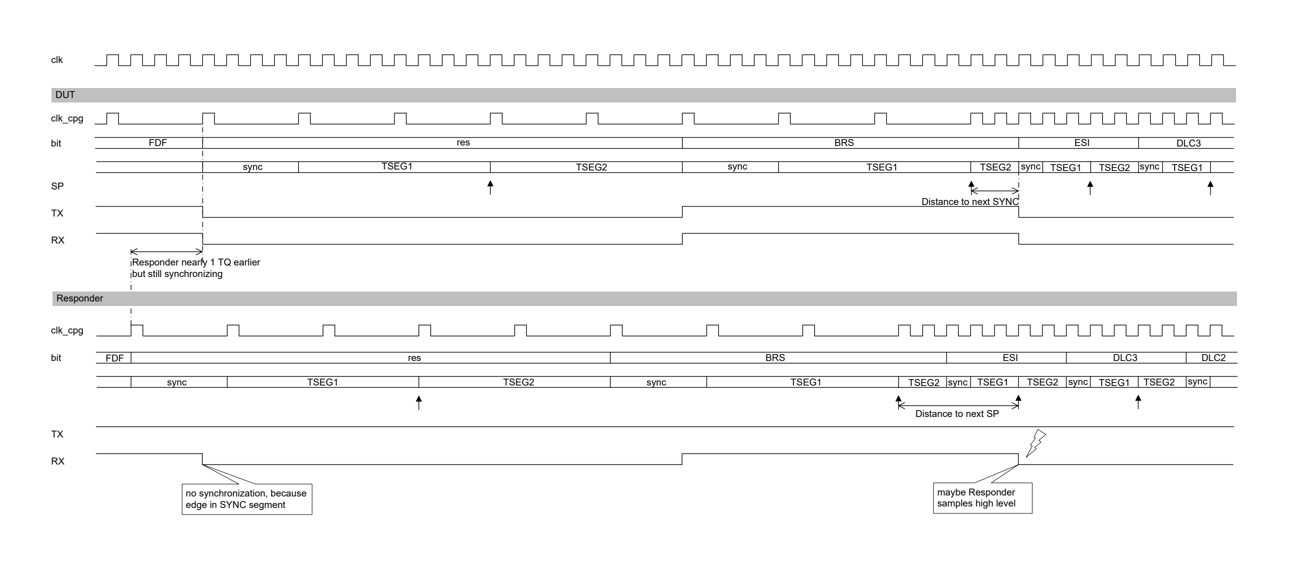


Figure 41.21 Loss of synchronization between two CAN nodes

The transmitter delay compensation measurement result is updated at the falling edge from FDF bit to RES bit when configured accordingly (CFDC0FDCFG.TDCE = 1, CFDC0FDCFG.TDCOC = 0).

Figure 41.22 shows the read flow to get the measured transmitter delay compensation result.

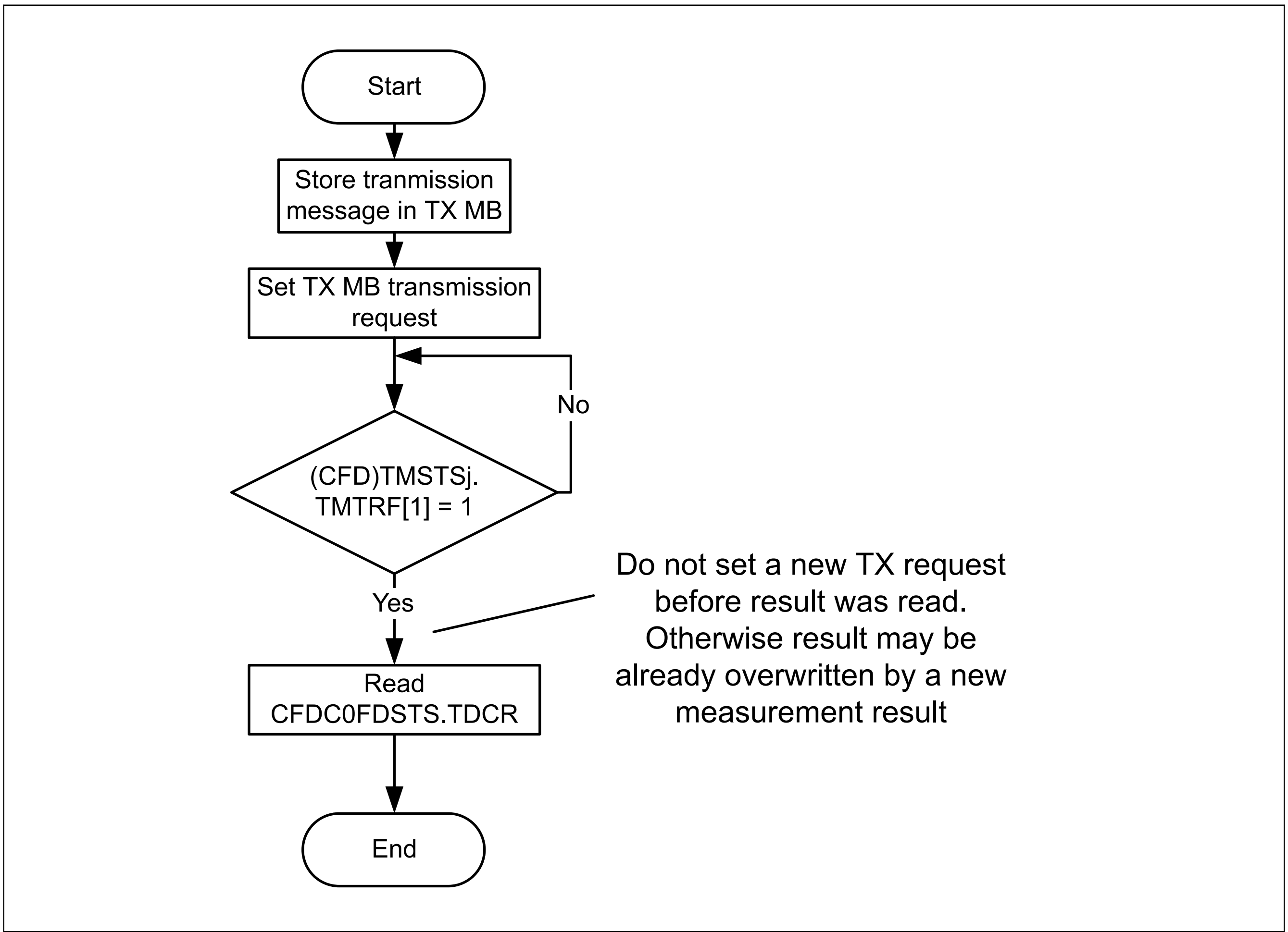


Figure 41.22 TDC result read flow

41.4.2 CAN Module Configuration after Hardware Reset

After a hardware reset (power on reset) or after setting and clearing a CFDGRSTC.SRST bit, the CANFD module enters Global Sleep mode automatically.

To enable configuration of the CANFD module, you must exit Sleep mode by clearing the Global Sleep Request bit CFDGCTR.GSLPR to 0.

After a hardware reset, the module starts RAM initialization, the CFDGSTS.GRAMINIT bit in the Global Status Register is set automatically to indicate that the CANFD logic is initializing the RAM.

After RAM initialization is complete, this bit is cleared automatically.

RAM initialization is necessary to avoid setting of false ECC error flag after hardware reset the random data presented in the RAM.

Do not access registers of CANFD in either read or write until RAM initialization is complete and the CFDGSTS.GRAMINIT bit is cleared.

Before going to communication mode, the Global Acceptance Filter List and message FIFO buffers must be configured. In addition, CAN channel must be configured such as CAN bit timing. For this configuration, CAN channel must be released from Channel Sleep mode and must be configured for communication in Channel Reset mode (Configuration mode).

Figure 41.23 shows the configuration procedure. For details about each step, see section 41.5. Acceptance Filtering Function using Global Acceptance Filter List (AFL), section 41.6. FIFO Buffers and Normal Message Buffer Configuration, section 41.7. Interrupts and DMA and section 41.4.1.3. Baud Rate.

The CANFD module does not perform the RAM initialization sequence after executing a software reset by setting CFDGRSTC.SRST.

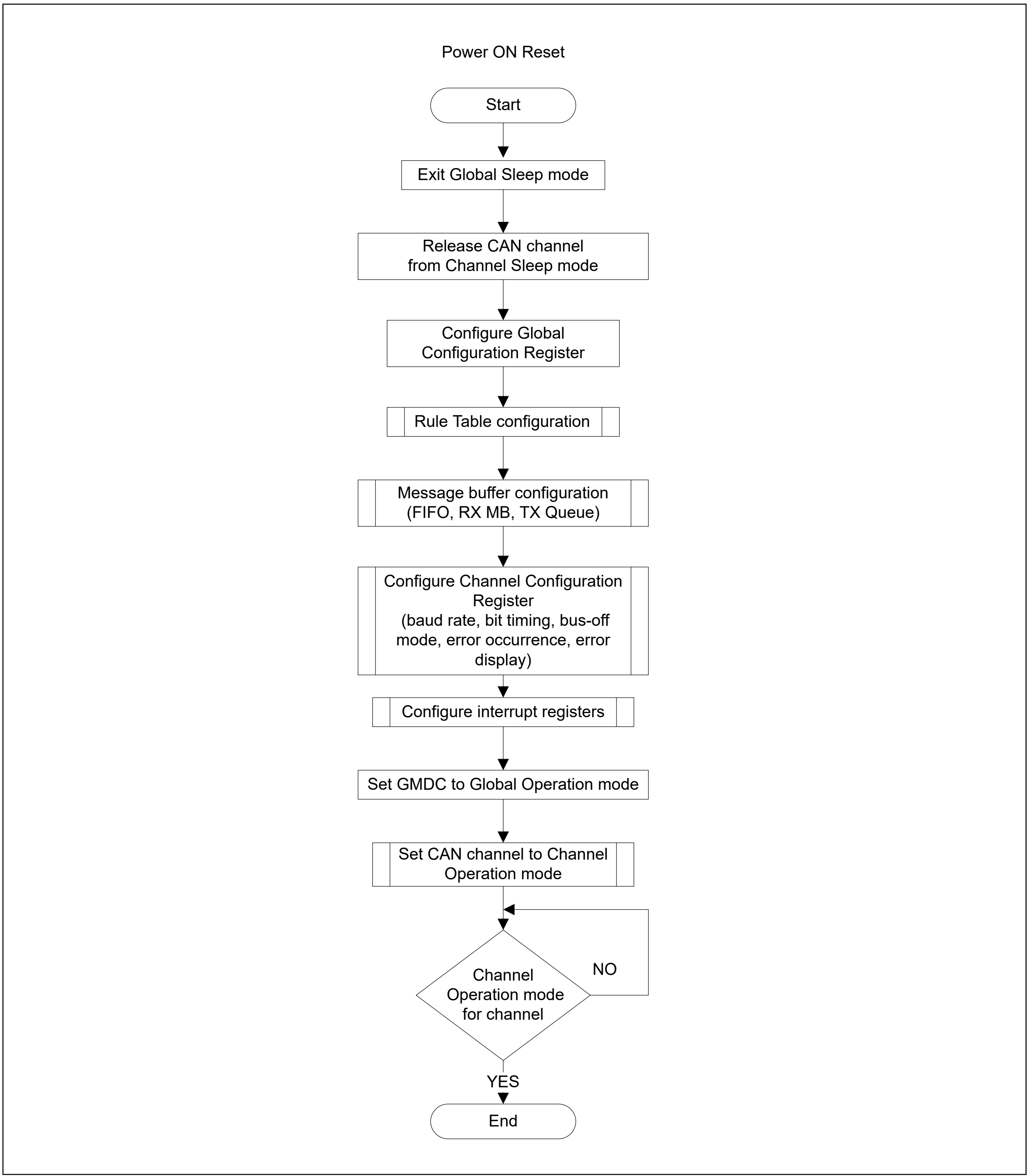


Figure 41.23 Configuration procedure after a hardware reset

41.5 Acceptance Filtering Function using Global Acceptance Filter List (AFL)

41.5.1 Overview

The CANFD module can handle message acceptance filtering with a global Acceptance Filter List (called AFL). Each element of the AFL defines a filter rule for messages received on a specific channel.

The following actions are performed based on the AFL entries:

- Acceptance filtering based on received CAN Identifier and masking
- DLC filtering based on received DLC value
- Message data payload according to the CFDGCFG.CMPOC bit*1
- Storage of accepted messages in the message buffer objects defined in the related AFL entry
- Attaching a 16-bit pointer to the stored messages defined in the related AFL entry, for example to support AUTOSAR applications
- Attaching a 2-bit information label to the stored messages defined in the related AFL entry

Note 1. This feature is not available in the classical CAN function.

The CANFD module allows a maximum of 16 AFL entries.

During acceptance filtering process, each AFL entry in a channel is checked against the received message by the acceptance filter unit. The check starts from the lowest AFL entry number for this channel.

AFL search stops when a match of the received identifier with a configured identifier/mask combination occurs or when the received identifier has been compared against all AFL entries defined for the related channel. If no match occurs, then the received message is rejected. No notification is given to the application in this case.

Additionally, an automatic DLC filtering is performed for each accepted message if DLC check is globally enabled. If the DLC value of the received message is equal to or higher than the configured DLC value in the matching AFL entry, the DLC check is passed.

If DLC replacement (CFDGCFG.DRE bit) is enabled, DLC value configured in the matching AFL entry is greater than 0x0 and DLC check passes, then the configured value of DLC in the matching AFL entry is stored in the destination RXMB or FIFO Buffer.

If the received value of DLC is greater than the configured DLC value in the matching AFL entry, then the additional data bytes received on the CAN Bus are not stored in the destination RXMB or FIFO Buffer. These additional data bytes are stored as 0x00 in the destination RXMB or FIFO Buffer.

If DLC replacement is enabled and DLC value of matching AFL entry is 0x0, then the received value of DLC is stored in the destination RX MB or FIFO Buffer.

If DLC replacement (the CFDGCFG.DRE bit) is disabled and DLC check passes, then the received value of DLC on the CAN bus is stored in the destination RXMB or FIFO buffer.

If the received value of DLC is greater than the configured DLC value in the matching AFL entry, then the additional data bytes received from the CAN bus are also stored in the destination RXMB or FIFO buffer.

If DLC value of the received message is less than the configured DLC value in the matching AFL entry, then DLC check fails. In this case, the received message is rejected and is not stored in any RXMB or FIFO buffer.

Additionally, DLC check failure is flagged by the DLC Error Flag in the Global Error Flag Register. If configured, an error interrupt is also generated. The DLC replacement configuration has no impact if the DLC check fails.

If a message has passed both acceptance filtering and DLC filtering, it is stored in a single reception message buffer and/or in FIFO buffers configured for reception function.

This message storage target information is also defined in the same AFL entry. Do not set a target at the AFL entry which is not configured.

Each accepted received message can be stored into a maximum of 2 different target destinations (single reception message buffer and/or FIFO buffers).

The programming of more than 2 target destinations is not allowed. If more destinations are programmed, then the internal timing might lead to a race condition that prevents the storage of received messages in the message RAM.. Correct configuration of the numbers of target destination is the responsibility of the application.

Additional protection mechanism is made for the case when a received message contains more data payload Bytes than possible to store in the target destination (CFDRMNB.RMPLS, CFDRFCCa.RFPLS or CFDCFCC.CFPLS).

If CFDGCFG.CMPOC = 0, the message is completely rejected and is stored in the target destination. When CFDGCFG.CMPOC = 0 and RX or Common FIFO full including the received message contains more data payload bytes than possible to store in the target destination (CFDRMNB.RMPLS, CFDRFCCa.RFPLS or CFDCFCC.CFPLS), the corresponding CFDFMSTS.RFxMLT or CFDFMSTS.CFxMLT bit is not set to 1, respectively.

When CFDGCFG.CMPOC = 1, the received data bytes greater than CFDRMNB.RMPLS is rejected. When CFDGCFG.CMPOC = 1 and RX or Common FIFO full including the received message contains more data payload bytes than possible to store in the target destination (CFDRMNB.RMPLS, CFDRFCCa.RFPLS or CFDCFCC.CFPLS), the corresponding CFDFMSTS.RFxMLT or CFDFMSTS.CFxMLT bit is set to 1, respectively.

Depending on the CFDGCFG.DRE bit, the original received DLC or the DLC value configured at the AFL entry is stored.

Regardless of the CFDGCFG.CMPOC bit setting, CFDGERFL.CMPOF is set to 1 if a payload overflow condition is detected.

The DLC filtering is performed before the payload overflow function. So for one reception frame, only one flag can be set at the same time with CFDGERFL.DEF or CFDGERFL.CMPOF*1.

Note 1. This bit is not available in the classical CAN function.

41.5.2 Allocation of AFL Entries

The number of AFL entries per channel can be configured using the dedicated field in the related Global Acceptance Filter Configuration Registers (see Figure 41.24).

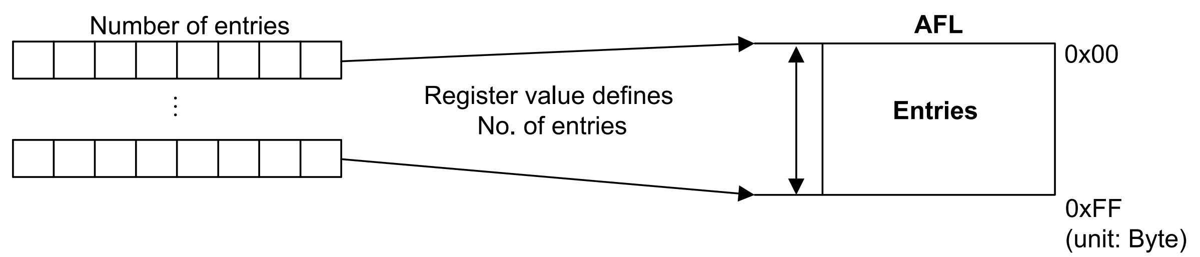


Figure 41.24 Configuration of AFL for each channel

The minimum number of entries for one channel is 0 (no entries defined for the channel) and the maximum number of entries is 16.

All entries are unique for a channel and overlapping or sharing of entries is not supported. Correct configuration of the AFL is the responsibility of the application.

The CANFD module does not flag errors related to the configuration of the AFL.

41.5.3 AFL Entry Description

Each AFL entry consists of 16 bytes. The fields in all entries are identical.

Each entry contains the following information for acceptance filtering and DLC filtering:

- Identifier (11 bits for Standard Frame format, 29 bits for Extended Frame format):
  Acceptance filter unit checks the identifier field of the received message against the identifier field of each AFL entry (full 29 bits masking of identifier bits is possible, see information that follows).
- IDE bit:
  Acceptance filter unit checks the IDE bit of the received message against this bit and selects the relevant part of the identifier field for acceptance filtering (masking of IDE bit is possible, see the information that follows).
- RTR bit:
  Acceptance filter unit only accepts data frames (RTR = 0) or remote frames (RTR = 1) according to the setting of this bit (masking of RTR bit is possible, see the information that follows).
- Loopback Configuration bit:
  This bit can enable or disable the AFL entry depending on the Loopback Configuration or Mirror mode condition.
- Mask for Identifier bits (29 bits):

Each bit in the identifier mask can mask the corresponding identifier bit in the AFL entry during acceptance filtering, see Figure 41.25.

- Mask for IDE bit:
  If this Mask bit masks the IDE bit of the AFL entry in both Standard Identifier and Extended Identifier format, messages can be accepted by this AFL entry. The identifier of the received message is compared against the Standard Identifier part of the AFL entry for Standard Identifier format messages and against the Extended Identifier part of the AFL entry for Extended Identifier format messages.
- Mask for RTR bit:
  If this Mask bit masks the RTR bit of the AFL entry in both frame formats, data frame and remote frame formats are accepted by this AFL entry.
- Pointer information (16 bits):
  This 16-bit pointer is attached to a received message accepted by the related AFL entry. The pointer is added during message storage in the message buffer area and can be used by application as support function. The pointer information can be used for example to support PDU identifier allocation for the received message in AUTOSAR systems.
- Information label (2 bits):
  This 2-bit label is attached to a received message accepted by the related AFL entry. The label is added during message storage in the message buffer area and can be used by application as support function.
- DLC value for automatic DLC filtering:
  If the DLC value of the received message is equal or higher than the configured DLC value, the DLC check is passed.

If the DLC value in this AFL entry is configured to 0, DLC filtering is effectively disabled for this entry (all accepted messages pass DLC filtering).

Each AFL entry contains the following information for the handling of received messages:

- Message buffer number of one single reception message buffer as target for received message storage
- Single reception message buffer enable bit to configure the single reception message buffer number to be valid or invalid, as target for received message storage
- FIFO direction pointer - each bit of the FIFO direction pointer configures a dedicated FIFO as possible target for a received message

There is no hardware protection against such storage of message. Therefore, the FIFO direction pointer must be configured carefully.

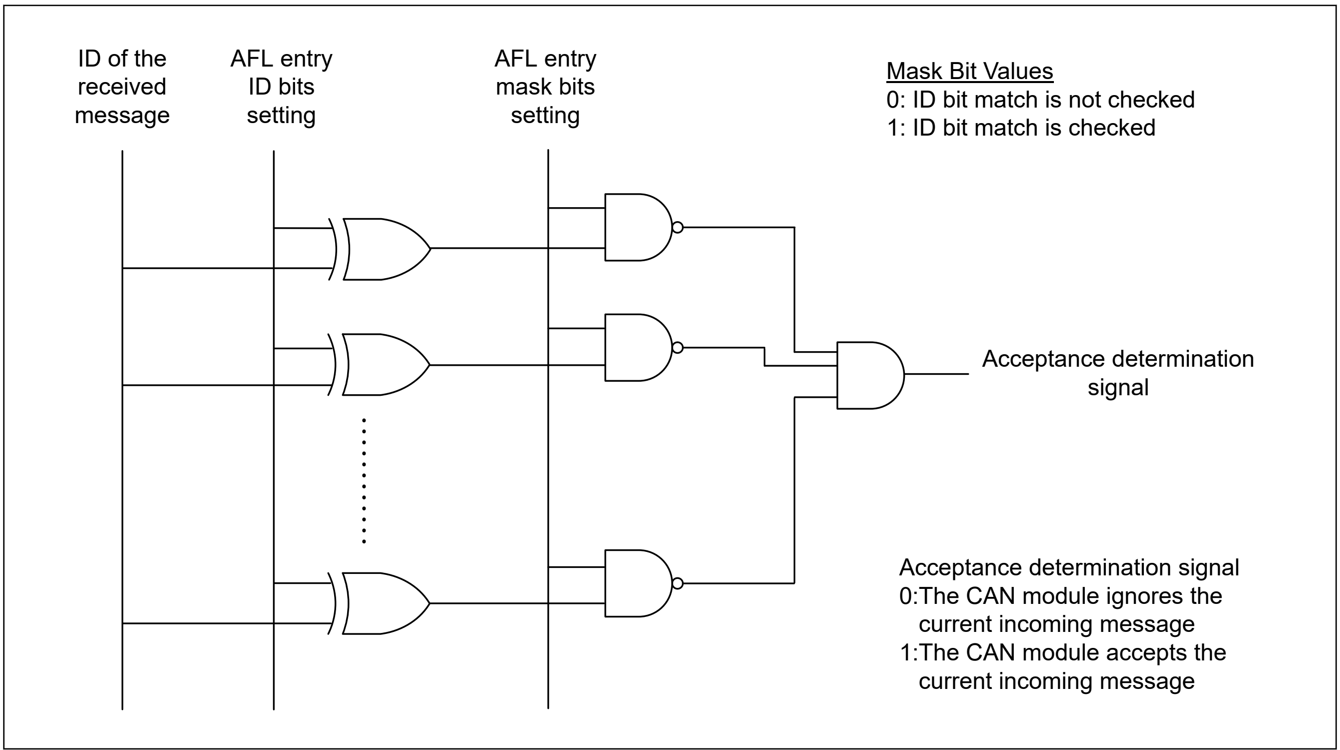


Figure 41.25 Acceptance function

41.5.4 Entering Entries in the AFL

Application software can enter one full entry into the AFL using the following registers:

- Global AFL ID Entry Register: Part 1 of the AFL entry
- Global AFL Mask Entry Register: Part 2 of the AFL entry
- Global AFL Pointer 0 Entry Register: Part 3 of the AFL entry
- Global AFL Pointer 1 Entry Register: Part 4 of the AFL entry.

16 sets of these registers form a group of AFL entries. The AFL should only be configured in CH_RESET or CH_HALT mode.

Follow the configuration shown in Figure 41.26 to program the AFL.

After entering all entries in Configuration mode, locking of the AFL access should be performed to protect unwanted write access to the AFL.

Write protection is active during all Global modes (GL_RESET, GL_HALT, and GL_OPERATION) if the lock bit is set.

Read access to AFL is still possible during all Global modes even when AFL data access is disabled (consistency check of AFL contents is possible during run time).

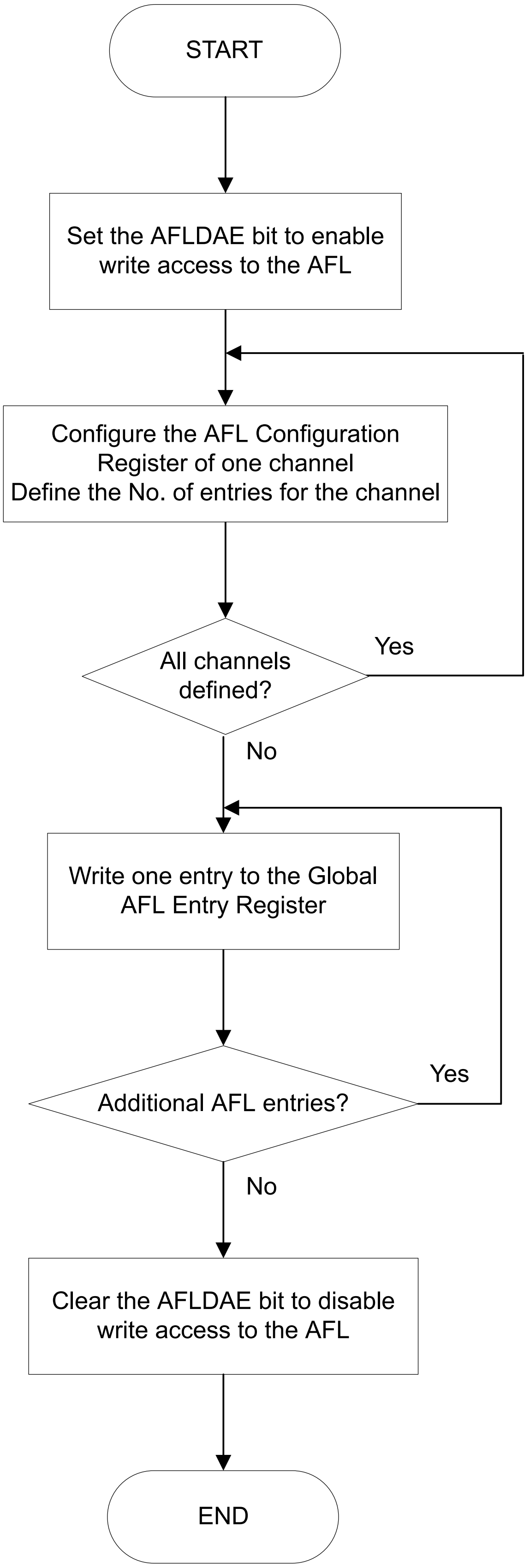


Figure 41.26 AFL configuration flow

41.5.5 Loopback Modes

If the Loopback Configuration bit is set, the AFL entry is only valid in Loopback test mode (Self-test mode 0 or Self-test mode 1) or in mirror mode when receiving messages that were transmitted by the respective CAN channel itself.

The AFL entry is not valid for received messages in loopback mode transmitted by other CAN nodes on the bus. The expression valid or invalid for the related entry means that this AFL entry is or is not compared against the received message ID respectively.

If the Loopback Configuration bit is 0, the AFL entry is only valid for:

- Received messages transmitted by other CAN nodes on the bus in normal (non-loopback mode) and mirror modes
- Received messages transmitted by other CAN nodes or the CAN channel itself in Loopback test mode.

The mirror mode can be enabled with the CFDGCFG.MME bit in the Global Configuration Register. If CFDGCFG.MME bit is set, then a successfully transmitted message can be stored back in an RX message buffer or FIFO buffer if a matching entry is configured in the AFL for that channel.

The Loopback Configuration bit in the matching AFL entry must be set to store this frame.

If Mirror mode and Loopback test mode are configured at the same time, the Loopback test mode behavior applies.

Table 41.22 shows the behavior of the acceptance filter unit depending on the setting of the related input signals.

Table 41.22 Behavior of acceptance filter based on the loopback configuration setting in AFL entry

Mirror Mode Enable (MME Configuration bit)	Loopback in test mode (Self-test mode 0 or Self-test mode 1)	Channel mode	Loopback Configuration bit in AFL entry	AFL entry
0	0	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Invalid
			1	Invalid
	1	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Valid
			1	Valid
1	0	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Invalid
			1	Valid
	1	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Valid
			1	Valid

Note: The expression valid or invalid for the related entry means that this AFL entry is or is not compared against the received message ID, respectively.

41.5.6 IDE Masking

When the GAFLIDEM bit is 0 in an AFL entry, the IDE bit configured in the AFL entry is not used for ID matching. In this case, the use of ID[10:0] or ID[28:0] matching is based on the received IDE bit.

Consider the following example:

- The ID and Mask fields of an AFL entry x is configured as follows:
  - CFDGAFLID [x] = 0xC0553A20 → IDE = 1, RTR = 1, LLB = 0, ID[10:0] = 0x220 / ID[28:0] = 0x00553A20
  - CFDGAFLMr = 0x0000FFFF → IDEM = 0. RTRM = 0, IDM[10:0] = 0x7FF / IDM[28:0] = 0x0000FFFF

- The comparison result for the four different received IDs with AFL entry x is described as follows:
  - If a frame with IDE = 0 and ID = 0x220 is received, this is considered as a match
  - If a frame with IDE = 0 and ID = 0x320 is received, this is not a match
  - If a frame with IDE = 1 and ID = 0x1FFF3A20 is received, this is considered as a match
  - If a frame with IDE = 1 and ID = 0x08803220 is received, this is not a match.

41.5.7 Updating AFL Entry During Communication

You can update the AFL entry without disabling all CAN communications. Choose the entry number to be updated by setting the AFL entry number, and ignore the enable bit.

This entry number is ignored from the AFL matching while the entry is being updated.

Figure 41.27 shows the update flow for an AFL entry.

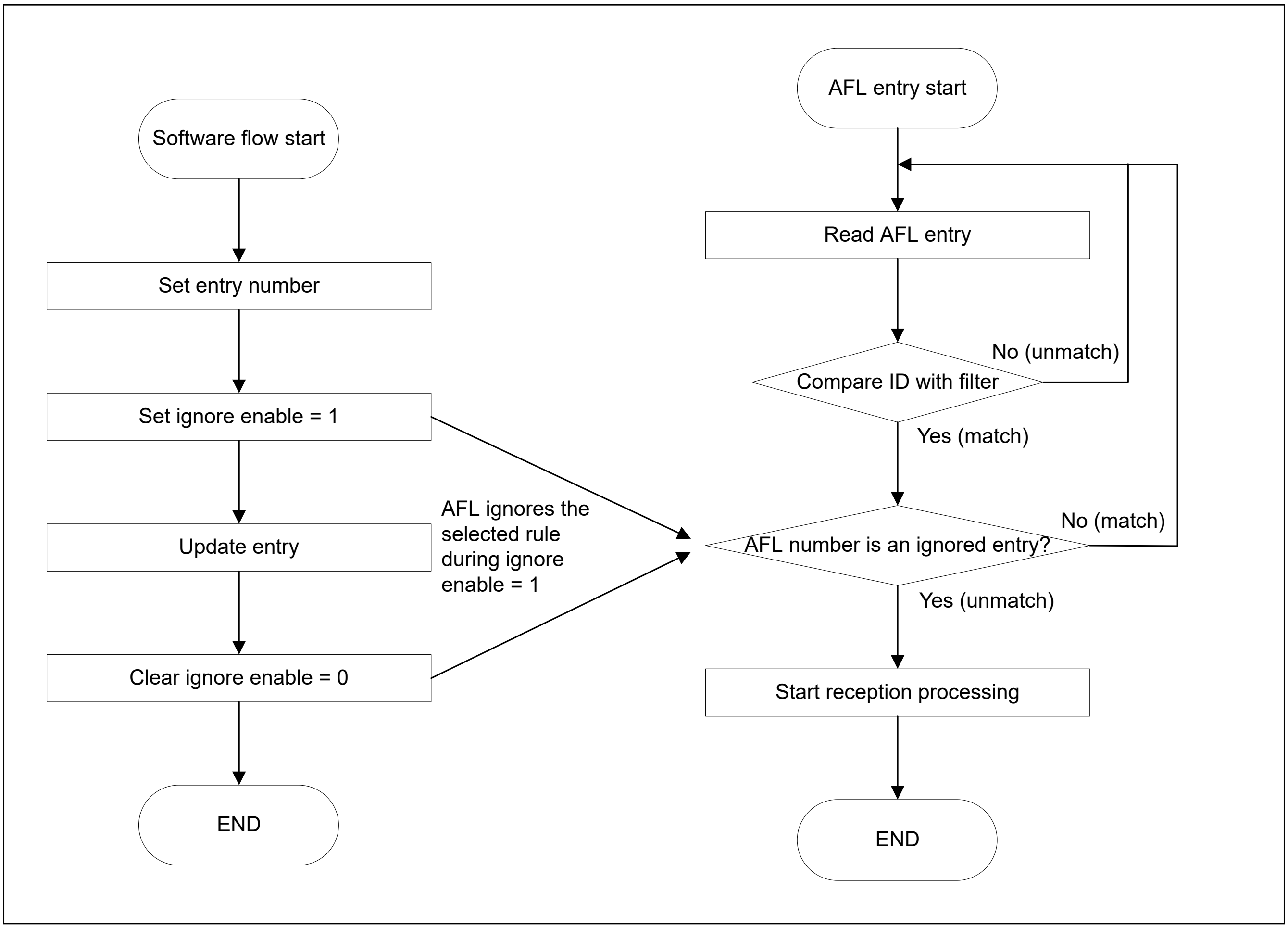


Figure 41.27 Update flow for an AFL entry

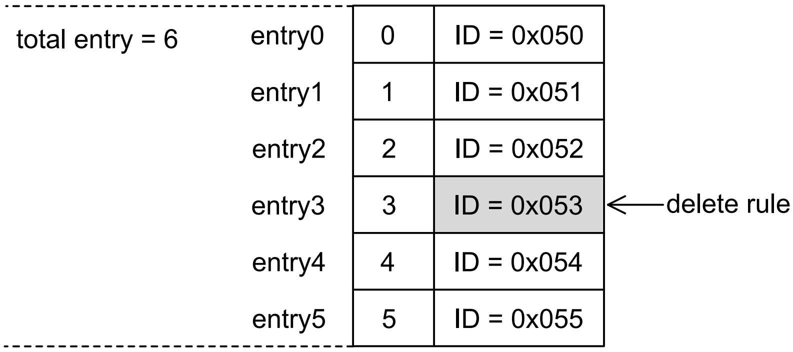
The method to update an AFL entry is as follows:

1. Set the entry number to CFDGAFLIGNENT register.
2. Set the value 0xC401 (key code and enable bit) to CFDGAFLIGNCTR register.
3. CFDGAFLECTR.AFLDAE is set to 1.
4. Set the new rule to CFDGAFLIDr, CFDGAFLMr, CFDGAFLP0r, CFDGAFLP1r registers.
5. CFDGAFLECTR.AFLDAE is cleared to 0.
6. Set the value 0xC400 (key code and clear enable bit) to CFDGAFLIGNCTR register.

Note: This entry number is ignored during the periods from (2) to (5).

(1) Example 1: Deleting an entry

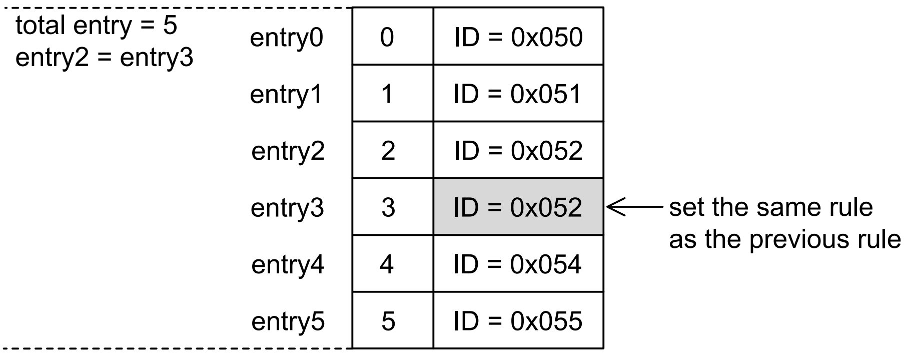
Deleting entry3 when the total number of entries is 6 channels.



How to delete an entry

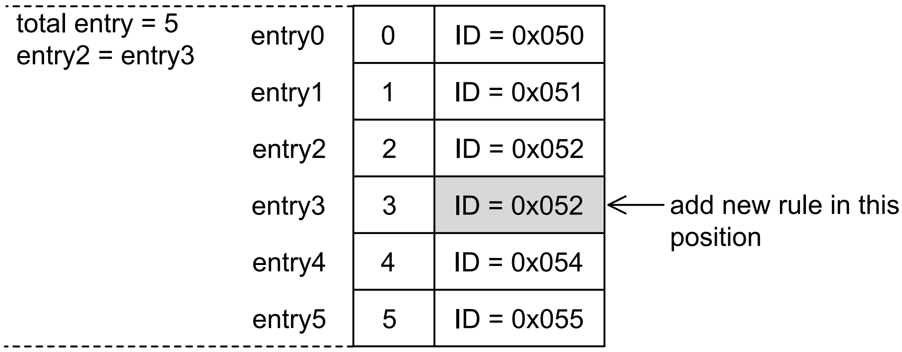
1. Set 0x00000003 to CFDGAFLIGNENT register.
2. Set 0x0000C401 to CFDGAFLIGNCTR register.
3. Set 0x00000100 to CFDGAFLECTR register.
4. Set the same rule as the previous rule by accessing CFDGAFLIDr, CFDGAFLMr, CFDGAFLP0r, CFDGAFLP1r (r = 3, this is entry3).
5. Set 0x00000000 to CFDGAFLECTR register.
6. Set 0x0000C400 to CFDGAFLIGNCTR register.

Entry3 is now deleted.



(2) Example 2: Adding an entry

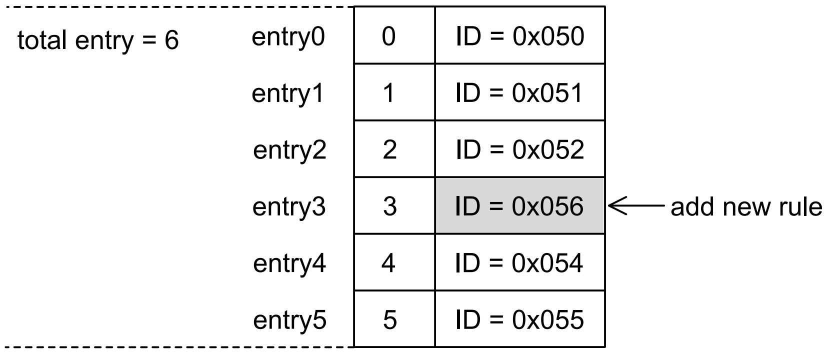
Adding a new entry to entry3 when the total number of entries is 6.



How to add an entry

1. Set 0x00000003 to CFDGAFLIGNENT register.
2. Set 0x0000C401 to CFDGAFLIGNCTR register.
3. Set 0x00000100 to CFDGAFLECTR register.
4. Set the new rule by accessing CFDGAFLIDr, CFDGAFLMr, CFDGAFLP0r, CFDGAFLP1r (r = 3, this is entry3).
5. Set 0x00000000 to CFDGAFLECTR register.
6. Set 0x0000C400 to CFDGAFLIGNCTR register.

The new entry is now added.



The AFL filter can be used to set CFDGAFLCFG, and addition/deletion of an entry is possible. Therefore, it is necessary to set the maximum number to be used to CFDGAFLCFG.

41.6 FIFO Buffers and Normal Message Buffer Configuration

This section describes the process for configuring the number of RX message buffers, the FIFO buffers, and the flat TX message buffers in the CANFD module. The message buffers are mapped as shown in Figure 41.28.

The RX message buffers can be accessed with the RX Message Buffer Registers.

The RX FIFO buffers and the common FIFO buffers configured in RX mode or TX mode can only be accessed with the FIFO Access Registers.

If the common FIFO is configured in TX mode, you can only write data into the FIFO buffer using the FIFO Access registers.

If the common FIFO is configured in RX mode, you can only read data from the FIFO Access Registers.

The TX message buffers can be accessed with the TX Message Buffer Registers.

If unused message buffer locations are read, the message buffer locations are read as unknown values.

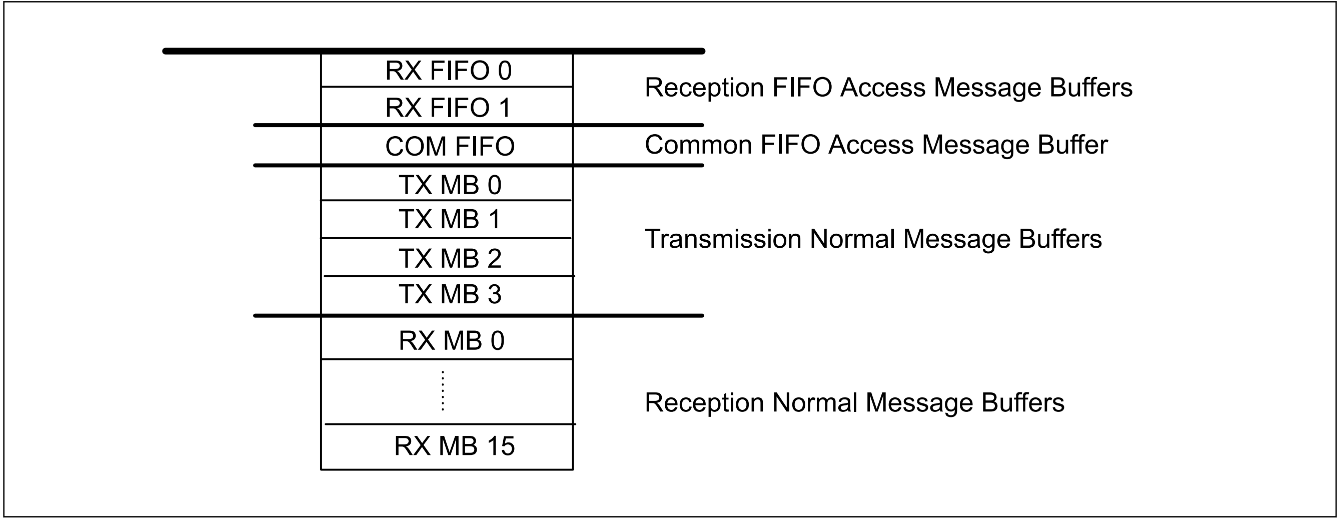


Figure 41.28 Message buffer configuration

41.6.1 Normal RX Message Buffers

In CANFD module, the frames received can be stored in normal RX message buffers based on the configuration of the AFL entries.

Additionally, the number of normal RX message buffers required in the system can be chosen up to a fixed maximum limit.

41.6.1.1 Normal RX Message Buffer Configuration

In CANFD module, the number of normal RX message buffers can be configured by writing to the RX Message Buffer Number Register.

The limiting values for the configuration of number of message buffers are:

- Minimum value = 0x00 (no normal RX MB)
- Maximum value = 0x10

Do not use values outside these limits.

The AFL entries for routing the received messages to normal RX message buffers must be configured to match the requirements of the system.

The AFL entries must also be configured properly, and an AFL entry for normal RX message buffers should not exceed the number of message buffers configured in the RX Message Buffer Number Register.

Note: There is no internal check procedure provided in CANFD module against wrong configuration of the AFL.

The data field size of the RX message buffer can be configured with the CFDRMNB.RMPLS bit. The default size is 8 bytes and the maximum data payload size is 64 bytes.

When the receiving frame exceeds the data field size, then the acceptance depends on the configuration of CFDGCFG.CMPOC (message rejecting or data payload cut).

Note: RMPLS and CMPOC bit is not available in the classical CAN function, so, these feature is not valid for classical CAN.

41.6.2 FIFO Buffers

The CANFD module provides a fixed number of FIFO buffers to support storage of frames for reception and transmission functions.

The number of reception-only FIFO buffers is fixed to 2. However, common FIFO buffer channel can be configured to store messages for transmission or reception function.

These FIFO buffers can be enabled or disabled, and the following parameters can be configured to match the system requirements:

- Size
- Interrupt structure
- Message lost mechanism
- Message overwrite mechanism of the FIFO buffers
- Location of the TX FIFO.

When the receiving frame exceeds the data field size, the acceptance depends on the configuration of the CFDGCFG.CMPOC bit (message rejecting or data payload cut).

41.6.2.1 FIFO Buffers Configuration

In CANFD module, the FIFO buffers can be configured to match the system requirements.

The total number of FIFO buffers = 2 RX FIFO buffers + 1 common FIFO buffer = 3 FIFO buffers.

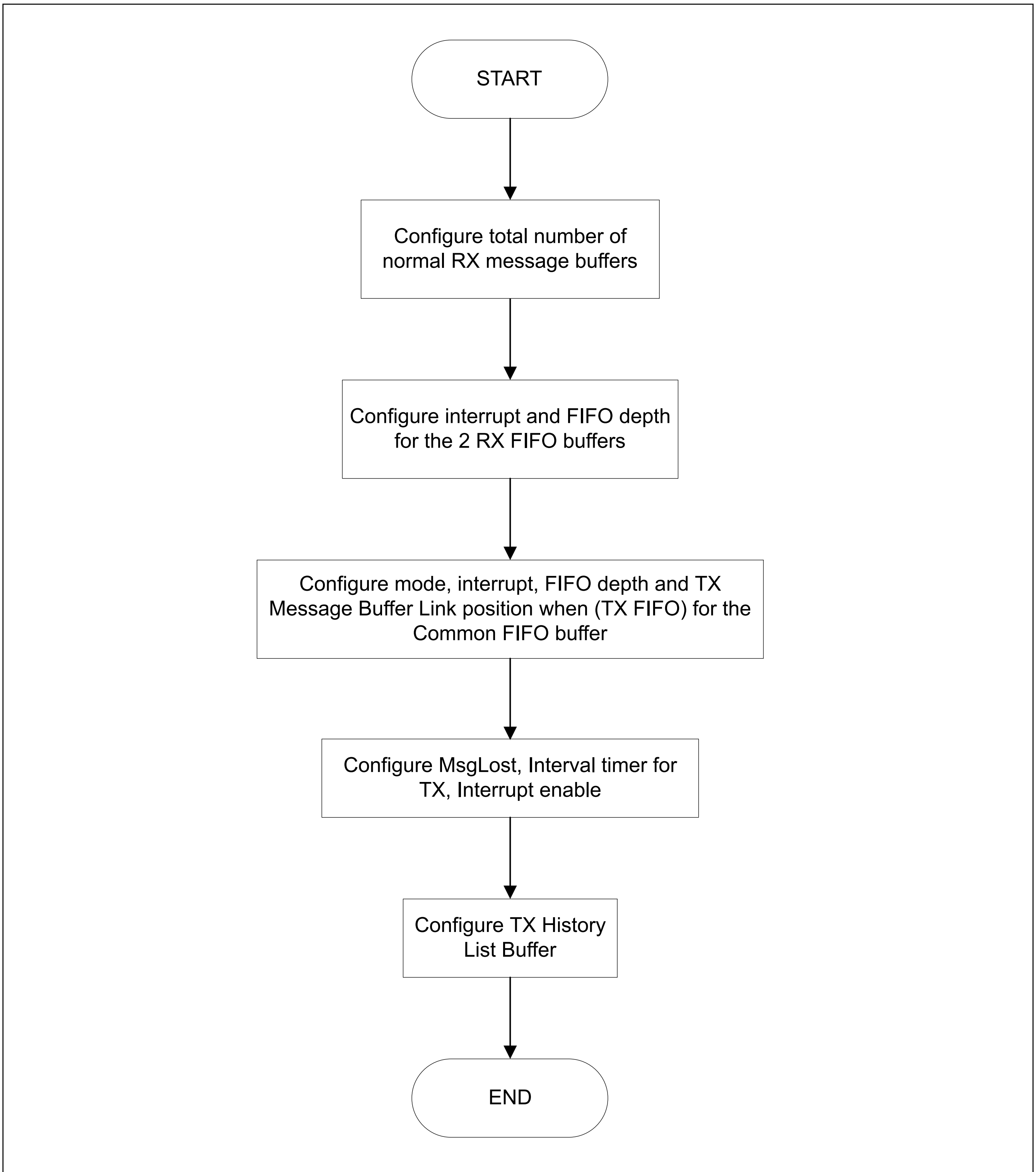


Figure 41.29 FIFO buffer configuration flow in CANFD module

As shown in Figure 41.29, the various FIFO buffers can be configured by writing to the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers.

For the 2 RX FIFO buffers, the following parameters can be configured:

- Interrupts
- FIFO depth
- FIFO payload data size.

For the common FIFO buffer, the following parameters can be configured:

- Mode
- Interrupts FIFO depth
- FIFO payload data size
- FIFO TX link position.

(1) FIFO mode configuration of Common FIFO buffer

The mode of the common FIFO buffer can be configured by writing to the CFDCFCC.CFM[1:0] bits in the Common FIFO Configuration/Control Register. The possible modes of configuration for Common FIFO buffer are:

- 0b RX mode (default mode after hardware reset)
- 1b TX mode

Messages can only be read from the RX FIFO buffers and the Common FIFO buffer configured in RX mode. Messages are stored by the CAN module in these FIFO buffers based on the AFL entries.

Messages can be read and written into the Common FIFO buffer configured in TX mode. These messages are transmitted on the appropriate CAN channel.

The pointers can only be incremented when a new message is stored in the FIFO buffer and decremented when a message is transmitted on the corresponding CAN channel by the CANFD module.

After a hardware reset, the Common FIFO buffer is configured in RX mode by default. Only enable the FIFO buffers after configuring the Common FIFO buffer in the required modes.

(2) FIFO TX message buffer link configuration

When the common FIFO is configured as TX FIFO, the FIFO buffer must be linked to a normal TX message buffer to participate in the transmission scan.

Do not write data into a TX message buffer that is linked to a Common FIFO buffer. Also, the TX message buffer linked to a Common FIFO buffer should not be a part of the TX Queue.

The TX message buffer link of each Common FIFO buffer can be configured by writing to the CFDCFCC.CFTML[1:0] bits in the Common FIFO Configuration/Control Registers. Available options for TX message buffer link configuration are:

- 0x00: TX Message Buffer 0
- 0x01: TX Message Buffer 1
- 0x10: TX Message Buffer 2
- 0x11: TX Message Buffer 3

(3) FIFO depth configuration

The depth of each FIFO buffer can be configured by writing to the CFDRFCCa.RFDC[2:0] bits and CFDCFCC.CFDC[2:0] bits in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers. The 6 available options for depth configuration are:

- 0x000: 0 Message (FIFO buffer cannot be enabled)
- 0x001: 4 Messages
- 0x010: 8 Messages
- 0x011: 16 Messages
- 0x100: 32 Messages
- 0x101: 48 Messages

The RAM allocation for RX message buffers along with FIFO buffers is limited to 16 messages with 64 data bytes. Configuration of the RX message buffers, along with FIFO buffers, that exceeds this maximum limit should not be done.

CANFD module logic does not check the validity of the configuration.

Note: If the FIFO depth of a common FIFO is 4 messages or more (CFDCFCC.CFDC[2:0] > 000b), then the Common FIFO TX message buffer link is valid when the FIFO is disabled or enabled.

If FIFO depth is 0 messages, then the Common FIFO TX message buffer link is not valid when the FIFO is disabled or enabled.

(4) FIFO payload size configuration

The data size of each FIFO buffer can be configured by writing to the CFDRFCCa.RFPLS[2:0] bits and CFDCFCC.CFPLS[2:0] bits in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers. The eight available options for depth configuration are:

- 000b: 8 bytes
- 001b: 12 bytes
- 010b: 16 bytes
- 011b: 20 bytes
- 100b: 24 bytes
- 101b: 32 bytes
- 110b: 48 bytes
- 111b: 64 bytes

The RAM allocation for RX message buffers along with FIFO buffers is limited to 16 messages with 64 data bytes. Configuration of the RX message buffers, along with FIFO buffers, that exceeds this maximum limit should not be done.

CANFD module logic does not check the validity of the configuration.

Note: This feature is not available in the classical CAN function.

(5) FIFO interrupt configuration

The Interrupt generation conditions for the FIFO buffers can be configured by writing to the CFDRFCCa.RFIM and CFDCFCC.CFIM bit in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers. The two available options are:

- 0:
  - RX FIFO mode: Interrupt generated when the Common FIFO counter reaches CFDRFCCa.RFIGCV/ CFDCFCC.CFIGCV value
  - TX FIFO mode: Interrupt generated when the Common FIFO transmits the last message successfully
- 1:
  - RX FIFO mode: Interrupt generated at the end of storage of every received message
  - TX FIFO mode: Interrupt generated for every successfully transmitted message

If the Interrupt Mode bit is 0 for a RX FIFO, then interrupt is generated based on the configuration of the CFDRFCCa.RFIGCV[2:0] bits.

Similarly, if the Interrupt Mode bit is 0 for a Common FIFO configured in RX mode, then interrupt is generated based on the configuration of CFDCFCC.CFIGCV[2:0] bits.

The eight available options for configuring the FIFO counter value for generation of an interrupt are:

- 000b: Interrupt generated when FIFO is 1/8th Full
- 001b: Interrupt generated when FIFO is 1/4th Full
- 010b: Interrupt generated when FIFO is 3/8th Full
- 011b: Interrupt generated when FIFO is 1/2 Full
- 100b: Interrupt generated when FIFO is 5/8th Full
- 101b: Interrupt generated when FIFO is 3/4th Full
- 110b: Interrupt generated when FIFO is 7/8th Full
- 111b: Interrupt generated when FIFO is Full.

In this case, an interrupt is generated when the message count matches the configured value.

However, there are some limitations on the configuration of the CFDRFCCa.RFIGCV[2:0] and CFDCFCC.CFIGCV[2:0] bits depending on the FDC[2:0] bits (FIFO Depth Configuration), see Table 41.23.

Table 41.23 FIFO interrupt generation counter and FIFO depth configuration

RFDC[2:0] (CFDC[2:0])	RFIGCV[2:0] (CFIGCV[2:0])							
	111b	110b	101b	100b	011b	010b	001b	000b
000b	Don’t care (FIFO cannot be enabled)							
001b	Allowed	Not allowed	Allowed	Not allowed	Allowed	Not allowed	Allowed	Not allowed
010b	Allowed							
011b	Allowed							
100b	Allowed							
101b	Allowed							
110b	Allowed							
111b	Allowed							

41.6.2.2 FIFO Buffers Control

The FIFO interrupt must be enabled by setting any one of the following bits in the RX FIFO Configuration/Control Registers:

- CFDRFCCa.RFIE

In addition, the FIFO interrupt must be enabled by setting any one of the following bits in the Common FIFO Configuration/Control Register:

- CFDCFCC.CFRXIE
- CFDCFCC.CFTXIE

After configuration is complete, each FIFO can be enabled by setting the CFDRFCCa.RFE and CFDCFCC.CFE bits in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Register to allow transmission and reception of messages.

41.7 Interrupts and DMA

41.7.1 Interrupts

The CANFD module generates several interrupts. The interrupt output, which is connected to the Interrupt Controller Unit (ICU), can be controlled by the corresponding interrupt enable bit.

The status flag is set independent of this enable bit.

The channel transmission interrupt has an additional status flag register. The status bits are set when the corresponding interrupt enables are set.

The status flag register supports the identification of the interrupt source for the channel transmission, as this interrupt is driven by several trigger sources.

The interrupts in the CANFD module can be classified into two groups, global interrupts and channel interrupts:

- Global interrupts:
  The CANFD module can generate 3 global interrupts:
  - Global interrupt for successful reception into the 2 RX FIFO buffers
  - Global error interrupt.
  - Global Interrupt for successful reception into the 16 RX message buffers
- Channel interrupts:
  Channel of the CANFD module can generate 3 channel interrupts:

1. Channel transmission
   - Transmission completion from channel
   - Transmission abort from channel
   - Transmission from TX Queue for a channel
   - Channel THL interrupt
   - Successful transmission from a Common FIFO in TX mode for a channel.
2. Channel error interrupt
3. Successful reception in a Common FIFO in RX mode for a channel.

The interrupts are cleared when the corresponding flag bits are cleared or the Interrupt enable bits are cleared.

Table 41.24 gives an overview of interrupt sources for the different interrupt outputs. The interrupt outputs are active-high.

Table 41.24 Interrupt source overview

Parameter	Interrupt	Name	Interrupt source	Interrupt clearing
Global Interrupts	Successful reception into at least one RX FIFO	CAN_RXF	Interrupt flag of corresponding RX FIFO for which interrupt is enabled	Clear the interrupt flag of corresponding RX FIFO buffer for which interrupt is enabled
	Global Error	CAN_GLERR	Any of the following: ● DLC Error flag ● Message Lost Status bit ● TX History Entry Lost Status bit ● CANFD Message Payload overflow flag	Clear all of : ● DLC Error flag ● Message Lost flags in all of the FIFO Status Registers ● TX History List Entry Lost flag ● CANFD Message Payload overflow flag
	Successful reception into at least one RXMB	CANn_RXMB (n = 0, 1)	Interrupt flag of corresponding RXMB for which interrupt is enabled	Clear the interrupt flag of corresponding RXMB buffer for which interrupt is enabled
Channel Transmission Interrupts	Channel successful transmission	CANn_TX (n = 0, 1)	Any channel related TXMB Successful flag when interrupt is enabled*1	Clear all channel related TXMB Result status bits for which the interrupt is enabled
	Channel Abort		Any channel related TXMB Abort flag when interrupt is enabled*1	Clear all channel related TXMB Result Status bits for which the interrupt is enabled globally
	Channel transmission from TX Queue		Related channel TX Queue Interrupt flag	Clear related channel TX Queue Interrupt flag
	Channel THL Interrupt		Channel THL Interrupt status flag	Clear the relevant THL Interrupt status flag
	Channel COM FIFO TX Interrupt		Interrupt Flag for Common FIFOs in TX mode belonging to the related channel	Clear the interrupt flags of Common FIFOs in TX mode belonging to the related channel
Channel Error Interrupt	Channel Error	CANn_CHERR (n = 0, 1)	Any channel related error flag in the Channel Error Flag Register for which interrupt is enabled in the Channel Error Interrupt Enable Register	Clear all channel related error flags in the Channel Error Flag Register for which interrupt is enabled in the Channel Error Interrupt Enable Register
Channel COM RX FIFO Interrupt	Channel COM FIFO RX Interrupt	CANn_COMFRX (n = 0, 1)	Interrupt flag for Common FIFOs in RX mode belonging to the related channel	Clear the interrupt flags of Common FIFOs in RX mode belonging to the related channel

Note 1. These interrupts are only set for TX Message Buffers that do not belong to an enabled TX Queue and are not pointing to a common FIFO.
Separate interrupts are provided for common FIFO buffers and TX Queue.

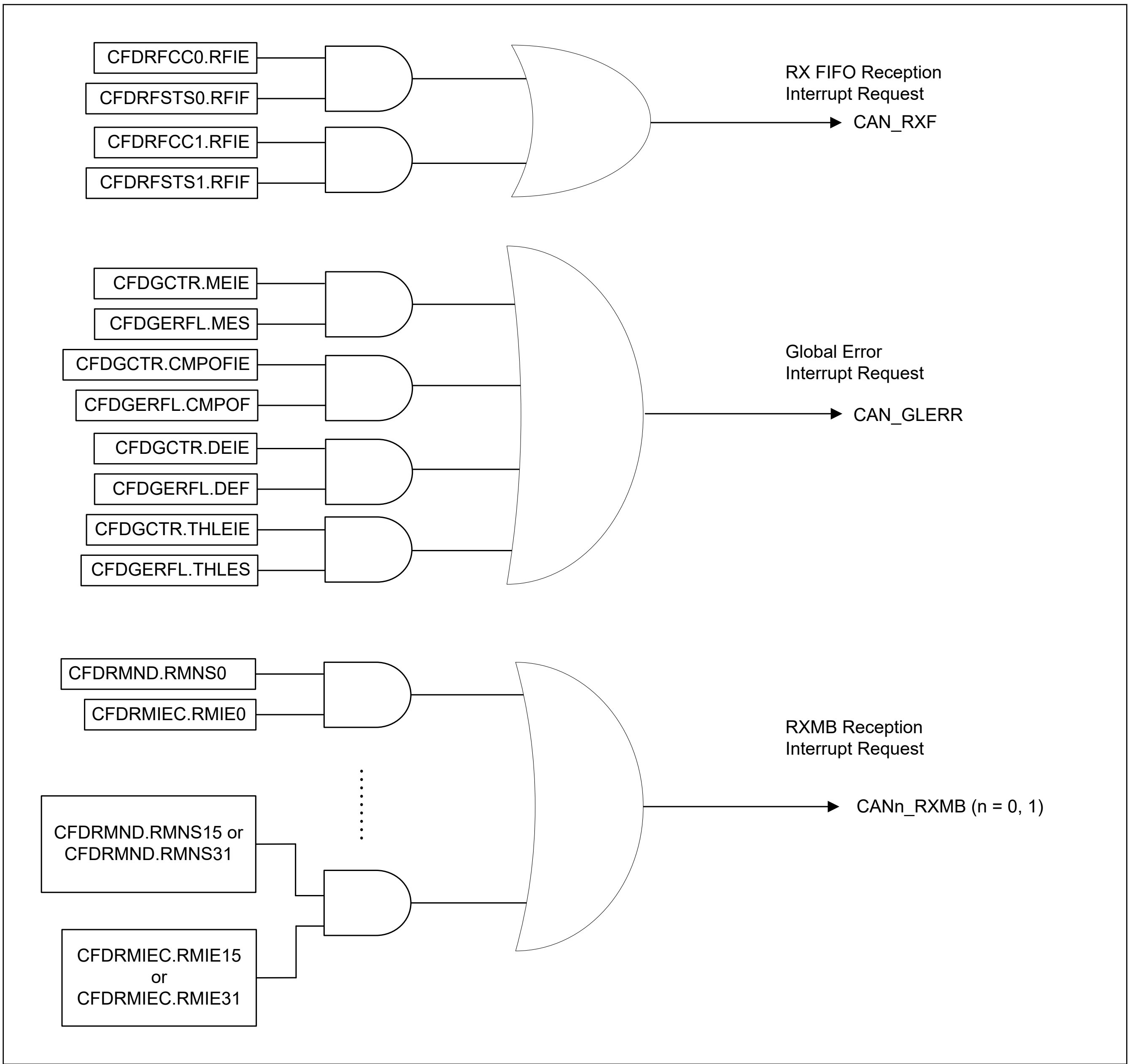


Figure 41.30 Global interrupt block diagram

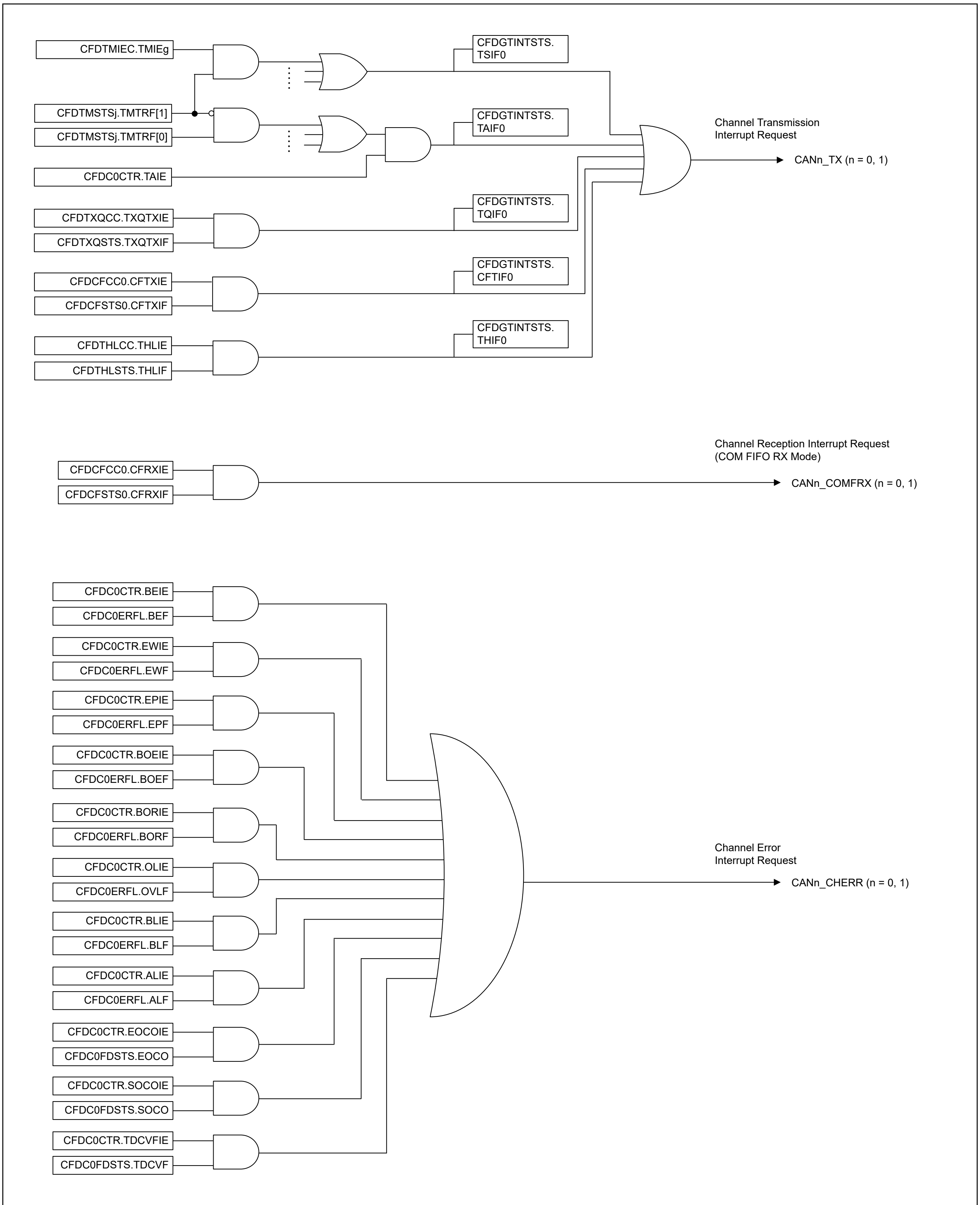


Figure 41.31 Channel interrupt block diagram

41.7.2 DMA Transfer

The CANFD module has message buffers that can be associated with a DMA channel:

- Reception DMA

– 2 RX FIFO message buffers
– Common FIFO Message Buffer

Figure 41.32 shows the potential DMA channels.

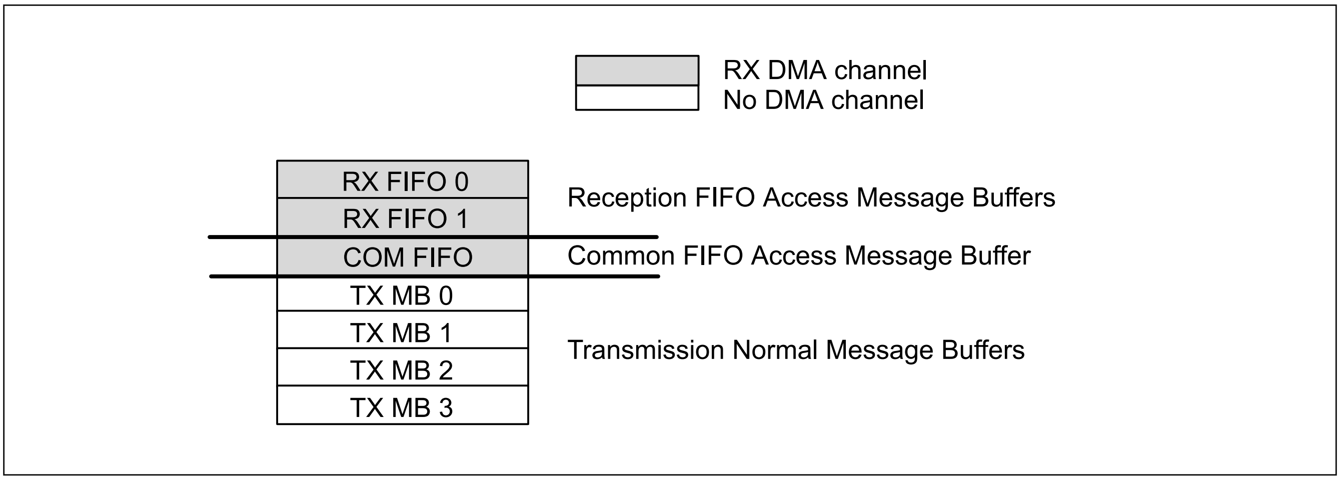


Figure 41.32 Message buffer connectable to a DMA channel

A DMA channel transfer request is generated for each FIFO entry to the DMAC when the related CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE is set to 1 and the belonging FIFO is not empty.

Reception FIFO Interrupt should be disabled for this particular FIFO (CFDRFCCa.RFIE or CFDCFCC.CFRXIE)

Use the regular start address for the DMA access window address. See Figure 41.33.

Table 41.25 DMA channel access window address

b = Message buffer component index	Message Buffer Component	Register	P	Regular Start Address
b = [0…1]	RFMBCPb[0]	CFDRFIDb	x	0x0520 + b × 0x004C
		CFDRFPTRb	x	0x0524 + b × 0x004C
		CFDRFFDSTSb	x	0x0528 + b × 0x004C
		CFDRFDFbp	[0…15]	0x052C + p × 0x0004 + b × 0x004C
—	CFMBCP0[0]	CFDCFID	x	0x05B8
		CFDCFPTR	x	0x05BC
		CFDCFFDCSTS	x	0x05C0
		CFDCFDFp	[0…15]	0x05C4 + p × 0x0004

DMA FIFO pointer decrement is done automatically by reading the last configured data payload byte (CFDRFCCa.RFPLS or CFDCFCC.CFPLS).

Note: The DMA must read the exact length of the configured data payload size (CFDRFCCa.RFPLS or CFDCFCC.CFPLS).

Note: This feature is not available for classical CAN function because CFDRFCCa.RFPLS and CFDCFCC.CFPLS are not in classical CAN.

Do not write to the FIFO control registers when DMA is enabled. The DMA enable of the particular DMA FIFO (CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE) can be set at any time. Figure 41.33 shows a configuration flow for an initial setup.

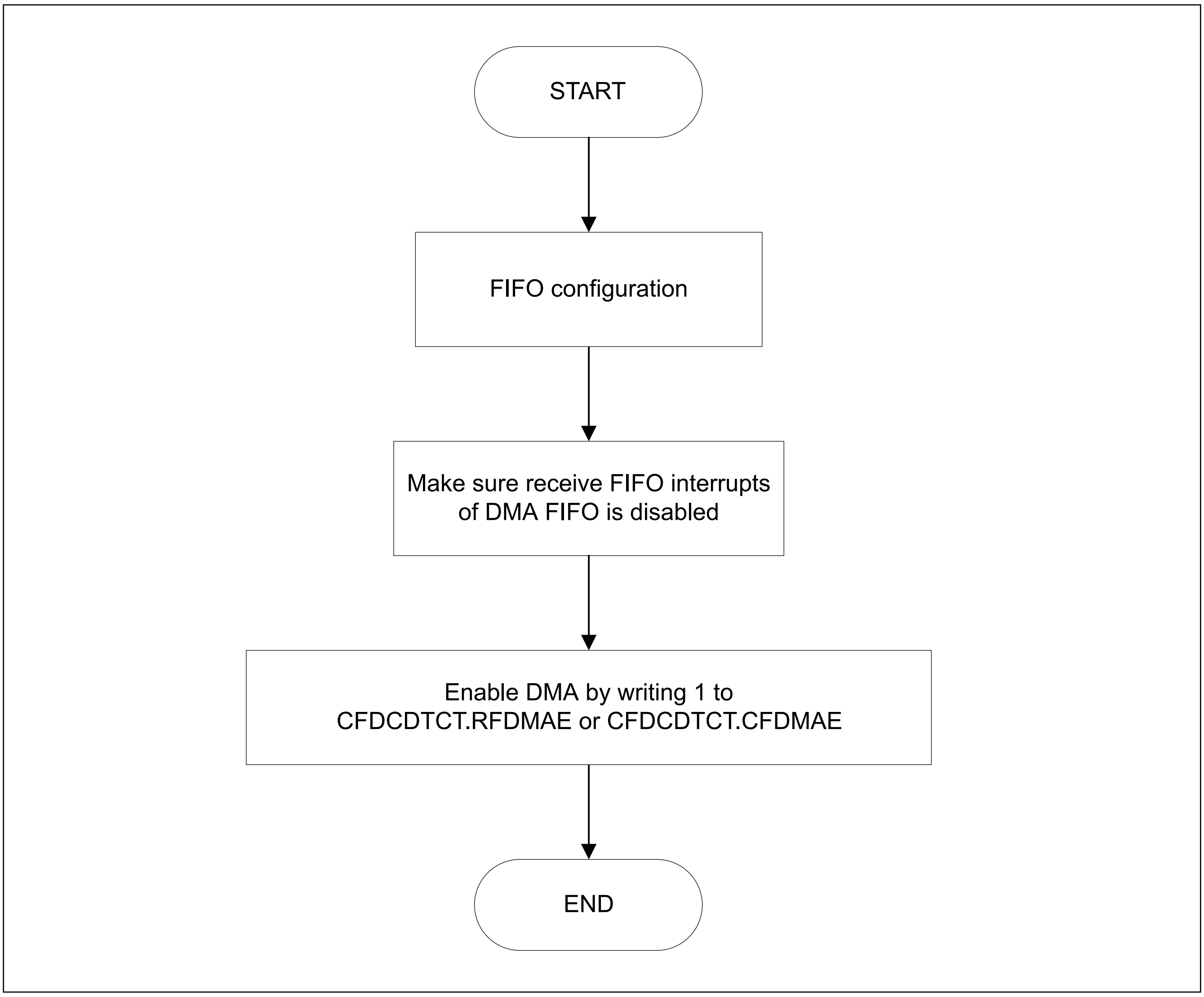


Figure 41.33 DMA enable flow

To disable a DMA transfer request, you must disable the particular DMA enable bit (CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE). If the disable is made during an ongoing transfer, then the transfer must be completed first before further action can be taken. The transfer status can be identified by the CFDCDTSTS.RFDMASTS or CFDCDTSTS.CFDMASTS bit. See Figure 41.34 for the DMA disable flow. When the DMA is disabled, consideration should be made for the remaining or new incoming messages to this particular reception FIFO.

When the FIFO is not disabled, reception to the FIFO continues.

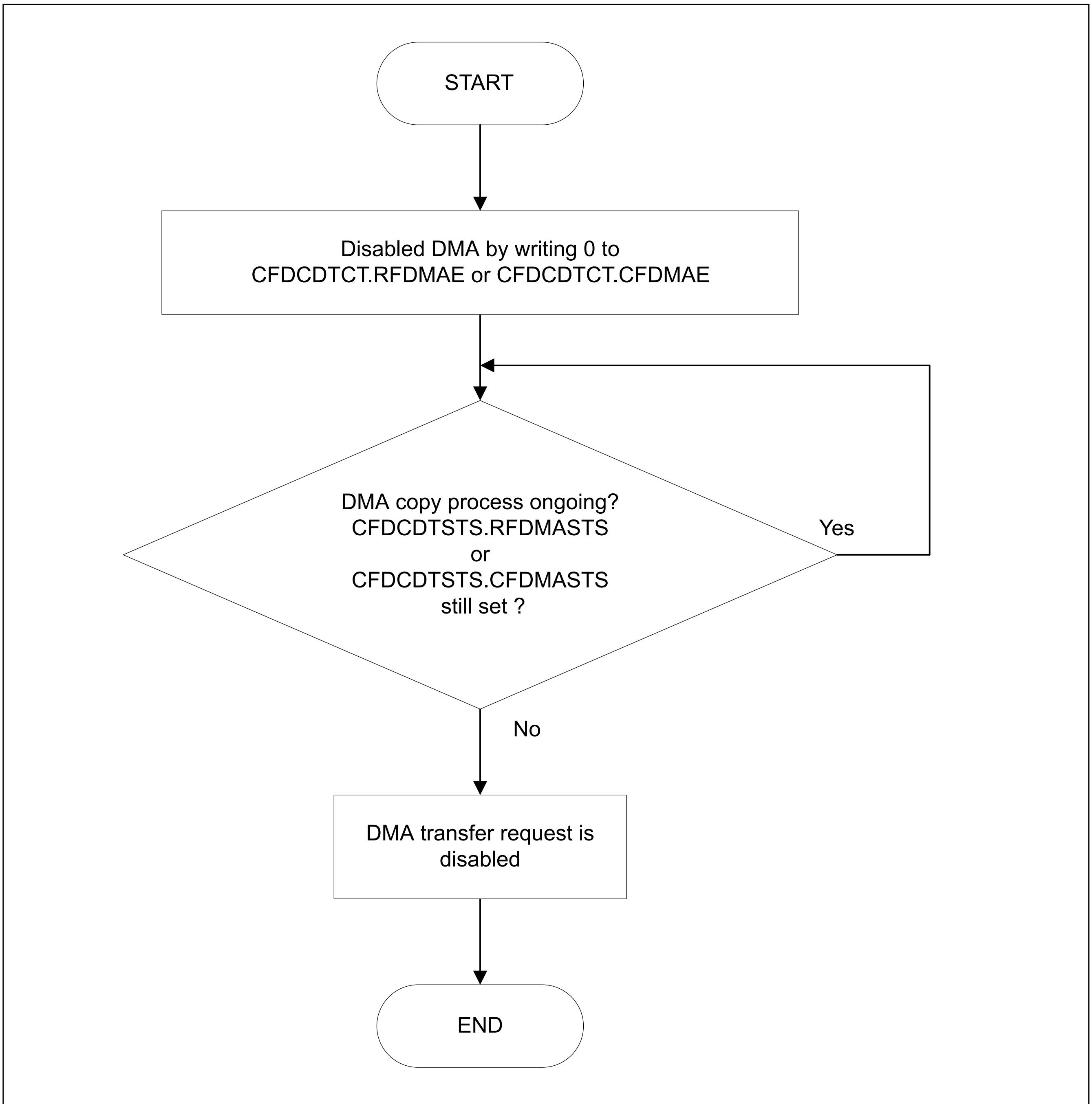


Figure 41.34 DMA disable flow

41.8 Reception and Transmission

41.8.1 Reception

In the CANFD module, CAN messages received on any of the channels are stored in RX message buffers, RX FIFO buffers, or Common FIFO buffers configured in RX mode depending on the Acceptance Filter List entries.

- Up to 16 RX message buffers can be configured
- 2 RX FIFO buffers available
- 1 Common FIFO Buffer can be configured in RX mode

41.8.1.1 Message Storage in RX Message Buffers

When a message is successfully received and stored in a RX message buffer, the corresponding New Data flag is set in the RX Message Buffer New Data Register.

The CAN message can be read from the corresponding RX message buffer.

If a new message is stored into a RX message buffer before the previous message in this message buffer can be read, then the original message is overwritten. There is no mechanism for preventing a new message from overwriting the current message in the RX message buffer. If such a loss of messages is not acceptable, then RX FIFO should be used for storing related messages.

Note: Users should do the same processing as the existing software flow also when using interrupt. (see Figure 41.36)

Note: Unused data bytes are filled with 0x00 depending on the DLC value.

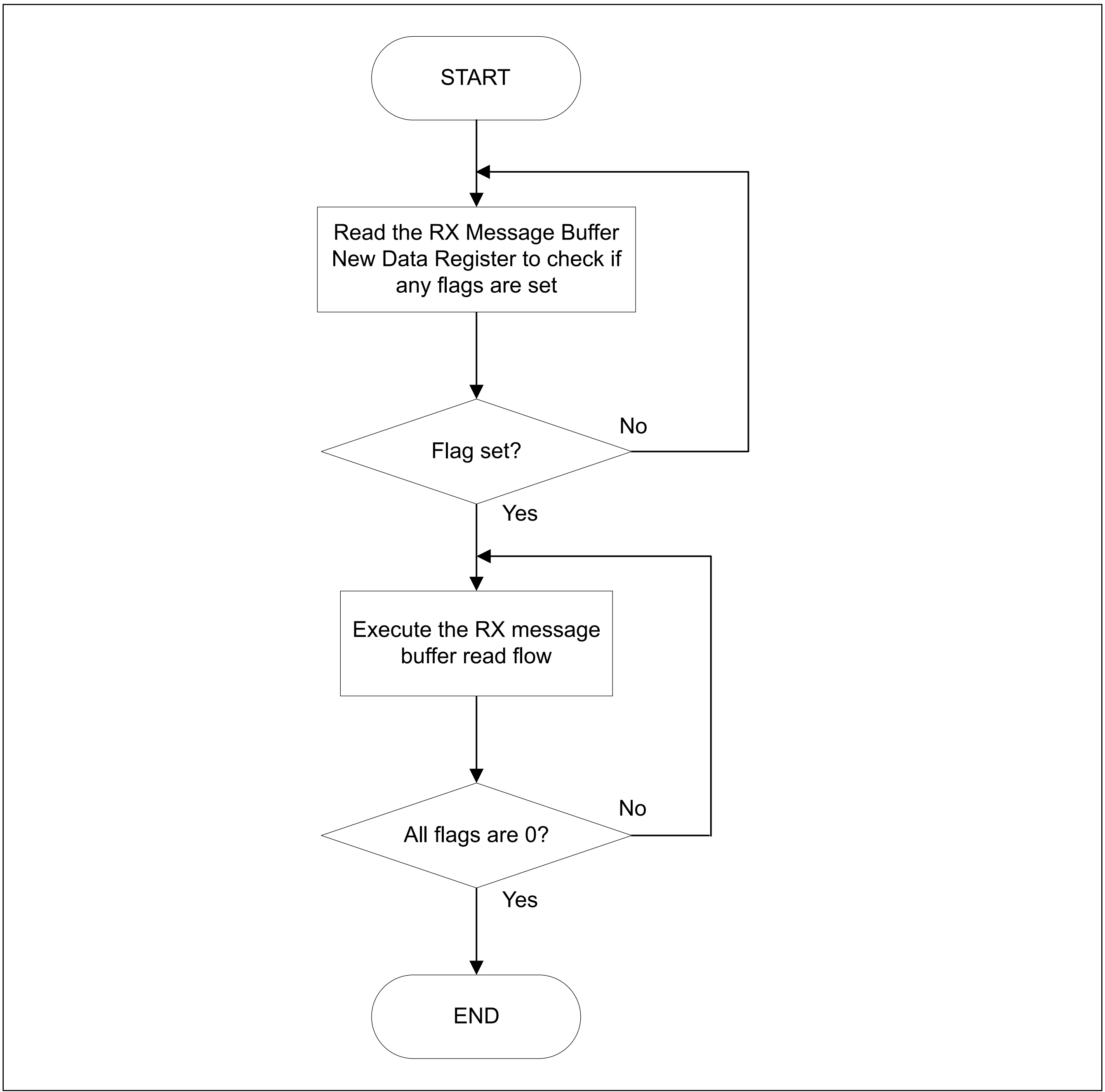


Figure 41.35 Access flow of RX message buffer (Polling)

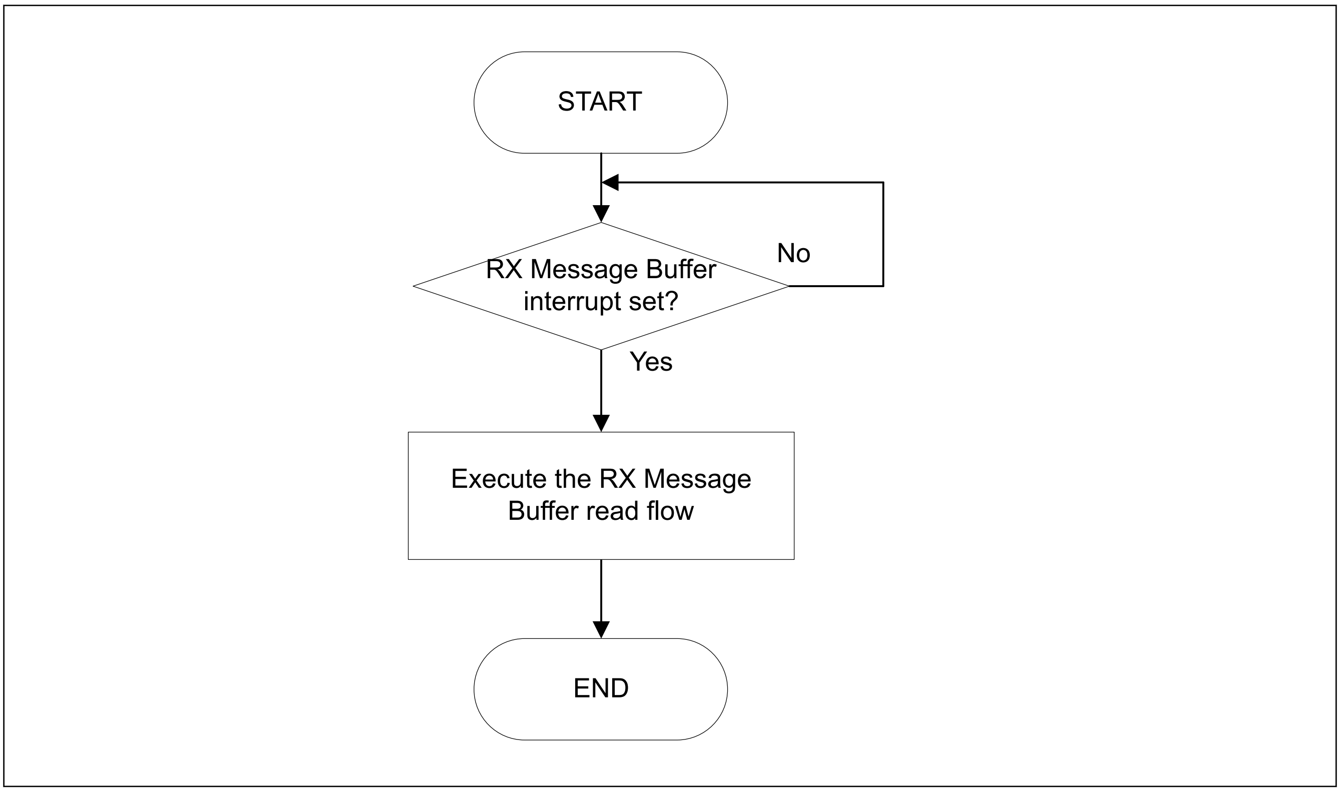


Figure 41.36 RX Message Buffer Message Access Flow (interrupt)

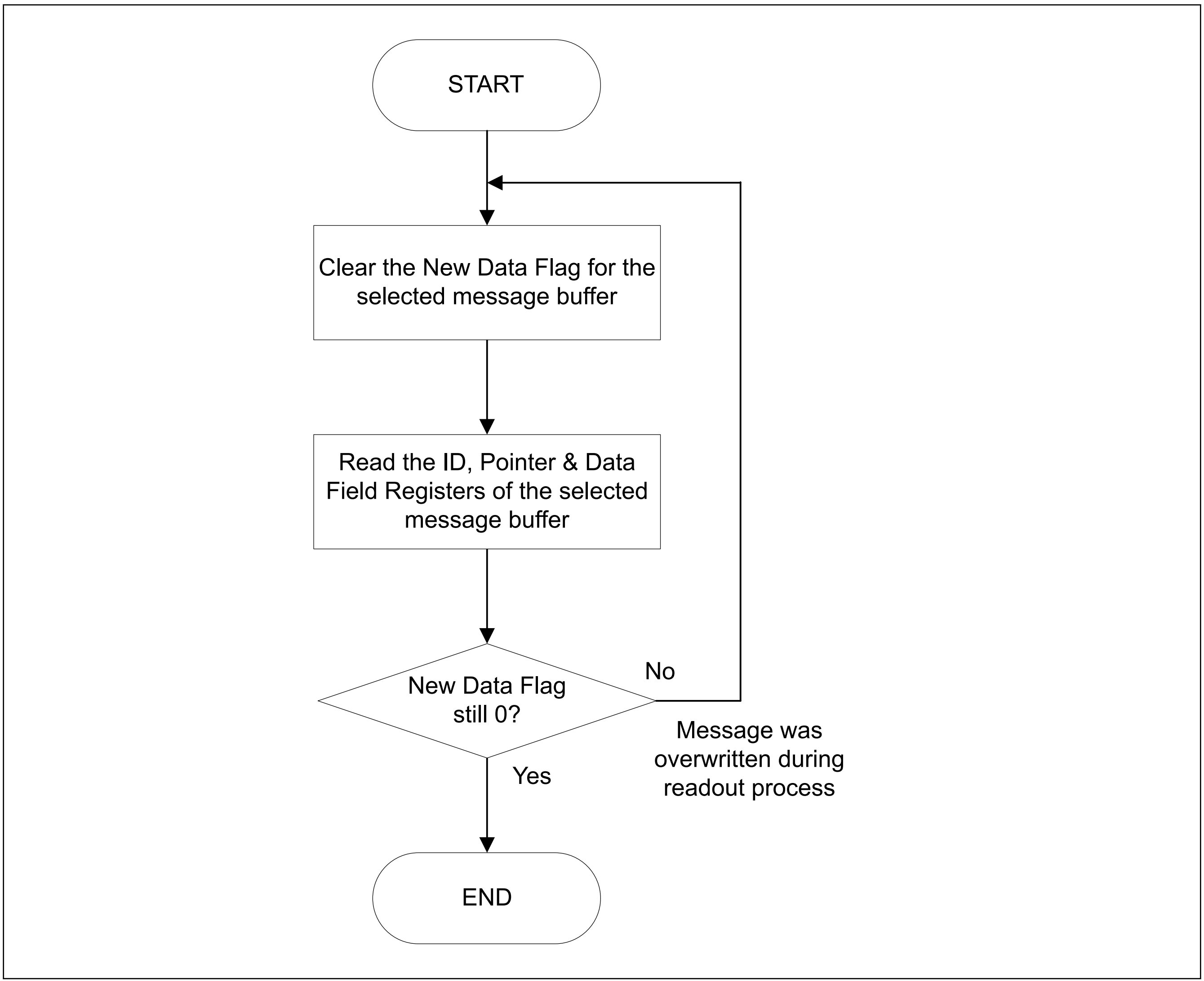


Figure 41.37 Read flow of RX message buffer

41.8.1.2 Message Storage in FIFO Buffers

The AFL entries for routing the received messages to RX FIFO buffers or Common FIFO buffer configured in RX mode should be configured based on system requirements.

The CFDGAFLP1r.GAFLFDP[8,1:0] field in the matching AFL entry selects the FIFO buffers to which the related reception message is stored.

When the received message is stored in one or more RX FIFO buffers or Common FIFO buffer configured in RX mode, the message counter value is incremented in the corresponding RX FIFO Status Registers or Common FIFO Status Register.

Depending on the configuration of the FIFO buffers, an interrupt might also be generated.

The message can be read from the corresponding FIFO Access registers.

Note: Because many messages can be stored in the FIFO buffers, reading more than one message may be required to read the latest message stored in a FIFO buffer.

If the message count value matches the FIFO depth, the FIFO Full flag is set.

When the value 0xFF is written to the corresponding FIFO Pointer Control Register, the message count is decremented by 1.

Only write 0xFF to the FIFO Pointer Control register after reading the complete message from the FIFO Access registers of the corresponding FIFO.

When all the messages stored in the FIFO are read, the FIFO Empty flag is set.

If a new message is stored into the FIFO when the FIFO message count matches the FIFO depth (FIFO full condition), the FIFO Message Lost flag is set and the new message is lost (no overwrite of already stored messages takes place).

An appropriate value can be configured as warning level to generate an interrupt before the FIFO full condition occurs to avoid loss of a message due to an overrun condition.

Note: The message lost can be set only in RX mode by CAN, and the flag is not set when the CPU is overloading the FIFO buffers.

The RX FIFO buffers and the Common FIFO buffers configured in RX mode can be disabled at any time by clearing the CFDRFCCa.RFE or CFDCFCC.CFE bit in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers.

When the CFDRFCCa.RFE or CFDCFCC.CFE bit is cleared, the message read and write pointers of the FIFO are cleared and are no longer active. Therefore, all messages in the FIFO buffers are lost and no further messages can be stored into the FIFO.

When the RX FIFO buffers or Common FIFO buffer configured in RX mode is assigned as a DMA channel, software should not access the FIFO Access Register of this FIFO buffer or write 0xFF to the FIFO Pointer Control Register (CFDCFPCTR.CFPC or CFDRFPCTRa.RFPC). This can lead to unintended FIFO message decrement. The DMA channel controls the FIFO decrement automatically.

Note: If the interrupt flag is set for a FIFO buffer and then the FIFO is disabled, the interrupt flag is not cleared automatically. The interrupt flag should be cleared before disabling the FIFO.

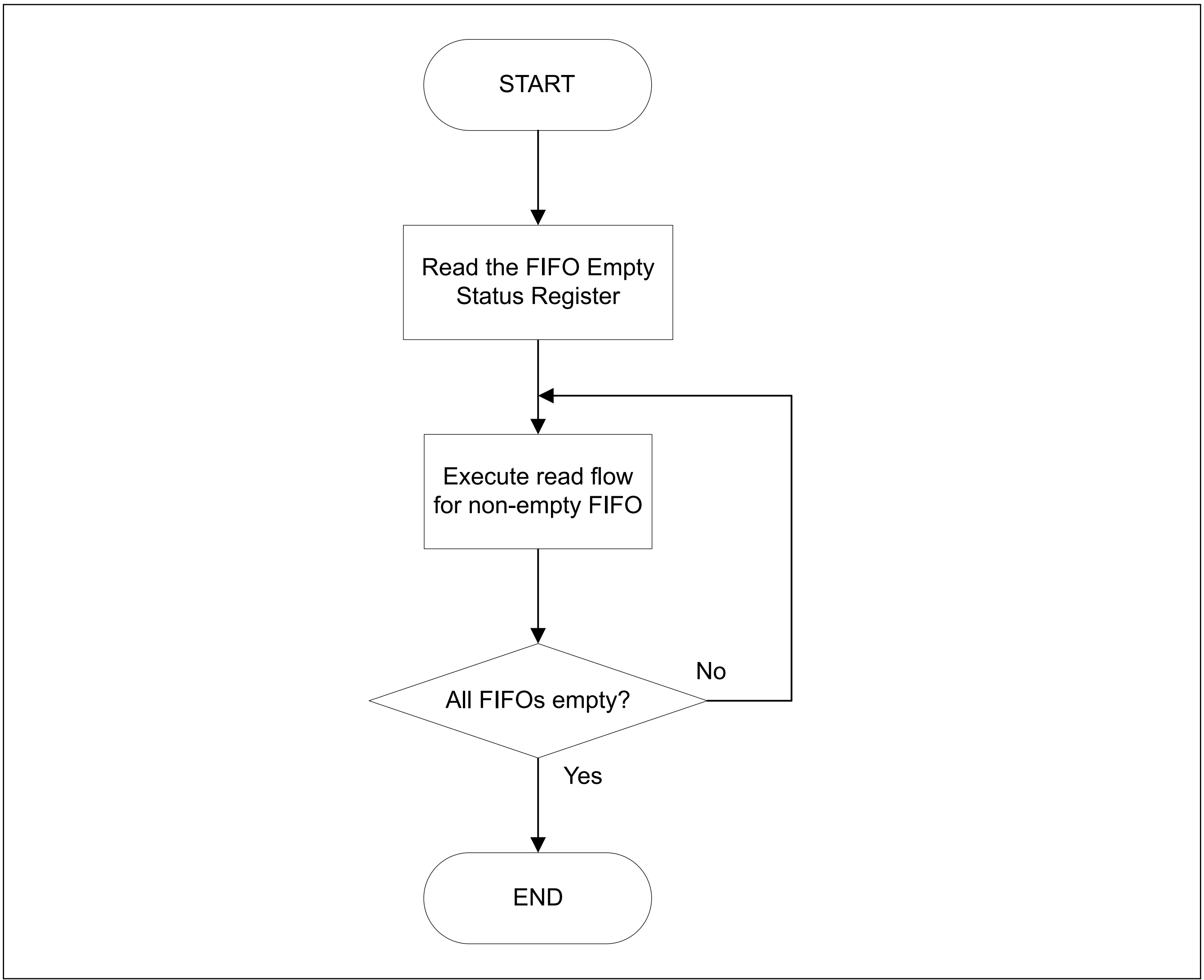


Figure 41.38 Access flow of FIFO buffer message (example for polling case)

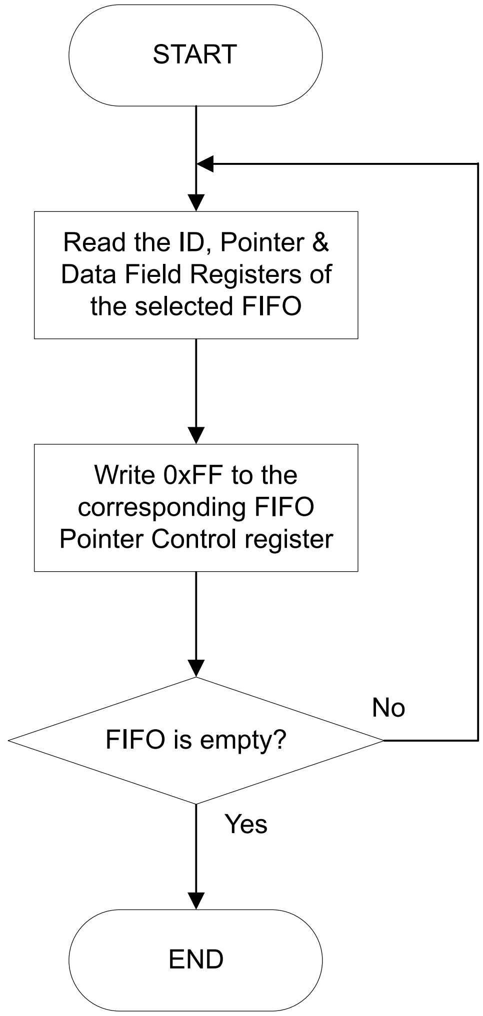


Figure 41.39 Read flow of RX FIFO buffer message (example for polling case)

Note: When the next frame is received before clearing the completion interrupt flag of reception, the completion interrupt of reception is not set again.

Even when an interruption flag is cleared after the completion processing of reception, the already received interrupt flag is not set.

It is necessary to perform the completion processing of reception even before the next completion of frame reception, and to clear an interruption flag.

When processing does not meet the condition, after checking that receiving data is empty, interrupt flag is cleared, and it checks that receiving data is empty again.

41.8.1.3 Timestamp

The timestamp counter is a free-running counter that can be used to check reception time of an incoming message or transmission time of successful transmitted messages. The Timestamp counter value is captured based on the CFDGFDCFG.TSCCFG[1:0] configuration (at the sample point of start of frame, point in time when the frame is valid, or for CANFD frames also at the sample point of the RES bit). For reception, it is stored together with the message ID and data into the target RX message buffer or RX FIFO.

For transmit message, the timestamp counter value is stored as part of the TX History List entry.

The counter can be clocked with the peripheral clock or with the CAN channel bit timing clock. The counter source clock can be configured with the CFDGCFG.TSSS bit of the Global Configuration Register. If this bit is 0, the peripheral clock is used. If the bit is 1, the selected CAN channel bit time clock is used.

The channel selection is performed with the CFDGCFG.TSBTCS bit of the Global Configuration Register.

Care must be taken when using selected CAN channel bit time clock as the clock source. When entering Channel Halt mode or Channel Reset node, for this channel, the timestamp counter is stopped. For other CAN channels, the timestamp counter value is not updated.

If peripheral clock is selected as the timestamp counter clock source, Channel modes do not affect the timestamp counter function.

The source clock for the timestamp counter can be divided by a factor defined by the CFDGCFG.TSP bits (timestamp prescaler) in the Global Configuration Register.

The timestamp counter can be reset to 0x0000 with the CFDGCTR.TSRST bit (timestamp reset).

41.8.2 Transmission

There are several possible transmission configurations:

- Normal transmission
- FIFO transmission
- TX Queue transmission
  A fixed number of transmission message buffers (4 TX message buffers) are dedicated. These message buffers are only used for transmission and cannot be configured for reception.
  Additionally, transmission from TX Queue or Common FIFO in TX mode can be configured in the following way (see Figure 41.40):
  - TX Queue: Up to four transmission message buffers can be grouped to form a TX Queue with a common access window.
    Upper transmission message buffers are used to form the TXQ.
    TXQ has an access window.
    - TXQ is transmission Message Buffer 0.
- Common FIFO (TX mode): Common FIFO in TX mode is linked to a dedicated channel.
  Channel has a fixed number of one Common FIFO assigned to it. Within the channel, a Common FIFO configured in TX mode, can be freely linked (assigned) between 0 and 3 transmission message buffers (only one FIFO to one transmission message buffer).
  The Common FIFO buffer then replaces the transmission message buffer linked to it.
  Transmission Control and Status registers of these transmission message buffers should not be used.

See Figure 41.28 for information about Common FIFO buffer assignment to related channel.

Note: Common FIFO buffer should not be linked to TX message buffers that are already part of a TX Queue.

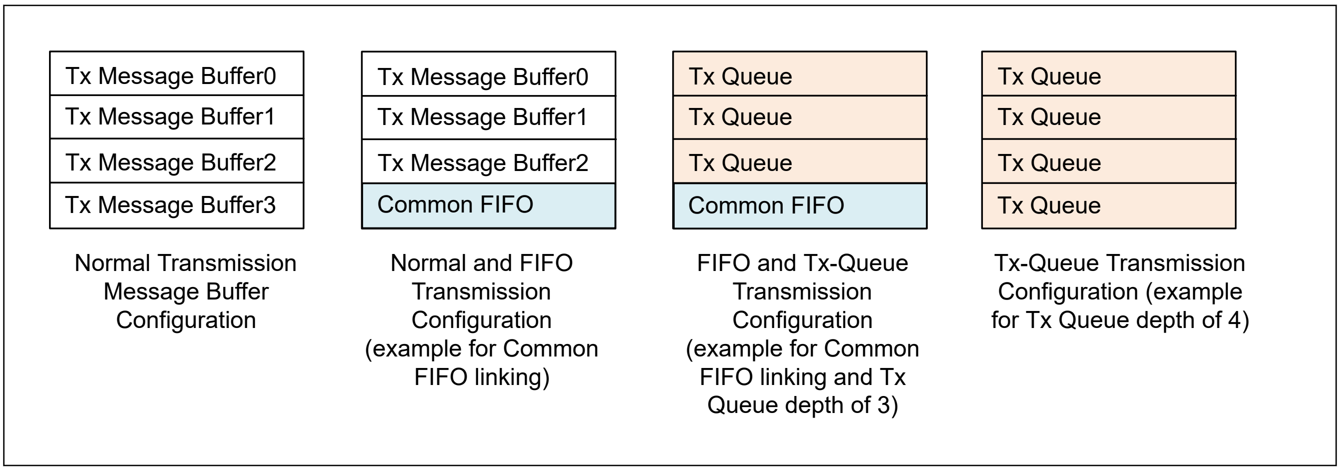


Figure 41.40 Configuration of channel transmission message buffer

41.8.2.1 Transmission Priority

If two or more transmission message buffers of a channel are configured for transmission, then the transmission priority in the CANFD module can be selected from the following two modes:

- CAN ID priority
- Message buffer number priority.

The transmission priority mode is common for all message buffers. It can be configured with the CFDGCFG.TPRI bit in the Global Configuration Register.

For message buffer number priority transmission, the smallest message buffer number with transmission request has the highest priority for transmission. This also includes the TX message buffers linked to the Common FIFO buffer configured in TX mode.

However, message buffer number priority should not be used if TX Queue is enabled.

For CAN ID priority transmission, ID priority complies with the CAN bus arbitration rule (as specified in ISO 11898-1 specification). All TX message buffers can enter the ID priority comparison for message buffers configured for transmission. This also includes the TX message buffers linked to the Common FIFO buffer configured in TX mode and includes the TX Queue message buffers.

If the ID of two or more message buffers is the same, then the smaller message buffer number has higher priority for transmission.

Note: For Common FIFO buffer configured in TX mode, only the message currently being pointed to by the FIFO read pointer can be included in the transmission arbitration.

If the message is being transmitted from the FIFO, then the next pending message within the same FIFO is considered in the transmission arbitration.

In contrast to this, all transmission message buffers of a TX Queue participate in internal transmission arbitration.

Figure 41.41 shows the transmission configuration flow.

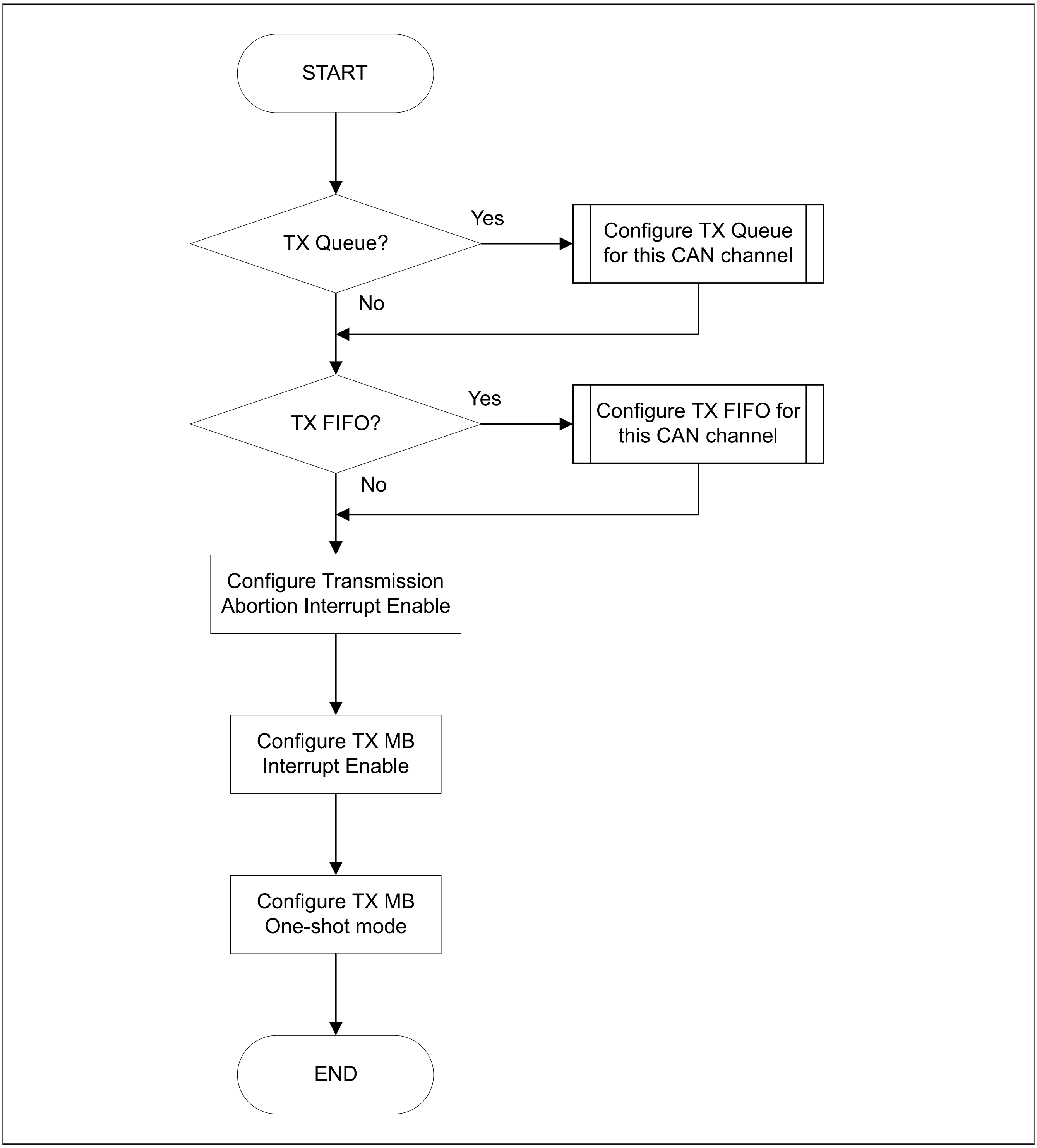


Figure 41.41 Flow for transmission configuration

41.8.2.2 Normal Transmission

Each transmission message buffer has two modes of message transmission:

1. Regular transmission mode
   If the message buffer is placed in regular transmission mode, the data frame or remote frame set in that message buffer can be transmitted.
   Completion of regular transmission can be checked through the related TX Message Buffer Transmission Result flag bits (CFDTMSTSj.TMTRF) in the TX Message Buffer Status Registers. These bits are set to 10b or 11b when the regular transmission is successful.

When arbitration is lost or an error occurs, message transmission is further attempted if no transmission abort request is set for this transmission message buffer.
New internal transmission arbitration for this channel is performed for all message buffers with transmission request.

2. One-shot transmission mode

When the CFDTMCi.TMOM bit of the TX Message Buffer Control Registers is set for a transmission message buffer, the message buffer is placed in One-shot transmission mode and attempts to transmit a message only once.

Completion of One shot transmission can be checked through the related TX Message Buffer Transmission Result Flag bits (CFDTMSTSj.TMTRF) in the TX Message Buffer Status Registers. The CFDTMSTSj.TMTRF bits are set to 10b or 11b when One-shot transmission is successful.

The CFDTMSTSj.TMTRF bits are set to 01b when arbitration is lost or an error occurs during transmission of the related message buffer.

Additional message transmission is not attempted in this case.

The regular transmission request procedure after a configuration is shown in Figure 41.42.

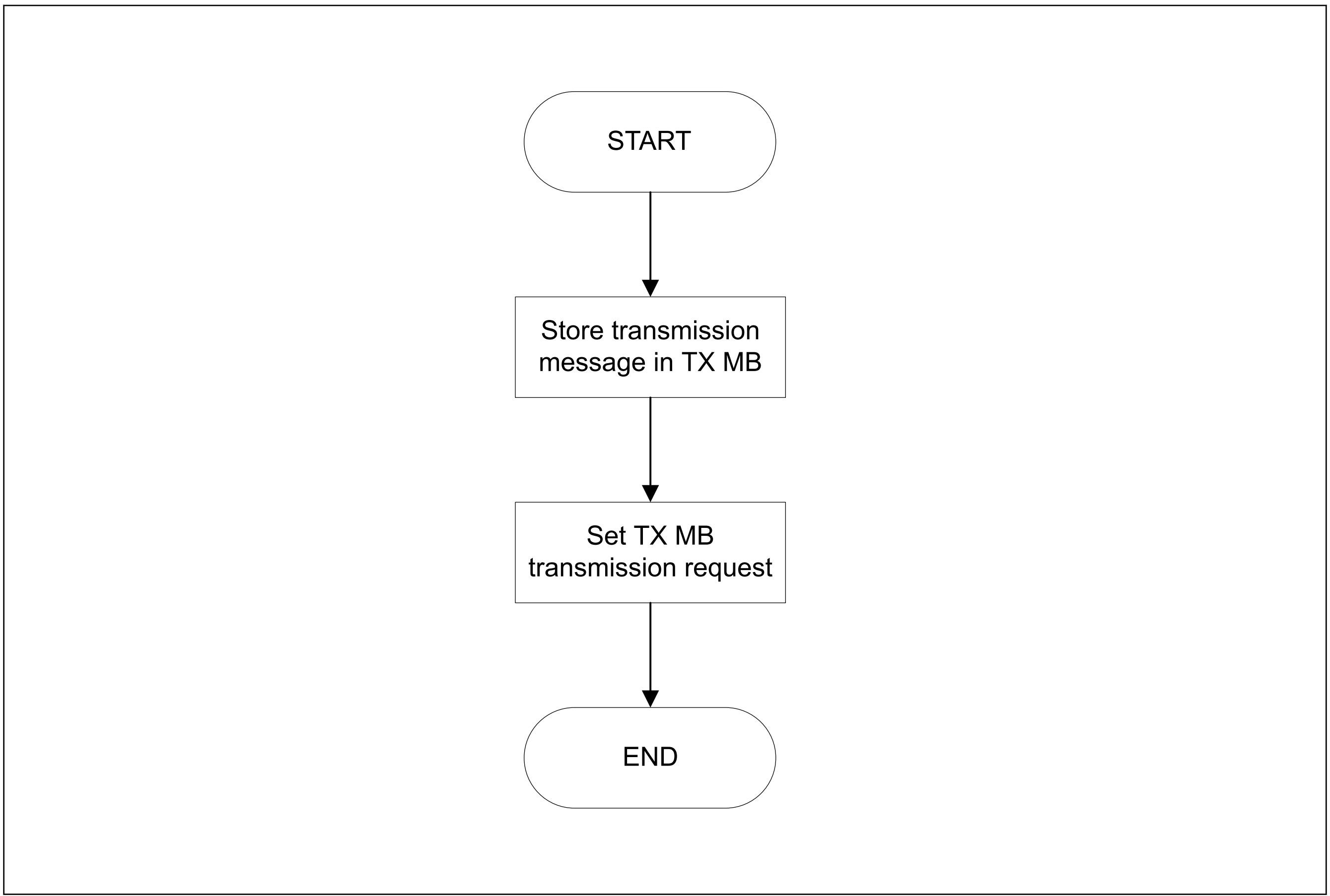


Figure 41.42 Transmission request procedure using normal TX Message Buffer mode

(1) Setting for TX Message Buffer Control Register

Table 41.26 shows configuration of a normal CAN transmission mode.

Table 41.26 Configuration of CAN transmission mode (1 of 2)

Transmission request CFDTMCi.TMTR	Transmission abort request CFDTMCi.TMTAR	One-shot enable CFDTMCi.TMOM	Communication activity
0	0	0	Message buffer disabled
0	0	1	Message buffer disabled

Table 41.26 Configuration of CAN transmission mode (2 of 2)

Transmission request CFDTMCi.TMTR	Transmission abort request CFDTMCi.TMTAR	One-shot enable CFDTMCi.TMOM	Communication activity
1	0	0	Configured as a transmission message buffer for a data frame or a remote frame
1	0	1	Configured as a one-shot transmission message buffer for a data frame or a remote frame
1	1	0	Transmission abort requested
1	1	1	One-shot transmission abort requested

The configuration bits can be configured in the TX Message Buffer Control Registers.

Figure 41.43 shows timings for successful transmission for two message buffers.

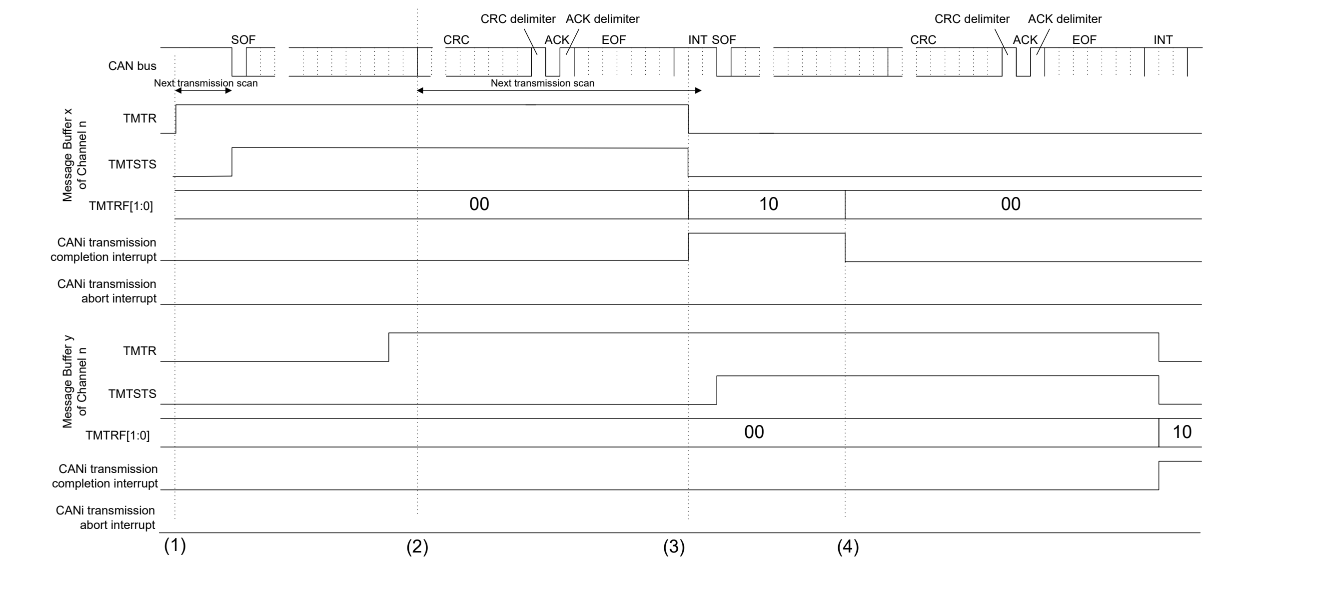


Figure 41.43 Timing of request and flag bits for successful transmission

1. If the CFDTMCi.TMTR bit in the TX Message Buffer Control Registers is set in the bus idle state, the message buffer scanning procedure determines the highest priority message buffer for transmission.
   When the transmission message buffer is determined, the CFDTMSTSj.TMTSTS bit in the related TX Message Buffer Status Registers is set (transmitting/transmitter), and CAN channel starts the transmission *1.
2. At the first bit of CRC, the transmission scanning procedure starts for the next possible transmission when pending transmission requests exist.
3. If the message has been successfully transmitted, the CFDTMSTSj.TMTRF[1:0] bits in the corresponding TX Message Buffer Status Registers are set to 10b and CFDTMSTSj.TMTSTS and the CFDTMCi.TMTR bits are cleared.
   When the TMIE bit in the TX Message Buffer Interrupt Enable Configuration Registers is set (interrupt enabled), the CAN successful transmission interrupt request is generated.
   To clear the related interrupt line, clear the CFDTMSTSj.TMTRF flag bits.
4. Before starting the next transmission, clear the CFDTMSTSj.TMTRF bits. Load the next message in the transmission message buffer and set the CFDTMCi.TMTR bit again.
   CFDTMCi.TMTR bit cannot be set again before CFDTMSTSj.TMTRF[1:0] bits are cleared.

Note 1. If arbitration is lost after the CAN channel starts the transmission, the CFDTMSTSj.TMTSTS bit is cleared.

The transmission scanning procedure is performed again to search for the highest priority transmission message buffer from the beginning of the first CRC bit.

If an error occurs either during transmission or following the loss of arbitration, then during the error frame, the transmission scanning procedure is performed again to search for the highest priority transmission message buffer.

Note: The setting point of CFDTMSTSj.TMTSTS is not always fixed at the start of the SOF. It may be delayed up to the start of the standard ID due to the synchronization logic implemented for the PLL bypass.

Figure 41.44 shows timings for transmission abort for two message buffers.

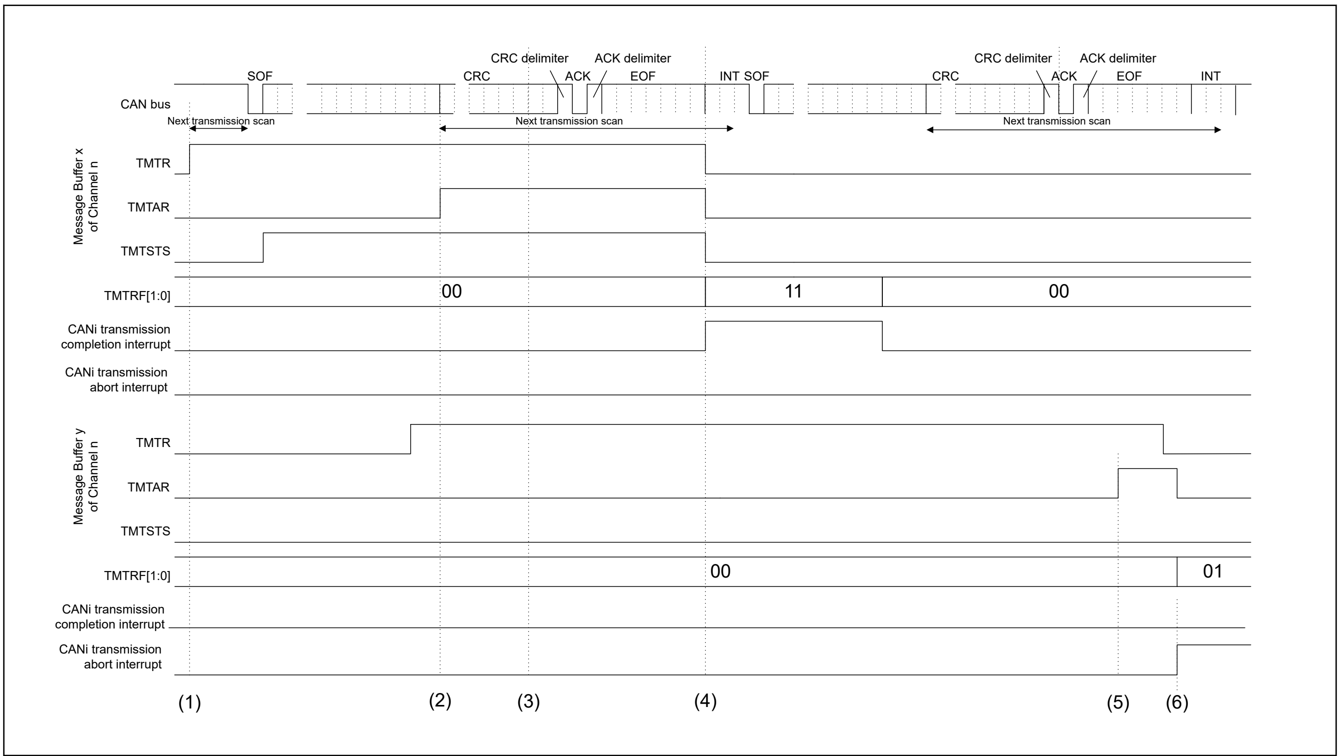


Figure 41.44 Timing of request and flag bits for transmission abort

1. If the CFDTMCi.TMTR bit in the TX Message Buffer Control Registers is set in the bus idle state, the message buffer scanning procedure determines the highest priority message buffer for transmission.
   When the transmission message buffer is determined, the CFDTMSTSj.TMTSTS bit in the TX Message Buffer Status Registers is set (transmitting/transmitter), and CAN channel starts the transmission*1.
2. If the CFDTMCi.TMTAR bit is set when the related message buffer is already selected for transmission or currently transmitting, the message is not aborted, if no error occurs or arbitration is lost.
3. At the first CRC bit, the transmission scanning procedure starts for the next transmission. In this example, timing chart message buffer y is not selected as the next transmission message buffer.
4. If the message has been successfully transmitted, the CFDTMSTSj.TMTRF[1:0] bits in the corresponding TX Message Buffer Status Registers are set to 11b and the CFDTMSTSj.TMTSTS and CFDTMCi.TMTR bits are cleared.
   When the TMIE bit in the TX Message Buffer Interrupt Enable Configuration Registers is set (interrupt enabled), the CAN successful transmission interrupt request is generated.
   To clear the related interrupt line, clear the CFDTMSTSj.TMTRF[1:0] bits.
5. Another CAN node is transmitting on the CAN bus (CFDTMSTSj.TMTSTS is not set). If the CFDTMCi.TMTAR bit is set when the related channel is under transmission scan, the transmission request cannot be cleared.
6. After internal processing time, the transmission is aborted and the CFDTMSTSj.TMTRF[1:0] bits are set to 01b.
   If the message buffer is not transmitting or selected as the next transmission message buffer or under transmit scan, then the abort is immediately accepted and the corresponding CFDTMSTSj.TMTRF[1:0] bits in the TX Message Buffer Status Registers are set to 01b.
   In addition, CFDTMCi.TMTR, and CFDTMCi.TMTAR bits are cleared automatically.
   When the transmission abort interrupt enable TAIE bit of the related Channel Control Register is set then an interrupt is generated for successful transmission abort.
   To clear the related interrupt line the CFDTMSTSj.TMTRF[1:0] bits have to be cleared.

Note 1. If arbitration is lost after the CAN channel starts the transmission, the CFDTMSTSj.TMTSTS bit is cleared.

The transmission scanning procedure is performed again to search for the highest priority transmission message buffer from the beginning of the first CRC bit.

If an error occurs, either during transmission, or following the loss of arbitration, then during the error frame, the transmission scanning procedure is performed again to search for the highest priority transmission message buffer.

41.8.2.3 TX FIFO Transmission

One common FIFO buffer is assigned to CANFD module. The FIFO buffer can be linked to any normal TX message buffer position for this channel with the CFDCFCC.CFTML bits in the Common FIFO Configuration/Control Register if configured in TX mode.

When the transmission scan starts and the FIFO buffer corresponding to this TX message buffer is enabled, the relevant message in the FIFO buffer participates in the transmission scan.

Configuration of a TX message buffer linked to a FIFO buffer configured in TX mode should not be done.

(1) TX FIFO Operation

CAN messages can be written into the TX FIFO by writing to the corresponding FIFO Access registers.

When the value 0xFF is written into the corresponding FIFO Pointer Control Register, the message count of the related FIFO is incremented by 1.

Only write to the FIFO Pointer Control register after writing the complete message to the corresponding FIFO Access registers. If the message count matches the FIFO depth, the FIFO Full flag is set.

The oldest message in the TX FIFO is included in the scan for transmission by the corresponding CANFD module channel logic.

When a message is successfully transmitted from the TX FIFO, the message count value is decremented by 1. When all the messages from the FIFO are transmitted, the FIFO Empty flag is set.

The interrupt generation conditions for the TX FIFO buffer can be configured by configuring the CFDCFCC.CFIM bit in the corresponding Common FIFO Configuration/Control Register.

If CFDCFCC.CFIM bit is 0, then interrupt is generated when the last message is successfully transmitted from the TX FIFO buffer.

If CFDCFCC.CFIM bit is 1, then interrupt is generated for every successfully transmitted message from the TX FIFO buffer.

The Common FIFO can set interrupt when CAN frame transmission is complete.

The Common FIFO buffer configured in TX Mode can be disabled by clearing the CFDCFCC.CFE bit in the Common FIFO Configuration/Control Register. If this bit is cleared to 0, the FIFO Empty flag is set as follows:

- Immediately if the message from the TX FIFO is neither scheduled for the next transmission nor in transmission
- Following the transmission completion, the detection of an error on the CAN bus, loss of arbitration or transition to Channel or Global Halt mode if the transmission from the TX FIFO is already scheduled for transmission or already in transmission.

Note: The Common FIFO buffer is considered as disabled after clearing the CFDCFCC.CFE bit only when the Empty flag is set for the corresponding Common FIFO buffer.

Other possible messages pending from the TX FIFO are lost and their transmission must be requested again. Before CFDCFCC.CFE is set again, ensure that CFDCFSTS.CFEMP bit is set and that there are no pending aborts from the TX FIFO.

When the CFDCFCC.CFE bit is cleared, the message read and write pointers of the FIFO are cleared and are no longer active. Therefore, all messages in the FIFO buffers are lost and no further message can be stored into the FIFO.

The FIFO transmission request procedure after configuration is shown in Figure 41.45.

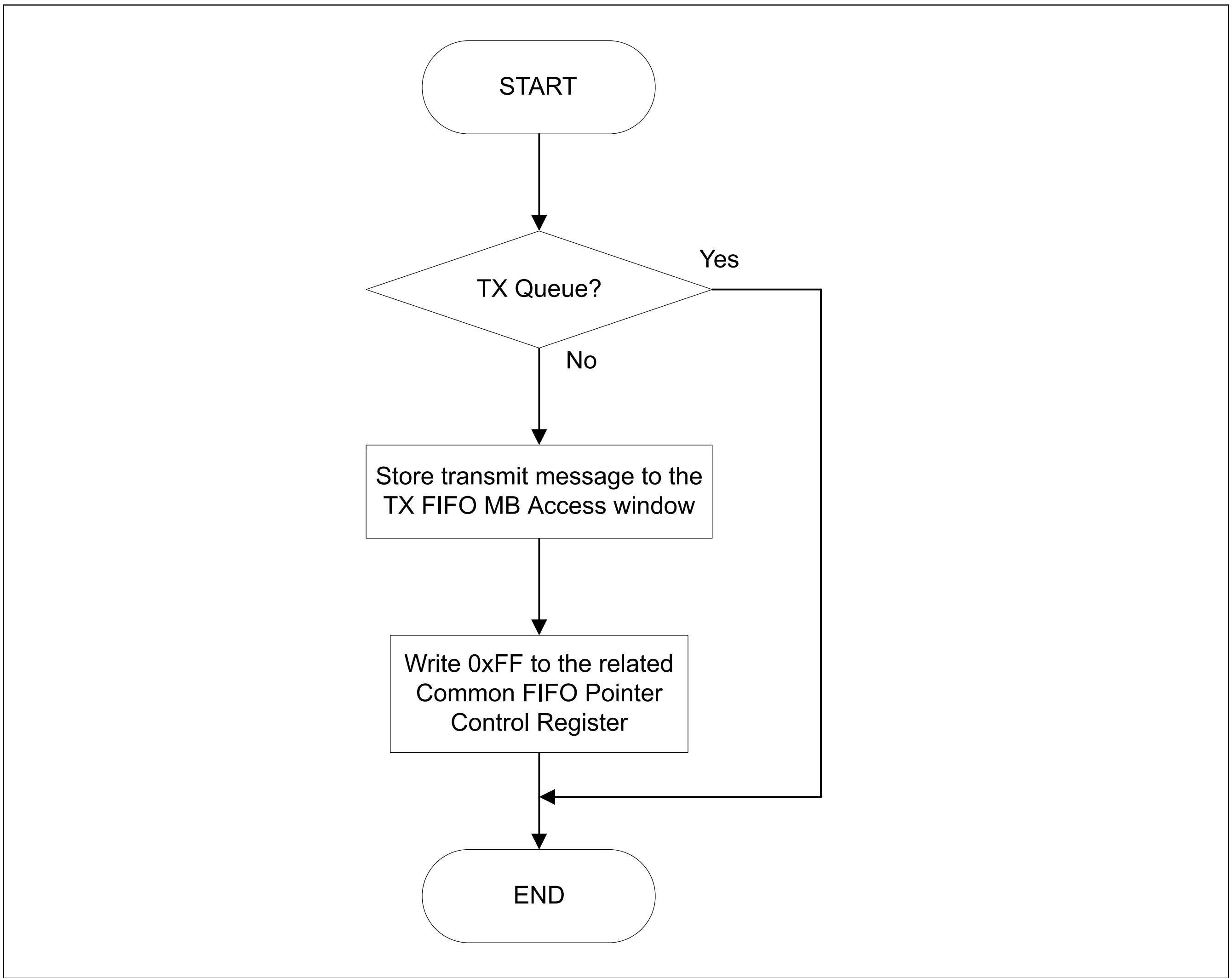


Figure 41.45 Request procedure for TX FIFO transmission

(2) Interval Timer for FIFO Transmission

For each Common FIFO in TX mode, it is possible to specify a delay between two consecutive messages that are configured for transmission from the same FIFO buffer. This delay is called interval time. This interval time starts after the first message has been successfully transmitted from the FIFO buffer after the CFDCFCC.CFE bit is set.

When the Common FIFO in TX mode is enabled, the first message is transmitted without considering this interval time.

The interval timer stops counting when:

- FIFO is disabled by clearing the CFDCFCC.CFE bit.
- CAN channel is in CH_RESET mode.

The interval time is specified by the CFDCFCC.CFITT value from 0 to 255 timer units in the Common FIFO Configuration/ Control Register.

The timer unit can be defined based on two different source clocks for the interval timer. To disable the interval timer for FIFO transmission, select a value of 0.

The timer source can be selected with the configuration bit CFITSS in the Common FIFO Configuration/Control Register.

If CAN channel bit time clock is configured as the clock source, and the CAN channel enters CH_HALT, CH_RESET, or CH_SLEEP mode, the interval timer is stopped for that channel.

If peripheral clock is selected as the interval timer clock source, the interval timer is stopped only when the CAN channel is in CH_RESET or CH_SLEEP mode.

The reference clock can be used to configure the interval time in fixed time units. It is based on the peripheral clock. The reference clock prescaler value CFDGCFG.ITRCP in the Global Configuration Register defines the relation between the peripheral clock frequency/period and the reference clock period.

See Table 41.27 for CFDGCFG.ITRCP configuration values to achieve different reference clock periods based on the peripheral clock frequency and period.

Table 41.27 Configuration example for the reference clock of the FIFO interval timer

Reference clock/Peripheral clock	1 µs	100 µs	500 µs
16 MHz/62.5 ns	16	1600	8000
20 MHz/50 ns	20	2000	10000
32 MHz/31.25 ns	32	3200	16000
50 MHz/20 ns	50	5000	25000

The reference clock resolution can be specified by the interval timer reference clock resolution value CFDCFCC.CFITR in the Common FIFO Configuration/Control Register.

The interval time is based on the reference clock period multiplied by the configured value (x1 or x10). The reference clock based interval timer can be used to satisfy the requirements of the ISO 15765-2 Separation Time. The whole range for the separation time from 100 µs to 127 ms can be covered.

The specified interval time starts after successful transmission event (after EOF7 state of the CAN protocol).

When the interval time has elapsed, the next transmission request is raised by the related TX FIFO. Therefore, the interval time defines the minimum time between two messages transmitted from one FIFO.

The next message is sent at earliest after this interval time. Figure 41.46 shows an example timing of the internal processing.

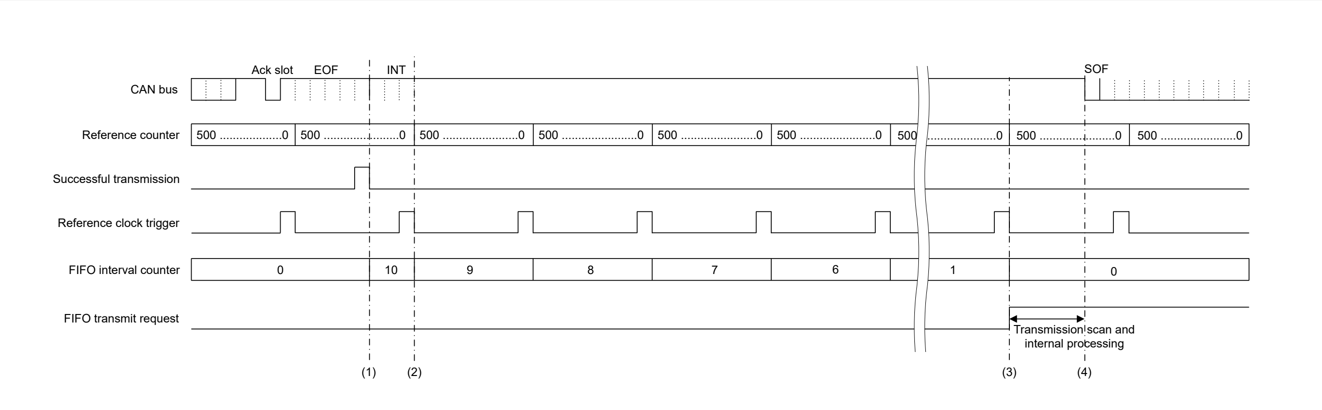


Figure 41.46 Example for interval processing time

The configuration for the timing in Figure 41.46 is as follows:

- Peripheral clock frequency = 50 MHz
- Interval timer reference clock (CFDGCFG.ITRCP) = 500 times
- Reference clock from the settings in Figure 41.46 = 10 µs
- Common FIFO interval timer source selection (CFDCFCC.CFITSS) = 0
- Common FIFO interval timer resolution (CFDCFCC.CFITR) = 0
- Common FIFO interval transmission time (CFDCFCC.CFITT) = 10 times
- Theoretical message separation interval = 100 µs

1. Internal FIFO interval timer is restarted with the occurrence of successful transmission result. This restart is not synchronized to the reference clock trigger. Therefore, the first interval is counting less or equal to 1 reference clock interval.
2. With the next reference clock trigger the FIFO interval timer is decremented.

3. When the FIFO interval timer reached the value 0, the FIFO transmit request is set.
4. When the FIFO is selected for transmission, the transmission starts. Due to internal processing, this usually takes less than 3 CAN bit time, between the internal FIFO transmit request set in step 3. and the actual transmission.

In the worst case when multiple events such as a reception scan, an internal message routing, a transmit scan on all channels occur, it can take up to 120 peripheral clock cycles.

As shown in Figure 41.46, it is not guaranteed that the minimum interval is always equal to the configured value. If a minimum time must never be breached, configure CFDCFCC.CFITT to the required minimum value plus 1.

If additional TX message buffers or TX FIFO are configured for transmission of the same channel, the real delay between two messages transmitted from a TX FIFO can be much longer than specified by the interval time. This is due to higher priority message transmission from these TX message buffers or TX FIFO.

Figure 41.47 shows a block diagram of the FIFO interval time generation circuit.

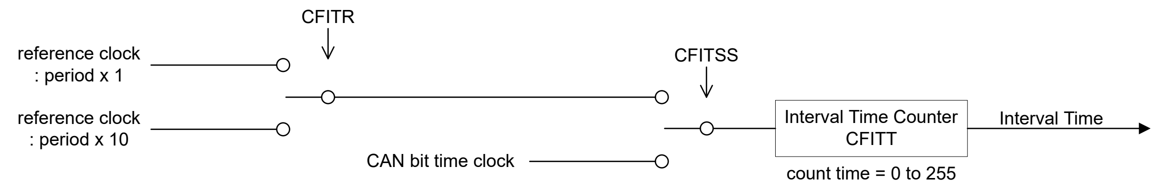


Figure 41.47 Block diagram of FIFO interval timer

41.8.2.4 TX Queue

Each enabled TX Queue for a specific channel consists of 3 to 4 TX message buffers, which are accessed through one access window.

- The first TX Queue can be configured with a depth of three up to four buffers and uses TX Message Buffer No. 0 as access window (referred to as TXQ)

All the TXQ messages enter the priority comparison for the transmission, which should be only ID Priority (CFDGCFG.TPRI = 0).

The registers for TXQ are:

- CFDTXQCC
- CFDTXQSTS
- CFDTXQPCTR

See related access registers TX Message Buffer ID Registers (TMID[m]), TX Message Buffer Pointer Registers (TMPTR[m]), TX Message Buffer Data Field 0 Registers, and TX Message Buffer Data Field 1 Registers (TMDF[0:1][m]) when access window TXQ0 is used.

The depth of each TXQ buffer can be configured by writing to the CFDTXQCC.TXQDC[1:0] bits of the TX Queue Configuration/Control Register. TXQ can be set from TXMB0 to TXMB3 as a queue buffer at the maximum.

The 4 available options for the depth configuration of TXQ buffer are:

- 0x00: TX Queue disabled
- 0x01: reserved
- 0x10: 3 Messages
- 0x11: 4 Messages

Do not access all the TX message buffers forming the TX Queue directly (except TX Message Buffer No. 0, which act as TX Queue access window).

When a system writes in TXQ, it writes in send data, after checking the state of TXQ.

Do not access or configure the related TX Message Buffer Control Registers.

The messages stored to the TX Queue access window are internally stored to a free buffer of the TX Queue.

When the buffer is full, no further access should be done to the queue, until it is no longer full. If access is a software write when the buffer of TXQ is full, send data is overwritten.

The TX Queue can be disabled by clearing the TXQE bit in the TX Queue Configuration/Control Register. If this bit is cleared, the TX Queue Empty flag is set as follows:

- Immediately if the message from the TX Queue is neither scheduled for the next transmission nor in transmission
- Following the transmission completion, the detection of an error on the CAN bus, loss of arbitration or transition to Channel or Global Halt mode if the transmission from the TX Queue is already scheduled for transmission or already in transmission.

Note: The TX Queue is disabled only when the Empty flag is set after clearing the TXQE bit for the corresponding TX Queue.

Other possible messages pending from the TX Queue are lost and their transmission must be requested again.

Before TXQE is set again, ensure that the CFDTXQSTS.TXQEMP bit is set and that there is no pending abort from the TX Queue.

When the TXQE bit is cleared, all messages in the TX Queue buffers are lost and no further message should be stored in the TX Queue.

When a message has been stored to the TX Queue, write 0xFF in the TX Queue Pointer Control Register. This sets the transmit request automatically and changes the internal message buffer pointer to the next free message buffer location of the TX Queue.

Note: If two messages with the same ID are stored in the TX Queue, the order of transmission of these messages can be different from the order in which they were stored in the TX Queue.

To avoid this condition, it is important to confirm that the previous message with the same ID was successfully transmitted before a new message with the same ID is stored in the TX Queue.

For the TX Queue, a dedicated interrupt can be enabled by setting the TXQIE bit of the TX Queue Configuration/Control Register.

The interrupt mode can be configured with the CFDTXQCC.TXQIM bit of the same register either to generate an interrupt for every transmitted message or for the last transmitted message.

The TX Queue transmission request procedure after configuration is shown in Figure 41.48.

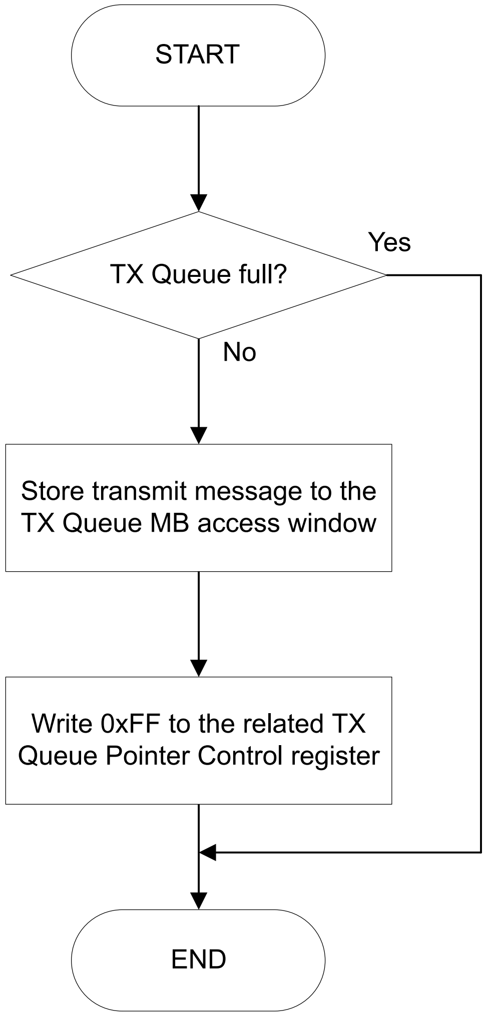


Figure 41.48 TX Queue transmission request

41.8.2.5 TX History List

The TX History List function records the information of the successfully transmitted message in the TX History List Buffers. Two TX History List buffers are provided and THL buffer can store up to 8 TX History List entries.

The CFDTHLCC.THLDTE bit of the TX History List Configuration/Control Register can be used to configure if only message information from TX FIFO or TX Queue is stored, or if all transmit message information from TX Queue, TX FIFO, or normal TX message buffers is stored in the TX History List.

Each transmit message can be individually configured for acceptance to the TX History List with the CFDCFID.THLEN bit in the Message Buffer Pointer Register.

The message information is stored to the TX History List Buffer of a CAN channel after the message is successfully transmitted.

Storing to the list is not synchronized with the status of CFDTMSTSj.TMTRF[1:0] bits in the TX Message Buffer Status Register.

Due to internal processing, the storage to the list can happen with a delay after the successful transmission indication.

Storing the TX History List data can be recognized by the condition that the THLIF is set to 1 when the THLIE bit is configured to 1 or when the TX History List counter CFDTHLSTS.THLMC[5:0] is increased.

In the worst case when multi events like reception scan, internal message routing on happen.

- Maximum delay time from setting the CFDTMSTSj.TMTRF to store the TX History List data is 70 peripheral bus clock cycles.

The History list records the following information of a transmitted message:

- Buffer type:
  - 001: TX Message Buffer
  - 010: TX FIFO
  - 100: TX Queue
- Buffer number:
  TX message buffer, TX Queue message buffer or TX message buffer link for the Common FIFO buffer from which transmission occurred. The number depends on the buffer type. See Table 41.28.
- Transmission ID:
  Transmission pointer stored in the transmission message
- Transmit timestamp:
  Message timestamp captured at capture point as configured by CFDGFDCFG.TSCCFG.
- Transmission information label:
  Transmission information label stored in the transmission message.

Table 41.28 TX History List Buffer number entry

Buffer Number	BT[2:0] Buffer Type		
	001b TX Message Buffer	101b TX FIFO	100b TX Queue
00b	Message Buffer 0	Number shown corresponds to the common FIFO. TX Message Buffer Link CFTML of the related Common FIFO configuration	Number shown corresponds to the Message Buffer belonging to the TX Queue which the frame was transmitted
01b	Message Buffer 1		
10b	Message Buffer 2		
11b	Message Buffer 3		

The Transmission ID entry is used to identify which message of a TX FIFO or TX Queue has been successfully transmitted because the TX FIFO or TX Queue number alone is not sufficient.

Therefore, a unique number can be attached to each transmission message stored in a TX FIFO or TX Queue. This unique identification number should be written to the CFDCFFDCSTS.CFPTR[15:0] part of the Common FIFO Access Pointer Register for a TX FIFO or to the CFDTMFDCTRb.TMPTR[15:0] part of the TX Message Buffer Pointer Register of the TX Queue access window message buffer.

When the message is successfully transmitted, this identification number is stored together with the other message related information to the TX History List and can be read using the Transmission ID (TID) of the TX History List Access Register.

Also, for normal TX message buffers, the CFDTMFDCTRb.TMPTR[15:0] part of the TX Message Buffer Pointer Register is stored in the Transmission History List and the information label is the same.

Figure 41.49 shows a transmission preparation flow when TX History List is used.

Read access to the TX History List Access Register is done for every single entry.

After reading one entry, 0xFF must be written to the corresponding TX History List Pointer Control Register to be able to access the next entry until TX History List is empty.

Figure 41.50 shows an example flow for processing the TX History List information.

The TX History Lists have dedicated interrupts, which can be configured with the CFDTHLCC.THLIM bit of the corresponding TX History List Configuration/Control Register and enabled with the CFDTHLCC.THLIE bit of the same registers, either to generate an interrupt when the History List reached a filling level of 75% or for every new TX History List entry.

An entry lost indication is flagged by the CFDTHLSTS.THLELT bit in the TX History List Status Register. The status of this bit is also shown by the THLES bit in the Global Error Flag Register.

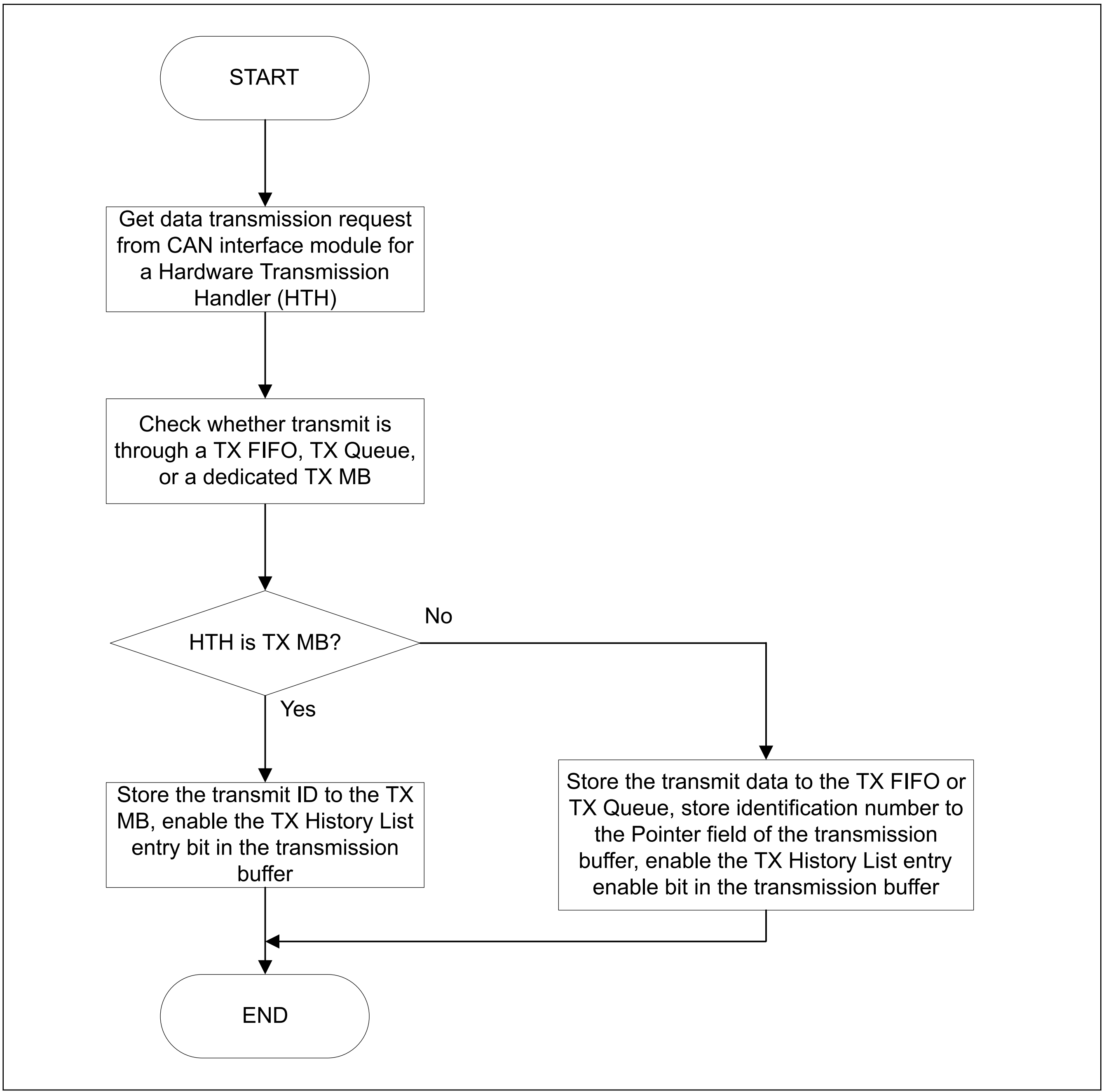


Figure 41.49 TX History List preparation flow

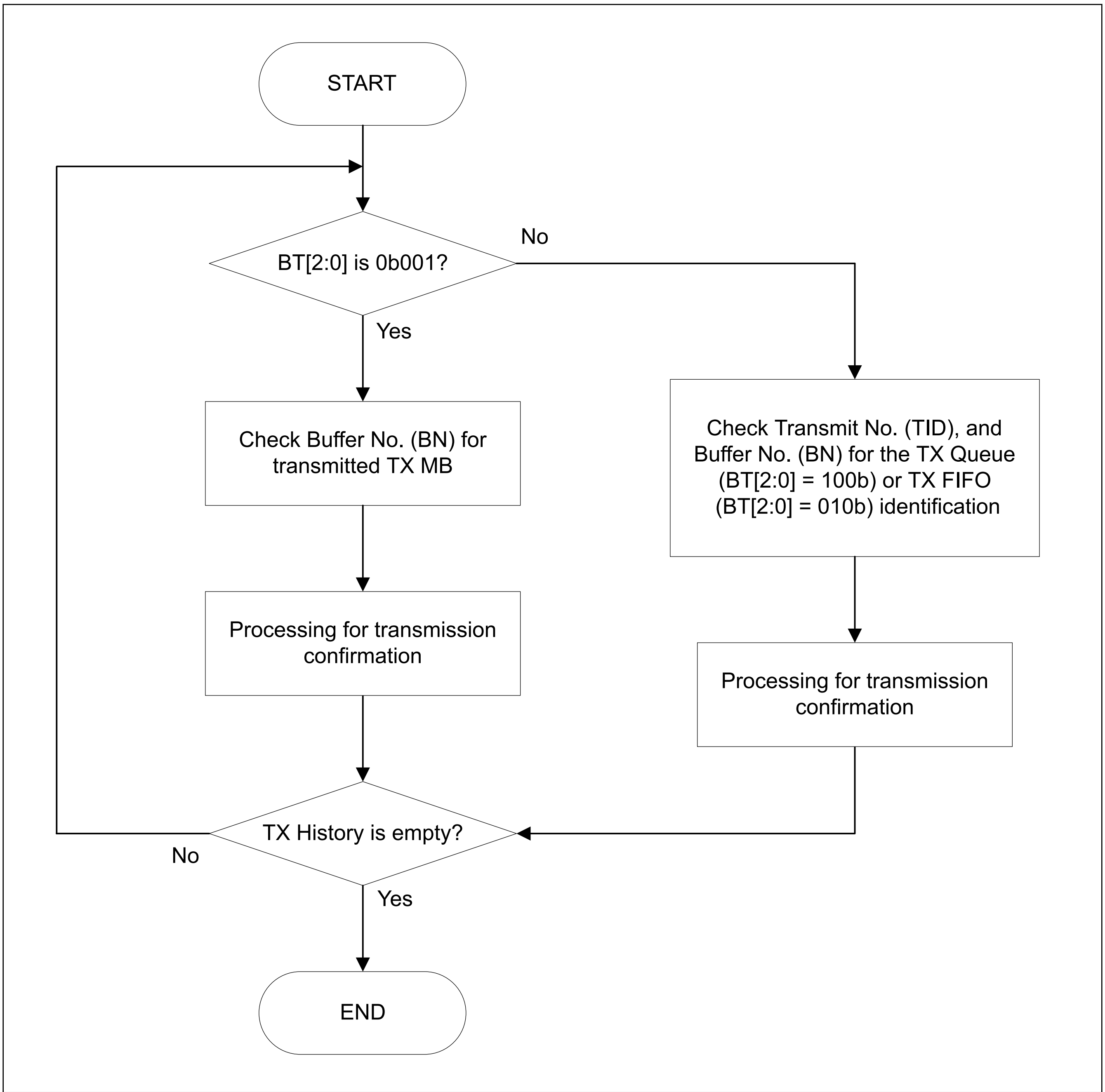


Figure 41.50 TX History List processing flow

41.8.2.6 TX Data Padding

This chapter is not valid for classical CAN.

If the data length code (DLC) of the transmitting message has a higher number of data bytes than the buffer size, the data bytes beyond the restricted range are replaced by bytes with the value of 0xCC.

This can happen for Common FIFO configured as (TX mode) when the transmit message DLC is higher than the CFDCFCC.CFPLS.

This can also happen in FD only mode, if a Classical frame is configured with a DLC bigger than 8.

41.9 Test Mode

The CANFD module can be configured into test modes to allow testing of certain features. These features are provided only for special purposes and care must be taken when configuring the CANFD module in test modes.

Note: All test modes are mutually exclusive unless it is explicitly stated that some functions can be enabled across other test modes.

Do not enable any combination of the various test modes specified in this section.

The test modes can be broadly split into 2 groups:

- Channel specific test modes
- Global test modes.

41.9.1 Channel Specific Test Modes

CAN channel can be configured into the following test modes:

- Basic test mode
- Listen-only mode
- Self-test mode 0 (External loop back mode)
- Self-test mode 1 (Internal loop back mode)
- Restricted operation mode.

41.9.1.1 Basic Test Mode

The basic test mode should be used when there is requirement for a particular test setting to be enabled other than when in Listen-only and Self-test modes.

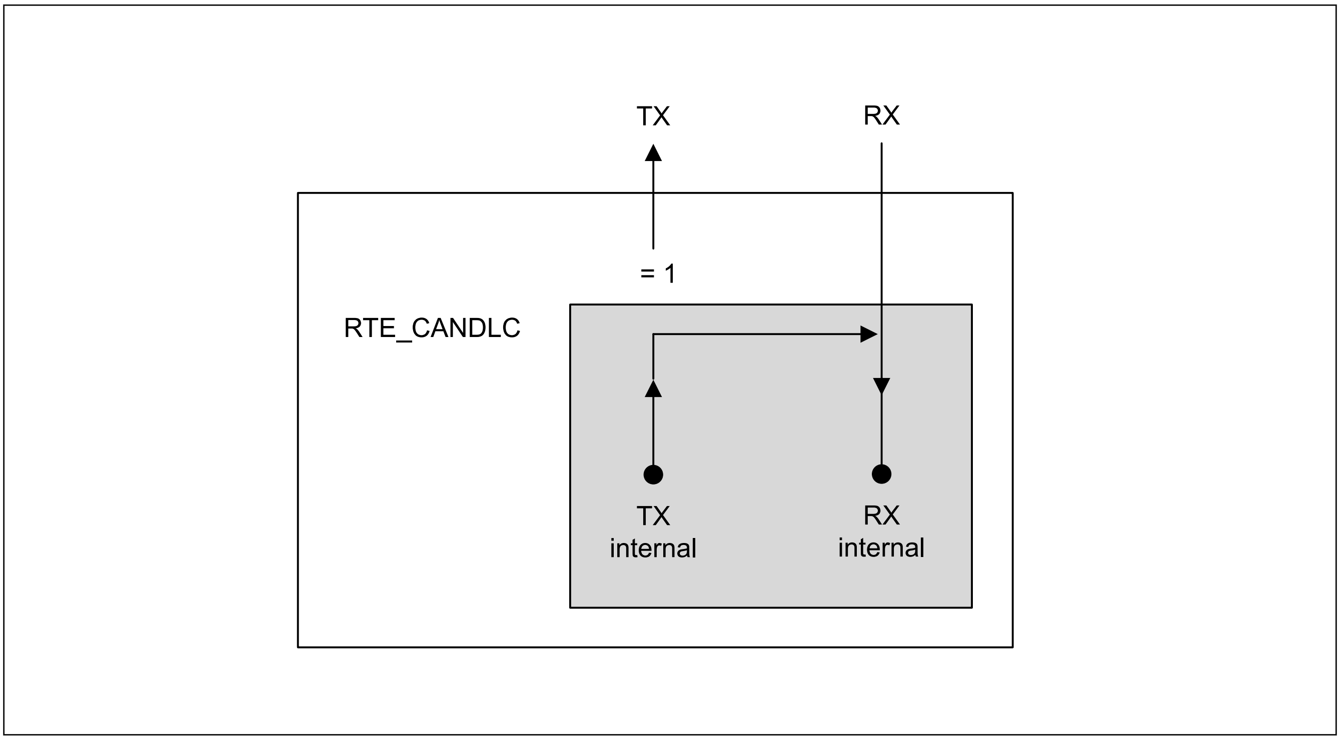
41.9.1.2 Listen-only Mode

The ISO 11898-1 recommends an optional bus-monitoring mode. In this mode, the CAN channel is able to receive valid data frames and valid remote frames. However, it sends only recessive bits on the CAN bus and is not allowed to transmit.

If the CAN engine is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is routed internally so that the CAN engine monitors this as dominant. The external TX pin remains in recessive state.

This mode can be used for baud rate detection. In this mode, an error interrupt is generated if a bus error occurs and the interrupt is enabled.

In this mode, it is not permitted to request transmission from any normal TX message buffer or TX FIFO.

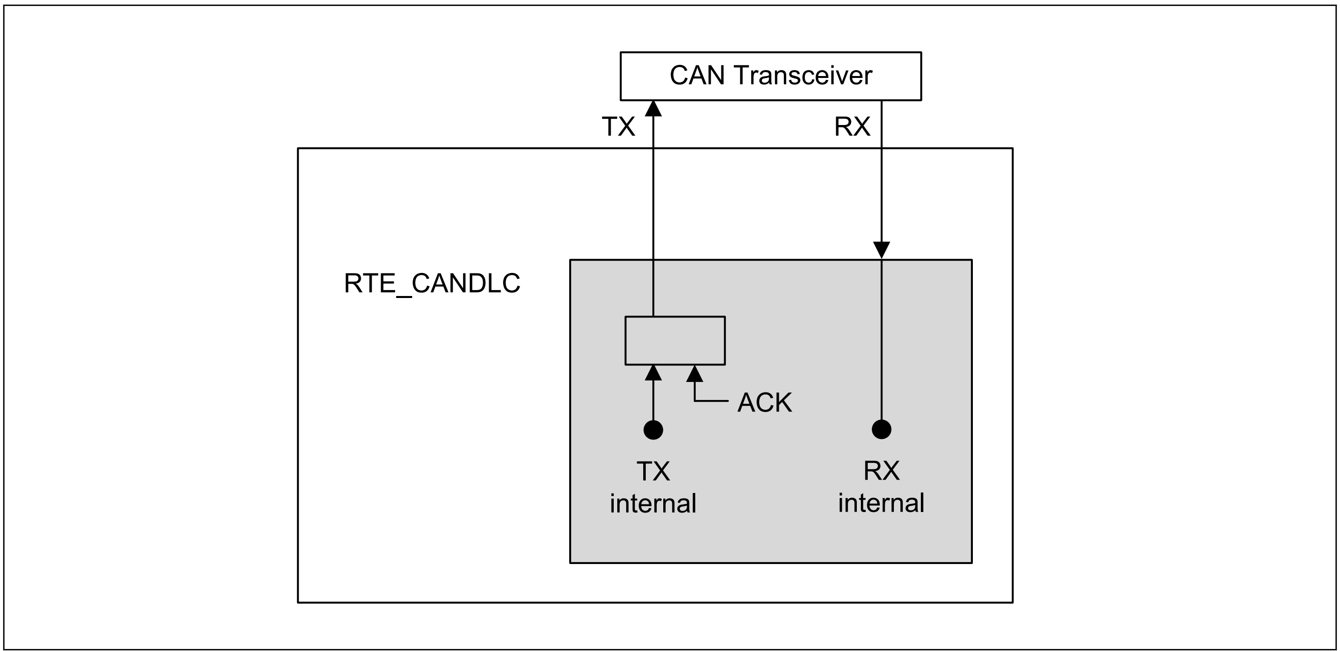


41.9.1.3 Self-test Mode 0 (External loopback mode)

In Self-test mode 0, the CAN engine treats its own transmitted messages as received messages through the CAN transceiver and stores them into its receive message buffers.

To be independent of external stimulation, the engine generates its own Acknowledge bit.

This test can be used for CAN transceiver tests and the RX/TX pins should be connected to the transceiver.

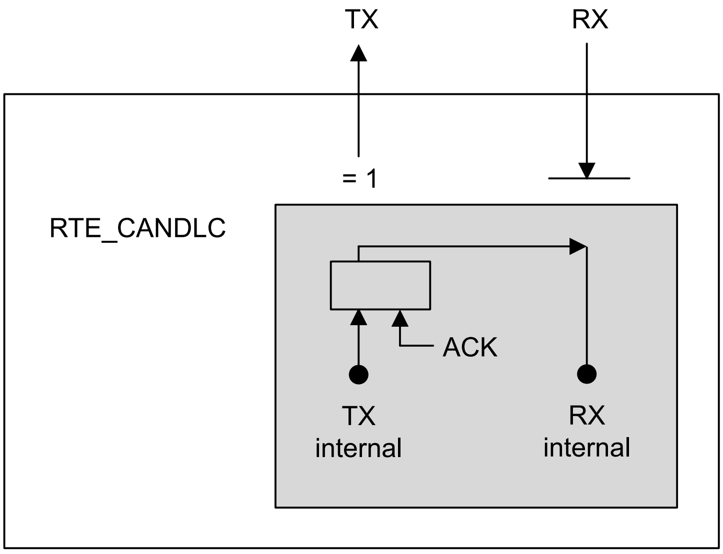


41.9.1.4 Self-test Mode 1 (Internal loopback mode)

In Self-test mode 1, the CAN engine treats its own transmitted messages as received messages and stores them into the receive buffer. This mode is provided for self-test functions. To be independent of external stimulation, the CAN engine generates its own Acknowledge bit. In this mode the CAN engine performs an internal feedback from TX internal to RX internal. The actual value of the external RX input is disregarded by the CAN engine.

The external TX pin outputs only recessive bits. The RX/TX pins do not need to be connected to the CAN bus or any external device.

Note: The channel pins are also disconnected from the internal CAN bus communication line.



41.9.1.5 Restricted Operation Mode

This chapter is not valid for classical CAN.

In Restricted operation mode, the CAN node is able to receive valid data and remote frames generating the Acknowledge bit.

Active error or overload frames cannot be transmitted, instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication after an error or overload condition occurs.

Additionally, the Receive and Transmit Error Counter (REC and TEC) are frozen independently of the occurrence of errors. The mode is specified in ISO 11898-1 and the setting of transmit request is permitted.

41.9.2 Global Test Modes

The CANFD module can be configured into the following test modes:

- RAM test mode
- Bit Flip Test

The test modes in the following table are protected by a special software procedure to enable the mode. This software procedure enables write access to the test mode by a specific unlock key as shown in the table.

Test mode	Unlock key 1	Unlock key 2
RAM test mode	0x7575	0x8A8A

If the software sequence of the two consecutive unlock key write accesses (half-word or word access) is interrupted by any other write access to the register or if incorrect data is written to the Global Unlock Key Register, the corresponding test mode cannot be set and the sequence must be restarted.

After the two unlock key write accesses, the next write access should be to set the corresponding test mode enable bit. If this is not followed, the unlock mechanism reset and the test mode enable bit cannot be set and the unlock sequence must be restarted.

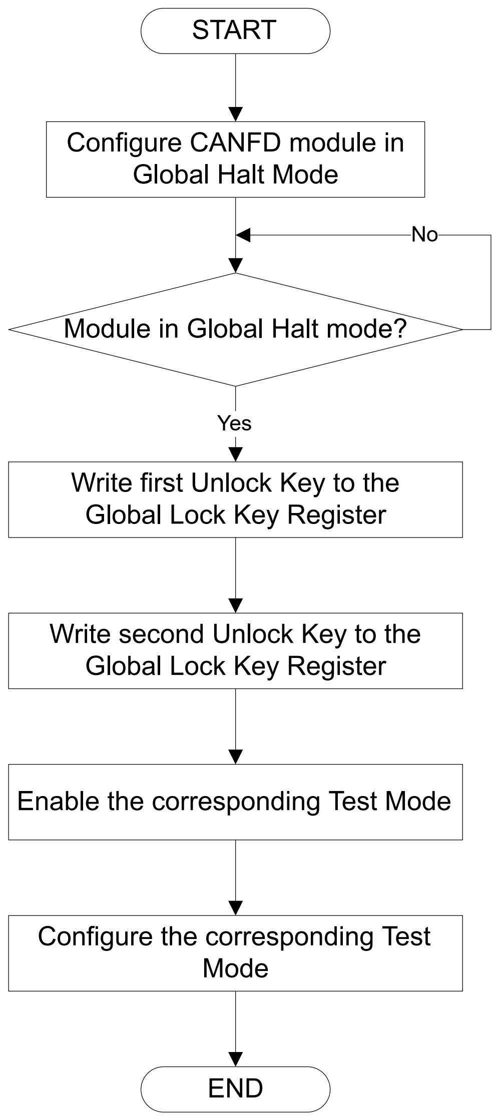


Figure 41.51 Unlock software protection routine

41.9.2.1 RAM Test Mode

The CANFD module can be configured in RAM test mode by setting the CFDGTSTCTR.RTME bit in the Global Test Control Register when the corresponding lock key is previously written. This is a special test mode, in which, the complete RAM area can be accessed.

Note: The actual RAM size is bigger than the RAM area initialized after a hardware reset. Therefore, ECC error flag (of the ECC macro) may be set if CPU reads data from this uninitialized RAM area while CANFD module is in RAM test mode.

In this mode, the RAM area is split into number of pages (pn) of 256 bytes, each which can be accessed with the CFDRPGACCk register.

The page should be selected for read/write access by writing to the CFDGTSTCFG.RTMPS[3:0] bits in the Global Test Control Register. Data can then be read from or written in to the RAM Test Page Access Registers.

Figure 41.52 shows the structure of the pages in the RAM when performing a RAM test mode.

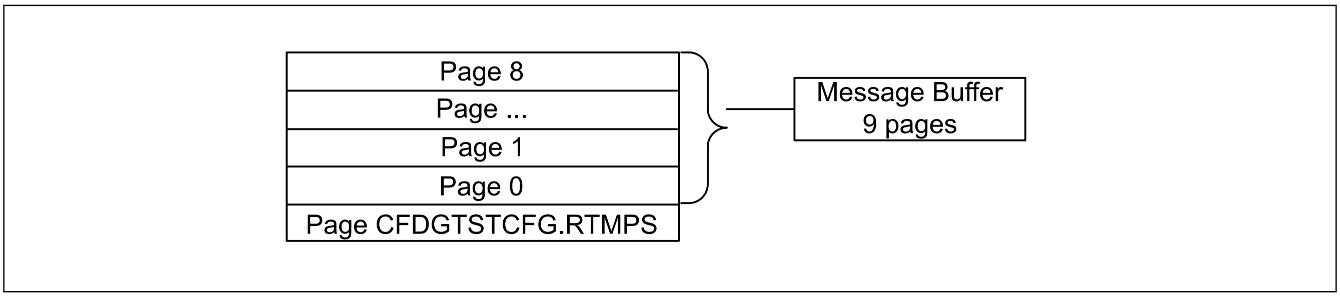


Figure 41.52 RAM page structure

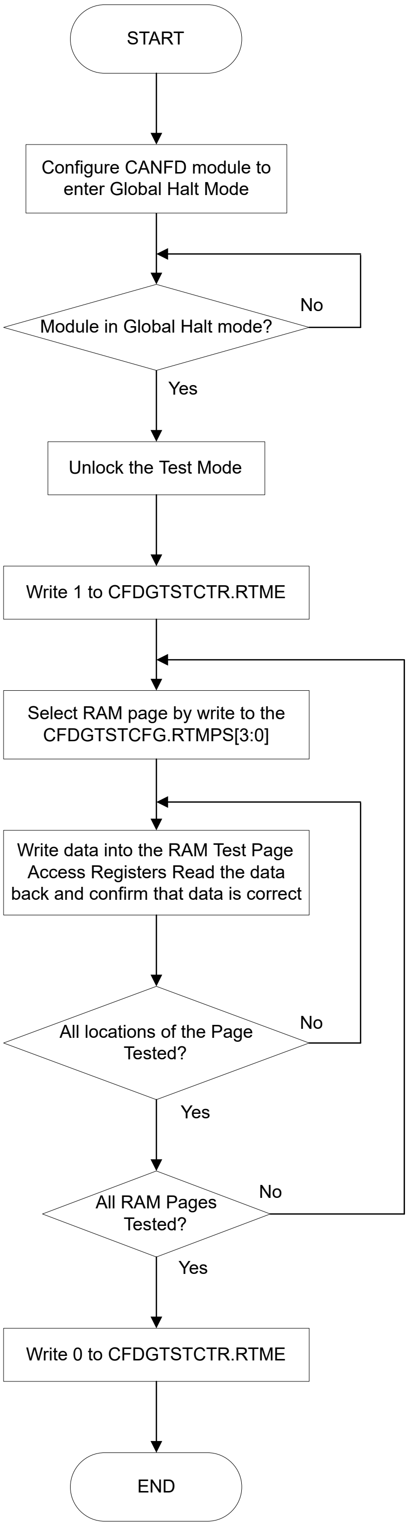
The total available RAM size is 2072 bytes for the Message Buffer RAM.

The pn and CFDGTSTCFG.RTMPS[3:0] values for the MB RAMs are calculated in the following way:

pn = ceil (total RAM size in bytes / number of bytes per page)

- MB RAM:
  pn = ceil (2072 / 256) = 9 pages
  CFDGTSTCFG.RTMPS[3:0] = 0 to 8 inclusive

Figure 41.53 shows the software flow for RAM test mode.



Note: Change into the following status before changing to RAMTEST.

- Cancel of a request for transmission.
- Disable of all the FIFO and TXQ.
- Clear of the receiving flag for a receiving buffer.

Figure 41.53 Software flow for RAM test mode

To exit this test mode, the CFDGTSTCTR.RTME bit must be cleared. The CFDGTSTCTR.RTME bit is cleared by writing 0 to it.

The CFDGTSTCTR.RTME bit is cleared automatically when the CANFD module enters Global Reset mode from the test mode.

41.9.2.2 Bit Flip Test

Bit Flip Test can invert the bit (the 1st bit of ID) of the beginning of the bit stream to receive.

If this function is used by a transmitting node, a bit error or an arbitration lost will occur.

If this function is used by a receiving node, a CRC error or a stuff error will occur.

Users should refer to the bit stuffing rule when using this feature, as there is the possibility of receiving a stuff error (due to the inversion) rather than a CRC error.

The following sequence should be used to perform CRC Error testing. In the sequence below CANFD module is the receiver.

1. Set the CFDC0CTR.BFT bit to 1’b1, in order to invert the first bit of the incoming bit stream from sending node
2. Wait for the CANn_CHERR (n = 0, 1) output signal to set to 1’b1
3. Read either the CFDC0ERFL.CRCREG or the CFDC0FDCRC.CRCREG (depending on the received frame type: Classical or FD). The value should be different from the received CRC value of the reference message from sending node.
4. Check that CFDC0ERFL.CERR is 1’b1

As the CRC generator logic is shared for RX and TX there is no need to create a separate TX CRC Error test.

41.10 Usage Notes

41.10.1 Enter Software Standby Mode

The following steps are required, before the MCU enters Software Standby mode.

1. Set the global halt mode request (CFDGCTR.GMDC = 10b).
2. Wait for completion of transition to global halt mode (CFDGSTS.GHLTSTS = 1b).
3. Set the global reset mode request (CFDGCTR.GMDC = 01b).
4. Wait for completion of transition to global reset mode (CFDGSTS.GRSTSTS = 1b).
5. Set the global sleep mode request (CFDGCTR.GSLPR = 1b).
6. Wait for completion of transition to global sleep mode (CFDGSTS.GSLPSTS = 1b).
7. Read the CFDGLOCKK register.

41.10.2 Return from Software Standby Mode

The procedures described in section 41.4.2. CAN Module Configuration after Hardware Reset are required, after the MCU returns from Software Standby mode.

41.10.3 Module-Stop Function Setting

CANFD operation can be disabled or enabled using Module Stop Control Register C (MSTPCRC). The CANFD module is initially stopped after reset. Releasing the module-stop state enables access to the registers.

42. CANFD ECC (CNECC)

42.1 Overview

MBRAM have ECC function of 2-bit ECC error detection and 1-bit ECC error detection and correction*1. The ECC module adds 7 bits ECC data to 32 bits RAM data.

Note 1. The ECC module cannot detect 3 or more bits error. In this case, the ECC module detects 1-bit or 2-bit error, does not detect errors, or corrects the erroneous bit to erroneous data by setting. When all RAM data are fixed to 0 or 1, it is detected as 2-bit ECC error.

42.2 Register Descriptions

42.2.1 EC710CTL : ECC Control Register

Base address: ECCMBn = 0x4036_F200 + 0x0100 × n (n = 0, 1)
ECCMBn_NS = 0x5036_F200 + 0x0100 × n (n = 0, 1)

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECDEDF0	ECSEDF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EMCA[1:0]		—	—	ECOVFF	ECER2C	ECER1C	—	—	ECERVF	EC1ECP	EC2EDIC	EC1EDIC	ECER2F	ECER1F	ECEMF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
0	ECEMF	ECC Error Message Flag 0: There is no bit error in present RAM output data 1: There is bit error in present RAM output data	R
1	ECER1F	ECC Error Detection and Correction Flag 0: After clearing this bit, 1-bit error correction has not occurred 1: 1-bit error has occurred	R
2	ECER2F	2-bit ECC Error Detection Flag 0: After clearing this bit, 2-bit error has not occurred 1: 2-bit error has occurred	R
3	EC1EDIC	ECC 1-bit Error Detection Interrupt Control 0: Disable 1-bit error detection interrupt request 1: Enable 1-bit error detection interrupt request	R/W
4	EC2EDIC	ECC 2-bit Error Detection Interrupt Control 0: Disable 2-bit error detection interrupt request 1: Enable 2-bit error detection interrupt request	R/W
5	EC1ECP	ECC 1-bit Error Correction Permission 0: At 1-bit error detection, the error correction is executed 1: At 1-bit error detection, the error correction is not executed	R/W
6	ECERVF	ECC Error Judgment Enable Flag 0: Error judgment disable 1: Error judgment enable	R/W
8:7	—	These bits are read as 0. The write value should be 0.	R/W
9	ECER1C	Accumulating ECC Error Detection and Correction Flag Clear 0: No effect 1: Clear accumulating ECC error detection and correction flag	R/W
10	ECER2C	2-bit ECC Error Detection Flag Clear 0: No effect 1: Clear 2-bit ECC error detection flag	R/W

Bit	Symbol	Function	R/W
11	ECOVFF	ECC Overflow Detection Flag 0: No effect 1: ECC overflow detection flag	R
13:12	—	These bits are read as 0. The write value should be 0.	R/W
15:14	EMCA[1:0]	Access Control to ECC Mode Select bit These bits enable or disable write access to ECERVF bit.	R/W
16	ECSEDF0	ECC Single bit Error Address Detection Flag 0: There is no bit error in EC710EAD0 after reset or clearing ECER1F bit 1: Address captured in EC710EAD0 shows that 1-bit error occurred and captured	R
17	ECDEDF0	ECC Dual Bit Error Address Detection Flag 0: There is no bit error in EC710EAD0 after reset or clearing ECER2F bit 1: Address captured in EC710EAD0 shows that 2-bit error occurred and captured	R
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

ECEMF bit (ECC Error Message Flag)

The ECEMF bit shows that there is error in present read data bus. This bit is updated by every RAM output data.

When RAM output data is undefined and the ECERVF bit is set to 1, the value of this bit is undefined.

[Setting condition]

There is bit error in present RAM output data under the condition that error judgment is enabled.

[Clearing condition]

- Under the condition that there is no 1-bit error in input data to decode circuit
- When ECC error judgment is disabled (ECERVF = 0).

ECER1F bit (ECC Error Detection and Correction Flag)

The ECER1F bit shows that the bit errors are detected in the one part of RAM read data [38:0] at RAM read access when the error judgment is enabled.

When the 1-bit error interrupt output is enabled, error interrupt is generated by setting this flag.

This bit is read-only, so writing 1 or 0 has no effect.

At clearing, write 1 to the ECER1C bit.

When 1-bit error is detected again under the condition that this bit is set, the interrupt is not generated.

[Setting condition]

When the error judgment is enabled and there is 1-bit error to RAM output data (when not setting ECER1C = 1).

[Clearing condition]

- Writing ECER1C = 1
- When ECC error judgment is disabled (ECERVF = 0).

ECER2F bit (2-bit ECC Error Detection Flag)

The ECER2F bit shows that the bit errors are detected in the two parts of RAM read data [38:0] at RAM read access when the error judgment is enabled.

When the 2-bit error interrupt output is enabled, error interrupt is generated by setting this flag.

This bit is read-only, so writing 1 or 0 has no effect.

At clearing, write 1 to the ECER2C bit.

When 2-bit error is detected again under the condition that this bit is set, the interrupt is not generated.

[Setting condition]

When the error judgment is enabled and there is 2-bit error to RAM output data (when not setting ECER2C = 1).

[Clearing condition]

- Writing ECER2C = 1
- When ECC error judgment is disabled (ECERVF = 0).

EC1EDIC bit (ECC 1-bit Error Detection Interrupt Control)

The EC1EDIC controls the interrupt output at detecting 1-bit error. By setting 1 to this bit, the 1-bit error interrupt is outputted when 1-bit error detected.

EC2EDIC bit (ECC 2-bit Error Detection Interrupt Control)

The EC2EDIC controls the interrupt output at detecting 2-bit error. By setting 1 to this bit, the 2-bit error interrupt is outputted when 2-bit error detected.

EC1ECP bit (ECC 1-bit Error Correction Permission)

The EC1ECP sets enable or disable to correct the 1-bit error when ECC error detection and correction is valid. By setting 1 to this bit, the non-corrected data is outputted if 1-bit error is detected.

ECERVF bit (ECC Error Judgment Enable Flag)

Setting the ECERVF bit to 1 enables the judgment of error. The correction of output data and the interrupt output depend on setting of the EC1ECP bit, EC2EDIC bit, and EC1EDIC bit.

The write access to this bit is valid when the write value of the EMCA[1:0] is 01b. So only the 16 bits or 32 bits operation command is valid in the case of the write access to this bit.

ECER1C bit (Accumulating ECC Error Detection and Correction Flag Clear)

The ECER1C bit clears the status flag of the ECER1F bit.

The read value is always 0. By writing 0, the internal condition is not changed. When the competition between writing 1 to this bit and setting the ECER1F bit, the former has priority.

The ECER1F bit is cleared by writing 1 to this bit while the ECER1F bit is set. Additionally, the Overflow Detection flag (ECOVFF), ECC Dual Bit Error flag (ECDEDF0) and ECC Single Bit Error flag (ECSEDF0) are also cleared.

ECER2C bit (2-bit ECC Error Detection Flag Clear)

The ECER2C bit clears the status flag of the ECER2F bit.

The read value is always 0. By writing 0, the internal condition is not changed. When the competition between writing 1 to this bit and setting the ECER2F bit, the former has priority.

The ECER2F bit is cleared by writing 1 to this bit while the ECER2F bit is set. Additionally, the Overflow Detection flag (ECOVFF), ECC Dual Bit Error flag (ECDEDF0), and ECC Single Bit Error flag (ECSEDF0) are also cleared.

ECOVFF bit (ECC Overflow Detection Flag)

The ECOVFF bit is set and the overflow interruption is outputted by detecting the new error address under the condition that error address is already captured in the EC710EAD0 register. The overflow interrupt is outputted again when this bit is set and new error is detected.

This bit is read-only, so writing 1 or 0 has no effect.

To clear this bit, write 1 to the ECER2C bit and the ECER1C bit.

[Setting condition]

When new error address is captured under the condition that error address is already captured in the EC710EAD0 register (when not setting ECER2C = 1 or ECER1C = 1).

[Clearing condition]

- Writing ECER2C = 1 and ECER1C = 1
- When ECC error judgment is disabled (ECERVF = 0).

EMCA[1:0] bit (Access Control to ECC Mode Select bit)

The EMCA[1:0] bits are the write trigger reserved bits to the ECERVF bit. The read value is always 0. When the value of these bits is 01b, it is possible to have write access to the ECERVF bit. If these bits are not 01b, write access to the ECERVF bit is ignored and the value is not written.

ECSEDF0 bit (ECC Single bit Error Address Detection Flag)

The ECSEDF0 bit shows that the error is captured in the error address register when error detection is valid. This bit is set by 1-bit error detection.

When 1-bit error is detected after the 2-bit error address is already captured in the EC710EAD0 register, this bit is not updated and the EC710EAD0 register is not updated.

This bit is read-only, so writing 1 or 0 has no effect. To clear this bit, write 1 to the ECER2C bit and the ECER1C bit.

[Setting condition]

When there is 1-bit error to RAM output data and error address is captured in EC710EAD0 under the condition that the error judgment is permitted (when not setting ECER1C = 1).

[Clearing condition]

- Writing ECER2C = 1 and ECER1C = 1
- When ECC error judgment is disabled (ECERVF = 0).

ECDEDF0 bit (ECC Dual Bit Error Address Detection Flag)

The ECDEDF0 bit shows that the error is captured in the error address register when error detection is valid. This bit is set by 2-bit error detection.

When 2-bit error is detected after the 1-bit error address is already captured in the EC710EAD0 register, this bit is not updated but the EC710EAD0 register is updated.

This bit is read-only, so writing 1 or 0 has no effect. To clear this bit, write 1 to the ECER2C bit and the ECER1C bit.

[Setting condition]

When there is 2-bit error to RAM output data and error address is captured in EC710EAD0 under the condition that the error judgment is permitted (when not setting ECER2C = 1).

[Clearing condition]

- Writing ECER2C = 1 and ECER1C = 1
- When ECC error judgment is disabled (ECERVF = 0).

42.2.2 EC710TMC : ECC Test Mode Control Register

Base address: ECCMBn = 0x4036_F200 + 0x0100 × n (n = 0, 1)
ECCMBn_NS = 0x5036_F200 + 0x0100 × n (n = 0, 1)

Offset address: 0x04

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ETMA[1:0]		—	—	—	—	—	—	ECTMCE	—	—	—	—	—	ECDCS	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	ECDCS	ECC Decode Input Select 0: Input lower 32 bits of RAM output data to data area of decode circuit 1: Input ECEDB31-0 in EC710TED register to data area of decode circuit	R/W
6:2	—	These bits are read as 0. The write value should be 0.	R/W
7	ECTMCE	ECC Test Mode Control Enable 0: The access to test mode register and bit is disabled 1: The access to test mode register and bit is enabled	R/W
13:8	—	These bits are read as 0. The write value should be 0.	R/W
15:14	ETMA[1:0]	ECC Test Mode Bit Access Control These bits enable or disable write access to ECTMCE bit.	R/W

Note: S-TYPE-3, P-TYPE-3

ECDCS bit (ECC Decode Input Select)

The ECDCS bit selects either the lower 32 bits data value from RAM or value from the internal test register (EDEDB[31:0] in EC710TED) as input signal to decoder.

The write access to this bit is valid under the condition of ECTMCE = 1 (it is possible to set them at the same time.)

This bit is cleared by setting ECTMCE = 0.

ECTMCE bit (ECC Test Mode Control Enable)

The ECTMCE bit selects the access enable or disable to test register and test control bit.

The write access to this bit is valid under the condition that the value of the ETMA[1:0] bits is 10b.

ETMA[1:0] bits (ECC Test Mode Bit Access Control)

The ETMA[1:0] bits are the write trigger reserved bits to the ECTMCE bit. The read value is always 0. When the value of these bits is 10b, it is possible to have write access to the ECTMCE bit. If these bits are not 10b, the write access to the ECTMCE bit is ignored and the value is not written.

42.2.3 EC710TED : ECC Test Substitute Data Register

Base address: ECCMBn = 0x4036_F200 + 0x0100 × n (n = 0, 1)
ECCMBn_NS = 0x5036_F200 + 0x0100 × n (n = 0, 1)

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ECEDB[31:0]																															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	ECEDB[31:0]	ECC Test Substitute Data Substitute data in ECC test mode.	R/W

Note: S-TYPE-3, P-TYPE-3

This register is for the 32 bits data for ECC decode. It is possible to read and write using 32 bits operation command when ECTMCE = 1. When ECTMCE = 0, all bits are always 0.

ECEDB[31:0] bits (ECC Test Substitute Data)

When ECDCS in EC710TMC register is 1, the value of this register is bits [31:0] of the input data to the decode circuit.

42.2.4 EC710EAD0 : ECC Error Address Register

Base address: ECCMBn = 0x4036_F200 + 0x0100 × n (n = 0, 1)
ECCMBn_NS = 0x5036_F200 + 0x0100 × n (n = 0, 1)

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	ECEAD[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	ECEAD[9:0]	ECC Error Address	R

Bit	Symbol	Function	R/W
31:10	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

This is a read-only register to hold the ECC error address.

ECEAD[9:0] bits (ECC Error Address)

When ECC error is detected for permitting ECC error judgment, RAM address is captured by the detected signal as a trigger and is hold as the error occurring address. The error address is not captured when the error occurred again to the one held by the same factor.

If 2-bit error occurred under the condition that 1-bit error address is already captured, the 2-bit error address is over-written but the ECDEDF0 bit is not set to 1.

If 1-bit error occurred under the condition that 2-bit error address is already captured, the 1-bit error address is not overwritten and the ECSEDF0 bit is not set to 1.

This register is reset by clearing the status flag (the ECER2F bit or the ECER1F bit).

42.3 Operation

42.3.1 ECC Function Setting

Figure 42.1 shows a procedure for ECC function setting.

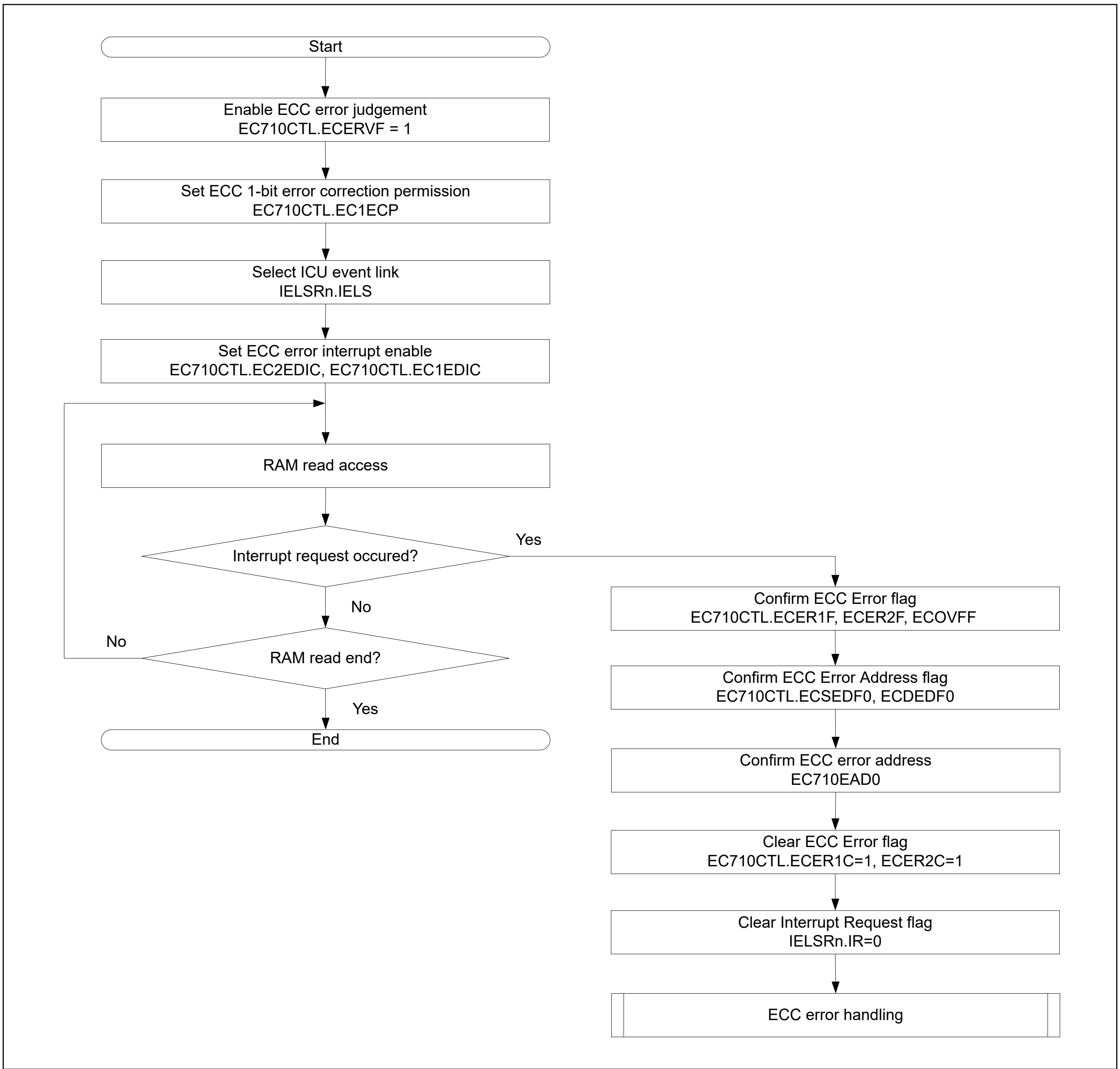


Figure 42.1 Setting procedure for ECC function

42.3.2 ECC Decoder Testing

ECC interrupts can be intentionally generated by ECC test mode. Figure 42.2 shows a procedure for ECC decoder testing.

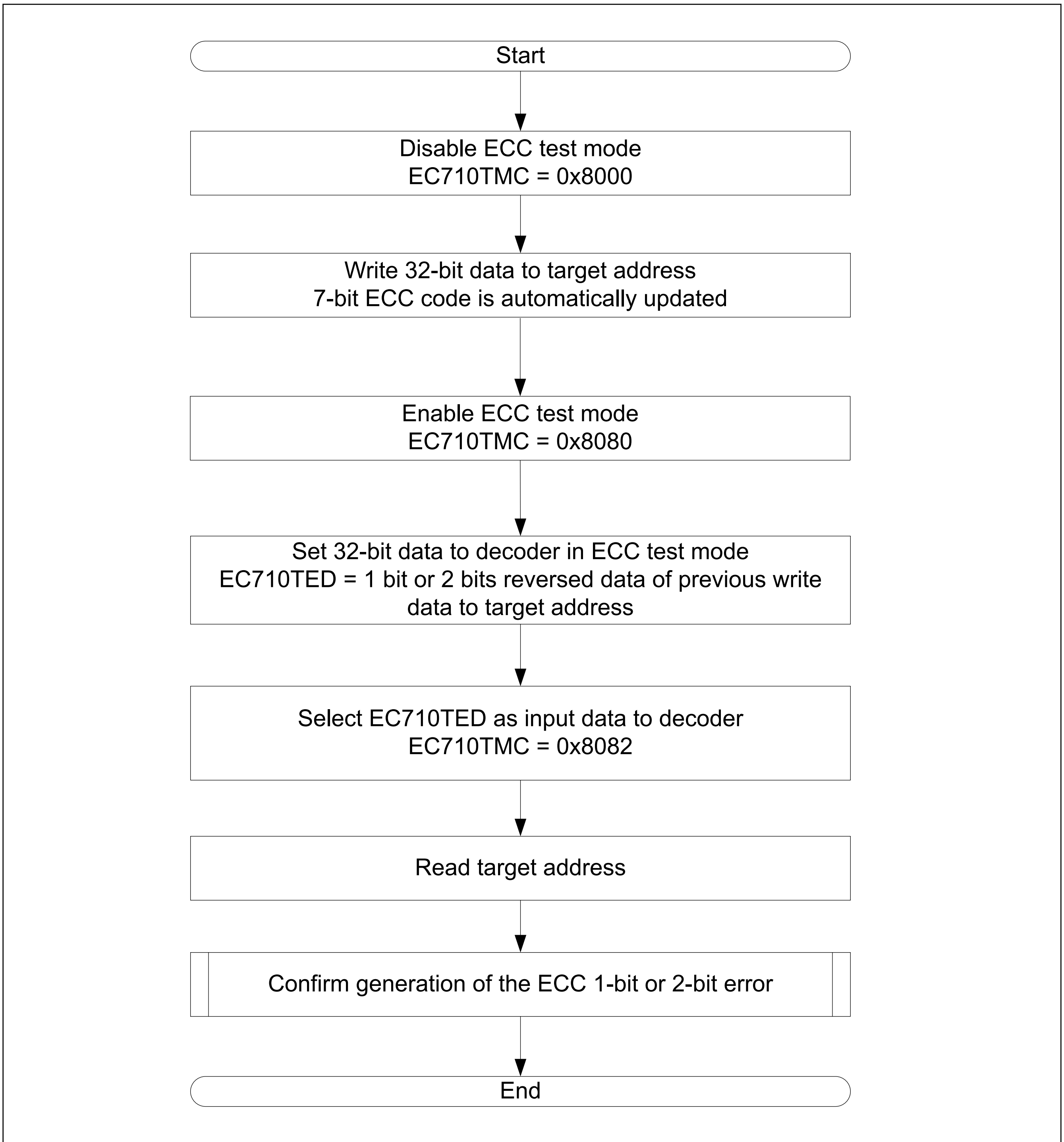


Figure 42.2 Testing procedure for ECC decoder

42.4 Interrupts

The ECC module issues three interrupt requests:

- CANn_MRAM_ERI (n =0, 1)

Interrupt sources of each interrupt request include:

- 1-bit ECC error
- 2-bit ECC error
- ECC error overflow.

42.5 Usage Notes

42.5.1 Enter Software Standby Mode

The following steps are required, before the MCU enters Software Standby mode.

1. To clear the ECC error flags and ECC error address, write 1 to the ECER2C bit and the ECER1C bit.
2. To disable ECC error judgment, write 0 to the ECERVF bit.

42.5.2 Return from Software Standby Mode

The following steps are required, after the MCU returns from Software Standby mode.

1. Make initial settings.
2. To enable ECC error judgment, write 1 to the ECERVF bit.

43. Serial Peripheral Interface (SPI)

This is the SPI_B version of the SPI peripheral module.

SPI_B is referred to as SPI in this chapter.

43.1 Overview

The Serial Peripheral Interface (SPI) has 2 channels. The SPI provides high-speed full-duplex synchronous serial communications with multiple processors and peripheral devices. Table 43.1 lists the SPI specifications, Figure 43.1 shows a block diagram of SPI, Figure 43.2 shows a clock source selector block diagram and Table 43.2 lists the I/O pins.

In this section, PCLK refers to PCLKA.

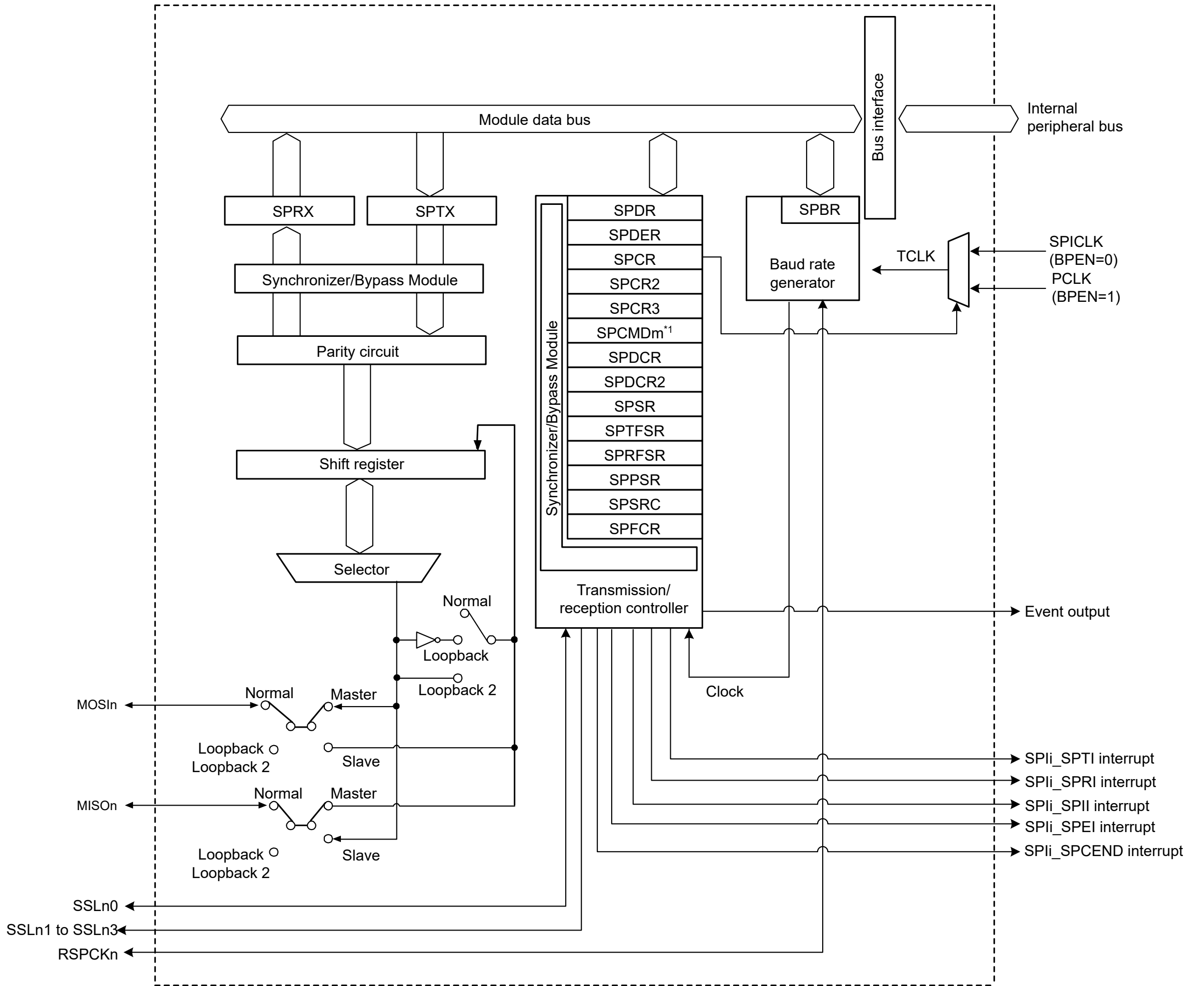
Table 43.1 SPI specifications (1 of 2)

Parameter	Specifications
Number of channels	Two channels
SPI transfer functions	• Use of MOSI (master out/slave in), MISO (master in/slave out), SSL (slave select), and RSPCK (SPI clock) signals allows serial communications through SPI operation (4-wire method) or clock synchronous operation (3-wire method) • Transmit-only operation available • Receive-only operation is available • Communication mode selectable to full-duplex, transmit-only or receive-only • RSPCK polarity switching • RSPCK phase switching
Data format	• MSB-first or LSB-first selectable • Transfer bit length selectable from 4 to 32 bits • 32 bit × 4 stages FIFO is available as transmit buffer or receive buffer • Byte swap operating function • Transmit/receive data can be inverted.
Operation clock (TCLK)	Synchronized clock (PCLK) or independent clock (SPICLK) can be selected.
Bit rate	• In master mode, the on-chip baud rate generator generates RSPCK by frequency-dividing TCLK (the division ratio ranges from divided by 2 to divided by 4096) • In slave mode, the minimum TCLK clock divided by 2 can be input as RSPCK (TCLK divided by 2 is the maximum RSPCK frequency) Width at high level: 1 TCLK cycle; width at low level: 1 TCLK cycle
Buffer configuration	• Double buffer configuration for the transmit and receive buffers
Error detection	• Mode fault error detection • Underrun error detection • Overrun error detection*1 • Parity error detection • Receive data ready detection
SSL control function	[motorola SPI mode/TI SSP mode common] • Four SSL pins (SSLni: SSLn0 to SSLn3) (n = A, B) for each channel • In single-master mode, SSLn0 to SSLn3 pins are output • In multi-master mode, SSLn0 pin for input, and SSLn1 to SSLn3 pins either for output or unused • In slave mode, SSLn0 pin for input and SSLn1 to SSLn3 pins unused • Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Function for changing SSL polarity • Delay between frames in burst transfer is settable [only Motorola mode] • Controllable delay from RSPCK stop to SSL output negation (SSL negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) [only TI SSP mode] • Controllable delay from RSPCK stop to SSL output negation (SSL negation delay) Range: 0 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable delay from OE output assertion to RSPCK operation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units)
Communication protocol	• Motorola SPI • TI SSP (Synchronous Serial Protocol)

Table 43.1 SPI specifications (2 of 2)

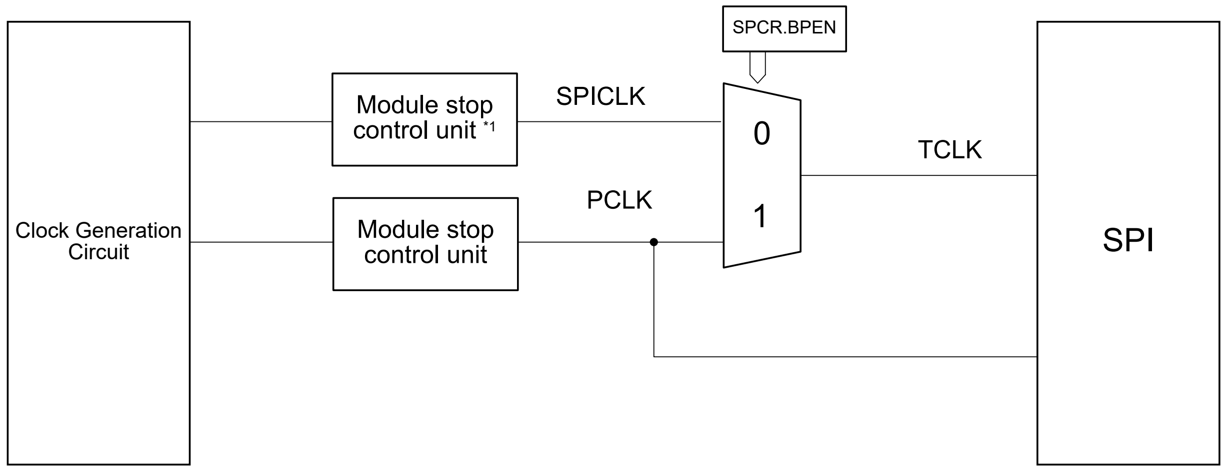
Parameter	Specifications
Synchronization bypass function	Synchronization circuit can be bypassed only when the same clock is input to bus clock (PCLK) and operation clock (TCLK).
Control in master transfer	● Transfers of up to eight commands each can be executed sequentially in looped execution ● For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity and phase, transfer data length, MSB- or LSB-first, burst, RSPCK delay, SSL negation delay, and next-access delay ● Transfers can be initiated by writing to the transmit buffer ● MOSI signal value specifiable in SSL negation ● RSPCK auto-stop function
Interrupt sources	Interrupt sources: ● Receive buffer full / Receive data ready interrupt ● Transmit buffer empty interrupt ● SPI error interrupt (mode fault error, under run error, over run error, parity error, receive data ready) ● SPI idle interrupt (SPI idle) ● Communication end interrupt
Event link function	The following events can be output to the Event Link Controller (ELC): ● Receive buffer full / receive data ready signal ● Transmit buffer empty signal ● Mode fault, underrun, overrun, parity error, or receive data ready signal ● SPI idle signal ● Communication end signal
Other functions	● Switching between CMOS output and open-drain output ● SPI initialization function ● Loopback mode ● SPE bit status polling function
Module-stop function	Module-stop state can be set for each channel to reduce power consumption.
TrustZone Filter	Security and Privilege attribution can be set for each channel.

Note 1. In master reception and when the RSPCK auto-stop function is enabled, an overrun error does not occur because the transfer clock is stopped on overrun error detection.



Note: i = 0, 1
Note: n = A, B
Note 1. m = 0 to 7

Figure 43.1 SPI block diagram



Note 1. If the module-stop state of any SPI channel is canceled, SPICLK is supplied.

Figure 43.2 Clock source selector block diagram

The SPI automatically switches the I/O direction of the SSLn0 pin. SSLn0 is set as an output when the SPI is a single master, and as an input when the SPI is a multi-master or a slave. The RSPCKn, MOSIn, and MISOn pins are automatically set as inputs or outputs based on the master or slave setting and the level input on the SSLn0 pin. For details, see section 43.3.2. Controlling the SPI Pins.

Table 43.2 SPI I/O pins

Channel	Pin name	I/O	Description
SPI0	RSPCKA	I/O	Clock input/output pin
	SSLA0	I/O	Slave selection input/output
	SSLA1 to SSLA3	Output	Slave selection output
	MOSIA	I/O	Master transmit data input/output
	MISOA	I/O	Slave transmit data input/output
SPI1	RSPCKB	I/O	Clock input/output pin
	SSLB0	I/O	Slave selection input/output
	SSLB1 to SSLB3	Output	Slave selection output
	MOSIB	I/O	Master transmit data input/output
	MISOB	I/O	Slave transmit data input/output

Note: Pin names are indicated as “...A” or “...An” for SPI0, and “...B” or “...Bn” for SPI1 (n = 0, 1, 2, or 3).

43.2 Register Descriptions

43.2.1 SPDR : SPI Data Register

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x00

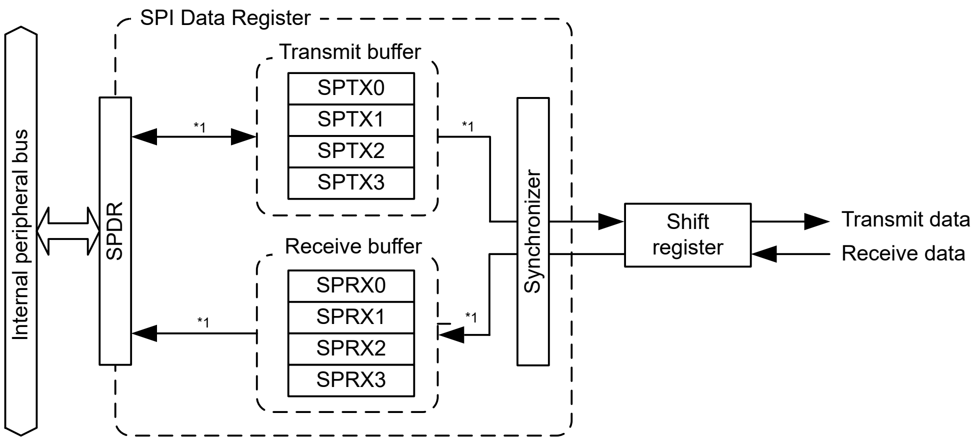
Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SPD[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SPD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SPD[31:0]	These bits are the interface with the buffers that hold data for transmission and reception by the SPI.	R/W

Note: S-TYPE-3, P-TYPE-3

SPDR is the interface with the buffers that hold data for transmission and reception by the SPI. When accessing this register in words, access SPDR. The transmit buffer (SPTX) and receive buffer (SPRX) are independent, but both are mapped to SPDR. Figure 43.3 shows the configuration of the SPDR register.



Note 1. The destination buffer and stage for access is automatically switched by hardware.

Figure 43.3 Structure of SPDR

32 bits × 4 stage transmit FIFO and 32 bits × 4 stage receive FIFO are provided. These 8 stage FIFO are mapped to one address in the SPDR. Transmit buffers (SPTXn, n = 0 to 3) can be written by writing data to SPDR to transmit written data.

Upon completion of receiving data, receive buffers store received data. When an overrun error occurs, data in the receive buffer is not updated.

(1) Bus Interface

The SPI data register has 32 bits × 4 stage transmit FIFO and 32 bits × 4 stage receive FIFO (32 bytes in total). These 32 bytes are mapped to the 4-byte space of SPDR. Write transmit data from the LSB. Received data is stored from the LSB.

SPDR register write operation and read operation are described below.

1. Write

A transmit buffer write pointer is provided for transmit buffers. When data is written to SPDR, the pointer automatically switches to the next buffer. The following illustrates the structure of the transmit buffer bus interface (write).

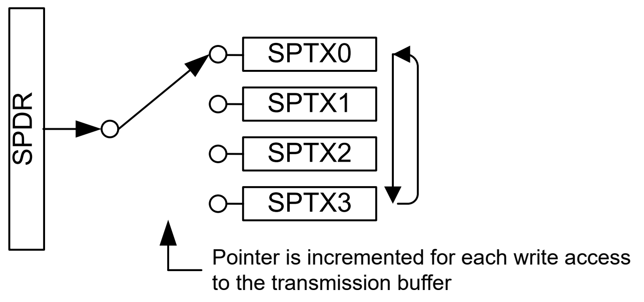


Figure 43.4 Structure of SPDR (Write)

The transmit buffer (SPTX0 to SPTX3) switching order:
SPTX0→SPTX1→SPTX2→SPTX3→SPTX0→SPTX1→…
When writing transmit data to transmit buffers (SPTXn), write transmit data of frames +1 specified by the Transmission FIFO threshold setting bits of SPI data control register 2 (SPDCR2.TTRG[1:0]) while an SPI transmit buffer empty interrupt is present (SPSR.SPTEF flag = 1). Writing to the transmit buffer (SPTXn, n = 0 to 3) in the state where there is no empty stage in the transmit FIFO does not update the buffer value.

2. Read
Values can be read from receive buffers (SPRXn, n = 0 to 3) or transmit buffers (SPTXn, n = 0 to 3) by reading the SPDR register. Reading a receive buffer or reading a transmit buffer can be selected by the SPI receive data or transmit data select bit (SPDCR.SPRDTD) in the SPI data control register.
The SPDR register is read according to the independent receive buffer read pointer and the transmit buffer read pointer. The following illustrates the structure of the receive buffer and transmit buffer bus interface (read).

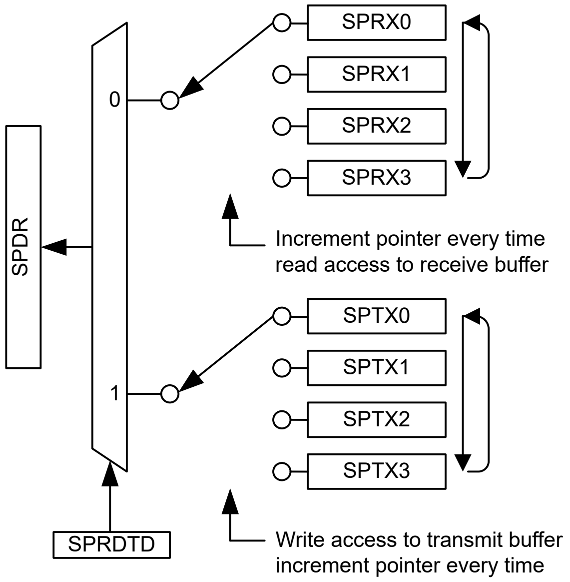


Figure 43.5 Structure of SPDR (Read)

When a receive buffer is read, the receive buffer read pointer automatically switches to the next buffer. The receive buffer read pointer switches in the same order as the transmit buffer write pointer.
The transmit buffer read pointer is updated during the SPDR write access, but it is not updated during the transmit buffer read access. When a transmit buffer is read, the value written to SPDR last can be read.

43.2.2 SPDECR : SPI Delay Control Register

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SPNDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SLNDL[2:0]			—	—	—	—	—	SCKDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SCKDL[2:0]	RSPCK Delay 0 0 0: 1RSPCK 0 0 1: 2RSPCK 0 1 0: 3RSPCK 0 1 1: 4RSPCK 1 0 0: 5RSPCK 1 0 1: 6RSPCK 1 1 0: 7RSPCK 1 1 1: 8RSPCK	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
10:8	SLNDL[2:0]	SSL Negation Delay [Master Mode] 0 0 0: 1RSPCK 0 0 1: 2RSPCK 0 1 0: 3RSPCK 0 1 1: 4RSPCK 1 0 0: 5RSPCK 1 0 1: 6RSPCK 1 1 0: 7RSPCK 1 1 1: 8RSPCK [Motorola-SPI case in Slave Mode] 0 0 0: 1RSPCK Others: Setting prohibited [TI-SSP case in Slave Mode] 0 0 0: 1 TCLK 0 0 1: 2 TCLK 0 1 0: 3 TCLK 0 1 1: 4 TCLK 1 0 0: 5 TCLK 1 0 1: 6 TCLK 1 1 0: 7 TCLK 1 1 1: 8 TCLK	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
18:16	SPNDL[2:0]	SPI Next-Access Delay 0 0 0: 1RSPCK + 5TCLK 0 0 1: 2RSPCK + 5TCLK 0 1 0: 3RSPCK + 5TCLK 0 1 1: 4RSPCK + 5TCLK 1 0 0: 5RSPCK + 5TCLK 1 0 1: 6RSPCK + 5TCLK 1 1 0: 7RSPCK + 5TCLK 1 1 1: 8RSPCK + 5TCLK	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

SCKDL[2:0] bit (RSPCK Delay)

[In the Motorola-SPI case]

The RSPCK delay bits (SCKDL) are used to set the period (RSPCK delay) from SSL signal assertion start until RSPCK oscillates while the SCKDEN bit in the SPI command register (SPCMD) is 1. If SCKDL is modified while the MSTR bit and the SPE bit in the SPI control register (SPCR) are 1, subsequent operation is not guaranteed.

To use the SPI in slave mode, set SCKDL[2:0] bits to 000b.

[In the TI-SSP case]

The RSPCK delay bits (SCKDL) are used to set the period (RSPCK delay) from SSL signal assertion start until RSPCK oscillates while the SCKDEN bit in the SPI command register (SPCMD) is 1. Also that is used to set the period until the SSL signal is negated. If SCKDL is modified while the MSTR bit and the SPE bit in the SPI control register (SPCR) are 1, subsequent operation is not guaranteed.

To use the SPI in slave mode, set SCKDL[2:0] bits to 000b.

SLNDL[2:0] bit (SSL Negation Delay)

[In the Motorola-SPI case]

The SSL negation delay bits (SLNDL) are used to set the period (SSL negation delay) after the SPI in master mode sends the final RSPCK edge during serial transfer until it negates the SSL signal while the SLNDEN bit in the SPI command register (SPCMD) is 1. If SLNDL is modified while the MSTR bit and the SPE bit in the SPI control register (SPCR) are 1, subsequent operation is not guaranteed.

To use the SPI in slave mode except TI-SSP, set SLNDL[2:0] bits to 000b.

[In the TI-SSP case]

The SSL negation delay bits (SLNDL) are used to set the period (OE negation delay) after the SPI in master mode sends the final RSPCK edge during serial transfer until it negates the OE signal while the SLNDEN bit in the SPI command register (SPCMD) is 1. Also, that is used to set the period from when the SPI in slave mode detects the last RSPCK edge of serial transfer to when the OE signal is negated. If SLNDL is modified while the SPE bit in the SPI control register (SPCR) are 1, subsequent operation is not guaranteed.

SPNDL[2:0] bit (SPI Next-Access Delay)

The SPI next-access delay register (SPDECR.SPNDL) is used to set the SSL signal inactive period (next-access delay) after completion of serial transfer while the SPNDEN bit in the SPI command register (SPCMD) is 1. If SPNDL is modified while the MSTR bit and the SPE bit in the SPI control register (SPCR) are 1, subsequent operation is not guaranteed.

These bits are used to set the next-access delay value when the SPNDEN bit in SPCMD is 1. To use the SPI in slave mode, set SPNDL[2:0] bits to 000b.

43.2.3 SPCR : SPI Control Register

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	BPEN	MSTR	TXMD[1:0]		—	—	SPFRF	SPMS	—	—	CENDIE	SPTIE	SPDRES	SPIIE	SPRIE	SPEIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	MODFEN	BFDS	SCKASE	PTE	—	SPOE	SPPE	—	—	—	—	—	—	—	SPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPE	SPI Function Enable 0: SPI function is disabled. 1: SPI function is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	SPPE	Parity Enable 0: A parity bit is not added to transmit data. Received-data parity check is not performed. 1: A parity bit is added to transmit data. Received-data parity check is performed.	R/W
9	SPOE	Parity Mode 0: Even parity is used for transmission and reception. 1: Odd parity is used for transmission and reception.	R/W
10	—	This bit is read as 0. The write value should be 0.	R/W
11	PTE	Parity Self-Diagnosis Enable 0: Parity circuit self-diagnosis function is disabled. 1: Parity circuit self-diagnosis function is enabled.	R/W
12	SCKASE	RSPCK Auto-Stop Function Enable 0: RSPCK auto-stop function is disabled. 1: RSPCK auto-stop function is enabled.	R/W
13	BFDS	Between Burst Transfer Frames Delay Select 0: Delay (RSPCK delay, SSL negation delay and next-access delay) between frames is inserted in burst transfer 1: Delay between frames is not inserted in burst transfer.	R/W
14	MODFEN	Mode Fault Error Detection Enable 0: Mode fault error detection is disabled. 1: Mode fault error detection is enabled.	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
16	SPEIE	SPI Error Interrupt Enable 0: SPI error interrupt request is disabled. 1: SPI error interrupt request is enabled.	R/W
17	SPRIE	SPI Receive Buffer Full Interrupt Enable 0: SPI receive buffer full interrupt request is disabled. 1: SPI receive buffer full interrupt request is enabled.	R/W
18	SPIIE	SPI Idle Interrupt Enable 0: Idle interrupt request is disabled. 1: Idle interrupt request is enabled.	R/W
19	SPDRES	SPI receive data ready error select Select the interrupt request to be generated when the reception data ready is detected 0: Receive data full interrupt 1: Error interrupt	R/W
20	SPTIE	SPI Transmit Buffer Empty Interrupt Enable 0: SPI transmit buffer empty interrupt request is disabled. 1: SPI transmit buffer empty interrupt request is enabled.	R/W
21	CENDIE	SPI Communication End Interrupt Enable 0: Communication end interrupt request is disabled. 1: Communication end interrupt request is enabled.	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
24	SPMS	SPI Mode Select 0: SPI operation (4-wire) 1: Clock synchronous operation (3-wire)	R/W
25	SPFRF	SPI Frame Format Select 0: Motorola-SPI 1: TI-SSP Note: When SPMS = 1 (clock synchronous operation (3-wire)), this bit setting is invalid.	R/W

Bit	Symbol	Function	R/W
27:26	—	These bits are read as 0. The write value should be 0.	R/W
29:28	TXMD[1:0]	Communication Mode Select 0 0: Transmit-Receive 0 1: Transmit only Others: Receive only	R/W
30	MSTR	SPI Master/Slave Mode Select 0: Slave mode 1: Master mode	R/W
31	BPEN	Synchronization Circuit Bypass Enable 0: Non-Bypass 1: Bypass	R/W

Note: S-TYPE-3, P-TYPE-3

The SPI control register (SPCR) is used to set operating mode of the SPI. If the set BPEN, MSTR, TXMD[1:0], SPFRF, SPMS, MODFEN, BFDS, SCKASE, PTE, SPOE, SPPE bit value is modified while the SPE bit = 1, subsequent operation is not guaranteed.

SPE bit (SPI Function Enable)

This bit is used to enable or disable SPI functions. Setting this bit to 1 enables SPI functions. When the MODF flag in the SPI status register (SPSR) is 1, the SPE bit is cleared to 0 and the SPE bit cannot be set to 1 until the MODF flag is cleared to 0. (See section 43.3.10. Error Detection) Setting the SPE bit to 0 disables SPI functions and initializes a part of module functions. (See section 43.3.11. Initializing the SPI)

SPPE bit (Parity Enable)

This bit is used to enable or disable the parity function.

SPOE bit (Parity Mode)

This bit is used to specify even parity or odd parity.

In even parity mode, the parity bit is determined so that the sum of 1 (parity bit + transmit characters or receive characters) becomes an even number. In the same way, in odd parity mode, a parity bit is determined so that the sum of 1 (parity bit + transmit characters or receive characters) becomes an odd number. The SPOE bit is valid only when the SPPE bit in SPCR is set to 1.

PTE bit (Parity Self-Diagnosis Enable)

This bit is used to enable or disable self-diagnosis of the parity circuit to confirm that the parity function is normal.

SCKASE bit (RSPCK Auto-Stop Function Enable)

This bit is used to enable or disable the RSPCK auto-stop function. When this function is enabled, the RSPCK clock stops immediately before an overrun error occurs during data reception in master mode. For details, see section 43.3.10.1. Overrun Errors.

BFDS bit (Between Burst Transfer Frames Delay Select)

This bit controls whether insert the delay time between the burst transfer frames.

Valid in the master mode (SPCR.MSTR = 1) for frames with the SPCMDn.SSLKP bit set to 1.

This bit should be set to 0 in slave mode. The usage of SSL delay control between transfer frames is shown as below. For details, see section 43.3.12.1. Master Mode Operation.

1. Non-burst transmits
2. Burst transmit with delay between frames

    2-1. From the 1st frame to the last previous frame

    2-2. The last frame
3. Burst transmit with no delay between frames

    3-1. From the 1st frame to the last previous frame

3-2. The last frame

Table 43.3 Usage of SSL delay control between transfer frames (Master mode)

	SPCMDn. SSLKP bit	SPCR.BF DS bit	SSL delay control register[*1] (RSPCK clock delay, SSL negation delay, next access delay)
1	0	0	Any given value. You can control each delay value according to setting for RSPCK clock delay, SSL negation delay and next access delay
2-1	1	0	
2-2	0	0	
3-1	1	1	Any given value. But delay is inserted only below. ● RSPCK clock delay of the 1st frame ● SSL negation delay and next access delay of the last frame
3-2	0	1	

Note 1. Whether the setting value of following bits are valid or not depends on the setting value of the SPCMD.SPNDEN bit. (See section 43.2.6. SPCMDm : SPI Command Register (m = 0 to 7).)
The SPDECR.SCKDL[2:0] bits: RSPCK delay
The SPDECR.SLNDL[2:0] bits: SSL negate delay
The SPDECR.SPNDL[2:0] bits: Next access delay

< Setting / operation example > (Motorola SPI, BFDS = 1 Case)

SPCMD0.SSLKP = 1 → Burst transfer / no interframe delay between 0 and 1 (SSL keep active)

SPCMD1.SSLKP = 1 → Burst transfer / no interframe delay between 1 and 2 (SSL keep active)

SPCMD2.SSLKP = 1 → Burst transfer / no interframe delay between 2 and 3 (SSL keep active)

SPCMD3.SSLKP = 1 → Burst transfer / no interframe delay between 3 and 4 (SSL keep active)

SPCMD4.SSLKP = 0 → do not Burst Transfer, and inactive SSL. (BFDS setting is invalid because it does not Burst Transfer.)

SPCMD5.SSLKP = 1 → Burst transfer / no interframe delay between 5 and 6 (SSL keep active)

SPCMD6.SSLKP = 1 → Burst transfer / no interframe delay between 6 and 7 (SSL keep active)

SPCMD7.SSLKP = 0 → do not Burst Transfer, and inactive SSL. (BFDS setting is invalid because it does not Burst Transfer.)

MODFEN bit (Mode Fault Error Detection Enable)

This bit is used to enable or disable detection of a mode fault error. (See section 43.3.10. Error Detection.) The SPI determines SSL0 pin input or output direction according to the combination of the MODFEN and MSTR bits. (See section 43.3.2. Controlling the SPI Pins.)

SPEIE bit (SPI Error Interrupt Enable)

This bit is used to enable or disable an SPI error interrupt request when the SPI detects a mode fault error or an underrun error and sets the MODF flag in the SPI status register (SPSR) to 1, when the SPI detects an overrun error and sets the OVRF flag in SPSR to 1, or when the SPI detects a parity error and sets the PERF flag in SPSR to 1. (See section 43.3.10. Error Detection)

SPRIE bit (SPI Receive Buffer Full Interrupt Enable)

This bit is used to enable or disable a receive buffer full interrupt request of the SPI.

SPIIE bit (SPI Idle Interrupt Enable)

This bit is used to enable or disable an idle interrupt request of the SPI after the SPI detects the idle state and sets the IDLNF flag in the SPI status register (SPSR) to 0.

SPDRES bit (SPI receive data ready error select)

When a receive data ready is detected (SPSR.SPDRF = 1), select whether to use SPIi_SPRI (i = 0, 1) interrupt request or SPIi_SPEI (i = 0, 1) interrupt request.

SPTIE bit (SPI Transmit Buffer Empty Interrupt Enable)

This bit is used to enable or disable a transmit buffer empty interrupt request of the SPI.

A transmit buffer empty interrupt request at the beginning of transmission is generated by setting the SPE bit to 1 simultaneously when or after the SPTIE bit is set to 1. Note that a transmit buffer empty interrupt is generated while the SPTIE bit is 1 even though SPI functions are disabled (SPE bit = 0).

CENDIE bit (SPI Communication End Interrupt Enable)

This bit controls generation of a communication end interrupt request.

SPMS bit (SPI Mode Select)

This bit is used to select SPI operation (4-wire) or clock synchronous operation (3-wire).

For clock synchronous operation, the SSL pin is not used but three pins RSPCK, MOSI, and MISO are used for communication. When SPMS = 1 (clock synchronous operation (3-wire)), the setting of the SPFRF bit is invalid.

To perform clock synchronous operation in master mode (SPCR.MSTR = 1), set the CPHA bit in the SPI command register (SPCMD) to 0 or 1. To perform clock synchronous operation in slave mode (SPCR.MSTR = 0), set the CPHA bit to 1. If this bit is set to 0 for clock synchronous operation in slave mode (SPCR.MSTR = 0), subsequent operation is not guaranteed.

The communication status according to the settings of the MSTR bit, TXMD[1:0] bits, SPFRF bit, and SPMS bit of the SPI control register (SPCR) as follows.

Table 43.4 SPI Communication Status

SPCR. MSTR	SPCR.TXMD[1]	SPCR.TXMD[0]	SPCR. SPFRF	SPCR. SPMS	Communication Status	Communication Status No
1	0	0	0	0	Transmit-Receive Master / Motorola SPI / SPI operation (4-wire)	1-(1)
1	0	0	1	0	Transmit-Receive Master / TI-SSP / SPI operation (4-wire)	1-(2)
1	0	0	—	1	Transmit-Receive Master /Clock synchronous operation (3-wire)	1-(3)
1	0	1	0	0	Transmit only Master / Motorola SPI / SPI operation (4-wire)	1-(4)
1	0	1	1	0	Transmit only Master / TI-SSP / SPI operation (4-wire)	1-(5)
1	0	1	—	1	Transmit only Master /Clock synchronous operation (3-wire)	1-(6)
1	1	—	0	0	Receive only Master / Motorola SPI / SPI operation (4-wire)	1-(7)
1	1	—	1	0	Receive only Master / TI-SSP / SPI operation (4-wire)	1-(8)
1	1	—	—	1	Receive only Master /Clock synchronous operation (3-wire)	1-(9)
0	0	0	0	0	Transmit-Receive Slave / Motorola SPI / SPI operation (4-wire) (default)	0-(1)
0	0	0	1	0	Transmit-Receive Slave / TI-SSP / SPI operation (4-wire)	0-(2)
0	0	0	—	1	Transmit-Receive Slave /Clock synchronous operation (3-wire)	0-(3)
0	0	1	0	0	Transmit only Slave / Motorola SPI / SPI operation (4-wire)	0-(4)
0	0	1	1	0	Transmit only Slave / TI-SSP / SPI operation (4-wire)	0-(5)
0	0	1	—	1	Transmit only Slave /Clock synchronous operation (3-wire)	0-(6)
0	1	—	0	0	Receive only Slave / Motorola SPI / SPI operation (4-wire)	0-(7)
0	1	—	1	0	Receive only Slave / TI-SSP / SPI operation (4-wire)	0-(8)
0	1	—	—	1	Receive only Slave /Clock synchronous operation (3-wire)	0-(9)

SPFRF bit (SPI Frame Format Select)

This bit selects the communication protocol.

The format of the SPI terminal (RSPCK, SSL0 to 7) can be set according to the set communication protocol.

When SPMS = 1 (clock synchronous operation (3-wire)), this bit is invalid because SSL is not used.

TXMD[1:0] bit (Communication Mode Select)

This bit is used to select the transmit-receive, transmit-only, and receive-only serial communication.

When TXMD[1:0] is set to 01 for communication, transmit-only is performed without reception.

When TXMD[1] is set to 1 for communication, receive-only is performed without transmission.

When TXMD[1:0] is set to 01 for communication, a receive buffer full interrupt request cannot be used.

When TXMD[1] is set to 1 for communication, a transmit buffer empty interrupt request cannot be used.

(See section 43.3.6. Communication Operating Mode.)

MSTR bit (SPI Master/Slave Mode Select)

This bit is used to select master mode or slave mode of the SPI. The SPI determines input/output directions of pins RSPCK, MOSI, MISO, and SSL1 to SSL3 according to the MSTR bit setting.

BPEN bit (Synchronization Circuit Bypass Enable)

This bit selects whether to enable or disable the synchronization bypass function. Set this bit to 1 when the bus clock (PCLK) is also used as operation clock (TCLK), see Figure 43.2.

43.2.4 SPCR2 : SPI Control Register 2

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	MOIFE	MOIFV	—	—	SPLP2	SPLP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SPDRC[7:0]								RMSTTG	RMEDTG	—	RMFM[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	RMFM[4:0]	Frame processing count setting in Master Receive only The number of received frames can be adjusted in master receive only. 0x00: This function is not used[*1] 0x01: Automatically stop communication after processing 1 received frame ⋮ 0x1F: Automatically stop communication after processing 31 received frames	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	RMEDTG	End Trigger in Master Receive only 1: Receive End (Writable only when Master Receive only) Reading value is always 0	W
7	RMSTTG	Start Trigger in Master Receive only 1: Receive Start (Writable only when Master Receive only) Reading value is always 0	W
15:8	SPDRC[7:0]	SPI received data ready detect adjustment 0x00: Disable receive data ready detection function 0x01: Performs reception data ready judgment after 1 TCLK ⋮ 0xFF: Performs reception data ready judgment after 255 TCLK	R/W
16	SPLP	SPI Loopback 0: Normal mode 1: Loopback mode (inverted transmit data = receive data)	R/W
17	SPLP2	SPI Loopback 2 0: Normal mode 1: Loopback mode (transmit data = receive data)	R/W

Bit	Symbol	Function	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
20	MOIFV	MOSI Idle Fixed Value 0: The fixed value of MOSI idle = 0. 1: The fixed value of MOSI idle = 1.	R/W
21	MOIFE	MOSI Idle Fixed Value Enable 0: The MOSI output value is the last data of previous transfer. 1: The MOSI output value is the set MOIFV bit value.	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. See SW flow in the section 43.3.12.1. Master Mode Operation.

RMFM[4:0] bit (Frame processing count setting in Master Receive only)

The number of received frames can be adjusted when operating in master receive only. Valid only when the master mode (SPCR.MSTR = 1) and the communication operation mode select bits (SPCR.TXMD [1:0]) are 10b.

Only the start bit in master mode reception automatically stops communication after starts frame processing according to the value set in this bit after reception starts.

If the RMFM [4:0] bits are rewritten while the SPE bit of the SPI control register (SPCR) is 1, subsequent operations are not guaranteed.

RMEDTG bit (End Trigger in Master Receive only)

This bit is used to end reception when master receive only. Valid only when the master mode (SPCR.MSTR = 1) and the communication mode select bits (SPCR.TXMD [1:0]) are 10b.

RMSTTG bit (Start Trigger in Master Receive only)

This bit is used to start reception when master receive only. Valid only when the master mode (SPCR.MSTR = 1) and the communication mode select bits (SPCR.TXMD [1:0]) are 10b.

Writing 1 to this bit during reception is not accepted. Write again after reception is completed.

SPDRC[7:0] bit (SPI received data ready detect adjustment)

The receive data ready detection function can be disabled or, if used, the period until detection can be set from 1 to 255 TCLK.

The value set in the SPDRC [7:0] bits is used to 1 set the SPDRF flag. For details, see the description of SPDRF in section 43.2.9. SPSR : SPI Status Register.

If the set value is changed while the SPE bit is 1, subsequent operations are not guaranteed.

SPLP bit (SPI Loopback)

When the SPLP bit is set to 1, the SPI shuts down the route between the MISO pin and the shift register (when the MSTR bit in the SPI control register is 1) or shuts down the route between the MOSI pin and the shift register, inverts the input route value in the shift register, and then connects the route to the output route (when the MSTR bit in the SPI control register is 0) (loopback mode).

SPLP2 bit (SPI Loopback 2)

When the SPLP2 bit is set to 1, the SPI shuts down the route between the MISO pin and the shift register (when the MSTR bit in the SPI control register is 1) or shuts down the route between MOSI pin and the shift register and then connects the route to the output route without inverting the input route value in the shift register (when the MSTR bit in the SPI control register is 0) (loopback mode). If this bit is set to 1 together with the SPLP bit, setting this bit takes precedence.

MOIFV bit (MOSI Idle Fixed Value)

This bit is used to select the MOSI pin output value during the SSL negation period (including SSL retention period in burst transfer) when the MOIFE bit is 1 in master mode.

If this bit is modified with the SPE bit in the SPI control register (SPCR) set to 1, subsequent operation is not guaranteed.

MOIFE bit (MOSI Idle Fixed Value Enable)

This bit is used for the SPI in master mode to fix the MOSI output value during the SSL negation period (including SSL retention period in burst transfer). When MOIFE bit = 0, the SPI outputs the last data of the previous serial transfer to MOSI during the SSL negation period. When MOIFE bit = 1, the SPI outputs the fixed MOIFV bit value to MOSI.

If this bit is modified with the SPE bit in the SPI control register (SPCR) set to 1, subsequent operation is not guaranteed.

43.2.5 SPCR3 : SPI Control Register 3

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	SPSLN[2:0]			—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SPBR[7:0]								—	—	—	—	SSL3P	SSL2P	SSL1P	SSL0P
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SSL0P	SSL0 Signal Polarity [In the Motorola-SPI case] 0: The SSL0 signal is active low. 1: The SSL0 signal is active high. [In the TI-SSP case] 0: The SSL0 signal is active high. 1: The SSL0 signal is active low.	R/W
1	SSL1P	SSL1 Signal Polarity [In the Motorola-SPI case] 0: The SSL1 signal is active low. 1: The SSL1 signal is active high. [In the TI-SSP case] 0: The SSL1 signal is active high. 1: The SSL1 signal is active low.	R/W
2	SSL2P	SSL2 Signal Polarity [In the Motorola-SPI case] 0: The SSL2 signal is active low. 1: The SSL2 signal is active high. [In the TI-SSP case] 0: The SSL2 signal is active high. 1: The SSL2 signal is active low.	R/W
3	SSL3P	SSL3 Signal Polarity [In the Motorola-SPI case] 0: The SSL3 signal is active low. 1: The SSL3 signal is active high. [In the TI-SSP case] 0: The SSL3 signal is active high. 1: The SSL3 signal is active low.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
15:8	SPBR[7:0]	SPI Bit Rate	R/W
23:16	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
26:24	SPSLN[2:0]	SPI Sequence Length 0 0 0: Sequence Length is 1 (Referenced SPCMDn, n = 0→0→…) 0 0 1: Sequence Length is 2 (Referenced SPCMDn, n = 0→1→0→…) 0 1 0: Sequence Length is 3 (Referenced SPCMDn, n = 0→1→2→0→…) 0 1 1: Sequence Length is 4 (Referenced SPCMDn, n = 0→1→2→3→0→…) 1 0 0: Sequence Length is 5 (Referenced SPCMDn, n = 0→1→2→3→4→0→…) 1 0 1: Sequence Length is 6 (Referenced SPCMDn, n = 0→1→2→3→4→5→0→…) 1 1 0: Sequence Length is 7 (Referenced SPCMDn, n = 0→1→2→3→4→5→6→0→…) 1 1 1: Sequence Length is 8 (Referenced SPCMDn, n = 0→1→2→3→4→5→6→7→0→…)	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

SSLiP bits (SSL Signal Polarity Bits)

These bits are used to specify the polarity of SSL signals. The set SSLiP bit (i = 0 to 3) values indicate the active polarity of SSLi signals.

If any of these SSLiP bits is modified with the SPE bit in the SPI control register (SPCR) set to 1, subsequent operation is not guaranteed.

Note: SSL0 is different from SSL1-SSL3. When slave or multi-master, it functions as an input.

For details, see section 43.3.3.2. Single-master/Single-slave with the MCU as a Slave, and section 43.3.3.5. Multi-master/Multi-slave with the MCU as a Master.

SPBR[7:0] bit (SPI Bit Rate)

The SPI bit rate bits (SPBR) is used to set the bit rate in master mode. If SPBR is modified while the MSTR bit in the SPI control register (SPCR) is 1, subsequent operation is not guaranteed.

When the SPI is used in slave mode, the bit rate depends on the input clock bit rate regardless of the SPCMD.BRDV setting. (Specify a bit rate that meets electrical characteristics.)

The bit rate is determined by a combination of the set SPBR value and the set BRDV[1:0] bits value in the SPI command register (SPCMD0 to SPCMD7).

The bit rate is calculated by the following expression, where n is the set SPBR value (0 to 255) and N is the set BRDV[1:0] bits value (0 to 3).

$$\text{Bit rate} = \frac{f(\text{TCLK})}{2 \times (n+1) \times 2^N}$$

The following table shows an example of correspondence between bit rates and set values of SPBR and BRDV[1:0].

Table 43.5 Corresponding Between Bit Rates and Set Values (Example)

SPBR Value (n)	BRDV Value (N)	Division Ratio	Bit Rate				
			TCLK = 32 MHz	TCLK = 36 MHz	TCLK = 40 MHz	TCLK = 50 MHz	TCLK = 120MHz
0	0	2	16.0 Mbps	18.0 Mbps	20.0 Mbps	25.0 Mbps	60.0Mbps
1	0	4	8.00 Mbps	9.00 Mbps	10.0 Mbps	12.5 Mbps	30.0Mbps
2	0	6	5.33 Mbps	6.00 Mbps	6.67 Mbps	8.33 Mbps	20.0Mbps
3	0	8	4.00 Mbps	4.50 Mbps	5.00 Mbps	6.25 Mbps	15.0Mbps
4	0	10	3.20 Mbps	3.60 Mbps	4.00 Mbps	5.00 Mbps	12.0Mbps
5	0	12	2.67 Mbps	3.00 Mbps	3.33 Mbps	4.16 Mbps	10.0Mbps
5	1	24	1.33 Mbps	1.50 Mbps	1.67 Mbps	2.08 Mbps	5.0Mbps
5	2	48	677 kbps	750 kbps	833 kbps	1.04 Mbps	2.5Mbps
5	3	96	333 kbps	375 kbps	417 kbps	521 kbps	1.25Mbps
255	3	4096	7.81 kbps	8.80 kbps	9.78 kbps	12.2 kbps	29.3kbps

SPSLN[2:0] bit (SPI Sequence Length)

These bits are used to set the sequence length for the SPI in master mode to perform sequence operation. According to the sequence length specified by SPSLN[2:0] bits, the SPI in master mode changes SPI command registers 0 to 7 (SPCMD0 to SPCMD7) to be referenced and the reference sequence. For details, see section 43.3.13.1. Master Mode Operation.

The SPI in slave mode always references SPCMD0.

43.2.6 SPCMDm : SPI Command Register (m = 0 to 7)

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x14 + 0x04 × m

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	SSLA[2:0]			—	—	—	SPB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SCKDEN	SLNDEN	SPNDEN	LSBF	—	—	—	—	SSLKP	—	—	—	BRDV[1:0]		CPOL	CPHA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CPHA	RSPCK Phase 0: Data is sampled at an odd edge and changes at an even edge. 1: Data changes at an odd edge and is sampled at an even edge.	R/W
1	CPOL	RSPCK Polarity 0: RSPCK in idle state is 0. 1: RSPCK in idle state is 1.	R/W
3:2	BRDV[1:0]	Bit Rate Division 0 0: Base bit rate 0 1: Base bit rate divided by 2 1 0: Base bit rate divided by 4 1 1: Base bit rate divided by 8	R/W
6:4	—	These bits are read as 0. The write value should be 0.	R/W
7	SSLKP	SSL Signal Level Hold 0: All SSL signals are negated at the end of transfer. 1: SSL signal level is held after the transfer ends until the next access starts.	R/W
11:8	—	These bits are read as 0. The write value should be 0.	R/W
12	LSBF	SPI LSB First 0: MSB first 1: LSB first	R/W
13	SPNDEN	SPI Next-Access Delay Enable 0: Next-access delay is 1RSPCK + 5TCLK 1: Next-access delay is the set value of the SPI next-access delay (SPDECR.SPNDL).	R/W
14	SLNDEN	SSL Negation Delay Setting Enable 0: [Master] SSL negation delay is 1RSPCK. [Slave in the TI-SSP] SSL negation delay is 1TCLK 1: SSL negation delay is the set value of the slave select negation delay (SPDECR.SLNDL).	R/W

Bit	Symbol	Function	R/W
15	SCKDEN	RSPCK Delay Setting Enable [In the Motorola-SPI case] 0: RSPCK delay is 1 RSPCK. 1: RSPCK delay is the set value of the RSPCK delay (SPDECR.SCKDL). [In the TI-SSP case] 0: RSPCK delay is 0 RSPCK. 1: RSPCK delay is the set value of the RSPCK delay (SPDECR.SCKDL).	R/W
20:16	SPB[4:0]	SPI Data Length 0x00 to 0x02: Setting prohibited 0x03: 4 bits 0x04: 5 bits 0x05: 6 bits ⋮ 0x1E: 31 bits 0x1F: 32 bits	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
26:24	SSLA[2:0]	SSL Signal Assertion 0 0 0: SSL0 0 0 1: SSL1 0 1 0: SSL2 0 1 1: SSL3 Others: Setting prohibited	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The SPI has eight SPI command registers (SPCMD0 to SPCMD7) that are used to set the transfer format of the SPI in master mode. Furthermore, some bits in SPCMD0 are used to set the transfer format of the SPI in slave mode. The SPI in master mode sequentially references SPCMD0 to SPCMD7 according to the setting of the SPSLN[2:0] bits in the SPI Control register 3 (SPCR3), and then performs serial transfer specified in the referenced SPCMD.

SPI set the SPCMD register before setting data to be transmitted by referencing the SPCMD when the transmit buffer is empty (while the next-transfer data has not been set).

The SPCMD referenced by the SPI in master mode is indicated by SPCP[2:0] in the SPI status register (SPSR). If SPCMD0 is modified while the SPI in slave mode is enabled (SPCR.SPE = 1), subsequent operation is not guaranteed.

CPHA bit (RSPCK Phase)

This bit is used to set the RSPCK phase of the SPI in master mode or slave mode. To perform data communication between SPI modules, the same RSPCK phase must be set for both modules.

When SPCR.SPMS = 0 and SPCR.SPFRF = 1 (in TI SSP mode), setting CPHA = 0 is invalid.

CPOL bit (RSPCK Polarity)

This bit is used to set the RSPCK polarity of the SPI in master mode or slave mode. To perform data communication between SPI modules, the same RSPCK polarity must be set for both modules.

BRDV[1:0] bit (Bit Rate Division)

This register is used to determine the bit rate with a combination of the set values of the BRDV[1:0] bits and the SPI bit rate register (SPCR3.SPBR). The set SPBR value determines the base bit rate. The set BRDV[1:0] bits value is used to select undivided, 2-divided, 4-divided, or 8-divided base bit rate. SPCMD0 to SPCMD7 enable setting of different BRDV[1:0] values. This makes it possible to perform serial transfer with a different bit rate for each command.

SSLKP bit (SSL Signal Level Hold)

This bit is used to set whether to hold or negate the SSL signal level of the current command during a period from the SSL negation timing for the current command to the SSL assertion timing for the next command when the SPI in master mode performs serial transfer. Setting this bit to 1 enables burst transfer in SPI operation master mode. For details, see section 43.3.12.1. Master Mode Operation.

To use the SPI in slave mode, set SSLKP bit to 0.

LSBF bit (SPI LSB First)

This bit is used to set the data format of the SPI in master mode or slave mode to MSB first or LSB first.

SPNDEN bit (SPI Next-Access Delay Enable)

This bit is used to set the period (next-access delay) after the SPI in master mode inactivates the SSL signal at the end of serial transfer until it enables SSL signal assertion of the next access. When SPNDEN bit = 0, the SPI sets the next-access delay to 1 RSPCK + 5TCLK. When SPNDEN bit = 1, the SPI inserts the next-access delay in accordance with the SPI next-access delay register (SPDECR.SPNDL) setting.

To use the SPI in slave mode, set SPNDEN bit to 0.

SLNDEN bit (SSL Negation Delay Setting Enable)

[In the Motorola-SPI case]

This bit is used to set the period (SSL negation delay) after the SPI in master mode stops RSPCK oscillation until it inactivates the SSL signal. When SLNDEN bit = 0, the SPI sets the SSL negation delay to 1 RSPCK. When SLNDEN bit = 1, the SPI negates the SSL signal with the RSPCK delay in accordance with the slave select negation delay register (SPDECR.SLNDL) setting.

To use the SPI in slave mode, set SLNDEN bit to 0.

[In the TI-SSP case]

This bit is used to set the period from when the master mode SPI stops RSPCK oscillation to when the OE signal is inactivated, or when the slave mode SPI detects the last edge of RSPCK and then negates the OE signal. When the SLNDEN bit is 0, the SSL negate delay is 1 RSPCK in master mode and 1 TCLK in slave mode. When SLNDEN bit = 1, the SPI negates the SSL signal with the RSPCK delay in accordance with the slave select negation delay register (SPDECR.SLNDL) setting.

When using SPI in slave mode except TI SSP setting, set the SLNDEN bit to 0.

SCKDEN bit (RSPCK Delay Setting Enable)

[In the Motorola-SPI case]

This bit is used to set the period (RSPCK delay) after the SPI in master mode activates the SSL signal until it oscillates RSPCK. When SCKDEN bit = 0, the SPI sets the RSPCK delay to 1 RSPCK. When SCKDEN bit = 1, the SPI starts RSPCK oscillation with the RSPCK delay in accordance with the RSPCK delay register (SPDECR.SCKDL) setting.

To use the SPI in slave mode, set SCKDEN bit to 0.

[In the TI-SSP case]

This bit is used to set the period from the start of assertion of the SSL signal to the RSPCK oscillation (RSPCK delay) and the period of the SSL signal to negation by the SPI in master mode. When SCKDEN bit = 0, the SPI does not set the RSPCK delay. When SCKDEN bit = 1, the SPI starts RSPCK oscillation with the RSPCK delay in accordance with the RSPCK delay register (SPDECR.SPCKDL) setting.

To use the SPI in slave mode, set SCKDEN bit to 0.

SPB[4:0] bit (SPI Data Length)

These bits are used to set the transfer data length of the SPI in master mode or slave mode.

SSLA[2:0] bit (SSL Signal Assertion)

These bits are used to control SSL signal assertion for the SPI in master mode to perform serial transfer. The set SSLA[2:0] bits value controls assertion of the SSL3 to SSL0 signals. The signal polarity when the SSL signal is asserted depends on the set value of the SPI slave select polarity register (SPCR3.SSLiP). When SSLA[2:0] bits are set to 000b in multi-master mode, serial transfer is performed with all SSL signals negated (because SSL0 is input).

To use the SPI in slave mode, set SSLA[2:0] bits to 000b.

43.2.7 SPDCR : SPI Data Control Register

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x40

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SPFC[1:0]		—	—	—	SINV	SPRDTD	—	—	BYSW
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BYSW	Byte Swap Operating Mode Select 0: Byte Swap OFF 1: Byte Swap ON	R/W
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	SPRDTD	SPI Receive Data or Transmit Data Select 0: The SPDR reads the receive buffer. 1: The SPDR reads the transmit buffer	R/W
4	SINV	Serial data invert bit 0: Not invert serial data 1: Invert serial data.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
9:8	SPFC[1:0]	Frame Count 0 0: 1 frame 0 1: 2 frames 1 0: 3 frames 1 1: 4 frames	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The SPI data control register (SPDCR) controls the data format.

If the value set in this register is changed while the SPE bit is 1, subsequent operations are not guaranteed.

BYSW bit (Byte Swap Operating Mode Select)

It is a setting bit, that is to swap a transmit/receive data in byte units. A data after byte swap is different by a data length (setting of SPCMD.SPB[4:0]).

When byte swap, A data length (setting of SPB[4:0]) must be set to 32bit or 16bit. Other case of data length (i.e. 4 to 15, 17 to 31-bit length), byte swap is not guaranteed. For the arrangement of data before and after swapping data lengths of 32 bits and 16 bits, see section 43.3.4.3. Byte Swap Transmission and section 43.3.4.4. Byte Swap Reception.

When the parity function set to valid, the behavior is not guaranteed.

SPRDTD bit (SPI Receive Data or Transmit Data Select)

This bit is used to select receive buffer or transmit buffer from which the SPI data register (SPDR) value is read.

When the transmit buffer is read, the value that was written to SPDR immediately before is read.

SINV bit (Serial data invert bit)

This bit is used to invert transmit data and receive data.

When the SINV bit is set to 1, transmit buffer (SPTX) data is inverted to invert transmit data and receive data, and then the inverted data is stored in the receive buffer (SPRX). The parity bit is the value corresponding to the inverted transmission/reception data.

SPFC[1:0] bit (Frame Count)

Used for the condition to set the CENDF flag in slave receive only mode.

For details on the CENDF flag setting conditions, see section 43.2.9. SPSR : SPI Status Register.

Note that this bit is invalid except in the slave receive only mode.

43.2.8 SPDCR2 : SPI Data Control Register 2

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x44

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TTRG[1:0]		—	—	—	—	—	—	RTRG[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	RTRG[1:0]	Receive FIFO threshold setting 0 0: threshold 0 0 1: threshold 1 1 0: threshold 2 1 1: threshold 3	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	TTRG[1:0]	Transmission FIFO threshold setting 0 0: threshold 0 0 1: threshold 1 1 0: threshold 2 1 1: threshold 3	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

SPI data control register 2 (SPDCR2) controls the FIFO threshold. If the value set in this register is changed while the SPE bit is 1, subsequent operations are not guaranteed.

RTRG[1:0] bit (Receive FIFO threshold setting)

Set the receive FIFO threshold.

When the number of data stored in the receive FIFO > the number of frames set by RTRG[1:0], the receive buffer full flag is set.

TTRG[1:0] bit (Transmission FIFO threshold setting)

Set the transmit FIFO threshold.

When the number of empty stages in the transmit FIFO > the number of frames set in TTRG[1:0], the transmit buffer empty flag is set.

43.2.9 SPSR : SPI Status Register

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x50

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SPRF	CENDF	SPTEF	UDRF	PERF	MODF	IDLNF	OVRF	SPDRF	—	—	—	—	—	—	—
Value after reset:	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	SPECM[2:0]			—	SPCP[2:0]			—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R
10:8	SPCP[2:0]	SPI Command Pointer 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
11	—	This bit is read as 0. The write value should be 0.	R
14:12	SPECM[2:0]	SPI Error Command 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
22:15	—	These bits are read as 0. The write value should be 0.	R
23	SPDRF	SPI Receive Data Ready Flag 0: Receive data ready not detected 1: Receive data ready detected	R
24	OVRF	Overrun Error Flag 0: No overrun error is present. 1: An overrun error is present.	R
25	IDLNF	SPI Idle Flag 0: The SPI is in the idle state. 1: The SPI is in the transfer state.	R
26	MODF	Mode Fault Error Flag 0: Neither mode fault error nor underrun error is present. 1: A mode fault error or underrun error is present.	R
27	PERF	Parity Error Flag 0: No parity error is present. 1: A parity error is present.	R
28	UDRF	Underrun Error Flag 0: When MODF=0, neither mode fault error nor underrun error is present. When MODF=1, a mode fault error is present. 1: When MODF=0, neither mode fault error nor underrun error is present. When MODF=1, an underrun error is present.	R

Bit	Symbol	Function	R/W
29	SPTEF	SPI Transmit Buffer Empty Flag 0: The number of empty stages in the transmit FIFO ≤ the value set in SPDCR2.TTRG 1: The number of empty stages in the transmit FIFO > the value set in SPDCR2.TTRG	R
30	CENDF	Communication End Flag 0: The SPI is not communicating or communicating. 1: The SPI communication completed.	R
31	SPRF	SPI Receive Buffer Full Flag 0: The number of data stored in the receive FIFO ≤ number of frames set by the SPDCR2.RTRG bit. 1: The number of data stored in the receive FIFO > number of frames set by the SPDCR2.RTRG bit.	R

Note: S-TYPE-3, P-TYPE-3

The SPI status register (SPSR) stores flags that indicate SPI's operating status.

SPCP[2:0] bit (SPI Command Pointer)

These bits indicate SPI command registers 0 to 7 (SPCMD0 to SPCMD7) indicated by the current pointer in the SPI sequence control. For details about the SPI sequence control, see section 43.3.13.1. Master Mode Operation.

SPECM[2:0] bit (SPI Error Command)

These bits indicate SPI command registers 0 to 7 (SPCMD0 to SPCMD7) indicated by the command pointer (SPCP[2:0] bits) when an error was detected in the SPI sequence control. The SPI updates the SPECM[2:0] bits value only when an error is detected. When no error is present (OVRF, MODF, and PERF flags in SPSR are 0), the SPECM[2:0] bits value has no meaning. For the SPI's error detection function, see section 43.3.10. Error Detection. For the SPI's sequence control, see section 43.3.13.1. Master Mode Operation.

SPDRF bit (SPI Receive Data Ready Flag)

During communication (SPCR.SPE = 1), a certain period of time has elapsed while the number of data stored in the reception FIFO ≤ the reception FIFO threshold.

This bit is set to 0 when the reception operation is not performed (SPCR.TXMD[1:0] = 01b).

[Setting condition]

All the following two conditions are met.

- SPCR2.SPDRC[7:0] ≠ 0x00.
- After the receive FIFO has been written, when the number of data stored in the receive FIFO ≤ the receive FIFO threshold and the value set by SPDRC[7:0] has elapsed

[Clearing condition]

- When 1 is written to the SPSRC.SPDRFC bit.

OVRF bit (Overrun Error Flag)

This flag indicates whether an overrun error is present. When the RSPCK clock auto-stop function is enabled (SPCR.SCKASE = 1) in master mode (SPCR.MSTR = 1), no overrun error occurs and, therefore, this flag is not set to 1. For details, see section 43.3.10.1. Overrun Errors.

[Setting condition]

When serial transfer is completed in one of the following two conditions with data stored in the receive FIFO for the number of FIFO stages.

- SPCR.TXMD[1:0] = 00b. (transmit-receive mode)
- SPCR.TXMD[1:0] = 10b. (receive only)

[Clearing condition]

- When 1 is written to the SPSRC.OVRFC bit.

IDLNF bit (SPI Idle Flag)

This flag indicates transfer status of the SPI.

[Setting condition]

[Transmit-Receive, Transmit-only Master mode]

- None of the clearing conditions (Transmit-Receive / Transmit-only in Master mode) below is met.

[Receive-only Master mode]

- When 1 is written to RMSTTG of SPCR2.

[Slave mode]

- The SPE bit in SPCR is 1 (SPI function enabled).

[Clearing conditions]

Communication status: 1-(1) to (6) * For details of communication status, see Table 43.4.

[Transmit-Receive, Transmit-only Master mode]

Any of the following two conditions is met.

- The SPE bit in SPCR is 0 (SPI initialization).
- All the following three conditions are met.
  - The next transfer data is not set in the transmission buffer (SPTXn, n = 0 to 3)
  - The SPCP bits in SPSR are 000b (at the beginning of sequence control).
  - The operation completed by the next access delay (the master main state machine has transitioned to the idle state)

[Receive-only Master mode]

Communication status: 1-(7) to (9)

Any of the following two conditions is met.

- The SPE bit in SPCR is 0 (SPI initialization).
- Any of the following 3 conditions is met.
  When RMFM[4:0] = 0x00, after writing 1 to RMEDTG, the operation completed by the next access delay (the master main state machine has transitioned to the idle state)
  When RMFM[4:0] ≠ 0x00, after writing 1 to RMEDTG, the operation completed by the next access delay (the master main state machine has transitioned to the idle state)
  When RMFM[4:0] ≠ 0x00, the operation completed by the next access delay after processing is completed for the number of received frames set in RMFM[4:0] (the master main state machine has transitioned to the idle state)

[Slave mode]

Communication status: 0-(1) to (9)

- The SPE bit in SPCR is 0 (SPI initialization).

MODF bit (Mode Fault Error Flag)

This flag indicates whether a mode fault error or an underrun error is present. The UDRF flag allows you to see which error (mode fault error or underrun error) has occurred.

[Setting condition]

[Multi-master mode]

- The SSL0 pin input level becomes active level while the SPCR.MSTR bit = 1 (master mode) and the SPCR.MODFEN bit = 1 (mode fault error detection enabled), and then the SPI has detected a mode fault error.

[Slave, Motorola-SPI mode]

Any of the following two conditions is met.

- The SSL0 pin is negated before the RSPCK cycles necessary for data transfer end while the SPCR.MSTR bit = 0 (slave mode), SPCR.SPFRF bit = 0 (Motorola-SPI) and the SPCR.MODFEN bit = 1 (mode fault error detection enabled), and then the SPI has detected a mode fault error.
- Serial transfer is started before transmit data output becomes ready while the SPCR.SPE bit = 1 (SPI function enabled), and then the SPI has detected an underrun error.

[Slave, TI-SSP mode]

Any of the following two conditions is met.

- The SSL0 pin is asserted before the RSPCK cycles necessary for data transfer end while the SPCR.MSTR bit = 0 (slave mode), SPCR.SPFRF bit = 1 (TI-SSP) and the SPCR.MODFEN bit = 1 (mode fault error detection enabled), and then the SPI has detected a mode fault error.
- Serial transfer is started before transmit data output becomes ready while the SPCR.SPE bit = 1 (SPI function enabled), and then the SPI has detected an underrun error.

The SSL0 signal active level depends on the SPCR3.SSLiP bits (SSL signal polarity bits).

[Clearing condition]

- When 1 is written to the SPSRC.MODFC bit.

PERF bit (Parity Error Flag)

This flag indicates whether a parity error is present.

[Setting condition]

When the serial transfer ends and a parity error is detected with the SPPE bit of SPCR set to 1 under any of the following 2 conditions.

- SPCR.TXMD[1:0] = 00b. (transmit-receive master mode or transmit-receive slave mode)
- SPCR.TXMD[1:0] = 10b. (receive-only master mode or receive-only slave mode)

[Clearing condition]

- When 1 is written to the SPSRC.PERFC bit.

UDRF bit (Underrun Error Flag)

This flag indicates that a mode fault error or an underrun error is present.

[Setting condition]

- Serial transfer is started before transmit data output becomes ready while the SPCR.MSTR bit = 0 and the SPCR.TXMD[1:0] bit = 00b or 01b (transmit-receive slave mode or transmit-only slave mode) and the SPCR.SPE bit = 1 (SPI function enabled), and then the SPI has detected an underrun error.

[Clearing condition]

- When 1 is written to the SPSRC.UDRFC bit.

SPTEF bit (SPI Transmit Buffer Empty Flag)

This flag indicates the transmit buffer (SPTX) status in the SPI data register (SPDR).

[Setting condition]

Any of the following 3 conditions is met.

- The SPE bit is set to 0 (SPI initialization).
- When the number of empty transmission FIFO stages > the value set in SPDCR2.TTRG[1:0].
- When 1 is written to SPFCR.SPFRST.

[Clearing condition]

Any of the following two conditions is met.

- At the time of final access when transmission data is written to SPDR (SPTXn, n = 0 to 3) in one processing routine using DTC / DMAC.
- When 1 is written to the SPSRC.SPTEFC bit.

Writing a value to the SPDR register is enabled only while the SPTEF flag = 1. If a value is written to the SPDR register while the SPTEF flag = 0, transmit buffer data is not updated.

CENDF bit (Communication End Flag)

This flag indicates communication end status of SPI. It turns 1 at communication end and turns 0 at starting next communication.

[Setting condition]

[Transmit-Receive / Transmit-only Master mode]

Communication status: 1-(1) to (6) * For details of communication status, see Table 43.4.

The following 3 conditions are met.

- The next transfer data is not set in the transmission buffer (SPTXn, n = 0 to 3)
- The SPSR.SPCP[2:0] are 000b. (It means the head of the sequential control.)
- The operation completed by the next access delay (the master main state machine has transitioned to the idle state)

[Receive-only Master mode]

Communication status: 1-(7) to (9)

Any of the following 3 conditions is met.

- When RMFM[4:0] = 0x00, after writing 1 to RMEDTG, the operation completed by the next access delay (the master main state machine has transitioned to the idle state)
- When RMFM[4:0] ≠ 0x00, after writing 1 to RMEDTG, the operation completed by the next access delay (the master main state machine has transitioned to the idle state)
- When RMFM[4:0] ≠ 0x00, the operation completed by the next access delay after processing is completed for the number of received frames set in RMFM[4:0] (the master main state machine has transitioned to the idle state)

[Transmit-receive / transmit-only slave, Motorola-SPI mode at SPI serial communication (4-wire: the SPCR.SPMS bit is 0)]

Communication status: 0-(1), (4)

The following 3 conditions are met.

- The next transfer data is not set in the transmission buffer
- The transmission shift register is empty. (It means SPI does not do serial transfer.)
- SSL0 was negated.

[Transmit-receive / transmit-only slave, TI-SSP mode at SPI serial communication (4-wire: the SPCR.SPMS bit is 0)]

Communication status: 0-(2), (5)

The following 3 conditions are met.

- The next transfer data is not set in the transmission buffer
- The transmission shift register is empty. (It means SPI does not do serial transfer.)
- When the SSL0 negate delay is completed.

[Transmit-receive / transmit only slave mode at clock synchronous (3-wire: the SPCR.SPMS bit is 1)]

Communication status: 0-(3), (6)

The following 3 conditions are met.

- The next transfer data is not set in the transmission buffer
- The transmission shift register is empty. (It means SPI does not do serial transfer.)
- The last even edge of RSPCK of the frame was detected. (When the SPCMD.CPHA bit is 1.)

[Receive only slave, Motorola-SPI mode at SPI serial communication (4-wire: the SPCR.SPMS bit is 0)]

Communication status: 0-(7)

The following condition is met.

- SSL0 input was negated after getting frames for SPDCR.SPFC set value in the receive buffer.

[Receive only slave, TI-SSP mode at SPI serial communication (4-wire: the SPCR.SPMS bit is 0)]

Communication status: 0-(8)

The following condition is met.

- SSL0 negate delay is completed after getting frames for SPDCR.SPFC set value in the receive buffer.

[Receive only slave mode at clock synchronous (3-wire: the SPCR.SPMS bit is 1)]

Communication status: 0-(9)

The following condition is met.

- The last even edge of RSPCK of the Last frame received for SPFC sets value. (When the SPCMD.CPHA bit is 1.)

[Clearing condition]

[Transmit-Receive / Transmit-only Master mode]

Communication status: 1-(1) to (6)

Any of the following 2 conditions is met.

- The next transmit data was written to the transmit buffer (SPTX).
- When 1 is written to the SPSRC.CENDFC bit.

[Receive -only Master mode]

Communication status: 1-(7) to (9)

Any of the following 2 conditions is met.

- When 1 is written to the SPCR2.RMSTTG bit with SPE = 1.
- When 1 is written to the SPSRC.CENDFC bit.

[Transmit-receive / transmit only slave mode]

Communication status: 0-(1) to (6)

Satisfy one of following.

- The next transmit data was written to the transmit buffer (SPTX).
- When 1 is written to the SPSRC.CENDFC bit.

[Receive only slave mode at SPI serial communication (4-wire: the SPCR.SPMS bit is 0)]

Communication status: 0-(7) to (8)

Satisfy one of following.

- SSL0 assertion of next data was detected.
- When 1 is written to the SPSRC.CENDFC bit.

[Receive only slave mode at clock synchronous (3-wire: the SPCR.SPMS bit is 1)]

Communication status: 0-(9)

Satisfy one of following.

- The first edge of RSPCK of the next data was detected.
- When 1 is written to the SPSRC.CENDFC bit.

SPRF bit (SPI Receive Buffer Full Flag)

This flag indicates the receive buffer (SPRX) status in the SPI data register (SPDR).

[Setting condition]

When the number of data stored in the receive FIFO > the number of frames set in the SPDCR2.RTRG[1:0] bits in Transmit-Receive, receive-only mode. However, the SPRF flag does not change from 0 to 1 while the OVRF flag = 1. (See section 43.3.10. Error Detection.)

[Clearing condition]

Any of the following 3 conditions is met.

- At the last access when read data is read from SPDR (SPRXn, n = 0 to 3) in one processing routine using DTC / DMAC
- When 1 is written to the SPSRC.SPRFC bit
- When 1 is written to the SPFCR.SPFRST bit

43.2.10 SPTFSR : SPI Transfer FIFO Status Register

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x58

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	TFDN[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
2:0	TFDN[2:0]	Transmit FIFO data empty stage number 0 0 0: Number of empty stages 0 ⋮ 1 0 0: Number of empty stages 4	R
31:3	—	These bits are read as 0. The write value should be 0.	R

TFDN[2:0] bit (Transmit FIFO data empty stage number)

Displays the number of empty transmission FIFO stages. By clearing the SPCR.SPE bit, TFDN[2:0] will be the initial value after reset (= all empty).

43.2.11 SPRFSR : SPI Receive FIFO Status Register

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x5C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	RFDN[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	RFDN[2:0]	Receive FIFO data store stage number 0 0 0: Number of store stages 0 ⋮ 1 0 0: Number of store stages 4	R
31:3	—	These bits are read as 0. The write value should be 0.	R

Note: S-TYPE-3, P-TYPE-3

RFDN[2:0] bit (Receive FIFO data store stage number)

Displays the number of stores receive FIFO stages. RFDN [2:0] is cleared by clearing the SPCR.SPE bit.

43.2.12 SPPSR : SPI Polling Register

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x60

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPEPS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPEPS	SPI Polling Status 0: SPCR.SPE is 0 1: SPCR.SPE is 1	R
31:1	—	These bits are read as 0. The write value should be 0.	R

SPEPS bit (SPI Polling Status)

This bit indicates status of SPCR.SPE bit after synchronization from bus clock (PCLK) to operation clock (TCLK).

43.2.13 SPSRC : SPI Status Clear Register

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x68

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SPRFC	CENDFC	SPTEFC	UDRFC	PERFC	MODFC	—	OVRFC	SPDRFC	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
22:0	—	The write value should be 0.	W
23	SPDRFC	SPI Receive Data Ready Flag Clear By writing 1, the SPI Receive Data Ready Flag can be cleared. Reading value is always 0.	W

Bit	Symbol	Function	R/W
24	OVRFC	Overrun Error Flag Clear By writing 1, the Overrun Error Flag can be cleared. Reading value is always 0.	W
25	—	The write value should be 0.	W
26	MODFC	Mode Fault Error Flag Clear By writing 1, the Mode Fault Error Flag can be cleared. Reading value is always 0.	W*1
27	PERFC	Parity Error Flag Clear By writing 1, the Parity Error Flag can be cleared. Reading value is always 0.	W
28	UDRFC	Underrun Error Flag Clear By writing 1, the Underrun Error Flag can be cleared. Reading value is always 0.	W*2
29	SPTEFC	SPI Transmit Buffer Empty Flag Clear By writing 1, the SPI Transmit Buffer Empty Flag can be cleared. Reading value is always 0.	W
30	CENDFC	Communication End Flag Clear By writing 1, the Communication End Flag can be cleared. Reading value is always 0.	W
31	SPRFC	SPI Receive Buffer Full Flag Clear By writing 1, the SPI Receive Buffer Full Flag can be cleared. Reading value is always 0.	W

Note: S-TYPE-3, P-TYPE-3
Note 1. Before setting MODFC and UDRFC, make sure that SPSR.MODF and UDRF are set to 1.
Note 2. When clearing the UDRF flag, clear the MODF flag at the same time (MODFC = 1).

The SPI status clear register (SPSRC) is a register that clears the status flag (SPSR) that indicates the operating status of SPI.

43.2.14 SPFCR : SPI FIFO Clear Register

Base address: SPIn_B = 0x4035_C000 + 0x0100 × n (n = 0, 1)
SPIn_B_NS = 0x5035_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x6C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPFRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPFRST	SPI FIFO clear By writing 1, the pointer in the FIFO and the stored data are initialized. Reading value is always 0.	W
31:1	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The FIFO clear register (SPFCR) is used to clear the FIFO.

If SPFCR is rewritten while the SPE bit of the SPI control register (SPCR) is 1, subsequent operations are not guaranteed.

SPFRST bit (SPI FIFO clear)

Initializing the pointer and stored data in the transmit / receive FIFO by writing 1

43.3 Operation

In this section, the serial transfer period refers to the period from the beginning of driving valid data to the fetching of the final valid data.

43.3.1 Overview of SPI Operation

The SPI is capable of synchronous serial transfers in the following modes:

- Slave mode (SPI operation)
- Single-master mode (SPI operation)
- Multi-master mode (SPI operation)
- Slave mode (clock synchronous operation)
- Master mode (clock synchronous operation)

The SPI mode can be selected by using the MSTR, MODFEN, SPMS, and SPFRF bits in SPCR. Table 43.6 lists the relationship between SPI modes and SPCR settings, and a description of each mode.

Table 43.6 Relationship between SPCR settings and SPI modes (1 of 2)

Mode	Slave (SPI operation)	Single-master (SPI operation)	Multi-master (SPI operation)	Slave (clock synchronous operation)	Master (clock synchronous operation)
MSTR bit setting	0	1	1	0	1
MODFEN bit setting	0 or 1	0	1	0	0
SPMS bit setting	0	0	0	1	1
SPFRF bit setting	valid	valid	valid	Invalid	Invalid
RSPCKn pins	Input	Output	Output/Hi-Z	Input	Output
MOSIn pin	Input	Output	Output/Hi-Z	Input	Output
MISOn pin	Output/Hi-Z	Input	Input	Output	Input
SSLn0 pins	Input	Output	Input	Hi-Z*1	Hi-Z*1
SSLn1 to SSLn3 pins	Hi-Z*1	Output	Output/Hi-Z	Hi-Z*1	Hi-Z*1
SSL polarity change function	Supported	Supported	Supported	—	—
Max transfer rate	TCLK/2	TCLK/2	TCLK/2	TCLK/2	TCLK/2
Clock source	RSPCK input	On-chip baud rate generator	On-chip baud rate generator	RSPCK input	On-chip baud rate generator
Clock polarity	Two				
Clock phase	Two*6	Two*6	Two*6	One (CPHA = 1)	Two
Transfer data length	4 to 32 bits				
Burst transfer	Possible (CPHA = 1)	Possible (CPHA = 0, 1)	Possible (CPHA = 0, 1)	—	—
RSPCK delay control	Not supported	Supported	Supported	Not supported	Supported
SSL negation delay control	Not supported*7	Supported	Supported	Not supported	Supported
Next-access delay control	Not supported	Supported	Supported	Not supported	Supported
Transfer trigger	SSL input active or RSPCK oscillation	Write to transmit buffer on generation of transmit buffer empty interrupt request (SPTEF = 1)	Write to transmit buffer on generation of transmit buffer empty interrupt request (SPTEF = 1)	RSPCK oscillation	Write to transmit buffer on generation of transmit buffer empty interrupt request (SPTEF = 1)
Sequence control	Not supported	Supported	Supported	Not supported	Supported
Transmit buffer empty detection	Supported*5				
Receive buffer full detection	Supported*2				
Overrun error detection	Supported*2	Supported*2 *4	Supported*2 *4	Supported*2	Supported*2

Table 43.6 Relationship between SPCR settings and SPI modes (2 of 2)

Mode	Slave (SPI operation)	Single-master (SPI operation)	Multi-master (SPI operation)	Slave (clock synchronous operation)	Master (clock synchronous operation)
Parity error detection	Supported*3 *2				
Mode fault error detection	Supported (MODFEN = 1)	Not supported	Supported	Not supported	Not supported
Underrun error detection	Supported*5	Not supported	Not supported	Supported*5	Not supported

Note 1. This function is not supported in this mode.
Note 2. When SPI is transmit-master mode or transmit-slave mode (see Table 43.4), detection of receive buffer full, overrun error, and parity error are not performed.
Note 3. When the SPCR.SPPE bit is 0, parity error detection is not performed.
Note 4. When the SPCR.SCKASE bit is 1, overrun error detection does not proceed.
Note 5. When SPI is receive only slave mode, none of transmit buffer empty and underrun error is detected.
Note 6. CPHA = 0 is invalid in TI SSP mode. (If it is set, the operation is the same as when CPHA = 1.)
Note 7. Supported only in TI SSP mode.

43.3.2 Controlling the SPI Pins

Based on the settings of the MSTR, MODFEN, and SPMS bits in SPCR and the PmnPFS.NCODR bit for I/O Ports, the SPI can switch pin states. Table 43.7 lists the relationship between pin states and bit settings. Setting the PmnPFS.NCODR bit for an I/O port to 0 selects the CMOS output. Setting it to 1 selects the open-drain output. The I/O port settings must follow this relationship.

Table 43.7 Relationship between pin states and bit settings

Mode	Pin	Pin state*2	
		PmnPFS.NCODR bit for I/O ports = 0	PmnPFS.NCODR bit for I/O ports = 1
Single-master mode (SPI operation) (MSTR = 1, MODFEN = 0, SPMS = 0)	RSPCKn	CMOS output	Open-drain output
	SSLn0 to SSLn3	CMOS output	Open-drain output
	MOSIn	CMOS output	Open-drain output
	MISOn	Input	Input
Multi-master mode (SPI operation) (MSTR = 1, MODFEN = 1, SPMS = 0)	RSPCKn*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	SSLn0	Input	Input
	SSLn1 to SSLn3*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MOSIn*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MISOn	Input	Input
Slave mode (SPI operation) (MSTR = 0, SPMS = 0)	RSPCKn	Input	Input
	SSLn0	Input	Input
	SSLn1 to SSLn3*5	Hi-Z*1	Hi-Z*1
	MOSIn	Input	Input
	MISOn*4	CMOS output/Hi-Z	Open-drain output/Hi-Z
Master mode (clock synchronous operation) (MSTR = 1, MODFEN = 0, SPMS = 1)	RSPCKn	CMOS output	Open-drain output
	SSLn0 to SSLn3*5	Hi-Z*1	Hi-Z*1
	MOSIn	CMOS output	Open-drain output
	MISOn	Input	Input
Slave mode (clock synchronous operation) (MSTR = 0, SPMS = 1)	RSPCKn	Input	Input
	SSLn0 to SSLn3*5	Hi-Z*1	Hi-Z*1
	MOSIn	Input	Input
	MISOn	CMOS output	Open-drain output

Note 1. This function is not supported in this mode.
Note 2. SPI settings are not reflected in multiplexed pins for which the SPI function is not selected.
Note 3. Motorola-SPI: When SSLn0 is at the active level, the pin state is Hi-Z. Whether or not the input signal is at the active level determines the setting of the SPCR3.SSL0P bit.
TI-SSP: From when SSL0 is at the active level until communication is completed, the pin state is Hi-Z under the condition SPCR.SPE = 1.
Note 4. Motorola-SPI: When SSLn0 is at the non-active level or the SPCR.SPE bit is 0, the pin state is Hi-Z. Whether or not the input signal is at the active level determines the setting of the SPCR3.SSL0P bit.
TI-SSP: When SSL0 is except the communication period or when the SPE bit of SPCR is 0 (assertion after SPE = 1 and communication is completed), the pin status changes to Hi-Z.
Note 5. These pins are available for use as I/O port pins.

The SPI in single-master mode (SPI operation) or multi-master mode (SPI operation) determines the MOSI signal values during the SSL negation period (including the SSL retention period during a burst transfer) based on the MOIFE and MOIFV bit settings in SPCR2, as listed in Table 43.8.

Table 43.8 MOSI signal value determination during SSL negation

MOIFE bit	MOIFV bit	MOSIn signal value during SSL negation
0	0, 1	Final data from previous transfer
1	0	Low
1	1	High

43.3.3 SPI System Configuration Examples

This configuration example describes that 0 level of SSL0n signals is active level.

When connecting and using in a multi-slave or multi-master mode, the transfer format of the connected device should be unified to either Motorola-SPI or TI-SSP.

43.3.3.1 Single-master/Single-slave with the MCU as a Master

Figure 43.6 shows a single-master/single-slave SPI system configuration example where the MCU is used as a master. In the single-master/single-slave configuration, the SSLni outputs of the MCU (master) are not used. The SSL input of the SPI slave is fixed to the low level, and the SPI slave is maintained in the selected state.*1

Note 1. In the transfer format configured when the SPCMDm.CPHA bit is 0, the SSL signal for some slave devices cannot be fixed to an active level. In this case, always connect the SSLni output of the MCU to the SSL input of the slave device.

The MCU (master) drives the RSPCKn and MOSIn signals. The SPI slave drives the MISO signal.

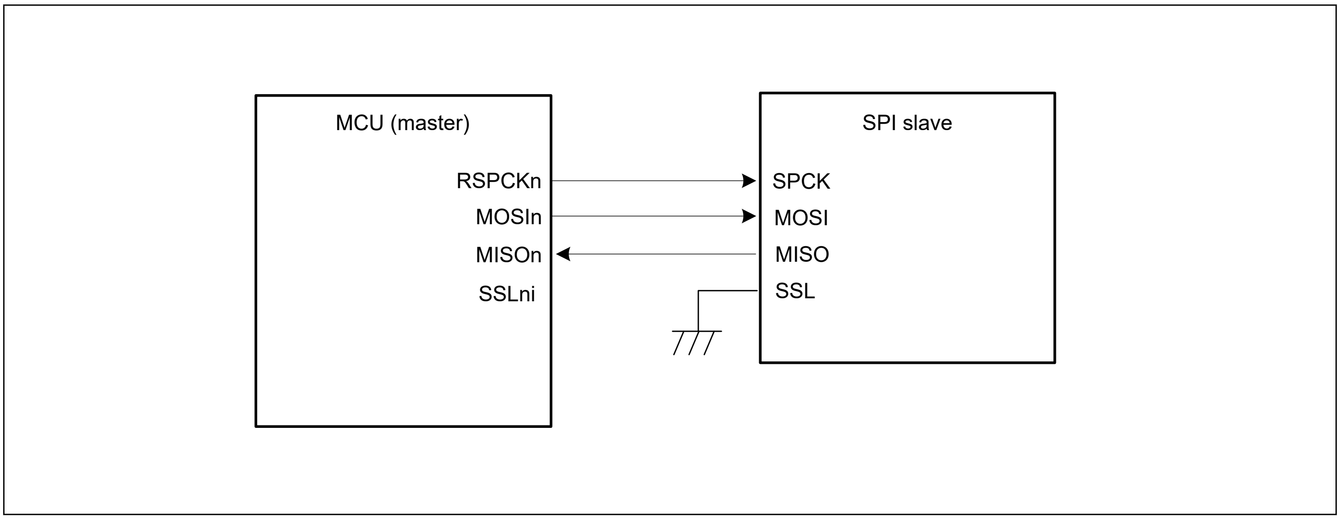


Figure 43.6 Single-master/single-slave configuration example with the MCU as a master

43.3.3.2 Single-master/Single-slave with the MCU as a Slave

Figure 43.7 shows a single-master/single-slave SPI system configuration example where the MCU is used as a slave. When the MCU operates as a slave, the SSLn0 pin is used as SSL input. The SPI master drives the RSPCK and MOSI signals. The MCU (slave) drives the MISOn signal.*1

Note 1. When SSLn0 is at a non-active level, the pin state is Hi-Z.

In the single-slave configuration when the SPCMDm.CPHA bit is set to 1, the SPCR.SPFRF bit is set to 0, and SPCR.SPMS is set to 0, the SSLn0 input of the MCU (slave) is fixed to the low level and the MCU (slave) is maintained in the selected state. This enables serial transfer execution (Figure 43.8). However, the communication end interrupt does not output when SSL0 input is fixed as Figure 43.8.

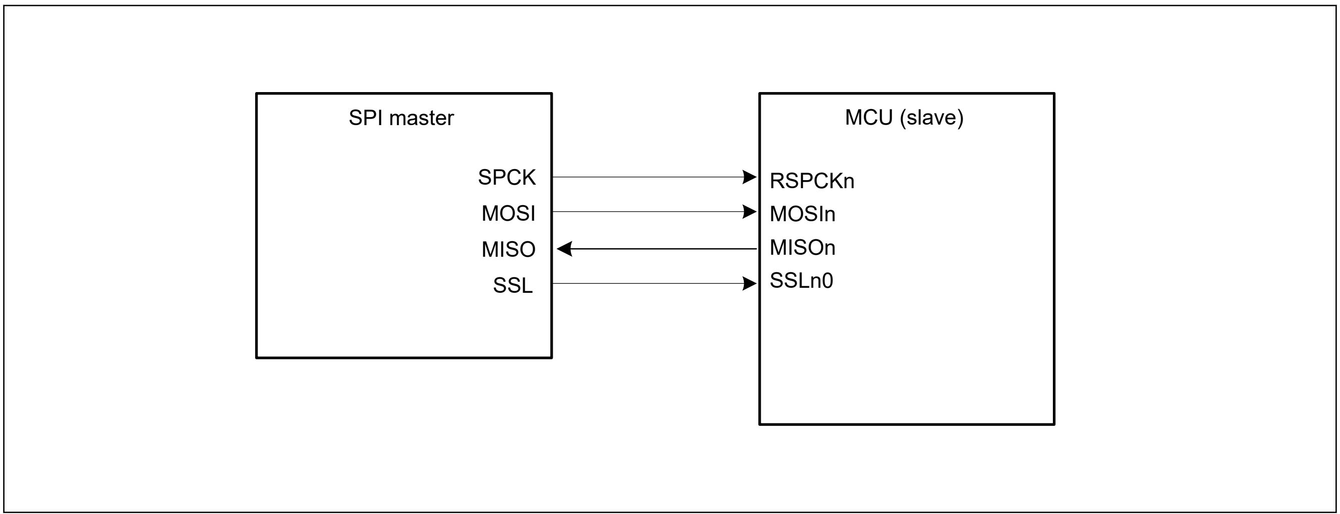


Figure 43.7 Single-master/single-slave configuration example with the MCU as a slave and CPHA = 0

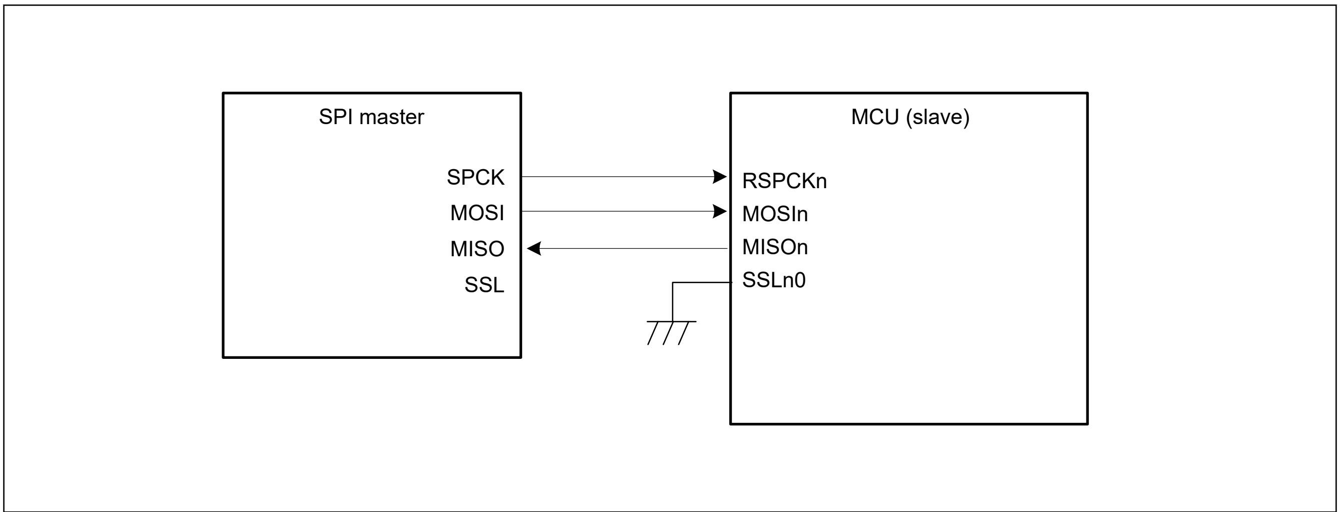


Figure 43.8 Single-master/single-slave configuration example with the MCU as a slave and CPHA = 1

43.3.3.3 Single-master/Multi-slave with the MCU as a Master

Figure 43.9 shows a single-master/multi-slave SPI system configuration example where the MCU is used as a master. In this example, the SPI system includes the MCU (master) and four slaves (SPI slave 0 to SPI slave 3).

The RSPCKn and MOSIn outputs of the MCU (master) are connected to the RSPCK and MOSI inputs of SPI slaves 0 to 3. The MISO outputs of SPI slaves 0 to 3 are all connected to the MISOn input of the MCU (master). The SSLn0 to SSLn3 outputs of the MCU (master) are connected to the SSL inputs of SPI slaves 0 to 3, respectively.

The MCU (master) drives the RSPCKn, MOSIn, and SSLn0 to SSLn3 signals. Out of the SPI slaves 0 to 3, the slave that receives low-level input into the SSL input drives the MISO signal.

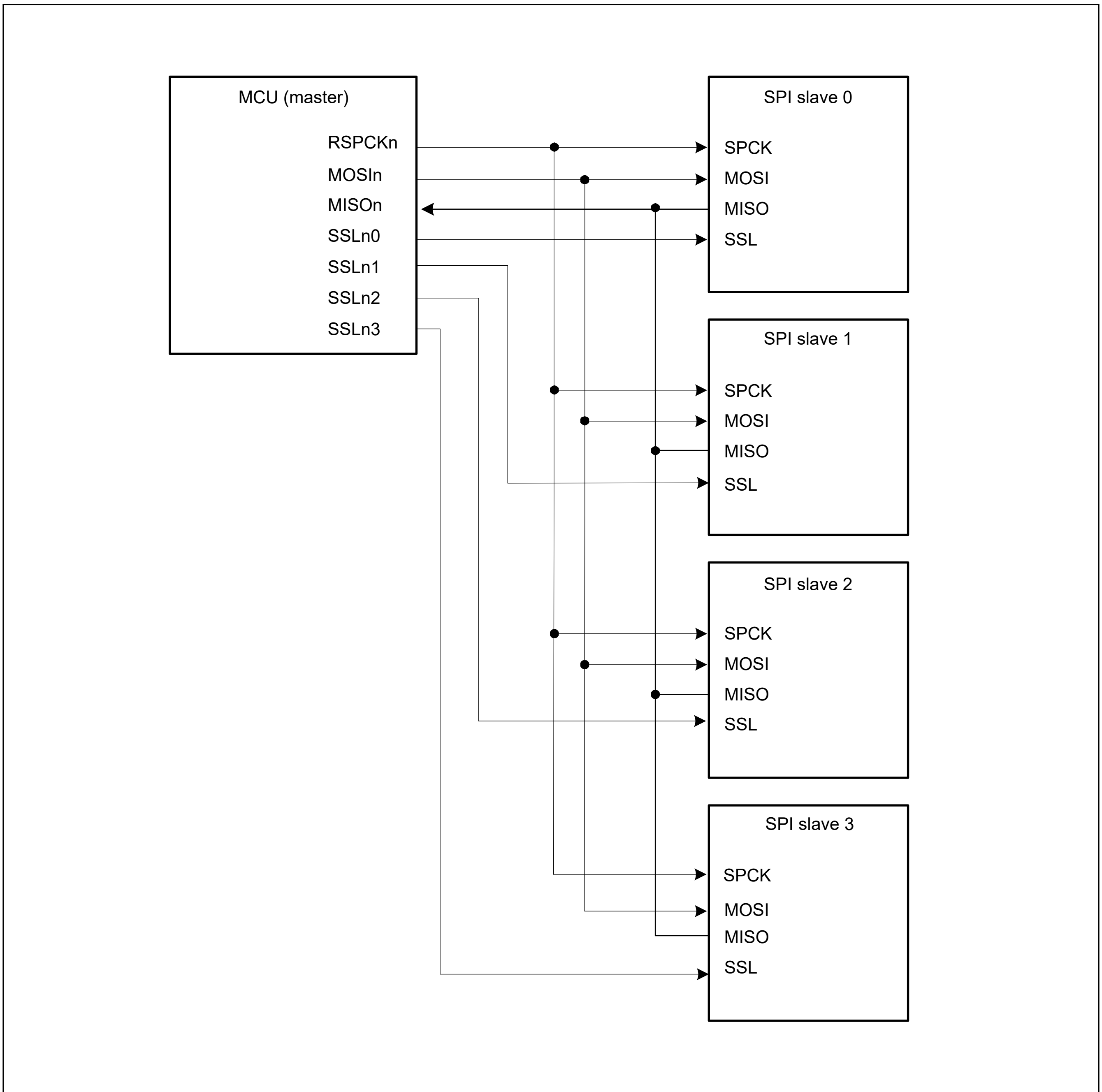


Figure 43.9 Single-master/multi-slave configuration example with the MCU as a master

43.3.3.4 Single-master/Multi-slave with the MCU as a Slave

Figure 43.10 shows a single-master/multi-slave SPI system configuration example where the MCU is used as a slave. In this example, the SPI system includes an SPI master and two MCUs (slaves X and Y).

The SPCK and MOSI outputs of the SPI master are connected to the RSPCKn and MOSIn inputs of the MCUs (slaves X and Y). The MISOn outputs of the MCUs (slaves X and Y) are all connected to the MISO input of the SPI master. The SSLX and SSLY outputs of the SPI master are connected to the SSLn0 inputs of the MCUs (slaves X and Y, respectively).

The SPI master drives the SPCK, MOSI, SSLX, and SSLY signals. Of the MCUs (slaves X and Y), the slave that receives low-level input into the SSLn0 input drives the MISOn signal.

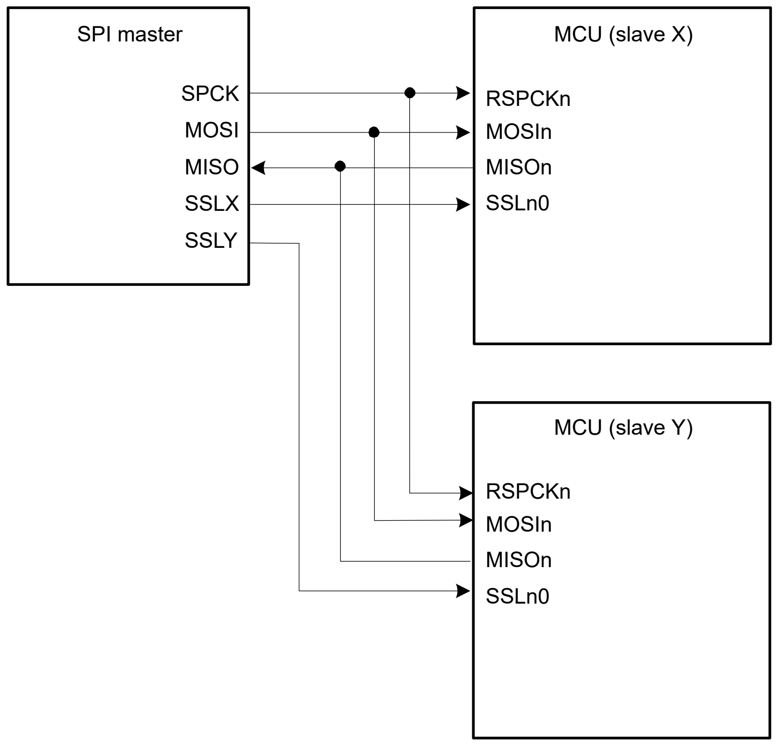


Figure 43.10 Single-master/multi-slave configuration example with the MCU as a slave

43.3.3.5 Multi-master/Multi-slave with the MCU as a Master

Figure 43.11 shows a multi-master/multi-slave SPI system configuration example where the MCU is used as a master. In this example, the SPI system includes two MCUs (masters X and Y) and two SPI slaves (SPI slaves 1 and 2).

The RSPCKn and MOSIn outputs of the MCUs (masters X and Y) are connected to the RSPCK and MOSI inputs of SPI slaves 1 and 2. The MISO outputs of SPI slaves 1 and 2 are connected to the MISOn inputs of the MCUs (masters X and Y). Any generic port Y output from the MCU (master X) is connected to the SSLn0 input of the MCU (master Y). Any generic port X output of the MCU (master Y) is connected to the SSLn0 input of the MCU (master X). The SSLn1 and SSLn2 outputs of the MCUs (masters X and Y) are connected to the SSL inputs of the SPI slaves 1 and 2. In this configuration example, because the system can be comprised solely of SSLn0 input, and SSLn1 and SSLn2 outputs for slave connections, the SSLn3 output of the MCU is not required.

The MCU drives the RSPCKn, MOSIn, SSLn1, and SSLn2 signals when the SSLn0 input level is high. When the SSLn0 input level is low, the MCU detects a mode fault error, sets RSPCKn, MOSIn, SSLn1, and SSLn2 to Hi-Z, and releases the SPI bus directly to the other master. Of the SPI slaves 1 and 2, the slave that receives low-level input into the SSL input drives the MISO signal.

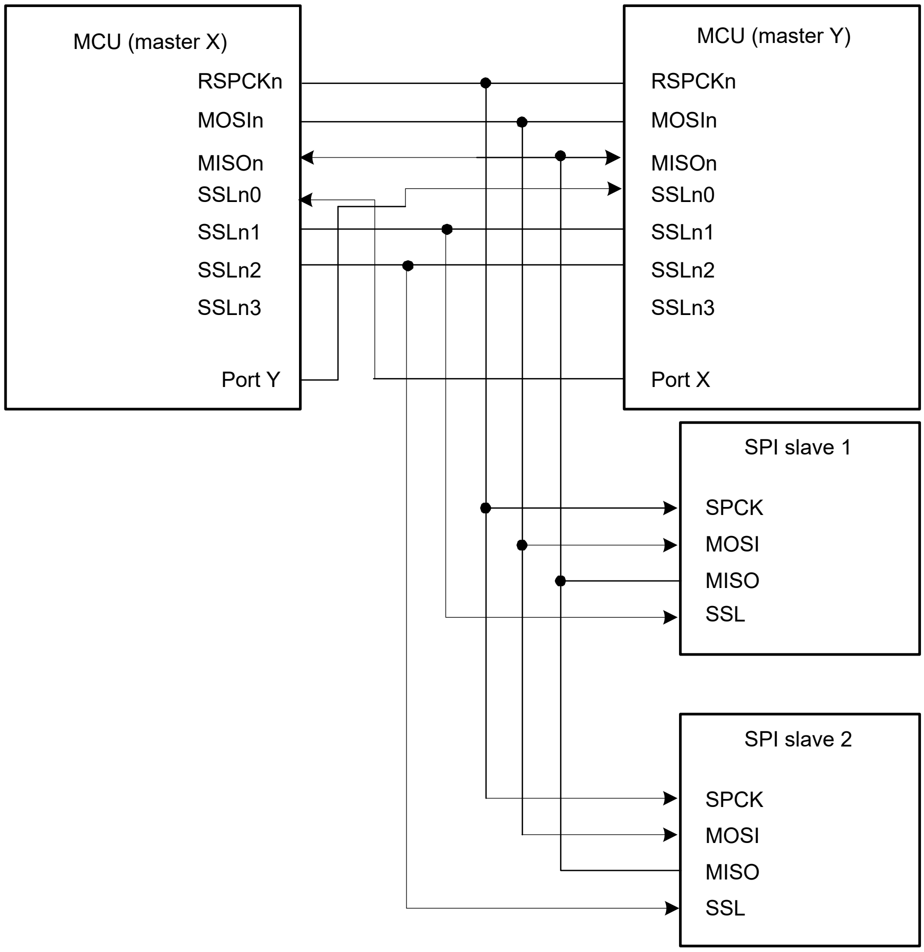


Figure 43.11 Multi-master/multi-slave configuration example with the MCU as a master

When setting TI SSP, enter the following levels for port X and port Y.

- Start of communication: the value of SPCR3.SSL0P of the other master.
- End of communication: the inverted value of SPCR3.SSL0P of the other master.

43.3.3.6 Master and Slave in Clock Synchronous Mode with the MCU Configured as a Master

Figure 43.12 shows a master and slave in clock synchronous mode configuration example where the MCU is used as a master. In this configuration, SSLni of the MCU (master) are not used.

The MCU (master) drives the RSPCKn and MOSIn signals. The SPI slave drives the MISO signal.

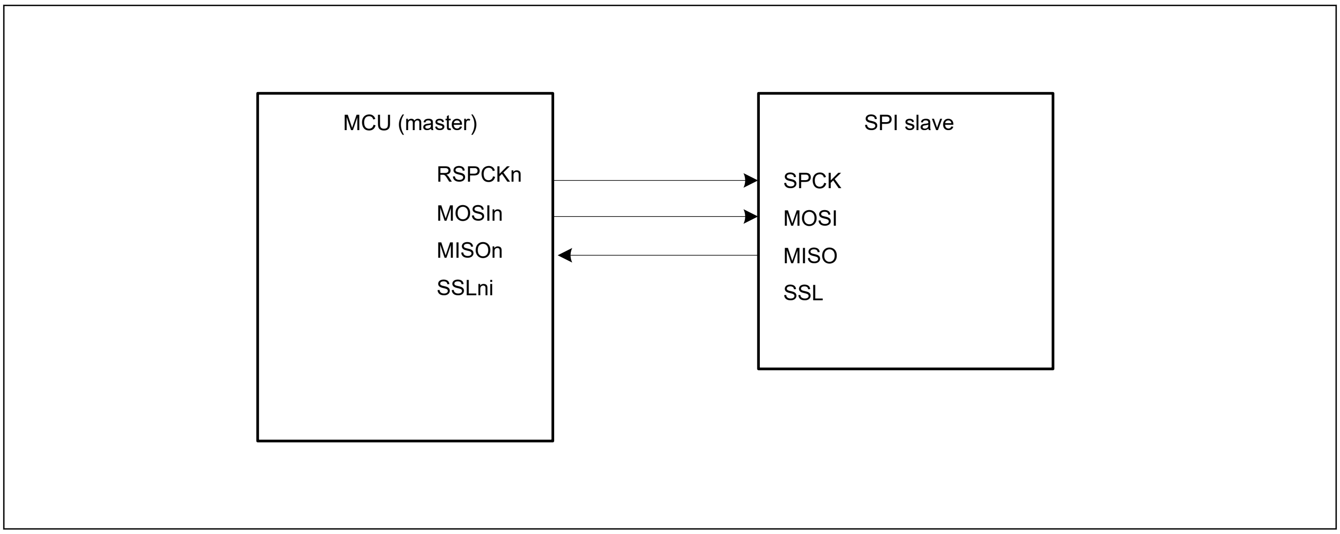


Figure 43.12 Clock synchronous master/slave configuration example with the MCU as a master

43.3.3.7 Master and Slave in Clock Synchronous Mode with the MCU as a Slave

Figure 43.13 shows a master and slave in clock synchronous mode configuration example where the MCU is used as a slave. When the MCU operates as a slave (clock synchronous operation), the MCU (slave) drives the MISOn signal and the SPI master drives the SPCK and MOSI signals. In addition, SSLn0 to SSLn3 of the MCU (slave) are not used.

The MCU (slave) can only execute serial transfers in the single-slave configuration when the SPCMDm.CPHA bit is set to 1.

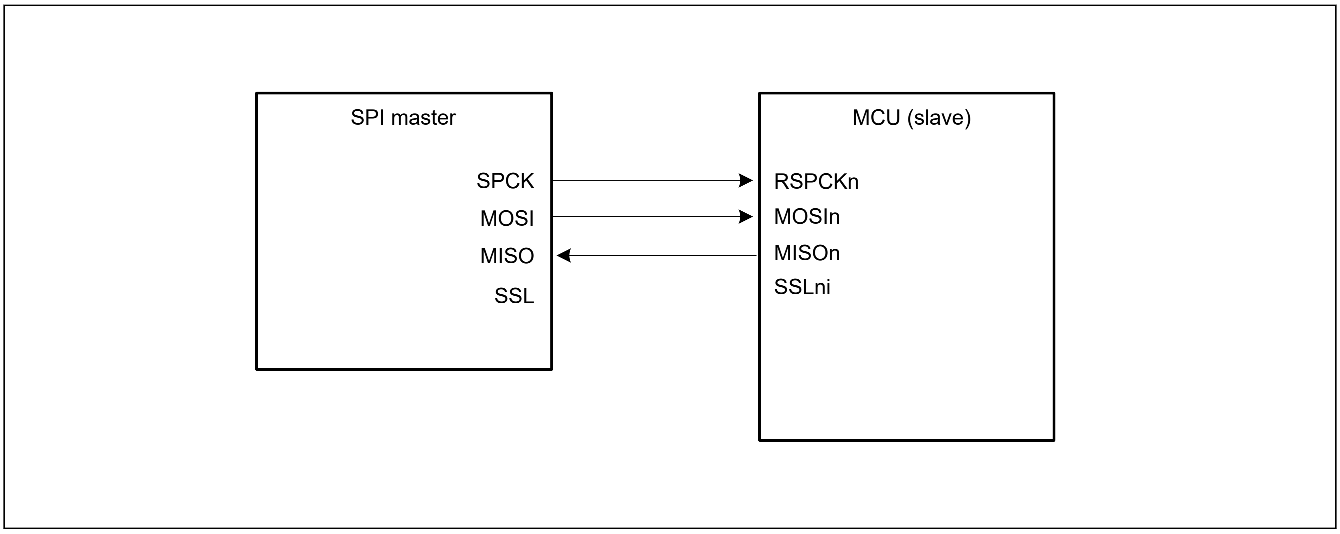


Figure 43.13 Clock synchronous master/slave configuration example with the MCU as a slave and CPHA = 1

43.3.4 Data Formats

The data format of the SPI depends on the settings in SPI Command Register m (SPCMDm) and the parity enable bit in SPI Control Register (SPCR.SPPE). Regardless of whether the MSB or LSB is first, the SPI treats the range from the LSB bit in the SPI Data Register (SPDR) to the bit associated with the selected data length, as transfer data.

This section shows the format of one frame of data before or after transfer.

Data format with parity disabled

When parity is disabled, transmission or reception of data proceeds with the length in bits selected in the SPI data length setting in SPI Command Register m (SPCMDm.SPB[4:0]).

Data format with parity enabled

When parity is enabled, transmission or reception of data proceeds with the length in bits selected in the SPI data length setting in SPI Command Register m (SPCMDm.SPB[4:0]). In this case, however, the last bit is a parity bit.

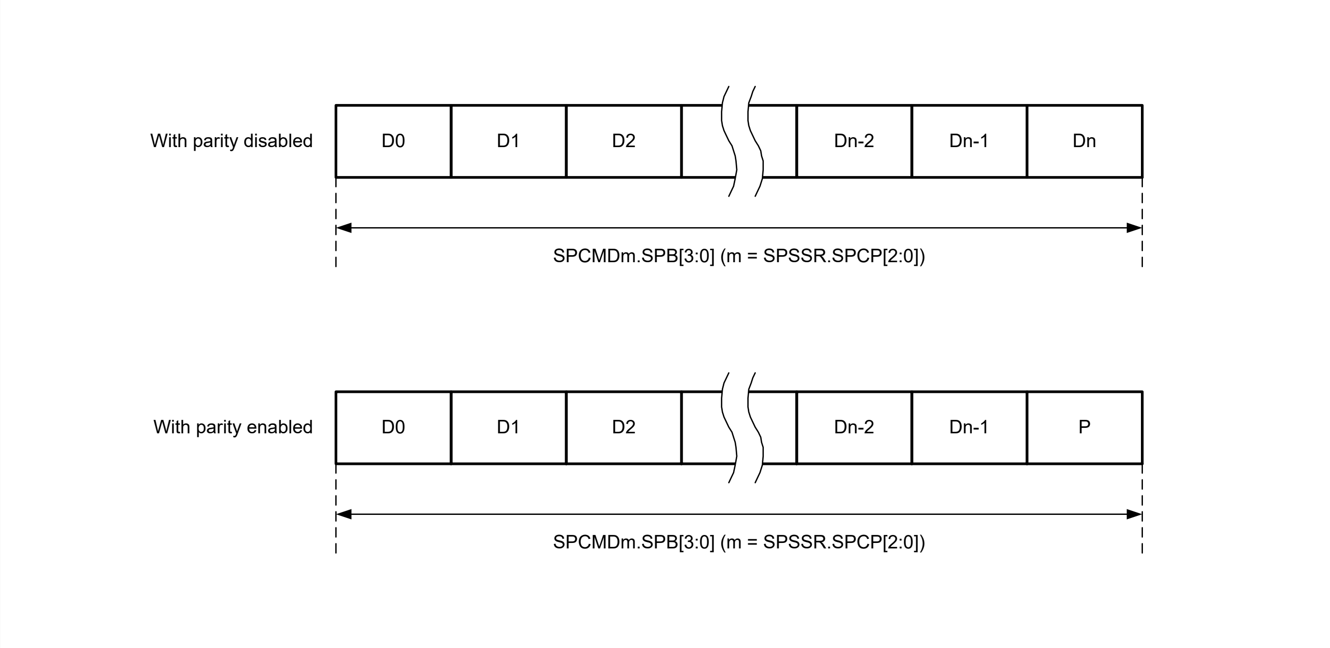


Figure 43.14 Data format with parity disabled and enabled

43.3.4.1 Operation When Parity is Disabled (SPCR.SPPE = 0)

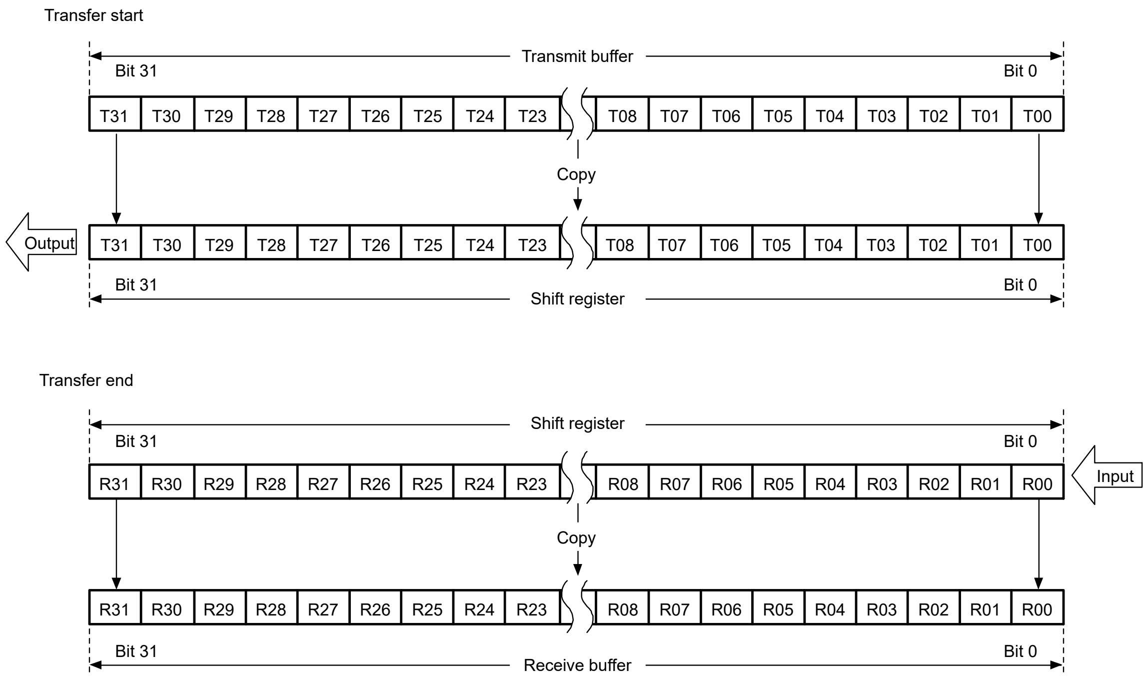
When parity is disabled, data for transmission is copied to the shift register with no pre-processing. This section describes the connection between the SPI Data Register (SPDR) and the shift register in terms of the combination of MSB- or LSB-first order and data length.

(1) MSB-first transfer with 32-bit data

Figure 43.15 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity disabled, an SPI data length of 32 bits, and MSB-first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are copied to the shift register. Data for transmission is shifted out from the shift register from T31 to T30, and continuing to T00.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R31 to R00 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

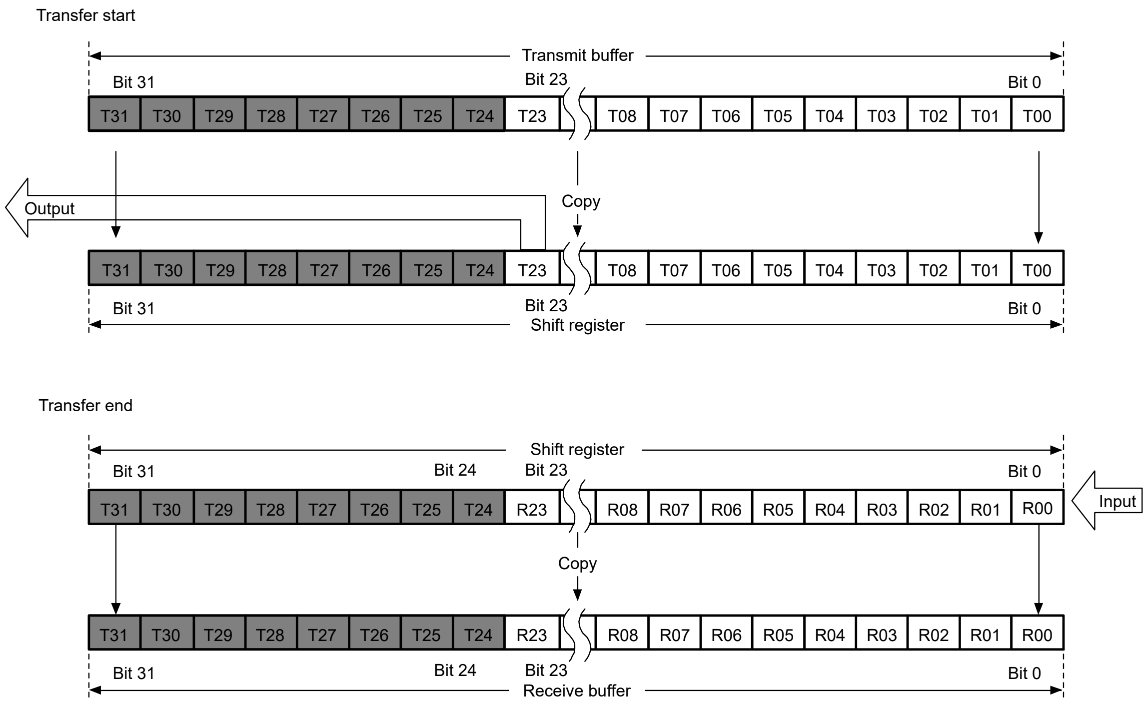
Figure 43.15 MSB-first transfer with 32-bit data and parity disabled

(2) MSB-first transfer with 24-bit data

Figure 43.16 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity disabled, an SPI data length of 24 bits for an example that is not 32 bits, and MSB-first selected.

In transmission, the lower 24 bits (T23 to T00) from the current stage of the transmit buffer are copied to the shift register. Data for transmission is shifted out from the shift register from T23 to T22, and continuing to T00.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R23 to R00 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

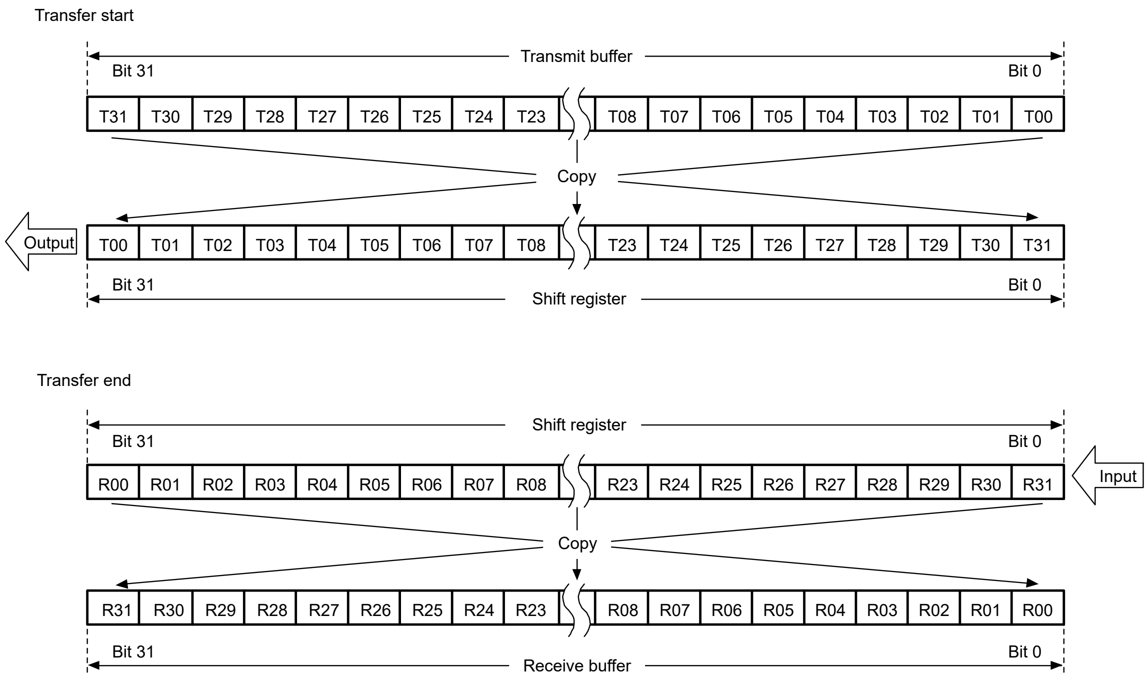
Figure 43.16 MSB-first transfer with 24-bit data and parity disabled

(3) LSB-first transfer with 32-bit data

Figure 43.17 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity disabled, an SPI data length of 32 bits, and LSB-first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are reordered bit-by-bit to obtain the order T00 to T31 for copying to the shift register. Data for transmission is shifted out from the shift register in order from T00 to T01, and continuing to T31.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R00 to R31 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

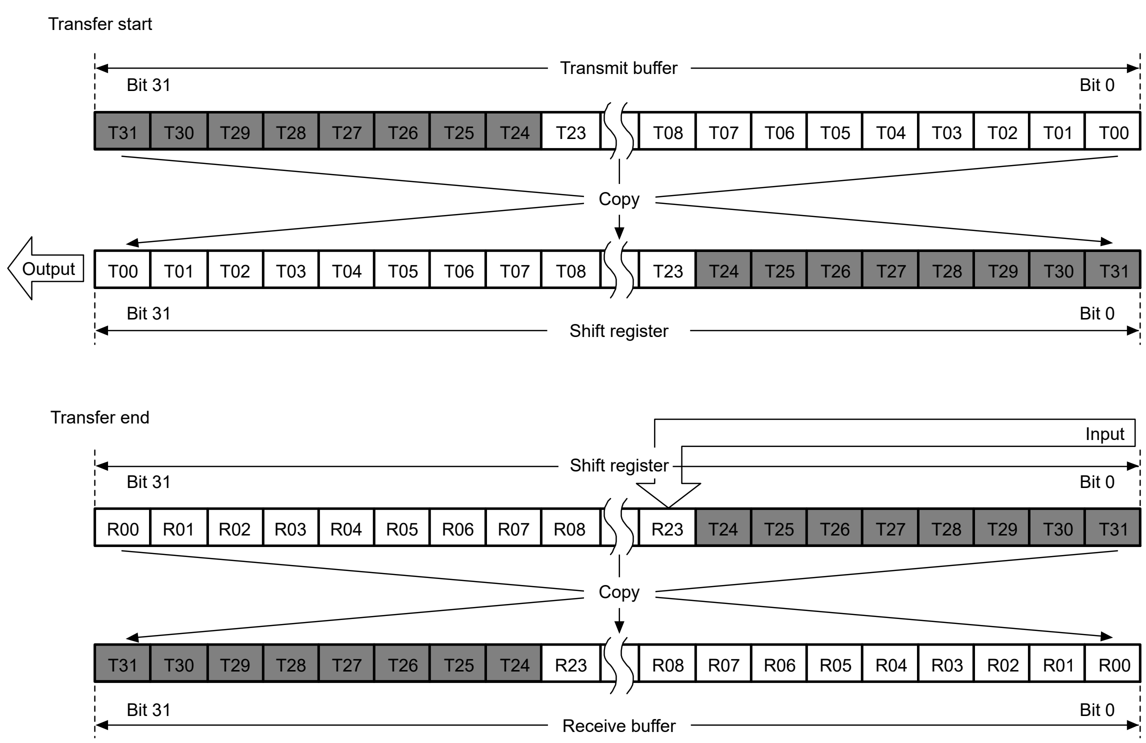
Figure 43.17 LSB-first transfer with 32-bit data and parity disabled

(4) LSB-first transfer with 24-bit data

Figure 43.18 shows the operation of the SPI Data Register (SPDR) and the shift register in transfers with parity disabled, an SPI data length of 24 bits for an example that is not 32, and LSB-first selected.

In transmission, the lower 24 bits (T23 to T00) from the current stage of the transmit buffer are reordered bit-by-bit to obtain the order T00 to T23 for copying to the shift register. Data for transmission is shifted out from the shift register from T00 to T01, and continuing to T23.

In reception, received data is shifted in bit-by-bit through bit[8] of the shift register. When the R00 to R23 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 43.18 LSB-first transfer with 24-bit data and parity disabled

43.3.4.2 Operation When Parity is Enabled (SPCR.SPPE = 1)

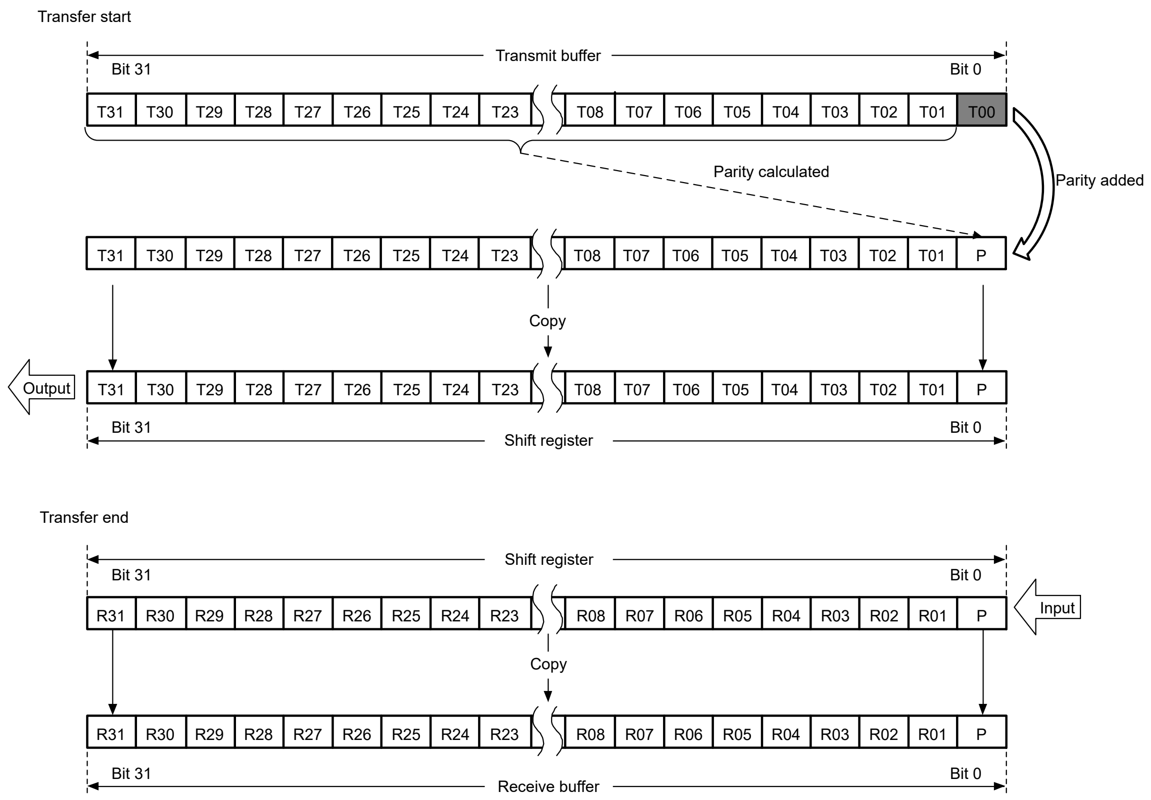
When parity is enabled, the lowest-order bit of the data for transmission becomes a parity bit. Hardware calculates the value of the parity bit.

(1) MSB-first transfer with 32-bit data

Figure 43.19 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, an SPI data length of 32 bits, and MSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T31 to T01. This replaces the final bit, T00, and the whole value is copied to the shift register. Data is transmitted in the order T31, T30, …, T01, and P.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R31 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R31 to P is checked for parity.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

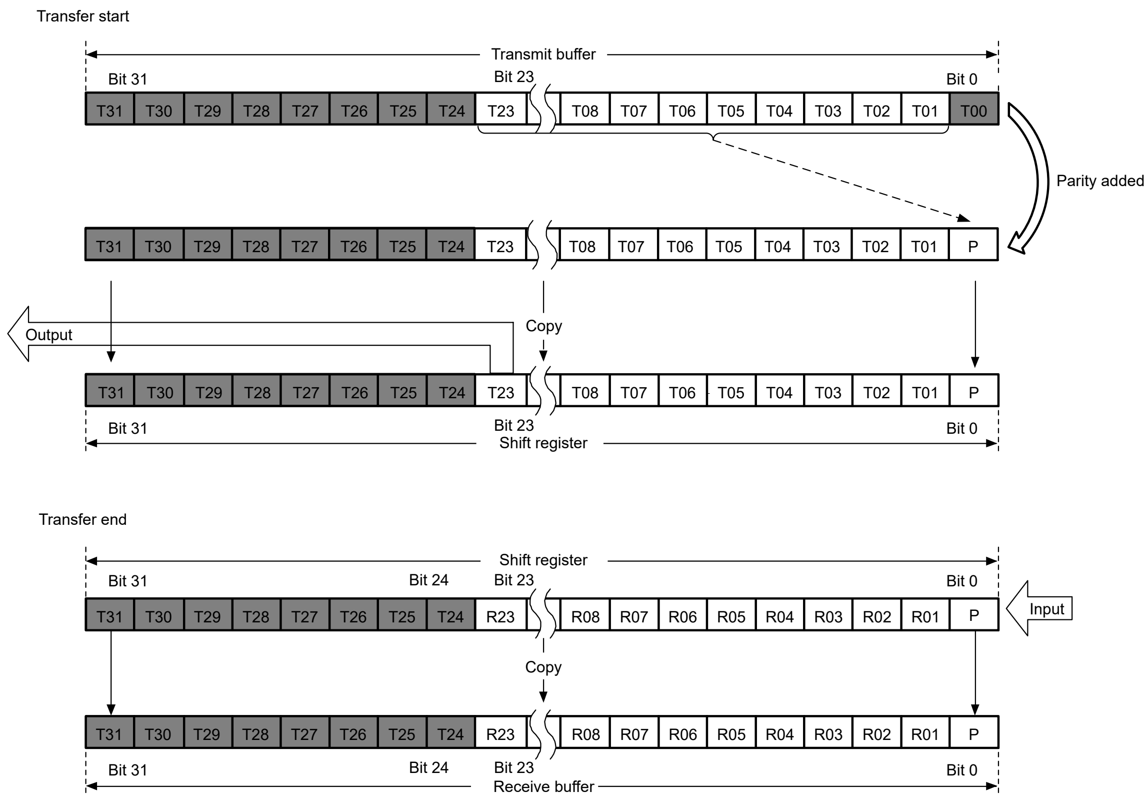
Figure 43.19 MSB-first transfer with 32-bit data and parity enabled

(2) MSB-first transfer with 24-bit data

Figure 43.20 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, an SPI data length of 24 bits, and MSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T23 to T01. This replaces the final bit, T00, and the whole value is copied to the shift register. Data is transmitted in the order T23, T22, …, T01, and P.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R23 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R23 to P is checked for parity.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

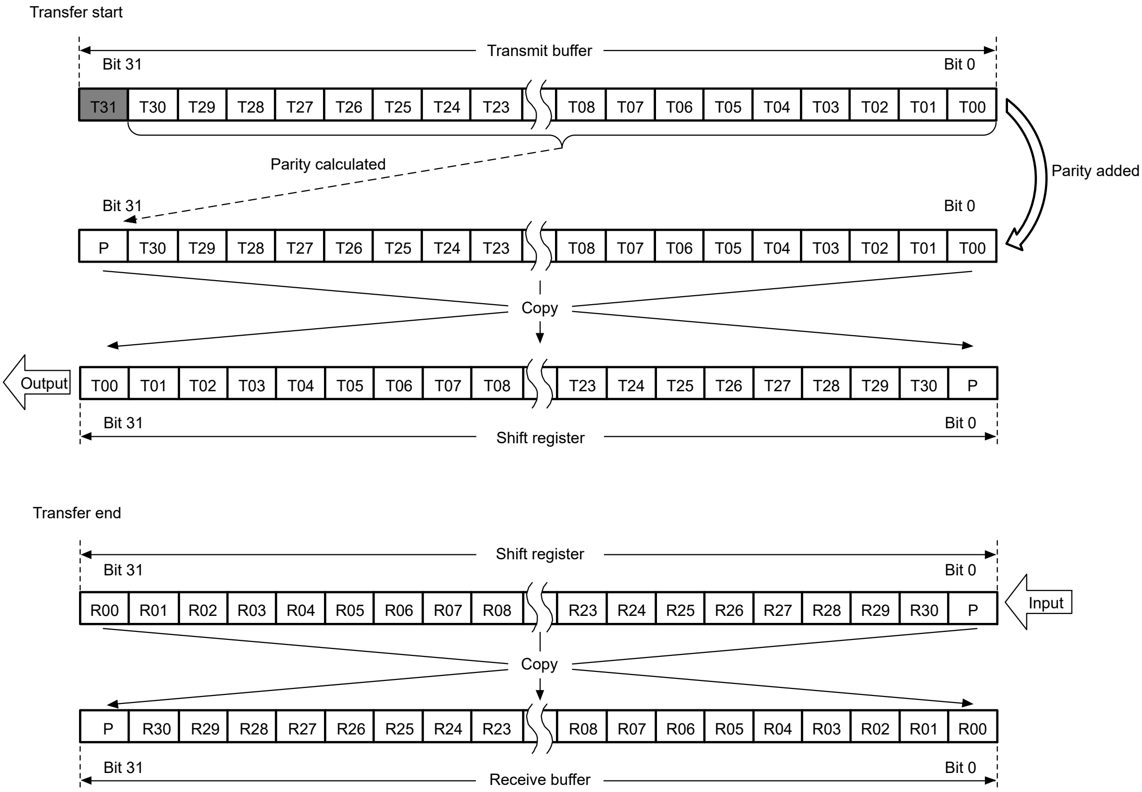
Figure 43.20 MSB-first transfer with 24-bit data and parity enabled

(3) LSB-first transfer with 32-bit data

Figure 43.21 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, an SPI data length of 32 bits, and LSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T30 to T00. This replaces the final bit, T31, and the whole value is copied to the shift register. Data is transmitted in the order T00, T01, …, T30, and P.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R00 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P is checked for parity.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

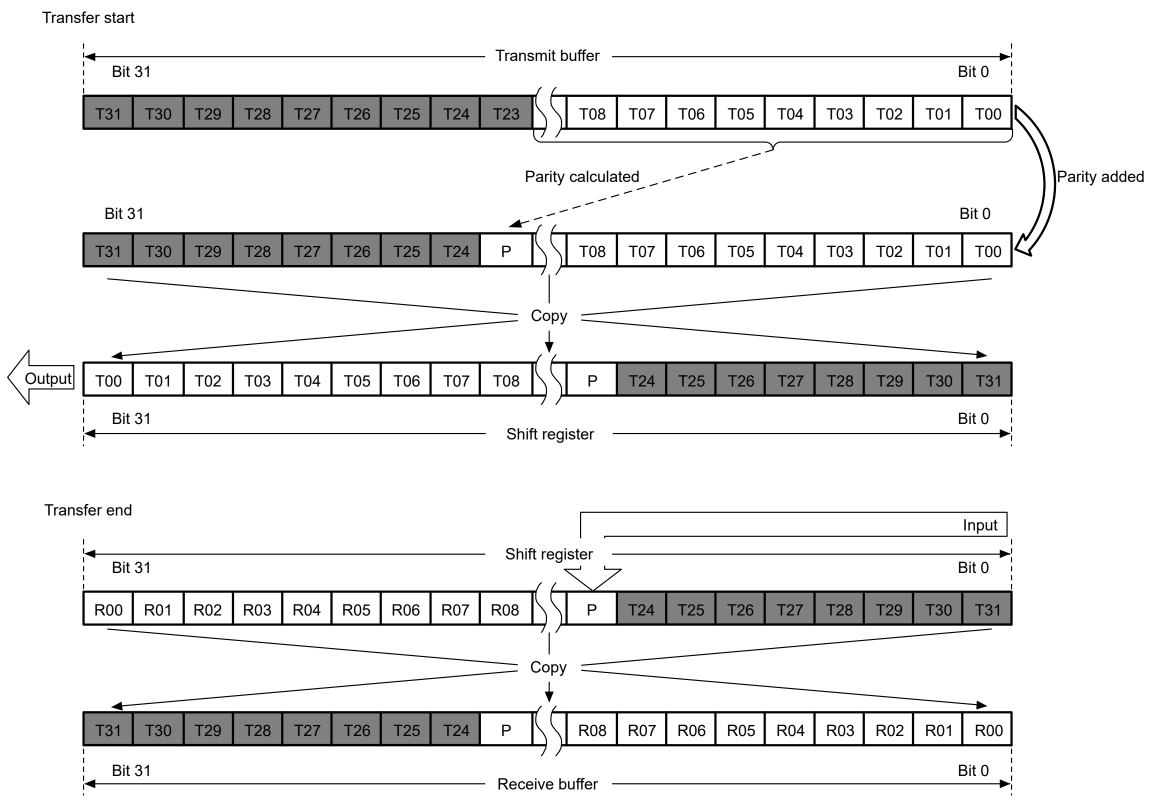
Figure 43.21 LSB-first transfer with 32-bit data and parity enabled

(4) LSB-first transfer with 24-bit data

Figure 43.22 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, an SPI data length of 24 bits, and LSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T22 to T00. This replaces the final bit, T23, and the whole value is copied to the shift register. Data is transmitted in the order T00, T01, …, T22, and P.

In reception, received data is shifted in bit-by-bit through bit[8] of the shift register. When the R00 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P is checked for parity.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 43.22 LSB-first transfer with 24-bit data and parity enabled

43.3.4.3 Byte Swap Transmission

When byte swapping is enabled, the data in the transmission buffer, swapped in 8-bit units, is copied to the shift register. Figure 43.23 shows the relationship between the SPDR (transmit buffer) and the shift register when transferring data with a 32-bit data length, using a combination of MSB / LSB first and with / without byte swap.

(1) MSB-first transfer. (When the byte swap is disabled.)

Data (Byte3 [T31 to T24] to Byte0 [T07 to T00]) in the transmit buffer are copied to the shift register.

Bit values in the shift register are shifted and transmitted in the order of T31 → T30 → … → T00 as transmit data.

(2) MSB-first transfer. (When the byte swap is enabled.)

Byte values of the transmit buffer (Byte3 [T31 to T24] to Byte0 [T07 to T00]) are reversed in byte units and are copied to the shift register in the order of Byte0 [T07 to T00] to Byte3 [T31 to T24].

Bit values in the shift register are shifted and transmitted in the order of T07 → T06 → … → T00 → T15 → T14 → … → T08 → T23 → T22 → … → T16 → T31 → T30 → … → T24 as transmit data.

(3) LSB-first transfer. (When the byte swap is disabled.)

Bit values of the transmit buffer (Byte3 [T31 to T24] to Byte0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte0 [T00 to T07] to Byte3 [T24 to T31].

Bit values in the shift register are shifted and transmitted in the order of T00 → T01 → … → T31 as transmit data.

(4) LSB-first transfer. (When the byte swap is enabled.)

Bit values of each byte of the transmit buffer (Byte3 [T31 to T24] to Byte0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte3 [T24 to T31] to Byte0 [T00 to T07].

Bit values in the shift register are shifted and transmitted in the order of T24 → T25 → … → T31 → T16 → T17 → … → T23 → T08 → T09 → … → T15 → T00 → T01 → … → T07 as transmit data.

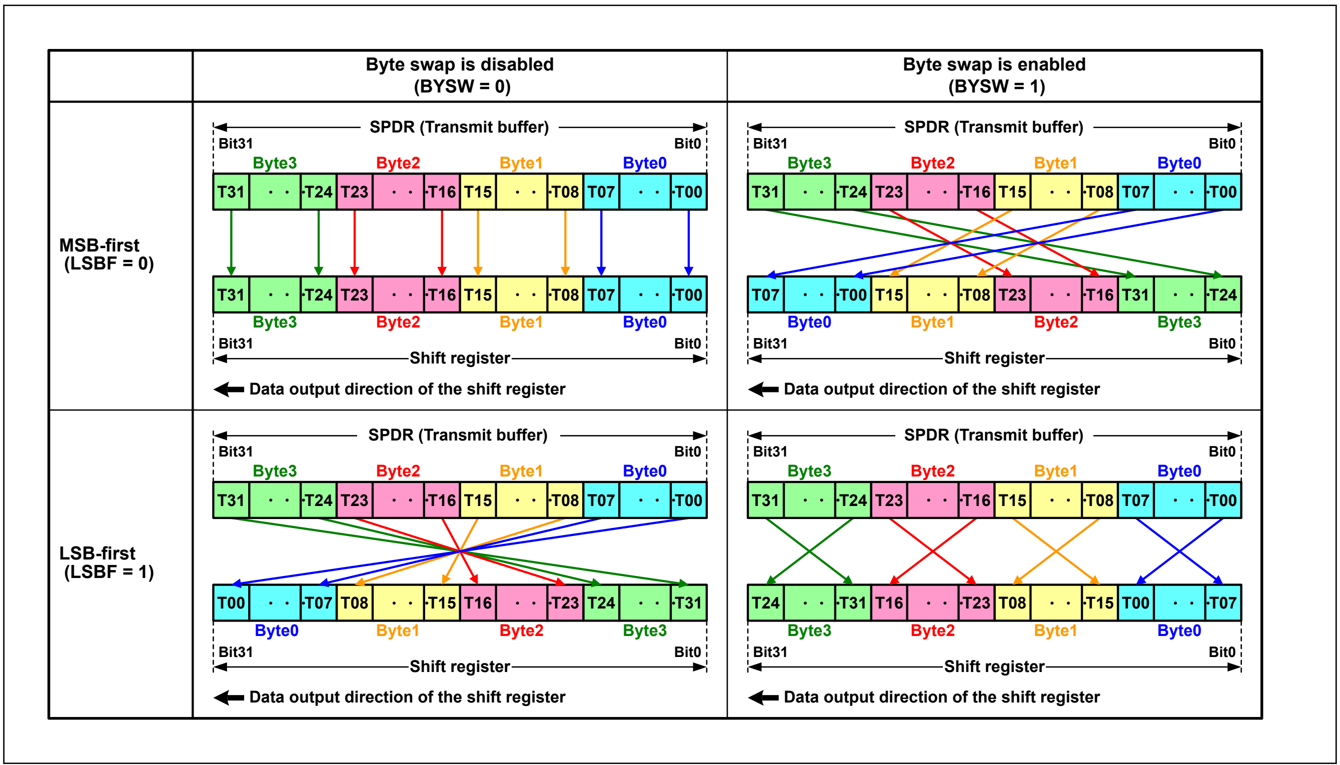


Figure 43.23 Byte swap with MSB/LSB transfer (32bit)

Figure 43.24 shows the relationship between the SPDR (transmit buffer) and the shift register when transferring data with a 16-bit data length, using a combination of MSB / LSB first and with / without byte swap.

1. MSB-first transfer. (When the byte swap is disabled.)
   Data (Byte1 [T15 to T08] to Byte0 [T07 to T00]) in the transmit buffer are copied to the shift register in the order of Byte1 [T15 to T08] to Byte0 [T07 to T00], Byte1 [T15 to T08] to Byte0 [T07 to T00]. Bit values in the shift register are shifted and transmitted in the order of T15 → T14 → … T00 as transmit data.
2. MSB-first transfer. (When the byte swap is enabled.)
   Byte values of the transmit buffer (Byte1 [T15 to T08] to Byte0 [T07 to T00]) are reversed in byte units and are copied to the shift register in the order of Byte0 [T07to T00] to Byte1 [T15 to T08], Byte0 [T07 to T00] to Byte1 [T15 to T08]. Bit values in the shift register are shifted and transmitted in the order of T07 → T06 → … T00 → T15 → T14 → … T08 as transmit data.
3. LSB-first transfer. (When the byte swap is disabled.)
   Bit values of the transmit buffer (Byte1 [T15 to T08] to Byte0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte0 [T00 to T07] to Byte1 [T08 to T15], Byte0 [T00 to T07] to Byte1 [T08 to T15]. Bit values in the shift register are shifted and transmitted in the order of T00 → T01 → … T15 as transmit data.
4. LSB-first transfer. (When the byte swap is enabled.)
   Bit values of each byte of the transmit buffer (Byte1 [T15 to T08] to Byte0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte1 [T08 to T15] to Byte0 [T00 to T07], Byte1 [T08 to T15] to Byte0 [T00 to T07]. Bit values in the shift register are shifted and transmitted in the order of T08 → T09 → … T15 → T00 → T01 → … T07 as transmit data.

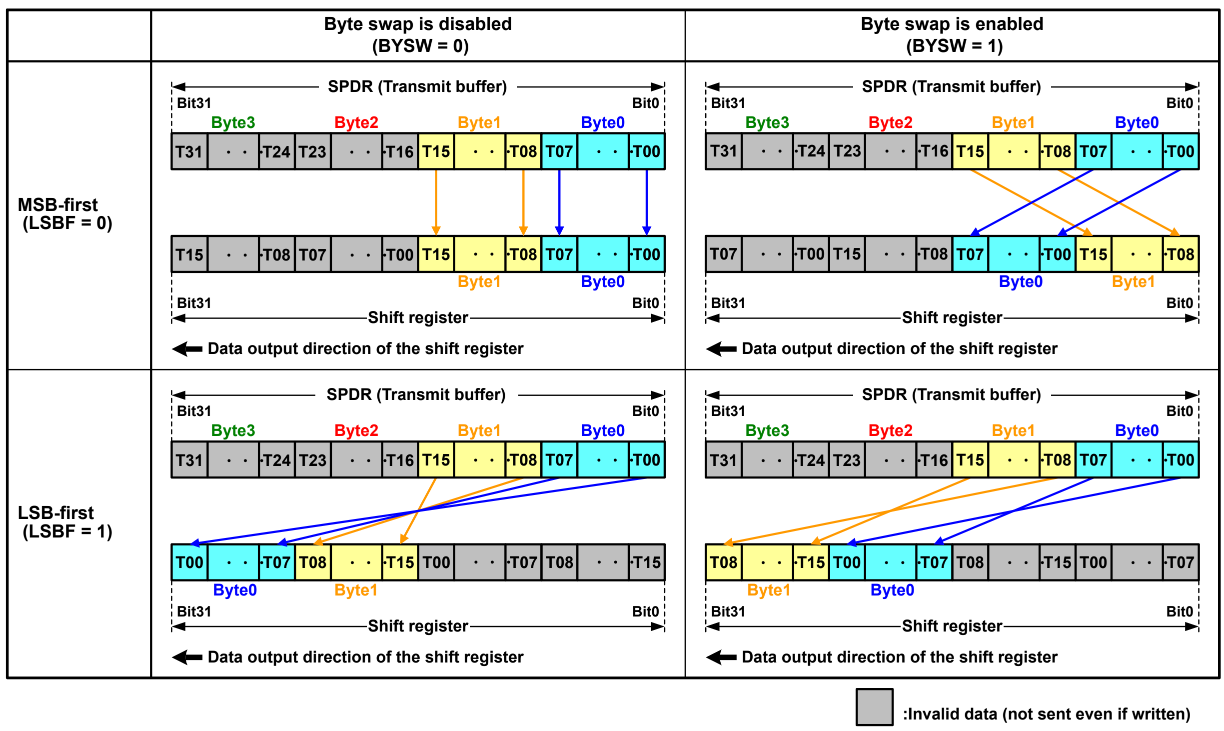


Figure 43.24 Byte swap with MSB/LSB transfer (16bit)

Note: 1. When using the byte swap, set 16 bits or 32 bits to the data length (SPCMDm.SPB[4:0] setting). If setting the other length, the behavior is not guaranteed.
2. When the byte swap is valid, set the parity function as invalid (SPCR.SPPE bit = 0). If setting the parity function as valid (SPPE bit = 1), the behavior is not guaranteed.
3. Set SPDCR.BYSW bit, when SPCR.SPE bit is 0. If rewriting BYSW bit, when SPE bit is 1, the behavior after it is not guaranteed.

43.3.4.4 Byte Swap Reception

When byte swap is enabled, the data in the shift register, swapped in 8-bit units, is copied to the receive buffer. Figure 43.25 shows the relationship between the shift register and SPDR (reception buffer) when transferring data with a 32-bit data length, using a combination of MSB / LSB first and with / without byte swap.

(1) MSB-first transfer (When the byte swap is disabled)

The first received data (R31) is stored in bit 0 of the shift register, and received data is shifted in the order of R31 → R30 → … → R00.

When necessary RSPCK cycles are input and data is stored from Byte3 [R31 to R24] to Byte0 [R07 to R00], the shift register value is copied to the receive buffer.

(2) MSB-first transfer (When the byte swap is enabled)

The first received data (R07) is stored in bit 0 of the shift register, and received data is shifted in the order of R07 → R06 → … → R00 → R15 → R14 → … → R08 → R23 → R22 → … → R16 → R31 → R30 → … → R24.

When necessary RSPCK cycles are input and data is stored from Byte0 [R07 to R00] to Byte3 [R31 to R24], byte values in the shift register are reversed in byte units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].

(3) LSB-first transfer (When the byte swap is disabled)

The first received data (R00) is stored in bit 0 of the shift register, and received data is shifted in the order of R00 → R01 → … → R31.

When necessary RSPCK cycles are input and data is stored from Byte0 [R00 to R07] to Byte3 [R24 to R31], bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].

(4) LSB-first transfer (When the byte swap is enabled)

The first received data (R24) is stored in bit 0 of the shift register, and received data is shifted in the order of R24 → R25 → … → R31 → R16 → R17 → … → R23 → R08 → R09 → … → R15 → R00 → R01 → … → R07.

When necessary RSPCK cycles are input and data is stored from Byte3 [R24 to R31] to Byte0 [R00 to R07], bit values of each byte in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].

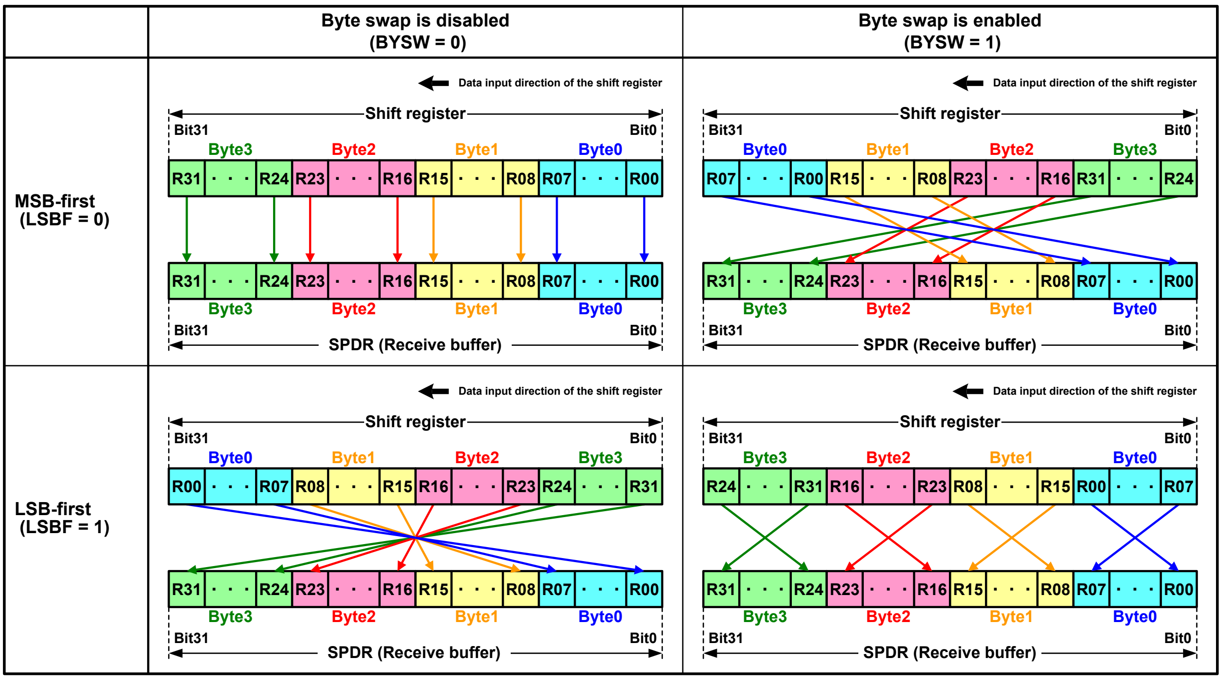


Figure 43.25 Byte swap with MSB/LSB transfer (32-bit)

Figure 43.26 shows the relationship between the shift register and SPDR (reception buffer) when transferring data with a 16-bit data length, using a combination of MSB / LSB first and with / without byte swap.

1. MSB-first transfer (when the byte swap is disabled)
   The first received data (R15) is stored in bit 0 of the shift register, and received data is shifted in the order of R15 → R14 → … R00. When necessary RSPCK cycles are input and data is stored from Byte3 [R31 to R24] to Byte0 [R07 to R00], the shift register value is copied to the receive buffer.
2. MSB-first transfer (when the byte swap is enabled)
   The first received data (R07) is stored in bit 0 of the shift register, and received data is shifted in the order of R07 → R06 → … R00 → R15 → R14 → … R08. When necessary RSPCK cycles are input and data is stored from Byte0 [R07 to R00] to Byte1 [R15 to R08], byte values in the shift register are reversed in byte units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].
3. LSB-first transfer (when the byte swap is disabled)
   The first received data (R00) is stored in bit 15 of the shift register, and received data is shifted in the order of R00 → R01 → … R07 → R08 → R09 → … R15. When necessary RSPCK cycles are input and data is stored from Byte0 [R00 to R07] to Byte1 [R08 to R15], bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07-R00].
4. LSB-first transfer (when the byte swap is enabled)
   The first received data (R08) is stored in bit 15 of the shift register, and received data is shifted in the order of R08 → R09 → … R15 → R00 → R01 → … R07. When necessary RSPCK cycles are input and data is stored from Byte1 [R08

to R15] to Byte0 [R00 to R07], bit values of each byte in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].

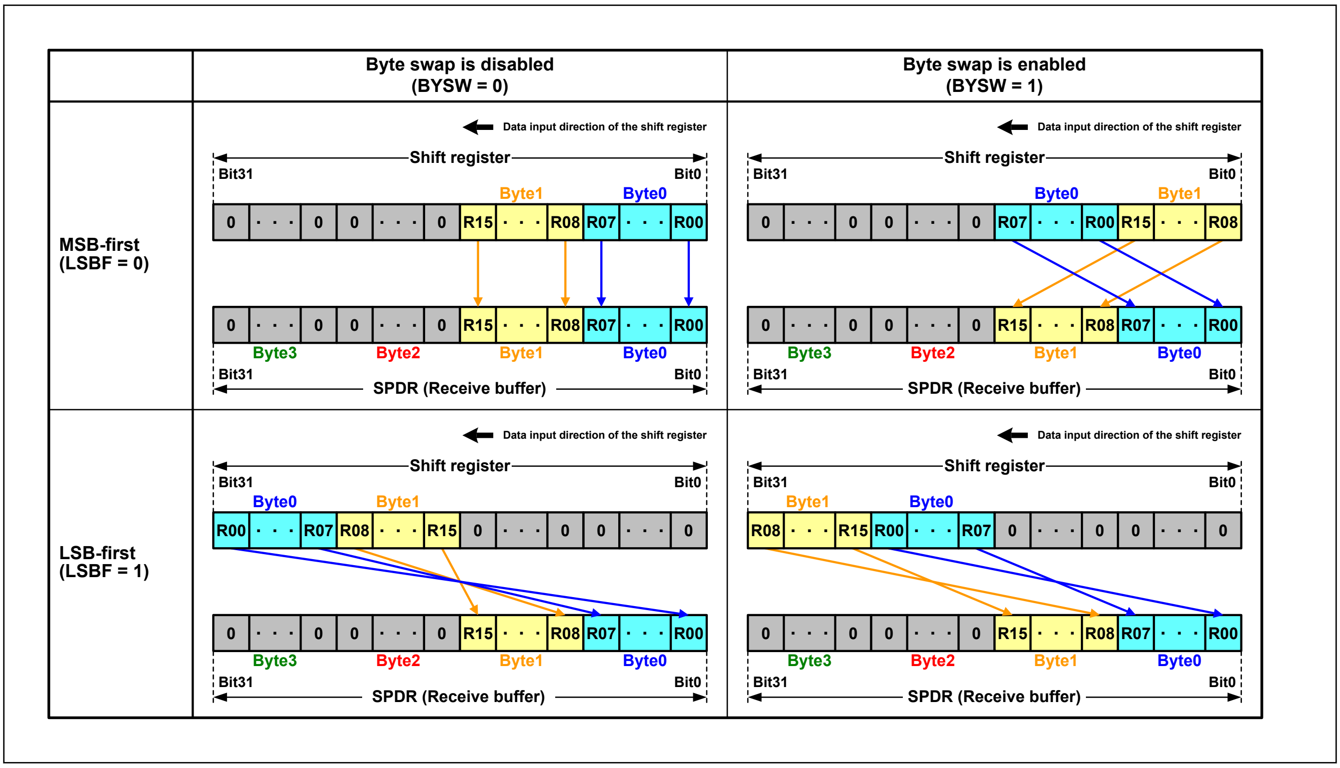


Figure 43.26 Byte swap with MSB/LSB transfer (16-bit)

Note: 1. When using the byte swap, set 16 bits or 32 bits to the data length (SPCMDm.SPB[4:0] setting). If setting the other length, the behavior is not guaranteed.
2. When the byte swap is valid, set the parity function as invalid (SPCR.SPPE bit = 0). If setting the parity function as valid (SPPE bit = 1), the behavior is not guaranteed.
3. Set SPDCR.BYSW bit, when SPCR.SPE bit is 0. If rewriting BYSW bit, when SPE bit is 1, the behavior after it is not guaranteed.

43.3.5 Transfer Formats

43.3.5.1 When CPHA = 0

Figure 43.27 shows an example transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 0. Do not perform clock synchronous operation (SPCR.SPMS = 1) when the SPI operates in slave mode (SPCR.MSTR = 0) and the CPHA bit is 0. In Figure 43.27, RSPCKn (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMDm.CPOL bit is 0, and RSPCKn (CPOL = 1) indicates the RSPCKn signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the SPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the SPI settings. For details, see section 43.3.2. Controlling the SPI Pins.

When the SPCMDm.CPHA bit is 0, the driving of valid data to the MOSIn and MISOn signals begins at an SSLni signal assertion. The first RSPCKn signal change that occurs after the SSLni signal assertion becomes the first transfer data fetch. After this, data is sampled every 1 RSPCKn cycle. The change timing for MOSIn and MISOn signals is 1/2 RSPCK cycles after the transfer data fetch timing. The CPOL bit setting does not affect the RSPCKn signal operation timing as it only affects the signal polarity.

t1 denotes the RSPCK delay, the period from an SSLni signal assertion to RSPCKn oscillation. t2 denotes the SSL negation delay, the period from the termination of RSPCKn oscillation to an SSLni signal negation. t3 denotes the next-access delay, the period in which SSLni signal assertion is suppressed for the next transfer after the end of serial transfer. t1, t2, and t3 are controlled by a master device running on the SPI system. For a description of t1, t2, and t3 when the SPI is in master mode, see section 43.3.12.1. Master Mode Operation.

[In the Motorola-SPI case]

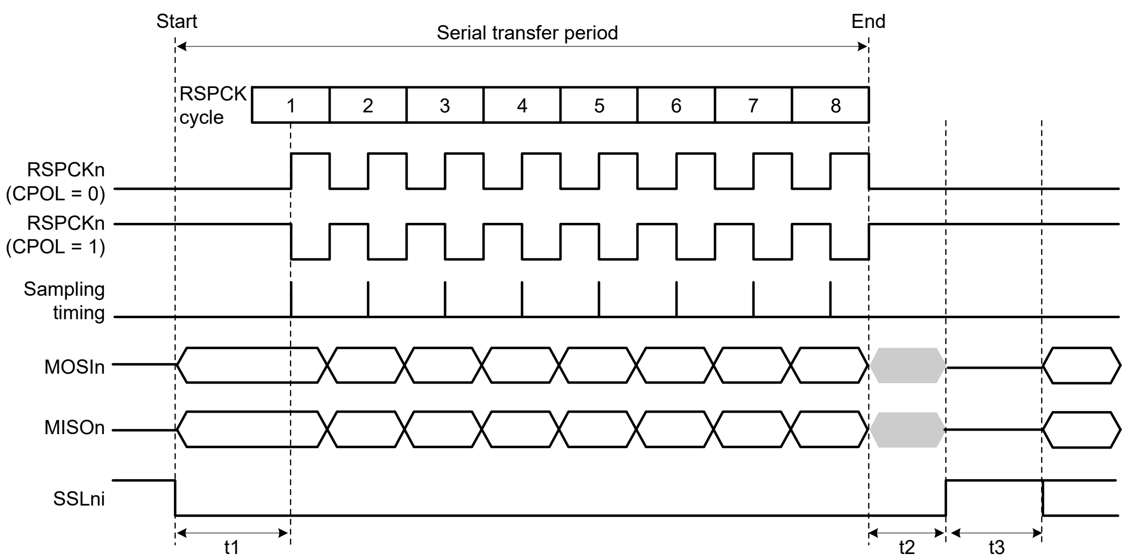


Figure 43.27 SPI transfer format when CPHA = 0, SPFRF = 0

[In TI-SSP case]

Not supported in CPHA = 0

43.3.5.2 When CPHA = 1

Figure 43.28 shows an example transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 1. However, when the SPCR.SPMS bit is 1, the SSLni signals are not used, and only the three signals RSPCKn, MOSIn, and MISOn handle communications. In Figure 43.28, RSPCKn (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMDm.CPOL bit is 0 and RSPCKn (CPOL = 1) indicates the RSPCKn signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the SPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the SPI mode (master or slave mode). For details, see section 43.3.2. Controlling the SPI Pins.

When the SPCMDm.CPHA bit is 1, the driving of invalid data to the MISOn signal begins at an SSLni signal assertion. The output of valid data to the MOSIn and MISOn signals begins at the first RSPCKn signal change that occurs after the SSLni signal assertion. After this, data is updated every 1 RSPCK cycle. The transfer data fetch timing is 1/2 RSPCK cycles after the data update timing. The SPCMDm.CPOL bit setting does not affect the RSPCKn signal operation timing. It only affects the signal polarity.

t1, t2, and t3 are the same as those when CPHA = 0. For a description of t1, t2, and t3 when the SPI of the MCU is in master mode, see section 43.3.12.1. Master Mode Operation.

[In the Motorola-SPI case]

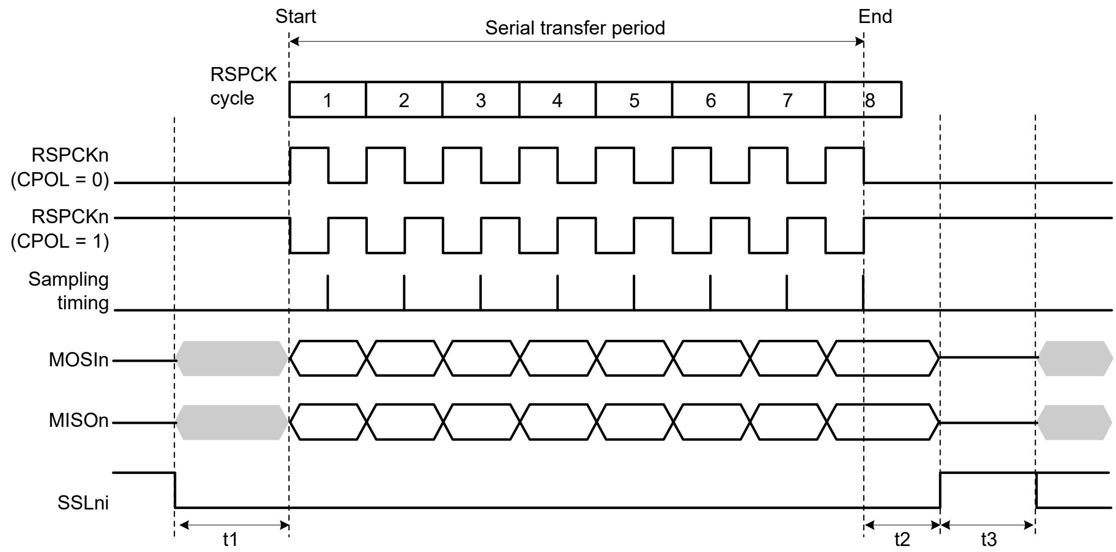


Figure 43.28 SPI transfer format when CPHA = 1, SPFRF = 0

[In the TI-SSP case]

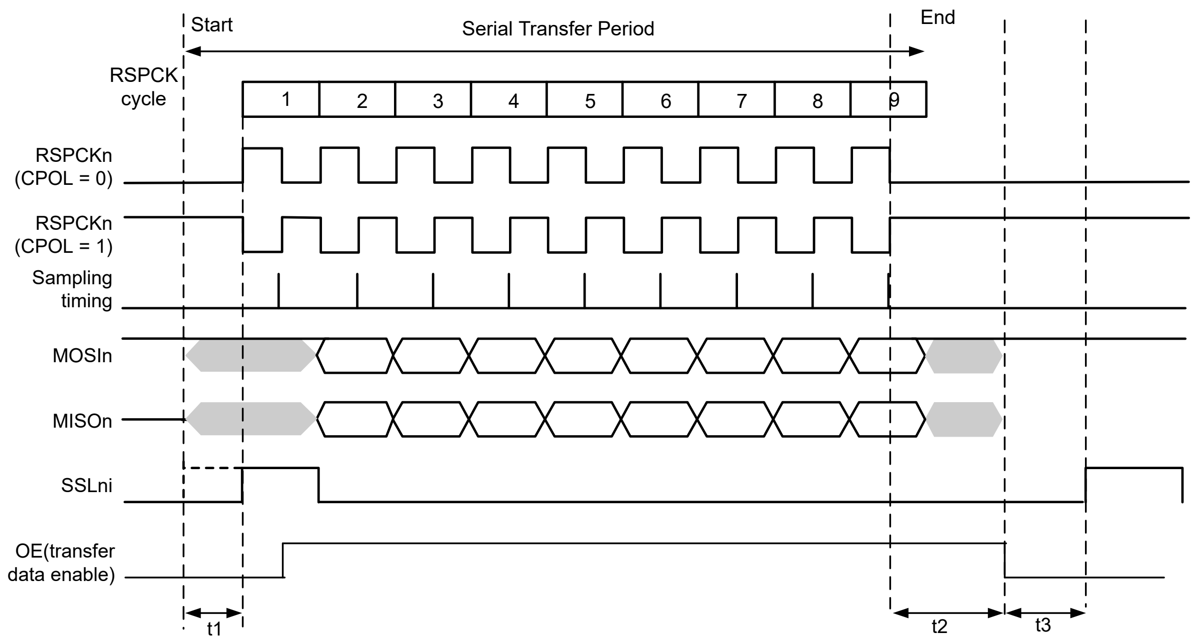


Figure 43.29 SPI transfer format when CPHA = 1, SPFRF = 1

43.3.6 Communication Operating Mode

Transmit-Receive serial communication, transmit-only operation, and Receive-only operation are selected by setting the Communication Mode Select bits (TXMD [1:0]) of the SPI control register (SPCR).

SPDR access described in Figure 43.30, Figure 43.31, Figure 43.32 shows an access to the SPI data register (SPDR). W shows a write cycle.

43.3.6.1 Transmit-Receive Serial Communication (TXMD[1:0] = 00b)

Figure 43.30 shows an example of operation when the communication mode select bit (TXMD[1:0]) in the SPI control register (SPCR) is set to 00b. In the example, the SPI performs an 8-bit serial transfer when the SPDCR2.TTRG is 0, the SPDCR2.RTRG is FIFO stage - 1, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

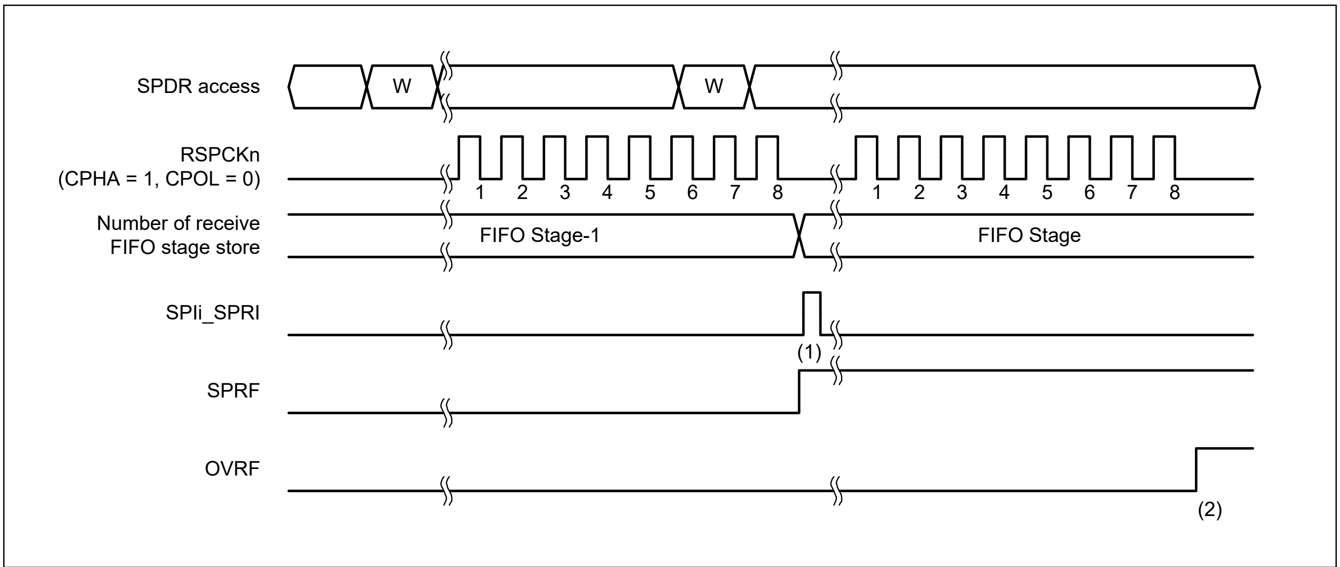


Figure 43.30 Operation example when SPCR.TXMD[1:0] = 00b

The operation of the flags at timings (1) and (2) in Figure 43.30 is as follows:

1. When serial transfer ends while the number of SPDR receive buffer store matches the number of frames set in SPDCR2.RTRG, the SPI generates a receive buffer full interrupt request (SPIi_SPRI), the SPI sets the SPSR.SPRF flag to 1, and copies the received data in the shift register to the receive buffer.
2. When serial transfer ends with data for the number of FIFO stages stored in the SPDR receive buffer, the SPI sets the SPSR.OVRF flag to 1, and discards the received data in the shift register. For details about the operation of the SPSR.OVRF flag, see section 43.3.10.1. Overrun Errors.

In Transmit-Receive serial communication (TXMD[1:0] = 00b), transmit data is transmitted and receive data is received. Therefore, the SPRF flag and the OVRF flag are set to 1 at timings (1) and (2) respectively.

43.3.6.2 Transmit-Only Serial Communications (TXMD[1:0] = 01b)

Figure 43.31 shows an example of operation when the communication mode select bit (TXMD[1:0]) in the SPI control register (SPCR) is set to 01b. In this example, the SPI performs an 8-bit serial transfer when the SPDCR2.TTRG is 0, the SPDCR2.RTRG is 0, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

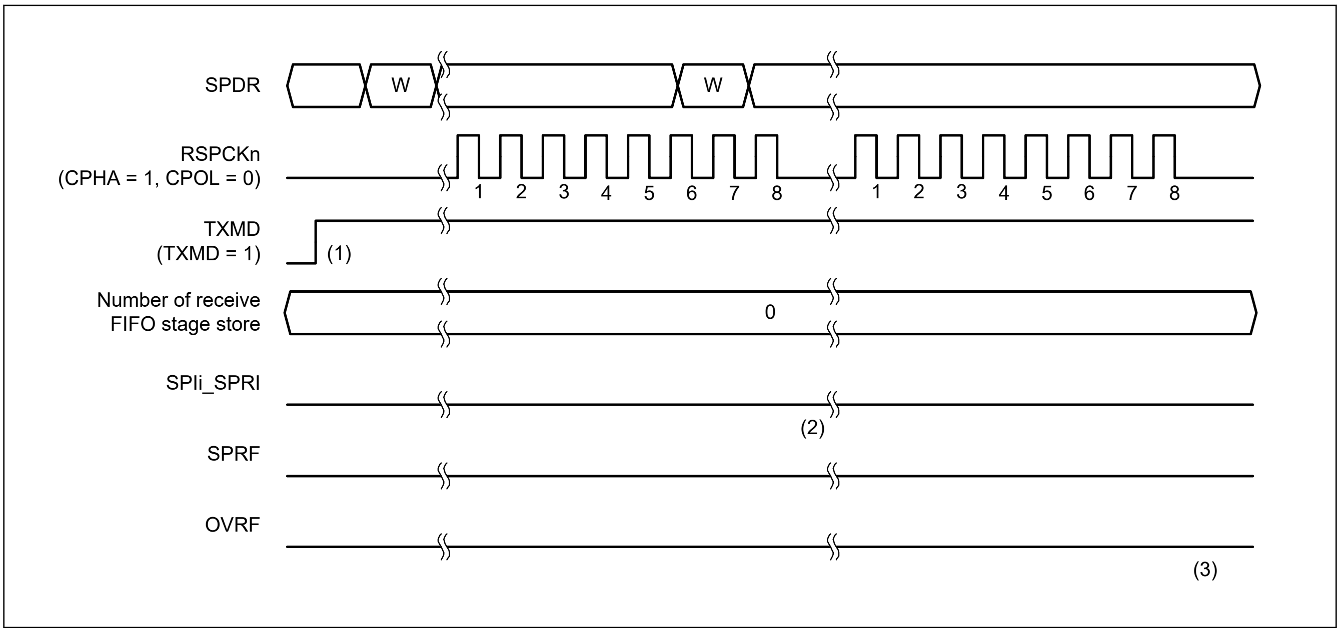


Figure 43.31 Operation example when SPCR.TXMD[1:0] = 01b

The operation of the flags at timings (1) to (3) in Figure 43.31 is as follows:

1. Make sure there is no data left in the receive buffer (the SPSR.SPRF flag is 0) and the SPSR.OVRF flag is 0 before entering transmit-only mode (SPCR.TXMD[1:0] = 01b).
2. When a serial transfer ends without receiving data in the receiving FIFO of SPDR, if the transmit-only mode is selected (SPCR.TXMD[1:0] = 01b), the SPSR.SPRF flag retains the value of 0, and the SPI does not copy the data in the shift register to the receive buffer.
3. Because the receive buffer of SPDR does not hold data that was received in the previous serial transfer, even when a serial transfer ends, the SPSR.OVRF flag retains the value of 0, and the data in the shift register is not copied to the receive buffer.

In transmit-only mode (SPCR.TXMD[1:0] = 01b), the SPI transmits data but does not receive data. Therefore, the SPSR.SPRF and SPSR.OVRF flags remain 0 at timings (1) to (3).

43.3.6.3 Receive-Only Serial Communication (TXMD[1:0] = 10b)

Figure 43.32 shows an example of operation when the communication mode select bit (TXMD[1]) in the SPI control register (SPCR) is set to 1. In this example, the SPI performs an 8-bit serial transfer when the SPDCR2.TTRG is FIFO stage - 1, the SPDCR2.RTRG is 0, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

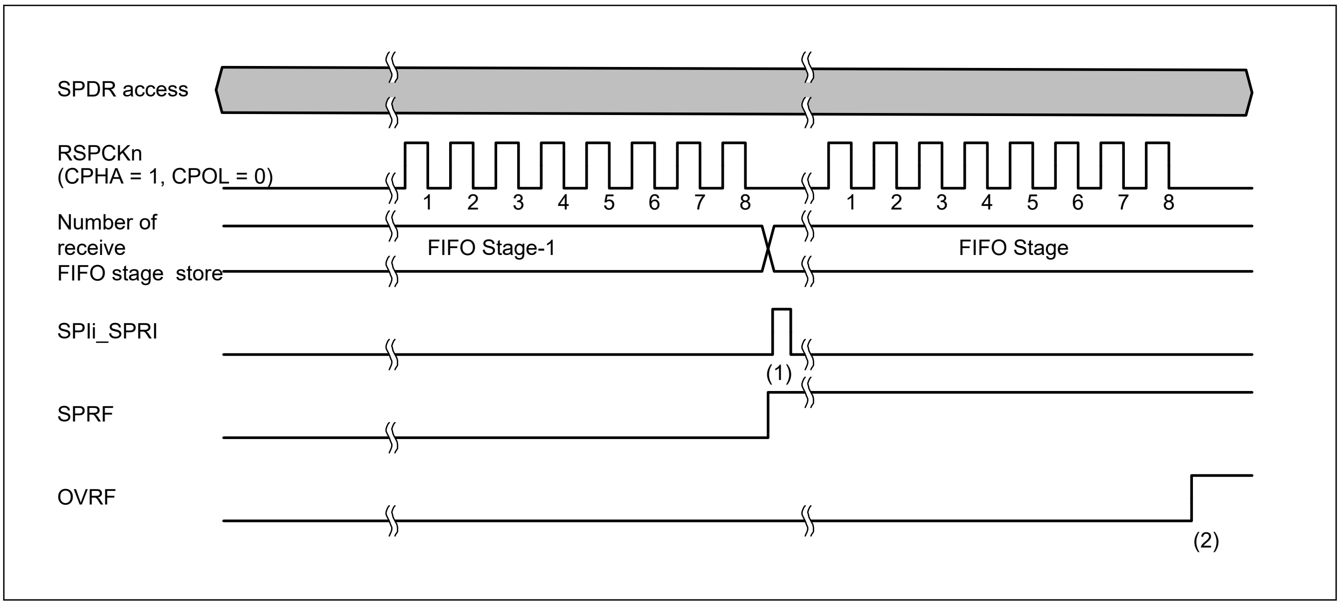


Figure 43.32 Example of Operation when SPCR.TXMD[1:0] = 10b

The following describes operation of flags at timings (1) and (2) in the figure above.

1. When serial transfer ends while the number of SPDR receive buffer store matches the number of frames set in SPDCR2.RTRG, the SPI generates a receive buffer full interrupt request (SPIi_SPRI), the SPI sets the SPSR.SPRF flag to 1, and copies the received data in the shift register to the receive buffer
2. When serial transfer ends with data for the number of FIFO stages stored in the SPDR receive buffer, the SPI sets the SPSR.OVRF flag to 1 and discards the received data in the shift register.

43.3.7 Transmit Buffer Empty and Receive Buffer Full Interrupts

Figure 43.33 show examples of operation of the transmit buffer empty interrupt (SPIi_SPTI (i = 0, 1)) and the receive buffer full interrupt (SPIi_SPRI). The SPDR register accesses shown in these figures indicate the conditions of access to the register, where W denotes a write cycle and R a read cycle. In Figure 43.33, the SPI performs an 8-bit serial transfer when SPCR.TXMD[1:0] bits are 00b, the SPDCR2.TTRG bit is 0, the SPDCR2.RTRG bit is 0, the SPCMDm.CPHA bit is 0, and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

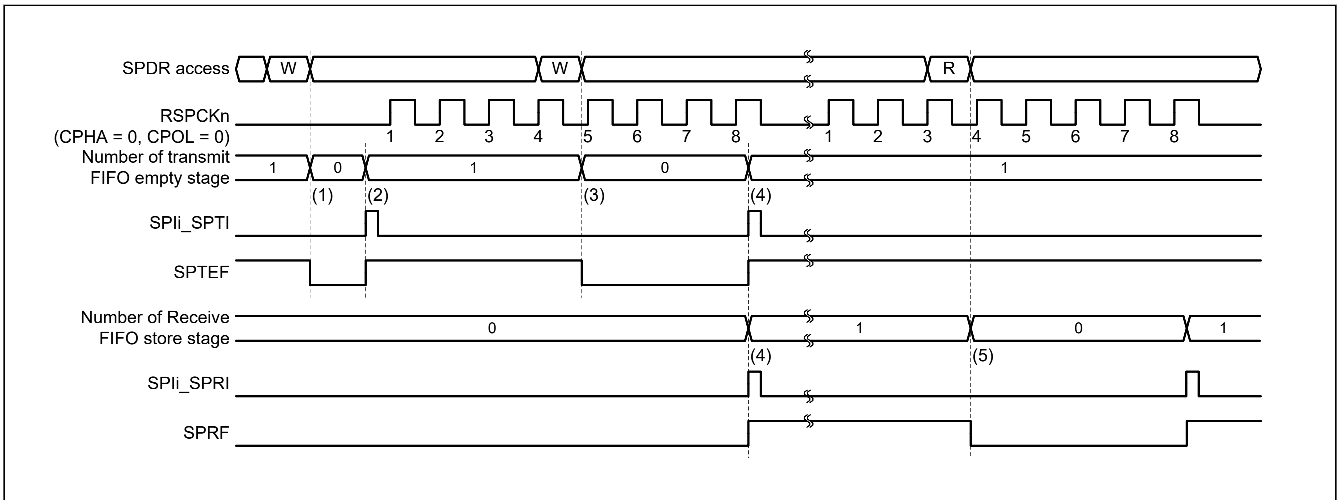


Figure 43.33 Operation example of the SPIi_SPTI and SPIi_SPRI interrupts when CPHA = 0 and CPOL = 0 in master mode

The operation of the SPI at timings (1) to (5) in Figure 43.33 is as follows:

1. When transmit data is written to SPDR with the transmit buffer of SPDR is before the next transfer data is set, the SPI writes data to the transmit buffer. When transmit data is written to SPDR in one processing routine using DTC / DMAC, the SPSR.SPTEF flag is cleared to 0 at the last access.
2. If the shift register is empty, the SPI copies the data in the transmit buffer to the shift register. At this time, if transmit FIFO empty stage number > TTRG value, then the SPI generates a transmit buffer empty interrupt request (SPIi_SPTI), and sets the SPSR.SPTEF flag to 1. How a serial transfer is started depends on the SPI mode. For details, see section 43.3. Operation, and section 43.3.13. Clock Synchronous Operation.
3. When transmit data is written to SPDR either by the transmit buffer empty interrupt routine, or by the processing of the transmit buffer empty using the SPTEF flag, the SPI writes data to the transmit buffer. When the transmit data is written to SPDR in one processing routine using DTC / DMAC, the SPTEF flag is cleared to 0 at the last access. Because the data being transferred serially is stored in the shift register, the SPI does not copy the data in the transmit buffer to the shift register.
4. When the serial transfer ends with the receive buffer of SPDR > FIFO stage number, the SPI copies the receive data in the shift register to the receive buffer, generates a receive buffer full interrupt request (SPIi_SPRI), and sets the SPRF flag to 1. Because the shift register becomes empty on completion of the serial transfer, if the next transfer data is set in the transmit FIFO before the serial transfer ended, the SPI sets the SPTEF flag to 1 and copies data in the transmit buffer to the shift register. Even when received data is not copied from the shift register to the receive buffer in an overrun error status, on completion of the serial transfer, the SPI determines that the shift register is empty, so data transfer from the transmit buffer to the shift register is enabled.
5. When SPDR is read either by the receive buffer full interrupt routine or processing of the receive buffer full interrupt using the SPRF flag, the receive data can be read. If the received data is read from SPDR in one processing routine using DTC / DMAC, the SPRF flag is cleared to 0 at the last access.

When transmit data is written to the SPDR register while no empty stages in the transmit FIFO, the SPI does not update data in the transmit buffer. When writing to SPDR, always use either a transmit buffer empty interrupt request or check the empty or processing of the transmit buffer empty interrupt using the SPTEF flag. To use a transmit buffer empty interrupt, set the SPTIE bit in SPCR to 1. If the SPI function is disabled (the SPCR.SPE bit is 0), set the SPTIE bit to 0.

When serial transfer ends while data is stored in the receive FIFO for the number of FIFO stages, the SPI does not copy data from the shift register to the receive buffer, and it detects an overrun error (see section 43.3.10. Error Detection). To prevent a receive data overrun error, read the received data using a receive buffer full interrupt request before the next serial transfer ends. To use an SPI receive buffer full interrupt, set the SPCR.SPRIE bit to 1.

Transmission and reception interrupts or the associated IELSRn.IR flags (where n is the interrupt vector number) in the ICU can be used to confirm the states of the transmit and receive buffers.

Similarly, the SPTEF and SPRF flags can be used to confirm the states of the transmit and receive buffers. See section 14, Interrupt Controller Unit (ICU) for the interrupt vector numbers.

43.3.8 Idle Interrupt

When the SPCP[2:0] of the SPI status register (SPSR) becomes 000b (start of sequence control), the IDLNF flag in the SPI status register (SPSR) is set to 1 and an idle interrupt request is made during master mode operation. An interrupt request is also made by clearing the SPCR.SPE bit to 0.

[In the Motorola-SPI case]

Figure 43.34 shows an example of idle interrupt operation during normal operation.

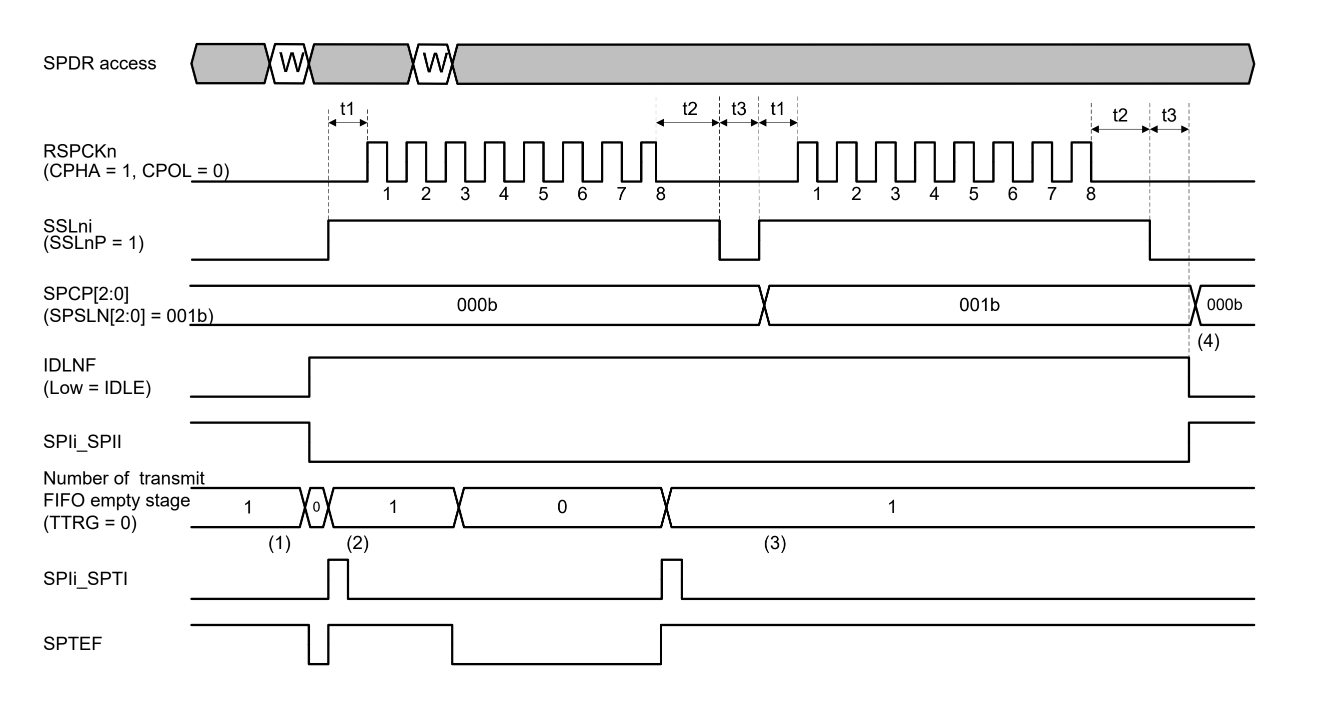


Figure 43.34 Example of Idle Interrupt Operation (Master mode / Motorola-SPI)

1. At the start of transmission, if the next transfer data is not set in the transmission buffer, the IDLNF flag is 0 (IDLE). Writing transmit data sets the IDLNF flag to 1 (BUSY). When the SPIIE bit in the SPI control register (SPCR) is set to 1 before transmit data is written, interrupt processing is required before transmission start. For this reason, set the SPIIE bit to 0 before starting transmission.
2. After transmission has started, the IDLNF flag remains 1 (BUSY) regardless of the transmit buffer state.
3. The SPCP[2:0] bits change the command to the next command at the end of t3 cycle. When the next command is not 000b, the IDLNF flag remains unchanged even when the next transmit data has not been written.
4. The IDLNF flag is cleared to 0 (IDLE) at the end of t3 cycle because the next command is 000b and the next transmit data is not present. When the SPIIE bit is 1 currently, an SPIi_SPII (i = 0, 1) interrupt is output.

[In the TI-SSP case]

Figure 43.35 shows an example of idle interrupt operation during normal operation.

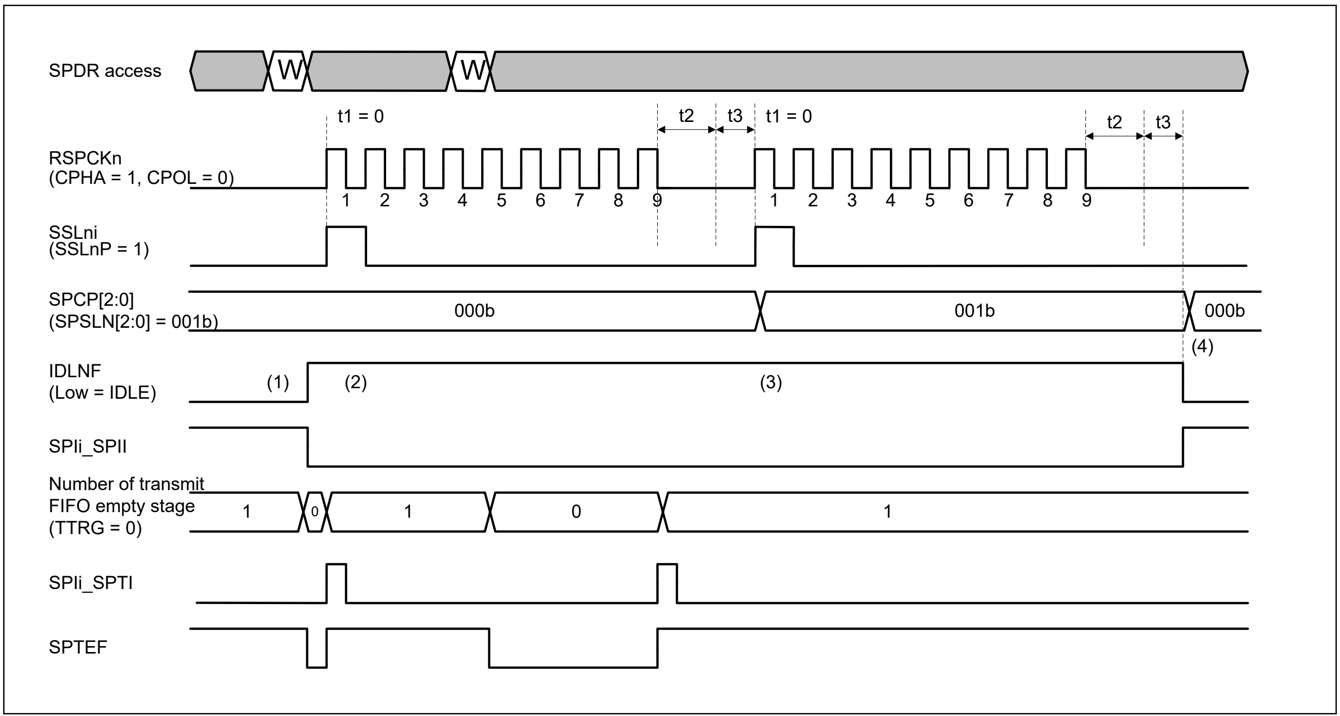


Figure 43.35 Example of Idle Interrupt Operation (Master mode / TI-SSP)

1. At the start of transmission, if the next transfer data is not set in the transmission buffer, the IDLNF flag is 0 (IDLE). Writing transmit data makes sets the IDLNF flag to 1 (BUSY). When the SPIIE bit in the SPI control register (SPCR) is set to 1 before transmit data is written, interrupt processing is required before transmission start. For this reason, set the SPIIE bit to 0 before starting transmission.
2. After transmission has started, the IDLNF flag remains 1 (BUSY) regardless of the transmit buffer state.
3. The SPCP[2:0] bits change the command to the next command at the end of t3 cycle. When the next command is not 000b, the IDLNF flag remains unchanged even when the next transmit data has not been written.
4. The IDLNF flag is cleared to 0 (IDLE) at the end of t3 cycle because the next command is 000b and the next transmit data is not present. When the SPIIE bit is 1 currently, an SPIi_SPII interrupt is output.

43.3.9 Communication End Interrupt

43.3.9.1 Transmit-Receive/Transmit in Master Mode

See the description of the CENDF bit in section 43.2.9. SPSR : SPI Status Register for the setting / clearing conditions of the communication completion flag during Transmit-Receive/Transmit-only in Master Mode.

[In the Motorola-SPI case]

Figure 43.36 shows an example of communication end interrupt operation during transmit-receive/transmit master mode.

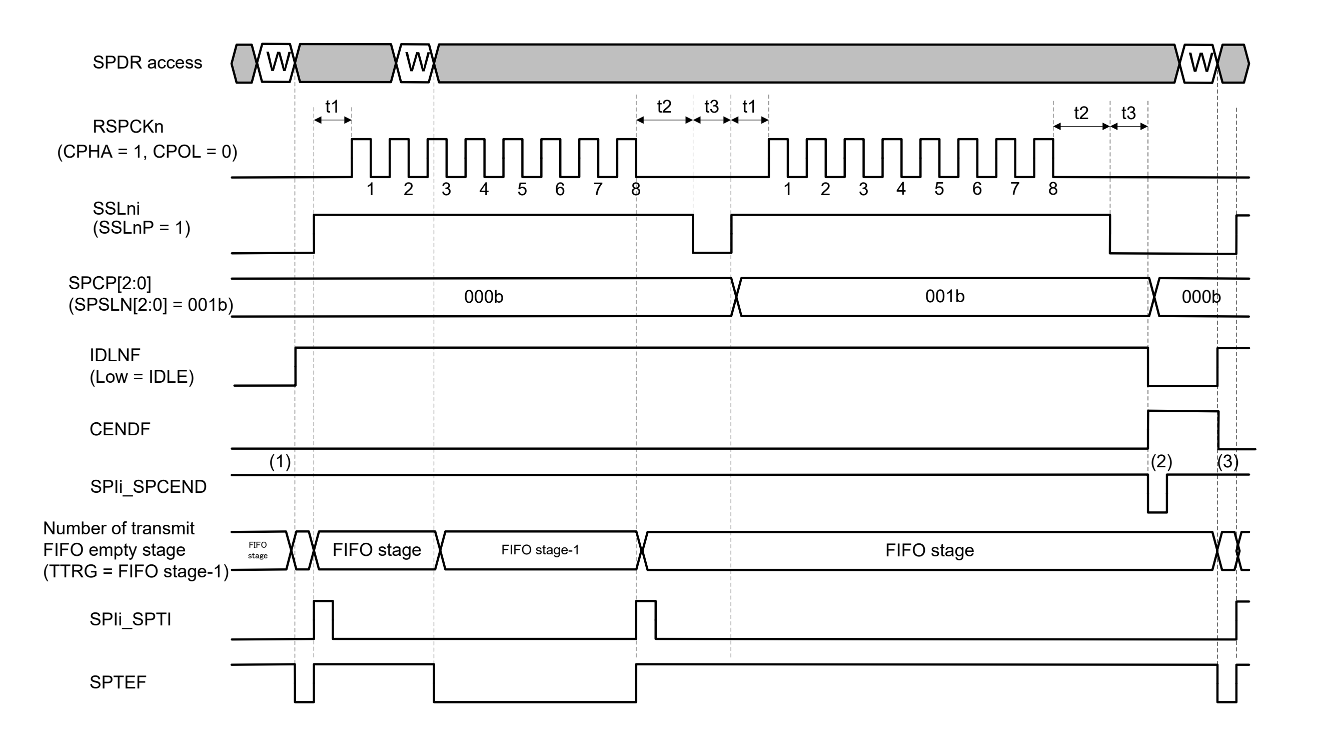


Figure 43.36 Example of Communication End Interrupt Operation (Transmit-Receive/Transmit Master mode/Motorola-SPI)

1. The CENDF flag is 0 and the level of SPIi_SPCEND (i = 0, 1) is 1 before communication start, and these have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the end of t3 cycle because the next command is 000b and there is no next transmit data, and then the SPIi_SPCEND interrupt output when the CENDIE bit is 1.
3. The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

[In the TI-SSP case]

Figure 43.37 shows an example of communication end interrupt operation during transmit-receive/transmit-only master mode.

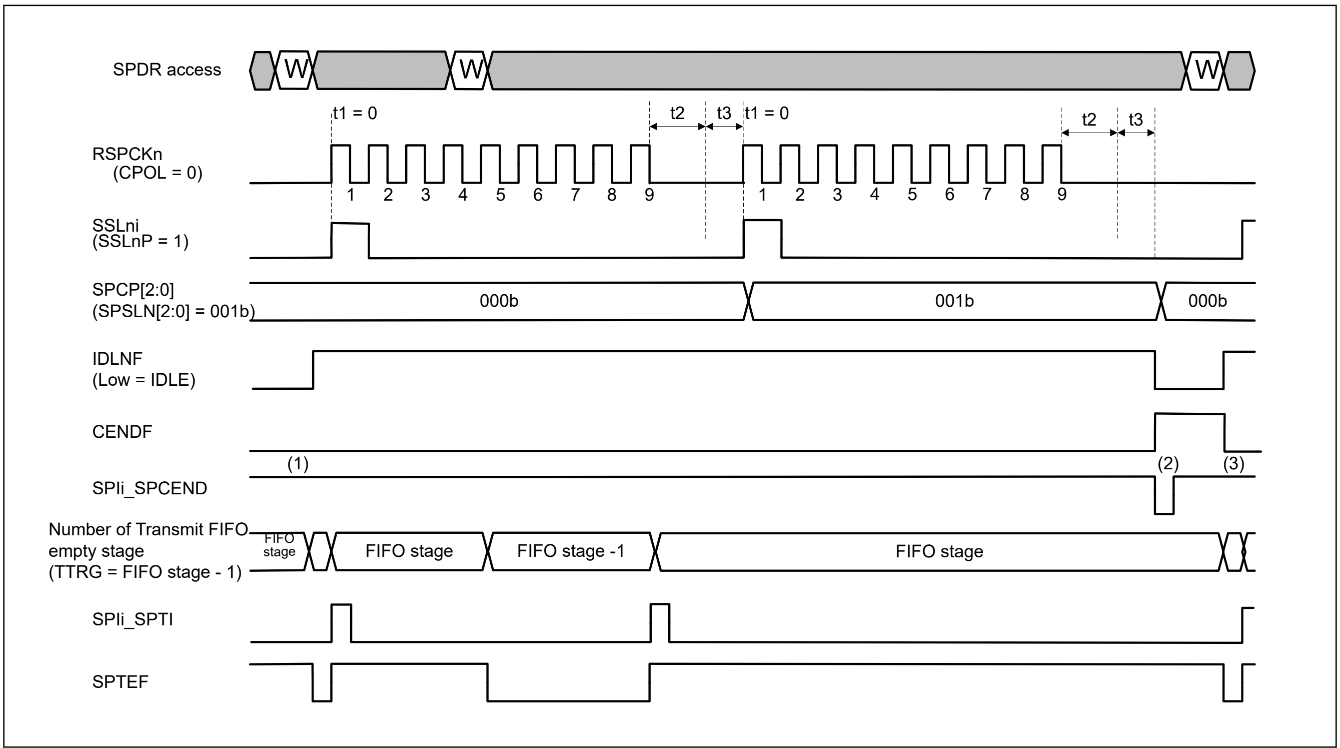


Figure 43.37 Example of Communication End Interrupt Operation (Transmit-Receive/Transmit-only Master mode/TI-SSP)

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start, and these have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the end of t3 cycle because the next command is 000b and there is no next transmit data, and then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

In slave mode operation, the output timing of the communication end interrupt is different due to the value of the SPCR.SPMS bit (SPI mode select bit), and the clear timing of the communication end interrupt is different due to the communication mode (transmit-receive or transmit-only or receive-only).

43.3.9.2 Receive-only in Master Mode

See the description of the CENDF bit in section 43.2.9. SPSR : SPI Status Register for the setting / clearing conditions of the communication completion flag during Receive-only in Master Mode.

Figure 43.38 shows an example of communication end interrupt operation during receive-only master mode at RMFM[4:0] = 0.

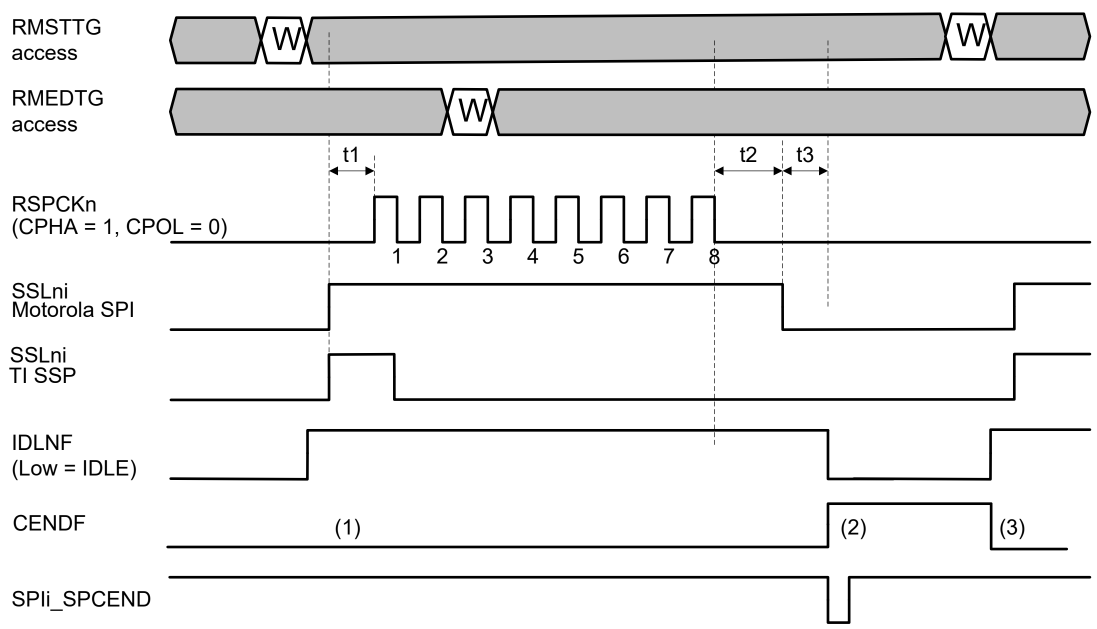


Figure 43.38 Example of Communication End Interrupt Operation (Receive-only Master mode / Motorola-SPI) at RMFM [4:0] = 0

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start. These have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the end of t3 cycle by writing 1 to RMEDTG during the communication frame. Then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared when writing 1 to RMSTTG. Also when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

Figure 43.39 shows an example of communication end interrupt operation during receive-only master mode at RMFM[4:0] ≠ 0.

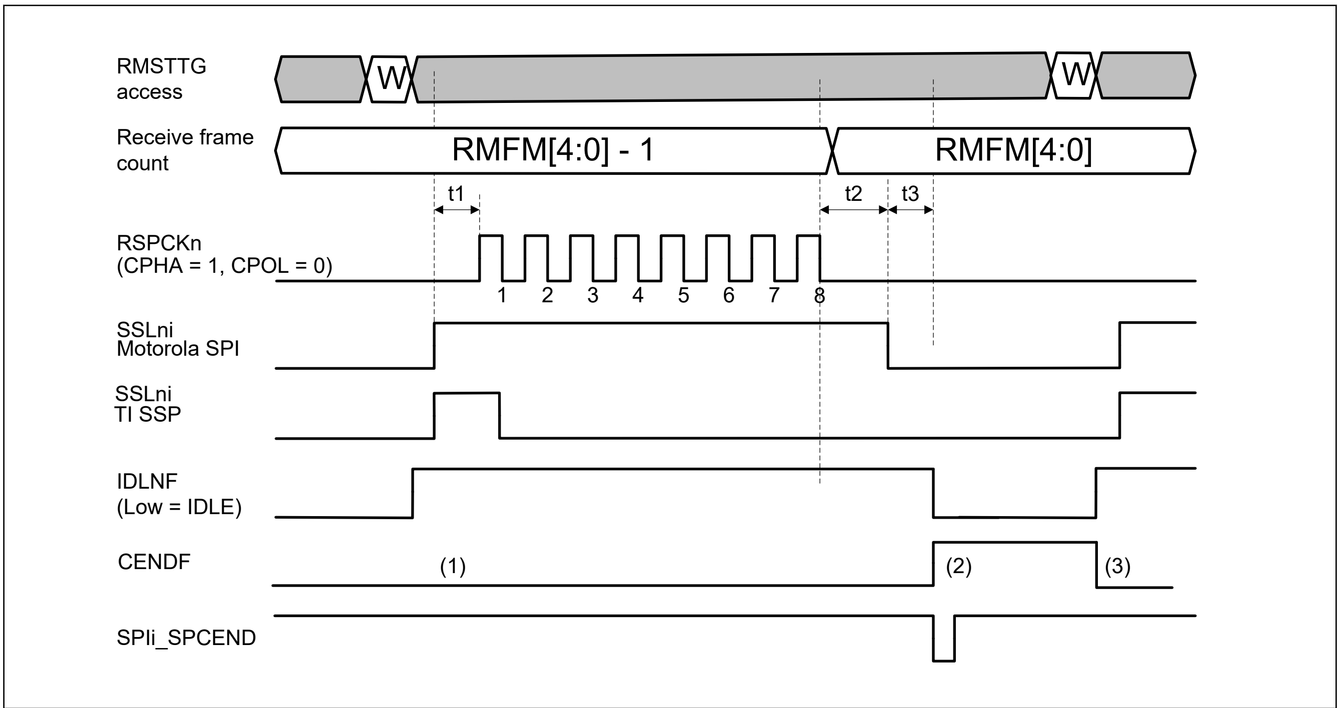


Figure 43.39 Example of Communication End Interrupt Operation (Receive-only Master mode / Motorola-SPI) at RMFM [4:0] ≠ 0

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start. These have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the end of t3 cycle after receiving the number of frames set by RMFM[4:0]. Then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared when writing 1 to RMSTTG. Also when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

43.3.9.3 Transmit-Receive/Transmit in Slave Mode on SPI Operation (4-wire)

See the description of the CENDF bit in section 43.2.9. SPSR : SPI Status Register for the setting / clearing conditions of the communication completion flag during Transmit-Receive/Transmit-only in Slave Mode (4-wire).

[In the Motorola-SPI case]

Figure 43.40 shows an example of communication end interrupt operation during transmit-receive/transmit slave mode on SPI operation.

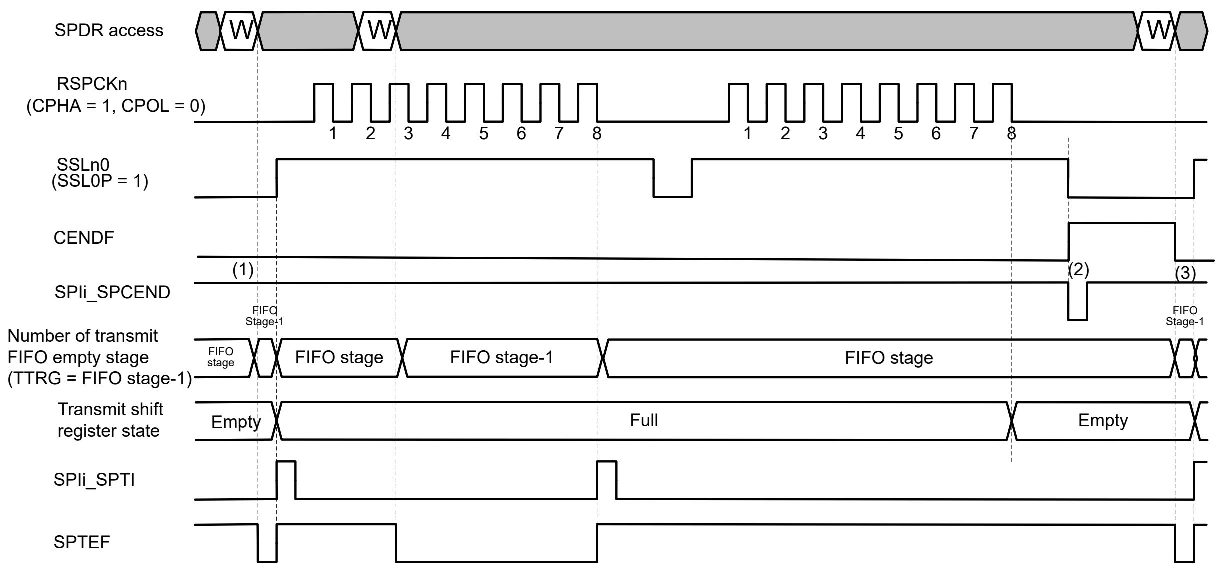


Figure 43.40 Example of Communication End Interrupt Operation (Transmit-Receive/Transmit Slave mode on SPI Operation/Motorola-SPI)

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start, and these have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the timing of SSLn0 negate when the next transfer data is not set in the transmit FIFO and the transmit shift register is empty. Then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

[In the TI-SSP case]

Figure 43.41 shows an example of communication end interrupt operation during transmit-receive/transmit-only slave mode on SPI operation.

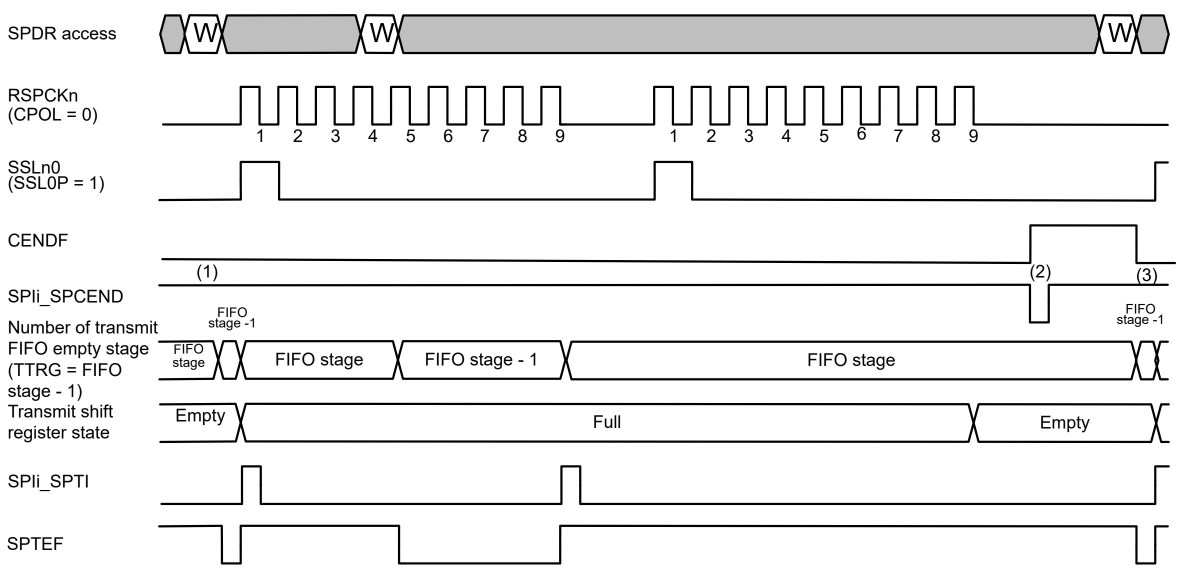


Figure 43.41 Example of Communication End Interrupt Operation (Transmit-Receive/Transmit-only Slave mode on SPI Operation / TI-SSP)

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start, and these have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the RSPCKn last data bit sampling when the next transfer data is not set in the transmit FIFO and the transmit shift register is empty. Then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

43.3.9.4 Receive Only in Slave Mode on SPI Operation (4-wire)

See the description of the CENDF bit in section 43.2.9. SPSR : SPI Status Register for the setting / clearing conditions of the communication completion flag during Receive-only in Slave Mode (4-wire).

[In the Motorola-SPI case]

Figure 43.42 shows an example of communication end interrupt operation during receive only slave mode on SPI operation (4-wire).

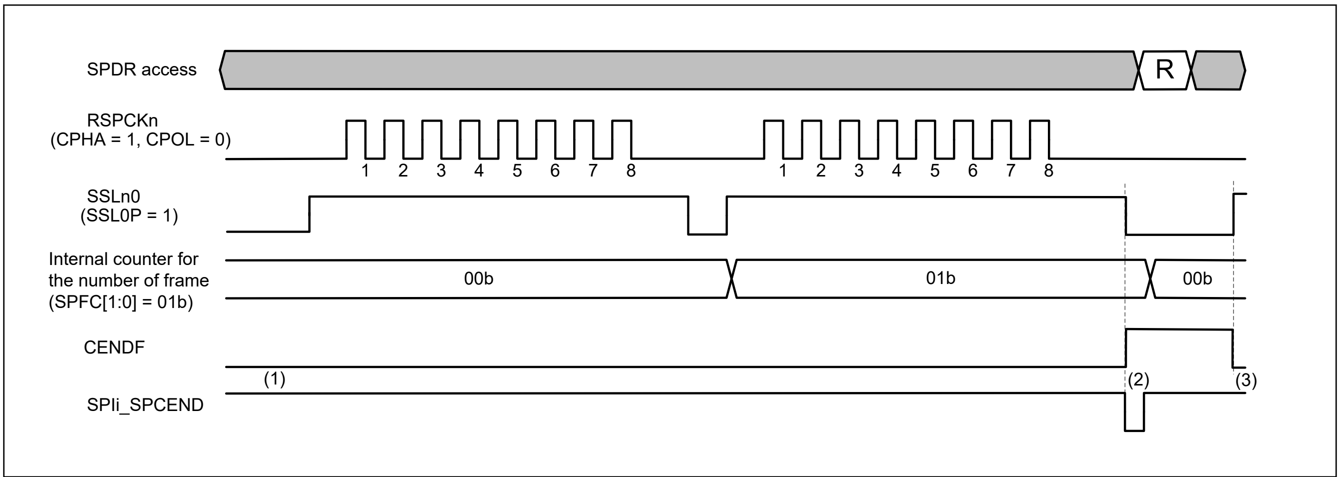


Figure 43.42 Example of Communication End Interrupt Operation (Receive only Slave mode on SPI Operation / Motorola-SPI)

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start, and these have kept during communication.
2. After the frames for SPFC set value in the SPI data control register (SPDCR) are stored in the receive buffer, the CENDF flag becomes 1 (communication completed) at the timing of SSLn0 negation. Then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared at the SSLn0 assert when the next transmission start. Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

[In the TI-SSP case]

Figure 43.43 shows an example of communication end interrupt operation during receive only slave mode on SPI operation (4-wire).

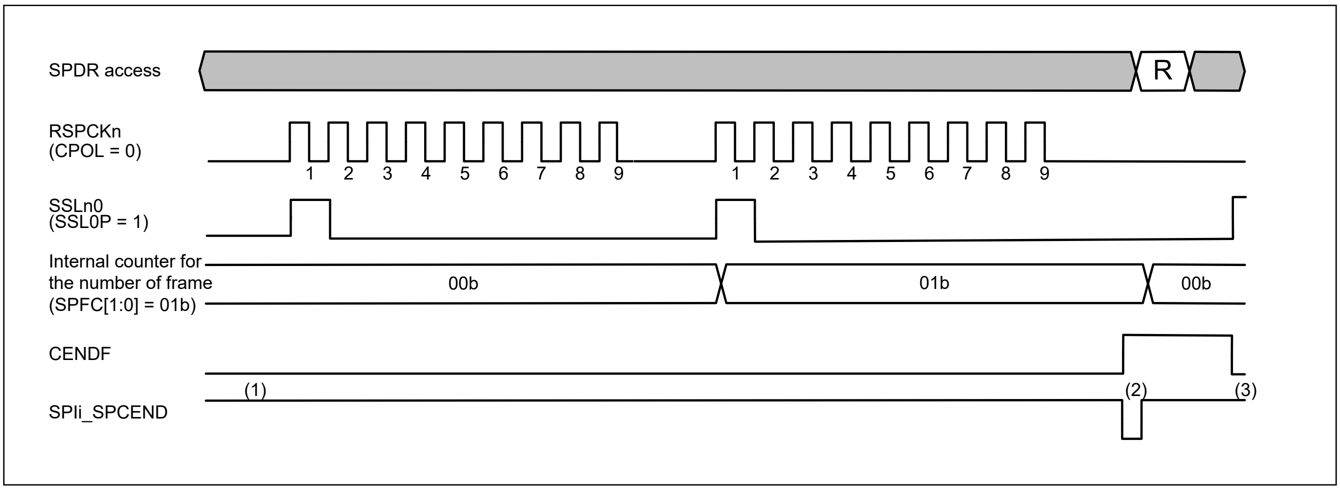


Figure 43.43 Example of Communication End Interrupt Operation (Receive-only Slave mode on SPI Operation / TI-SSP)

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start, and these have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the RSPCK last data bit sampling when the last frame transmission ends. Then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared at the SSLn0 assert when the next transmission start. Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

43.3.9.5 Transmit-Receive/Transmit in Slave Mode on Clock Synchronous Operation (3-wire)

See the description of the CENDF bit in section 43.2.9. SPSR : SPI Status Register for the setting / clearing conditions of the communication completion flag during Transmit-Receive/Transmit-only in Slave Mode on Clock Synchronous (3-wire).

Figure 43.44 shows an example of communication end interrupt operation during transmit-receive/transmit slave mode on clock synchronous operation (3-wire).

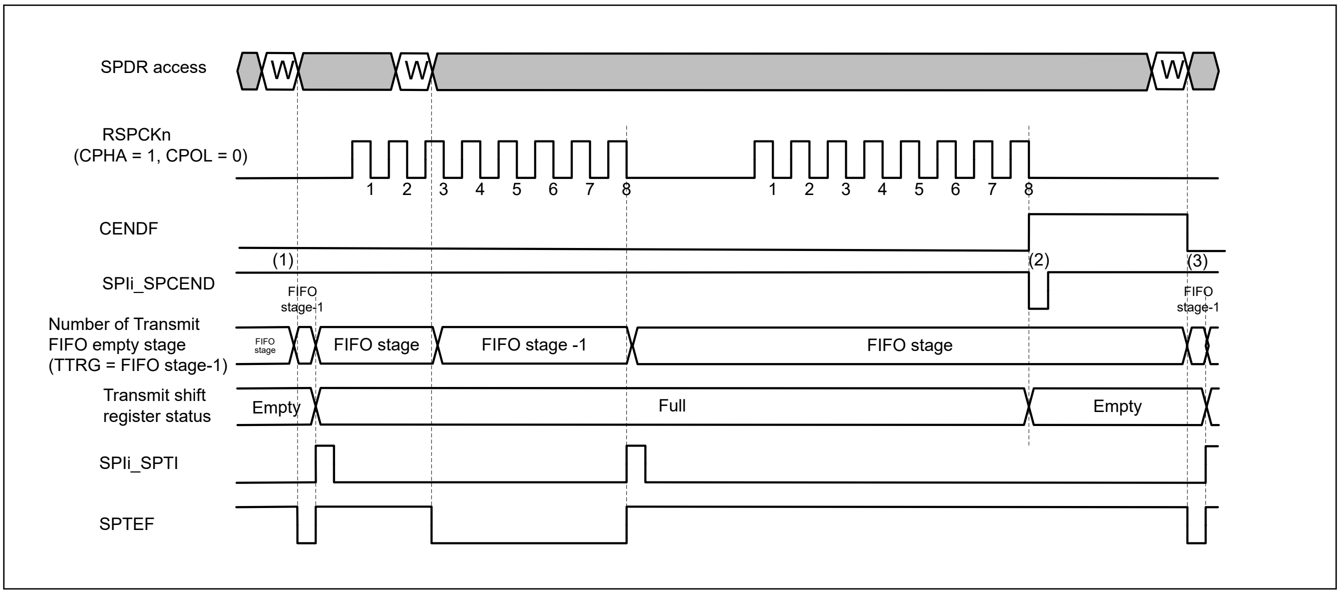


Figure 43.44 Example of Communication End Interrupt Operation (Transmit-Receive/Transmit Slave mode on Clock Synchronous Operation)

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start, and these have kept during communication.

2. When the next transfer data is not set in the transmit FIFO and the transmit shift register is empty, then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

43.3.9.6 Receive Only in Slave Mode on Clock Synchronous Operation (3-wire)

See the description of the CENDF bit in section 43.2.9. SPSR : SPI Status Register for the setting / clearing conditions of the communication completion flag during Receive -only in Slave Mode on Clock Synchronous (3-wire).

Figure 43.45 shows an example of communication end interrupt operation during receive only slave mode on clock synchronous operation.

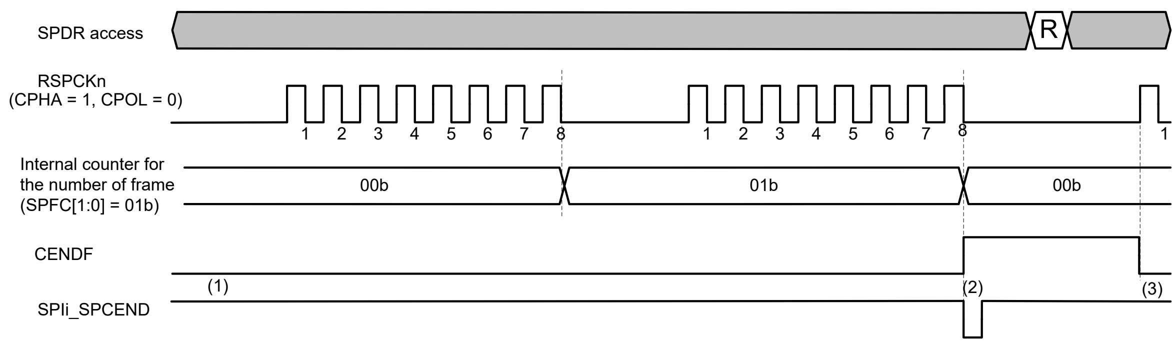


Figure 43.45 Example of Communication End Interrupt Operation (Receive-only Slave mode on Clock Synchronous Operation)

1. The CENDF flag is 0 and the level of SPIi_SPCEND is 1 before communication start, and these have kept during communication.
2. The CENDF flag is set to 1 (communication completed) at the timing of the last data bit sampling of RSPCKn in the last frame communication when the last frame of the SPI data control register (SPDCR) SPFC set value is received. Then the SPIi_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared at the first edge of RSPCKn for the next transmission. Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

43.3.9.7 Common Operation

In this chapter, the operation common to each mode / area option communication in section 43.3.9.1. Transmit-Receive/ Transmit in Master Mode to section 43.3.9.6. Receive Only in Slave Mode on Clock Synchronous Operation (3-wire) is explained. When the enable of SPI communication end interrupt (CENDIE) is 0, at the time of communication completion, a flag of communication end (CENDF) is set and an event of communication end is output, but no interrupt is output. However, if the enable of communication end interrupt (CENDIE) is set to 1 before clearing the flag of communication end (CENDF) while the enable of SPI function (SPE) is 1, the communication end interrupt (SPIi_SPCEND) is output.

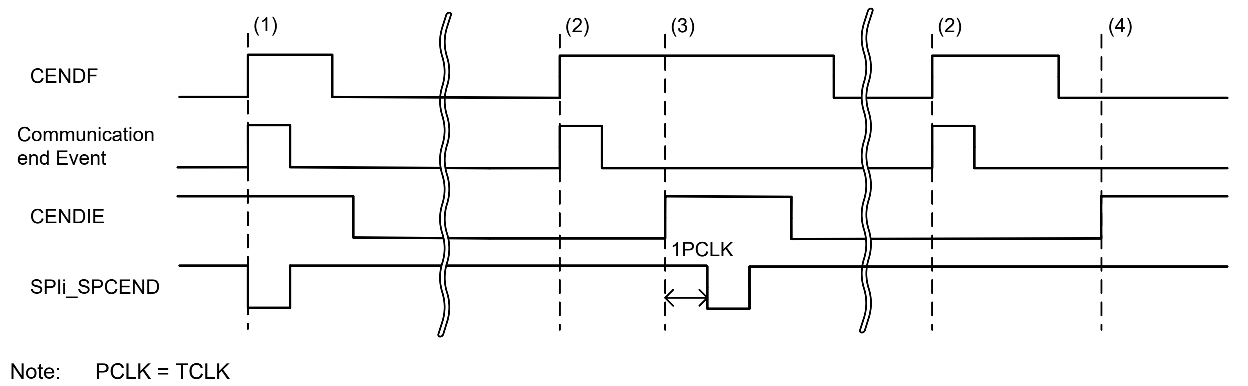


Figure 43.46 Example of Communication End Interrupt Operation (Enable control)

1. When the enable of SPI communication end interrupt (CENDIE) is 1, at the time of communication completion, the following three are the same timing.
   - A flag of communication end (CENDF)
   - An event of communication end
   - The communication end interrupt (SPIi_SPCEND)
2. When the enable of SPI communication end interrupt (CENDIE) is 0, at the time of communication completion, the following two are the same timing, but no interrupt.
   - A flag of communication end (CENDF)
   - An event of communication end
3. After (2), if the enable of communication end interrupt (CENDIE) is set when the enable of SPI function (SPE) and the flag of communication end (CENDF) are 1, the communication end interrupt (SPIi_SPCEND) is output after 1 TCLK.
4. After (2), even if the enable of communication end interrupt (CENDIE) is set when the enable of SPI function (SPE) or the flag of communication end (CENDF) is 0, the communication end interrupt (SPIi_SPCEND) is not output.

43.3.10 Error Detection

In normal SPI serial transfers, data written to the transmit buffer of SPDR is transmitted, and received data can be read from the receive buffer of SPDR. If access is made to SPDR, an abnormal transfer might occur, depending on the status of the transmit or receive buffer or the status of the SPI at the beginning or end of serial transfer.

If an abnormal transfer occurs, the SPI detects the event as an underrun error, overrun error, parity error, or mode fault error. Table 43.9 lists the relationship between non-normal transfer operations and the SPI error detection function.

Table 43.9 Relationship between non-normal transfer operations and SPI error detection (1 of 2)

Operation	Occurrence condition	SPI operation	Error detection
1	SPDR is written while no empty stages in the transmit FIFO.	• The contents of the transmit buffer are kept • Write data is missing	None
2	SPDR is read while no data stored in receive FIFO.	The contents of the receive buffer and previously received data are output.	None
3	Serial transfer is started in slave mode when the SPI is not able to transmit data.	• Serial transfer is suspended • Transmit or receive data is missing • Driving of the MISOn output signal is stopped • SPI function is disabled	Underrun error
4	Serial transfer ends when data is stored in the receive FIFO for the number of FIFO stages.	• Keeps the contents of the receive FIFO • Missing receive data	Overrun error

Table 43.9 Relationship between non-normal transfer operations and SPI error detection (2 of 2)

Operation	Occurrence condition	SPI operation	Error detection
5	An incorrect parity bit is received during full-duplex synchronous serial communication with the parity function enabled in following mode: ● Transmit-receive master mode ● Receive-only master mode ● Transmit-receive slave mode ● Receive-only slave mode	The parity error flag is asserted	Parity error
6	The SSLn0 input signal is asserted when the serial transfer is idle in multi-master mode.	● Driving of the RSPCKn, MOSIn, SSLn1 to SSLn3 output signals is stopped ● SPI function is disabled	Mode fault error
7	The SSLn0 input signal is asserted during serial transfer in multi-master mode.	● Serial transfer is suspended ● Transmit or receive data is missing ● Driving of the RSPCKn, MOSIn, SSLn1 to SSLn3 output signals is stopped ● SPI function is disabled	Mode fault error
8	[In the Motorola-SPI case] The SSLn0 input signal is negated during serial transfer in slave mode.	● Serial transfer is suspended ● Transmit or receive data is missing ● Driving of the MISOn output signal is stopped ● SPI function is disabled	Mode fault error
9	[In the TI-SSP case] The SSLn0 input signal is asserted during serial transfer in slave mode.	● Serial transfer is suspended ● Transmit or receive data is missing ● Driving of the MISOn output signal is stopped ● SPI function is disabled	Mode fault error
10	After data is stored in the receive FIFO with SPDRES = 1, the number of stored data is less than the threshold value and no receive data is written for the set value of SPDRC [7:0]	Assert the receive data ready flag	Receive data ready

In operation 1 described in Table 43.9, the SPI does not detect an error. To prevent data omission during writes to SPDR, the writes to SPDR must be executed using a transmit buffer empty interrupt request (when the SPSR.SPTEF flag is 1).

Similarly, the SPI does not detect an error in operation 2. To prevent extraneous data from being read, SPDR read must be executed with an SPI receive buffer full interrupt request (when the SPSR.SPRF flag is 1).

For information on the other errors, see the following sections:

- Underrun error, indicated in operation 3, see section 43.3.10.4. Underrun Errors
- Overrun error, indicated in operation 4, see section 43.3.10.1. Overrun Errors
- Parity error, indicated in operation 5, see section 43.3.10.2. Parity Errors
- Mode fault error, indicated in operations 6 to 9, see section 43.3.10.3. Mode Fault Errors
- For the transmit and receive interrupts, see section 43.3.7. Transmit Buffer Empty and Receive Buffer Full Interrupts.
- For the reception data ready in operations 10, see section 43.3.10.5. Received Data Ready.

43.3.10.1 Overrun Errors

If a serial transfer ends when the receive buffer of SPDR is full, the SPI detects an overrun error and sets the SPSR.OVRF flag to 1. When the OVRF flag is 1, the SPI does not copy data from the shift register to the receive buffer, so the data prior to the error occurrence is retained in the receive buffer. To set the OVRF flag to 0, issue a system reset or 1 is written to the SPSRC.OVRFC bit.

Figure 43.47 shows an example of operation of the OVRF and SPRF flags. The SPSRC and SPDR accesses shown in Figure 43.47 indicate the condition of accesses to the SPSRC and SPDR register, where W denotes a write cycle, and R a read cycle. In this example, the SPI performs an 8-bit serial transfer when SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

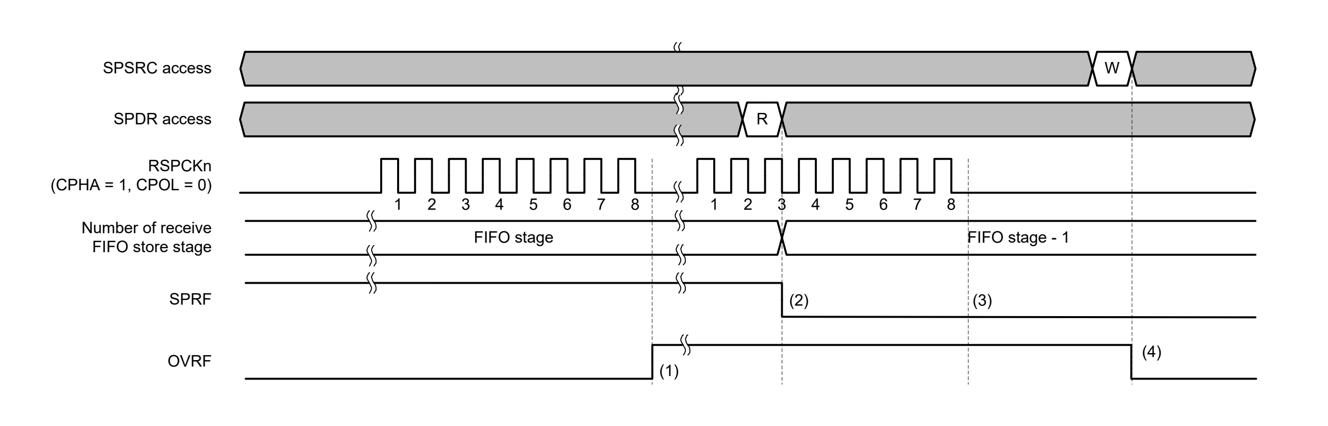


Figure 43.47 Operation example of the OVRF and SPRF flags

The operation of the flags at timings (1) to (4) in Figure 43.47 is as follows:

1. When serial transfer ends while data is stored for the number of FIFO stages, the SPI detects an overrun error and sets the OVRF flag to 1. The SPI does not copy shift register data to the receive buffer. The SPI does not detect a parity error even when SPPE = 1. In master mode, the SPI copies the value of pointer to the SPI command register (SPCMDm) to the SPECM[2:0] bits in the SPI status register (SPSR).
2. When SPDR is read, the SPI outputs the data in the receive buffer. At this time, the SPRF flag is cleared to 0 at the last access when the received data is read from SPDR in one processing routine using DTC / DMAC.
3. If the serial transfer ends with the OVRF flag set to 1 (overrun error occurred), the SPI does not copy data in the shift register to the receive buffer (the SPRF flag does not set to 1). A receive buffer full interrupt is not generated. Even when the SPPE bit is 1, parity errors are not detected. In an overrun error state when the SPI does not copy the received data from the shift register to the receive buffer, on termination of the serial transfer, the SPI determines that the shift register is empty. This enables data transfer from the transmit buffer to the shift register.
4. When 1 is written to the SPSRC.OVRFC bit, the SPSR.OVRF flag is cleared.

The occurrence of an overrun can be checked either by reading SPSR or by using an SPI error interrupt and reading SPSR. When executing a serial transfer, you must ensure that overrun errors are detected early, for example by reading SPSR immediately after SPDR is read.

If an overrun error occurs and the OVRF flag sets to 1, normal reception operations cannot be performed until the OVRF flag is set to 0.

When the RSPCK auto-stop function is enabled (SPCR.SCKASE = 1) in master mode, an overrun error does not occur. Figure 43.48 and Figure 43.49 show the clock stop waveform when a serial transfer continues while the receive buffer is full in master mode.

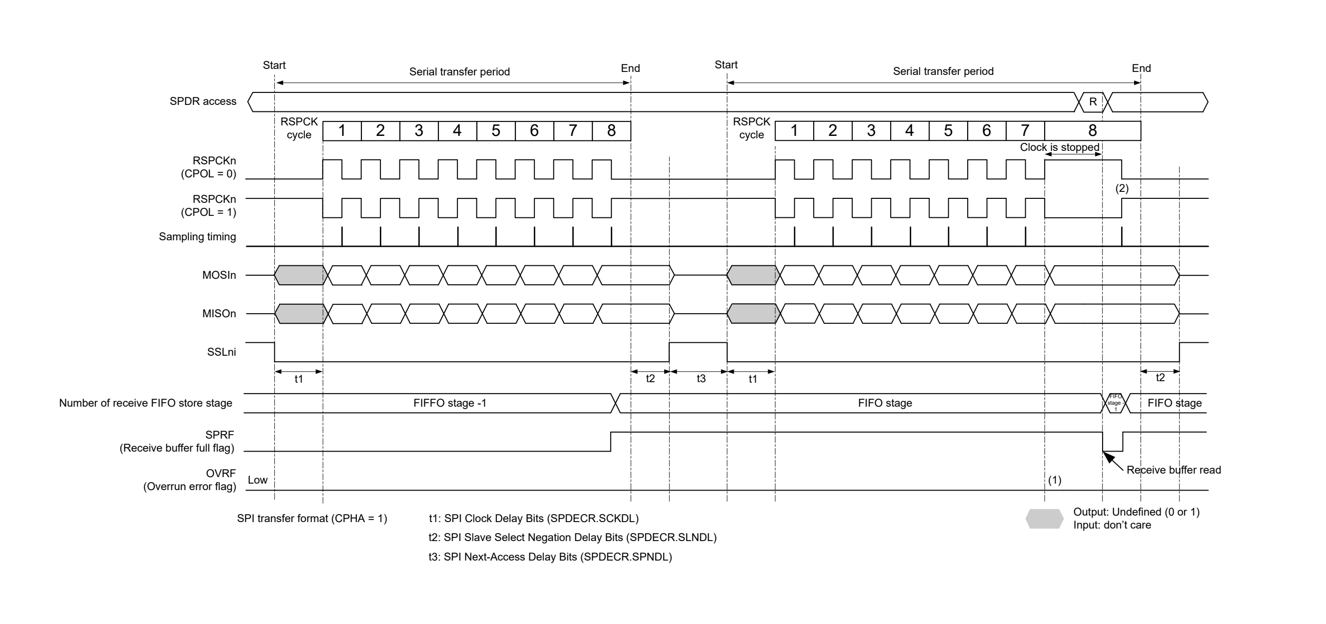


Figure 43.48 Clock stop waveform when serial transfer continues with data is stored for the number of FIFO stages in master mode (CPHA = 1)

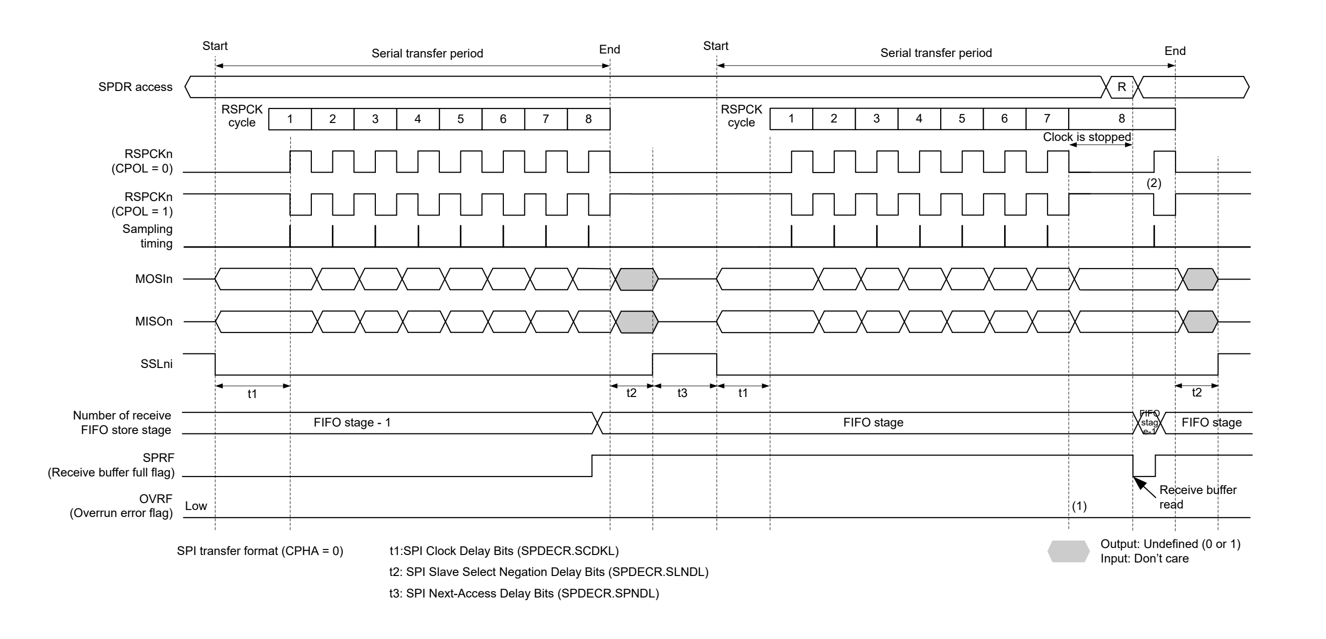


Figure 43.49 Clock stop waveform when serial transfer continues with data is stored for the number of FIFO stages in master mode (CPHA = 0)

The operation of the flags at timings (1) and (2) in Figure 43.48 and Figure 43.49 is as follows:

1. While data is stored in the receive FIFO for the number of FIFO stages, the RSPCK clock is deactivated and no overrun error occurs.
2. If SPDR is read while the clock is stopped, data in the receive buffer can be read. The RSPCK clock restarts.

Overrun error does not occur when RSPCK automatic stop function is enabled for transfer with no delay of between frames during burst transfer in master mode. Figure 43.50 and Figure 43.51 show the clock stop waveform, when there is no delay between frames at burst transfer and the serial transfer continues in the data is stored in the receive FIFO for the number of FIFO stages.

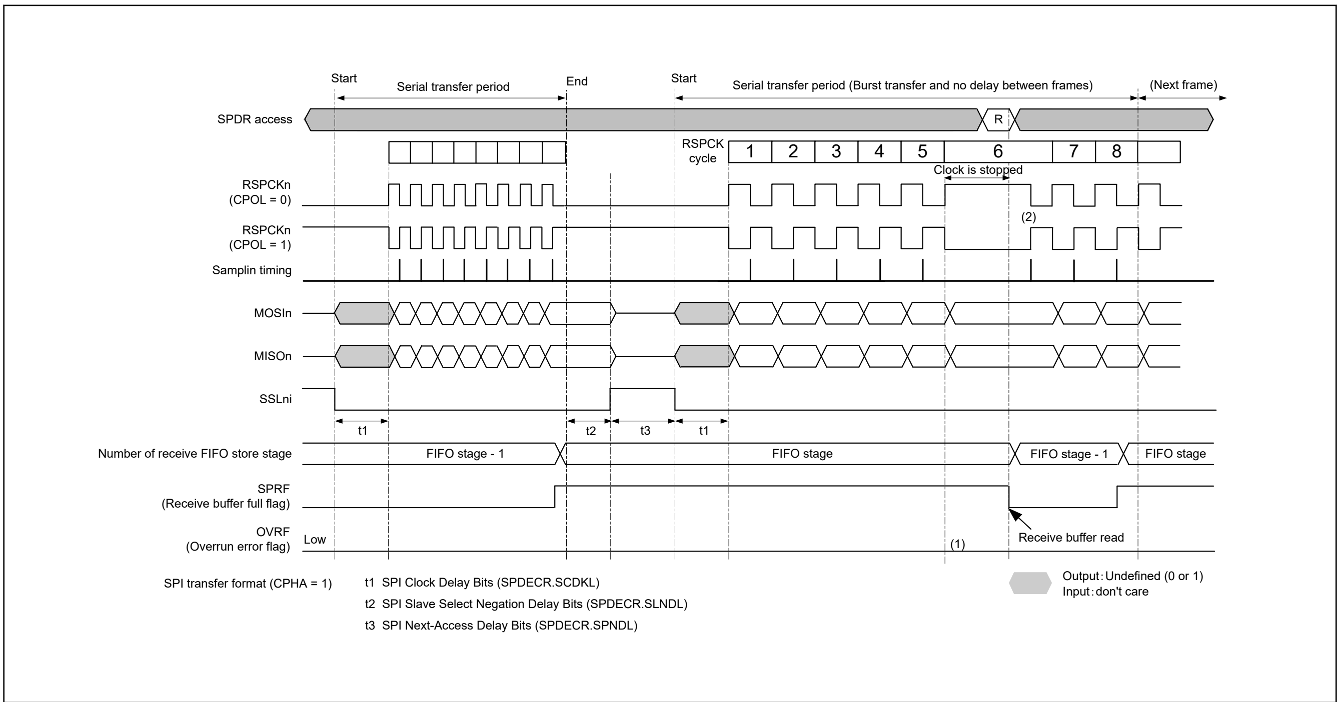


Figure 43.50 Clock Stop Waveform when Serial Transfer Continues in the Receive Buffer Full with data is stored for the number of FIFO stages (at burst transfer and no delay between frames CPHA = 1)

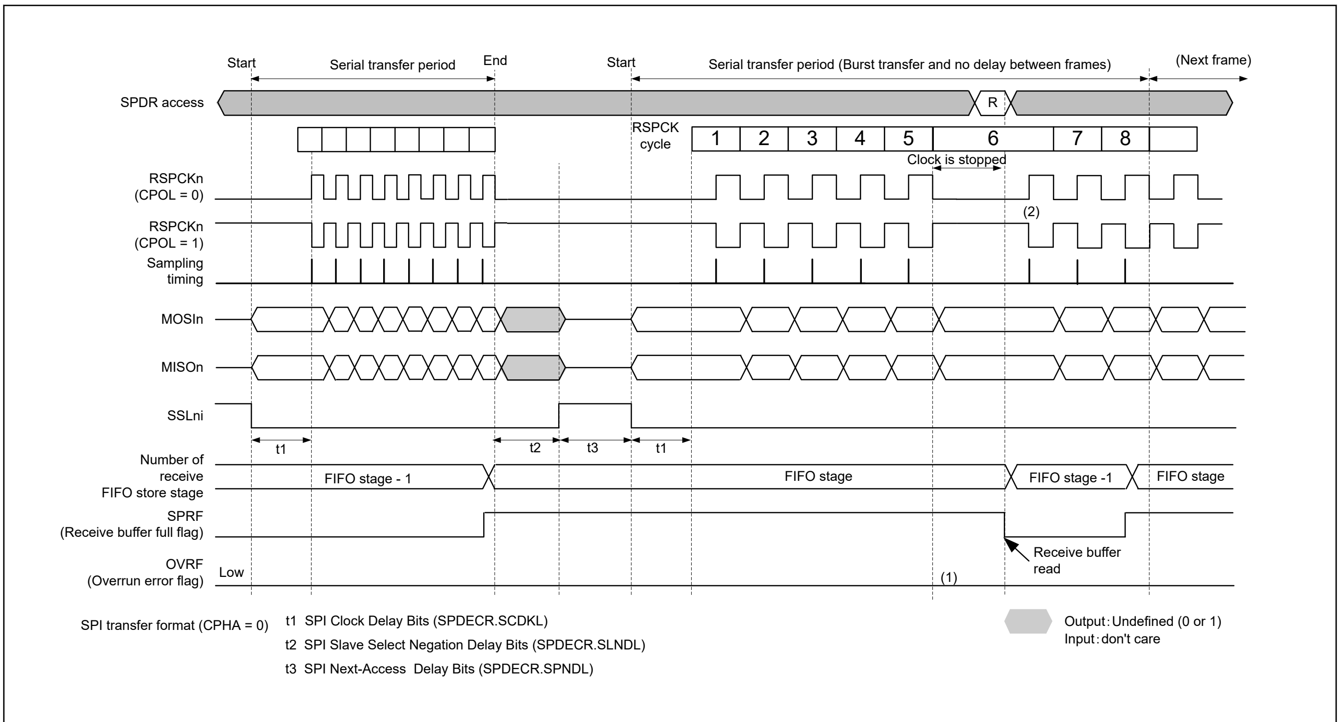


Figure 43.51 Clock Stop Waveform when Serial Transfer Continues with data is stored for the number of FIFO stages in Master Mode (at burst transfer and no delay between frames CPHA = 0)

The following describes operation of flags at timings (1) and (2) in the figure above.

1. While the data is stored for the number of FIFO stages, the RSPCK clock is deactivated and no overrun error occurs.
2. Receive buffer data can be read by reading SPDR during clock stop. After the receive buffer data has been read the RSPCK clock restarts.

43.3.10.2 Parity Errors

After transfer in transmit-receive or receive-only master mode, transmit-receive slave mode or receive only slave mode while the SPPE bit in the SPI control register (SPCR) is 1, the SPI checks occurrence of a parity error. When the SPI detects a parity error in received data, the PERF flag in the SPI status register (SPSR) is set to 1. While the OVRF flag is 1, the SPI does not copy shift register data to the receive buffer. Therefore, parity error in received data is not detected. To clear the PERF flag in SPSR to 0, issue a system reset or 1 is written to the SPSRC.PERFC bit.

Figure 43.52 shows an example of operation of the OVRF and PERF flags. The SPSR access shown in Figure 43.52 indicates the condition of access to the register, where W denotes a write cycle, and R a read cycle. In this example, full-duplex serial communication is performed while the SPCR.SPPE bit is 1. The SPI performs an 8-bit serial transfer when SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

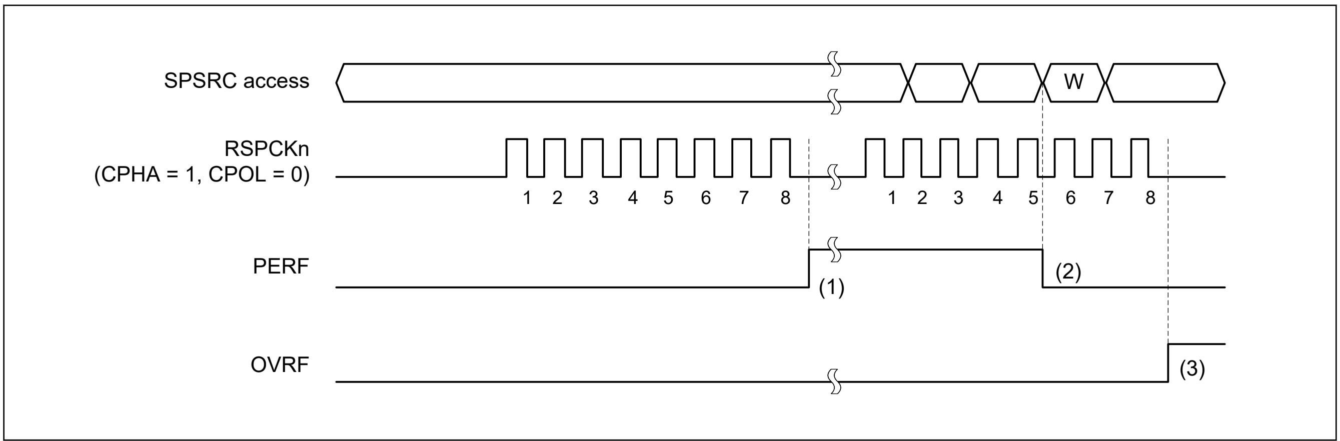


Figure 43.52 Operation example of the OVRF and PERF flags

The operation of the flags at timings (1) to (3) in Figure 43.52 is as follows:

1. When the SPI does not detect an overrun error and terminates the serial transfer, the SPI copies shift register data to the receive buffer. When the SPI checks the received data and detects a parity error at this time, the PERF flag is set to 1. In master mode, the SPI copies the value of pointer to the SPI command register (SPCMDm) to the SPECM[2:0] bits in the SPI Data control register 2 (SPDCR2).
2. When 1 is written to the SPSRC.PERFC bit, then clear the PERF flag.
3. When the SPI detects an overrun error and serial transfer is terminated, the data in the shift register is not copied to the receive buffer. The SPI does not perform parity error detection at this time.

Parity errors can be checked for by either reading the SPSR register or using an SPI error interrupt and reading the SPSR register. When executing a serial transfer, such checks are required to ensure early detection of parity errors. When the SPI is used in master mode, the pointer value to the SPCMDm register at the occurrence of the error can be checked by reading the SPDCR2.SPECM[2:0] bits.

43.3.10.3 Mode Fault Errors

The SPI operates in multi-master mode when the SPCR.MSTR bit is 1, the SPCR.SPMS bit is 0, and the SPCR.MODFEN bit is 1.

If the active level is input for the SSLn0 input signal of the SPI in multi-master mode, the SPI detects a mode fault error regardless of the status of the serial transfer, and sets the SPSR.MODF flag to 1.

On detecting the mode fault error, the SPI copies the value of the pointer to SPCMD to the SPECM[2:0] bits.

The active level of the SSLn0 signal is determined by the SPCR3.SSL0P bit.

When the MSTR bit is 0, the SPI operates in slave mode.

When the SPCR.MODFEN bit = 1 and the SPMS bit = 0 in slave mode, if the SSLn0 input signal is negated during the serial transfer period (from valid data drive start to final valid data latch), the SPI detects a mode fault error, while any of the following 2 conditions is met.

[In the Motorola-SPI case]

When the SSLn0 input signal is negated while serial data transfer.

[In the TI-SSP case]

When the SSLn0 input signal is asserted while serial data transfer. However, during burst transfer, no error is detected even if the SSLn0 input signal is asserted during the last bit of frame.

When the SPI detects a mode fault error, it stops driving output signals and clears the SPE bit in the SPCR register. When the SPE bit is cleared, the SPI function is disabled (as described in section 43.3.12. SPI Operation). In a multi-master configuration, the mastership can be released by stopping driving output signals and disabling the SPI function by using a mode fault error.

Whether a mode fault error is present can be checked by reading SPSR or by reading an SPI error interrupt and SPSR. To detect a mode fault error without using an SPI error interrupt, poll SPSR. When the SPI is used in master mode, the pointer value to SPCMD when an error is present can be checked by reading the SPECM[2:0] bits in SPSR.

While the MODF flag = 1, the SPI ignores writing 1 to the SPE bit. To enable the SPI function after a mode fault error is detected, clear the MODF flag to 0 without fail.

43.3.10.4 Underrun Errors

While the SPI is operating in slave mode (SPCR.MSTR bit = 0) and the communication mode select bit (TXMD[1:0]) in the SPI control register (SPCR) is set to 00b or 01b, if serial transfer is started before transmit data output is ready with the SPCR.SPE bit set to 1 (SPI function enabled), the SPI detects an underrun error and sets the SPSR.MODF and SPSR.UDRF flags to 1.

On detecting an underrun error, the SPI stops the driving of output signals and clears the SPCR.SPE bit to 0 (see section 43.3.11. Initializing the SPI).

The occurrence of underrun errors can be checked either by reading SPSR or by using an SPI error interrupt and reading SPSR. Detecting underrun errors without using the SPI error interrupt requires polling of SPSR.

When the MODF flag is 1, writing 1 to the SPE bit is ignored by the SPI. To enable the SPI function after the detection of an underrun error, the MODF flag must be set to 0.

43.3.10.5 Received Data Ready

When SPCR.TXMD[1:0] = 00b, 01b, or 11b, and SPCR2.SPDRC[7:0] ≠ 0x00, after receiving data in the receive FIFO during communication (SPE = 1), SPSR.SPDRF flag is set to 1 when the received data is not stored even after the number of received FIFOs is equal to or less than ≤ the threshold value and the value set in SPDRC[7:0] has elapsed.

When the receive data ready is detected, the interrupt and event link output can be selected as SPIi_SPRI or SPIi_SPEI with the SPCR.SPDRES bit.

Figure 43.53 shows an example of reception data ready detection operation.

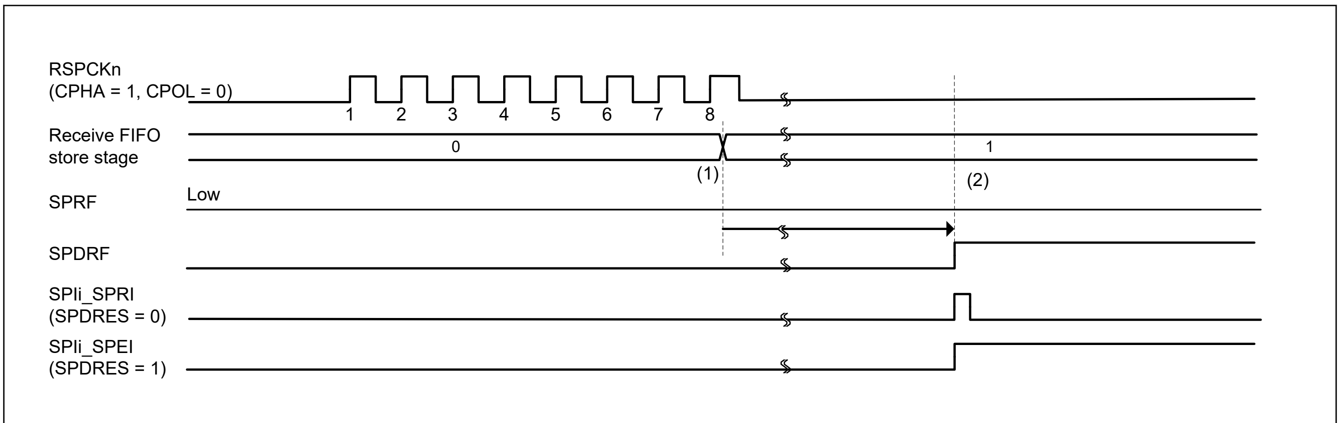


Figure 43.53 Received data ready

The following describes the operation at the timings indicated by (1) and (2) in the figure.

(1) Store the received data in the receive FIFO. SPRF is 0, because receive FIFO store stage ≤ number of frames set by SPDCR2.RTRG[1:0].

(2) Set SPDRF and assert SPIi_SPRI or SPIi_SPEI because there is no writing to the receive FIFO for the amount of SPDRFC [7:0] set from above (1).

43.3.11 Initializing the SPI

If 0 is written to the SPCR.SPE bit or if the SPI sets the SPE bit to 0 because it detected a mode fault error or an underrun error, the SPI disables the SPI function and initializes some of the module functions. When a system reset is generated, the SPI initializes all of the module functions. This section describes initialization by clearing of the SPCR.SPE bit and by a system reset.

43.3.11.1 Initialization by Clearing of the SPCR.SPE bit

When the SPCR.SPE bit is set to 0, the SPI initializes by:

- Suspending any serial transfer that is being executed
- Stopping the driving of output signals (Hi-Z) in slave mode
- Initializing the internal state of the SPI
- Initializing the transmit buffer of the SPI (the SPSR.SPTEF flag sets to 1)

Initialization by clearing of the SPE bit does not initialize the control bits of the SPI. For this reason, the SPI can be started in the same transfer mode in use prior to initialization when the SPE bit is set to 1 again.

The SPSR.CENDF, SPSR.SPRF, SPSR.OVRF, SPSR.MODF, SPSR.PERF, and SPSR.UDRF flags are not initialized, and the value of the SPDCR2.SPECM[2:0] and SPDCR2.SPCP[2:0] bits are not initialized. Therefore, even after the SPI is initialized, data from the receive buffer can be read to check the communication completion status and the error status during an SPI transfer.

The transmit buffer is initialized to an empty state (the SPSR.SPTEF flag sets to 1). Therefore, if the SPCR.SPTIE bit is set to 1 after SPI initialization, a transmit buffer empty interrupt is generated. To disable any transmit buffer empty interrupts when the SPI is initialized, write 0 to the SPTIE bit simultaneously while writing 0 to the SPE bit.

43.3.11.2 Initialization by System Reset

A system reset completely initializes the SPI by initializing all SPI control bits, status bits, and data registers, in addition to the requirements described in section 43.3.11.1. Initialization by Clearing of the SPCR.SPE bit.

43.3.12 SPI Operation

43.3.12.1 Master Mode Operation

The only difference between single- and multi-master mode operation is the use of mode fault error detection (see section 43.3.10. Error Detection). In single-master mode, the SPI does not detect mode fault errors whereas in multi-master mode, it does. This section explains operations that are common to both modes.

(1) Starting a serial transfer

When data is written to the SPI data register (SPDR) while the next transfer data is not set in the transmit FIFO, the SPI updates the transmit buffer (SPTXn, n = 0 to 3) data in SPDR. When the shift register is empty after the number of frames set in the SPDCR.SPFC[1:0] bits are written to the SPDR, the SPI copies data from the transmit buffer to the shift register and starts serial transfer. On copying transmit data to the shift register, the SPI changes the status of the shift register to full. On termination of the serial transfer, it changes the status of the shift register to empty. The status of the shift register cannot be referenced.

The polarity of the SSLni output pins depends on the SPCR3.SSLnP (n = 0 to 3) bits settings. For details on the SPI transfer format, see section 43.3.5. Transfer Formats.

(2) Terminating a serial transfer

[Except Receive-only in Master Mode]

After the SPI detects the RSPCKn edge corresponding to the final sampling timing regardless of the CPHA bit value in the SPI command register (SPCMD), the SPI terminates serial transfer. When the number of data stored in the receive FIFO is less than the number of FIFO stages, data is copied from the shift register to the receive buffer in the SPI data register (SPDR) after serial transfer.

The final sampling timing varies depending on the bit length of transfer data. In master mode, the SPI data length depends on the SPCMDm.SPB[4:0] bit settings. The polarity of the SSLni output pin depends on the SPCR3.SSLnP (n = 0 to 3) bits settings. For details on the SPI transfer format, see section 43.3.5. Transfer Formats.

[Receive-only in Master Mode]

When any of the following 2 conditions is met, then SPI terminating the serial transfer.

- After the SPI detects the RSPCKn edge corresponding to the final sampling timing regardless of the CPHA bit value in the SPI command register (SPCMD), the SPI terminates serial transfer.
- When writing SPCR2.RMEDTG = 1 during the serial transfer period, SPI terminating the serial transfer.

When the number of data stored in the receive FIFO is less than the number of FIFO stages, data is copied from the shift register to the receive buffer in the SPI data register (SPDR) after serial transfer.

The final sampling timing varies depending on the transfer data bit length. The data length of the SPI in master mode depends on the set value of the SPB[4:0] bits in the SPI command register (SPCMD). The SSLni output signal polarity depends on the set SPI SSLi signal polarity bit (SPCR3.SSLiP) (i = 0 to 3) value. For details about the SPI transfer format, see section 43.3.5. Transfer Formats.

(3) Sequence control

The transfer format in master mode is determined as follows.

The transfer format used in master mode is determined by the SPCR3, SPCMDm, and SPDECR registers.

The SPCR3.SPSLN[2:0] bits determine the sequence configuration for serial transfers that are executed by the SPI in master mode. The following items are set in the SPCMDm register:

- SSLni pin output signal value
- MSB- or LSB-first
- Data length
- Some of the bit rate settings
- RSPCKn polarity and phase
- Whether SPDECR.SCKDL is to be referenced
- Whether SPDECR.SLNDL is to be referenced
- Whether SPDECR.SPNDL is to be referenced

SPCR3.SPBR holds some of the bit rate settings, including SPDECR.SCKDL (SPI clock delay), SPDECR.SLNDL (SSL negation delay), and SPDECR.SPNDL (next-access delay).

Based on the sequence length assigned in SPCR3.SPSLN, the SPI makes up a sequence comprised of a part or all of the SPCMDm register. The SPI contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPDCR2.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the SPI function is enabled, the SPI loads the pointer to the commands in SPCMD0, and incorporates the SPCMD0 settings into the transfer format at the beginning of serial transfer. The SPI increments the pointer each time the next-access delay period for a data transfer ends. On completion of the serial transfer that corresponds to the final command in the sequence, the SPI sets the pointer to SPCMD0, and in this way the sequence is executed repeatedly.

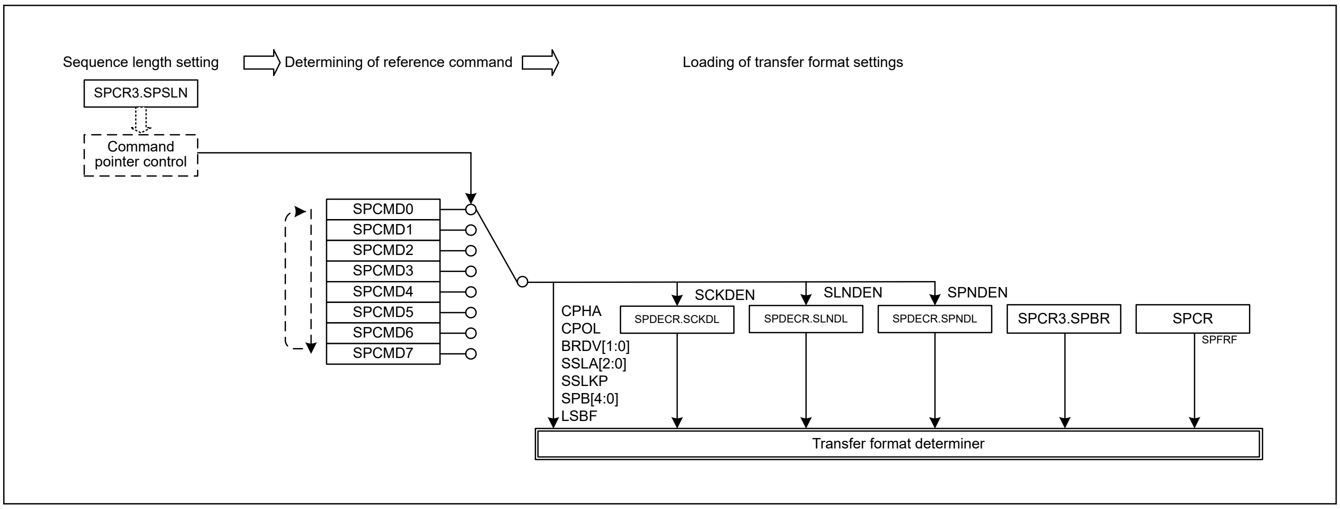


Figure 43.54 Procedure for determining the form of a serial transfer in master mode

In this section, a frame is the combination of the data in SPDR and the settings in SPCMDm.

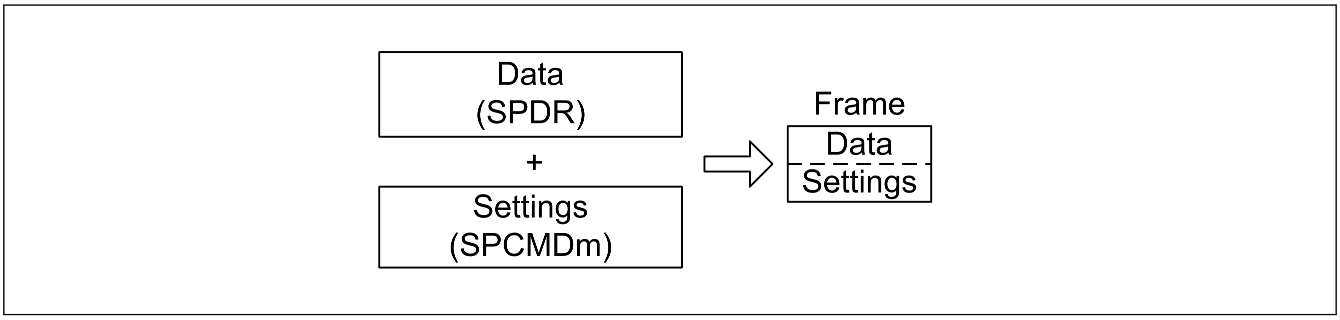


Figure 43.55 Conceptual diagram of frames

Figure 43.56 shows the correspondence between the commands and the transmit and receive buffers in the sequence of operations specified by the settings.

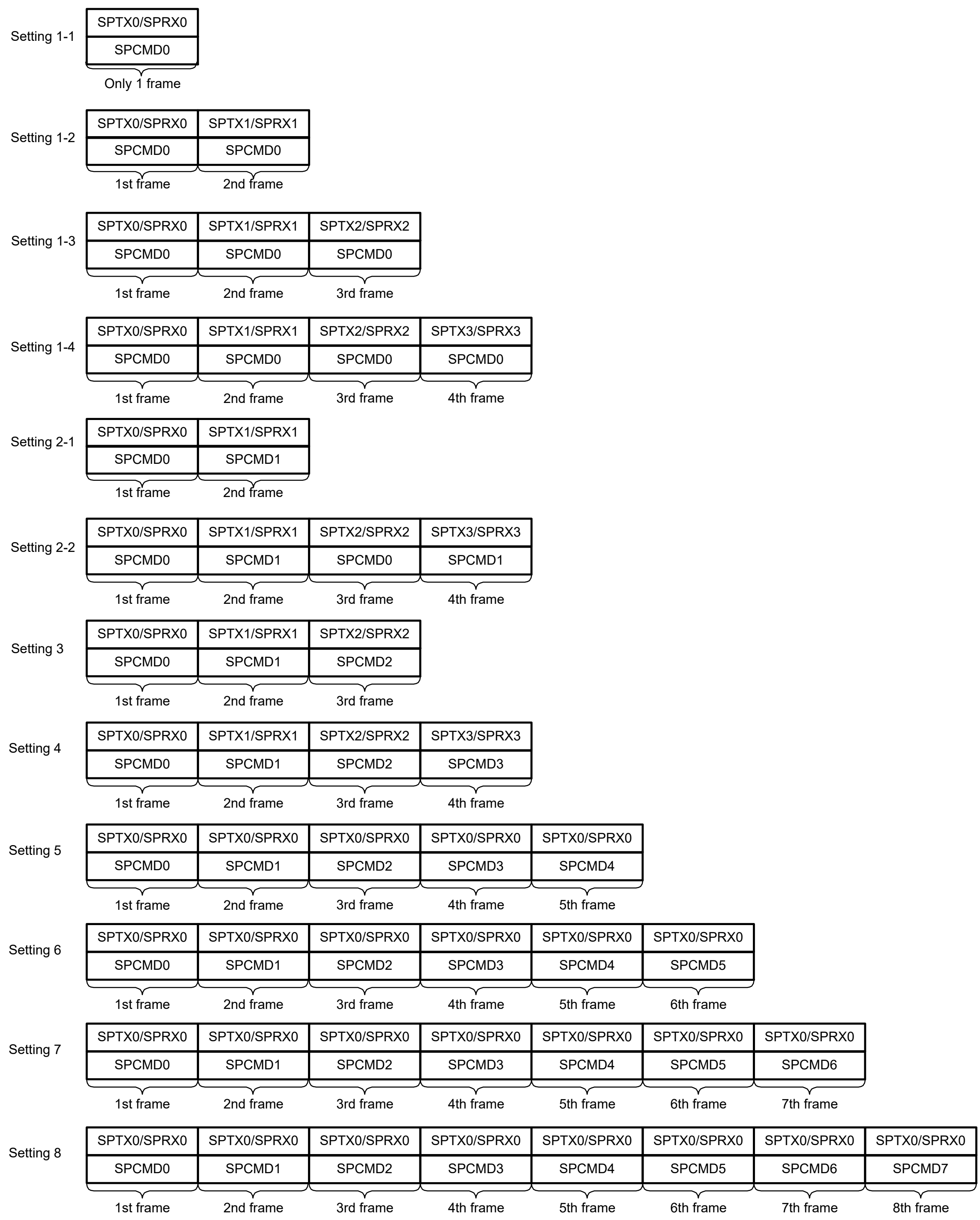


Figure 43.56 Correspondence between SPI Command Register and transmit and receive buffers in sequence operations

(4) Burst transfers

This section describes burst transfer during transmit-receive/transmit-only operation.

[In the Motorola-SPI case]

If the SPCMDm.SSLKP bit that the SPI references during the current serial transfer is 1, the SPI maintains the SSLni signal level during the serial transfer until the beginning of the SSLni signal assertion for the next serial transfer. If the SSLni signal level for the next serial transfer is the same as the SSLni signal level for the current serial transfer, the SPI can execute continuous serial transfers while keeping the SSLni signal assertion status (burst transfer).

- When Between Burst Transfer Frames Delay Select bit (BFDS) of SPI control register (SPCR) is 0.

Figure 43.57 shows an example of an SSLni signal operation for a burst transfer that is implemented using the SPCMD0 and SPCMD1 register settings. This section describes SPI operations (1) to (8) shown in Figure 43.57.

Note: The polarity of the SSLni output signal depends on the SPCR3.SSLnP (n = 0 to 3) bits settings.

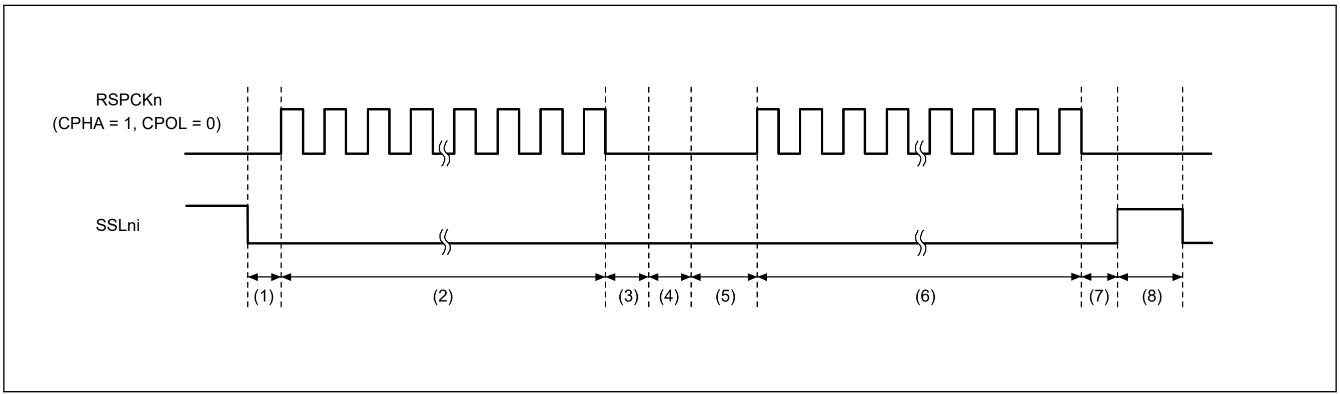


Figure 43.57 Example of burst transfer operation using the SSLKP bit (BFDS = 0, SPFRF = 0)

The SPI operation at times (1) to (8) in the figure is as follows:

1. Based on the SPCMD0 settings, the SPI asserts the SSLni signal and inserts RSPCK delays.
2. The SPI executes serial transfers in accordance with the SPCMD0 settings.
3. The SPI inserts an SSL negation delay.
4. Because the SPCMD0.SSLKP bit is 1, the SPI keeps the SSLni signal value specified in SPCMD0. This period additionally continues for 5 TCLK cycles (at minimum) that is the same as the next-access delay time of SPCMD0. If the shift register is empty after the passage of the minimum period, this period is sustained until the transmit data is stored in the shift register for the next transfer.
5. Based on the SPCMD1 settings, the SPI asserts the SSLni signal and inserts RSPCK delays.
6. The SPI executes serial transfers in accordance with the SPCMD1 settings.
7. Insert SSL negate delay.
8. Because the SPCMD1.SSLKP bit is 0, the SPI negates the SSLni signal. In addition, a next-access delay is inserted in accordance with SPCMD1.

If the SSLni signal output settings in the SPCMDm register where 1 is assigned to the SSLKP bit are different from the SSLni signal output settings in the SPCMDm register to be used in the next transfer, the SPI switches the SSLni signal status to SSLni signal assertion as shown in (5) in Figure 43.57. This corresponds to the command for the next transfer.

Note: If such an SSLni signal switching occurs, the slaves that drive the MISOn signal compete, and collision of signal levels might occur.

The SPI in master mode references the SSLni signal operation within the module when the SSLKP bit is not used. When the SPCMDm.CPHA bit is 0, the SPI can accurately start serial transfers by using the SSLni signal assertion for the next transfer that is detected internally.

- When Between Burst Transfer Frames Delay Select bit (BFDS) of SPI control register (SPCR) is 1.

Figure 43.58 shows an example of SSLni signal operation when burst transfer is achieved by using the settings of SPCMD0 and SPCMD1. The following describes SPI operations of (1) to (6) shown in Figure 43.58. The SSLni output signal polarity depends on the set SPCR3.SSLnP (n = 0 to 3) value.

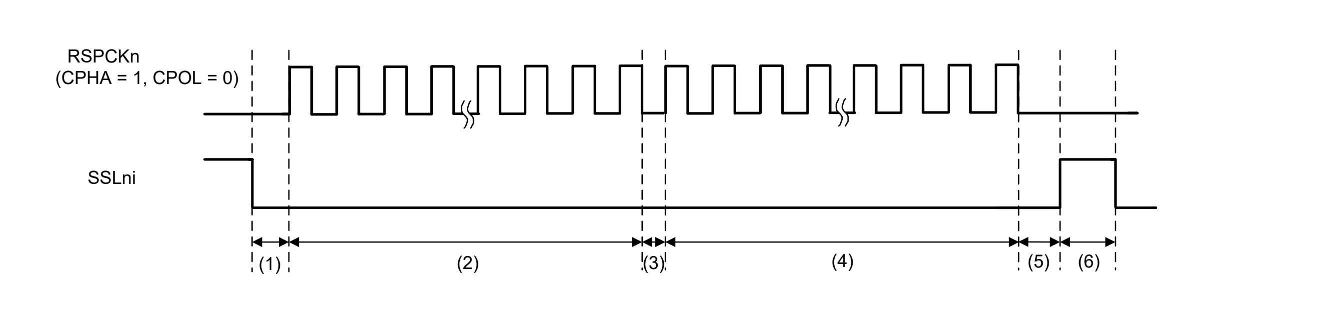


Figure 43.58 Example of Burst Transfer Operation Using SSLKP Bit (BFDS = 1, SPFRF = 0)

1. Assert the SSLni signal and insert an RSPCK delay according to SPCMD0. The RSPCK delay is inserted only the first frame of burst transmission.
2. Perform serial transfer according to SPCMD0. Wait last clock until the next transmit data is stored in the shift register, if the shift register is empty during RSPCK negate period between frames.
3. The value of SSLni signal according to SPCMD0 was hold, because the SPCMD0.SSLKP bit is 1. RSPCK negate period between frames is 0.5RSPCK, if the shift register is not empty.
4. Perform serial transfer according to SPCMD1.
5. Insert SSLni negate delay for the last frame.
6. The SSLni signal is negated because the SSLKP bit in SPCMD1 is 0. Furthermore, the next-access delay is inserted according to SPCMD1.

[In the TI-SSP case]

SPI asserts the SSLni signal for one cycle at the start of serial transfer.

Serial transfer can be executed continuously by asserting the SSLni signal for one cycle at the start of the next serial transfer (burst transfer).

- When the SSLni signal level holding bit (SSLKP) of the SPI command register (SPCMD) is 1 and the burst transfer frame delay selection bit (BFDS) of the SPI control register (SPCR) is 1, SPCMD0 to SPCMD1 are shown in Figure 43.59. The following shows an example of SSLni signal operation and serial data MISOn/MOSIn when burst transfer is realized using the settings. The SSLni output signal polarity depends on the set SPI SSLi signal polarity bit (SPCR3.SSLiP) (i = 0 to 3) value.

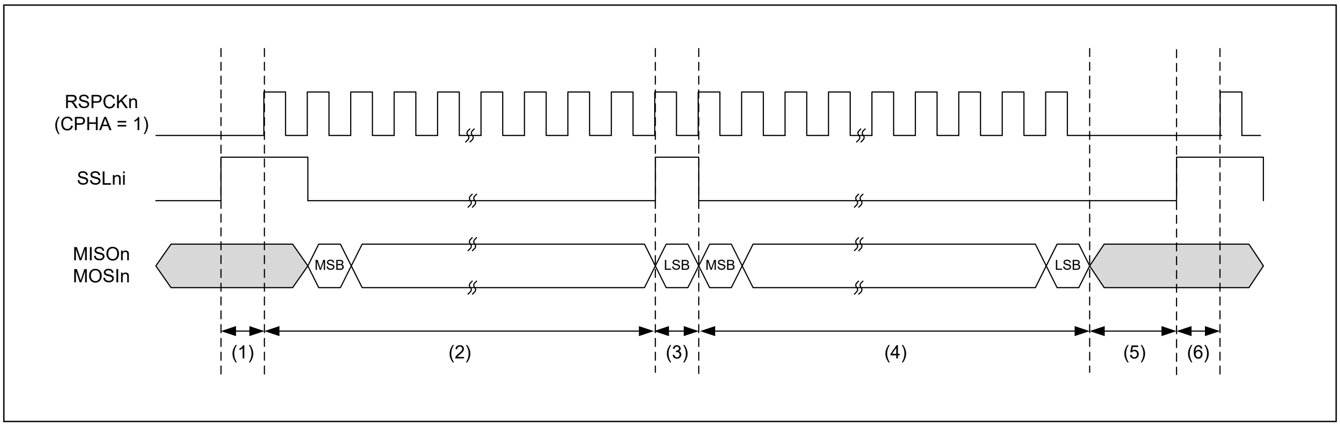


Figure 43.59 Example of Burst Transfer Operation (SPFRF = 1)

1. Assert the SSLni signal and insert an RSPCK delay according to SPCMD0.
   The RSPCK delay is inserted only the first frame of burst transmission.
2. Perform serial transfer according to SPCMD0.
3. Final data transfer and SSLni assertion are performed simultaneously. If the shift register is empty during the RSPCK negation period between frames, wait for the output of the last clock until the transmission data for the next transfer is stored in the shift register.

4. Perform serial transfer according to SPCMD1.
5. Insert OE negate delay for the last frame.
6. Insert the next access delay according to SPCMD1.

If the SSLni signal output setting in SPCMD with the SSLKP bit set to 1 differs from the SSLni signal output setting in SPCMD to be used for the next transfer, the SPI changes the SSLni signal state when the SSLni signal corresponding to the next-transfer command is asserted ((5)). Note that, if an SSLni signal change like this takes place, slaves that drive the MISOn signal may conflict with each other, which may cause collision of signal level.

This section describes burst transfer during receive-only operation.

[In the Motorola-SPI case]

When the SSLKP bit in the SPI command register (SPCMD), which the SPI references in the current serial transfer, is 1, the SPI retains the SSLni signal level during serial transfer until the SSLni signal assertion of the next serial transfer starts. When the SSLni signal level in the next serial transfer is the same as the SSLni signal level in the current serial transfer, the SPI can continuously perform serial transfer while holding the SSLni signal assertion status (burst transfer).

- When Between Burst Transfer Frames Delay Select bit (BFDS) of SPI control register (SPCR) is 0.

Figure 43.60 shows an example of SSLni signal operation when burst transfer is achieved by using the settings of SPCMD0 and SPCMD1. The following describes SPI operations of (1) to (8) shown in Figure 43.60. The SSLni output signal polarity depends on the set SPI SSLni signal polarity bit (SPCR3.SSLiP) (i = 0 to 3) value.

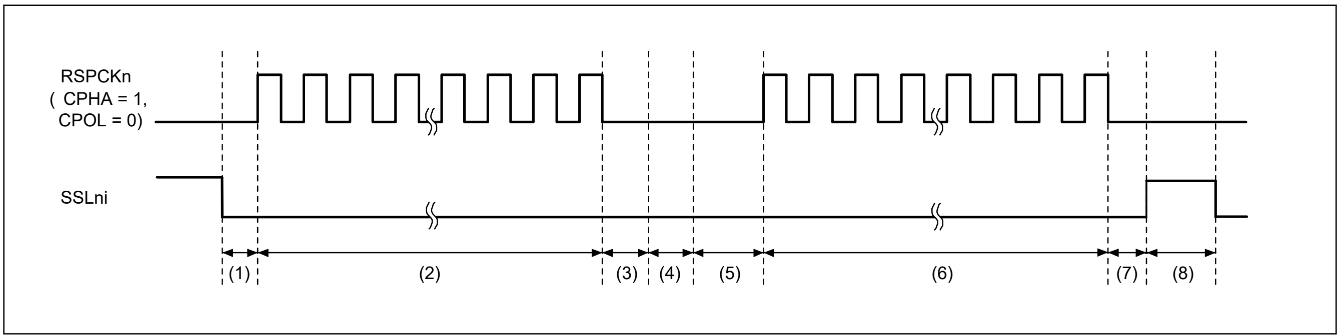


Figure 43.60 Example of Burst Transfer Operation Using SSLKP Bit (BFDS = 0, SPFRF = 0)

1. Assert the SSLni signal and insert an RSPCK delay according to SPCMD0.
2. Perform serial transfer according to SPCMD0.
3. Insert an SSLni negation delay.
4. The SSLni signal value in SPCMD0 is retained because the SSLKP bit in SPCMD0 is 1. This period additionally continues for 5 TCLK cycles (at minimum) that is the same as the next-access delay time of SPCMD0.
5. Assert the SSLni signal and insert an RSPCK delay according to SPCMD1.
6. Perform serial transfer according to SPCMD1.
7. Insert SSLni negate delay.
8. The SSLni signal is negated because the SSLKP bit in SPCMD1 is 0. Furthermore, the next-access delay is inserted according to SPCMD1.

If the SSLni signal output setting and the SSLni signal output setting between SPCMDs used for burst transfer are different, SPI switches the SSLni signal state when the SSLni signal corresponding to the next transfer command is asserted (5). Note that, if an SSLni signal change like this takes place, slaves that drive the MISOn signal may conflict with each other, which may cause collision of signal level.

The SPI in master mode references the SSLni signal operation in the module when SSLKP is not used.

Even when the CPHA bit in SPCMD is 0, the SPI can accurately start serial transfer by using the next transfer SSLni signal assertion detected internally. For this reason, burst transfer in master mode is enabled regardless of the set CPHA bit value. (See section 43.3.11. Initializing the SPI.)

- When Between Burst Transfer Frames Delay Select bit (BFDS) of SPI control register (SPCR) is 1.

Figure 43.61 shows an example of SSLni signal operation when burst transfer is achieved by using the settings of SPCMD0 and SPCMD1. The following describes SPI operations of (1) to (6) shown in Figure 43.61. The SSLni output signal polarity depends on the set SPI SSLi signal polarity bit (SPCR3.SSLiP) (i = 0 to 3) value.

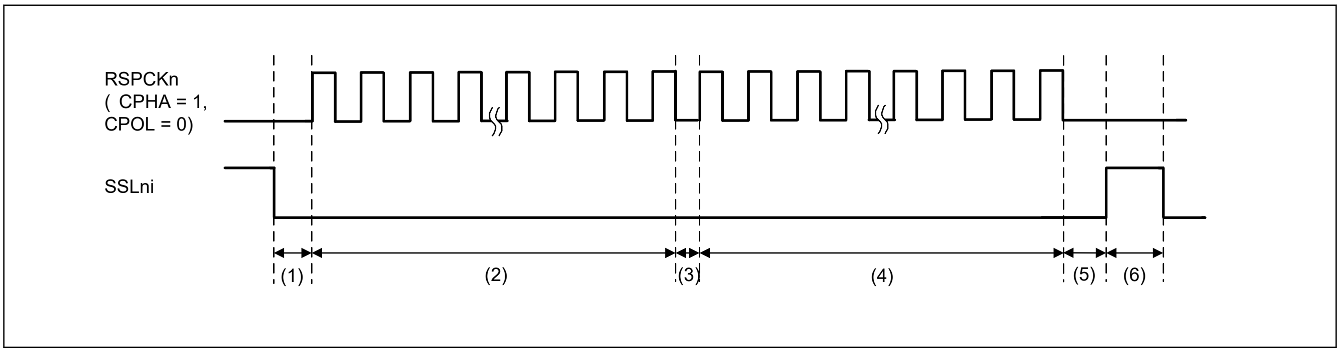


Figure 43.61 Example of Burst Transfer Operation Using SSLKP Bit (BFDS = 1, SPFRF = 0)

1. Assert the SSLni signal and insert an RSPCK delay according to SPCMD0.
   The RSPCK delay is inserted only the first frame of burst transmission.
2. Perform serial transfer according to SPCMD0.
3. Since it is not the last frame, the SSLni signal value at SPCMD0 is retained. RSPCKn negation between frames is 0.5 RSPCKn for the next frame.
4. Perform serial transfer according to SPCMD1.
5. Insert SSLni negate delay for the last frame.
6. The SSLni signal is negated. Furthermore, the next-access delay is inserted according to SPCMD1.

Note: Last frame: Frame set by RMFM[4:0] bits when SPCR2.RMFM[4:0] ≠ 0x00
Or, a frame in which SPCR2.RMEDTG = 1 has been accepted.

[In the TI-SSP case]

SPI asserts the SSLni signal for one cycle at the start of serial transfer.

Serial transfer can be executed continuously by asserting the SSLni signal for one cycle at the start of the next serial transfer (burst transfer).

- When the SSLni signal level holding bit (SSLKP) of the SPI command register (SPCMD) is 1 and the burst transfer frame delay selection bit (BFDS) of the SPI control register (SPCR) is 1, SPCMD0 to SPCMD1 are shown in Figure 43.62. The following shows an example of SSLni signal operation and serial data MISOn / MOSIn when burst transfer is realized using the settings. The SSLni output signal polarity depends on the set SPI SSLi signal polarity bit (SPCR3.SSLiP) (i = 0 to 3) value.

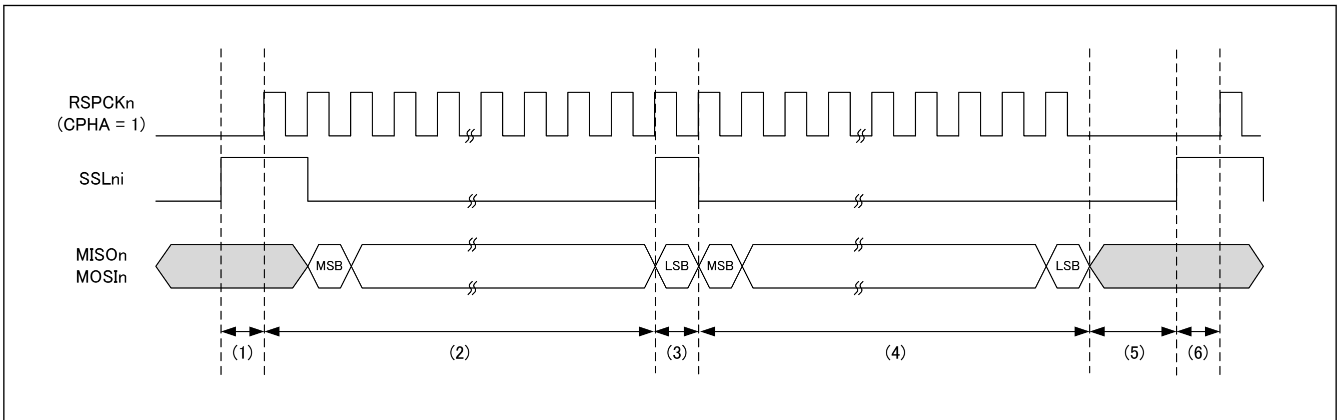


Figure 43.62 Example of Burst Transfer Operation (SPFRF = 1)

1. Assert the SSLni signal and insert an RSPCK delay according to SPCMD0.
   The RSPCK delay is inserted only the first frame of burst transmission.

2. Perform serial transfer according to SPCMD0.
3. Final data transfer and SSLni assertion are performed simultaneously.
4. Perform serial transfer according to SPCMD1.
5. Insert OE negate delay for the last frame.
6. Insert the next access delay according to SPCMD1.

Note: Last frame: Frame set by RMFM[4:0] bits when SPCR2.RMFM[4:0] ≠ 0x00
Or, a frame in which SPCR2.RMEDTG = 1 has been accepted.

If the SSLni signal output setting between the SPCMDs used for burst transfer differs from the SSLni signal output setting, SPI switches the SSLni signal state when the SSLni signal corresponding to the next transfer command is asserted (5). Note that, if an SSLni signal change like this takes place, slaves that drive the MISOn signal may conflict with each other, which may cause collision of signal level.

(5) RSPCK delay (t1)

The RSPCK delay value of the SPI in master mode depends on the SPCMDm.SCKDEN bit setting and the SPDECR.SCKDL[2:0] bits setting. The SPI determines the SPCMDm register to be referenced during a serial transfer by pointer control, and determines an RSPCK delay using the SPCMDm.SCKDEN bit and SPDECR.SCKDL[2:0] bits, as listed in Table 43.10. For a definition of RSPCK delay, see section 43.3.5. Transfer Formats.

RSPCK delay insert to only the first frame of burst transmission, when transmit without “Between Burst Transfer Frames Delay”. (The SPCMD.SSLKP bit is 1 and the SPCR.BFDS bit is 1.)

Table 43.10 Relationship between the SPCMDm.SCKDEN bit, SPDECR.SCKDL[2:0] bits, and RSPCK delay

SPCMDm.SCKDEN bit	SPDECR.SCKDL[2:0] bits	RSPCK delay	
		Motorola-SPI	TI-SSP
0	000b to 111b	1 RSPCK	0 RSPCK
1	000b	1 RSPCK	1 RSPCK
	001b	2 RSPCK	2 RSPCK
	010b	3 RSPCK	3 RSPCK
	011b	4 RSPCK	4 RSPCK
	100b	5 RSPCK	5 RSPCK
	101b	6 RSPCK	6 RSPCK
	110b	7 RSPCK	7 RSPCK
	111b	8 RSPCK	8 RSPCK

(6) SSL negation delay (t2)

The SSL negation delay value of the SPI in master mode depends on the SPCMDm.SLNDEN bit setting and the SPDECR.SLNDL[2:0] bits setting. The SPI determines the SPCMDm register to be referenced by pointer control during a serial transfer, and determines an SSL negation delay using the SPCMDm.SLNDEN bit and SPDECR.SLNDL[2:0] bits, as listed in Table 43.11. For a definition of SSL negation delay, see section 43.3.5. Transfer Formats.

An SSL negation delay is inserted to only the last frame of the burst transmission, that is, transmit without between burst transfer frames delay. (SPCMD.SSLKP bit is 1 and SPCR.BFDS bit is 1).

Table 43.11 Relationship between the SPCMDm.SLNDEN bit, SPDECR.SLNDL[2:0] bits, and SSL negation delay (1 of 2)

SPCMDm.SLNDEN bit	SPDECR.SLNDL[2:0] bits	SSL negation delay
0	000b to 111b	1 RSPCK

Table 43.11 Relationship between the SPCMDm.SLNDEN bit, SPDECR.SLNDL[2:0] bits, and SSL negation delay (2 of 2)

SPCMDm.SLNDEN bit	SPDECR.SLNDL[2:0] bits	SSL negation delay
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

(7) Next-access delay (t3)

The next-access delay value of the SPI in master mode depends on the SPCMDm.SPNDEN bit setting and the SPDECR.SPNDL[2:0] bits setting. The SPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and then determines a next-access delay during serial transfer using the SPCMDm.SPNDEN bit and SPDECR.SPNDL[2:0] bits, as listed in Table 43.12. For a definition of next-access delay, see section 43.3.5. Transfer Formats.

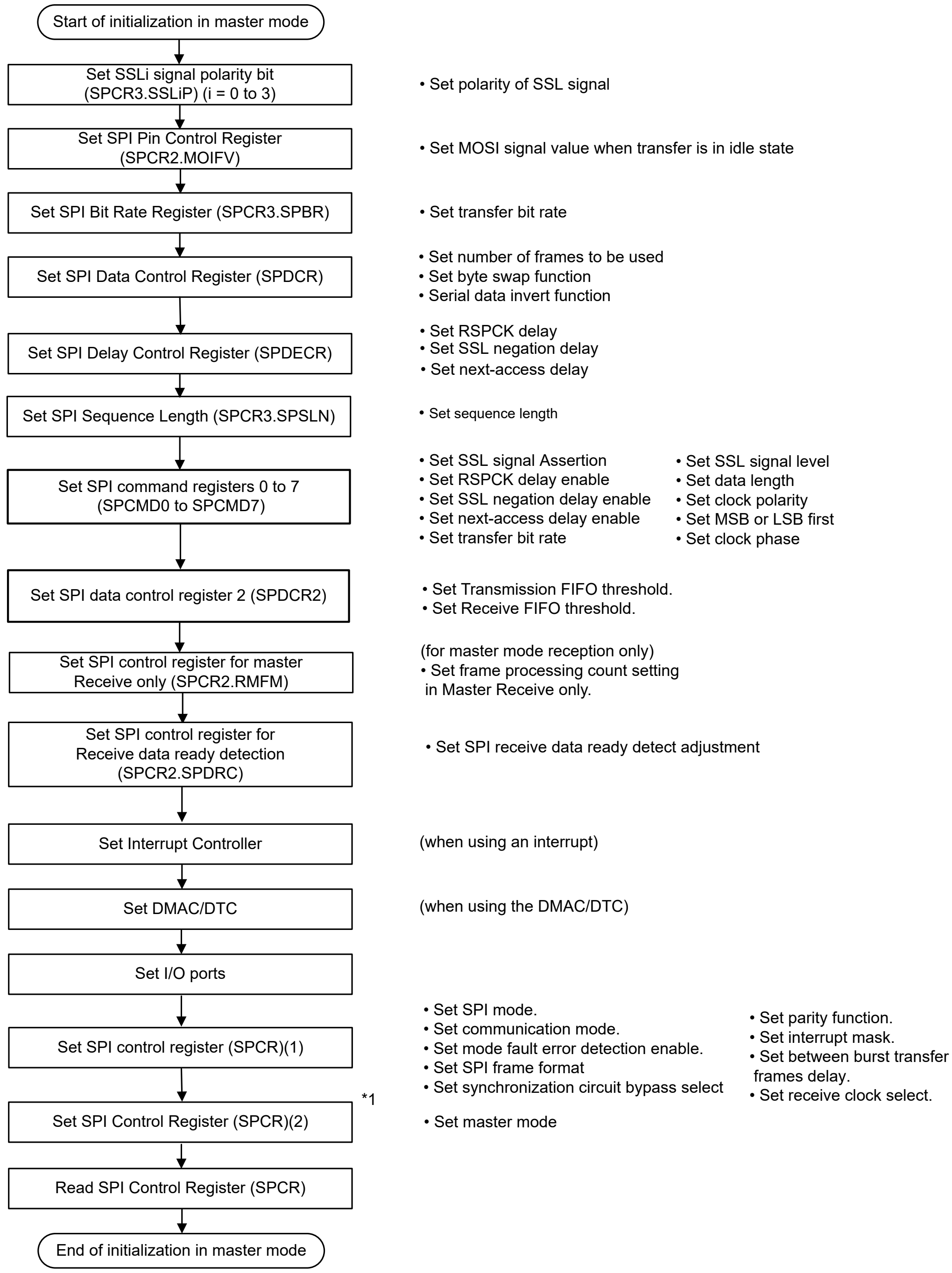
A next-Access delay is inserted to only the last frame of the burst transmission, that is, transmit without between burst transfer frames delay. (SPCMD.SSLKP bit is 1 and SPCR.BFDS bit is 1).

Table 43.12 Relationship between the SPCMDm.SPNDEN bit, SPDECR.SPNDL[2:0] bits, and next-access delay

SPCMDm.SPNDEN bit	SPDECR.SPNDL[2:0] bits	Next-access delay
0	000b to 111b	1 RSPCK + 5 TCLK
1	000b	1 RSPCK + 5 TCLK
	001b	2 RSPCK + 5 TCLK
	010b	3 RSPCK + 5 TCLK
	011b	4 RSPCK + 5 TCLK
	100b	5 RSPCK + 5 TCLK
	101b	6 RSPCK + 5 TCLK
	110b	7 RSPCK + 5 TCLK
	111b	8 RSPCK + 5 TCLK

(8) Initialization flow

Figure 43.63 shows an example of SPI initialization flow when the SPI is in master mode. For information on how to set up the Interrupt Controller Unit (ICU), DMAC or DTC, and I/O ports, see the descriptions given in the individual blocks.



Note 1. Regarding the setting of the SPI control register (SPCR), leave an interval of 1TCLK between setting (1) (initial setting other than MSTR bit) and setting (2) (MSTR setting).

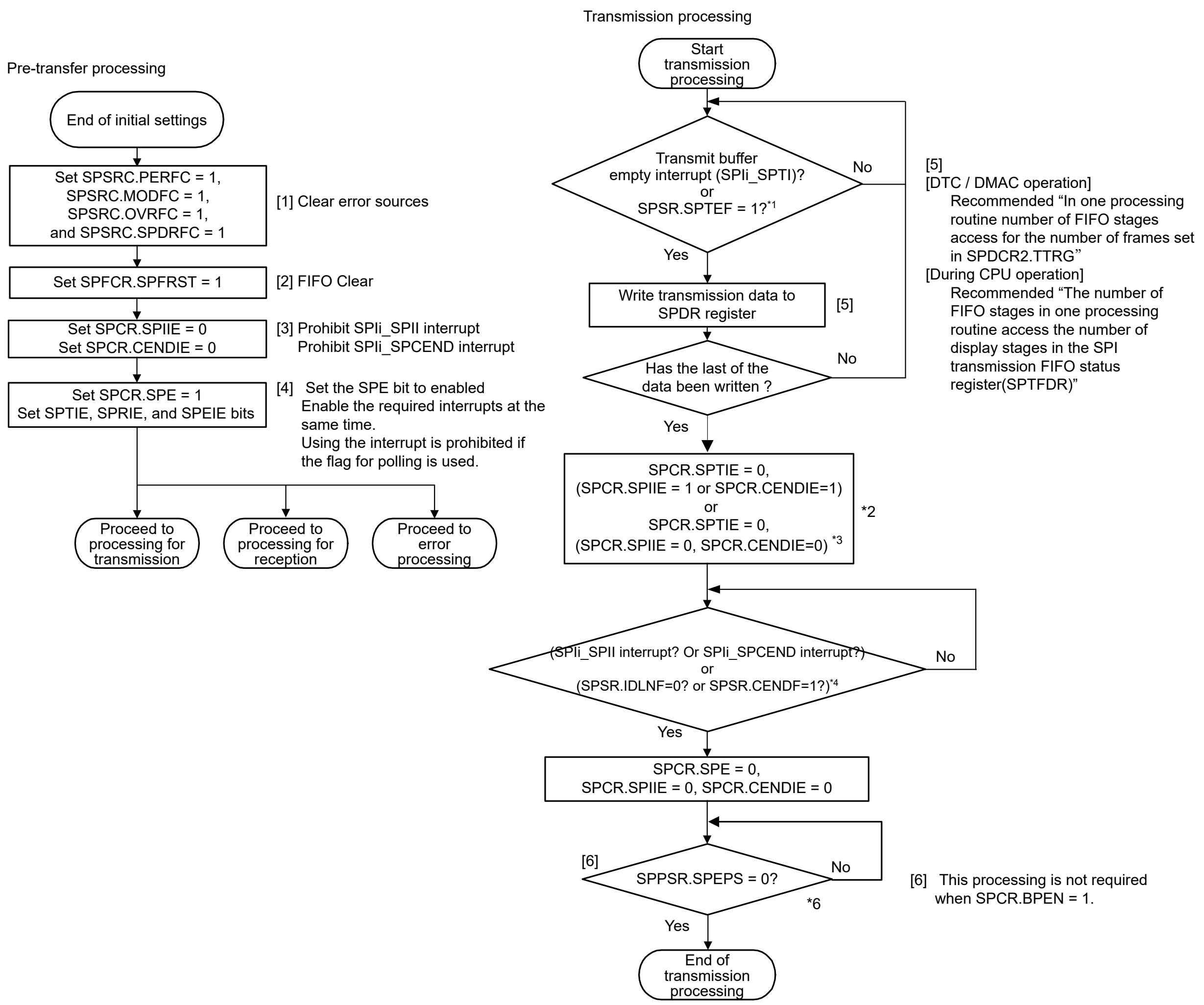
Figure 43.63 Example of initialization flow in master mode for SPI operation

(9) Software processing flow

Figure 43.64 to Figure 43.67 show examples of the software processing flow.

Transmit processing flow

When transmitting data, with the SPIi_SPII or SPIi_SPCEND interrupt enabled, the CPU is notified of the completion of data transmission after the last data write for transmission.

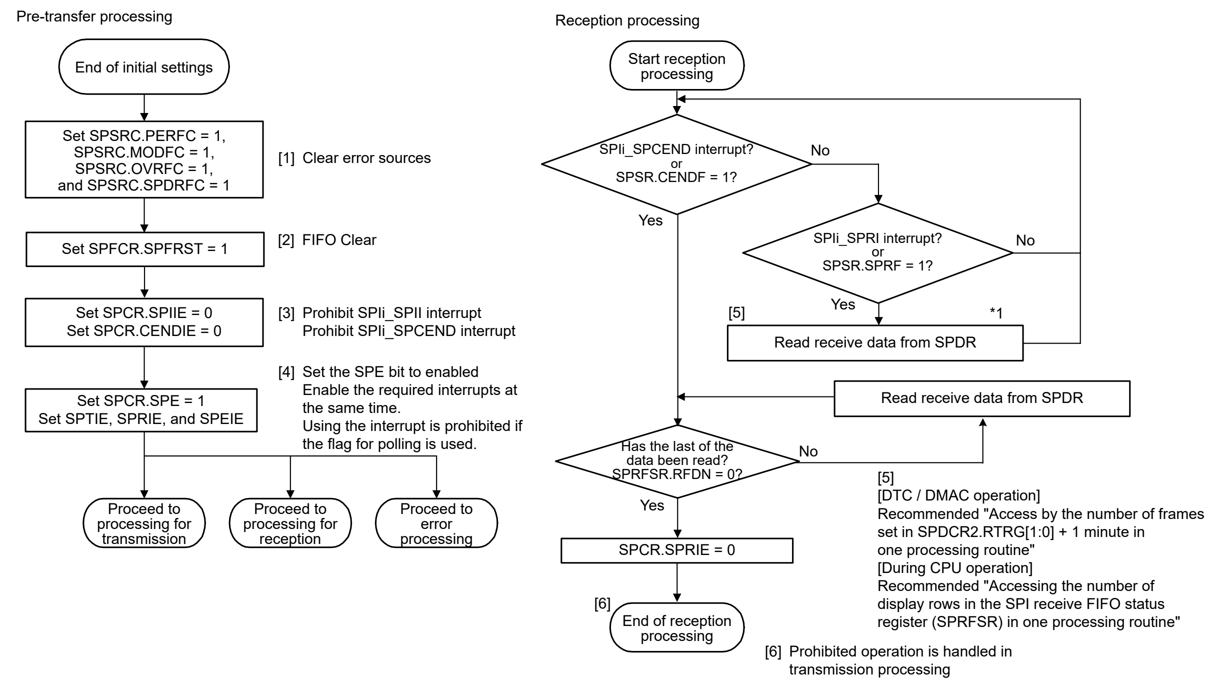


Note 1. Proceed to process of writing data for transmission after reading SPSR.SPTEF flag is 1, if the user uses the flag for polling.
Note 2. The user can know the communication end by SPIi_SPII interrupt (IDLNF flag) or SPIi_SPCEND interrupt (CENDF flag). Use either SPIi_SPII interrupt (IDLNF flag) or SPIi_SPCEND interrupt (CENDF flag).
Note 3. Set SPIi_SPII interrupt prohibited (SPCR.SPIIE = 0) and SPIi_SPCEND interrupt prohibited (SPCR.CENDIE = 0), if the user uses the flag for polling.
Note 4. Wait more than 1 PCLK, after writing data for transmission to SPDR, and before starting polling of SPSR.IDLNF or SPSR.CENDF, if the user uses the flag for polling.
Note 5. During CPU operation, Write 1 to SPI transmit buffer empty flag clear bit (SPSRC.SPTEFC) after one processing routine.
Note 6. When setting to 0 in slave mode and SPCR.BPEN = 0, check that SPPSR.SPEPS has become 0. (Not required when BPEN = 1.)

Figure 43.64 Transmission flow in master mode

Receive processing flow

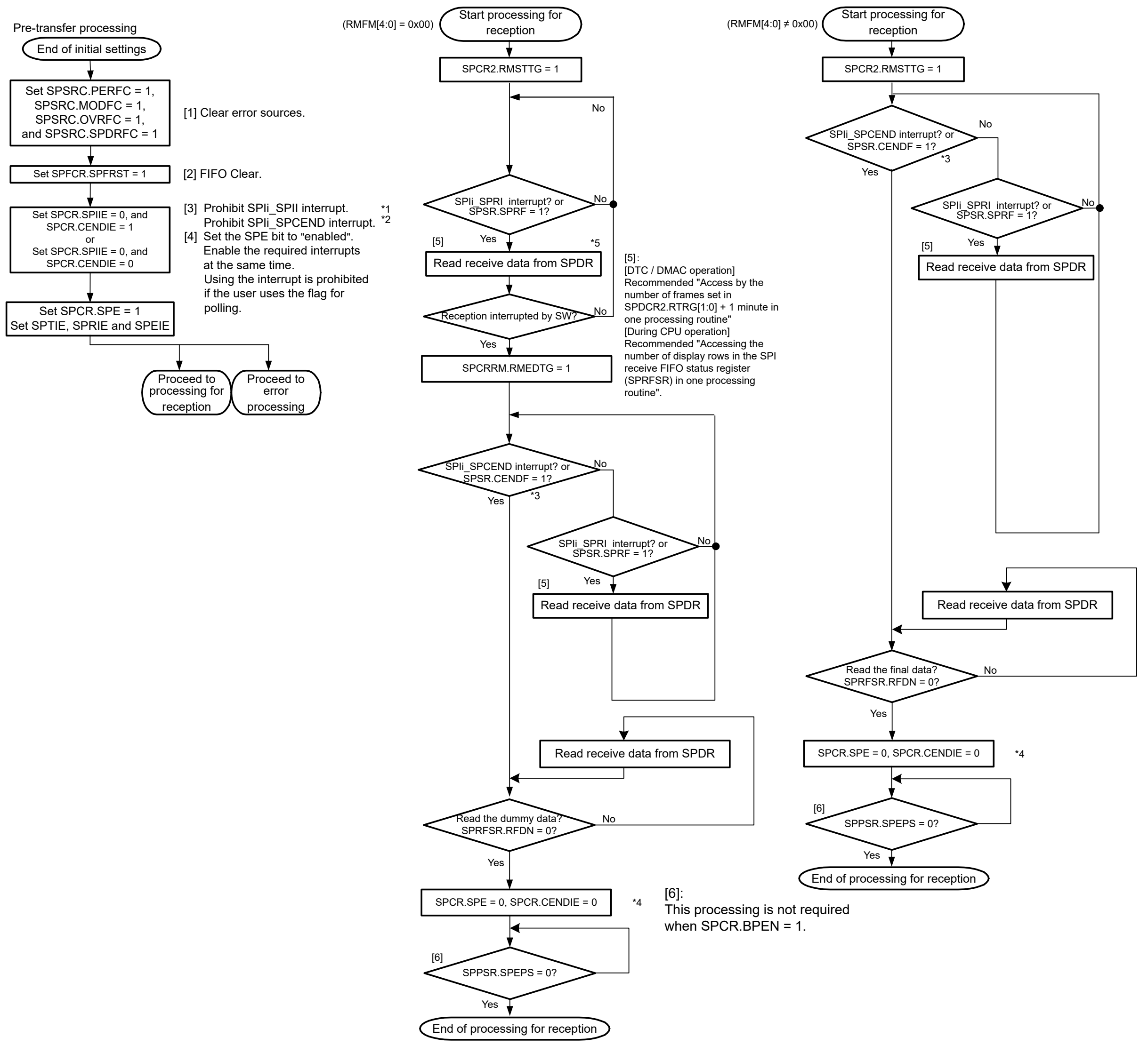
The SPI has receive only operation in slave mode.



Note 1. During CPU operation, write 1 to the SPSRC.SPRFC bit (SPI receive buffer full flag clear bit) after one processing routine.

Figure 43.65 Reception flow in master mode

Receive only processing flow in master mode



Note 1. The end of communication can be determined by the SPIi_SPCEND interrupt (CENDF flag). Use the SPIi_SPCEND interrupt (CENDF flag).
Note 2. Disable SPIi_SPCEND interrupt (SPCR.CENDIE = 0) during flag polling.
Note 3. A communication completion interrupt at reception only operation (Master) occurs at a predetermined timing under any of the following conditions.
- Frame number processing set by RMFM completed (RMFM[4:0] ≠ 0x00)
- RMEDTG = 1, when processing of the number of frames set by RMFM is not completed (including RMFM[4:0] = 0x00)

Note 4. Internal trigger is also cleared.
Note 5. During CPU operation, write 1 to the SPI receive buffer full flag clear bit (SPSRC.SPRFC) after one processing routine.

Figure 43.66 Software Processing Flowchart in Master Mode (Reception-only)

Error processing flow

The SPI detects the following errors:

- Mode fault error
- Underrun error
- Overrun error
- Parity error

When a mode fault error is generated, the SPCR.SPE bit is automatically cleared, stopping operations for transmission and reception. For errors from other sources, the SPCR.SPE bit is not cleared and operations for transmission and reception continue. Therefore, Renesas recommends clearing the SPCR.SPE bit to stop operations for errors other than mode fault errors. Not doing so leads to updating of the SPDCR2.SPECM[2:0] bits.

When an error is detected using an interrupt, clear the ICU.IELSRn.IR flag in the error processing routine. If this is not done, the ICU.IELSRn.IR flag might continue to indicate the SPIi_SPTI or SPIi_SPRI interrupt request. If the SPIi_SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the SPI.

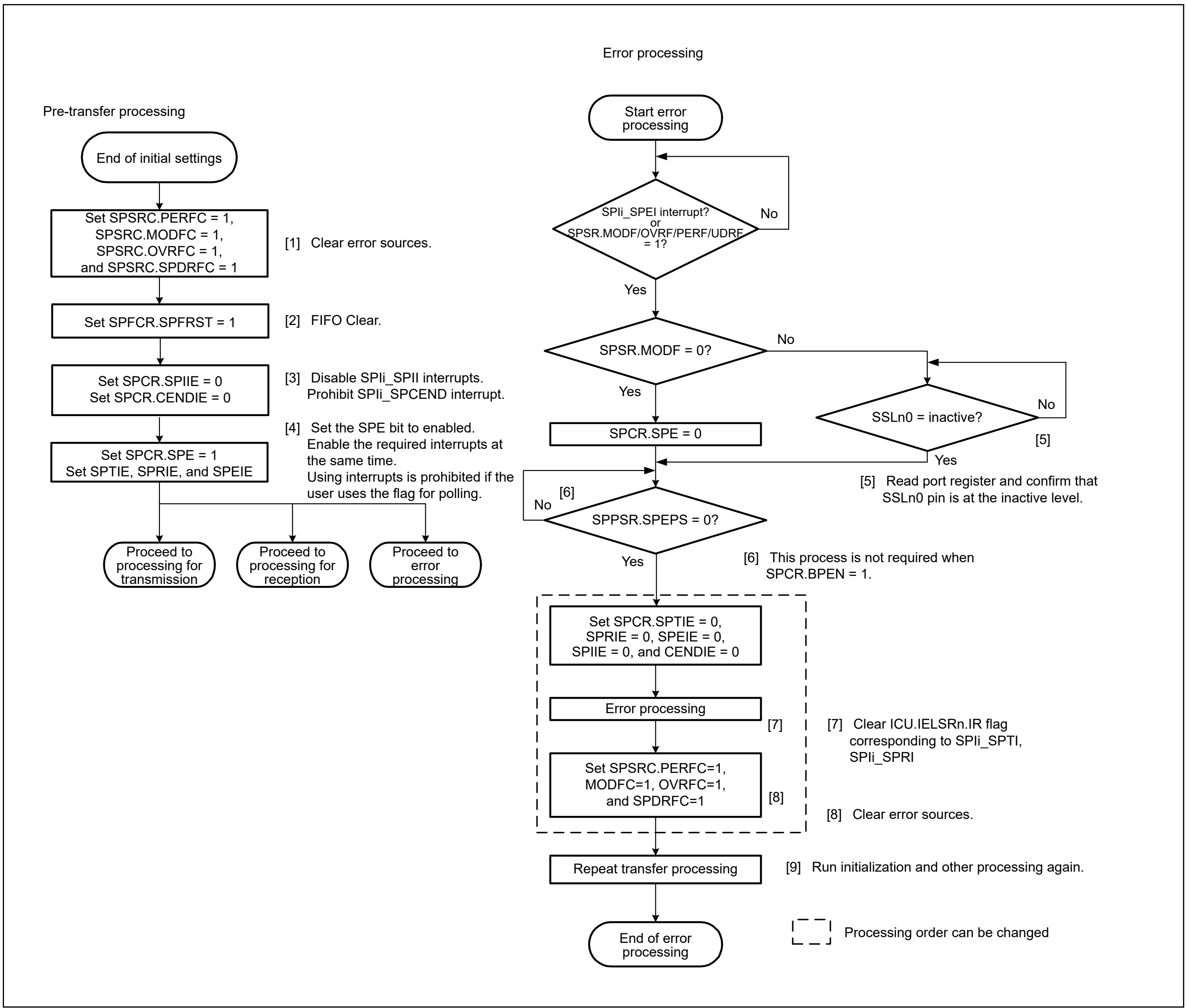


Figure 43.67 Error processing flow in master mode

43.3.12.2 Slave Mode Operation

(1) Starting a serial transfer

When the SPCMD0.CPHA bit is 0, if the SPI detects an SSLn0 input signal assertion, it must drive valid data to the MISOn output signal. For this reason, when the CPHA bit is 0, the assertion of the SSLn0 input signal triggers the start of a serial transfer.

When the CPHA bit is 1, if the SPI detects the first RSPCKn edge in an SSLn0 signal asserted condition, it must drive valid data to the MISOn output signal. For this reason, when the CPHA bit is 1, the first RSPCKn edge in an SSLn0 signal asserted condition triggers the start of a serial transfer.

Regardless of the CPHA bit setting, the SPI drives the MISOn output signal on SSLn0 signal assertion. The data that is output by the SPI is either valid or invalid, depending on the CPHA bit setting.

For details on the SPI transfer format, see section 43.3.5. Transfer Formats. The polarity of the SSLn0 input signal depends on the SPCR3.SSL0P setting.

(2) Terminating a serial transfer

Regardless of the SPCMD0.CPHA bit setting, the SPI terminates the serial transfer after detecting an RSPCKn edge corresponding to the final sampling timing. When the number of data stored in the receive FIFO is less than the number of FIFO stages, on termination of serial transfer, the SPI copies received data from the shift register to the receive buffer of the SPDR register. On termination of a serial transfer, the SPI changes the status of the shift register to empty, regardless of the receive buffer state. A mode fault error occurs if the SPI detects an SSLn0 input signal negation from the beginning of serial transfer to the end of serial transfer (see section 43.3.10. Error Detection).

The final sampling timing changes depending on the bit length of transfer data. In slave mode, the SPI data length is determined by the SPCMD0.SPB[4:0] bits setting. The polarity of the SSLn0 input signal is determined by the SPCR3.SSL0P bit setting. For details on the SPI transfer format, see section 43.3.5. Transfer Formats.

(3) Notes on single-slave operations

[In the Motorola-SPI case]

If the SPCMD0.CPHA bit is 0, the SPI starts serial transfers when it detects the assertion edge for an SSLn0 input signal. In the configuration shown in Figure 43.8, if the SPI is used in single-slave mode, the SSLn0 signal is fixed at an active state. Therefore, when the CPHA bit is set to 0, the SPI cannot correctly start a serial transfer. For the SPI to correctly execute transmit and receive operations in slave mode when the SSLn0 input signal is fixed at an active state, the CPHA bit must be set to 1. Do not fix the SSLn0 input signal if there is a requirement for setting the CPHA bit to 0.

[In the TI-SSP case]

When SPI is used as a single slave in the configuration shown in the Figure 43.8, the SSLn0 input signal is always fixed to the inactive state, so the SPI cannot start the serial transfer.

When using a single slave, use the configuration shown in the example in Figure 43.7.

(4) Burst transfer

[In the Motorola-SPI case]

If the SPCMD0.CPHA bit is 1, continuous serial transfer (burst transfer) can be executed while retaining the assertion state for the SSLn0 input signal. When the CPHA bit is 1, the serial transfer period is the period from the first RSPCKn edge to the sampling timing for the reception of the final bit in an SSLn0 signal active state. Even when the SSLn0 input signal remains at the active level, the SPI can accommodate burst transfers, because it can detect the start of an access.

When the CPHA bit is 0, the second and subsequent serial transfers during burst transfer cannot be executed correctly.

[In the TI-SSP case]

In serial transfer, data transfer starts after the SSLn0 input signal is asserted for RSPCK 1 cycle. Since frame transfer starts from the SSLn0 input signal, SSLn0 must be asserted between frames.

(5) Initialization flow

Figure 43.68 shows an example of initialization flow for SPI operation when the SPI is in slave mode. For a description of how to set up the ICU, DMAC or DTC, and I/O ports, see the descriptions given in the individual blocks.

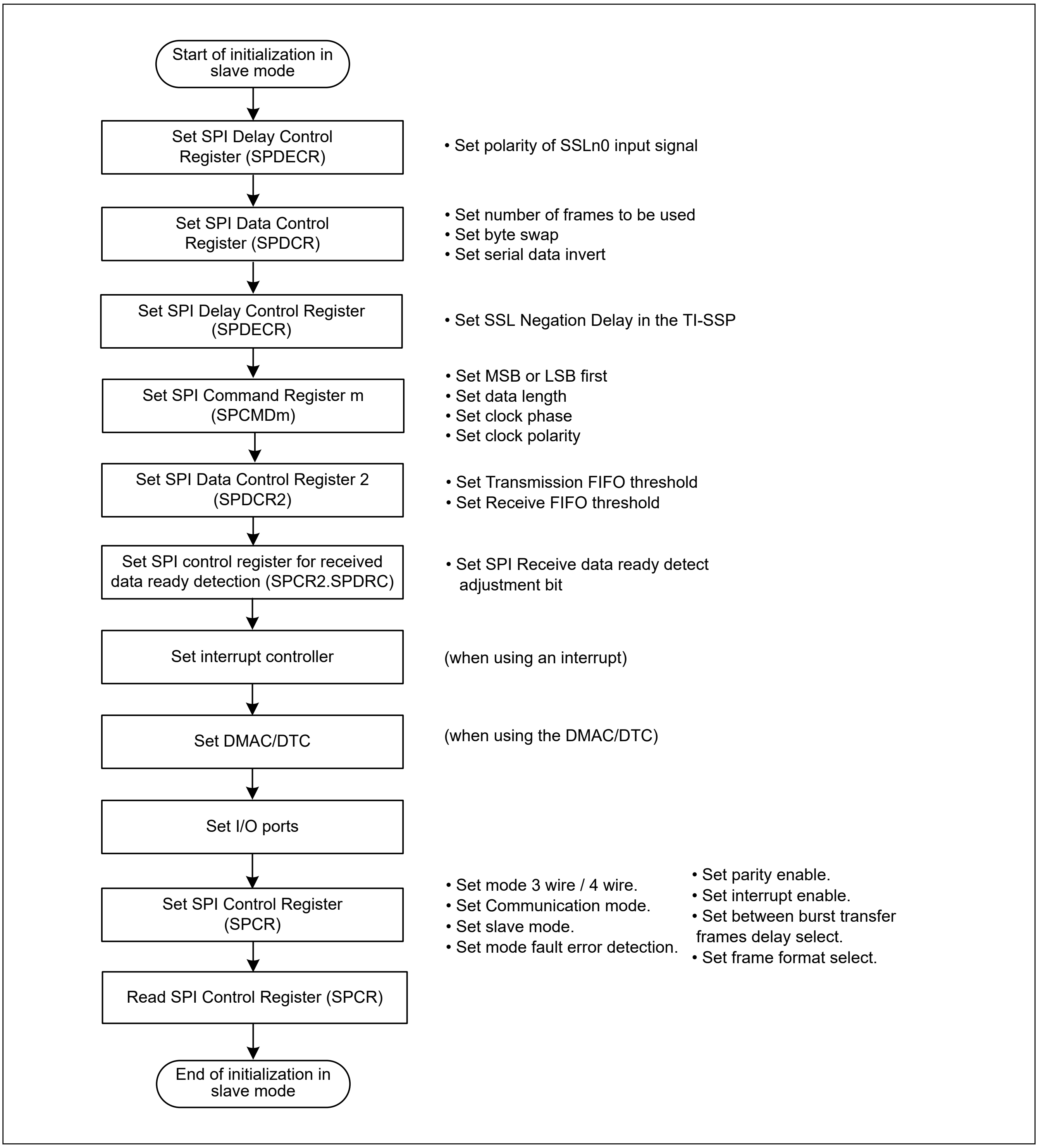
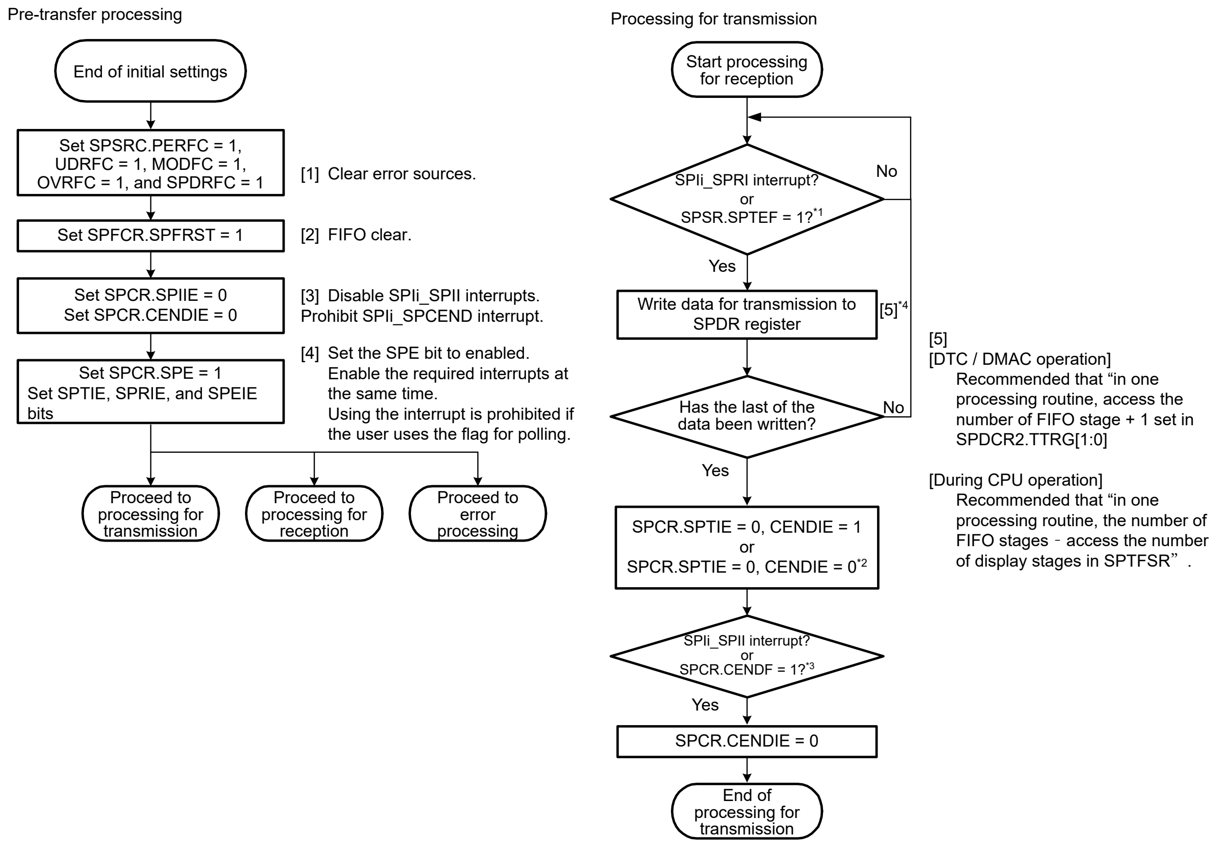


Figure 43.68 Example initialization flow in slave mode for SPI operation

(6) Software processing flow

Figure 43.69 to Figure 43.72 show examples of the flow of software processing.

Transmit processing flow



Note 1. Before writing data for transmission to SPDR register, check the transmit buffer empty interrupt by reading SPSR.SPTEF flag, if the flag for polling is used.

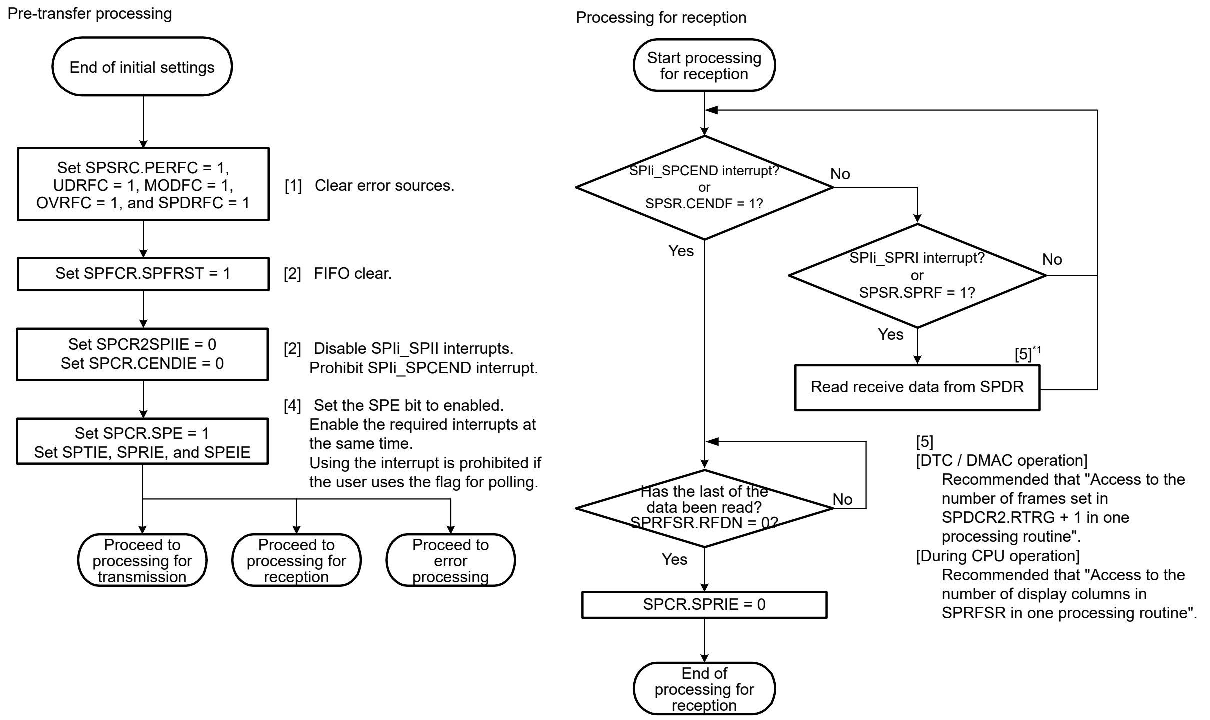
Note 2. Set SPIi_SPCEND interrupt prohibited (SPCR.CENDIE = 0), if the user uses the flag for polling.

Note 3. Wait more than 1 PCLK, after writing data for transmission to SPDR, and before starting polling of SPSR.IDLNF or SPSR.CENDF, if the user uses the flag for polling.

Note 4. During CPU operation, write 1 to SPI transmit buffer empty flag clear bit (SPSRC.SPTEFC) after one processing routine.

Figure 43.69 Transmission flow in slave mode

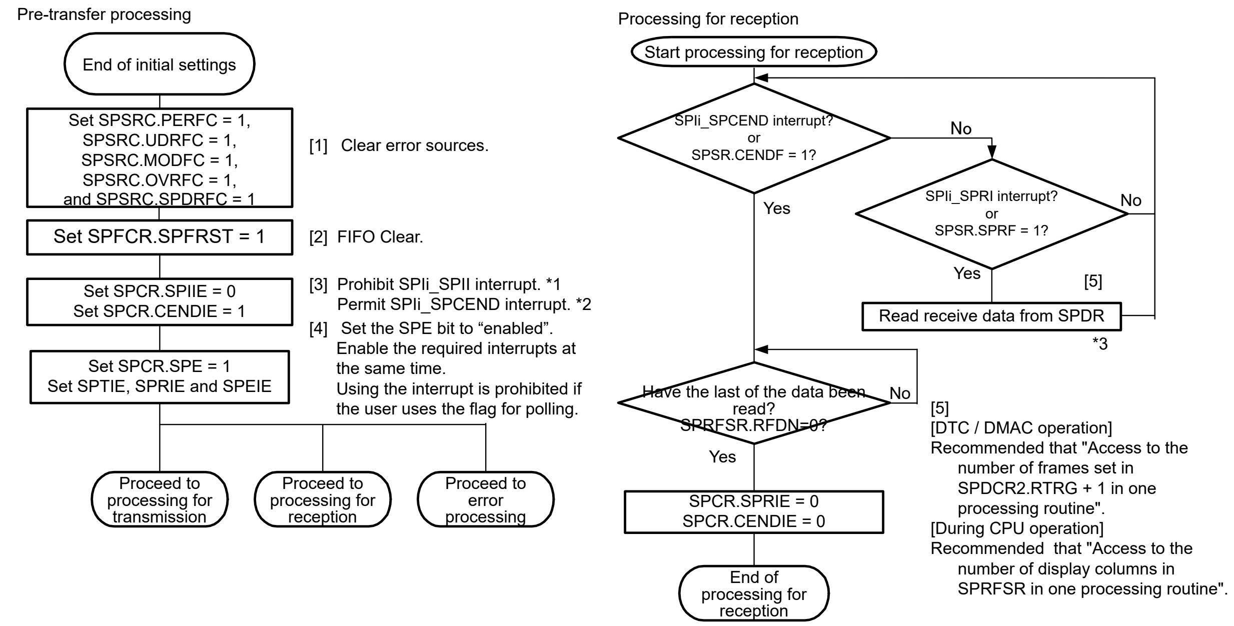
Receive processing flow



Note 1. During CPU operation, write 1 to the SPI receive buffer full flag clear bit (SPSRC.SPTEFC) after one processing routine.

Figure 43.70 Reception flow in slave mode

Master Reception-only processing flow



Note 1. The end of communication can be determined by the SPIi_SPCEND interrupt (CENDF flag). Use the SPIi_SPCEND interrupt (CENDF flag).
Note 2. Disable the SPIi_SPCEND interrupt (SPCR.CENDIE = 0) during flag polling.
Note 3. During CPU operation, write 1 to the SPI receive buffer full flag clear bit (SPSRC.SPTEFC) after one processing routine.

Figure 43.71 Software Processing Flowchart in Master Mode (Reception-only)

Error processing flow

In slave mode operation, even when a mode fault error is generated, the SPSR.MODF flag can be cleared regardless of the state of the SSLn0 pin.

When an error is detected by using an interrupt, clear the ICU.IELSRn.IR flag in the error processing routine. If this is not done, the ICU.IELSRn.IR flag might continue to indicate the SPIi_SPTI or SPIi_SPRI interrupt request. If the SPIi_SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the SPI.

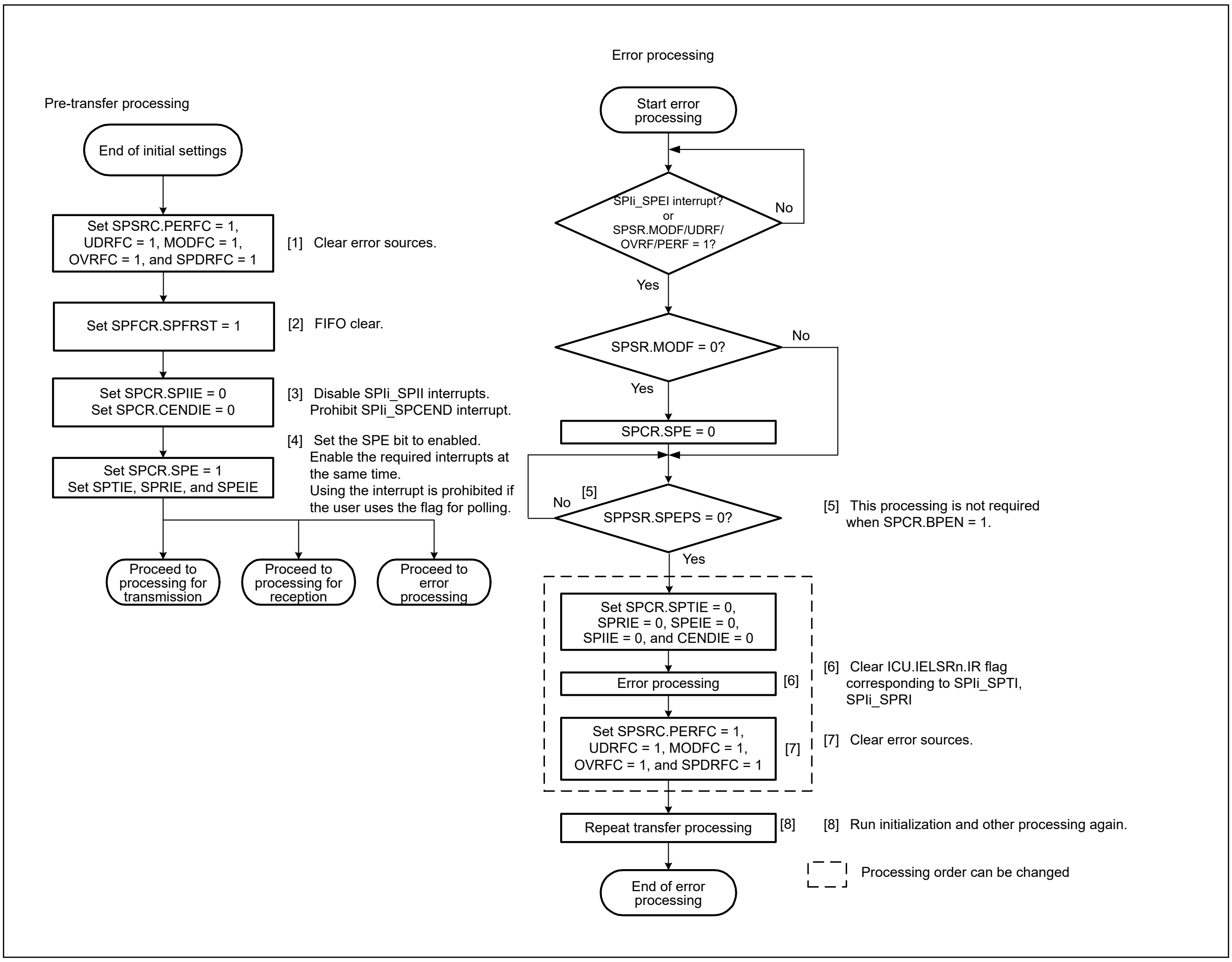


Figure 43.72 Error processing flow for slave mode

43.3.13 Clock Synchronous Operation

Setting the SPCR.SPMS bit to 1 selects clock synchronous operation of the SPI. In clock synchronous operation, the SSLni pin is not used, and the RSPCKn, MOSIn, and MISOn pins handle communications. All SSLni pins are available as I/O port pins.

Although clock synchronous operation does not require the use of the SSLni pin, operation of the module is the same as in SPI operation. In both master mode and slave mode operations, communications can be performed with the same flow as in SPI operation. However, mode fault errors are not detected, because the SSLni pin is not used.

Additionally, do not perform operation if clock synchronous operation is enabled when the SPCMDm.CPHA bit is set to 0 in slave mode (SPCR.MSTR = 0).

43.3.13.1 Master Mode Operation

(1) Starting serial transfer

When data is written to the SPI data register (SPDR) while the next transfer data is not set in the transmit FIFO, the SPI updates the transmit buffer (SPTXn, n = 0 to 3) data in SPDR. While the shift register is empty, the SPI copies transmit buffer data to the shift register to start serial transfer. After the SPI copies transmit data to the shift register, it changes the shift register status to full. Upon completion of serial transfer, the SPI changes the shift register status to empty. The shift register status cannot be monitored.

For details about the SPI transfer format, see section 43.3.5. Transfer Formats. In clock synchronous operation, however, the SSLn0 output signal is not used for communication.

(2) Terminating serial transfer

The SPI terminates the serial transfer after transmitting an RSPCKn edge corresponding to the sampling timing. If the number of data stored in the receive FIFO < the number of FIFO stages, on termination of serial transfer, the SPI copies data from the shift register to the receive buffer of the SPI Data Register (SPDR).

The final sampling timing varies depending on the bit length of transfer data. In master mode, the SPI data length depends on the SPCMDm.SPB[4:0] bits setting. Transfer in clock synchronous operation is conducted without the SSLn0 output signal. For details on the SPI transfer format, see section 43.3.5. Transfer Formats.

(3) Sequence control

The transfer format used in master mode is determined by the SPCR3, SPCMDm, and SPDECR registers. Although the SSLni signals are not output in clock synchronous operation, these settings are valid.

The SPCR3.SPSLN[2:0] bits determine the sequence configuration for serial transfers that are executed by the SPI in master mode. The following parameters are specified in the SPCMDm register:

- SSLni output signal value
- MSB or LSB first
- Data length
- Some of the bit rate settings
- RSPCKn polarity and phase
- Whether SPDECR.SCKDL is to be referenced
- Whether SPDECR.SLNDL is to be referenced
- Whether SPDECR.SPNDL is to be referenced

SPCR3.SPBR holds some of the bit rate settings such as SPDECR.SCKDL, an SPI clock delay value, SPDECR.SLNDL, an SSL negation delay, and SPDECR.SPNDL, a next-access delay value.

Based on the sequence length that is assigned to SPCR3, the SPI makes up a sequence comprised of a part or all of SPCMDm register. The SPI contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPDCR2.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the SPI function is enabled, the SPI loads the pointer to the commands in SPCMD0 register, and incorporates the SPCMD0 register setting into the transfer format at the beginning of serial transfer. The SPI increments the pointer each time the next-access delay period for a data transfer ends. On completion of the serial transfer that corresponds to the final command comprising the sequence, the SPI sets the pointer to the SPCMD0 register, and in this manner the sequence is executed repeatedly.

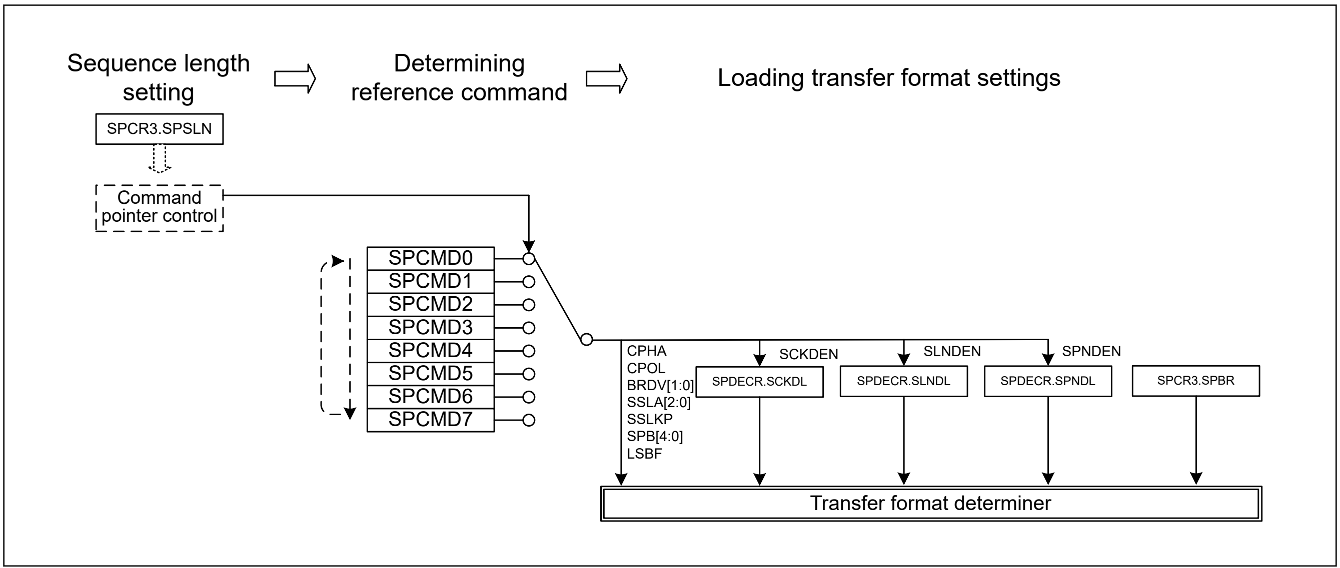


Figure 43.73 Procedure for determining the form of serial transmission in master mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

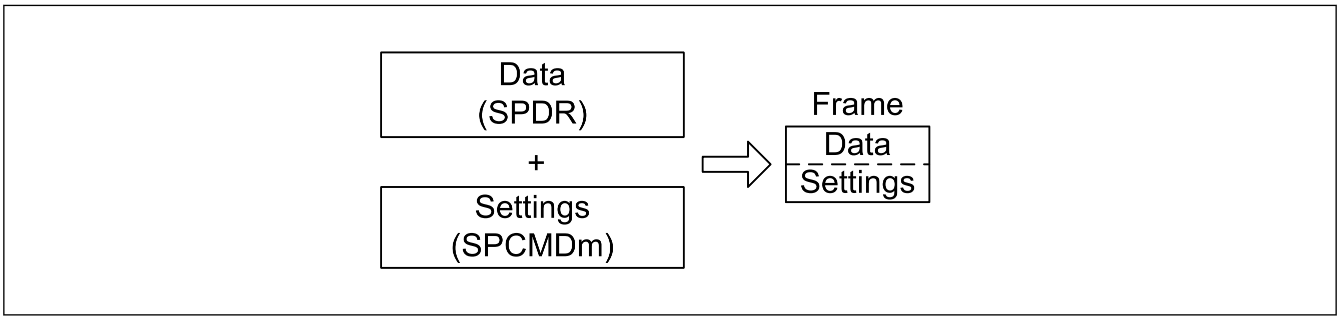


Figure 43.74 Conceptual diagram of frames

Figure 43.75 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings.

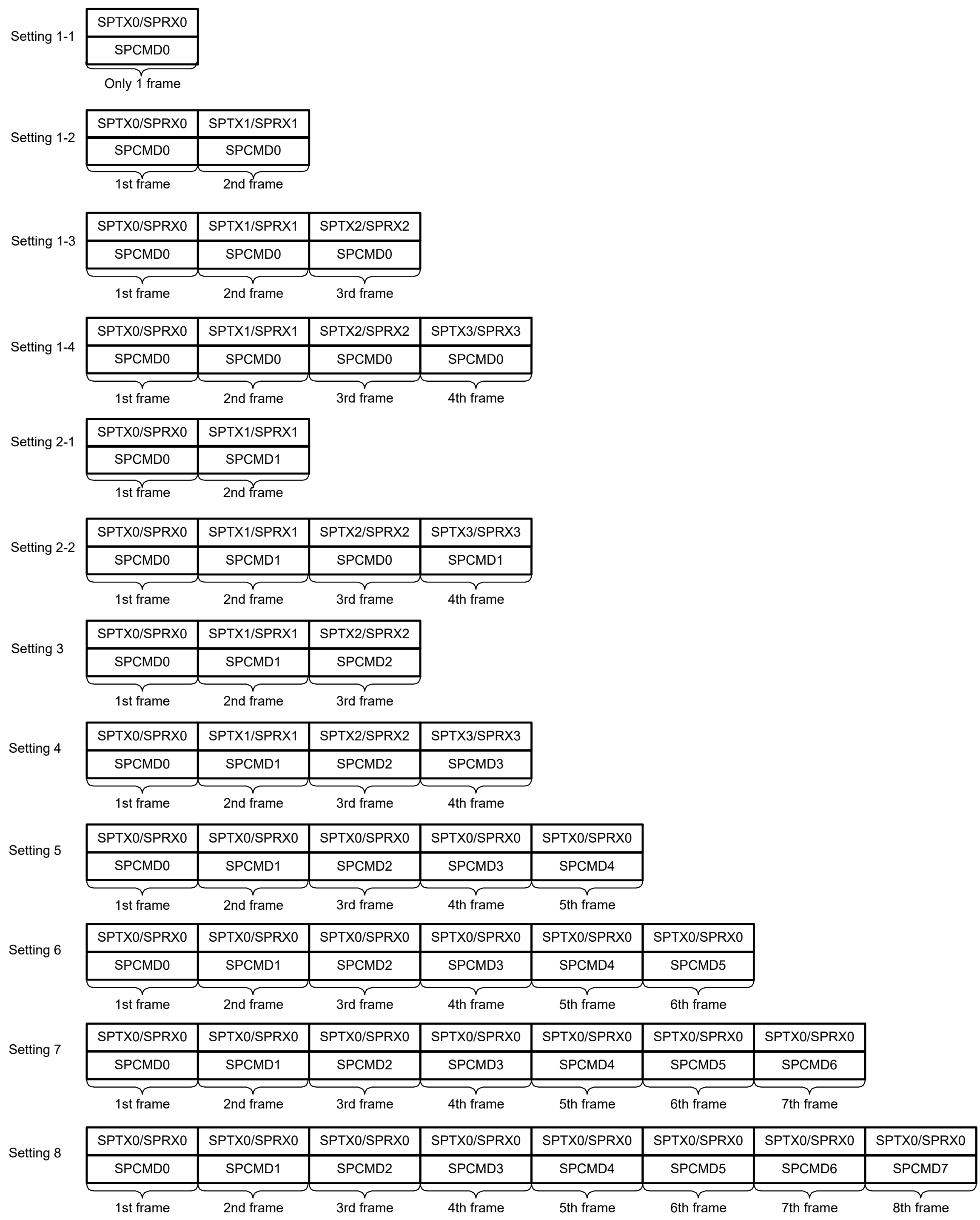


Figure 43.75 Correspondence between SPI Command Register and transmit and receive buffers in sequence operations

(4) Initialization flow

Figure 43.76 shows an example of initialization flow for clock synchronous operation when the SPI is used in master mode. For information on how to set up the ICU, DMAC or DTC, and I/O ports, see the descriptions given in the individual blocks.

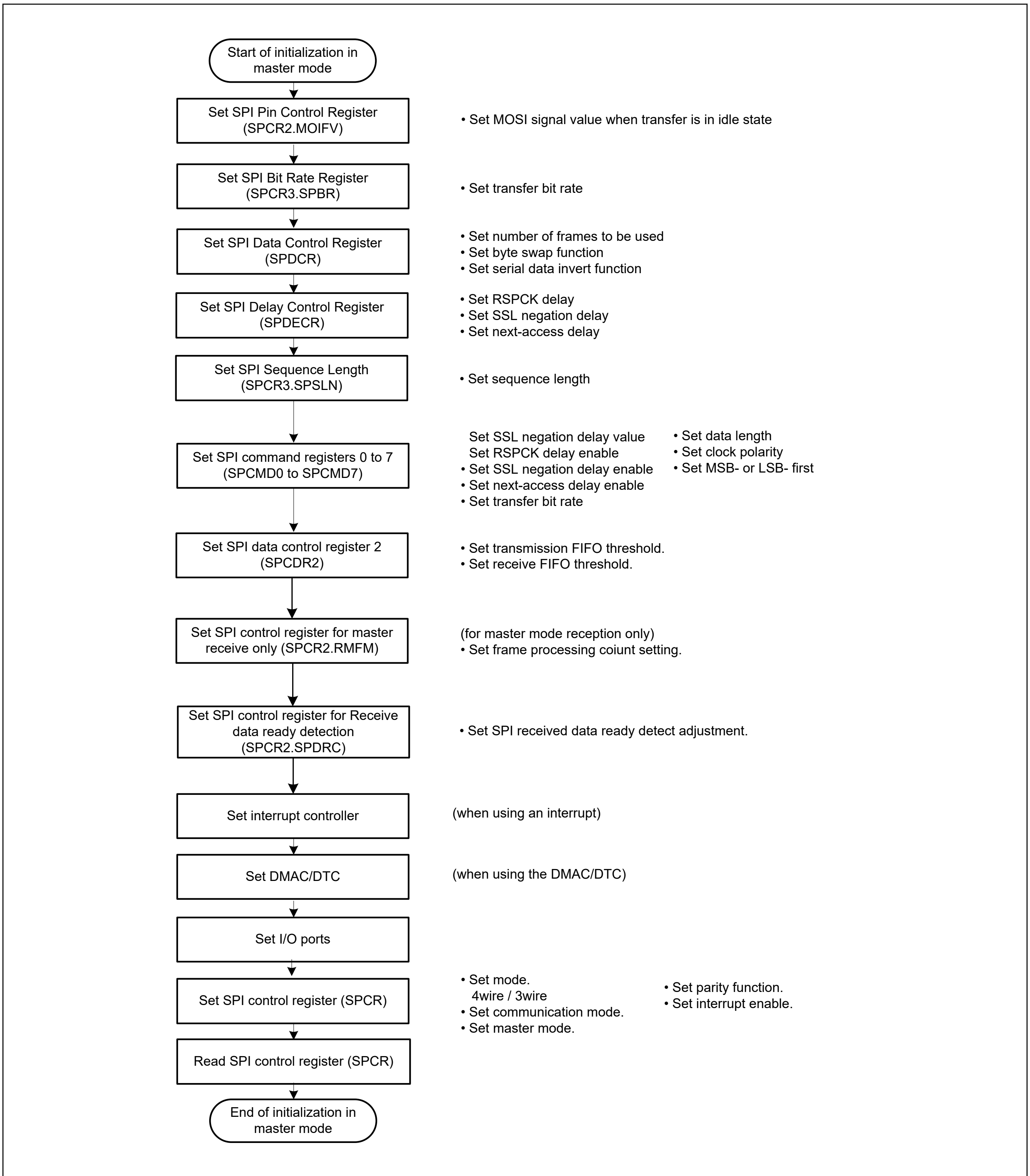


Figure 43.76 Example of initialization flow in master mode for clock synchronous operation

(5) Software processing flow

Software processing during clock synchronous master operation is the same as that for SPI master operation. For details, see (9) Software processing flow in section 43.3.12.1. Master Mode Operation. Mode fault errors do not occur in clock synchronous operation.

43.3.13.2 Slave Mode Operation

(1) Starting serial transfer

When the SPCR.SPMS bit is 1, the first RSPCKn edge triggers the start of a serial transfer in the SPI, and the SPI drives the MISOn output signal. The SSLn0 input signal is not used in clock synchronous operation. For details on the SPI transfer format, see section 43.3.5. Transfer Formats.

(2) Terminating serial transfer

The SPI terminates the serial transfer after detecting an RSPCKn edge corresponding to the final sampling timing. When the number of data stored in the receive FIFO < the number of FIFO stages, on termination of serial transfer, the SPI copies received data from the shift register to the receive buffer of the SPDR register. On termination of a serial transfer, the SPI changes the status of the shift register to empty regardless of the receive buffer.

The final sampling timing changes depending on the bit length of transfer data. In slave mode, the SPI data length depends on the SPCMD0.SPB[4:0] bits setting. For details on the SPI transfer format, see section 43.3.5. Transfer Formats.

(3) Initialization flow

Figure 43.77 shows an example of initialization flow for clock synchronous operation when the SPI is used in slave mode. For a description of how to set up the ICU, DMAC or DTC, and I/O ports, see the descriptions given in the individual blocks.

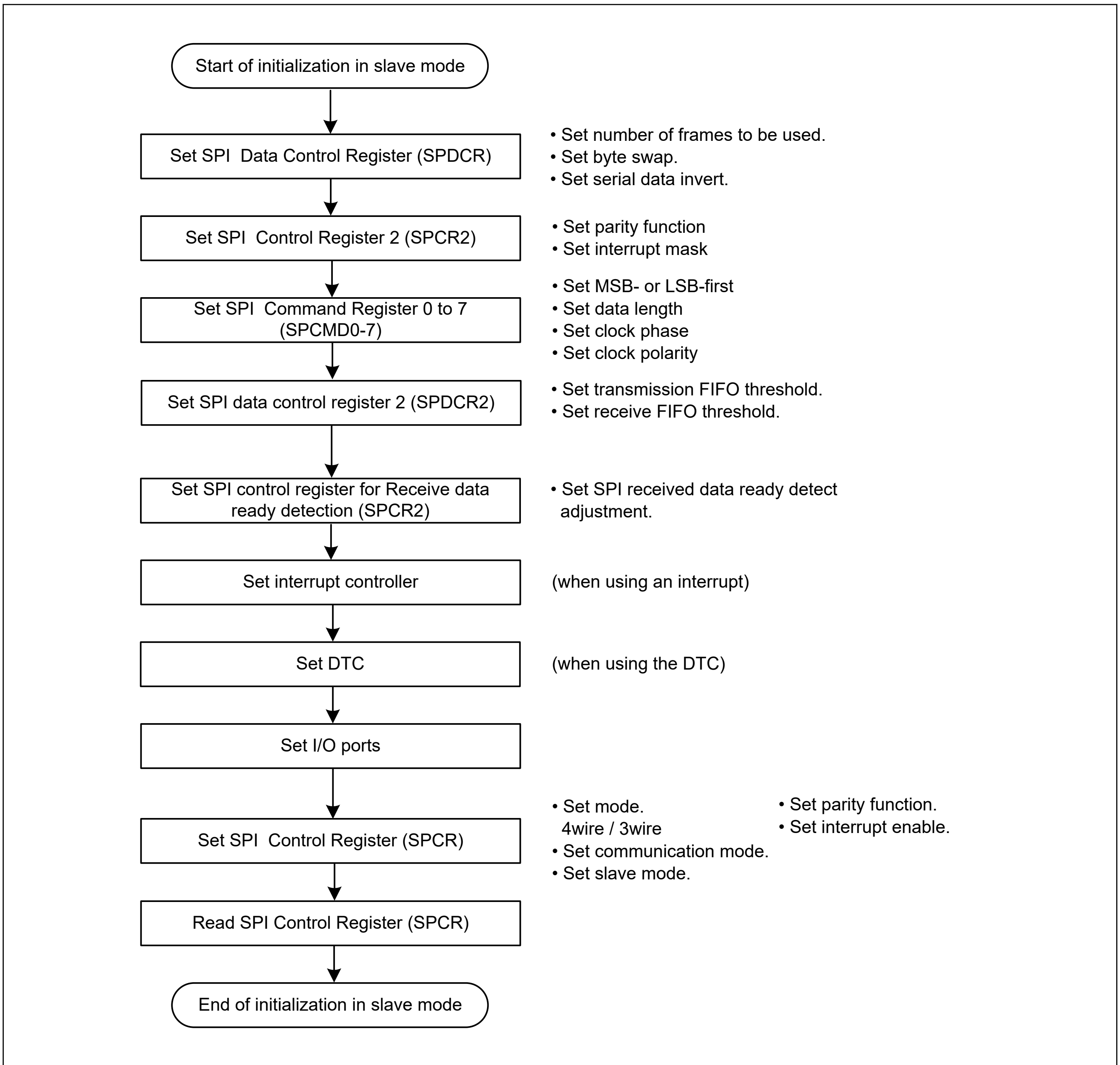


Figure 43.77 Example of initialization flow in slave mode for clock synchronous operation

(4) Software processing flow

Software processing during clock synchronous slave operation is the same as that for SPI slave operation. For details, see (6) Software processing flow. Mode fault errors do not occur in clock synchronous mode.

43.3.14 Loopback Mode

When 1 is written to the SPCR2.SPLP2 bit or SPCR2.SPLP bit, the SPI shuts off the path between the MISOn pin and the shift register if the SPCR.MSTR bit is 1, or between the MOSIn pin and the shift register if the SPCR.MSTR bit is 0, and connects the input and output paths of the shift register, establishing a loopback mode. The SPI does not shut off the path between the MOSIn pin and the shift register if the SPCR.MSTR bit is 1, or between the MISOn pin and the shift register if the SPCR.MSTR bit is 0. This is called loopback mode. When a serial transfer is executed in loopback mode, the transmit data for the SPI or the reversed transmit data becomes the received data for the SPI.

Table 43.13 lists the relationship between the SPLP2 and SPLP bits and the received data. Figure 43.78 shows the configuration of the shift register I/O paths when the SPI in master mode is set to loopback mode (SPCR2.SPLP2 = 0, SPCR2.SPLP = 1).

Table 43.13 SPLP2 and SPLP bit settings and received data

SPCR2.SPLP2 bit	SPCR2.SPLP bit	Received data
0	0	Input data from the MOSIn pin or MISOn pin
0	1	Inverted transmit data
1	0	Transmit data
1	1	Transmit data

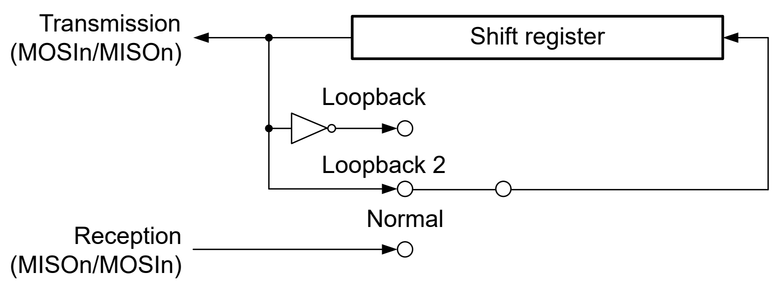


Figure 43.78 Configuration of shift register I/O paths in loopback mode for master mode

43.3.15 Self-Diagnosis of Parity Bit Function

The parity circuit consists of a parity bit adding unit used for transmit data and an error detecting unit used for received data. To detect defects in the parity bit adding unit and error detecting unit, the parity circuit performs self-diagnosis as shown in Figure 43.79.

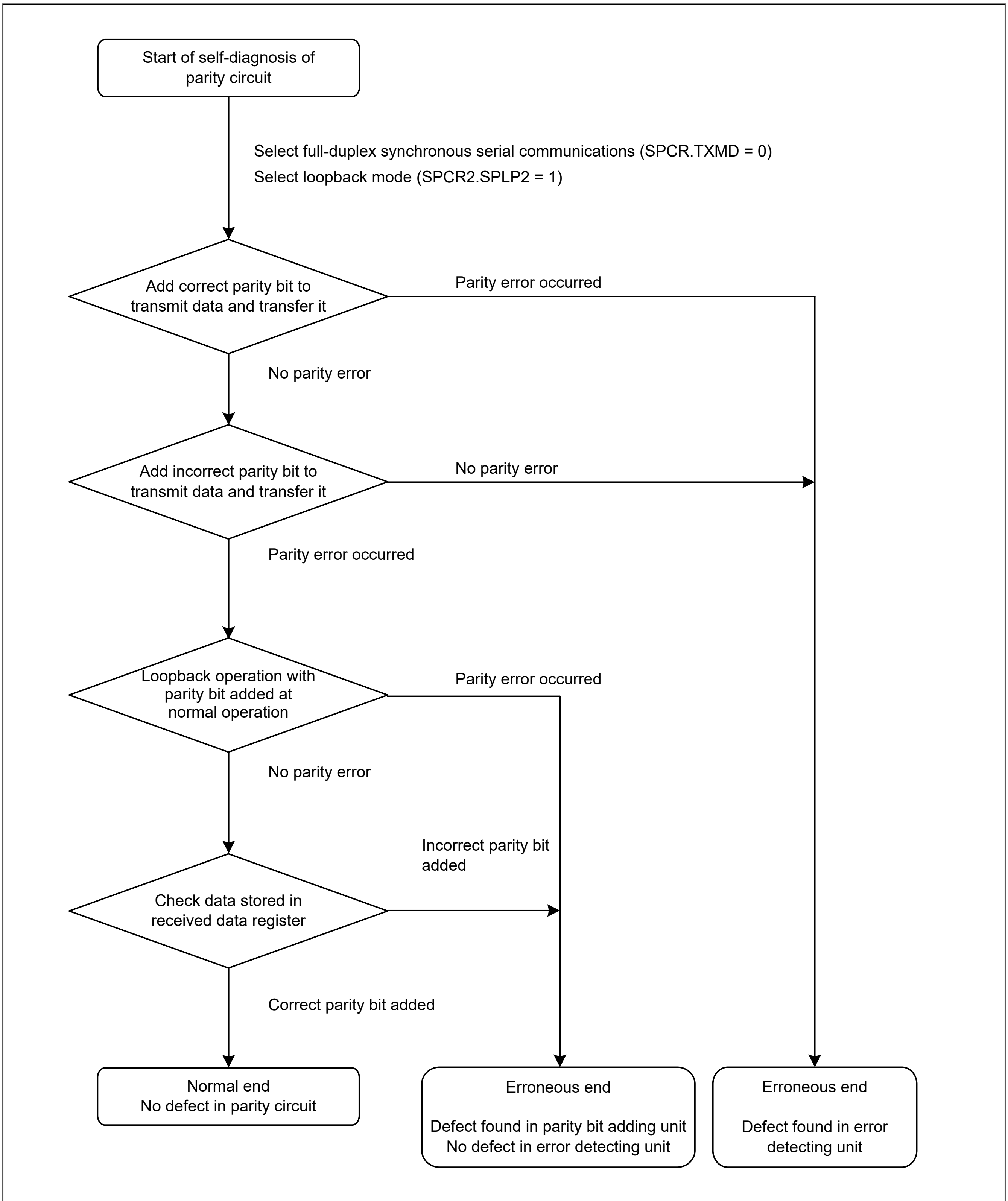


Figure 43.79 Self-diagnosis flow for parity circuit

43.3.16 Interrupt Sources

The SPI has the following interrupt sources:

- Receive buffer full
- Transmit buffer empty

- SPI error (mode-fault, underrun, overrun, or parity error)
- SPI idle
- Communication-end

The DMAC or DTC can be activated by the receive buffer full or transmit buffer empty interrupt to perform data transfer.

Because the vector address for the SPIi_SPEI (SPI error interrupt) is allocated to interrupt requests on mode-fault, underrun, overrun, and parity errors, the actual interrupt source must be determined from the flags. Interrupt sources for the SPI are listed in Table 43.14. An interrupt is generated on satisfaction of one of the interrupt conditions in Table 43.14. Clear the receive buffer full and transmit buffer empty sources through a data transfer.

When using the DMAC or DTC to perform data transmission and reception, you must first set up the DMAC or DTC to be in a transfer-enabled status before setting the SPI. For information on setting up the DMAC or DTC, see section 17, DMA Controller (DMAC) and section 18, Data Transfer Controller (DTC).

If the conditions for generating a transmit buffer empty or receive buffer full interrupt occur while the ICU.IELSRn.IR flag is 1, the interrupt is not output as a request for the ICU but is retained internally (the capacity for retention is one request per source). A retained interrupt request is output when the ICU.IELSRn.IR flag becomes 0. A retained interrupt request is automatically discarded when it is output as an actual interrupt request. The interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) for an internally retained interrupt request can also be set to 0.

Table 43.14 SPI interrupt sources

Interrupt source	Symbol	Interrupt condition	DTC/DMAC activation
Receive buffer full	SPIi_SPRI	The receive buffer becomes full (SPSR.SPRF flag is 1) while the SPCR.SPRIE bit is 1 or The receive data become ready (SPSR.SPDRF flag is 1) while the SPCR.SPDRES bit is 0	Possible
Transmit buffer empty	SPIi_SPTI	The transmit buffer becomes empty (SPSR.SPTEF flag is 1) while the SPCR.SPTIE bit is 1	Possible
SPI error (mode-fault, underrun, overrun, or parity error)	SPIi_SPEI	The SPSR.MODF, OVRF, or PERF flag sets to 1, or the SPSR.SPDRES and SPDRF flag set to 1 while the SPCR.SPEIE bit is 1	Impossible
SPI idle	SPIi_SPII	The SPSR.IDLNF flag sets to 0 while the SPCR.SPIIE bit is 1	Impossible
Communication-end	SPIi_SPCEND	CENDIE = 1 and CENDF = 1	Impossible

43.4 Event Link Controller Event Output

The Event Link Controller (ELC) can produce the following event output signals:

- Receive buffer full event output
- Transmit buffer empty event output
- Mode-fault, underrun, overrun, or parity error event output
- SPI idle event output
- Transmission-completed event output

The event link output signal is output regardless of the interrupt enable bit setting.

43.4.1 Receive Buffer Full Event Output

When the number of data stored in the receive FIFO > the threshold value, or when the number of data stored in the receive FIFO ≤ the threshold value and SPDRES = 0 has elapsed after writing to the receive FIFO, and the SPDRC [7:0] has elapsed outputs an event.

43.4.2 Transmit Buffer Empty Event Output

An event is output when the number of empty transmission FIFO stage > the threshold or when the SPCR.SPE bit changes from 0 to 1.

43.4.3 Mode-Fault, Underrun, Overrun, Parity Error, or Received Data Ready Event Output

This event signal is output when mode-fault, underrun, overrun, or parity error is detected. See section 43.5.4. Constraints on Mode-Fault, Underrun, Overrun, Parity Error, or Receive Data Ready Event Output if using this event signal.

(1) Mode-fault

Table 43.15 lists the conditions for occurrence of a mode-fault event.

Table 43.15 Conditions for mode-fault occurrence

SPI mode	SPCR.MODFEN bit	SSLn0 pin	Remarks
SPI operation (SPMS = 0) Slave (SPCR.MSTR = 0) Motorola-SPI (SPCR.SPFRF = 0)	1	Not active	Event is output only when the SSLn0 pin is deactivated during transmission
SPI operation (SPMS = 0) Slave (SPCR.MSTR = 0) TI-SSP (SPCR.SPFRF = 1)	1	active	Event is output only when the SSLn0 pin is activated during transmission

(2) Underrun

This event signal is output in response to an underrun when a serial transfer starts while the transmission data is not ready, and the value of the SPCR.MSTR bit is 0 and the SPCR.SPE bit is 1. Under these conditions, the MODF and UDRF flags are set to 1.

(3) Overrun

This event signal is output in response to an overrun when a serial transfer completes while the receive buffer contains unread data and the value of the SPCR.TXMD[1:0] bits are 00b or 10b. Under these conditions, the OVRF flag is set to 1.

(4) Parity error

This event signal is output in response to a parity error detected on completion of a serial transfer while the value of the SPPE bit in SPCR is 1.

(5) Receive Data Ready

When TXMD[1:0] of SPCR = 00b or 10b and SPDRES = 1 as the receive data ready event output condition, the number of data stored in the receive FIFO is received after writing the receive FIFO. An event will be output when the set value of SPDRC[7:0] has elapsed while the number is less than the FIFO threshold.

43.4.4 SPI Idle Event Output

(1) In master mode

In Transmit-Receive / Transmit-only master mode, an event is output when the IDLNF flag in SPSR changes from 1 to 0.

The IDLNF flag changes from 1 to 0 only when either of the conditions 1) and 2) below is met.

1. The SPE bit in SPCR is cleared to 0 (SPI initialized) during transmission.
2. All of the following three conditions are met.
    - The transmit buffer (SPTXn, n = 0 to 3) is empty (next transfer data has not been set).
    - The SPCP [2:0] bits in SPSR are 000b (at the start of sequence control).
    - Operation completed by the next access delay (when the master main state machine transitions to the idle state).

In receive only master mode

Any of the following 2 conditions is met.

1. SPE bit of SPCR is 0 (SPI initialization)
2. When any of the following is met

- When RMFM [4:0] = 0x00, after writing 1 to RMEDTG, operation completed by the next access delay (when the master main state machine transitions to the idle state).
- When RMFM [4:0] ≠ 0x00, after writing 1 to RMEDTG, operation completed by the next access delay (when the master main state machine transitions to the idle state).
- When RMFM [4:0] ≠ 0x00, the SPI internal sequencer transitions to the idle state after operation completed by the next access delay (when the master main state machine transitions to the idle state).

(2) In slave mode

In slave mode, an event is output when the SPCR.SPE bit is set to 0 (SPI is initialized).

43.4.5 Communication End Event Output

In master mode, an event is output when the IDLNF flag (SPI idle flag) changes from 1 to 0. In slave mode, an event occurs with conditions shown in Table 43.16 and Table 43.17

Table 43.16 Communication End Event Generating Conditions (transmit-receive/transmit slave mode)

	Transmit Buffer Status	Shift Register Status	Others
SPI operation (SPMS = 0, SPFRF = 0)	Empty	Empty	SSLn0 input is negated
SPI operation (SPMS = 0, SPFRF = 1)	Empty	Empty	SSL negation delay completed
Clock synchronous operation (SPMS = 1)	Empty	Empty	The last even edge of RSPCK of last data was detected (CPHA = 1)

Table 43.17 Communication End Event Generating Conditions (receive only slave mode)

	Others
SPI operation (SPMS = 0, SPFRF = 0)	After storing the frames corresponding to the SPFC setting value in the receive buffer, negate SSLn0 input.
SPI operation (SPMS = 0, SPFRF = 1)	After storing the frames corresponding to the SPFC setting value in the receive buffer, SSL negation delay completed
Clock synchronous operation (SPMS = 1)	RSPCK last even edge detection when receiving the last frame for the SPFC set value (CPHA = 1)

Regardless of master mode or slave mode, no event is output when 0 is written to the SPCR.SPE bit during transmission or when the SPCR.SPE bit is cleared due to a mode fault error or an underrun error.

A communication end event is output at the following timing. The communication end event output timing in master operation is omitted because it is output at the same timing as an idle event.

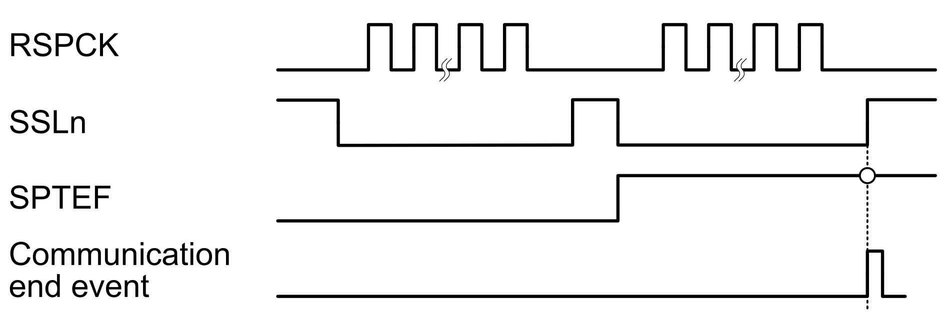


Figure 43.80 Communication End Event Output Timing (Transmit slave mode, Motorola SPI operation)

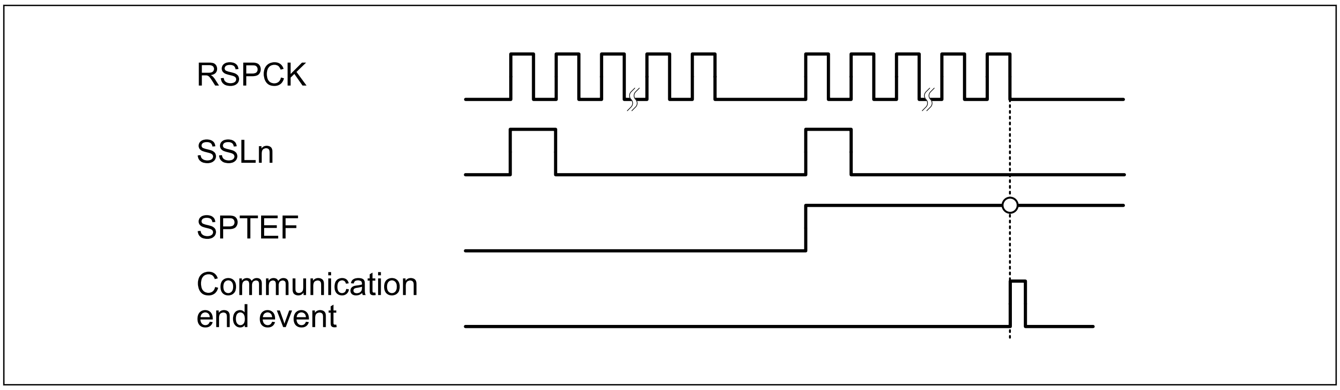


Figure 43.81 Communication End Event Output Timing (Transmit slave mode, TI-SSP Operation)

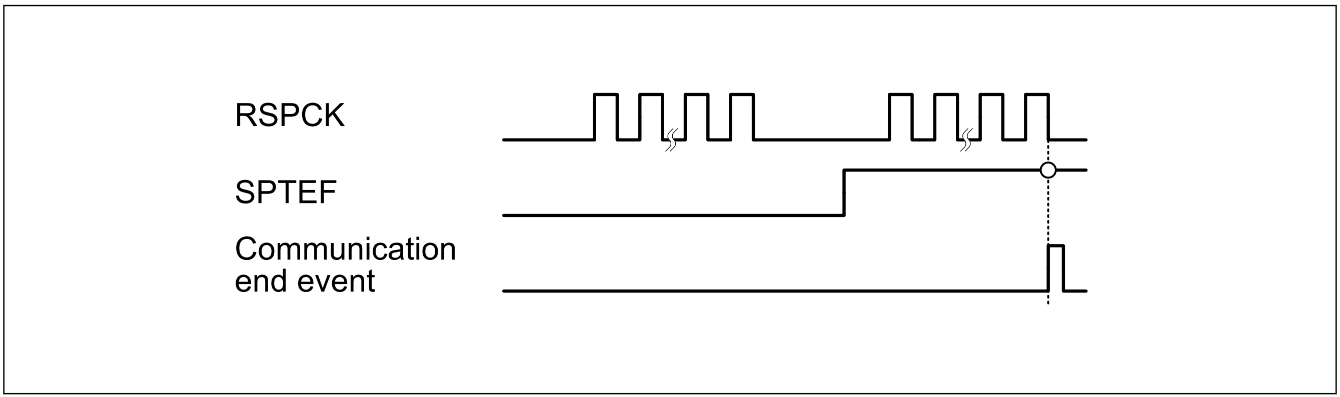


Figure 43.82 Communication End Event Output Timing (Transmit slave mode, Clock Synchronous Operation)

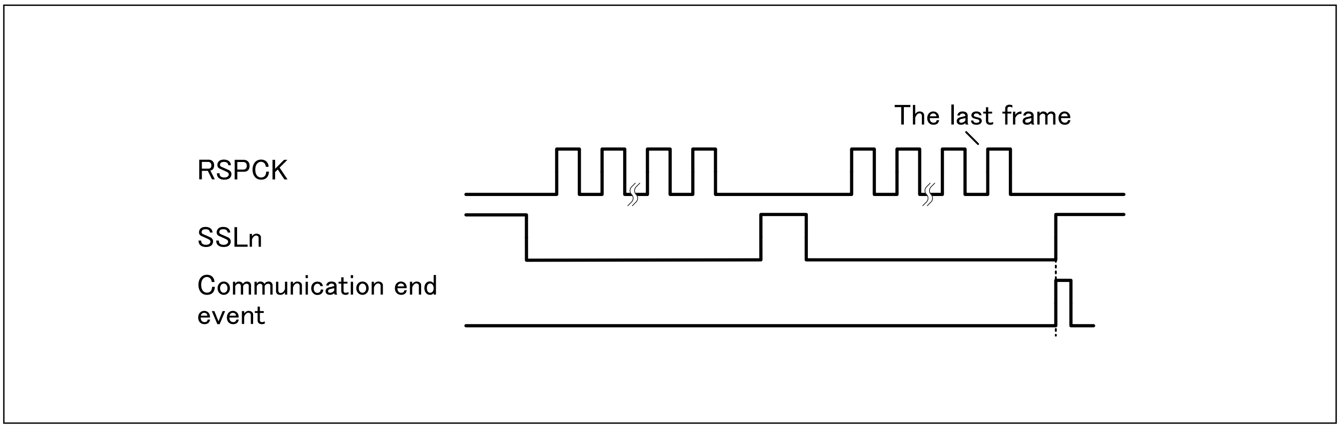


Figure 43.83 Communication End Event Output Timing (Receive only slave mode, Motorola SPI operation)

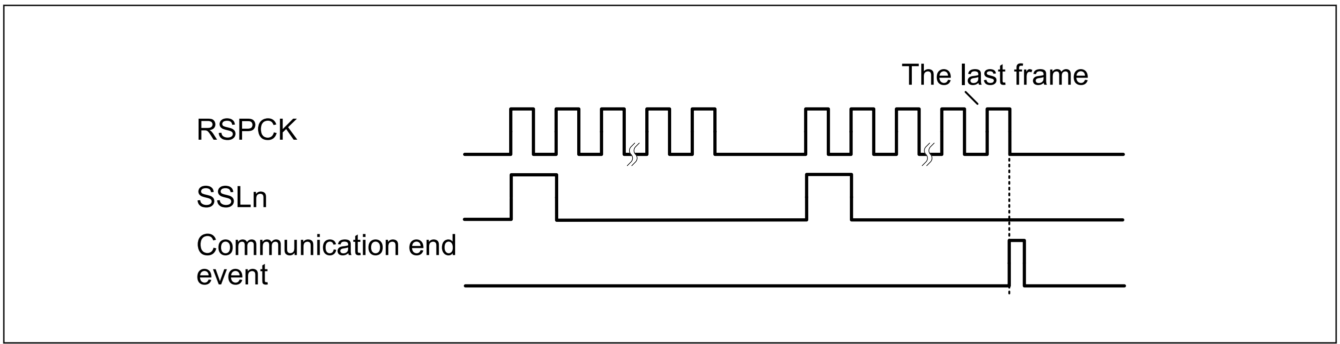


Figure 43.84 Communication End Event Output Timing (Receive only slave mode, TI-SSP Operation)

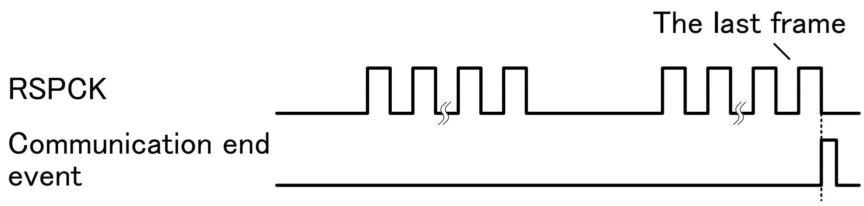


Figure 43.85 Communication End Event Output Timing (Receive only slave mode, Clock Synchronous Operation)

43.4.6 Synchronization Bypass Function

SPI has an internal clock (PCLK) and an operation clock (TCLK), and each has its own operation circuit. Therefore, a synchronization circuit is inserted between the signals between different clocks, and a signal delay between different clocks requires a synchronization delay time.

However, the synchronization circuit can be bypassed by the BPEN = 1 of the SPI control register (SPCR) only when the same clock is input as the internal bus clock and the operation clock. In this case, the synchronization delay time is excluded, and responsiveness is improved.

In addition, SPI has a synchronization circuit between the communication clock (RSPCK) and the operation clock (TCLK), but this synchronization circuit cannot be bypassed.

43.5 Usage Notes

43.5.1 Settings for the Module-Stop State

The Module Stop Control Register B (MSTPCRB) can enable or disable the SPI operation. The SPI is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details on the Module Stop Control Register B, see section 11, Low Power Mode.

43.5.2 Constraint on Low-Power Functions

When using the module-stop function and entering a low-power mode other than CPU Sleep mode or CPU Deep Sleep mode, set the SPCR.SPE bit to 0 before completing communication.

43.5.3 Constraints on Starting Transfer

If the ICU.IELSRn.IR flag is 1 when transfer starts, the interrupt request is internally retained, which can lead to unanticipated behavior of the ICU.IELSRn.IR flag.

To prevent this, use the following procedure to clear interrupt requests before enabling operations (by setting the SPCR.SPE bit to 1):

1. Confirm that transfer stopped (the SPCR.SPE bit is 0).
2. Set the associated interrupt enable bit (SPCR.SPTIE bit or SPCR.SPRIE bit) to 0.
3. Read the associated interrupt enable bit (SPCR.SPTIE bit or SPCR.SPRIE bit) and confirm that its value is 0.
4. Set the ICU.IELSRn.IR flag to 0.

43.5.4 Constraints on Mode-Fault, Underrun, Overrun, Parity Error, or Receive Data Ready Event Output

Using the mode-fault, underrun, overrun, parity error, or receive data ready event is prohibited if the SPI is in multi-master mode (when the SPCR.SPMS bit is 0, the SPCR.MSTR bit is 1, and the SPCR.MODFEN bit is 1).

43.5.5 Constraints on the SPSR.SPRF and SPSR.SPTEF Flags

If the polling flags, SPRF and SPTEF, are used, using the interrupts is prohibited, and you must set the SPCR.SPRIE and SPCR.SPTIE bits to 0. Either the interrupts or the flags can be used, but not both.

44. Octal Serial Peripheral Interface (OSPI)

This is the OSPI_B version of the OSPI peripheral module.

Flash memory ISSI IS25WX064-JWLE is packaged for SIP product.

OSPI_B is referred to as OSPI in this chapter.

44.1 Overview

The xSPI (eXpanded Serial Peripheral Interface) protocol specifies the interface for nonvolatile memory devices, which provides high data throughput, low signal count, and limited backward compatibility with legacy SPI devices. The electrical interface can deliver up to 333 MB per second raw data throughput. The OSPI is compliant with JEDEC standard JESD251 (profiles 1.0 and 2.0), JESD251-1 and JESD252.

JESD251 specifies two interface profiles where profile 1.0 is Octal SPI and profile 2.0 is HyperBus™ (HyperRAM™ and HyperFlash™). OSPI supports the QSPI protocol.

Table 44.1 lists the OSPI specifications, Figure 44.1 shows a block diagram, and Table 44.2 lists the I/O pins.

Table 44.1 OSPI specifications

Item	Description
Number of units	2 units
Protocol	Compliant with the xSPI protocol*1
Data transmission and reception	Issue transaction for up to 2 slaves as masters. Only one of the memory devices can operate at a time.
Transfer speed	Support the transfer at xSPI333
Mode	● Support the following protocol modes: – 1-, 4-, 8-pin with SDR/DDR (1S-1S-1S, 4S-4D-4D, 8D-8D-8D) – 2-, 4-pin with SDR (1S-2S-2S, 2S-2S-2S, 1S-4S-4S, 4S-4S-4S) ● Configurable address length ● Configurable initial access latency cycle ● Support XiP mode
OSPI function	● Support write data mask ● Support in-band reset ● Memory-mapping – Unit 0 supports up to 256 MB address space each CS – Unit 1 supports up to 128 MB address space each CS – 1 control channel per unit for the bus masters For more details, see section 15, Buses. – Prefetch function for burst-read with low latency – Outstanding buffer for burst-write with high throughput ● Manual command – Configurable up to 4 commands – Status register polling function ● Input strobe port timing shift
Decryption function	Decryption on-the-fly is available for memory-map read
Interrupt source	● Error interrupt ● Completion interrupt
Module-stop function	Module-stop state can be set to reduce power consumption
TrustZone Filter	● Security attribution can be set for the IO register area ● External address space is defined as non-secure

Note 1. The OSPI is compliant with JEDEC standard JESD251 (profiles 1.0 and 2.0), JESD251-1 and JESD252.

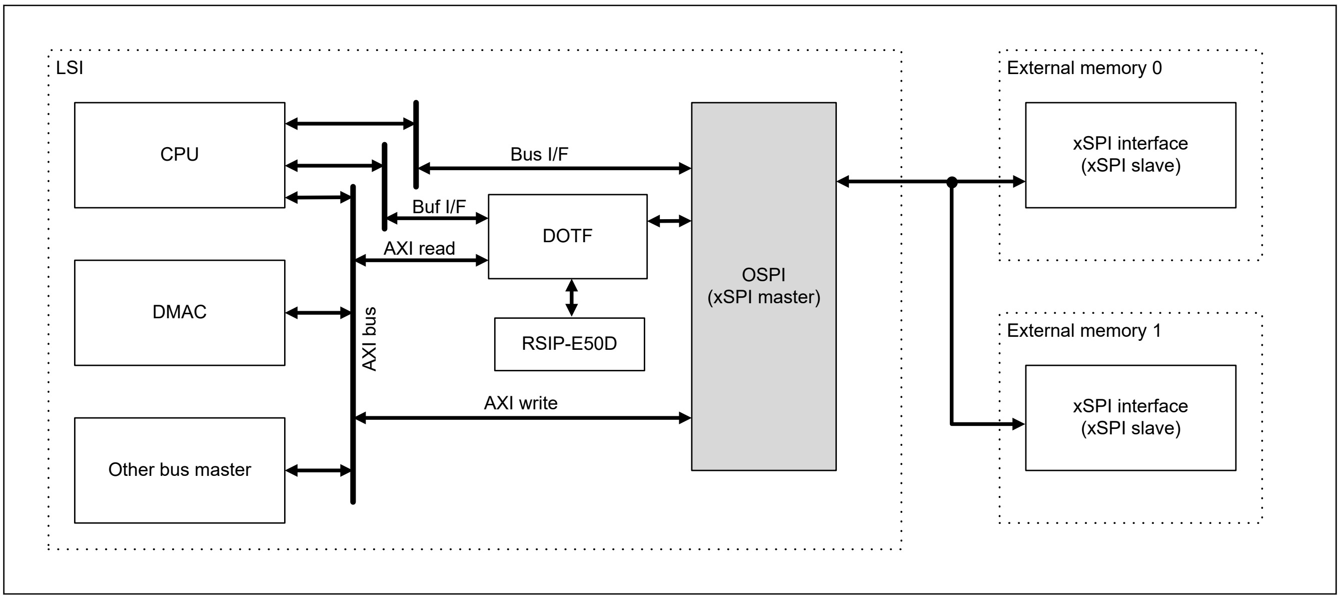


Figure 44.1 Block diagram

Table 44.2 OSPI I/O pins

Pin name	I/O	Function
OM_n_SCLK	Output	Clock positive
OM_n_SCLKN	Output	Clock negative
OM_n_CS0	Output	Chip selection for slave 0
OM_n_CS1	Output	Chip selection for slave 1
OM_n_DQS	I/O	Read data strobe/write data mask
OM_n_SIO0	I/O	Data 0 input/output
OM_n_SIO1	I/O	Data 1 input/output
OM_n_SIO2	I/O	Data 2 input/output
OM_n_SIO3	I/O	Data 3 input/output
OM_n_SIO4	I/O	Data 4 input/output
OM_n_SIO5	I/O	Data 5 input/output
OM_n_SIO6	I/O	Data 6 input/output
OM_n_SIO7	I/O	Data 7 input/output
OM_n_RESET	Output	Master reset status for slave 0, 1
OM_n_RSTO1	Input	Slave reset status for slave 1
OM_n_ECSINT1	Input	Interrupt for slave 1/error correction status for slave 1
OM_n_WP1	Output	Write protect for slave 1

Note: For OM_n_SIO7-0, OM_n_SCLK, OM_n_SCLKN and OM_n_DQS pins, 30Ω ±1% resistor must be put on the board to comply with JESD251 I/O driver definition. Renesas recommends this for the operation with proper signal quality.

Note: n = 0, 1

44.2 Register Descriptions

44.2.1 OSPI Configuration Registers

These registers configure the xSPI Master function. These registers should be configured in the initialization phase before issuing xSPI transactions.

44.2.1.1 WRAPCFG : OSPI Wrapper Configuration Register

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DSSFTCS1[4:0]					—	—	—	—	—	—	—	—	—	—	—	DSSFTCS0[4:0]					—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
12:8	DSSFTCS0[4:0]	DS Shift for Slave 0 These bits configure the number of delay cells for an OM_DQS port. A delay cell is used to adjust the DS sampling timing. When automatic calibration is enabled, it can be updated automatically. In this case, it should not be written by the user. 0x00: No shift 0x01: Add a delay of one cell ⋮ 0x1E: Add a delay of 30 cells 0x1F: Add a delay of 31 cells	R/W
23:13	—	These bits are read as 0. The write value should be 0.	R/W
28:24	DSSFTCS1[4:0]	DS Shift for Slave 1 The function is the same as one of slave 0.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The WRAPCFG register has functions to configure the xSPI Master function.

44.2.1.2 COMCFG : OSPI Common Configuration Register

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OENEGEX	OEASTEX
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
16	OEASTEX	Output Enable Asserting Extension This bit extends one cycle output enable of data and DS during output enable asserting. When set to 1, CS asserting should be extended (LIOCFGCSn.CSASTEX = 1 (n = 0, 1)). This bit is used for case with latency cycle because OSPI output data can be conflicted with OSPI input data. 0: Do not extend one cycle output enable 1: Extend one cycle output enable	R/W

Bit	Symbol	Function	R/W
17	OENEGEX	Output Enable Negating Extension This bit extends one cycle output enable of data and DS during output enable negating. This bit is used for case with latency cycle because OSPI output data can be conflicted with OSPI input data. 0: Do not extend one cycle output enable 1: Extend one cycle output enable	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The COMCFG register has functions to configure the xSPI Master function.

44.2.1.3 BMCFGCHn : OSPI Bridge Map Configuration Register Chn (n = 0)

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x008 + 0x004 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CMBTIM[7:0]								—	—	—	—	—	—	—	PREEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MWRSIZE[7:0]								MWRCOMB	—	—	—	—	—	—	WRMD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	WRMD	System Bus Write Response Mode This bit selects the timing of a system bus write response in memory-mapping mode. When set to 1, this bit returns the response after transmitting a frame on the xSPI bus. When the system bus write response mode is enabled, the memory write combination mode must be disabled. 0: Return response after storing to the internal write buffer 1: Return response after issuing a write transaction to the xSPI bus	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	MWRCOMB	Memory Write Combination Mode This bit selects to combine the OSPI data in write access of memory-mapping mode. When set to 0, the OSPI data size depends on the burst type and size of the system bus. When this field is set to 1, the data size depends on the MWRSIZE[7:0] field.*1 When set to 1, any write transaction can be temporarily held in this xSPI master. 0: Disable combination mode 1: Enable combination mode	R/W
15:8	MWRSIZE[7:0]	Memory Write Size These bits select the size to combine incremental address in memory-mapping mode. It transmits an xSPI frame with the data combined up to the configured size while the address is incremental. When a nonincremental address or a read transaction is detected before reaching the target size, it transmits the pending data into the xSPI bus. 0x00: Combine incremental address of up to 4 bytes 0x01: Combine incremental address of up to 8 bytes : 0x0E: Combine incremental address of up to 60 bytes 0x0F: Combine incremental address of up to 64 bytes 0xFF: Combine incremental address of up to 2 bytes Others: Setting prohibited	R/W
16	PREEN	Prefetch Enable This bit enables the prefetch function for a read transaction in memory-mapping mode. It can reduce the latency for read transaction with incremental address. 0: Disable prefetch function 1: Enable prefetch function	R/W

Bit	Symbol	Function	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W
31:24	CMBTIM[7:0]	Combination Timer These bits specify the expiration period of a combination timer. This timer is counted by PCLKA. 0x00 means disabling the combination timer. When the timer is expired, the data in the combination buffer is pushed to the memory device.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Writing to OSPI memory space other than 64 bit is prohibited during combination mode.

The BMCFGCHn register has functions to configure the xSPI Master function.

44.2.1.4 CMCFG0CSn : OSPI Command Map Configuration Register 0 CSn (n = 0, 1)

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x010 + 0x010 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ADDRPCD[7:0]								ADDRPEN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	ARYAMD	WPBSTMD	ADDSIZE[1:0]		FFMT[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	FFMT[1:0]	Frame Format These bits configure xSPI frame format in memory-mapping mode. See Table 44.7 for detail. 0 0: Normal format: Command 1 byte, address ADDSIZE, data up to system bus transaction 0 1: 8D-8D-8D profile 1.0 format: Command 2 bytes, address ADDSIZE, data up to system bus transaction 1 0: 8D-8D-8D profile 2.0 Command Modifier format: Command & Modifier 6 bytes, data up to system bus transaction 1 1: 8D-8D-8D profile 2.0 Commands with Extended Command Modifier format: Command & Modifier 6 bytes, data up to system bus transaction	R/W
3:2	ADDSIZE[1:0]	Address Size These bits configure the number of address byte in memory-mapping mode. For 8D-8D-8D profile 2.0, configure to 4 bytes. 0 0: 1 byte (256-byte address space) 0 1: 2 bytes (64-KB address space) 1 0: 3 bytes (16-MB address space) 1 1: 4 bytes (4-GB address space)	R/W
4	WPBSTMD	Wrapping Burst Mode When this bit is set to 1, the wrapping boundary between system bus access and xSPI memory should be matched. 0: Separate xSPI transfer at the wrapping address boundary 1: Do not separate xSPI transfer at the wrapping address boundary	R/W
5	ARYAMD	Array Address Mode When this bit is set to 1, address for memory is mapped as {A[25:10], A[9:4], 6{RSV}, A [3:0]} where A[25:0] is normal address, and RSV is reserved value(0b). This bit is valid only when FFMT = 1d. 0: Normal address mode 1: Array address mode	R/W
15:6	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
23:16	ADDRPEN[7:0]	Address Replace Enable These bits select the bits to replace the MSByte of system bus address in memory-mapping mode. 0: No replacement (xSPI frame address field is the same as system bus address) 1: Replacement <table><tr><td>ADDSIZE[1:0]</td><td>Replace target bits of system bus address</td></tr><tr><td>00</td><td>bit 7 to bit 0</td></tr><tr><td>01</td><td>bit 15 to bit 8</td></tr><tr><td>10</td><td>bit 23 to bit 16</td></tr><tr><td>11</td><td>bit 31 to bit 24</td></tr></table>	R/W
31:24	ADDRPCD[7:0]	Address Replace Code These bits configure the code to replace the MSByte of system bus address in memory-mapping mode. It replaces the corresponding bits when the Address Replace Enable bit is set to 1.	R/W

Note: S-TYPE-3, P-TYPE-3

The CMCFG0CSn register has functions to configure the xSPI Master function.

44.2.1.5 CMCFG1CSn : OSPI Command Map Configuration Register 1 CSn (n = 0, 1)

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x014 + 0x010 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	RDLATE[4:0]					RDCMD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RDCMD[15:0]	Read Command These bits configure the command field of a read transaction in memory-mapping mode. Normal format and 8D-8D-8D profile 2.0 format use only upper 1 byte. 8D-8D-8D profile 1.0 format uses 2 bytes.	R/W
20:16	RDLATE[4:0]	Read Latency Cycle These bits configure the latency cycle of a read transaction in memory-mapping mode. 0x00: No latency 0x01: 1 cycle ⋮ ⋮ 0x08: 8 cycles (default) ⋮ ⋮ 0x1E: 30 cycles 0x1F: 31 cycles Others: Setting prohibited	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The CMCFG1CSn register has functions to configure the xSPI Master function.

44.2.1.6 CMCFG2CSn : OSPI Command Map Configuration Register 2 CSn (n = 0, 1)

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x018 + 0x010 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	WRLATE[4:0]					WRCMD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	WRCMD[15:0]	Write Command These bits configure the command field of a write transaction in memory-mapping mode. Normal format and 8D-8D-8D profile 2.0 format use only upper 1 byte. 8D-8D-8D profile 1.0 format uses 2 bytes.	R/W
20:16	WRLATE[4:0]	Write Latency Cycle These bits configure the latency cycle of a write transaction in memory-mapping mode. 0x00: No latency 0x01: 1 cycle ⋮ ⋮ 0x08: 8 cycles (default) ⋮ ⋮ 0x1E: 30 cycles 0x1F: 31 cycles Others: Setting prohibited	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The CMCFG2CSn register has functions to configure xSPI the Master function.

44.2.1.7 LIOCFGCSn : OSPI Link I/O Configuration Register CSn (n = 0, 1)

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x050 + 0x004 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DDRSMPEX[3:0]				SDRSMPSFT[3:0]				SDRSMPMD	SDRDRV	CSNEGEX	CSASTEX	CSMIN[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	WRMSKMD	LATEMD	PRTMD[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PRTMD[9:0]	Protocol Mode These bits configure the protocol mode and the pin to sample data inputs. When using data strobe instead of SPI clock for sampling in SDR mode, set PRTMD[9:0] to 1. 0x000: 1S-1S-1S 0x3B2: 4S-4D-4D 0x3FF: 8D-8D-8D 0x048: 1S-2S-2S 0x049: 2S-2S-2S 0x090: 1S-4S-4S 0x092: 4S-4S-4S Others: Setting prohibited	R/W

<table>
<tr><th>Bit</th><th>Symbol</th><th>Function</th><th>R/W</th></tr>
<tr><td>10</td><td>LATEMD</td><td>Latency Mode
This bit selects the behavior of the initial access latency phase for both direct-manual mode and memory-mapping mode. When set to 0, the latency cycle is equal to each configured cycle from the transmitting Address field. When set to 1, the latency cycle is incremented from the last byte pair of the Address field and is extended 2 times of each configured cycle depending on the OM_DQS port.
Latency mode is used only for profile 2.0 frame format of 8D-8D-8D protocol mode with 6 bytes Command/Address field. See the xSPI protocol specification for detail.
<table>
<tr><th>Value: Function</th><th>Frame format</th><th>Usage</th></tr>
<tr><td rowspan="2">0: Configurable latency</td><td>Profile 2.0</td><td>The configurable latency cycle should be set as minus 1.</td></tr>
<tr><td>Others</td><td>The latency cycle increments after the Address field.</td></tr>
<tr><td rowspan="2">1: Variable latency</td><td>Profile 2.0</td><td>The latency cycle increments from address [23:16] and should not be set to 1.</td></tr>
<tr><td>Others</td><td>Not support</td></tr>
</table></td><td>R/W</td></tr>
<tr><td>11</td><td>WRMSKMD</td><td>Write Mask Mode
This bit selects to use the OM_DQS port as the write data mask. It can be useful for write access of odd byte.
Write mask mode is used only for 8D-8D-8D protocol mode.
0: Write mask disable
1: Write mask enable</td><td>R/W</td></tr>
<tr><td>15:12</td><td>—</td><td>These bits are read as 0. The write value should be 0.</td><td>R/W</td></tr>
<tr><td>19:16</td><td>CSMIN[3:0]</td><td>CS Minimum Idle Term
This bit configures the minimum cycle between xSPI frames.
0x0: 1 cycle
0x1: 2 cycles
⋮ ⋮
0x7: 8 cycles (default)
⋮ ⋮
0xE: 15 cycles
0xF: 16 cycles
Others: Setting prohibited</td><td>R/W</td></tr>
<tr><td>20</td><td>CSASTEX</td><td>CS Asserting Extension
This bit extends 1 cycle Chip Select pins when asserting.
0: No extension
1: Extend 1 cycle</td><td>R/W</td></tr>
<tr><td>21</td><td>CSNEGEX</td><td>CS Negating Extension
This bit extends 1 cycle Chip Select pins when negating.
0: No extension
1: Extend 1 cycle</td><td>R/W</td></tr>
<tr><td>22</td><td>SDRDRV</td><td>SDR Driving Timing
This bit configures the timing of data output in SDR.
This bit should not be set to 1 when there is no latency cycle because OSPI output data can be conflicted with OSPI input data.
0: Drive at 1/2 cycle before CK rising edge
1: Drive at CK rising-edge</td><td>R/W</td></tr>
<tr><td>23</td><td>SDRSMPMD</td><td>SDR Sampling Mode
This bit selects the edge of sampling in SDR. In DDR, regardless of the setting, this bit samples OM_SIOn (n = 0 to 7) ports with both edges of OM_DQS.
When set to 1, this bit samples at rising edge before falling edge.
0: Samples data input at falling edge
1: Samples data input at rising edge</td><td>R/W</td></tr>
</table>

Bit	Symbol	Function	R/W
27:24	SDRSMPSFT[3:0]	SDR Sampling window shift These bits shift the timing of CK sampling in SDR. When using OM_DQS in SDR, there is no influence on the behavior. For DDR or when using OM_DQS in SDR, set these bits to 0. 0x0: Sample without delay 0x1: Sample at 1-cycle delay ⋮ ⋮ 0x6: Sample at 6-cycle delay 0x7: Sample at 7-cycle delay Others: Setting prohibited	R/W
31:28	DDRSMPEX[3:0]	DDR Sampling Window Extend These bits configure the cycle of extending the sampling window in DDR. In DDR, the input data is sampled during the expected cycle soon after the latency cycle. The input data out of range is ignored. It can be configured depending on the OM_DQS propagation delay. These bits setting should be set to satisfy t_{CKHDSH} in section 62.3.11. OSPI Timing. t_{CKHDSH} means the output delay time of the external memory. For example, the DDRSMPEX[3:0] bits are set to 0x1 in the following conditions. ● VCC is 2.7 V or above ● OM_n_SCLK is 133 MHz ● Memory output delay time (t_{CKHDSH}) is 6 ns The maximum allowable t_{CKHDSH} value is: t_{PERIOD} × (1 + DDRSMPEX[3:0]) - 8.5 = 7.5 × (1 + 1) - 8.5 = 6.5[ns] The memory output delay time should be less than this value. 0x0: Expand no cycle 0x1: Expand 1 cycle ⋮ ⋮ 0x6: Expand 6 cycles 0x7: Expand 7 cycles Others: Setting prohibited	R/W

Note: S-TYPE-3, P-TYPE-3

The LIOCFGCSn register has functions to configure the xSPI Master function.

44.2.2 OSPI Control Registers

These registers control the xSPI Master function.

44.2.2.1 BMCTL0 : OSPI Bridge Map Control Register 0

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x060

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	CH0CS1ACC[1:0]		CH0CS0ACC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
1:0	CH0CS0ACC[1:0]	System bus ch0 to slave 0 memory area access enable This field enables the access from ch0 to CS0 memory. 0 0: Read/write disable 0 1: Read enable, write disable 1 0: Read disable, write enable 1 1: Read/write enable	R/W

Bit	Symbol	Function	R/W
3:2	CH0CS1ACC[1:0]	System bus ch0 to slave 1 memory area access enable This field enables the access from ch0 to CS1 memory. 0 0: Read/write disable 0 1: Read enable, write disable 1 0: Read disable, write enable 1 1: Read/write enable	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The BMCTL0 register has functions to control the xSPI Master function.

This register is configured in the initialization phase. When the setting requires modifications after the start of an xSPI transaction, stop all communication. See section 44.3.7.2. Flow of Communication Stop before changing the value of BMCTL0.

44.2.2.2 BMCTL1 : OSPI Bridge Map Control Register 1

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x064

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	PBUFCLRCH0	—	MWRPUSHCH0	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	The write value should be 0.	W
8	MWRPUSHCH0	Memory Write Data Push for ch0 This field requests to push the pending data in combination mode.*1 0: No command 1: Push request	W
9	—	The write value should be 0.	W
10	PBUFCLRCH0	Prefetch Buffer Clear for ch0 This field requests to clear the prefetch buffer when the prefetch function is enabled.*1 Do not set this bit during memory access (COMSTT.MEMACCCH0 = 1). 0: No command 1: Clear request	W
31:11	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

Note 1. Prefetch/Combination behavior is not defined when the cycle base race condition between asserting these bits and system bus access has occurred.

The BMCTL1 register has functions to control the xSPI Master function.

44.2.2.3 CMCTLCHn : OSPI Command Map Control Register CHn (n = 0)

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x0068 + 0x004 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	XIPEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	XIPEXCODE[7:0]								XIPENCODE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	XIPENCODE[7:0]	XiP Mode Enter Code These bits configure the code to enter XiP mode in memory-mapping mode.	R/W
15:8	XIPEXCODE[7:0]	XiP Mode Exit Code These bits configure the code to exit XiP mode in memory-mapping mode.	R/W
16	XIPEN	XiP Mode Enable This bit enables XiP mode in memory-mapping mode. When set to 1, XiP enter code is inserted in the latency field, and the command field in the next transaction is omitted. When set to 0, XiP exit code is inserted in the latency field. This bit is set to 0 automatically when transmitting a XiP disable pattern. XiP mode should not be used for 8D-8D-8D protocol mode profile 2.0 frame format. 0: Disable XiP mode 1: Enable XiP mode	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The CMCTLCHn register has functions to control the xSPI Master function.

44.2.2.4 CDCTL0 : OSPI Command Manual Control Register 0

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x070

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	PERREP[3:0]				—	—	—	PERITV[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	TRNUM[1:0]		CSSEL	—	PERMD	TRREQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TRREQ	Transaction Request This bit requests to issue the transaction of manual-command. When set to 1, it starts the transaction. This bit is cleared to 0 when the transaction is completed. The transaction is canceled by clearing to 0 while the transaction is ongoing. 0: No transaction 1: Request transaction	R/W

Bit	Symbol	Function	R/W
1	PERMD	Periodic Mode This bit enables the periodic transaction mode. When set to 1, it repeats a transaction periodically and compares the read value with the expected value. Periodic transaction mode alternates the status polling operation for external memory. 0: Direct manual-command mode 1: Periodic manual-command mode	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	CSSEL	Chip Select This bit selects a target memory to issue a manual-command. 0: CS0 1: CS1	R/W
5:4	TRNUM[1:0]	Transaction Number These bits configure the number of transactions in normal manual-command mode. In periodic manual-command, regardless of this setting, the read data of the last command is compared. 0 0: Issue 1 command (using command buffer 0) 0 1: Issue 2 commands (using command buffer 0-1) 1 0: Issue 3 commands (using command buffer 0-2) 1 1: Issue 4 commands (using command buffer 0-3)	R/W
15:6	—	These bits are read as 0. The write value should be 0.	R/W
20:16	PERITV[4:0]	Periodic Transaction Interval These bits configure the interval of transaction in periodic manual-command mode. Too short interval compared with CPU bus cycle can result in no store into command buffer0. The interval should be longer than 4 times the CPU bus cycle. 0x00: 2 (= 2^1) cycles 0x01: 4 (= 2^2) cycles ⋮ ⋮ 0x1E: 2,147,483,648 (= 2^{31}) cycles 0x1F: 4,294,967,296 (= 2^{32}) cycles	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
27:24	PERREP[3:0]	Periodic Transaction Repeat These bits configure the number of transaction repetitions in periodic manual-command mode. 0x0: 1 (= 2^0) time 0x1: 2 (= 2^1) times ⋮ ⋮ 0xE: 16384 (= 2^{14}) times 0xF: 32768 (= 2^{15}) times	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The CDCTL0 register has functions to control the xSPI Master function.

44.2.2.5 CDCTL1 : OSPI Command Manual Control Register 1

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x074

Bit position:	31	0
Bit field:	PEREXP[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	PEREXP[31:0]	Periodic Transaction Expected Value These bits configure the expected value to compare with the read value in periodic manual-command mode. For example, when comparing one byte, the lower byte should be configured.	R/W

Note: S-TYPE-3, P-TYPE-3

The CDCTL1 register has functions to control the xSPI Master function.

44.2.2.6 CDCTL2 : OSPI Command Manual Control Register 2

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x078

Bit position:	31–0
Bit field:	PERMSK[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	PERMSK[31:0]	Periodic Transaction Masked Value These bits configure the masked value for the expected value in periodic manual-command mode. When 1 is set to any bit, the corresponding bit is configured as the expected value (CDCTL1.PEREXP[31:0]) is ignored. In 8D-8D-8D, the data bytes are transferred only in byte pairs on the xSPI bus. This means the dummy read data can be stored. It should be masked for unused bits. For example, when reading a lower one byte, it should be configured to 0xFFFFFF00.	R/W

Note: S-TYPE-3, P-TYPE-3

The CDCTL2 register has functions to control the xSPI Master function.

44.2.2.7 CDTBUFn : OSPI Command Manual Type Buf n (n = 0 to 3)

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x080 + 0x010 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CMD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TRTYPE	—	LATE[4:0]					DATASIZE[3:0]				ADDSIZE[2:0]			CMDSIZE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CMDSIZE[1:0]	Command Size These bits configure the size of the Command field. In 8D-8D-8D, set these bits to 10b. Do not configure both command size and address size to 0. 0 0: 0 bytes (no command phase) 0 1: 1 byte 1 0: 2 bytes Others: Setting prohibited	R/W

Bit	Symbol	Function	R/W
4:2	ADDSIZE[2:0]	Address Size These bits configure the size of the address field. 0 0 0: 0 bytes (no address phase) 0 0 1: 1 byte 0 1 0: 2 bytes 0 1 1: 3 bytes 1 0 0: 4 bytes Others: Setting prohibited	R/W
8:5	DATASIZE[3:0]	Write/Read Data Size These bits configure the size of the Data field. In 8D-8D-8D, the data bytes are transferred only in byte pairs on the xSPI bus. For example, even when configuring 1 byte for read, 2 bytes of data are received. The last byte should be ignored. Do not configure 0 bytes for read transaction. 0x0: 0 bytes (no data phase) 0x1: 1 byte ⋮ ⋮ 0x7: 7 bytes 0x8: 8 bytes Others: Setting prohibited	R/W
13:9	LATE[4:0]	Latency Cycle These bits configure the latency cycle in manual-command mode. 0x0: No latency 0x1: 1 cycle ⋮ ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
14	—	This bit is read as 0. The write value should be 0.	R/W
15	TRTYPE	Transaction Type This bit selects the type of transaction. 0: Read transaction (readout data from slave device) 1: Not read transaction	R/W
31:16	CMD[15:0]	Command (1-2 bytes) These bits configure the Command field in manual-command mode. The number of bytes configured in Command Size bit is transferred. 1S-1S-1S, 4S-4D-4D: CMD[15:8] is Command field, CMD[7:0] is not used. 8D-8D-8D profile 1.0: CMD[15:8] is Command field, CMD[7:0] is Extension field. 8D-8D-8D profile 2.0: CMD[15:0] is the upper 2 bytes of the Command & Modifier fields (bits [47:32] in the xSPI protocol)	R/W

Note: S-TYPE-3, P-TYPE-3

The CDTBUFn register has functions to control the xSPI Master function.

44.2.2.8 CDABUFn : OSPI Command Manual Address Buf n (n = 0 to 3)

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x084 + 0x10 × n

Bit position:	31 ... 0
Bit field:	ADD[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	ADD[31:0]	Address These bits configure the Address field in manual-command mode. 1S-1S-1S, 4S-4D-4D, 8D-8D-8D profile 1.0: The Address field. 8D-8D-8D profile 2.0: The lower 4 bytes of the Command & Modifier fields (bits [31:0] in the xSPI protocol).	R/W

Note: S-TYPE-3, P-TYPE-3

The CDABUFn register has functions to control the xSPI Master function.

44.2.2.9 CDD0BUFn : OSPI Command Manual Data 0 Buf n (n = 0 to 3)

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x088 + 0x10 × n

Bit position:	31 … 0
Bit field:	DATA[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	DATA[31:0]	Write/Read Data These bits configure the Data field in manual-command mode. For write transaction, configure the write data. For read transaction, the read data is stored.	R/W

Note: S-TYPE-3, P-TYPE-3

The CDD0BUFn register has functions to control the xSPI Master function.

44.2.2.10 CDD1BUFn : OSPI Command Manual Data 1 Buf n (n = 0 to 3)

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x08C + 0x10 × n

Bit position:	31 … 0
Bit field:	DATA[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	DATA[31:0]	Write/Read Data These bits configure the Data field in manual-command mode. For write transaction, configure the write data. For read transaction, the read data is stored.	R/W

Note: S-TYPE-3, P-TYPE-3

The CDD1BUFn register has functions to control the xSPI Master function.

44.2.2.11 LPCTL0 : OSPI Link Pattern Control Register 0

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x100

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	XD2VAL	—	—	XD2LEN[4:0]					XD1VAL	—	—	XD1LEN[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	XDPIN[1:0]		CSSEL	—	—	PATREQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PATREQ	Pattern Request This bit requests to issue the pattern. When set to 1, it starts the pattern. It is cleared to 0 when the pattern is completed. 0: No request XiP disable pattern 1: Request XiP disable pattern	R/W
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	CSSEL	Chip Select This bit selects a target memory to issue a pattern. 0: Slave 0 (CS0) 1: Slave 1 (CS1)	R/W
5:4	XDPIN[1:0]	XiP Disable Pattern Pin These bits select the data output pins to transmit XiP disable pattern. 0 0: 1 pin 0 1: 2 pins 1 0: 4 pins 1 1: 8 pins	R/W
15:6	—	These bits are read as 0. The write value should be 0.	R/W
20:16	XD1LEN[4:0]	XiP Disable Pattern 1st Phase Length These bits select the length of 1st phase in XiP disable pattern. Do not configure the pattern with zero-length for both 1st phase and 2nd phase. 0x0: 0 cycles 0x1: 1 cycle ⋮ ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
22:21	—	These bits are read as 0. The write value should be 0.	R/W
23	XD1VAL	XiP Disable Pattern 1st Phase Value This bit selects the value of 1st phase in XiP disable pattern. 0: Low drive 1: High drive	R/W
28:24	XD2LEN[4:0]	XiP Disable Pattern 2nd Phase Length These bits select the length of 2nd phase in XiP disable pattern. 0x00: 0 cycles 0x01: 1 cycle ⋮ ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
30:29	—	These bits are read as 0. The write value should be 0.	R/W
31	XD2VAL	XiP Disable Pattern 2nd Phase Value This bit selects the value of 2nd phase in XiP disable pattern. 0: Low drive 1: High drive	R/W

Note: S-TYPE-3, P-TYPE-3

The LPCTL0 register has functions to control the xSPI Master function.

44.2.2.12 LPCTL1 : OSPI Link Pattern Control Register 1

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x104

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	RSTSU[2:0]			—	RSTWID[2:0]			—	—	RSTREP[1:0]		CSSEL	—	PATREQ[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PATREQ[1:0]	Pattern Request These bits request to issue the pattern. When set to 01b or 10b, it starts the pattern. It is cleared to 00b when the pattern is completed. 0 0: No request 0 1: Request Reset pattern 1 0: Request CS-only pattern 1 1: Setting prohibited	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	CSSEL	Chip select This bit selects a target memory to issue a pattern. 0: Slave 0 (CS0) 1: Slave 1 (CS1)	R/W
5:4	RSTREP[1:0]	Reset Pattern Repeat These bits select the repeating time to toggle CS from low to high. 0 0: 4 times (specified on Reset Signaling Protocol) 0 1: 5 times 1 0: 6 times 1 1: 7 times	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
10:8	RSTWID[2:0]	Reset Pattern Width These bits configure the width of a cycle in Reset pattern and CS-only pattern. It toggles CS with the configured cycle. 0 0 0: 2 (= 2^1) cycles 0 0 1: 4 (= 2^2) cycles ⋮ ⋮ 1 1 0: 128 (= 2^7) cycles 1 1 1: 256 (= 2^8) cycles	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
14:12	RSTSU[2:0]	Reset Pattern Data Output Setup Time These bits configure the number of setup cycles for data output based on the edge of CS in a Reset pattern. The number of setup cycles requires enough setup time because the xSPI slave samples any data at the rising edge of CS. This cycle of setup time should be less than the cycle of reset pattern width (RSTWID[2:0]). 0 0 0: 1 cycle 0 0 1: 2 cycles ⋮ ⋮ 1 1 0: 7 cycles 1 1 1: 8 cycles	R/W
31:15	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The LPCTL1 register has functions to control the xSPI Master function.

44.2.2.13 LIOCTL : OSPI Link I/O Control Register

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x108

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RSTCS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WPCS1	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	—	This bit is read as 1. The write value should be 1.	R/W
1	WPCS1	WP Drive for Slave 1 This bit controls the value of the OM_WP1 port. It can be useful only for xSPI slave with write protect port. 0: Drive low level 1: Drive high level	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
16	RSTCS0	Reset Drive This bit controls the value of OM_RESET port. It can be useful only for xSPI slave with reset port. 0: Drive low level 1: Drive high level	R/W
17	—	This bit is read as 1. The write value should be 1.	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The LIOCTL register has functions to control the xSPI Master function.

44.2.2.14 CCCTL0CSn : OSPI Command Calibration Control Register 0 CSn (n = 0, 1)

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x130 + 0x020 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	CASFTEND[4:0]					—	—	—	CASFTSTA[4:0]				
Value after reset:	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	CAITV[4:0]					—	—	—	—	—	—	CANOWR	CAEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CAEN	Automatic Calibration Enable This bit enables the automatic calibration. When set to 1, it transmits the calibration sequence periodically and adjusts the value of phase shift. When set to 0 during the calibration sequence, it stops after completion of the ongoing calibration sequence, this bit is then cleared. 0: Disable automatic calibration 1: Enable automatic calibration	R/W
1	CANOWR	Calibration No Write Mode This bit selects to omit write command in the calibration sequence. It can be used for any slave device with fixed calibration pattern data. 0: Calibration sequence with write command 1: Calibration sequence without write command	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
12:8	CAITV[4:0]	Calibration Interval These bits configure the intervals between calibration patterns. 0x00: 2 (= 2^1) cycle wait 0x01: 4 (= 2^2) cycle wait ⋮ 0x1E: 2,147,483,648 (= 2^{31}) cycle wait 0x1F: 4,294,967,296 (= 2^{32}) cycle wait	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
20:16	CASFTSTA[4:0]	Calibration DS Shift Start Value These bits configure the start value of DS shift.	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
28:24	CASFTEND[4:0]	Calibration DS Shift End Value These bits configure the end value of DS shift. The end value should be equal to or more than the start value (CASFTSTA[4:0]).	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The CCCTL0CSn register has functions to control the xSPI Master function.

44.2.2.15 CCCTL1CSn : OSPI Command Calibration Control Register 1 CSn (n = 0, 1)

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x134 + 0x020 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	CARDLATE[4:0]					—	—	—	CAWRLATE[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CADATASIZE[3:0]				CAADDSIZE[2:0]			CACMDSIZE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CACMDSIZE[1:0]	Command Size These bits configure the size of the Command field. In 8D-8D-8D, set these bits to 10b. Do not configure both command size and address size to 0. 0 0: 0 bytes (no command phase) 0 1: 1 byte 1 0: 2 bytes 1 1: Setting prohibited	R/W

Bit	Symbol	Function	R/W
4:2	CAADDSIZE[2:0]	Address Size These bits configure the size of the Address field. 0 0 0: 0 bytes (no address phase) 0 0 1: 1 byte 0 1 0: 2 bytes 0 1 1: 3 bytes 1 0 0: 4 bytes Others: Setting prohibited	R/W
8:5	CADATASIZE[3:0]	Write/Read Data Size These bits configure the size of the Data field. In 8D-8D-8D, the data size should be configured with an even byte. 0x0: 1 byte 0x1: 2 bytes ⋮ ⋮ 0xE: 15 bytes 0xF: 16 bytes	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
20:16	CAWRLATE[4:0]	Write Latency Cycle These bits configure the latency cycle in the calibration frame. 0x00: No latency 0x01: 1 cycle ⋮ ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
28:24	CARDLATE[4:0]	Read Latency Cycle These bits configure the latency cycle in the calibration frame. 0x00: No latency 0x01: 1 cycle ⋮ ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The CCCTL1CSn register has functions to control the xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

44.2.2.16 CCCTL2CSn : OSPI Command Calibration Control Register 2 CSn (n = 0, 1)

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x138 + 0x020 × n

Bit position:	31–16	15–0
Bit field:	CARDCMD[15:0]	CAWRCMD[15:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	CAWRCMD[15:0]	Calibration Pattern Write Command These bits configure the calibration pattern write command.	R/W
31:16	CARDCMD[15:0]	Calibration Pattern Read Command These bits configure the calibration pattern read command.	R/W

Note: S-TYPE-3, P-TYPE-3

The CCCTL2CSn register has functions to control the xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

44.2.2.17 CCCTL3CSn : OSPI Command Calibration Control Register 3 CSn (n = 0, 1)

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x13C + 0x020 × n

Bit position:	31–0
Bit field:	CAADD[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	CAADD[31:0]	Calibration Pattern Address These bits configure the calibration pattern address.	R/W

Note: S-TYPE-3, P-TYPE-3

The CCCTL3CSn register has functions to control the xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

44.2.2.18 CCCTL4CSn : OSPI Command Calibration Control Register 4 CSn (n = 0, 1)

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x140 + 0x020 × n

Bit position:	31–0
Bit field:	CADATA[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	CADATA[31:0]	Calibration Pattern Data These bits configure the calibration pattern data.	R/W

Note: S-TYPE-3, P-TYPE-3

The CCCTL4CSn register has functions to control the xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

44.2.2.19 CCCTL5CSn : OSPI Command Calibration Control Register 5 CSn (n = 0, 1)

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x144 + 0x020 × n

Bit position:	31–0
Bit field:	CADATA[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	CADATA[31:0]	Calibration Pattern Data These bits configure the calibration pattern data.	R/W

Note: S-TYPE-3, P-TYPE-3

The CCCTL5CSn register has functions to control the xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

44.2.2.20 CCCTL6CSn : OSPI Command Calibration Control Register 6 CSn (n = 0, 1)

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x148 + 0x020 × n

Bit position:	31–0
Bit field:	CADATA[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	CADATA[31:0]	Calibration Pattern Data These bits configure the calibration pattern data.	R/W

Note: S-TYPE-3, P-TYPE-3

The CCCTL6CSn register has functions to control the xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

44.2.2.21 CCCTL7CSn : OSPI Command Calibration Control Register 7 CSn (n = 0, 1)

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x14C + 0x020 × n

Bit position:	31–0
Bit field:	CADATA[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	CADATA[31:0]	Calibration Pattern Data These bits configure the calibration pattern data.	R/W

Note: S-TYPE-3, P-TYPE-3

The CCCTL7CSn register has functions to control the xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

44.2.2.22 RACTL : OSPI Register Access Control Register

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x208

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RAEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RAEN	Register Access Enable 0: Disable MER register access 1: Enable MER register access	R/W

Bit	Symbol	Function	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

44.2.2.23 MER : OSPI Map End Replace Register

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x228

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	MECS1								—	—	—	MERCS1
Value after reset:	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
11:4	—	These bits are read as 1. The write value should be 1.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
16	MERCS1	Map End Replace CS1 Enable 0: No replacement 1: Replacement	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
27:20	MECS1	Map End CS1 Replace Address The bit27 to bit20 of system bus address is replaced with this value when MERCS1 is set to 1.	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

For SiP product, the MER register should be set as described in section 44.4. SiP Product Configuration. For standard product, do not change the value after reset.

44.2.3 OSPI Status Registers

These registers monitor the status of xSPI Master.

44.2.3.1 COMSTT : OSPI Common Status Register

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x184

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	RSTOCS1	INTCS1	ECSCS1	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	WRBUFNECH0	—	PBUFNECH0	—	—	—	MEMACCCH0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MEMACCCH0	Memory Access Ongoing from Channel 0 This bit is only valid in the section 44.3.7.2. Flow of Communication Stop and section 44.3.7.6. Flow of Memory-mapping Stop flows. 0: System bus bridge CH0 is not accessing to memory 1: System bus bridge CH0 is accessing to memory	R
3:1	—	These bits are read as 0.	R
4	PBUFNECH0	Prefetch Buffer Not Empty for Channel 0 0: Empty 1: Not empty	R
5	—	This bit is read as 0.	R
6	WRBUFNECH0	Write Buffer Not Empty for Channel 0 This bit is only valid in the section 44.3.7.2. Flow of Communication Stop and section 44.3.7.6. Flow of Memory-mapping Stop flows. 0: Empty 1: Not empty	R
19:7	—	These bits are read as 0.	R
20	ECSCS1	ECS Monitor for Slave 1 This bit indicates the value of OM_ECSINT1 port. 0: Low level 1: High level	R
21	INTCS1	INT Monitor for Slave 1 This bit indicates the value of OM_ECSINT1 port. 0: Low level 1: High level	R
22	RSTOCS1	RSTO Monitor for Slave 1 This bit indicates the value of OM_RSTO1 port. 0: Low level 1: High level	R
31:23	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The COMSTT register indicates the status of the xSPI Master.

44.2.3.2 CASTTCSn : OSPI Calibration Status Register CSn (n = 0, 1)

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x188 + 0x004 × n

Bit position:	31 … 0
Bit field:	CASUC[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	CASUC[31:0]	Calibration Success These bits indicate the calibration success for each DS shift value. This bit is updated for each completed calibration sequence. CASUC[x] indicates calibration success in DS shift value = x.	R

Note: S-TYPE-3, P-TYPE-3

The CASTTCSn register indicates the status of xSPI Master.

44.2.4 OSPI Interrupt Registers

These registers control the interrupt function of xSPI Master.

44.2.4.1 INTS : OSPI Interrupt Status Register

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x190

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CASUCCS1	CASUCCS0	CAFAILCS1	CAFAILCS0	—	—	—	—	—	—	—	BUSERRCH0	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	INTCS1	—	—	—	ECSCS1	—	—	—	DSTOCS1	DSTOCS0	PERTO	—	PATCMP	CMDCMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMDCMP	Command Completed This bit is set to 1 when the requested manual-command is completed. In direct manual-command, it means all transactions completed. In periodic manual-command, it means the read data matches with the expected data. 0: No detection 1: Detection	R
1	PATCMP	Pattern Completed This bit is set to 1 when the requested pattern is completed. 0: No detection 1: Detection	R
2	—	This bit is read as 0.	R
3	PERTO	Periodic Transaction Timeout This bit is set to 1 when the read value does not match with the expected value in periodic manual-command mode. 0: No detection 1: Detection	R
4	DSTOCS0	DS Timeout for Slave 0 This bit is set to 1 for lost DS in read transaction when using DS. It means not receiving the data during the expected read phase. In this case, xSPI master stops the read transaction and the following transaction. This error may issue an error response to the system bus. 0: No detection 1: Detection	R
5	DSTOCS1	DS Timeout for Slave 1 This function is the same as DSTOCS0. 0: No detection 1: Detection	R
8:6	—	These bits are read as 0.	R
9	ECSCS1	ECC Error Detection for Slave 1 This bit is set to 1 when the falling edge is detected on the OM_ECSINT1 port. It can be useful only for xSPI slave with ECC detection function. 0: No detection 1: Detection	R
12:10	—	These bits are read as 0.	R
13	INTCS1	Interrupt Detection for Slave 1 This bit is set to 1 when the falling edge is detected on the OM_ECSINT1 port. It can be useful only for xSPI slave with interrupt function. 0: No detection 1: Detection	R
19:14	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
20	BUSERRCH0	System Bus Error for Channel 0 This bit is set to 1 when an error response occurs on system bus CH0.	R
27:21	—	These bits are read as 0.	R
28	CAFAILCS0	Calibration Failed for Slave 0 This bit is set to 1 when calibration failed. 0: No detection 1: Detection	R
29	CAFAILCS1	Calibration Failed for Slave 1 This function is the same as CAFAILCS0. 0: No detection 1: Detection	R
30	CASUCCS0	Calibration Success for Slave 0 This bit is set to 1 for success calibration. 0: No detection 1: Detection	R
31	CASUCCS1	Calibration Success for Slave 1 This function is the same as CASUCCS0. 0: No detection 1: Detection	R

Note: S-TYPE-3, P-TYPE-3

The INTS register indicates the status of an interrupt. The bits in this register are cleared to 0 when writing 1 on the corresponding bit of the INTC register.

44.2.4.2 INTC : OSPI Interrupt Clear Register

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x194

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CASUCCS1C	CASUCCS0C	CAFAILCS1C	CAFAILCS0C	—	—	—	—	—	—	—	BUSERRCH0C	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	INTCS1C	—	—	—	ECSCS1C	—	—	—	DSTOCS1C	DSTOCS0C	PERTOC	—	PATCMPC	CMDCMPC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMDCMPC	Command Completed Interrupt Clear 0: No change interrupt status 1: Clear interrupt status	W
1	PATCMPC	Pattern Completed Interrupt Clear 0: No change interrupt status 1: Clear interrupt status	W
2	—	The write value should be 0.	W
3	PERTOC	Periodic Transaction Timeout Interrupt Clear 0: No change interrupt status 1: Clear interrupt status	W
4	DSTOCS0C	DS Timeout for Slave 0 Interrupt Clear 0: No change interrupt status 1: Clear interrupt status	W

Bit	Symbol	Function	R/W
5	DSTOCS1C	DS Timeout for Slave 1 Interrupt Clear 0: No change interrupt status 1: Clear interrupt status	W
8:6	—	The write value should be 0.	W
9	ECSCS1C	ECC Error Detection for Slave 1 Interrupt Clear 0: No change interrupt status 1: Clear interrupt status	W
12:10	—	The write value should be 0.	W
13	INTCS1C	Interrupt Detection for Slave 1 Interrupt Clear 0: No change interrupt status 1: Clear interrupt status	W
19:14	—	The write value should be 0.	W
20	BUSERRCH0C	System Bus Error for ch0 Interrupt Clear 0: No change interrupt status 1: Clear interrupt status	W
27:21	—	The write value should be 0.	W
28	CAFAILCS0C	Calibration Failed for Slave 0 Interrupt Clear 0: No change interrupt status 1: Clear interrupt status	W
29	CAFAILCS1C	Calibration Failed for Slave 1 Interrupt Clear 0: No change interrupt status 1: Clear interrupt status	W
30	CASUCCS0C	Calibration Success for Slave 0 Interrupt Clear 0: No change interrupt status 1: Clear interrupt status	W
31	CASUCCS1C	Calibration Success for Slave 1 Interrupt Clear 0: No change interrupt status 1: Clear interrupt status	W

Note: S-TYPE-3, P-TYPE-3

The INTC register clears the status of interrupt.

44.2.4.3 INTE : OSPI Interrupt Enable Register

Base address: OSPIn_B = 0x4026_8000 + 0x0400 × n (n = 0, 1)
OSPIn_B_NS = 0x5026_8000 + 0x0400 × n (n = 0, 1)

Offset address: 0x198

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CASUCCS1E	CASUCCS0E	CAFAILCS1E	CAFAILCS0E	—	—	—	—	—	—	—	BUSERRCH0E	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	INTCS1E	—	—	—	ECSCS1E	—	—	—	DSTOCS1E	DSTOCS0E	PERTOE	—	PATCMPE	CMDCMPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMDCMPE	Command Completed Interrupt Enable 0: Disabled 1: Enabled	R/W
1	PATCMPE	Pattern Completed Interrupt Enable 0: Disabled 1: Enabled	R/W

Bit	Symbol	Function	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	PERTOE	Periodic Transaction Timeout Interrupt Enable 0: Disabled 1: Enabled	R/W
4	DSTOCS0E	DS Timeout for Slave 0 Interrupt Enable 0: Disabled 1: Enabled	R/W
5	DSTOCS1E	DS Timeout for Slave 1 Interrupt Enable 0: Disabled 1: Enabled	R/W
8:6	—	These bits are read as 0. The write value should be 0.	R/W
9	ECSCS1E	ECC Error Detection for Slave 1 Interrupt Enable 0: Disabled 1: Enabled	R/W
12:10	—	These bits are read as 0. The write value should be 0.	R/W
13	INTCS1E	Interrupt Detection for Slave 1 Interrupt Enable 0: Disabled 1: Enabled	R/W
19:14	—	These bits are read as 0. The write value should be 0.	R/W
20	BUSERRCH0E	System Bus Error for ch0 Interrupt Enable 0: Disabled 1: Enabled	R/W
27:21	—	These bits are read as 0. The write value should be 0.	R/W
28	CAFAILCS0E	Calibration Failed for Slave 0 Interrupt Enable 0: Disabled 1: Enabled	R/W
29	CAFAILCS1E	Calibration Failed for Slave 1 Interrupt Enable 0: Disabled 1: Enabled	R/W
30	CASUCCS0E	Calibration Success for Slave 0 Interrupt Enable 0: Disabled 1: Enabled	R/W
31	CASUCCS1E	Calibration Success for Slave 1 Interrupt Enable 0: Disabled 1: Enabled	R/W

Note: S-TYPE-3, P-TYPE-3

The INTE register enables the interrupt.

44.3 Operation

xSPI Master interface has the functions to issue the transaction for external memory with xSPI Slave interface. It allows writes to registers in external memory or reads from it.

This xSPI Master has two modes to issue transaction.

- Manual-command mode
- Memory-mapping mode

In manual-command mode, software configures all fields of xSPI frame and starts the transaction by a software request. In memory-mapping mode, software automatically converts a system bus for a preconfigured memory area into an xSPI transaction. It enables access from the system bus to the external memory area outside of the chip through xSPI bus.

This section describes the xSPI bus operation, the direct control of xSPI frame (manual-command), the control of memory access (memory-mapping), the error operation, and the flow to operation.

44.3.1 xSPI Bus

This section describes the xSPI bus operation.

Figure 44.2 shows an example of connections between OSPI and memory devices.

In this case, connect OM_CS0 to the RAM device and connect OM_CS1 to the Flash device.

Specify pull-up resistors according to the instructions for each device.

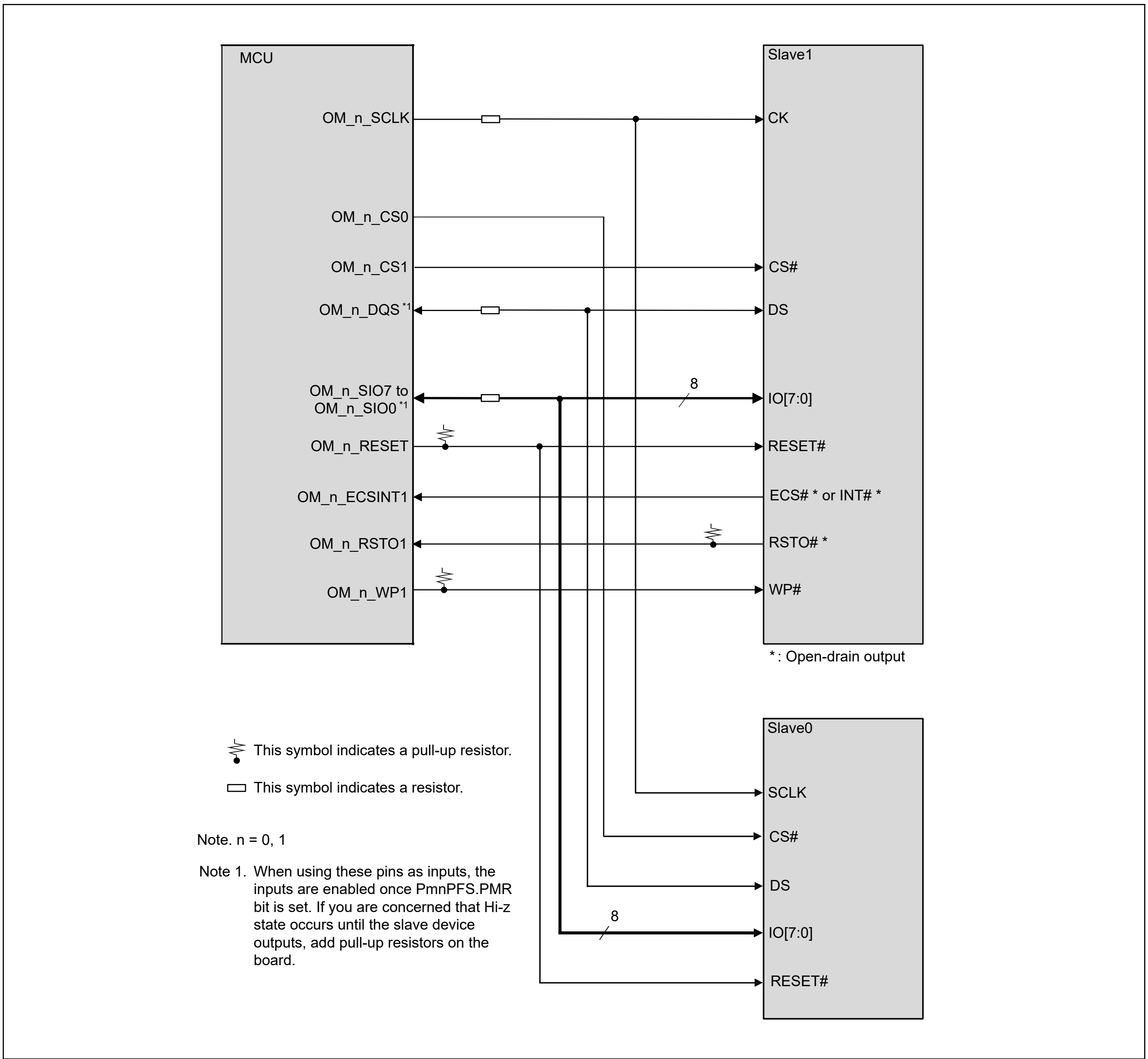


Figure 44.2 Example of connection between OSPI and memory devices

44.3.1.1 Supported Protocol Mode

This xSPI Master supports various protocol modes. It is configured by Protocol mode bits (LIOCFGCSn.PRTMD[9:0]). Table 44.3 shows the summary of protocol modes.

Table 44.3 Supported protocol mode (1 of 2)

Protocol mode	Function	PRTMD[9:0]	Note
1S-1S-1S	Command, Address and Data fields are transferred at SDR using 1 data input pin and 1 data output pin. Read data is sampled with CK.	0x000	Specified by xSPI protocol

Table 44.3 Supported protocol mode (2 of 2)

Protocol mode	Function	PRTMD[9:0]	Note
4S-4D-4D	Command field is transferred at SDR using 4 data pins. Address and Data fields are transferred at DDR using 4 data pins. Read data is sampled with OM_DQS.	0x3B2	Specified by xSPI protocol
8D-8D-8D	Command, Address and Data fields are transferred at DDR using 8 data pins. Read data is sampled with OM_DQS.	0x3FF	Specified by xSPI protocol
1S-2S-2S	Command field is transferred at SDR using 1 data pin. Address and Data fields are transferred at SDR using 2 data pins. Read data is sampled with CK.	0x048	—
2S-2S-2S	Command, Address and Data fields are transferred at SDR using 2 data pins. Read data is sampled with CK.	0x049	—
1S-4S-4S	Command field is transferred at SDR using 1 data pin. Address and Data fields are transferred at SDR using 4 data pins. Read data is sampled with CK.	0x090	—
4S-4S-4S	Command, Address, and Data fields are transferred at SDR using 4 data pins. Read data is sampled with CK.	0x092	—

Note: In XiP mode enable, XiP code is inserted in the Latency field. It is valid only for memory-mapping mode.

The bytes of Command and Address fields are transferred in the highest order to the lowest order sequence. The sequential bytes of the Data field are transferred in the lowest address to the highest address order. When using multiple pins, the least significant bit of each byte is placed on OM_SIO0 with each higher-order bit on the successively higher numbered OM_SIOn (n = 1 to 7) signals.

Figure 44.3 shows the timing diagram for 1S-1S-1S protocol mode. The OM_SIO0 signal is used for output data and the OM_SIO1 signal is used for input data.

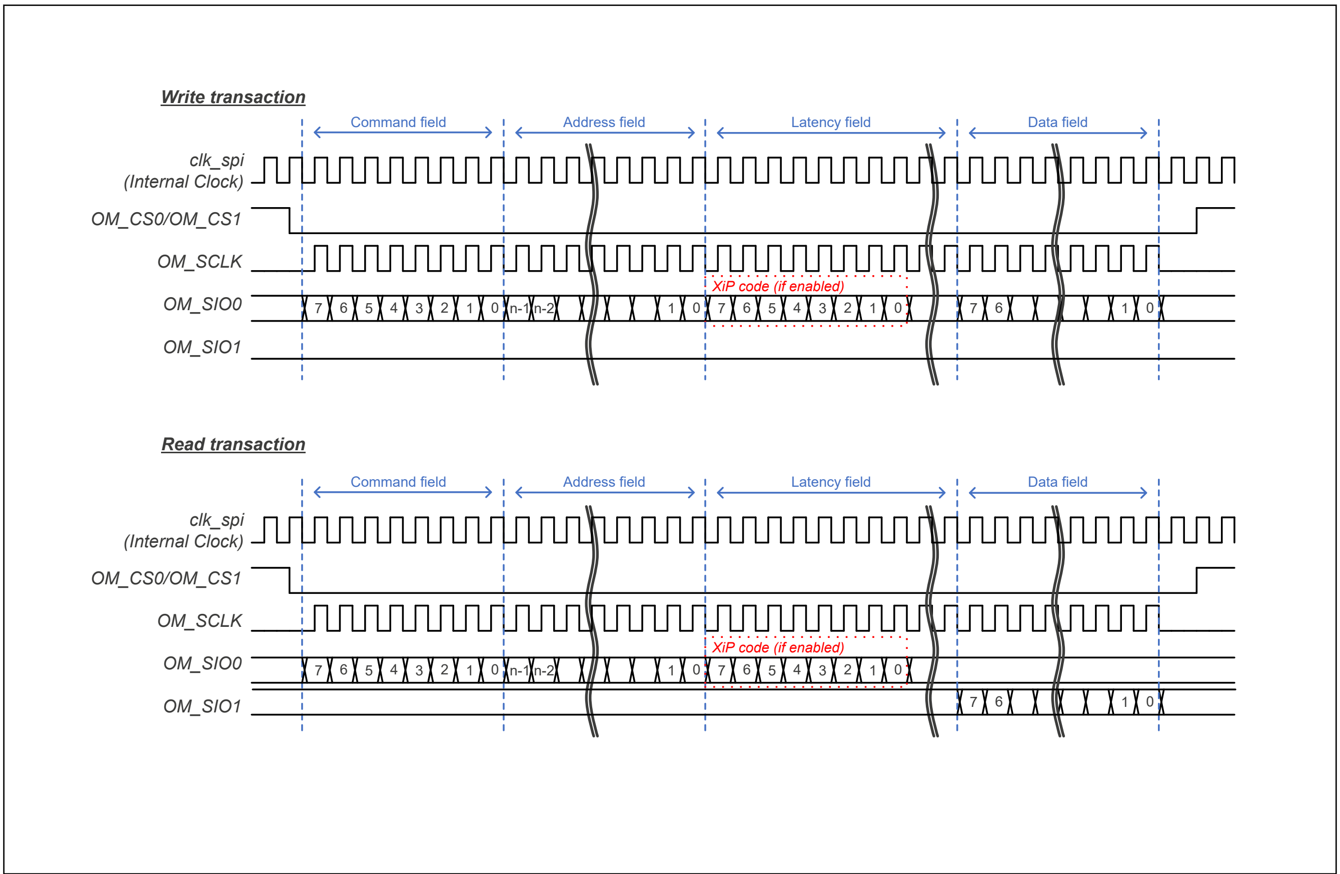


Figure 44.3 1S-1S-1S timing diagram

Figure 44.4 shows timing diagram for 1S-2S-2S protocol mode.

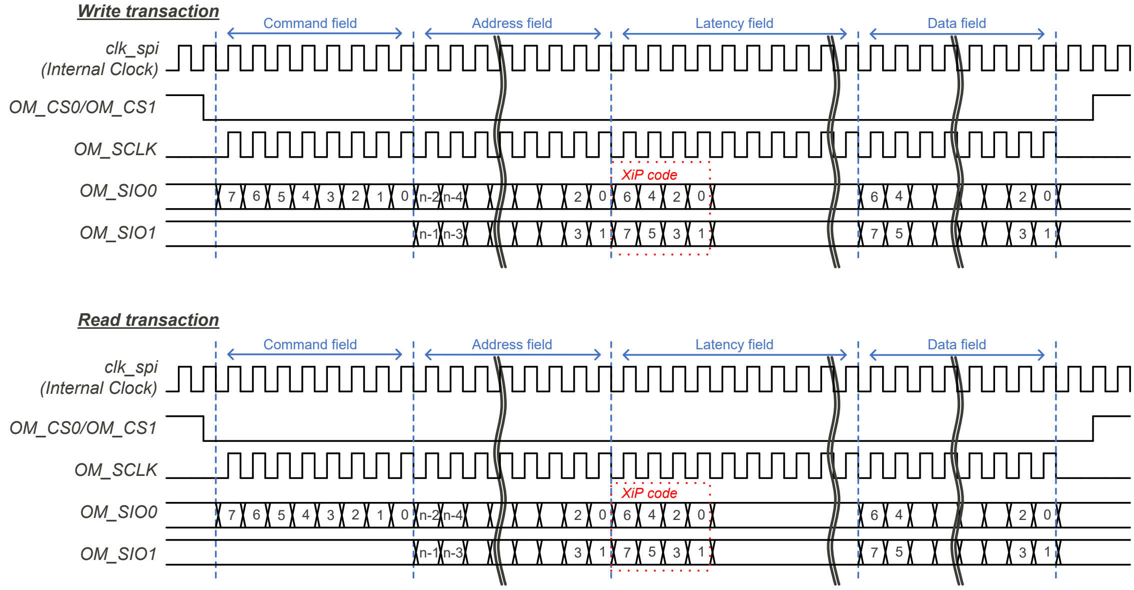


Figure 44.4 1S-2S-2S timing diagram

Figure 44.5 shows timing diagram for 4S-4D-4D protocol mode.

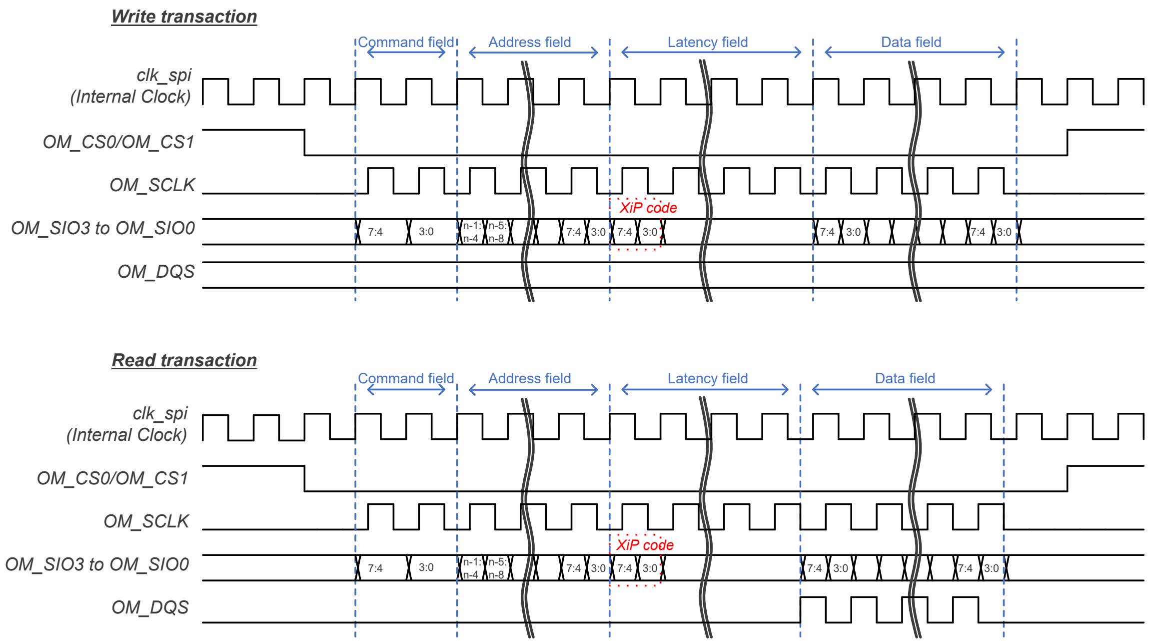


Figure 44.5 4S-4D-4D timing diagram

Figure 44.6 shows timing diagram for 8D-8D-8D profile 1.0 protocol mode. In 8D-8D-8D, the data is always transmitted with byte-pair. When each field is an odd byte, the last one byte is padded with invalid data.

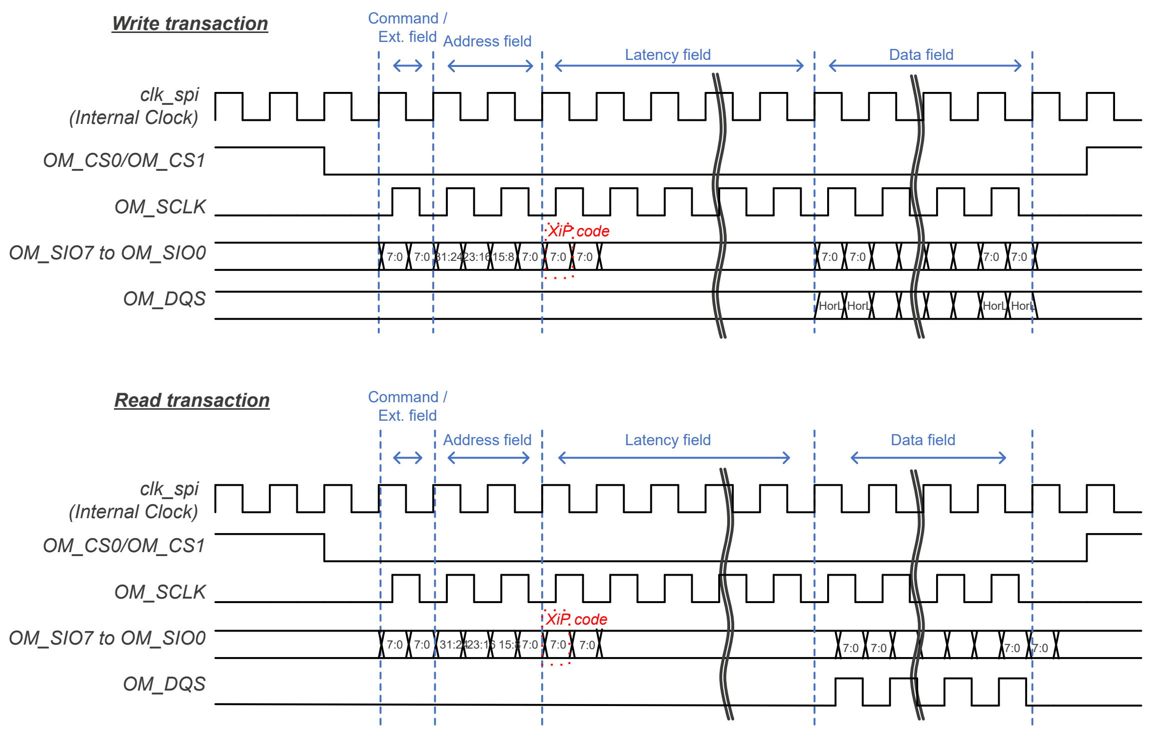


Figure 44.6 8D-8D-8D profile 1.0 timing diagram

Figure 44.7 shows timing diagram for 8D-8D-8D profile 2.0 protocol mode. In 8D-8D-8D, the data is always transmitted with byte-pair. When each field is an odd byte, the last one byte is padded with invalid data.

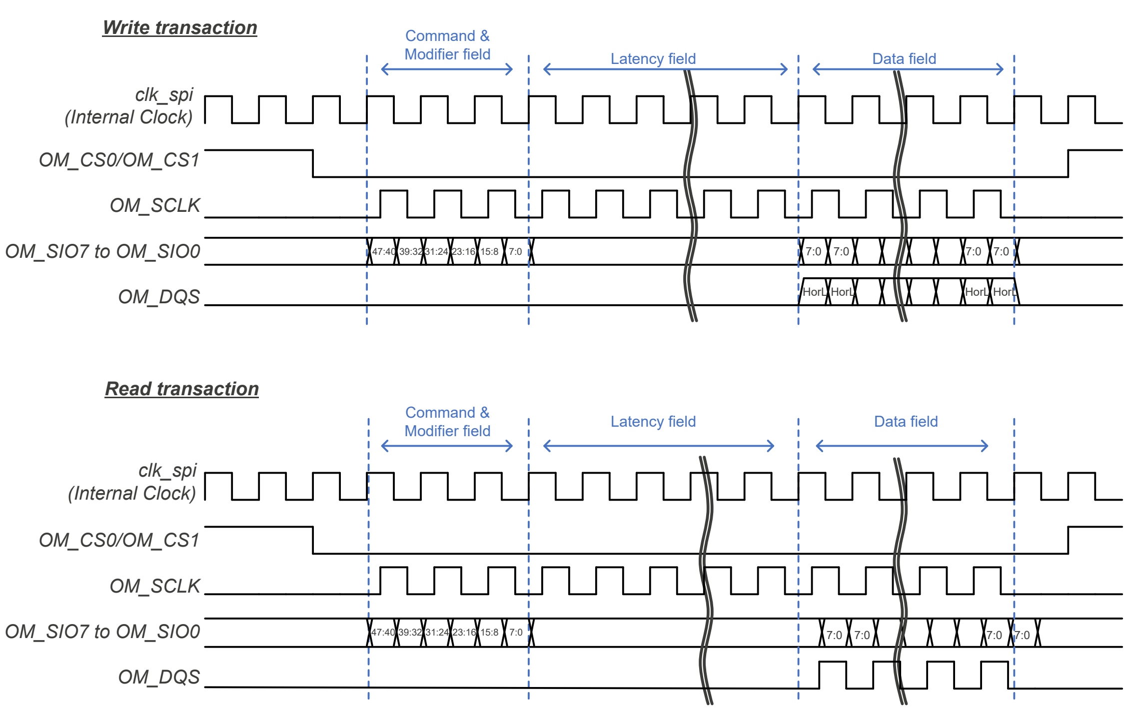


Figure 44.7 8D-8D-8D profile 2.0 timing diagram

44.3.1.2 xSPI Frame Interval

The interval between xSPI frames is configured with CS minimum idle term bits (LIOCFGCSn.CSMIN[3:0]). It depends on the specification of xSPI slave device.

44.3.1.3 OSPI Signals Timing Control

This xSPI Master supports both SDR and DDR. It is possible to sample input data with Data-Strobe (DS) signal at SDR. For various modes and easy implementation, this xSPI Master can adjust the timing to drive/sample xSPI interface signals statically. Table 44.4 shows the summary of xSPI interface signal timing control.

Table 44.4 Summary of OSPI signals timing control (1 of 2)

Signal	Mode	Default operation	Timing control (n = 0, 1)
OM_CS0, OM_CS1 drive	—	Asserting 1 cycle before the rising edge of first OM_SCLK	1 cycle extension for asserting with LIOCFGCSn.CSASTEX bit
		Negating 1.5 cycle after the falling edge of last OM_SCLK	1 cycle extension for negating with LIOCFGCSn.CSNEGEX bit
OM_SCLK drive	SDR without DS SDR with DS	Reference point	—
	DDR with DS	Reference point	—

Table 44.4 Summary of OSPI signals timing control (2 of 2)

Signal	Mode	Default operation	Timing control (n = 0, 1)
OM_SIOn (n = 0 to 7) output drive	SDR without DS SDR with DS	Falling edge of clk_spi (Internal Clock)	0 or 0.5 cycle shift with LIOCFGCSn.SDRDRV bit
	DDR with DS	Both edges of clk_spi (internal clock)	—
OM_SIOn (n = 0 to 7) input sample	SDR without DS	Falling edge of OM_SCLK on expected data size	0 to 7 cycle shift (1 cycle unit) with LIOCFGCSn.SDRSMPSFT[3:0] bits 0 or 0.5 cycle shift with LIOCFGCSn.SDRSMPMD bit
	SDR with DS	Falling edge of OM_DQS signal on expected data size	Sample at rising edge with LIOCFGCSn.SDRSMPMD bit 0 to 1 cycle phase shift with WRAPCFG.DSSFTCSn[4:0] bits 0 to 7 cycle extension with LIOCFGCSn.DDRSMPEX[3:0] bits
	DDR with DS	Both edges of OM_DQS signal on expected data size	0 to 1 cycle phase shift with WRAPCFG.DSSFTCSn[4:0] bits 0 to 7 cycle extension with LIOCFGCSn.DDRSMPEX[3:0] bits

Note: In DDR on xSPI protocol, DS should be aligned for the center of data. It means to shift the phase by 0.25 cycle (90 degrees). This xSPI master supports the adjustment of this phase depending on the usage conditions.

Figure 44.8 shows the default operation and timing control for SDR without DS.

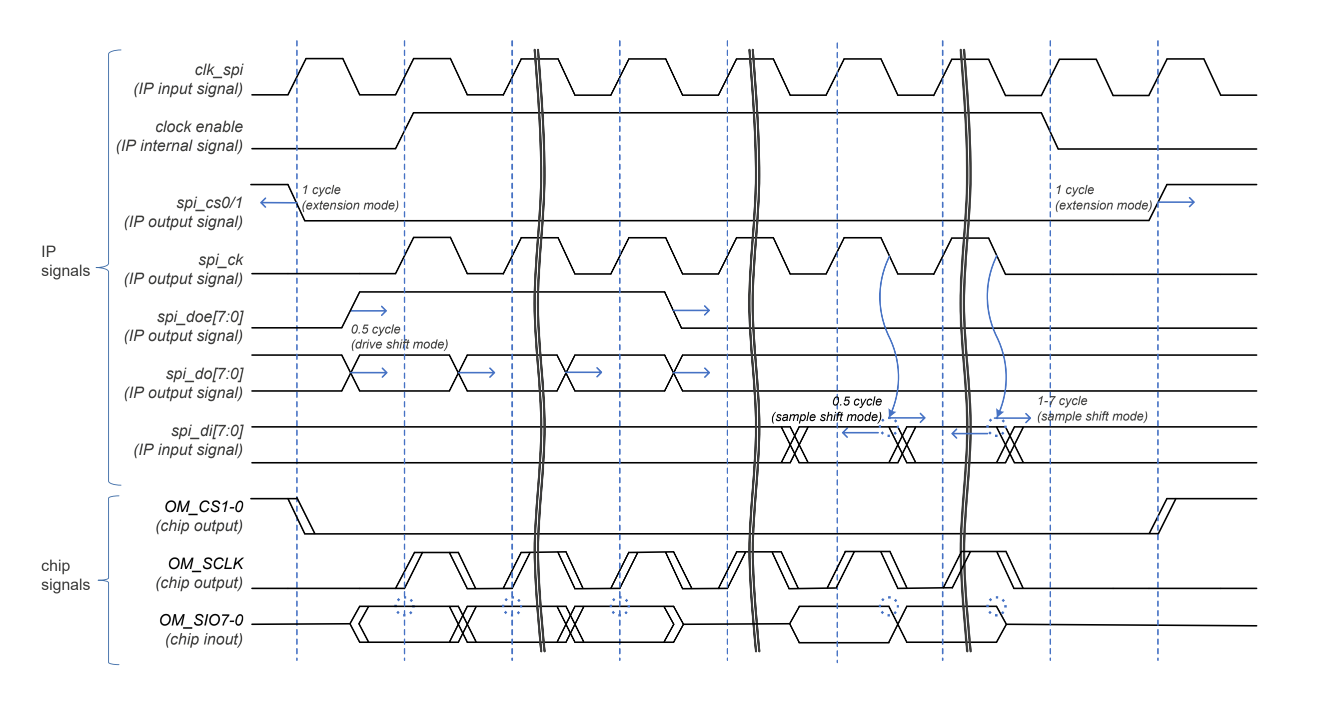


Figure 44.8 Timing control for SDR without DS

Figure 44.9 shows the default operation and timing control for SDR with DS.

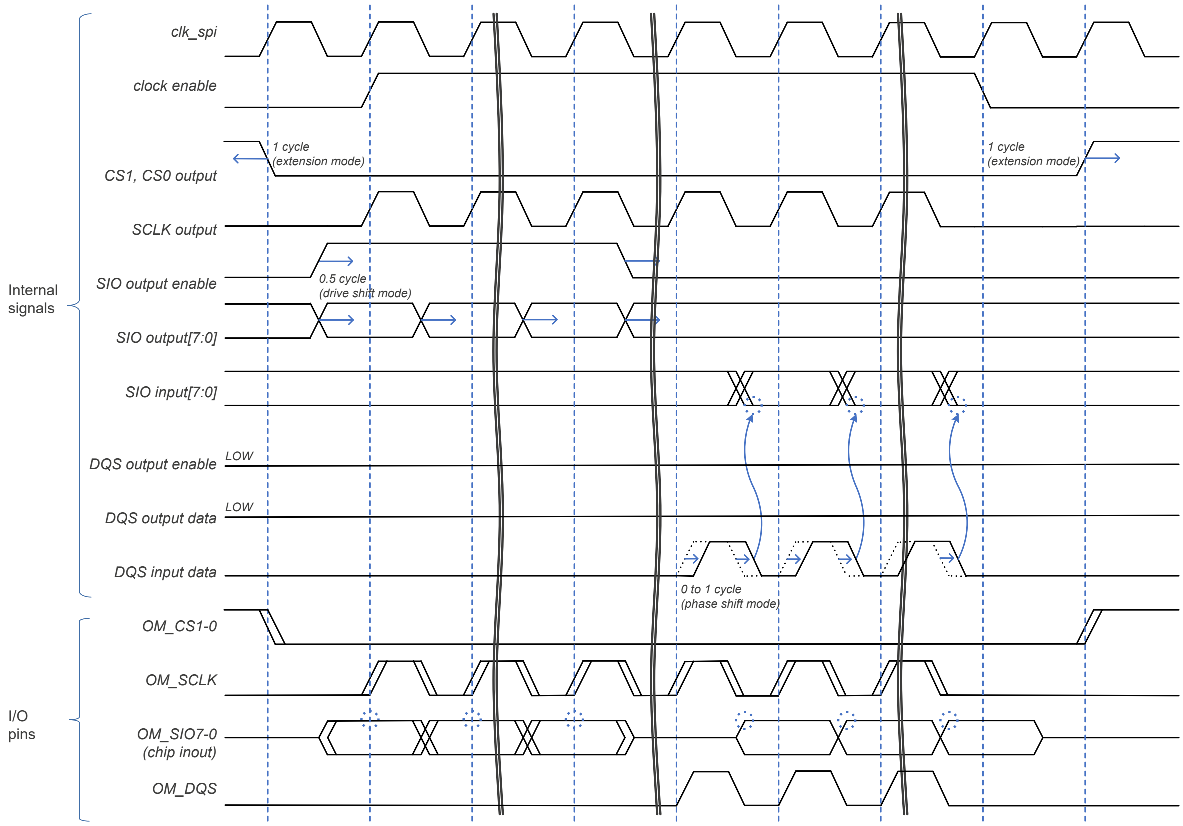


Figure 44.9 Timing control for SDR with DS

Figure 44.10 shows the default operation and timing control for DDR with DS.

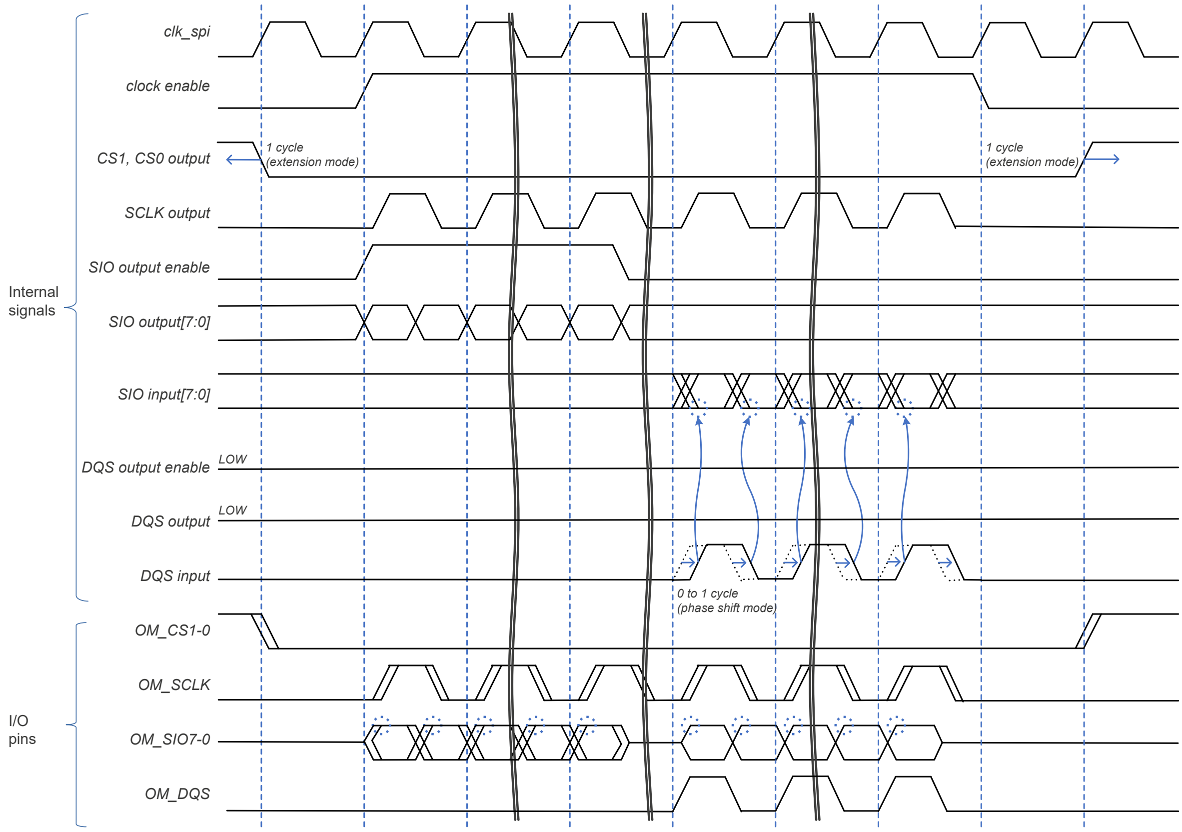


Figure 44.10 Timing control for DDR with DS

44.3.1.4 Automatic Calibration

The xSPI master supports the function to adjust DS shift value (WRAPCFG.DSSFTCSn (n = 0, 1)) automatically. When this function is enabled (CCCTL0CSn.CAEN = 1), this xSPI master transmits the calibration sequence periodically and adjusts the value of phase shift.

When all read compare is mismatched in read transaction during the calibration sequence, the calibration fail bit (INTS.CAFAILCSn = 1 (n = 0, 1)) is asserted and DS shift value is not updated. When at least one read compare is matched, the calibration success bit (INTS.CASUCCSn = 1 (n = 0, 1)) is asserted and DS shift value is updated. The result of each DS shift value can be monitored by the Calibration Status register (CASTTCSn). Figure 44.11 shows automatic calibration.

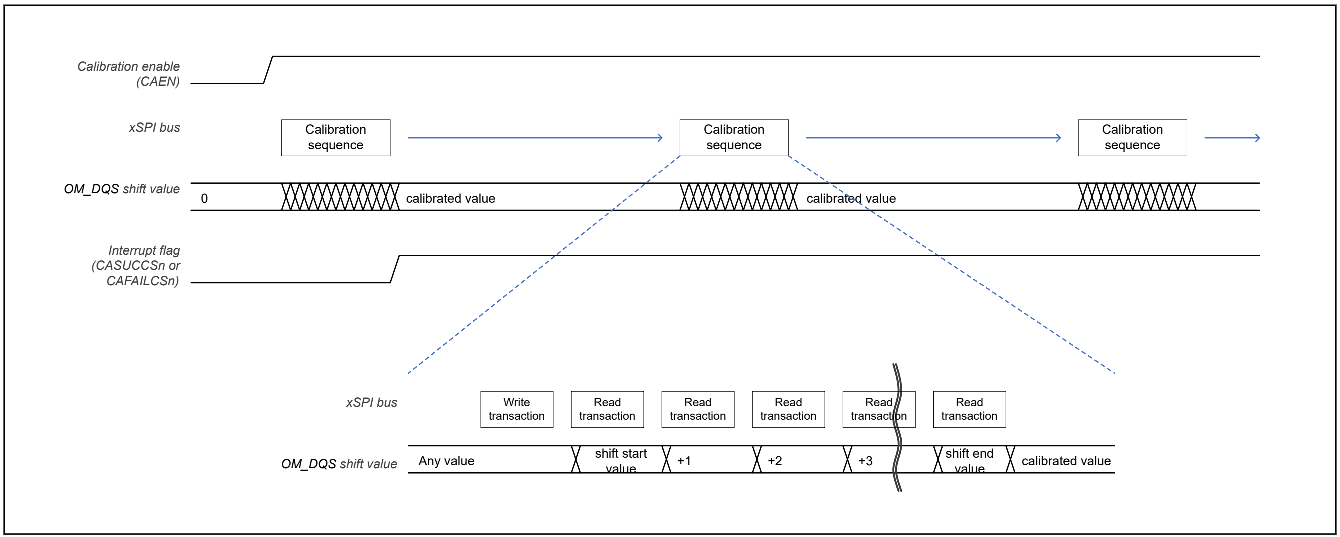


Figure 44.11 Automatic calibration

44.3.2 Manual-command

This section describes the manual-command mode. The manual-command has two functional modes:

- Direct mode
- Periodic mode.

44.3.2.1 Direct Mode

This mode sequentially can issue up to four xSPI transactions configured and requested by software. A series of transaction can be issued by a transaction request (CDCTL0.TRREQ = 1 with PERMD = 0). The number of transactions (CDCTL0.TRNUM[1:0]) can be configured up to 4. It can be used to change the mode or read the status of an xSPI slave device.

Table 44.5 shows the configured register bits for direct manual-command. Figure 44.25 shows the operating flow.

Table 44.5 Manual-command configuration for direct mode

Transaction	Transaction type	Command	Command size	Address	Address size	Data (x = 0, 1)	Data size	Latency cycle
1st Transaction	CDTBUF0. TRTYPE	CDTBUF0. CMD[15:0]	CDTBUF0. CMDSIZE [1:0]	CDABUF0. ADD[31:0]	CDTBUF0. ADDSIZE [2:0]	CDDxBUF0. DATA[31:0]	CDTBUF0. DATASIZE [3:0]	CDTBUF0. LATE[4:0]
2nd Transaction	CDTBUF1. TRTYPE	CDTBUF1. CMD[15:0]	CDTBUF1. CMDSIZE [1:0]	CDABUF1. ADD[31:0]	CDTBUF1. ADDSIZE [2:0]	CDDxBUF1. DATA[31:0]	CDTBUF1. DATASIZE [3:0]	CDTBUF1. LATE[4:0]
3rd Transaction	CDTBUF2. TRTYPE	CDTBUF2. CMD[15:0]	CDTBUF2. CMDSIZE [1:0]	CDABUF2. ADD[31:0]	CDTBUF2. ADDSIZE [2:0]	CDDxBUF2. DATA[31:0]	CDTBUF2. DATASIZE [3:0]	CDTBUF2. LATE[4:0]
4th Transaction	CDTBUF3. TRTYPE	CDTBUF3. CMD[15:0]	CDTBUF3. CMDSIZE [1:0]	CDABUF3. ADD[31:0]	CDTBUF3. ADDSIZE [2:0]	CDDxBUF3. DATA[31:0]	CDTBUF3. DATASIZE [3:0]	CDTBUF3. LATE[4:0]

44.3.2.2 Periodic Mode

This mode periodically issues an xSPI read transaction configured and requested by software, and can compare the read value up to 4 bytes with the expected value. The transaction is issued by a transaction request (CDCTL0.TRREQ = 1 with PERMD = 1). It can be used to alternate the status polling operation of an xSPI slave device.

The periodic term is configured in the periodic transaction interval bits (CDCTL0.PERITV[4:0]). The expected value is configured in the periodic transaction expected and masked value bits (CDCTL1.PEREXP[31:0] and CDCTL2.PERMSK[31:0]). Table 44.6 shows the configured register bits for periodic manual-command. Figure 44.26 shows the operating flow.

Table 44.6 Manual-command configuration for periodic mode

Transaction	Transaction type	Command	Command size	Address	Address size	Data (x = 0, 1)	Data size	Latency cycle
Read Transaction	CDTBUF0. TRTYPE	CDTBUF0. CMD[15:0]	CDTBUF0. CMDSIZE [1:0]	CDABUF0. ADD[31:0]	CDTBUF0. ADDSIZE [2:0]	CDDxBUF0. DATA[31:0]	CDTBUF0. DATASIZE [3:0]	CDTBUF0. LATE[4:0]

44.3.3 Memory-mapping

This section describes the memory-mapping mode. This mode automatically converts system bus access for a preconfigured memory area into an xSPI transaction.

44.3.3.1 Configuration

In this operation, the payload of the Address and Data fields are delivered from system bus signals. The information of the Command field and size are delivered from the configured register bits. Table 44.7 shows the register bits configured for memory-mapping.

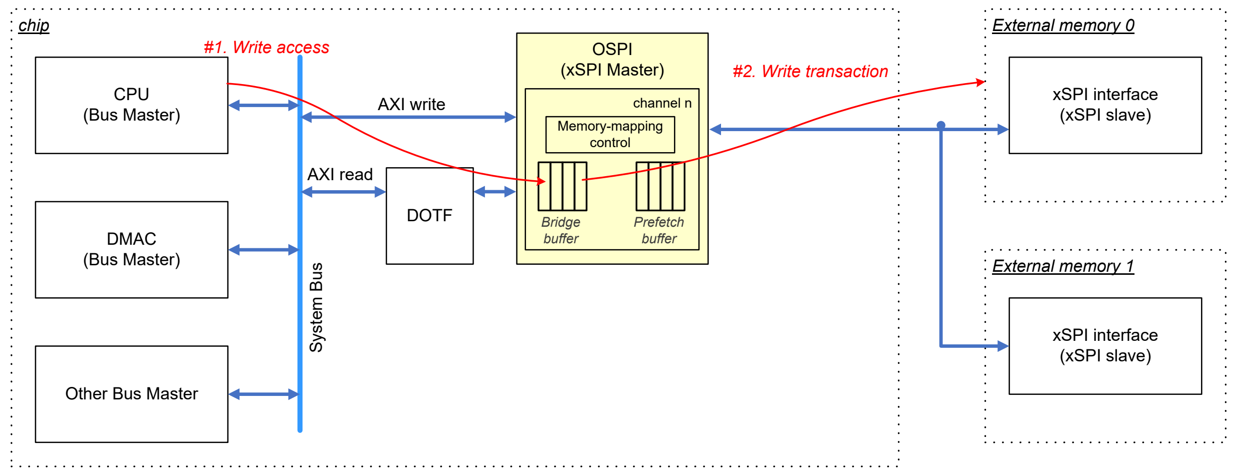
Table 44.7 Memory-mapping configuration for memory area access (n = 0, 1)

System bus transaction	Format change mode	Command	Command size	Address	Address size	Data	Data size	Latency cycle
Write for slave n memory area	Normal	CMCFG2CSn. WRCMD[15:8]	1 byte	SAWADDR[x:0]	CMCFG0CSn. ADDSIZE [1:0]	SWDATA	Up to SAWLEN and SAWSIZE	CMCFG2CSn. WRLATE[4:0]
	8D-8D-8D profile 1.0	CMCFG2CSn. WRCMD[15:0]	2 bytes					
	8D-8D-8D profile 2.0 Command Modifier	CMCFG2CSn. WRCMD[15:8]	1 byte	{SAWADDR[27:4], 0000000000000b, SAWADDR[3:1]}	5 bytes			
	8D-8D-8D profile 2.0 Extended Command Modifier	CMCFG2CSn. WRCMD[15:13]	3 bits	{0b, SAWADDR [31:4], 0000000000000b, SAWADDR[3:1]	45 bits			
Read for slave n memory area	Normal	CMCFG1CSn. RDCMD[15:8]	1 byte	SARADDR[x:0]	CMCFG0CSn. ADDSIZE [1:0]	SRDATA	Up to SARLEN and SARSIZE	CMCFG1CSn. RDLATE[4:0]
	8D-8D-8D profile 1.0	CMCFG1CSn. RDCMD[15:0]	2 bytes					
	8D-8D-8D profile 2.0 Command Modifier	CMCFG1CSn. RDCMD[15:8]	1 byte	{SARADDR[27:4], 0000000000000b, SARADDR[3:1]}	5 bytes			
	8D-8D-8D profile 2.0 Extended Command Modifier	CMCFG1CSn. RDCMD[15:13]	3 bits	{0b, SARADDR [31:4], 0000000000000b, SARADDR[3:1]}	45 bits			

Note: The MSByte of Address can be replaced with Address Replace Enable and Code bits (CMCFG0CSn.ADDRPEN[7:0]/ ADDRPCD[7:0]).

44.3.3.2 Write Access Operation

At accepting write access for memory area from system bus, this xSPI Master stores all payload data in an internal bridge buffer and then issues a write transaction into an xSPI slave. Figure 44.12 shows the operation summary.



Note: n = 0

Figure 44.12 Write access for memory area

The operation of xSPI bus changes depending on the burst type of the system bus. When the burst type is single type or incremental type, one system bus transaction triggers one xSPI frame. When the burst type is wrap type and the CMCFG0CSn.WPBSTMD is 0, one system bus transaction triggers two xSPI frames. Figure 44.13 shows the relationship between AXI and xSPI frames.

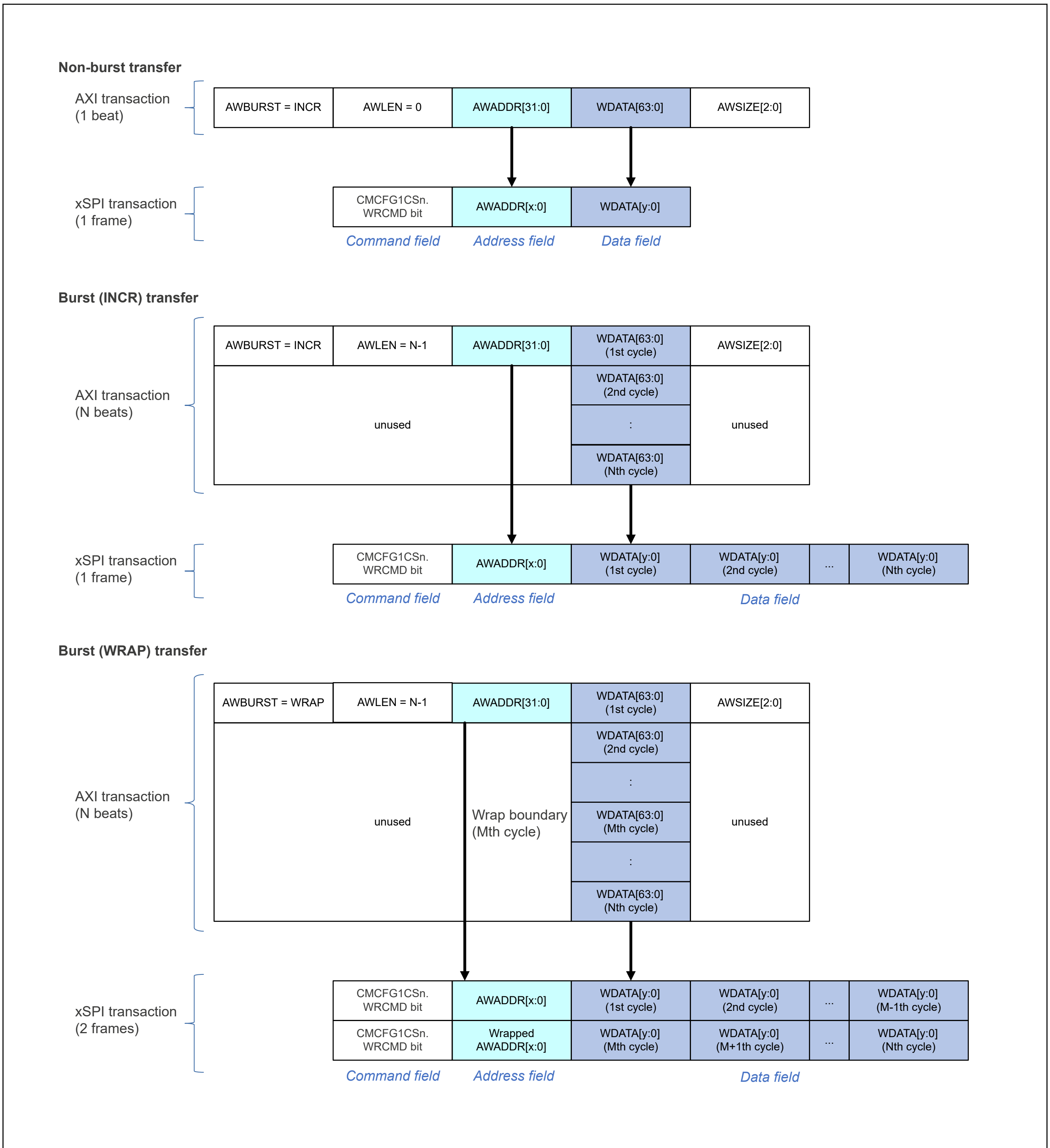


Figure 44.13 xSPI frame format in write access (Normal format)

44.3.3.3 Combination Function

At the system bus write access for memory area, this xSPI master has the function to combine the write data for high throughput on the xSPI bus. When this function is enabled (BMCFGCHn.MWRCOMB = 1), this xSPI master transmits a xSPI frame with the selected size while the sequential address is incremental*1. When one of the below conditions is detected, even though not reaching to the target size (BMCFGCHn.MWRSIZE[7:0]), this xSPI master transmits the pending data into the xSPI bus.

- Nonincremental address is detected
- Different burst type is detected

- Read transaction is detected
- Access for different slave is detected
- Memory Write Data Push bit (BMCTL1.MWRPUSHCHn (n = 0)) is set.

This function is useful for any slave device to request a chunk of data at a time. In such case, the system bus master continues to provide the fixed data size with an incremental address. For example, there is a device to request to write in page unit.

Figure 44.14 shows the operation when enabling the combination function.

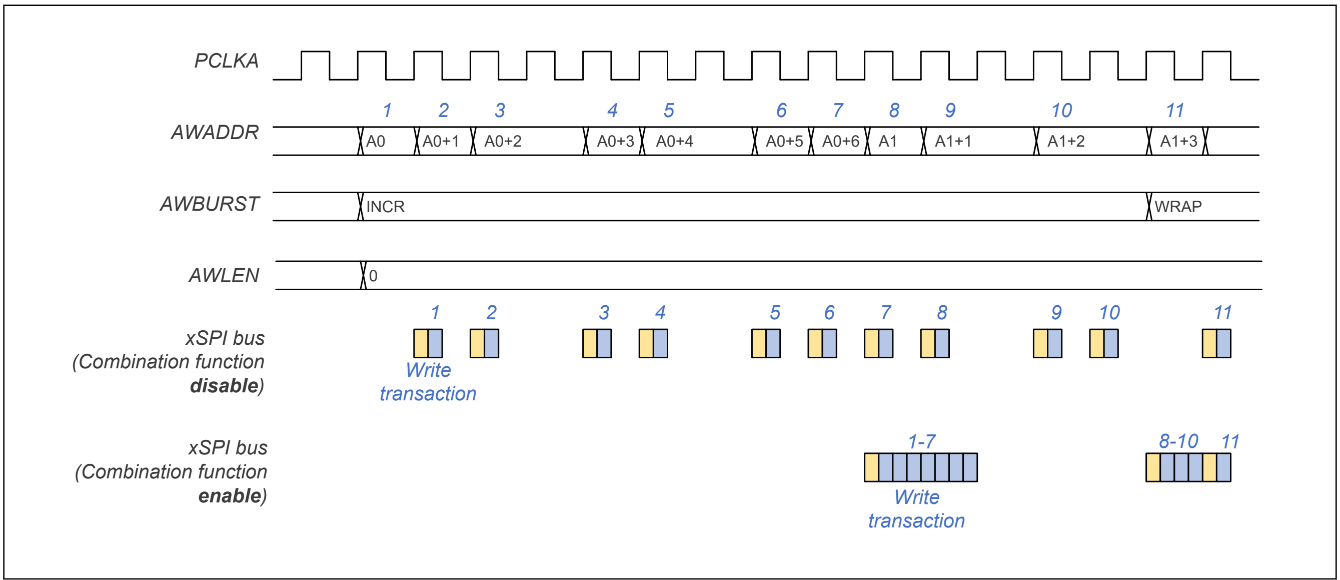


Figure 44.14 Combination function

Note 1. The access which complies with all the following conditions is considered as incremental address.

- Transaction type is INCR
- The start address of the access is continuous to the previous last write address.
  - The start position of the access for WSTRB is treated as the start address
  - The last WSTRB of the previous access is treated as the last write address.

Figure 44.15 shows a data combined example with AXI access.

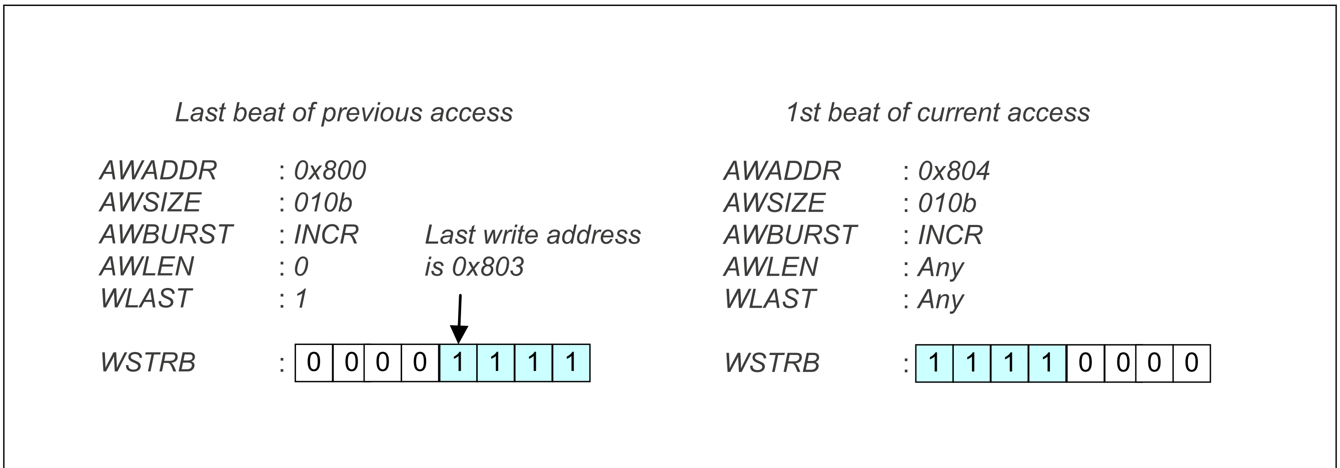


Figure 44.15 Data combined example with AXI access

44.3.3.4 Read Access Operation

At the read access for memory area, soon after detected the read access, this xSPI Master issues a read transaction into an xSPI slave. Figure 44.16 shows the operation summary.

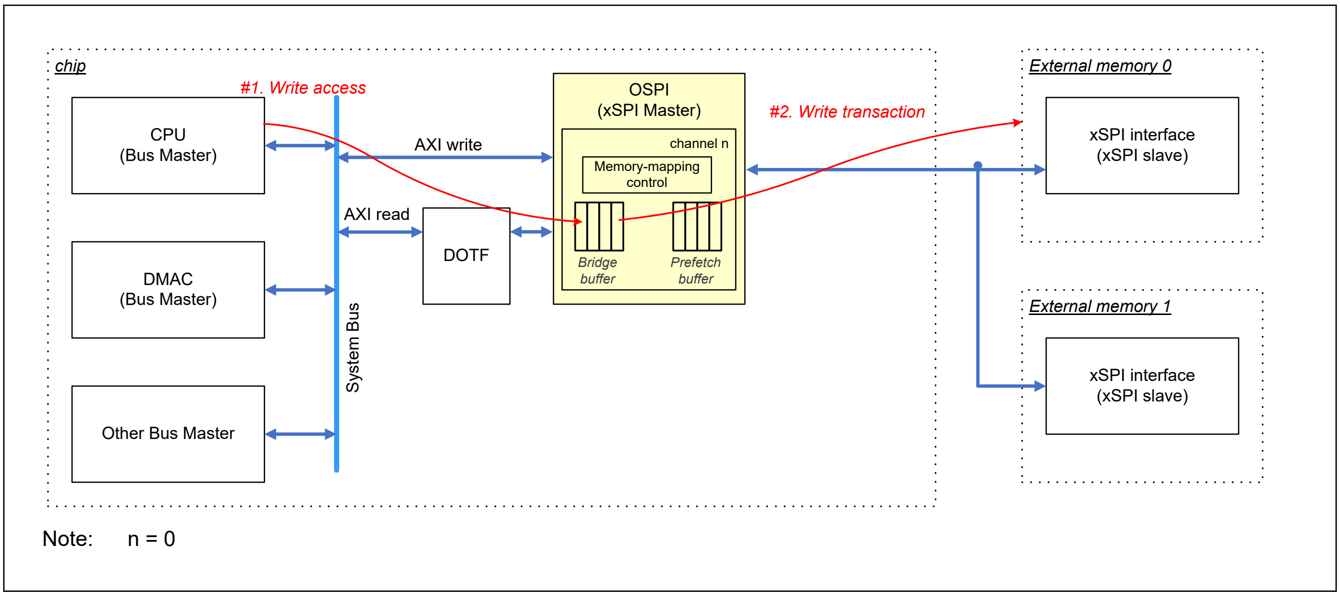


Figure 44.16 Read access for memory area

The operation of xSPI bus changes depending on the burst type. When the type is single or increment type, one read transaction of a system bus triggers one xSPI frame. When the type is wrap type and the CMCFG0CSn.WPBSTMD is 0, one read transaction of a system bus triggers two xSPI frames. Figure 44.17 shows the relationship between AXI and xSPI frames.

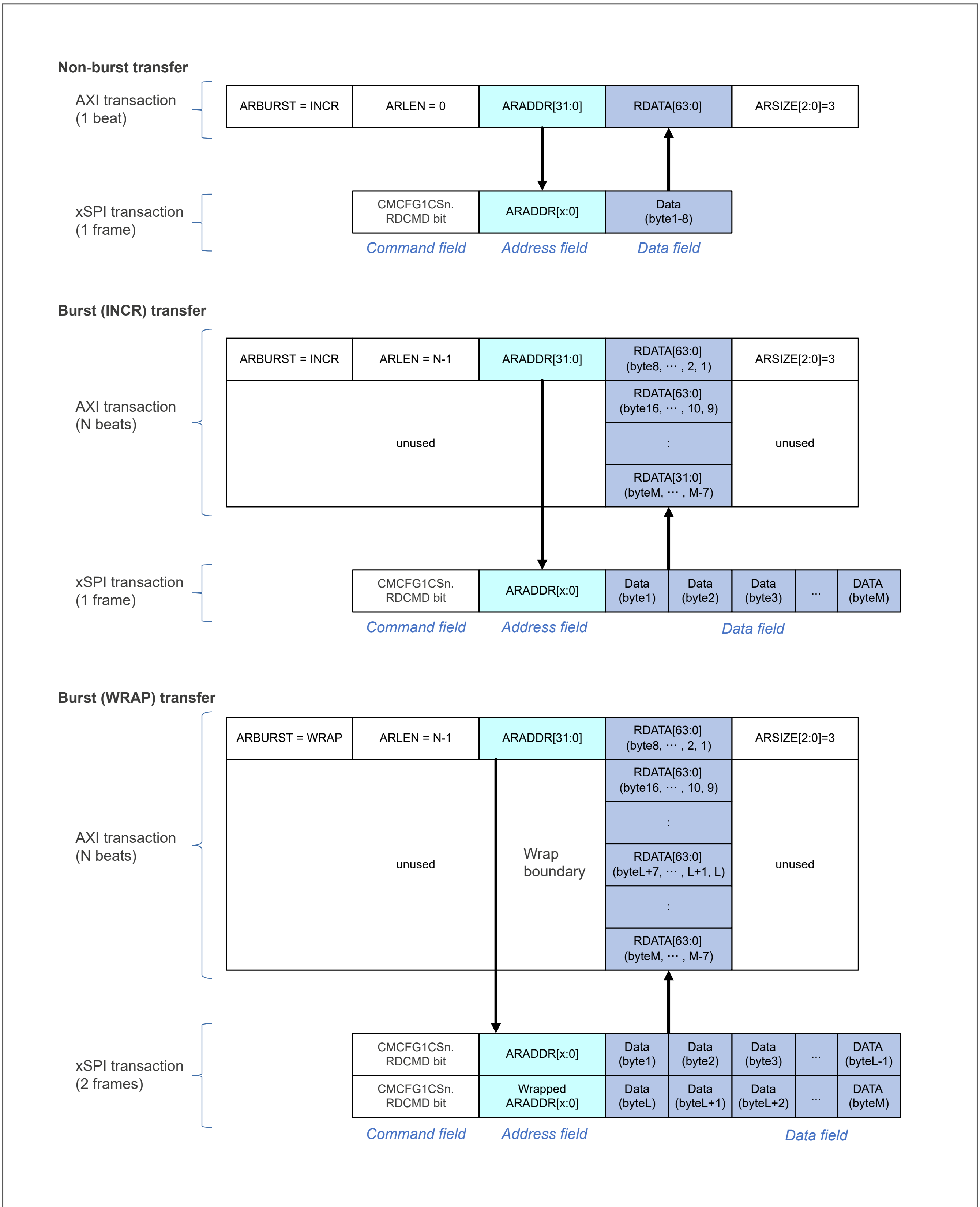


Figure 44.17 xSPI frame format in read access (Normal format)

44.3.3.5 Prefetch Function

At the read access for memory area from the system bus, this xSPI Master has the function to prefetch read data for reducing the latency. When enabled this function (BMCFGCHn.PREEN = 1), this xSPI Master continues to read the incremental address and store the read data from an xSPI slave in internal prefetch buffer. This xSPI master then searches in the prefetch buffer for the following read access from the system bus. If the target read data in the prefetch buffer is found, this xSPI Master returns the data from the prefetch buffer. If not found, this xSPI Master clears the prefetch buffer and issues a new read transaction into an xSPI slave. This function is effective in applications such as when the consecutive read addresses are close. But it is not effective in applications such as when the consecutive read addresses are not incremental because xSPI read frame for prefetch uses xSPI bus. Figure 44.18 shows the operation summary.

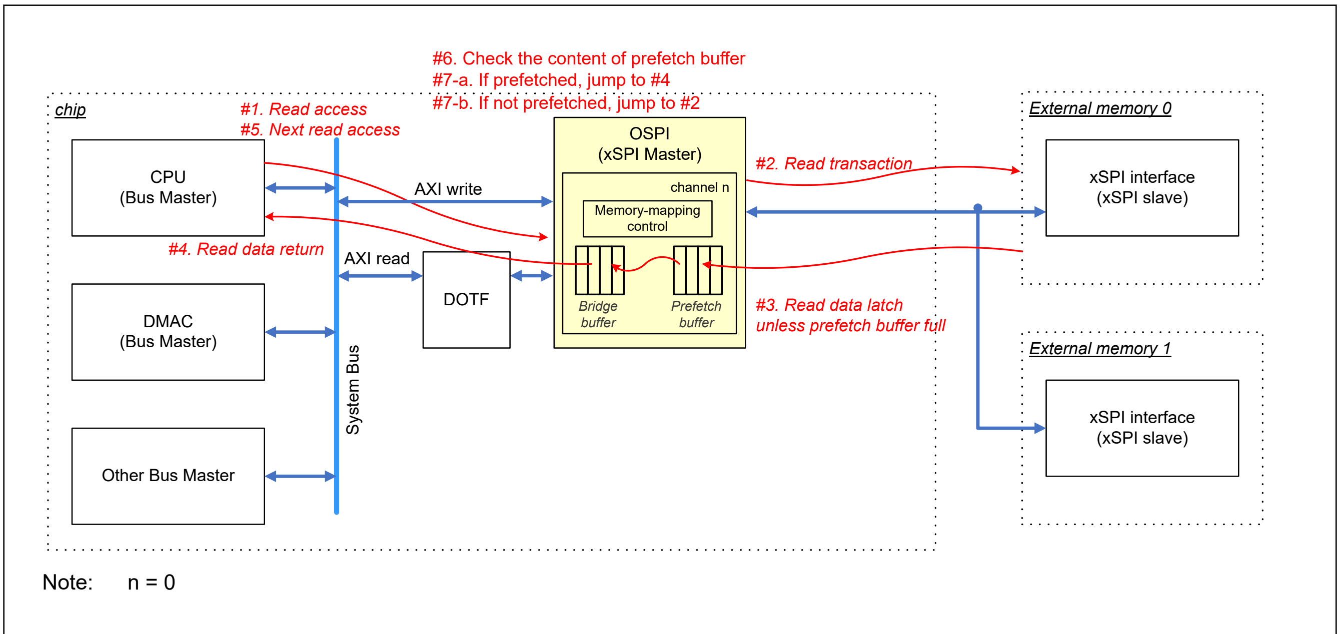


Figure 44.18 Read access for memory area with prefetch enabled

Note: When enabled this prefetch function, the bus master can read from the internal prefetch buffer but not from a slave device. When accessed to the same address from multiple bus masters, this xSPI Master does not guarantee to read the latest data. If the bus master wants to read the latest data from a slave device, it should read after clearing the prefetch buffer (BMCTL1.PBUFCLRCHn (n = 0)).

Note: Prefetch buffer is implemented as FIFO-based, and when a read access is issued, the data before the access address is discarded from the buffer. When the next access is issued to the region which is discarded at the previous access, this module issues the xSPI read access again to fill the prefetch buffer.

Note: This IP has a 1-line buffer which keeps the last 8-byte read data from the prefetch buffer. When a read access is issued to the same address and to the data in the 1-line buffer, this IP returns read data from the 1-line buffer.

44.3.3.6 XiP Mode

Some slave devices have a mode (XiP mode) in which the command phase is not required for lower latency. While in this mode, the xSPI master skips sending the command and the slave device implicitly performs the command that was executed in the previous transaction. When enabled XiP mode bit (CMCTLCHn.XIPEN = 1), this xSPI master inserts XiP enter code (CMCTLCHn.XIPENCODE[7:0]) in the Latency field. When disabled XiP mode bit (CMCTLCHn.XIPEN = 0), this xSPI master inserts XiP exit code (CMCTLCHn.XIPEXCODE[7:0]) in the Latency field. This function is available only for memory-mapping mode.

When the xSPI master transmits the XiP disable pattern, this master clears the XiP mode bit and disables XiP mode configured for both channels. It is not possible to disable XiP mode for only one channel by transmitting a XiP disable pattern.

Note: When enough latency cycle does not exist for XiP code, this xSPI Master cannot insert XiP code.

Note: XiP mode can be used only for unidirectional access to a slave. The write transaction and read transaction should be separated.

Note: The XiP exit code is inserted once when disabled. For more details, see Figure 44.30.

44.3.4 Pattern Control

This xSPI Master has the function to transmit three types of pattern which are not xSPI frame format. The pattern is triggered by the setting trigger bit (LPCTL0-1.PATREQ).

44.3.4.1 XiP Disable Pattern

XiP Disable pattern transmits any pattern with the configured length and value (LPCTL0.XD1LEN[4:0]/XD1VAL/XD2LEN[4:0]/XD2VAL). XiP Disable pattern uses OM_SCLK, OM_SIO7-0 signals. The number of output pins can be configured by the XiP Disable pattern pin bits (LPCTL0.XDPIN[1:0]). It may be used to disable XiP mode for legacy SPI. Figure 44.19 shows the timing diagram.

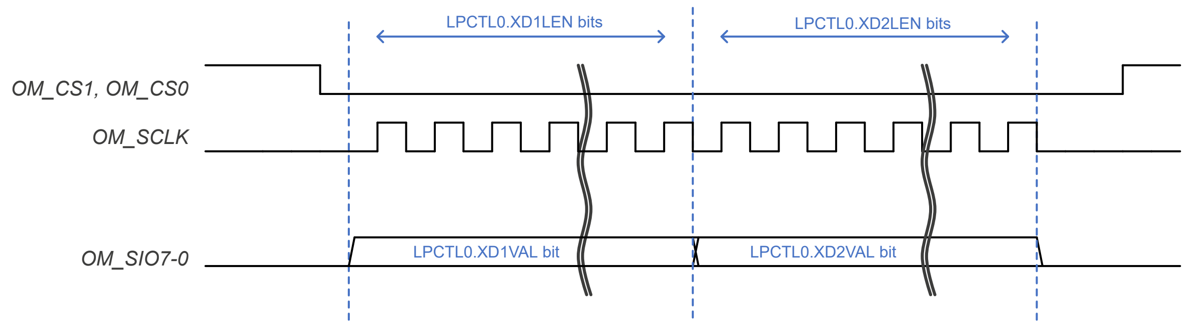


Figure 44.19 XiP Disable pattern

44.3.4.2 Reset Pattern

Reset pattern transmits the pattern specified in the Serial Flash Reset Signaling Protocol. Figure 44.20 shows the timing diagram.

CS Low/High width is configured with the Reset Pattern Length bits (LPCTL1.RSTWID[2:0]). The xSPI slave samples the data input at the rising edge of CS. Setup time for data output is configured with Reset pattern data output setup time bits (LPCTL1.RSTSU[2:0]). The setup time should always be less than Reset pattern width.

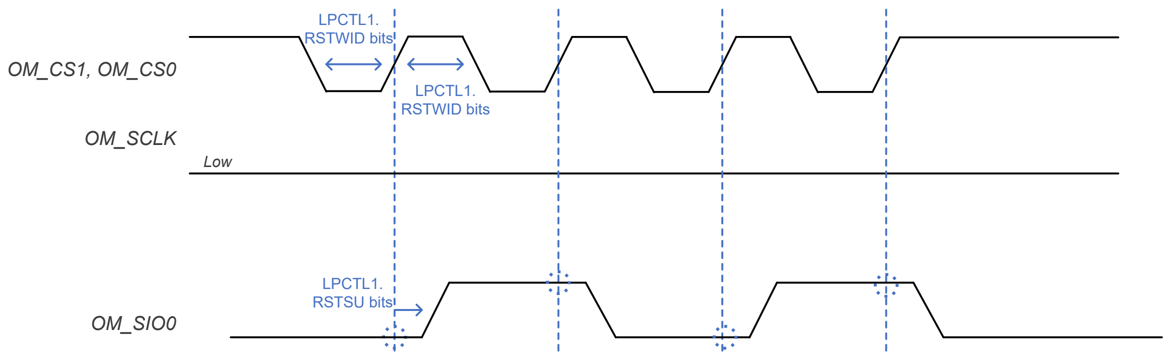


Figure 44.20 Reset pattern

Note: In the protocol, CS Low/High width is defined as minimum 500 ns and setup time is defined as minimum 6 ns.

44.3.4.3 CS-only Pattern

CS-only pattern activates the CS port with the configured length bits (LPCTL1.RSTWID[2:0]). It may be used to resume from the Deep Power Down state. Figure 44.21 shows the timing diagram.

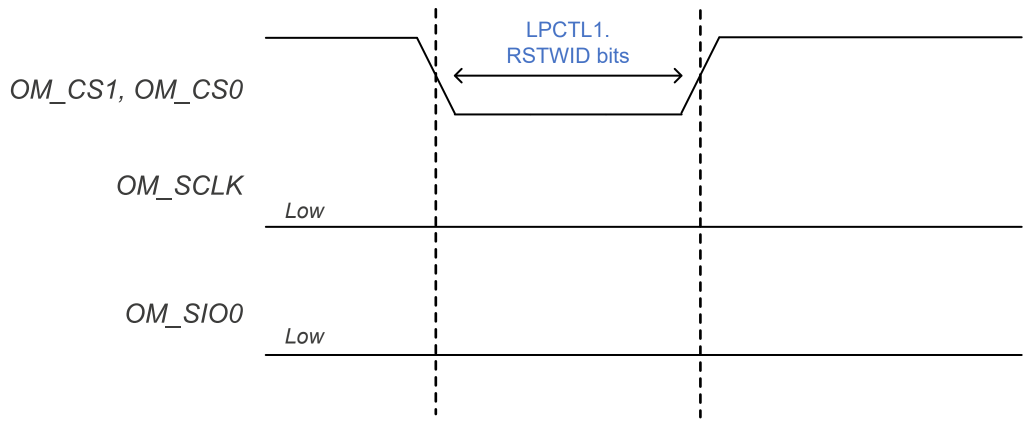


Figure 44.21 CS-only pattern

44.3.5 Integrity Checking

This xSPI Master can detect some errors. Table 44.8 shows the error list and the detail behavior.

Table 44.8 Error list (n = 0, 1)

Error type	Event	Flag bit	Note (action)
Calibration failed	When the read data did not match the expected value during automatic calibration.	INTS.CAFAILCSn	It results in writing unexpected data to xSPI slave.
System bus error	When an error response occurred on the AXI slave interface for memory-mapping.	INTS.BUSERRCHn	This xSPI master should be reset for fatal error.
ECC error detection	When detected the falling edge on OM_ECSINT1 port. It can be useful only for xSPI slave with ECC detection function.	INTS.ECSCS1	Only notify the error event of xSPI slave.
DS timeout	When DS does not toggle in read transaction with using DS.	INTS.DSTOCSn	If this error occurs during calibration, clear the interrupt status bit. Other than above, both xSPI master and xSPI slave should be reset for fatal error.
Periodic transaction timeout	When the read value does not match with the expected value in periodic manual-command mode.	INTS.PERTO	Depending on the status of the function.

44.3.6 Interrupts

This xSPI Master has an interrupt port. It can monitor with the Interrupt Status Register (INTS). During the initialization phase, it can be programmable with the Interrupt Enable register (INTE). Table 44.9 shows OSPI interrupt sources, and Table 44.10 shows the related register bit.

Note: The interrupt pulse port signal is not asserted when the corresponding bit of INTE is set after the interrupt event is detected.

Table 44.9 OSPI interrupt sources

Name	Interrupt sources	DMAC activation
OSPI0_ERR	Error	Not possible
OSPI0_CMP	Complete	Not possible

Table 44.10 Interrupt register bit

Flag bit	Enable bit	Clear bit	Interrupt sources
CASUCCS1	CASUCCS1E	CASUCCS1C	OSPI0_CMP
CASUCCS0	CASUCCS0E	CASUCCS0C	
CAFAILCS1	CAFAILCS1E	CAFAILCS1C	OSPI0_ERR
CAFAILCS0	CAFAILCS0E	CAFAILCS0C	
BUSERRCH0	BUSERRCH0E	BUSERRCH0C	OSPI0_ERR
INTCS1	INTCS1E	INTCS1C	OSPI0_ERR
ECSCS1	ECSCS1E	ECSCS1C	OSPI0_ERR
DSTOCS1	DSTOCS1E	DSTOCS1C	OSPI0_ERR
DSTOCS0	DSTOCS0E	DSTOCS0C	
PERTO	PERTOE	PERTOC	OSPI0_ERR
PATCMP	PATCMPE	PATCMPC	OSPI0_CMP
CMDCMP	CMDCMPE	CMDCMPC	OSPI0_CMP

44.3.7 Flows of Operations

44.3.7.1 Flow of Configuration

Figure 44.22 shows flow of configuration.

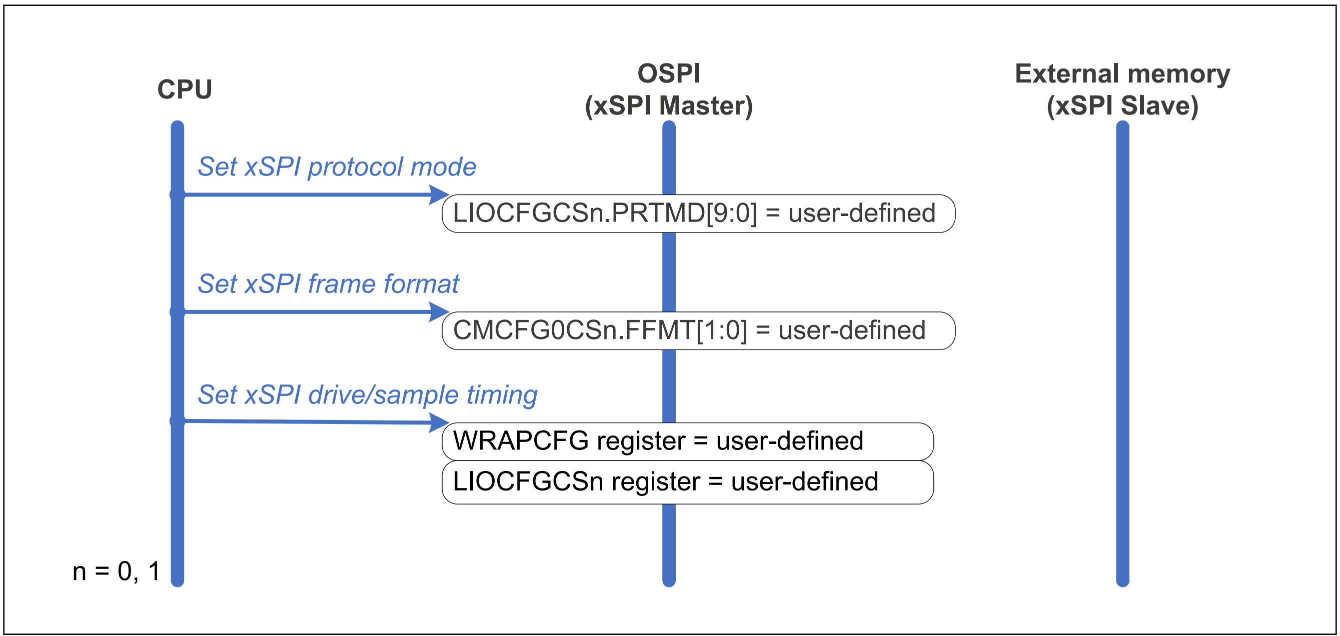


Figure 44.22 Flow of configuration

44.3.7.2 Flow of Communication Stop

Figure 44.23 shows the flow of a communication stop.

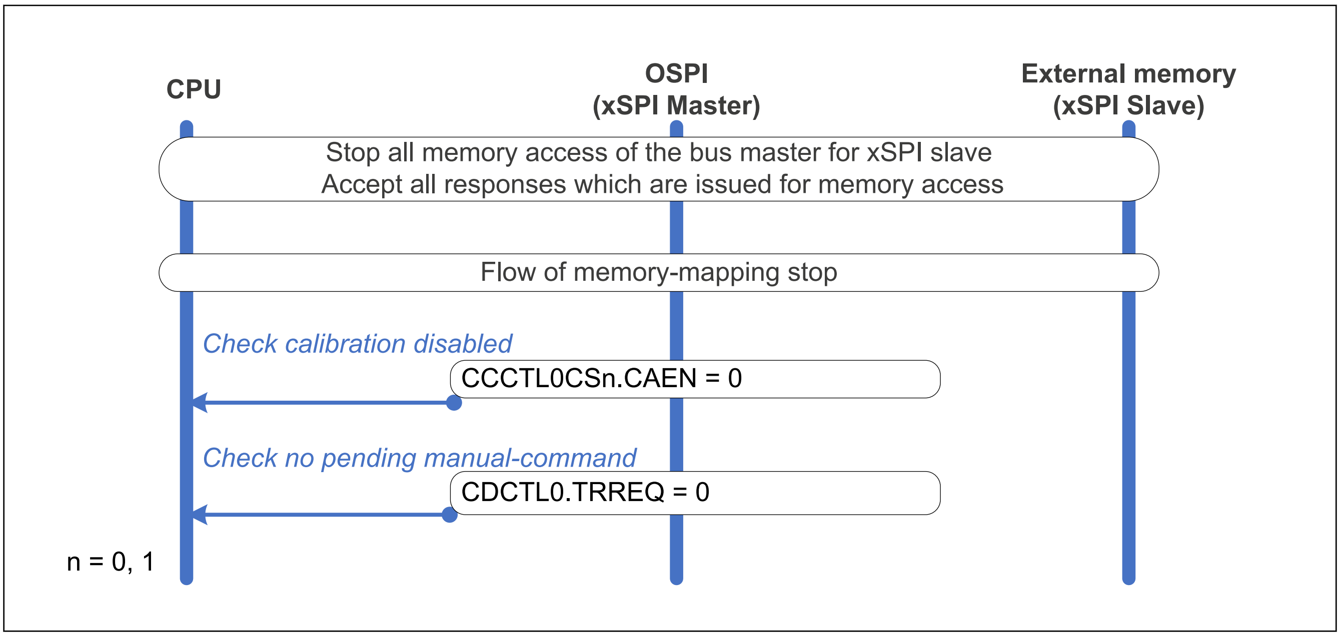


Figure 44.23 Flow of communication stop

Note: When reconfiguring any configuration register, stop all communication with the xSPI slave to avoid race condition between register setting and memory access. It means the automatic calibration is disabled, and there is no pending manual-command and memory-mapping access.

44.3.7.3 Flow of Automatic Calibration

Figure 44.24 shows the flow of an automatic calibration.

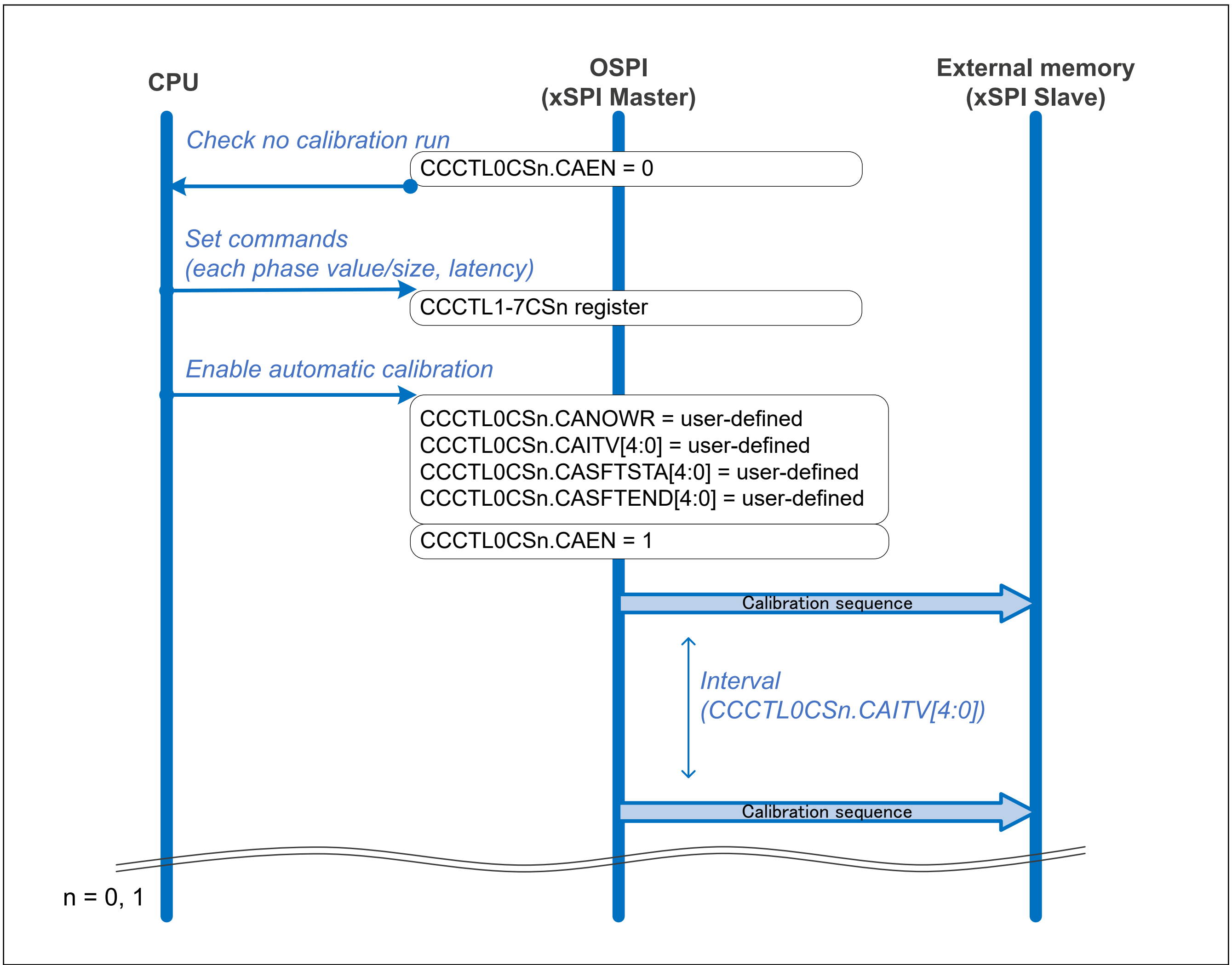


Figure 44.24 Flow of automatic calibration

44.3.7.4 Flow of Manual-command Procedure

Figure 44.25 shows the manual-command procedure for direct mode.

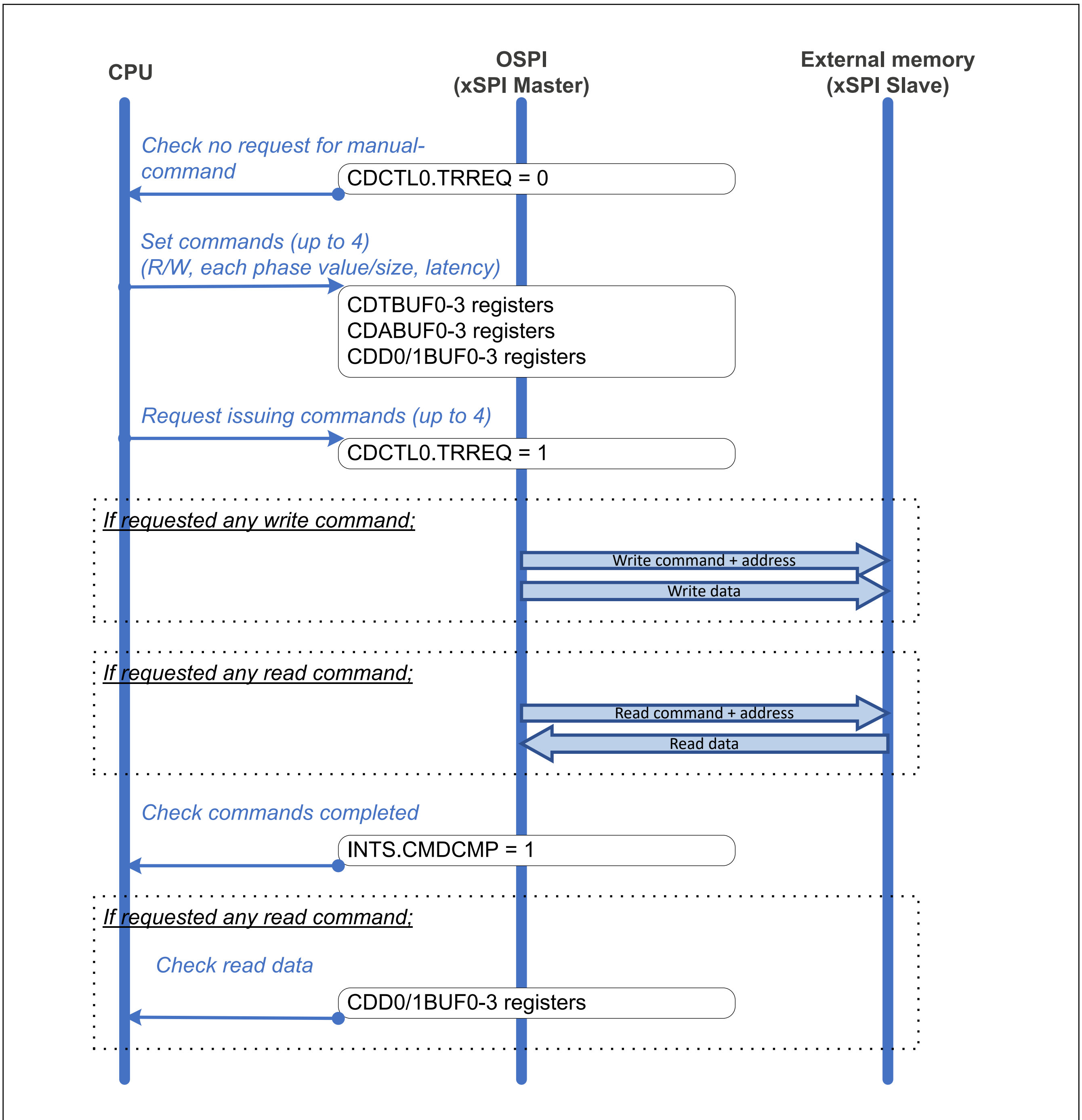


Figure 44.25 Flow of manual-command procedure for direct mode

Figure 44.26 shows the manual-command procedure for periodic mode.

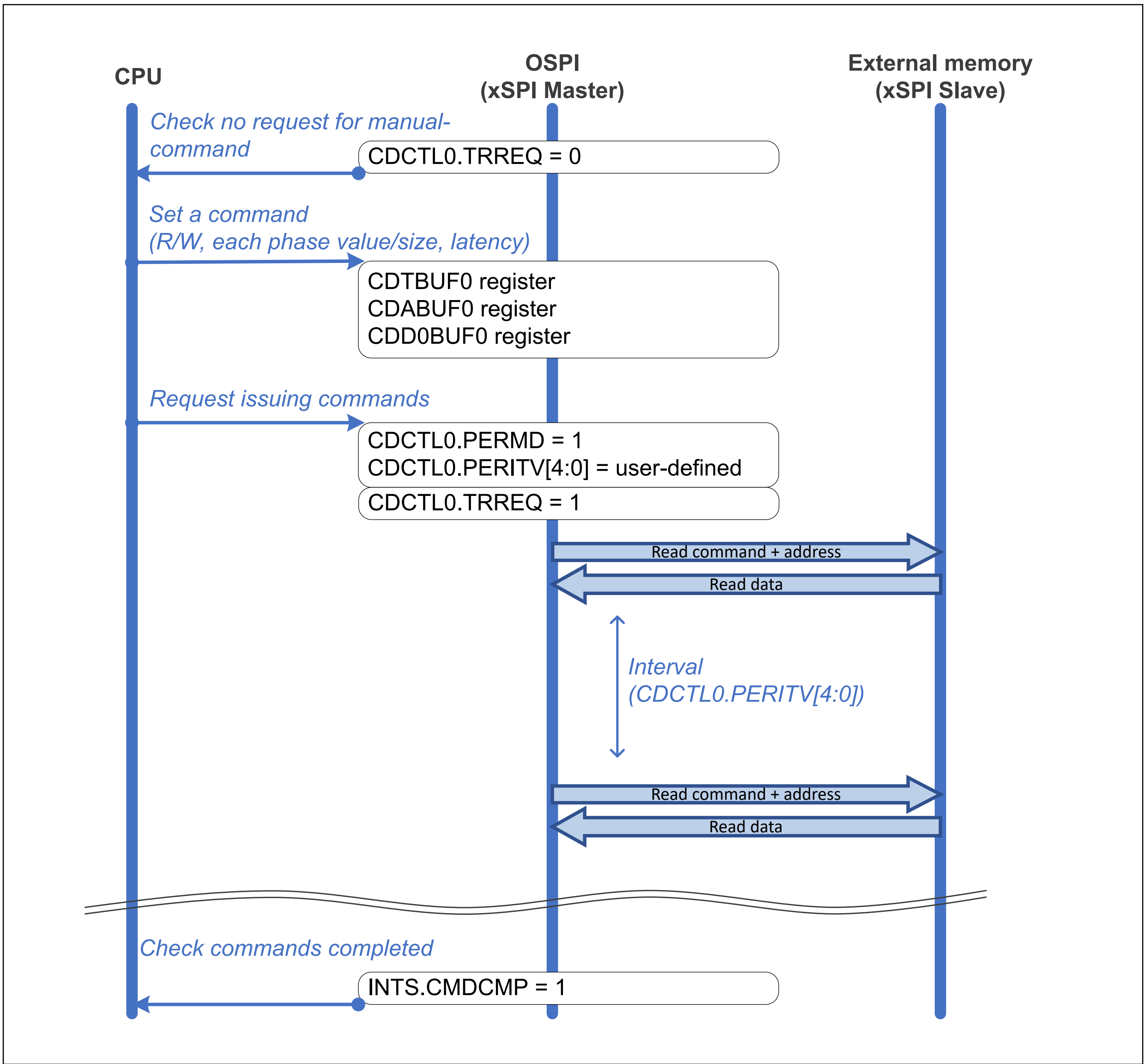


Figure 44.26 Flow of manual-command procedure for periodic mode

44.3.7.5 Flow of Memory-mapping

Figure 44.27 shows the flow of a memory-mapping.

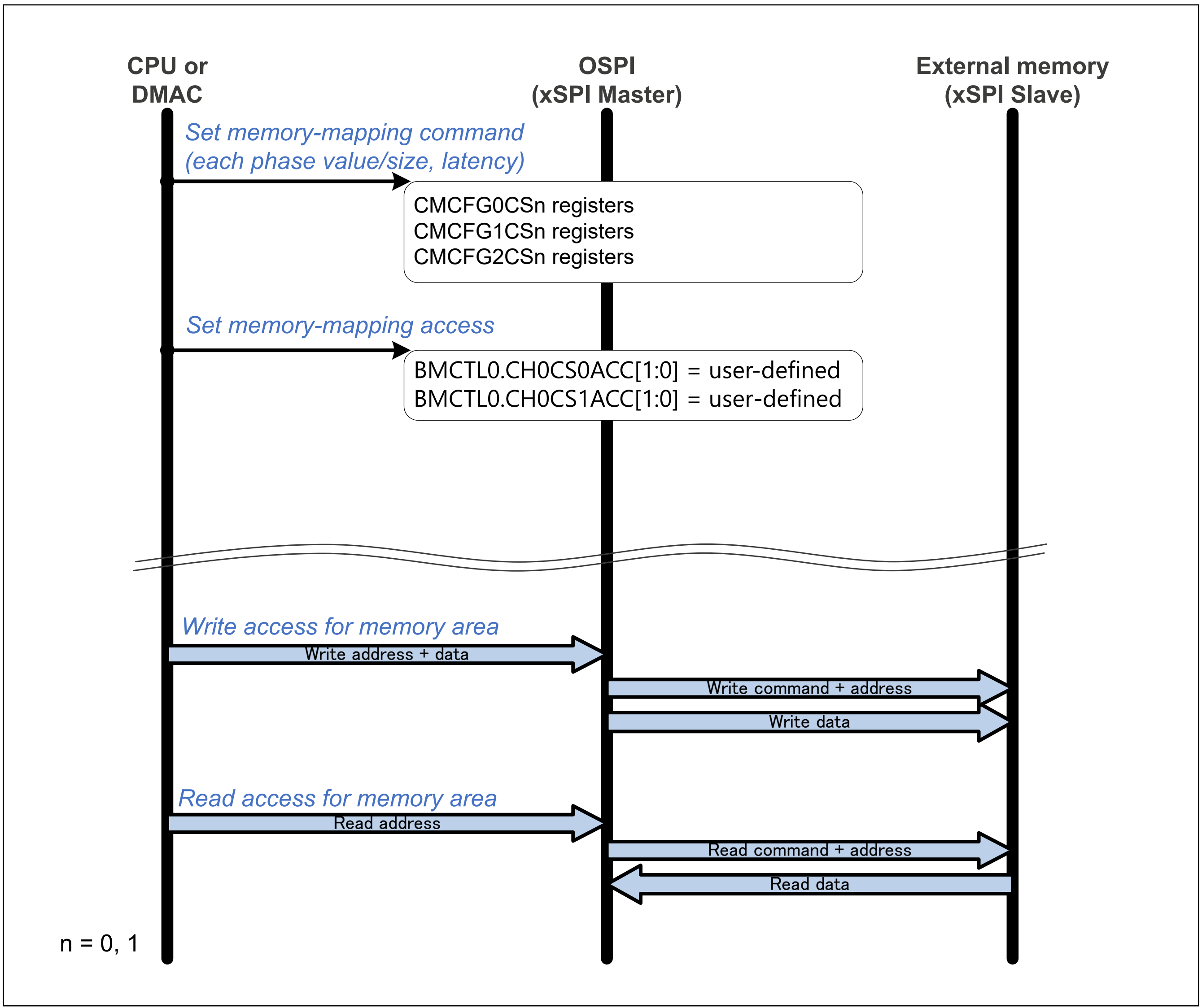


Figure 44.27 Flow of memory-mapping

44.3.7.6 Flow of Memory-mapping Stop

Figure 44.28 shows the flow of a memory-mapping stop.

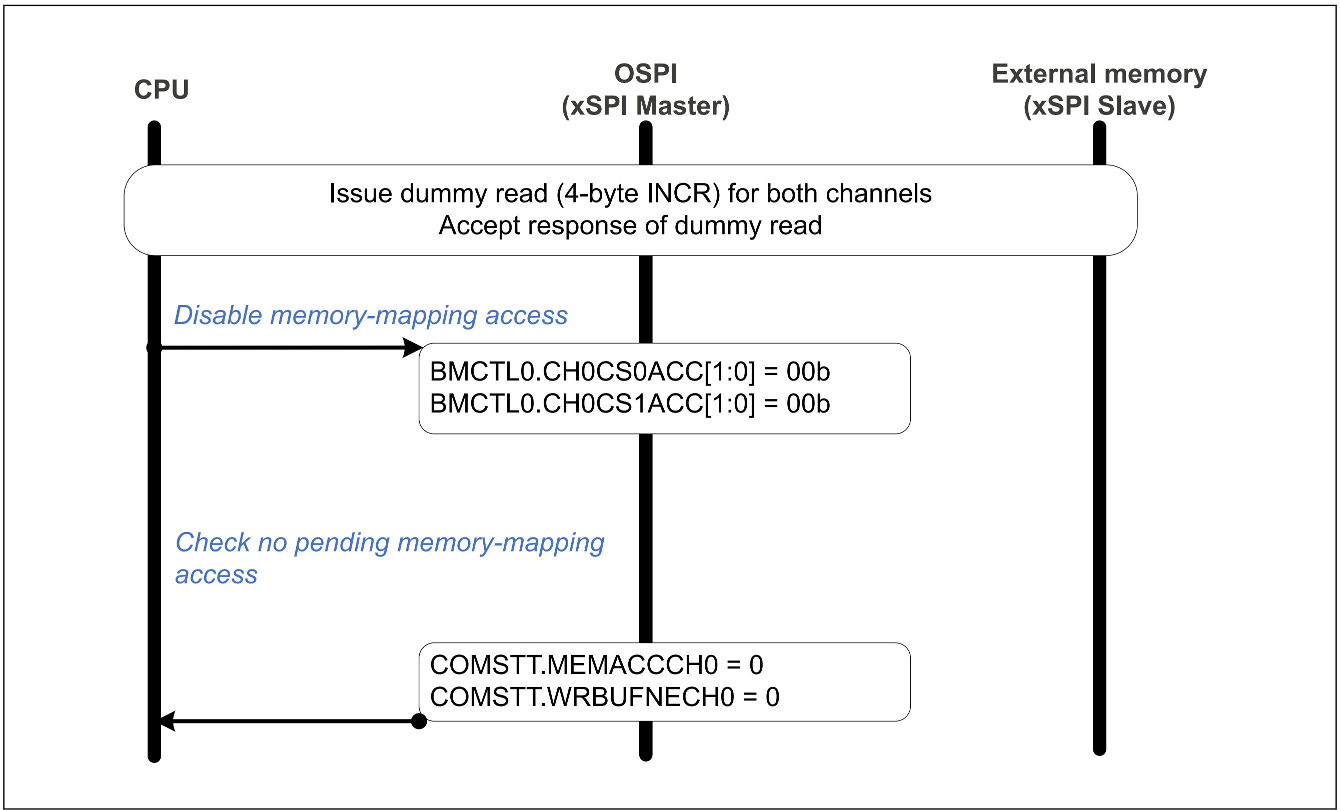


Figure 44.28 Flow of memory-mapping stop

44.3.7.7 Flow of Pattern Request

Figure 44.29 shows the flow of a pattern request. Before requesting any pattern, any ongoing commands should be completed or canceled.

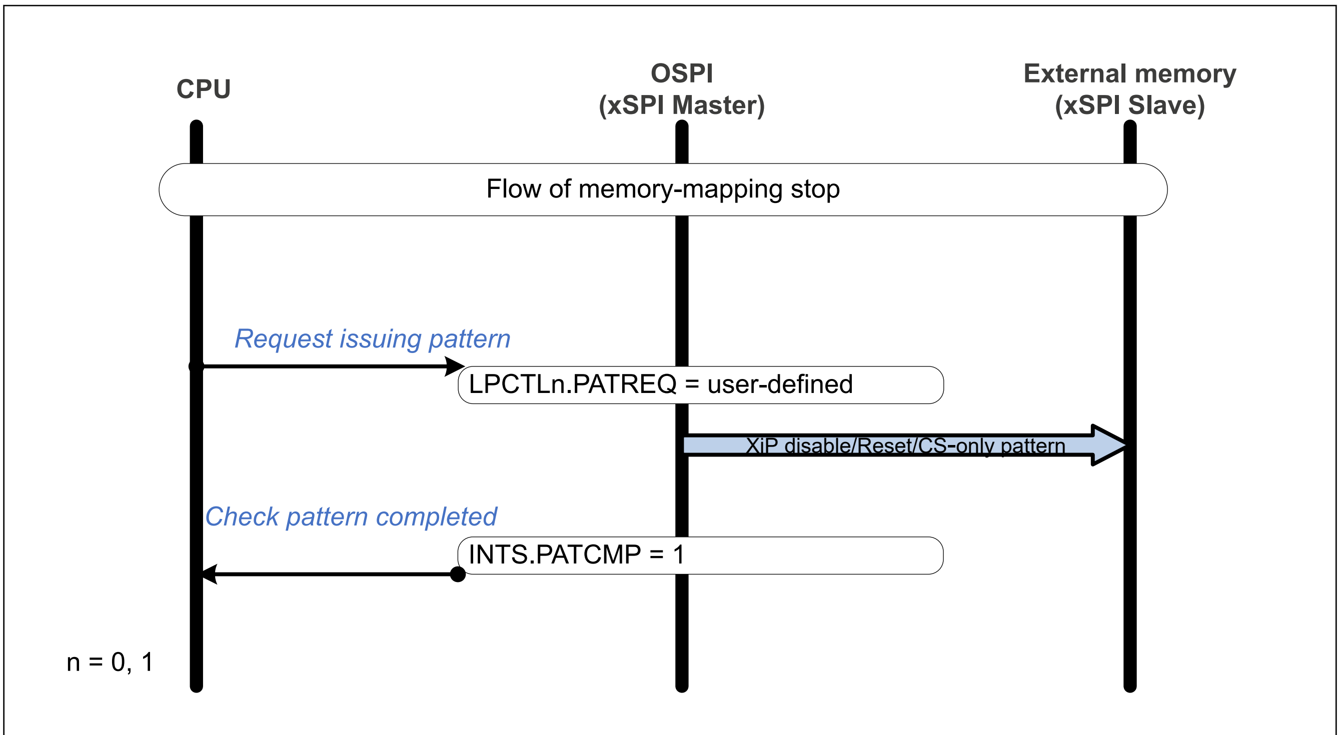


Figure 44.29 Flow of pattern request

44.3.7.8 Flow of XiP Mode

Figure 44.30 shows the flow of XiP mode enable/disable.

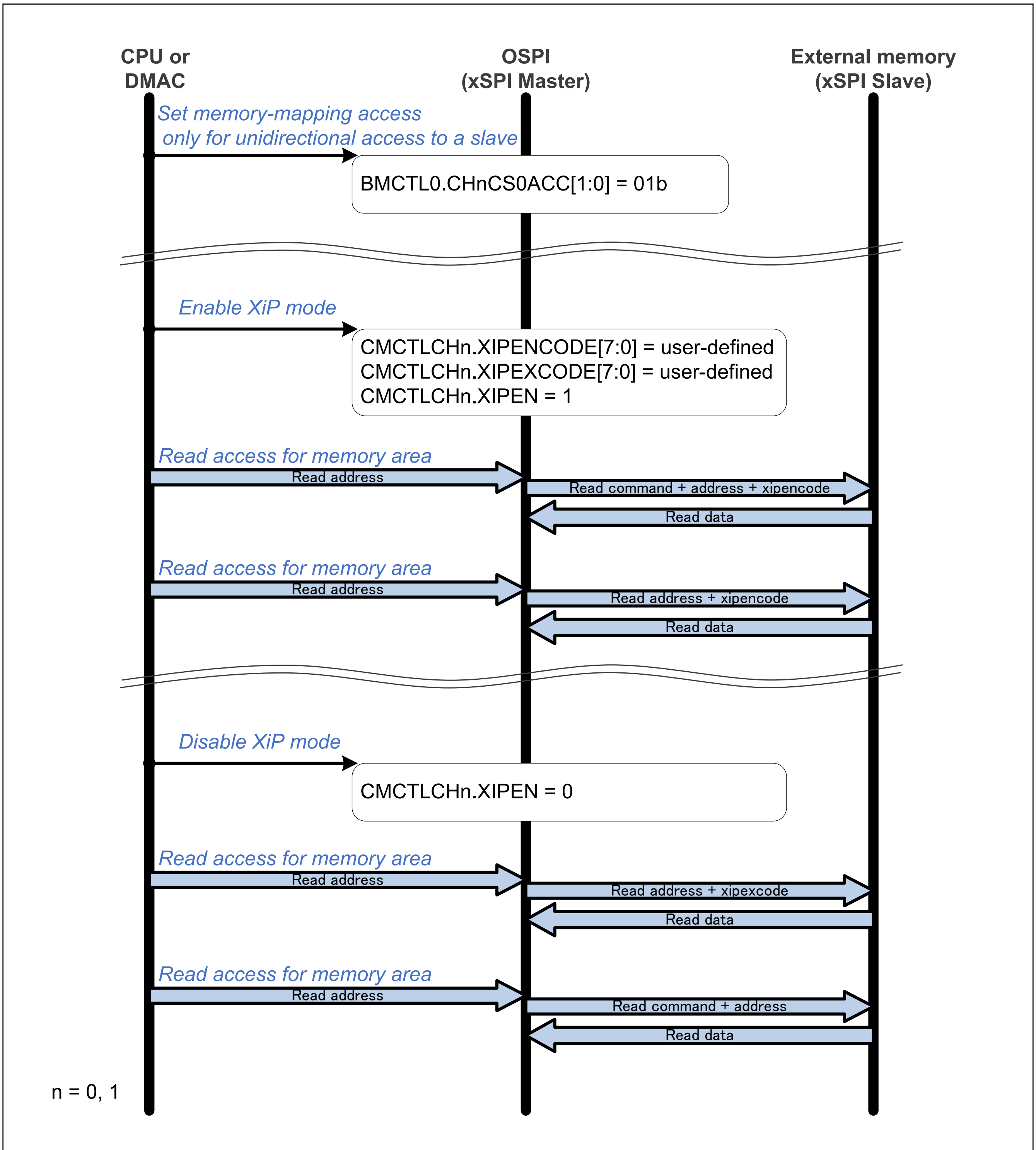


Figure 44.30 Flow of XiP mode enable/disable

44.3.7.9 Flow of Read While Write

Figure 44.31 shows the example flow of a read while write.

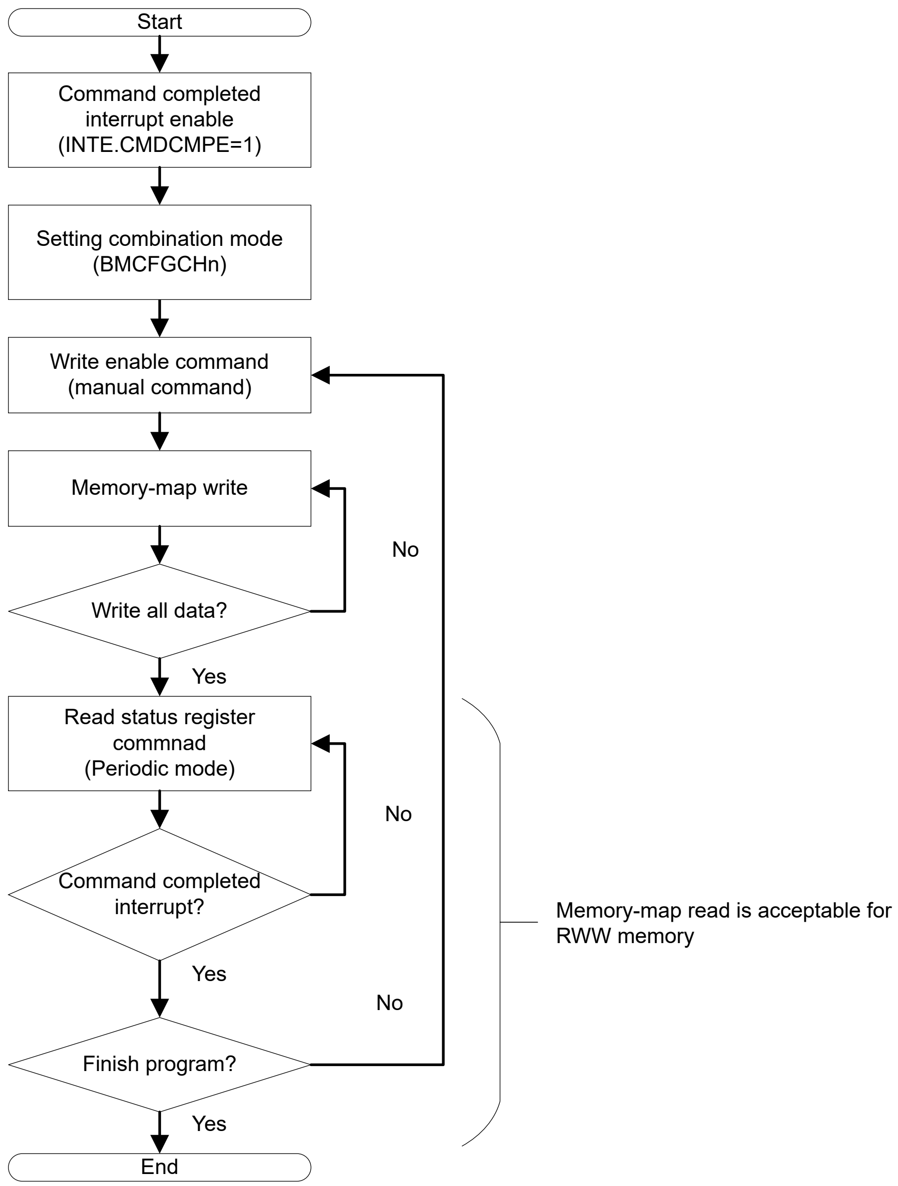


Figure 44.31 Flow of read while write

44.4 SiP Product Configuration

Flash memory ISSI IS25WX064-JWLE is packaged for SiP product.

- VCC2 = 1.70 V to 2.00 V
- OM_1_SCLK frequency:
  - SDR without OM_DQS: Up to 50 MHz
  - SDR with OM_DQS/DDR: Up to 133 MHz.

For details, see section 62, Electrical Characteristics and the datasheet of IS25WX064.

The following settings should be set before accessing the SiP Flash memory.

- Set OSPI (11100b) to the PmnPFS.PSEL[4:0] bits of P603 to P607 and PC00 to PC07
- Set the LVOCR.LVO1E bit to 1
- Set the PSARB.PSARB17 bit to 0 unless whole SiP Flash area is marked as Non-secure

- Set the end address of CS1
  - Set the RACTL.RAEN bit to 1
  - Set the MER register to 0x0871_0FF0 for 8 MB SiP-Flash product
  - Set the MER register to 0x0831_0FF0 for 4 MB SiP-Flash product
  - Set the RACTL.RAEN bit to 0
- Set the LIOCFGCS1.DDRSMPEX[3:0] bits
  The t_{CKHDSH} value of SiP Flash memory is $t_{PERIOD}/2 + 6$[ns]. When OM1_SCLK is 133 MHz, the DDRSMPEX[3:0] bits should be 0x2.

Figure 44.32 shows connection between the MCU and the SiP Flash memory. The external pull-up should be attached to the PUP pin.

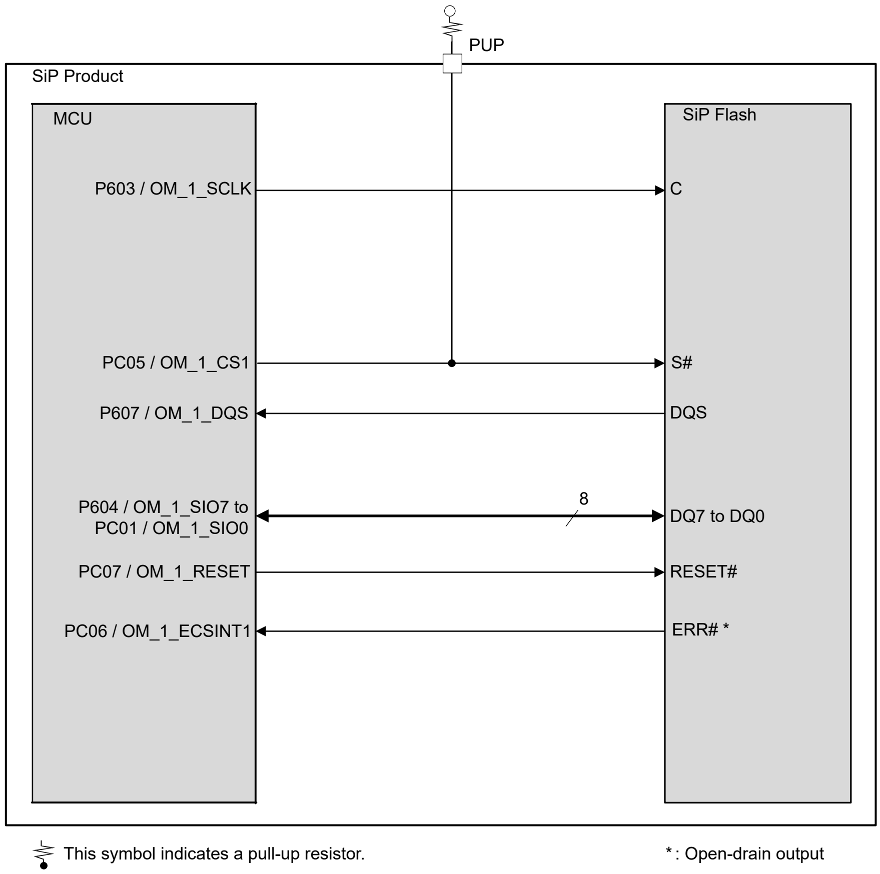


Figure 44.32 Connection between the MCU and the SiP Flash memory

44.5 Usage Notes

44.5.1 Software Standby Mode

Before transition to Software Standby mode, stop communication by Flow of Communication Stop, see section 44.3.7.2. Flow of Communication Stop.

44.5.2 Constraints on Memory Access

Memory access has the following constraints:

- Not using the prefetch function if the application has access to the last 128-byte area

- Enable the prefetch function when read HyperFlash by DMAC/DTC in Block transfer mode or Repeat block transfer mode and source address is Incremented mode.

44.5.3 Constraint on Burst Length

AXI bus master must have a burst length of 16 or less.

44.5.4 Memory Write Combination Mode

When BMCFGCHn.MWRCOMB = 1, the memory write combination mode is enabled. However, the combination does not work if the bus master is CPU0 and OSPI data is under 32 bits. Table 44.11 shows the possibility of a data write for each bus master.

Table 44.11 Data write possibility for each bus master

Bus master	Combination enable	Combination disable
CPU0 under 32-bit access	Not possible	Possible
CPU0 64-bit access	Possible	Possible
Other bus masters	Possible	Possible

44.5.5 Module-stop Function

OSPI operation can be disabled or enabled using Module Stop Control Register B (MSTPCRB). The OSPI module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Mode.

44.5.6 Byte Data Order in 8-pin DDR Mode

In 8-pin-DDR mode, OSPI read and write data in the order of byte 0 at the rising edge and byte 1 at the falling edge.

Some memory devices read and write in the swapped order in 8-pin DDR mode. For these devices, if 8-pin DDR read the data which is written in other than 8-pin DDR mode, this data should be written in swapped order.

44.5.7 Constraint on Prefetch Function

When all the following conditions are satisfied, prefetch function has no effect on the read access even if the PREEN bit is set to 1.

- Master is CPU0 or CPU1
- Memory attribution of OSPI region is set to cacheable memory by MPU in the CPU
- CPU cache is enabled
- DOTF is enabled.

44.5.8 Constraint on Block Protect setting for Flash 4MB SiP Product

Flash memory ISSI IS25WX064-JWLE has block protect function. There are some prohibited settings for Flash 4MB SiP Product (see section section 1.3. Part Numbering).

Table 44.12 shows block protect setting of TB and BP[3:0] bits in Status Register.

Table 44.12 Block protect setting for Flash 4MB SiP Product (1 of 2)

Status Register Bits					Byte Protected
TB	BP3	BP2	BP1	BP0	
1	0	0	0	0	0 KB
1	0	0	0	1	128 KB
1	0	0	1	0	256 KB
1	0	0	1	1	512 KB
1	0	1	0	0	1 MB

Table 44.12 Block protect setting for Flash 4MB SiP Product (2 of 2)

Status Register Bits					Byte Protected
TB	BP3	BP2	BP1	BP0	
1	0	1	0	1	2 MB
1	0	1	1	0	4 MB
Others: Setting prohibited					

44.5.9 Constraint on OTP Area for SiP Product

Flash memory ISSI IS25WX064-JWLE has 64-byte OTP area, but it is not supported in the SiP product.

45. Decryption On The Fly (DOTF)

45.1 Overview

DOTF has a function to decode read data of the AXI bus using AES core.

Table 45.1 lists the DOTF specifications and for a system using DOTF, see section 44.1. Overview.

Table 45.1 DOTF Specification

Item	Description
Number of channels	2 channels DOTF0 decodes the read data for OSPI0 and DOTF1 decodes the read data for OSPI1.
Clock Source	Register clock : PCLKB AES core clock : PCLKA
AES core function	● Utilize for on-the-fly decryption of encrypted software stored in external memory ● Block size: 128-bit ● Key size: 128-bit, 192-bit, 256-bit ● Support the following block cipher mode: – Counter (CTR) mode following NIST SP800-38A ● Supports side channel countermeasure function ● Supports self-test function ● Counter[127:0] = {IV[127:28], Address[31:4]}, where Address is the memory mapped address of the encrypted data, aligned to the AES block size of 128 bits (16 bytes). The IV is chosen according to the recommendations provided in Appendix B of SP800-38A. See section 44, Octal Serial Peripheral Interface (OSPI) and section 5, Address Space for details. See NIST SP800-38A for details of AES counter mode operation.
Tamper Resistance	Countermeasures available for side-channel attacks, including SPA/DPA and timing attacks
Module-stop function	Module-stop state can be set to reduce power consumption (same as OSPI module stop)
TrustZone Filter	Security and Privilege attribution can be set for each channel (same as TZF of OSPI)

45.2 Block Diagram

The block diagram of DOTF is shown in Figure 45.1. See Figure 44.1 for a diagram that includes the OSPI.

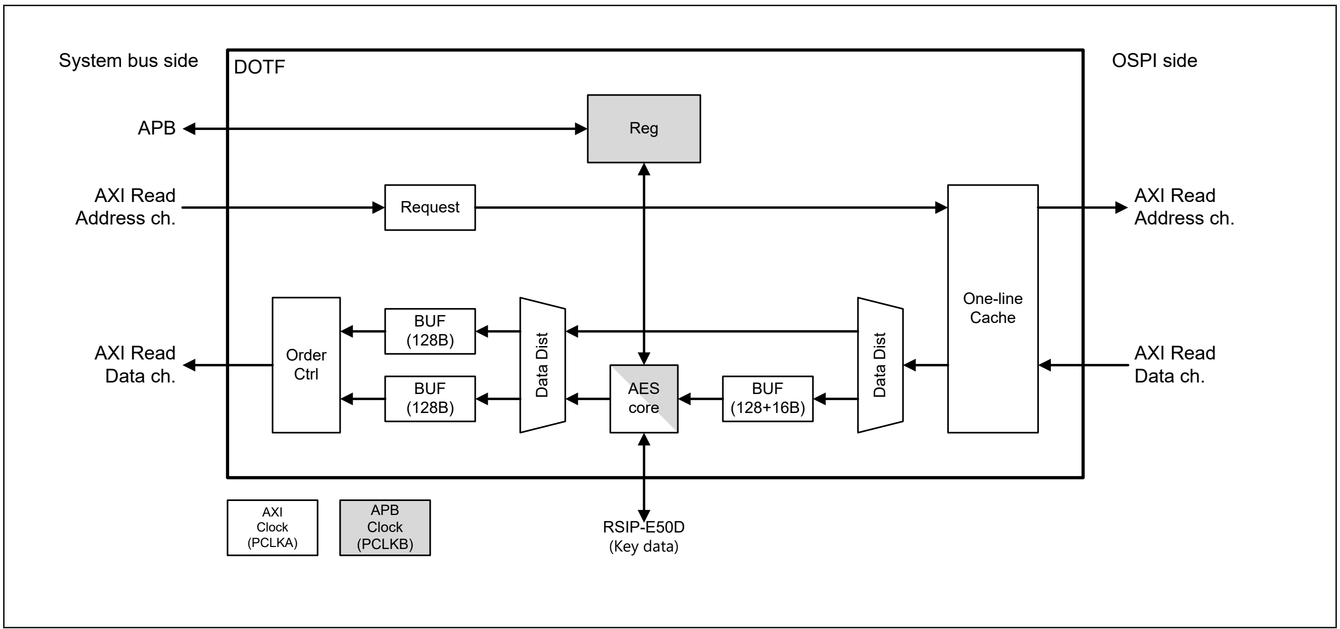


Figure 45.1 Block diagram

45.3 Register Descriptions

DOTF allows the entire area to be an accessible area, but the area where decryption processing is performed can be specified as the entire area or one contiguous area. The region can be specified by setting the start and end addresses to the CONVAREAST and CONAREAED registers, respectively.

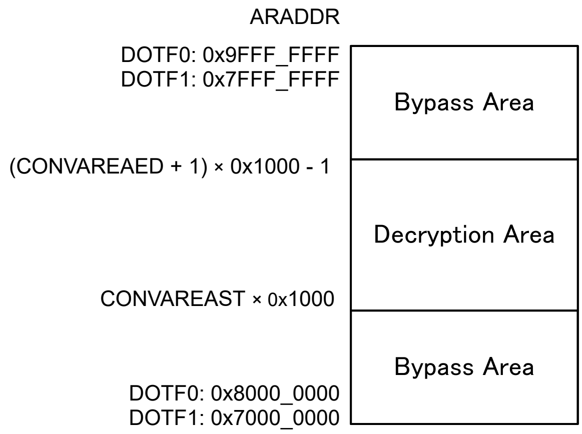


Figure 45.2 Image of decryption area setting

45.3.1 CONVAREAST : DOTF Conversion Area Start Address Register

Base address: DOTFn = 0x4026_8800 + 0x0100 × n (n = 0, 1)
DOTFn_NS = 0x5026_8800 + 0x0100 × n (n = 0, 1)

Offset address: 0x00

Bit position:	31																			12												0
Bit field:	CONVAREAST[31:12]																				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	—	These bits are read as 0.	R
31:12	CONVAREAST [31:12]	The first address of the decryption processing area. The actual address is CONVAREAST [31:12] × 0x1000.	R/W

Note: S-TYPE-3, P-TYPE-3

Specify the first address of the decryption processing area. Set before the AXI transfer request and do not make any further changes. Setting CONVAREAST[31:12] > CONVAREAED[31:12] is prohibited.

45.3.2 CONVAREAD : DOTF Conversion Area End Address Register

Base address: DOTFn = 0x4026_8800 + 0x0100 × n (n = 0, 1)
DOTFn_NS = 0x5026_8800 + 0x0100 × n (n = 0, 1)

Offset address: 0x04

Bit position:	31																			12												0
Bit field:	CONVAREAED[31:12]																				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
11:0	—	These bits are read as 1.	R

Bit	Symbol	Function	R/W
31:12	CONVAREAED [31:12]	The end address of the decryption processing area. The actual address is CONVAREAED [31:12] ×0x1000.	R/W

Note: S-TYPE-3, P-TYPE-3
Note: Setting CONVAREAST[31:12] > CONVAREAED[31:12] is prohibited.

Specify the tail address of the decryption processing area. Set before the AXI transfer request and do not make any further changes. Setting CONVAREAST[31:12] > CONVAREAED[31:12] is prohibited.

45.4 Operation

DOTF has one AXI Read Slave Interface and one AXI Read Master Interface.

An AXI transfer request received on the AXI Slave Address channel is issued to the AXI Master's Read Address channel, and the Slave data is received from the Master's AXI Read data channel and sent to the AXI Slave Read data channel.

At this time, the necessity of decryption is determined by the requested address area, and the decrypted data is transmitted through AES core if necessary. When decryption is not required, the received response is sent without conversion.

45.5 System Flow

The initialization flow after HW reset is shown in Figure 45.3.

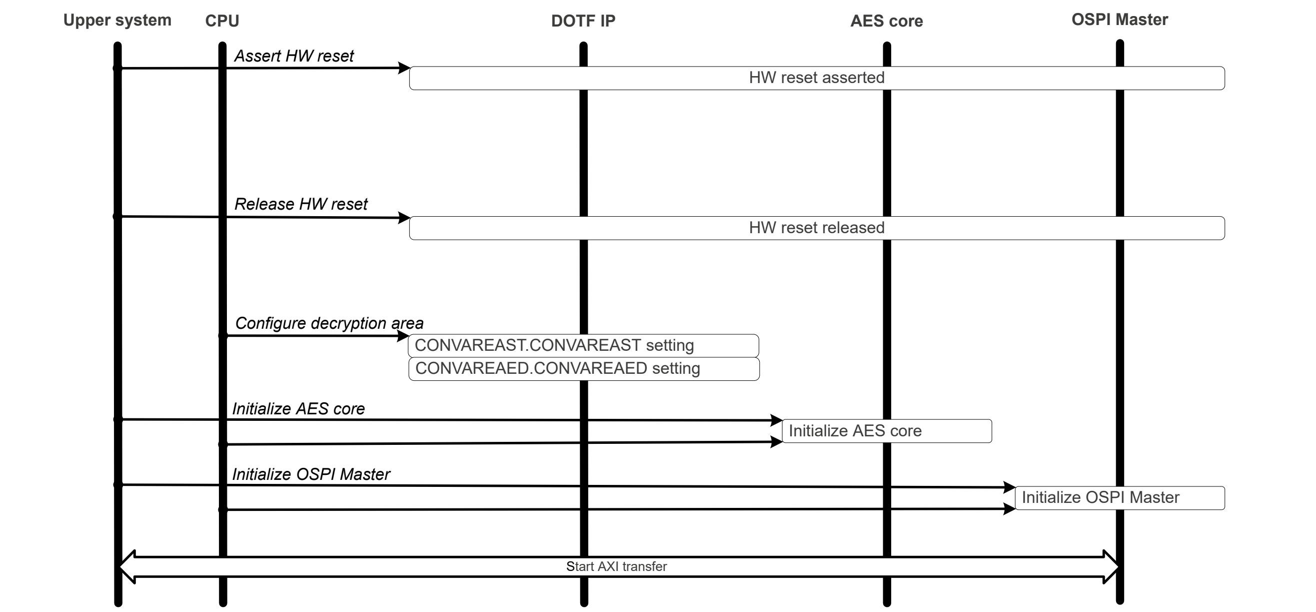


Figure 45.3 Initialization flow

45.6 Usage Notes

45.6.1 Module-stop Function

DOTF operation can be disabled or enabled using Module Stop Control Register B (MSTPCRB). The DOTF module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Mode.

46. Serial Sound Interface Enhanced (SSIE)

46.1 Overview

The Serial Sound Interface Enhanced (SSIE) can transmit and receive audio data to and from various devices that support any of audio data formats, such as I^2S, monaural, and TDM.

Table 46.1 lists the SSIE specifications, Figure 46.3, Figure 46.4 and Figure 46.5 show a block diagram of SSIE, and Table 46.4 lists the I/O pins.

Table 46.1 SSIE specifications

Item		Description
Number of channels		Two channels, SSIE0 and SSIE1
Communication mode		• Master/slave • Transmission/reception (SSIE0:full duplex communication, SSIE1:half-duplex communication)
Communication format		• I^2S format • Monaural format • TDM format
Serial data		• MSB first • Data can be left-justified or right-justified. • Data delay (1 clock cycle) or no delay selectable for the period from SSILRCKn/SSIFSn (n = 0, 1) to SSITXD0/SSIRXD0/SSIDATA1 • System word length: 8, 16, 24, 32, 48, 64, 128, or 256 bits • Data word length: 8, 16, 18, 20, 22, 24, or 32 bits • Padding polarity: Low or high
Bit clock (SSIBCKn (n = 0, 1))	In master mode	• Two clock sources available (AUDIO_CLK/GPT output (GTIOC2A)) • Clock source division ratio: 1/1, 1/2, 1/4, 1/6, 1/8, 1/12, 1/16, 1/24, 1/32, 1/48, 1/64, 1/96, and 1/128. • Supply/stop is selectable while communication is halted.
	In master/slave mode	• Polarity (rising edge or falling edge) selectable
LR clock/frame synchronization (SSILRCKn/SSIFSn (n = 0, 1))	In master mode	• Polarity (low level or high level) selectable • Supply/stop is selectable while communication is halted.
Transmit data (SSITXD0/SSIDATA1) and receive data (SSIRXD0/SSIDATA1)	Transmission	• Muting method (transmission of transmit FIFO data or transmission of data fixed to 0) selectable
FIFO	Capacity	• Transmit FIFO/receive FIFO: 4 bytes × 32 stages
	Data alignment	• Data alignment method (left-justification or right-justification) selectable for the data transfer between FIFO and shift register
Interrupt	Interrupt output	• Communication error/idle mode • Receive data full • Transmit data empty
Low power consumption function		• Whether to supply the audio clock selectable in master mode
Module stop function		Module stop state can be set for each channel to reduce power consumption.
TrustZone Filter		Security and Privilege attribution can be set for each channel.

The following table lists and defines the terms used for the communication formats SSIE can use:

Table 46.2 Definition of terms (1 of 2)

Term	Definition
Start trigger	First edge of the signal on the SSILRCKn/SSIFSn (n = 0, 1) pin when the signal is set to the value specified in LRCKP to enable communication

Table 46.2 Definition of terms (2 of 2)

Term	Definition
Frame boundary	Point where SSIE starts transferring the first data of a frame or the point where SSIE ends transferring the last data of the frame
Frame word number	Number of sound channels per frame
System word length	Number of bits per channel
Data word length	Number of significant bits per channel
Control bits for communication formats	• SSICR register: FRM, DWL, SWL, LRCKP, SPDP, SDTA, PDTA, and DEL bits • SSIFCR register: BSW bit • SSIOFR register: OMOD bit • SSISCR register: TDES[4:0] and RDFS[4:0] bits

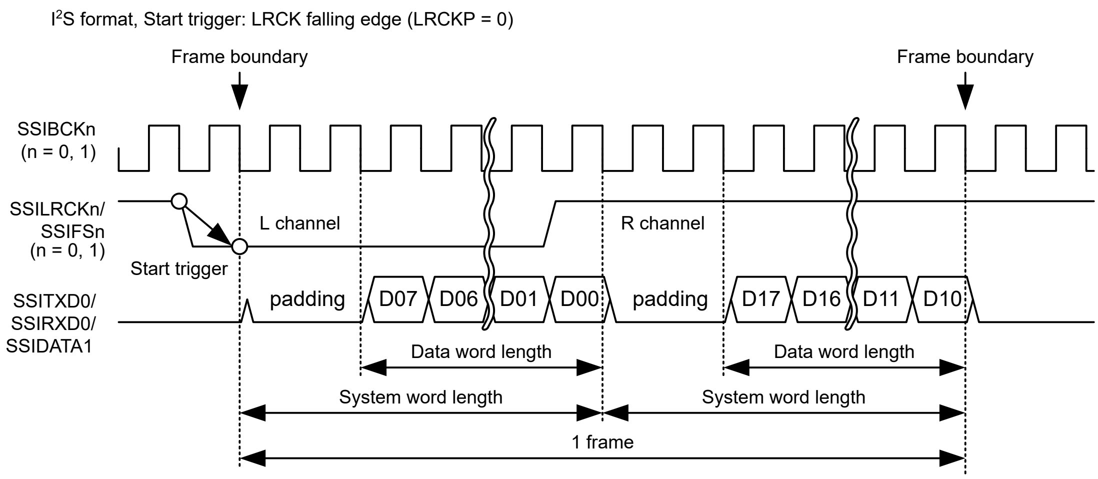
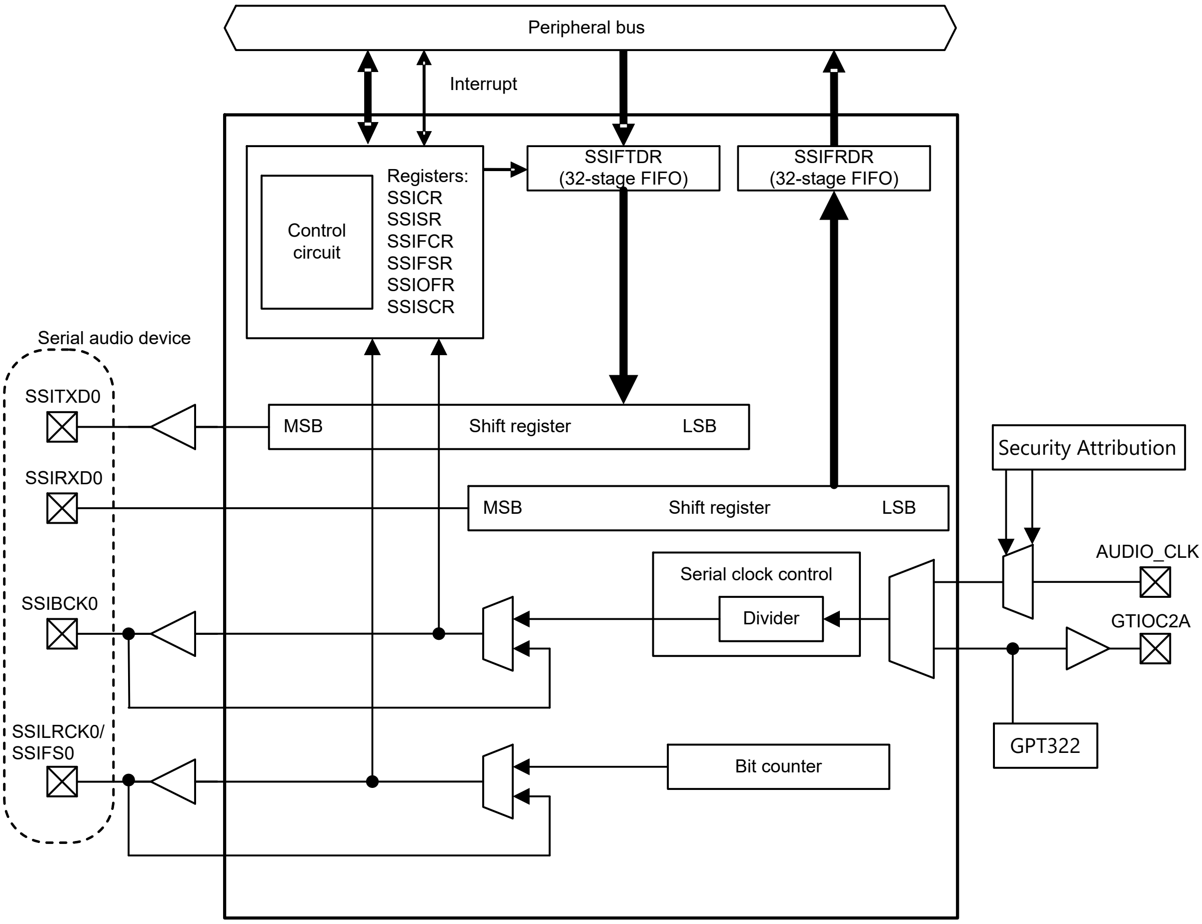


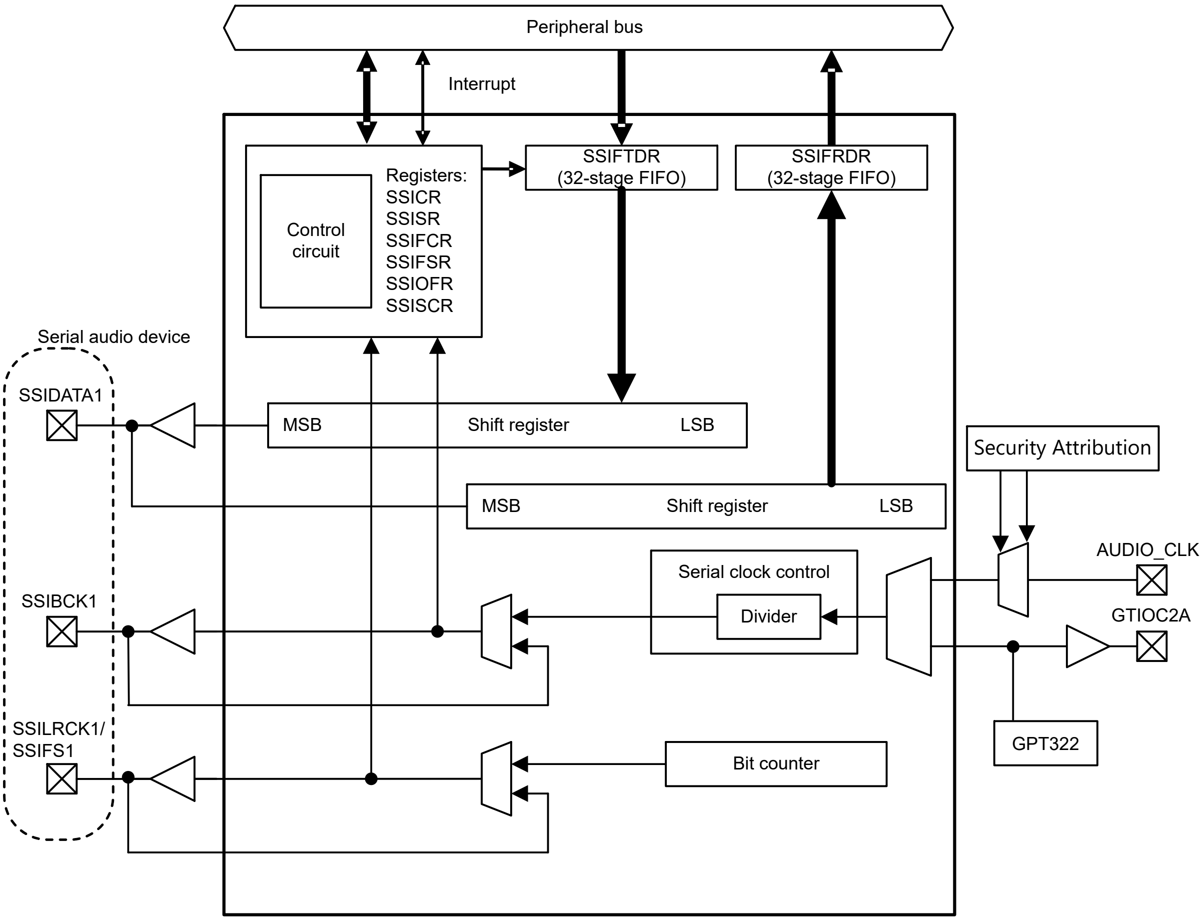
Figure 46.1 Definition of communication format

Figure 46.2 and Figure 46.3 shows a block diagram of SSIE.



[Legend]
SSICR: Control register
SSISR: Status register
SSIFCR: FIFO control register
SSIFSR: FIFO status register
SSIOFR: Audio format register
SSIFTDR: Transmit FIFO data register
SSIFRDR: Receive FIFO data register
SSISCR: Status control register

Figure 46.2 SSIE block diagram (SSIE0)



[Legend]

SSICR: Control Register
SSISR: Status Register
SSIFCR: FIFO Control Register
SSIFSR: FIFO Status Register
SSIOFR: Audio Format Register
SSIFTDR: Transmit FIFO Data Register
SSIFRDR: Receive FIFO Data Register
SSISCR: Status Control Register

Figure 46.3 SSIE block diagram (SSIE1)

Figure 46.4 shows the clock configuration of SSIE.

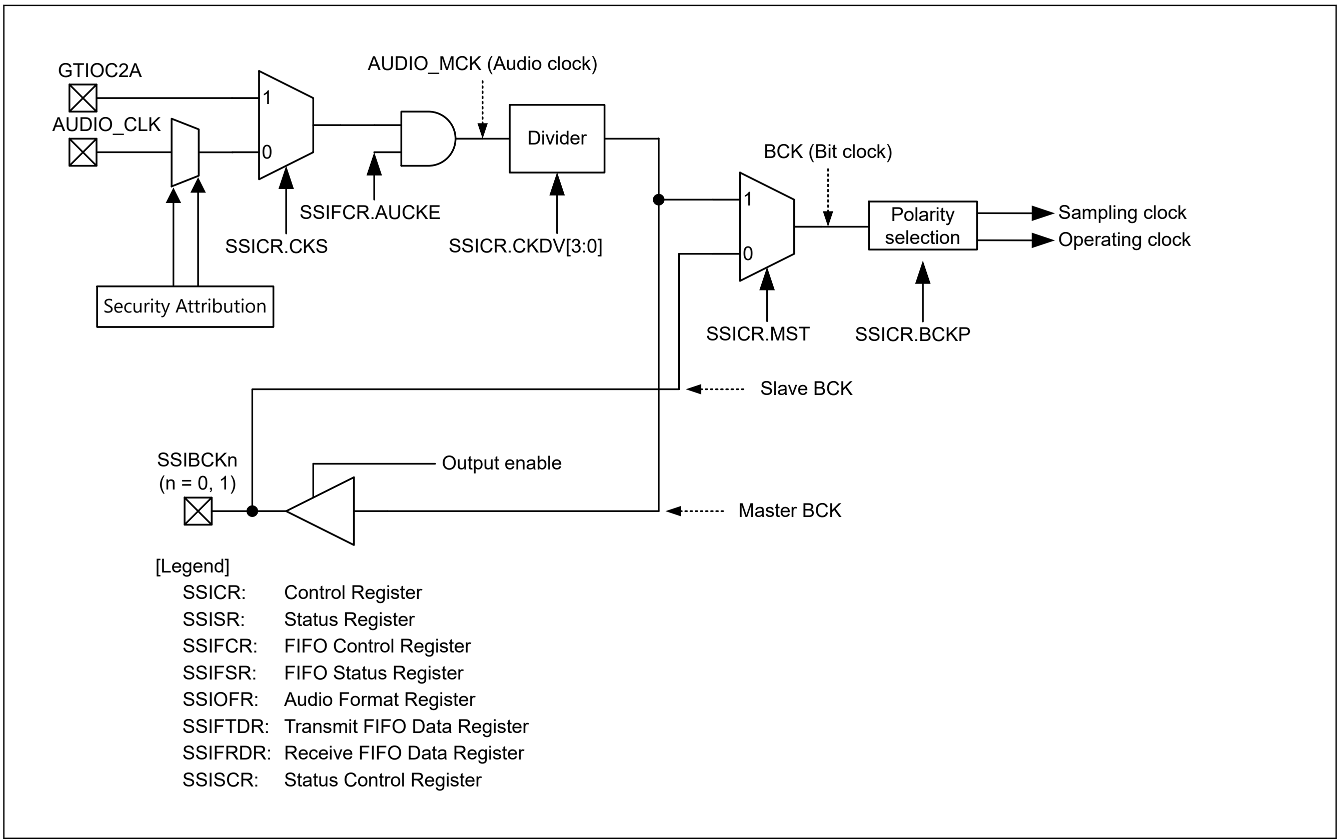


Figure 46.4 SSIE clock configuration

The AUDIO_CLK can be input according to the security attribution of Port and SSIE and SSICR.CKS bit setting as shown in the Table 46.3.

Table 46.3 The input condition of AUDIO_CLK by the security setting

PORT Security Attribution (PmSAR (m = 0 - 9, A, B))	SSIE0 Security Attribution (PSARm (B - E))	SSIE0's SSICR.CKS	SSIE1 Security Attribution (PSARm (B - E))	SSIE1's SSICR.CKS	AUDIO_CLK pin input
0 (secure)	0 (secure)	0 (AUDIO_CLK)	Don't care		AUDIO_CLK input enable
	Don't care		0 (secure)	0 (AUDIO_CLK)	
1 (non-secure)	1 (non-secure)	Don't care	0 (secure)	1 (GTIOC2A)	
			1 (non-secure)	Don't care	
	0 (secure)	1 (GTIOC2A)	1 (non-secure)	Don't care	
	1 (non-secure)	Don't care			
other than above					AUDIO_CLK input disable

Table 46.4 lists the I/O pins.

Table 46.4 SSIE I/O pins

Channel	Pin name	I/O	Function
Common	SSIBLKn	Input	Serial bit clock pins
	AUDIO_CLK	Input	External clock pin for audio (input oversampling clock)
	SSILRCKn/SSIFSn	Output	LR clock/frame synchronization pins
SSIE0	SSIRXD0	Input	Serial data input pin
	SSITXD0	Output	Serial data output pin
SSIE1	SSIDATA1	I/O	Serial data input/output pin

Note: n = 0, 1

46.2 Register Descriptions

46.2.1 SSICR : Control Register

Base address: SSIEn = 0x4025_D000 + 0x0100 × n (n = 0, 1)
SSIEn_NS = 0x5025_D000 + 0x0100 × n (n = 0, 1)

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	—	FRM[1:0]		DWL[2:0]			SWL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	MST	BCKP	LRCKP	SPDP	SDTA	PDTA	DEL	CKDV[3:0]				MUEN	—	TEN	REN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	REN	Reception Enable*2 0: Disables reception 1: Enables reception (starts reception)	R/W
1	TEN	Transmission Enable*2 0: Disables transmission 1: Enables transmission (starts transmission)	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	MUEN	Mute Enable 0: Disables muting on the next frame boundary 1: Enables muting on the next frame boundary	R/W
7:4	CKDV[3:0]	Selects Bit Clock Division Ratio*1 0x0: AUDIO_MCK 0x1: AUDIO_MCK/2 0x2: AUDIO_MCK/4 0x3: AUDIO_MCK/8 0x4: AUDIO_MCK/16 0x5: AUDIO_MCK/32 0x6: AUDIO_MCK/64 0x7: AUDIO_MCK/128 0x8: AUDIO_MCK/6 0x9: AUDIO_MCK/12 0xA: AUDIO_MCK/24 0xB: AUDIO_MCK/48 0xC: AUDIO_MCK/96 Others: Setting prohibited	R/W
8	DEL	Selects Serial Data Delay*1 In the monaural format, this bit controls the waveform of SSILRCKn/SSIFSn (n = 0, 1). For details, see section 46.3.2. Monaural Format. 0: Delay of 1 cycle of SSIBCKn (n = 0, 1) between SSILRCKn/SSIFSn (n = 0, 1) and SSITXD0/SSIRXD0/SSIDATA1 1: No delay between SSILRCKn/SSIFSn (n = 0, 1) and SSITXD0/SSIRXD0/SSIDATA1	R/W
9	PDTA	Selects Placement Data Alignment*1 0: Left-justifies placement data (SSIFTDR, SSIFRDR) 1: Right-justifies placement data (SSIFTDR, SSIFRDR)	R/W
10	SDTA	Selects Serial Data Alignment*1 0: Transmits and receives serial data first and then padding bits 1: Transmit and receives padding bits first and then serial data	R/W

<table>
<tr><th>Bit</th><th>Symbol</th><th>Function</th><th>R/W</th></tr>
<tr><td>11</td><td>SPDP</td><td>Selects Serial Padding Polarity*1
0: Padding data is at a low level
1: Padding data is at a high level</td><td>R/W</td></tr>
<tr><td>12</td><td>LRCKP</td><td>Selects the Initial Value and Polarity of LR Clock/Frame Synchronization Signal*1
0: The initial value is at a high level.
The start trigger for a frame is synchronized with a falling edge of SSILRCKn/ SSIFSn (n = 0, 1).
1: The initial value is at a low level.
The start trigger for a frame is synchronized with a rising edge of SSILRCKn/ SSIFSn (n = 0, 1).</td><td>R/W</td></tr>
<tr><td>13</td><td>BCKP</td><td>Selects Bit Clock Polarity*1
0: SSILRCKn/SSIFSn (n = 0, 1) and SSITXD0/SSIRXD0/SSIDATA1 change at a falling edge (SSILRCKn/SSIFSn (n = 0, 1) and SSIRXD0/SSIDATA1 are sampled at a rising edge of SSIBCKn (n = 0, 1)).
1: SSILRCKn/SSIFSn (n = 0, 1) and SSITXD0/SSIRXD0/SSIDATA1 change at a rising edge (SSILRCKn/SSIFSn (n = 0, 1) and SSIRXD0/SSIDATA1 are sampled at a falling edge of SSIBCKn (n = 0, 1)).</td><td>R/W</td></tr>
<tr><td>14</td><td>MST</td><td>Master Enable*1
0: Slave-mode communication
1: Master-mode communication</td><td>R/W</td></tr>
<tr><td>15</td><td>—</td><td>This bit is read as 0. The write value should be 0.</td><td>R/W</td></tr>
<tr><td>18:16</td><td>SWL[2:0]</td><td>Selects System Word Length*1
0 0 0: 8 bits
0 0 1: 16 bits
0 1 0: 24 bits
0 1 1: 32 bits
1 0 0: 48 bits
1 0 1: 64 bits
1 1 0: 128 bits
1 1 1: 256 bits</td><td>R/W</td></tr>
<tr><td>21:19</td><td>DWL[2:0]</td><td>Selects Data Word Length*1
0 0 0: 8 bits
0 0 1: 16 bits
0 1 0: 18 bits
0 1 1: 20 bits
1 0 0: 22 bits
1 0 1: 24 bits
1 1 0: 32 bits
1 1 1: Setting prohibited</td><td>R/W</td></tr>
<tr><td>23:22</td><td>FRM[1:0]</td><td>Selects Frame Word Number*1
<table>
<tr><th></th><th colspan="3">Communication format (SSIOFR.OMOD[1:0])</th></tr>
<tr><td>FRM[1:0]</td><td>I^{2}S (00b)</td><td>Monaural (10b)</td><td>TDM (01b)</td></tr>
<tr><td>00b</td><td>2</td><td>1</td><td>Setting prohibited</td></tr>
<tr><td>01b</td><td rowspan="3">Setting prohibited</td><td rowspan="3">Setting prohibited</td><td>4</td></tr>
<tr><td>10b</td><td>6</td></tr>
<tr><td>11b</td><td>8</td></tr>
</table></td><td>R/W</td></tr>
<tr><td>24</td><td>—</td><td>This bit is read as 0. The write value should be 0.</td><td>R/W</td></tr>
<tr><td>25</td><td>IIEN</td><td>Idle Mode Interrupt Output Enable
0: Disables idle mode interrupt output
1: Enables idle mode interrupt output</td><td>R/W</td></tr>
<tr><td>26</td><td>ROIEN</td><td>Receive Overflow Interrupt Output Enable
0: Disables receive overflow interrupt output
1: Enables receive overflow interrupt output</td><td>R/W</td></tr>
</table>

Bit	Symbol	Function	R/W
27	RUIEN	Receive Underflow Interrupt Output Enable 0: Disables receive underflow interrupt output 1: Enables receive underflow interrupt output	R/W
28	TOIEN	Transmit Overflow Interrupt Output Enable 0: Disables transmit overflow interrupt output 1: Enables transmit overflow interrupt output	R/W
29	TUIEN	Transmit Underflow Interrupt Output Enable 0: Disables transmit underflow interrupt output 1: Enables transmit underflow interrupt output	R/W
30	CKS	Selects an Audio Clock for Master-mode Communication*1 0: Selects the AUDIO_CLK input 1: Selects the GTIOC2A (GPT output)	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Writing to these bits while SSIE is in a communication state (SSISR.IIRQ = 0) is prohibited. If the value of these bits is changed by rewriting, subsequent operation is unpredictable.

Note 2. If the TEN bit or REN bit is rewritten, make sure that the SSISR.IIRQ bit is in the desired status. If the value of the TEN or REN bit is changed by rewriting, subsequent operation is unpredictable. For example, when transmission or reception is enabled, check that SSISR.IIRQ is 0; when transmission or reception is disabled, check that SSISR.IIRQ is 1.

With this register, select an audio clock, control interrupt requests, select data formats, and set an operation mode.

TEN and REN bits (Transmission and Reception Enable)

The TEN and REN bits enable/disable transmission and reception. When 1 is written to one of these bits, the corresponding communication operation starts in synchronization with a start trigger by the SSILRCKn/SSIFSn (n = 0, 1) signal. For details, see section 46.6.2. Transmission to section 46.6.4. Transmission and Reception. When 0 is written to this bit, the current communication operation stops at the next frame boundary. To use SSIE for both transmission and reception, always write 1 to these bits together. When stopping the communication using SSIE, always disable both transmission and reception (write 0 to the TEN and REN bits).

If you want to stop SSIE before a frame boundary is reached, perform a software reset procedure.

MUEN bit (Mute Enable)

The MUEN bit sets/clears the mute function for the data output from the SSITXD0/SSIDATA1 pin. When this bit is set to 1 in the middle of a frame, the SSITXD0/SSIDATA1 output changes to 0 at the next frame boundary. When this bit is set to 0 in the middle of a frame, the SSITXD0/SSIDATA1 output changes to the data of transmit FIFO data register at the next frame boundary. Note that this bit controls data only. Status flags and interrupt signals are normally generated.

Changing the value of this bit must be performed only after setting the communication format to be used.

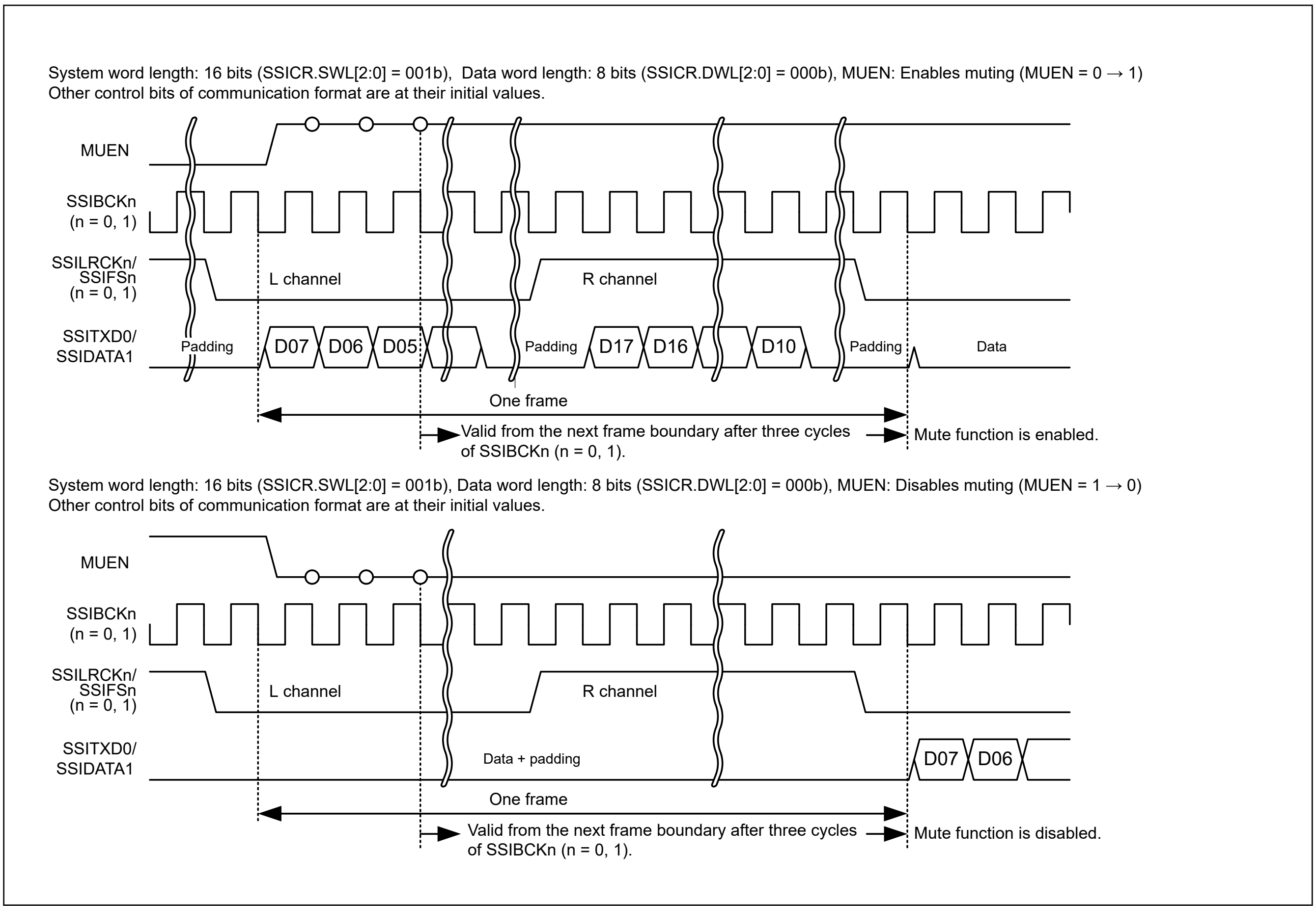


Figure 46.5 Transmit data with the mute function set

CKDV[3:0] bits (Selects Bit Clock Division Ratio)

The CKDV[3:0] bits set the division ratio of the bit clock based on AUDIO_MCK in master-mode communication (MST= 1). In slave-mode communication (MST = 0), setting of these bits are invalid.

Writing to this bit must be performed when the supply of AUDIO_MCK is stopped. For details about the timing, see the detailed description of the AUCKE bit in section 46.2.3. SSIFCR : FIFO Control Register.

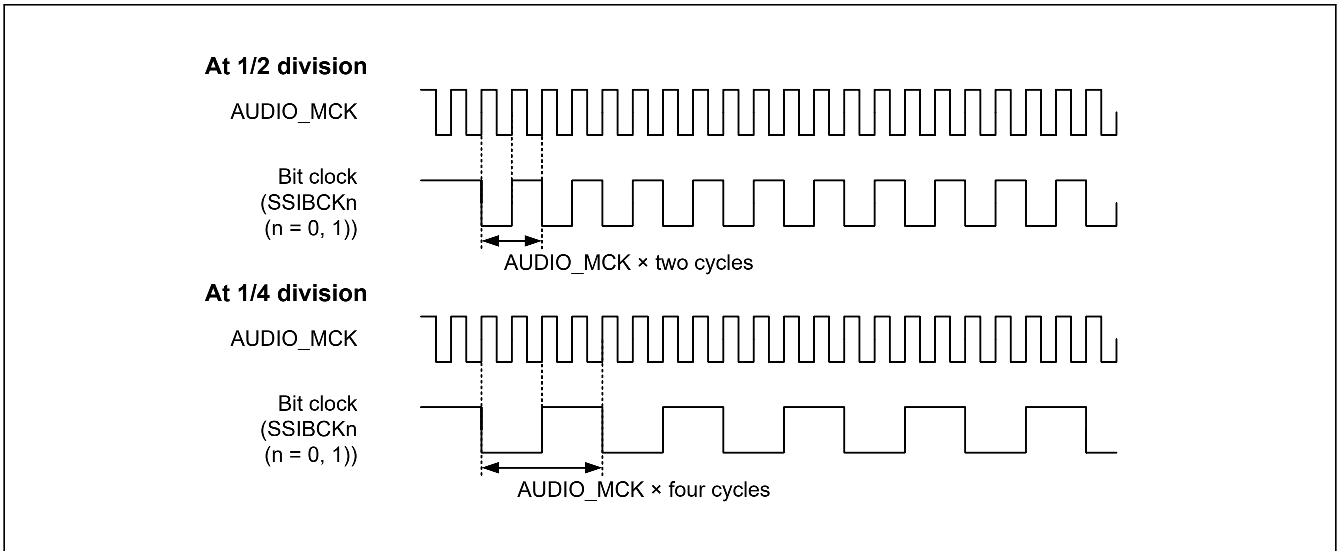


Figure 46.6 Sampling frequencies in master-mode communication

DEL bit (Selects Serial Data Delay)

The DEL bit sets whether or not there will be a delay between SSILRCKn/SSIFSn (n = 0, 1) and SSITXD0/SSIRXD0/ SSIDATA1.

For the I^{2}S or TDM format, set the DEL bit to 0. When the monaural format is used, setting of this bit changes the high period width of SSILRCKn/SSIFSn (n = 0, 1). For details, see section 46.3.2. Monaural Format. When using a compatible communication format, specify a setting of this bit that enables communication.

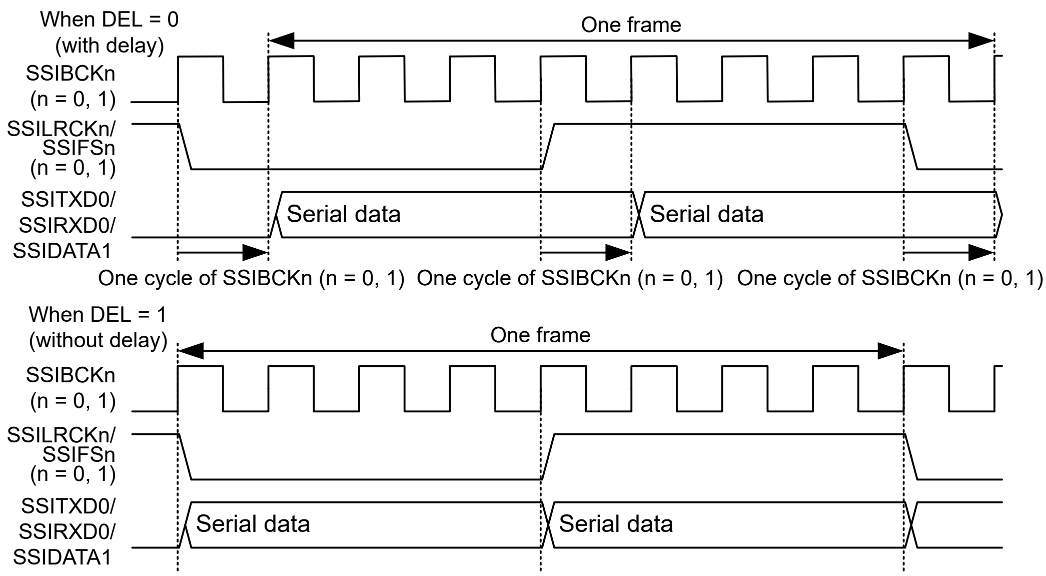


Figure 46.7 Setting of delay in serial data

PDTA bit (Selects Placement Data Alignment)

The PDTA bit sets how to align placement data. With the setting of data word length as 32 bits (SSICR.DWL[2:0] = 110b), this bit is invalid.

At transmission, see Figure 46.8.

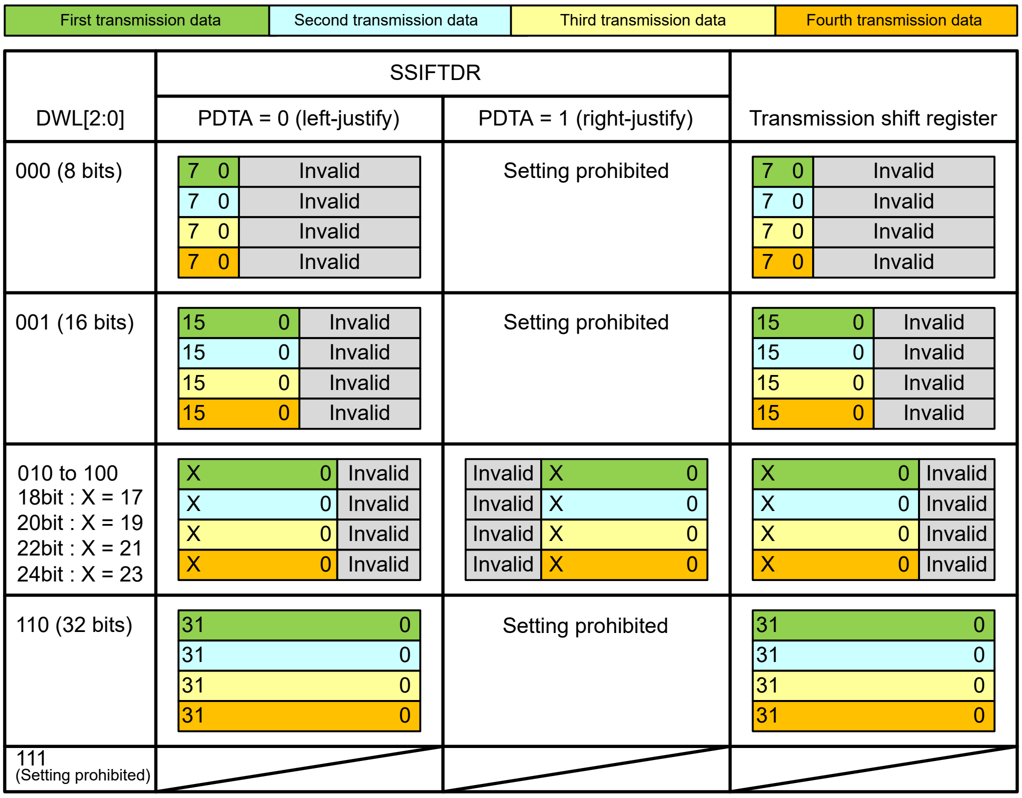


Figure 46.8 Alignment of placement data at transmission

At reception, see Figure 46.9.

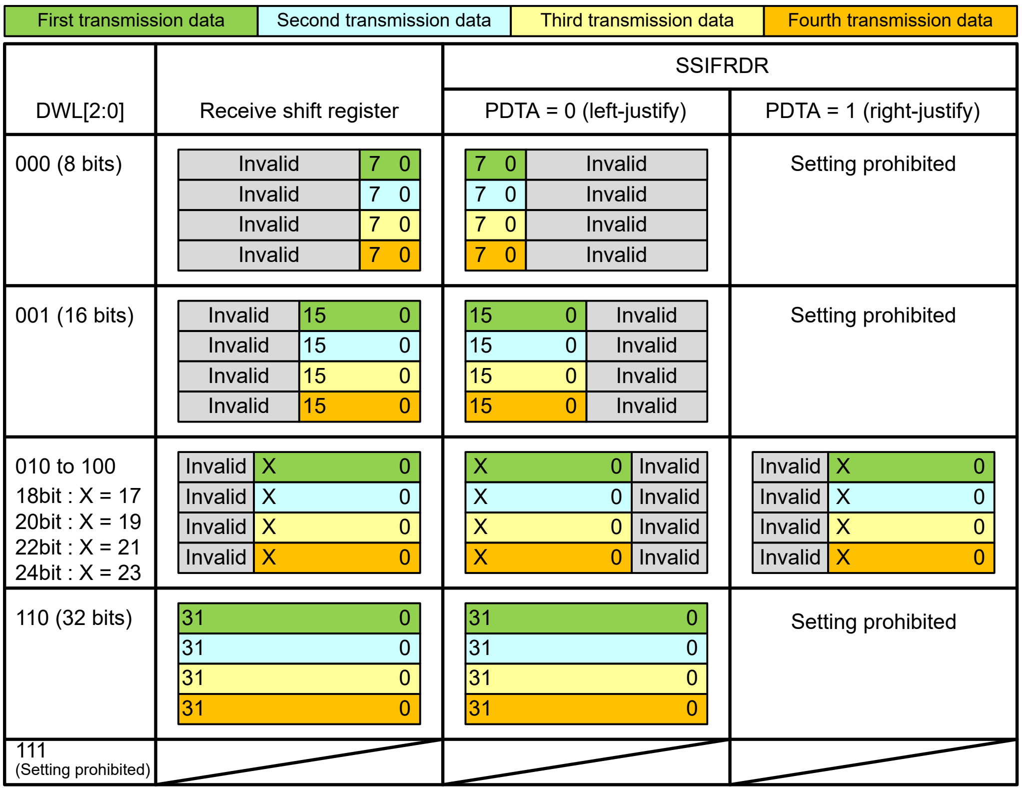


Figure 46.9 Alignment of placement data at reception

SDTA bit (Selects Serial Data Delay)

The SDTA bit sets how to align serial data and padding bits. For communication without padding bits, this bit is invalid.

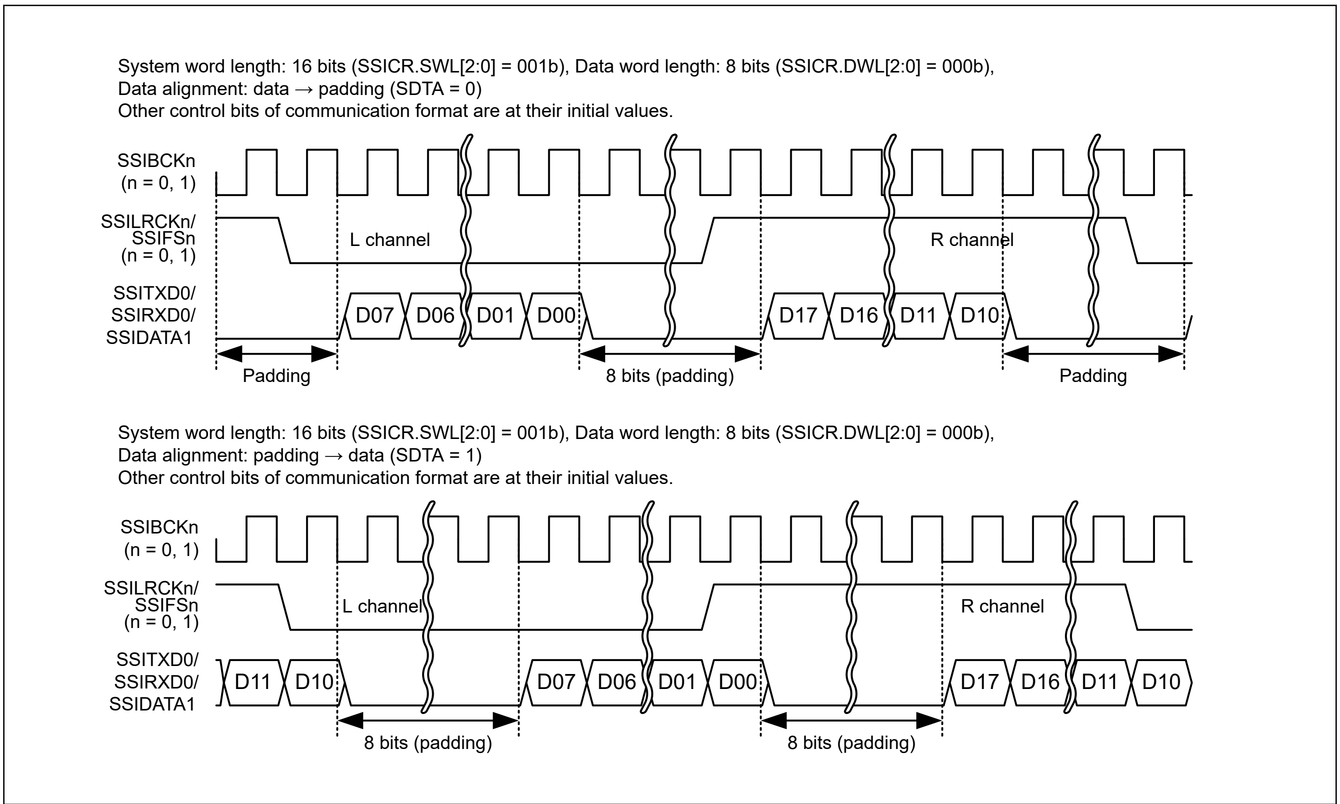


Figure 46.10 Alignment setting of serial data with padding bits

SPDP bit (Selects Serial Padding Polarity)

The SPDP bit sets polarity of padding bits.

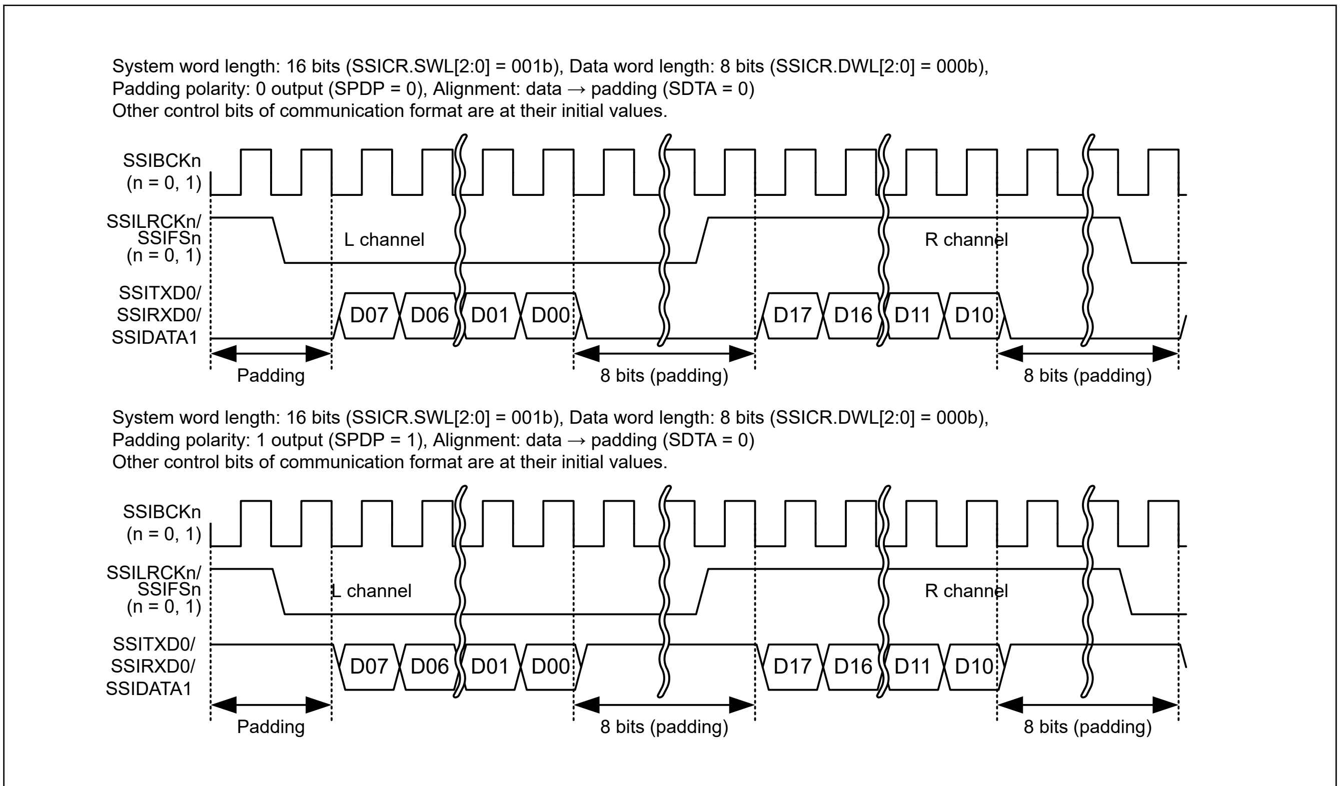


Figure 46.11 Padding bit polarity

LRCKP bit (Selects the Initial Value and Polarity of LR Clock/Frame Synchronization Signal)

The LRCKP bit sets the initial value and polarity of SSILRCKn/SSIFSn (n = 0, 1). Set this bit according to the communication format to be used in SSIE. See Table 46.5 Initial output value and polarity of SSILRCKn/SSIFSn (n = 0, 1) pin. For the slave-mode communication (MST = 0), only the start trigger is used.

Writing to these bits must be performed when the LR clock supply to the SSILRCKn/SSIFSn (n = 0, 1) pin is stopped. For details about the output of LR clock, see the detailed description of the LRCONT bit in section 46.2.7. SSIOFR : Audio Format Register.

Table 46.5 Initial output value and polarity of SSILRCKn/SSIFSn (n = 0, 1) pin

Communication Format	Expected Initial State	Setting Value of LRCKP
I^2S	High	0
Monaural	Low	1
TDM	Low	1

Note: When the format to be used is compatible with the I^2S, monaural, or TDM format, specify settings to enable communication with the respective formats.

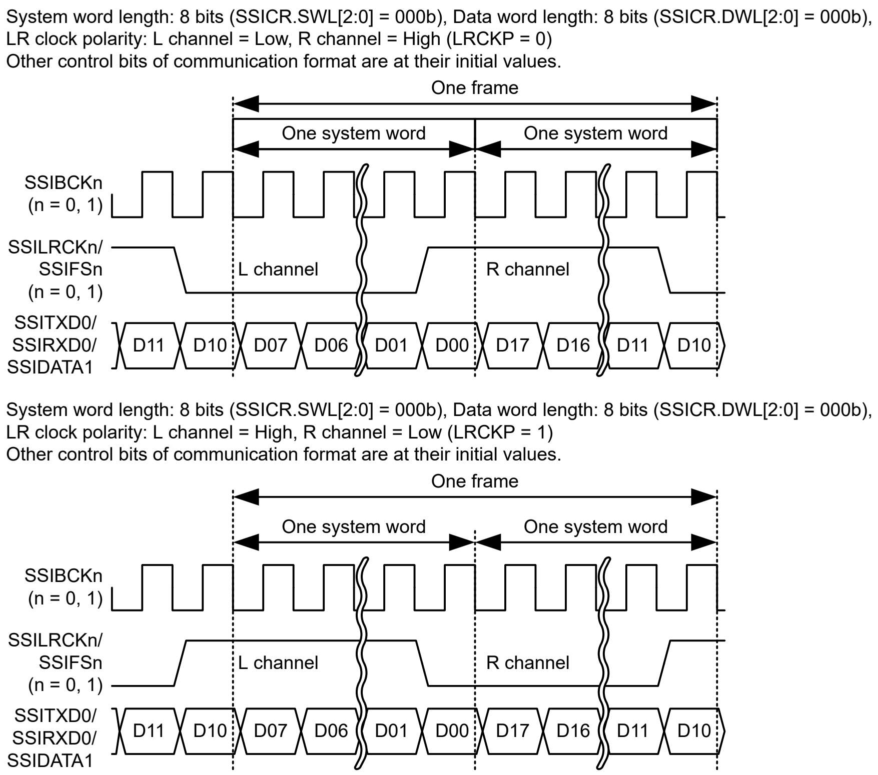


Figure 46.12 LR clock/frame synchronization polarity setting

BCKP bit (Selects Bit Clock Polarity)

The BCKP bit sets the bit clock polarity.

Writing to this bit must be performed when the supply of AUDIO_MCK is stopped. For details about the timing, see the detailed description of the AUCKE bit in section 46.2.3. SSIFCR : FIFO Control Register.

Table 46.6 Bit clock polarity

Communication	Master/Slave	Timing	BCKP = 0	BCKP = 1
Reception	Slave	At SSILRCKn/SSIFSn (n = 0, 1) sampling	SSIBCKn (n = 0, 1) rising edge	SSIBCKn (n = 0, 1) falling edge
	Master/slave	At SSIRXD0/SSIDATA1 sampling	SSIBCKn (n = 0, 1) rising edge	SSIBCKn (n = 0, 1) falling edge
Transmission	Master	At change of SSILRCKn/SSIFSn (n = 0, 1) output	SSIBCKn (n = 0, 1) falling edge	SSIBCKn (n = 0, 1) rising edge
	Master/slave	At change of SSITXD0/SSIDATA1 output	SSIBCKn (n = 0, 1) falling edge	SSIBCKn (n = 0, 1) rising edge

MST bit (Master Enable)

The MST bit sets master-/slave-mode communication.

Writing to this bit must be performed when the supply of AUDIO_MCK is stopped. For details about the timing, see the detailed description of the AUCKE bit in section 46.2.3. SSIFCR : FIFO Control Register.

SWL[2:0] bits (Selects System Word Length)

The SWL[2:0] bits set the number of bits in one system word. Padding bits are sent and received in relation with one data word set with DWL[2:0]. See Table 46.13 for details.

Writing to these bits must be performed when the LR clock supply to the SSILRCKn/SSIFSn (n = 0, 1) pin is stopped. For details about the output of LR clock, see the detailed description of the LRCONT bit in section 46.2.7. SSIOFR : Audio Format Register.

DWL[2:0] bits (Selects Data Word Length)

The DWL[2:0] bits set the number of bits in one data word. The data word length (number of bits per data word) must not exceed the system word length (number of bits per system word). For details, see Table 46.13.

FRM[1:0] bits (Selects Frame Word Number)

The FRM[1:0] bits set the frame word number in individual communication formats.

Writing to these bits must be performed when the LR clock supply to the SSILRCKn/SSIFSn (n = 0, 1) pin is stopped. For details about the output of LR clock, see the detailed description of the LRCONT bit in section 46.2.7. SSIOFR : Audio Format Register.

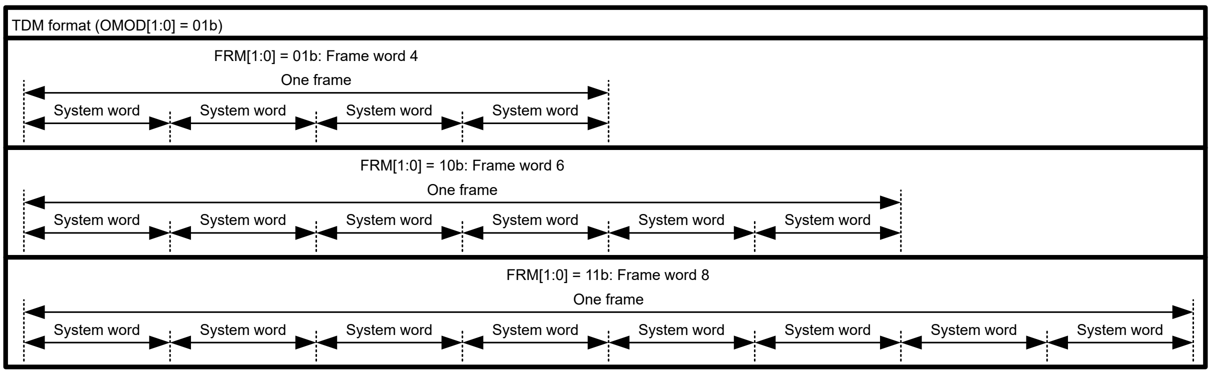


Figure 46.13 Frame word number

IIEN bit (Idle Mode Interrupt Output Enable)

The IIEN bit enables/disables output of idle mode interrupts. By enabling this bit (set it to 1), an interrupt is output at a rising edge of SSISR.IIRQ = 1. An interrupt is also output when this bit is changed from 0 to 1 while SSISR.IIRQ = 1.

ROIEN bit (Receive Overflow Interrupt Output Enable)

The ROIEN bit enables/disables output of receive overflow interrupts. By enabling this bit (set it to 1), an interrupt is output at a rising edge of SSISR.ROIRQ = 1. An interrupt is also output when this bit is changed from 0 to 1 while SSISR.ROIRQ = 1.

RUIEN bit (Receive Underflow Interrupt Output Enable)

The RUIEN bit enables/disables output of receive underflow interrupts. By enabling this bit (set it to 1), an interrupt is output at a rising edge of SSISR.RUIRQ = 1. An interrupt is also output when this bit is changed from 0 to 1 while SSISR.RUIRQ = 1.

TOIEN bit (Transmit Overflow Interrupt Output Enable)

The TOIEN bit enables/disables output of transmit overflow interrupts. By enabling this bit (set it to 1), an interrupt is output at a rising edge of SSISR.TOIRQ = 1. An interrupt is also output when this bit is changed from 0 to 1 while SSISR.TOIRQ = 1.

TUIEN bit (Transmit Underflow Interrupt Output Enable)

The TUIEN bit enables/disables output of transmit underflow interrupts. By enabling this bit (set it to 1), an interrupt is output at a rising edge of SSISR.TUIRQ = 1. An interrupt is also output when this bit is changed from 0 to 1 while SSISR.TUIRQ = 1.

CKS bit (Selects an Audio Clock for Master-mode Communication)

The CKS bit sets the audio clock in master-mode communication (MST = 1). In slave-mode communication (MST = 0), setting of this bit is invalid.

Writing to this bit must be performed when the supply of AUDIO_MCK is stopped. For details about the timing, see the detailed description of the AUCKE bit in section 46.2.3. SSIFCR : FIFO Control Register.

46.2.2 SSISR : Status Register

Base address: SSIEn = 0x4025_D000 + 0x0100 × n (n = 0, 1)
SSIEn_NS = 0x5025_D000 + 0x0100 × n (n = 0, 1)

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
24:0	—	These bits are read as 0. The write value should be 0.	R/W
25	IIRQ	Idle Mode Status Flag 0: In the communication state 1: In the idle state	R
26	ROIRQ	Receive Overflow Error Status Flag 0: No receive overflow error is generated. 1: A receive overflow error is generated.	R/W
27	RUIRQ	Receive Underflow Error Status Flag 0: No receive underflow error is generated. 1: A receive underflow error is generated.	R/W
28	TOIRQ	Transmit Overflow Error Status Flag 0: No transmit overflow error is generated. 1: A transmit overflow error is generated.	R/W

Bit	Symbol	Function	R/W
29	TUIRQ	Transmit Underflow Error Status flag 0: No transmit underflow error is generated. 1: A transmit underflow error is generated.	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register is configured with status flags that indicate SSIE operational state.

IIRQ flag (Idle Mode Status Flag)

The IIRQ flag is a status flag that indicates the idle state. It indicates whether SSIE is in the idle state or communication state.

For details, see Figure 46.14 and Figure 46.15.

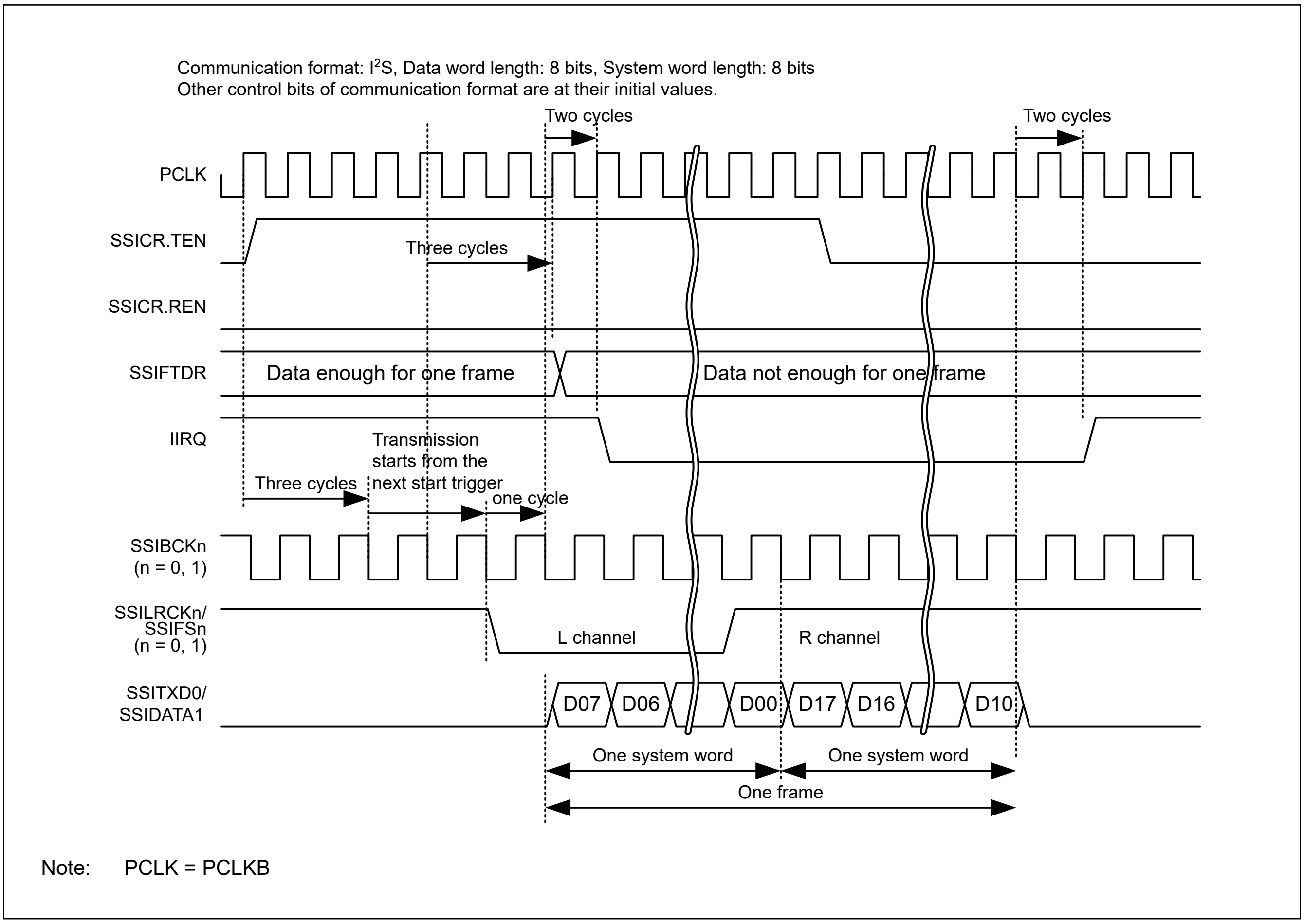


Figure 46.14 IIRQ setting timing (transmission)

- Transmitter (dedicated to transmission)

[Clearing condition]

While transmission was enabled (SSICR.TEN = 1 and SSICR.REN = 0), the transmit data for a transmission frame was written to the SSIFTDR register, and a start trigger was generated by the SSILRCKn/SSIFSn (n = 0, 1) signal.

[Clearing timing]

1 SSIBCKn (n = 0, 1) cycle + 2 PCLKB cycles after generation of the start trigger that is the clearing condition.

[Setting condition]

While transmission and reception were disabled (SSICR.TEN = 0 and SSICR.REN = 0), transmission of one frame was complete.

[Setting timing]

2 PCLKB cycles after the end of transmission (at a frame boundary) that is the setting condition.

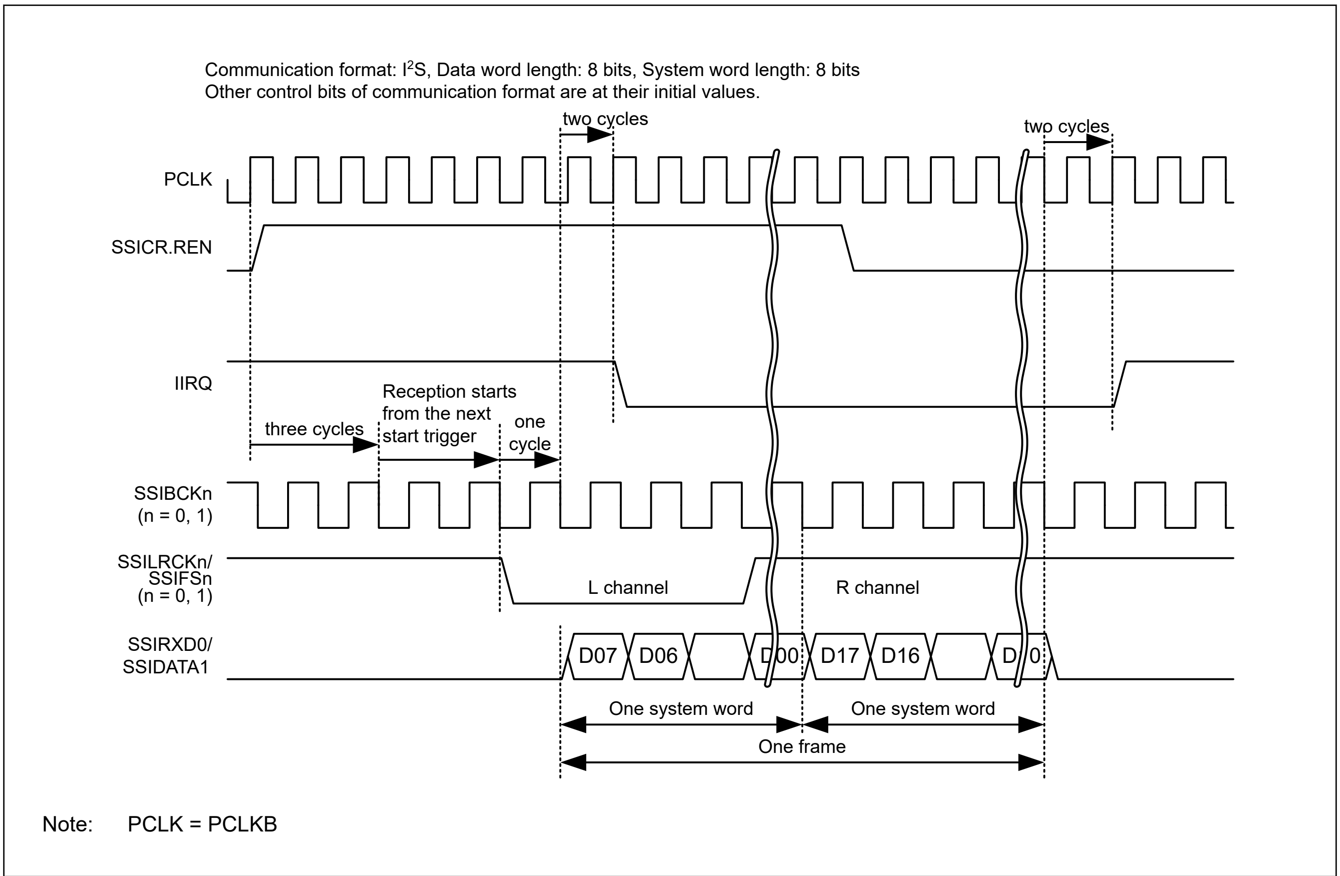


Figure 46.15 IIRQ setting timing (reception)

- Receiver (dedicated to reception)

[Clearing condition]

While reception was enabled (SSICR.TEN = 0 and SSICR.REN = 1, a start trigger was generated by the SSILRCKn/ SSIFSn (n = 0, 1) signal.

[Clearing timing]

1 SSIBCKn (n = 0, 1) cycle + 2 PCLKB cycles after generation of the start trigger that is the clearing condition.

[Setting condition]

While transmission and reception were disabled (SSICR.TEN = 0 and SSICR.REN = 0), reception of one frame was complete.

[Setting timing]

2 PCLKB cycles after the end of reception (at a frame boundary) that is the setting condition.

- Transceiver (transmission and reception)

[Clearing condition]

While transmission and reception were enabled (SSICR.TEN = 1 and SSICR.REN = 1), the transmit data for a transmission frame was written to the SSIFTDR register, and a start trigger is generated by the SSILRCKn/SSIFSn (n = 0, 1) signal.

[Clearing timing]

1 SSIBCKn (n = 0, 1) cycle + 2 PCLKB cycles after generation of the start trigger that is the clearing condition.

[Setting condition]

While transmission and reception were disabled (SSICR.TEN = 0 and SSICR.REN = 0), transmission of one frame was complete.

[Setting timing]

2 PCLKB cycles after the end of transmission (at a frame boundary) that is the setting condition.

ROIRQ flag (Receive Overflow Error Status Flag)

The ROIRQ flag is a status flag that indicates a receive overflow error. This flags is set by automatic determination, but it must be cleared by register access. This flag indicates that received data is supplied at a higher rate than requested. Data is not transferred from the receive shift register to SSIFRDR where a receive overflow error is generated. For the procedure to recover from the overflow error, see section 46.6.6. Error Handling. This flag is not cleared by a receive FIFO data register reset (SSIFCR.RFRST).

[Priority order for setting and clearing]

Setting is prioritized.*1

[Clearing condition]

When either of the following operations is done:

1. Writing 0 to this bit after reading 1 from this bit*2
2. Enabling communication (changing SSICR.REN from 0 to 1).

[Clearing timing]

Clearing timing corresponding to the above clearing condition:

1. When 0 is written to this bit after reading 1 from this bit (same as the timing in Figure 46.19)
2. 1 PCLKB cycle after writing 1 to SSICR.REN.*3

[Setting condition]
At completion of receiving new data while SSIFRDR is full.

[Setting timing]
3 cycles of PCLKB after reception is completed.

Note 1. This bit is cleared by a software reset (SSIFCR.SSIRST = 1). The software reset has priority over all the clearing conditions described above.

Note 2. After reading 1 from this bit, this bit is cleared when one of the following three conditions is met:
- A software reset (SSIFCR.SSIRST = 1) is done.
- After 1 has been read, writing of 0 is complete.
- 1 PCLKB cycle passes after 1 has been written to SSICR.REN.

Note 3. After communication is enabled (by changing the value of SSICR.REN bit from 0 to 1), the reception error flags (RUIRQ and ROIRQ in the SSISR register) are cleared. If, however, the SSISR register is read continuously, the cleared status of the reception error flags might be unable to be read.

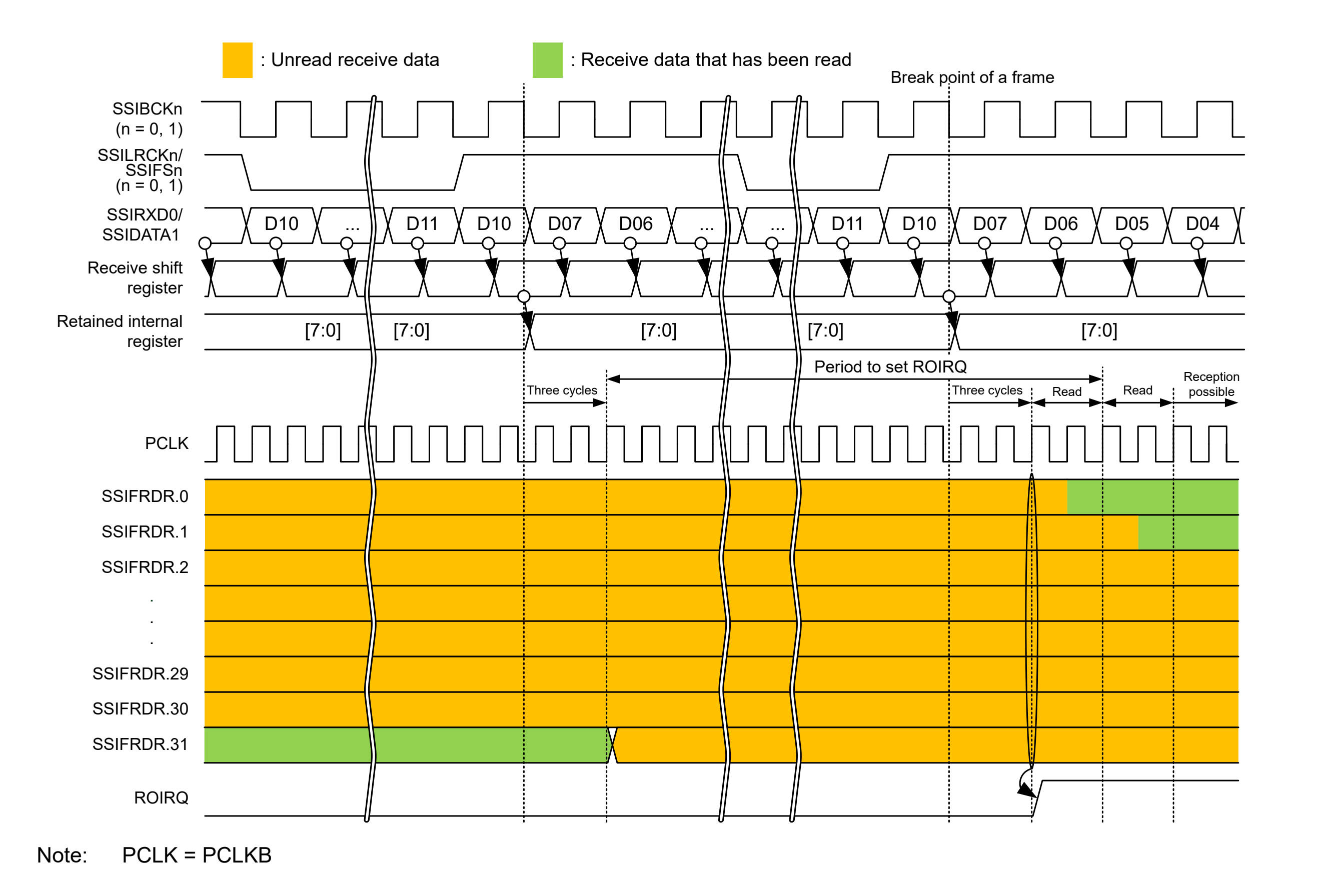


Figure 46.16 ROIRQ setting timing

RUIRQ flag (Receive Underflow Error Status Flag)

The RUIRQ flag is a status flag that indicates a receive underflow error. This flags is set by automatic determination, but it must be cleared by register access. This flag indicates that SSIFRDR is read while it is empty. Data read from SSIFRDR where a receive underflow error is generated is invalid. See section 46.6.6. Error Handling for the error recovery procedure. This flag is not cleared by a receive FIFO data register reset (SSIFCR.RFRST). Note, however, that this flag is not set even if the SSIFRDR register is read while the receive FIFO data register is reset (by setting SSIFCR.RFRST to 1).

[Priority order for setting and clearing]

Setting is prioritized.*1

[Clearing condition]

When either of the following operations is done:

1. Writing 0 to this bit after reading 1 from this bit*2
2. Enabling communication (changing SSICR.REN from 0 to 1).

[Clearing timing]
Clearing timing corresponding to the above clearing condition

1. When 0 is written to this bit after reading 1 from this bit (same as the timing in Figure 46.19)
2. 1 PCLKB cycle after writing 1 to SSICR.REN.*3

[Setting condition]

Reading from SSIFRDR while it is empty.

[Setting timing]

At completion of reading from SSIFRDR. See Figure 46.17.

Note 1. This bit is cleared by a software reset (SSIFCR.SSIRST = 1). The software reset has priority over all the clearing conditions described above.

Note 2. After reading 1 from this bit, this bit is cleared when one of the following three conditions is met:
- A software reset (SSIFCR.SSIRST = 1) is done.
- After 1 has been read, writing of 0 is complete.
- 1 PCLKB cycle passes after 1 has been written to SSICR.REN.

Note 3. After communication is enabled (by changing the value of SSICR.REN bit from 0 to 1), the reception error flags (RUIRQ and ROIRQ in the SSISR register) are cleared. If, however, the SSISR register is read continuously, the cleared status of the reception error flags might be unable to be read.

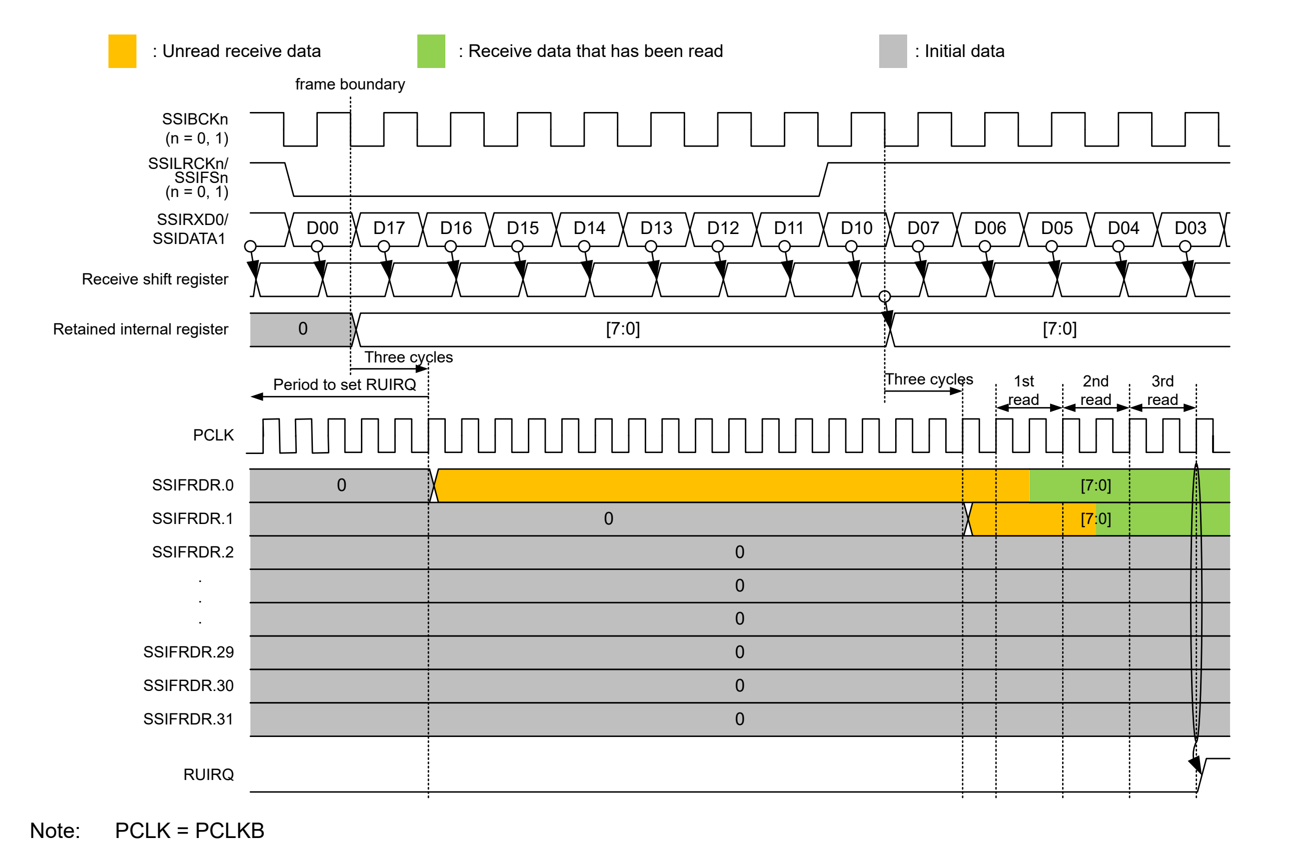


Figure 46.17 RUIRQ setting timing

TOIRQ flag (Transmit Overflow Error Status Flag)

The TOIRQ flag is a status flag that indicates a transmit overflow error. This flag is set by automatic determination, but it must be cleared by register access. This flag indicates that an attempt has been made to write data to the SSIFTDR register when the register is full of data. The data writing that causes a transmit overflow is ignored. For the procedure to recover from the overflow error, see section 46.6.6. Error Handling. This flag is not cleared by a transmit FIFO data register reset (SSIFCR.TFRST).

[Priority order for setting and clearing]

Setting is prioritized.*1

[Clearing condition]

When either of the following operations is done:

1. Writing 0 to this bit after reading 1 from this bit*2
2. Enabling communication (changing SSICR.TEN from 0 to 1).

[Clearing timing]

Clearing timing corresponding to the above clearing condition

1. When 0 is written to this bit after reading 1 from this bit (same as the timing in Figure 46.19)
2. 1 PCLKB cycle after writing 1 to SSICR.TEN.*3

[Setting condition]

An attempt was made to write data to the SSIFTDR register when the register is full of data.

[Setting timing]

At completion of writing to SSIFTDR. For details, see Figure 46.18.

Note 1. This bit is cleared by a software reset (SSIFCR.SSIRST = 1). The software reset has priority over all the clearing conditions described above.

Note 2. After reading 1 from this bit, this bit is cleared when one of the following three conditions is met:
- A software reset (SSIFCR.SSIRST = 1) is done.
- After 1 has been read, writing of 0 is complete.
- 1 PCLKB cycle passes after 1 has been written to SSICR.TEN.

Note 3. After communication is enabled (by changing the value of SSICR.TEN bit from 0 to 1), the transmission error flags (TOIRQ and TUIRQ in the SSISR register) are cleared. If, however, the SSISR register is read continuously, the cleared status of the transmission error flags might be unable to be read.

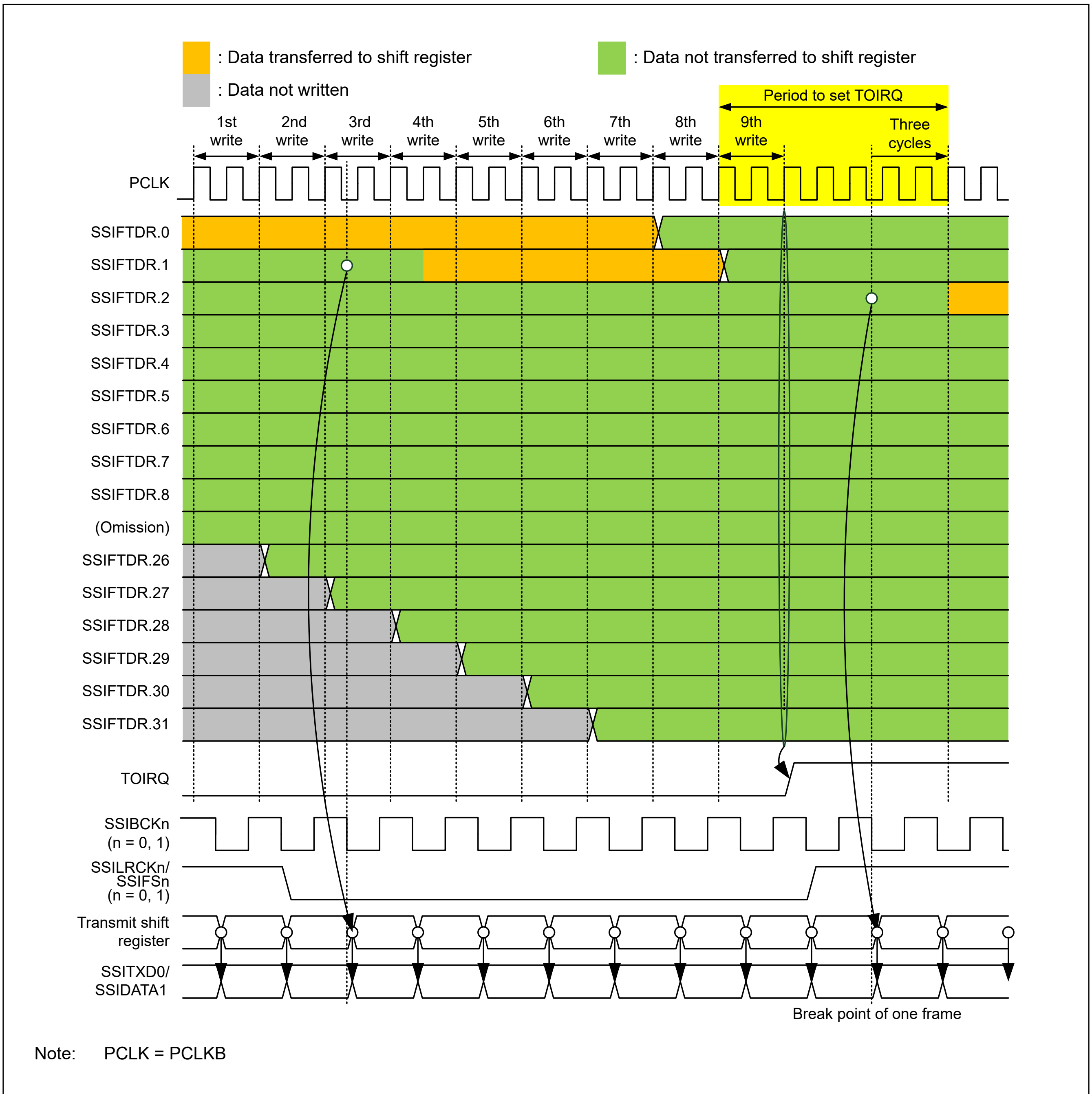


Figure 46.18 TOIRQ setting timing

TUIRQ flag (Transmit Underflow Error Status flag)

The TUIRQ flag is a status flag that indicates a transmit underflow error. This flag is set by automatic determination, but it must be cleared by register access. This flag indicates that writing the serial data required for a frame to SSIFTDR did not catch up with transmission of the frame. Even if this flag is cleared after it has been set, the SSITXD0/SSIDATA1 output remains to be 0. To output the data written to the transmit FIFO data register (SSIFTDR) to the SSITXD0/SSIDATA1 pin, follow the communication stop procedure in Figure 46.56 and error-handling procedure in Figure 46.57. For the procedure to recover from an error, see section 46.6.6. Error Handling. This flag is not cleared by a reset of transmit FIFO data register (by the SSIFCR.TFRST signal).

[Priority order for setting and clearing]

Setting is prioritized.*1

[Clearing condition]

When either of the following operations is done:

1. Writing 0 to this bit after reading 1 from this bit*2
2. Enabling communication (changing SSICR.TEN from 0 to 1).

[Clearing timing]

Clearing timing corresponding to the above clearing condition

1. When 0 is written to this bit after reading 1 from this bit
2. 1 PCLKB cycle after writing 1 to SSICR.TEN.*3

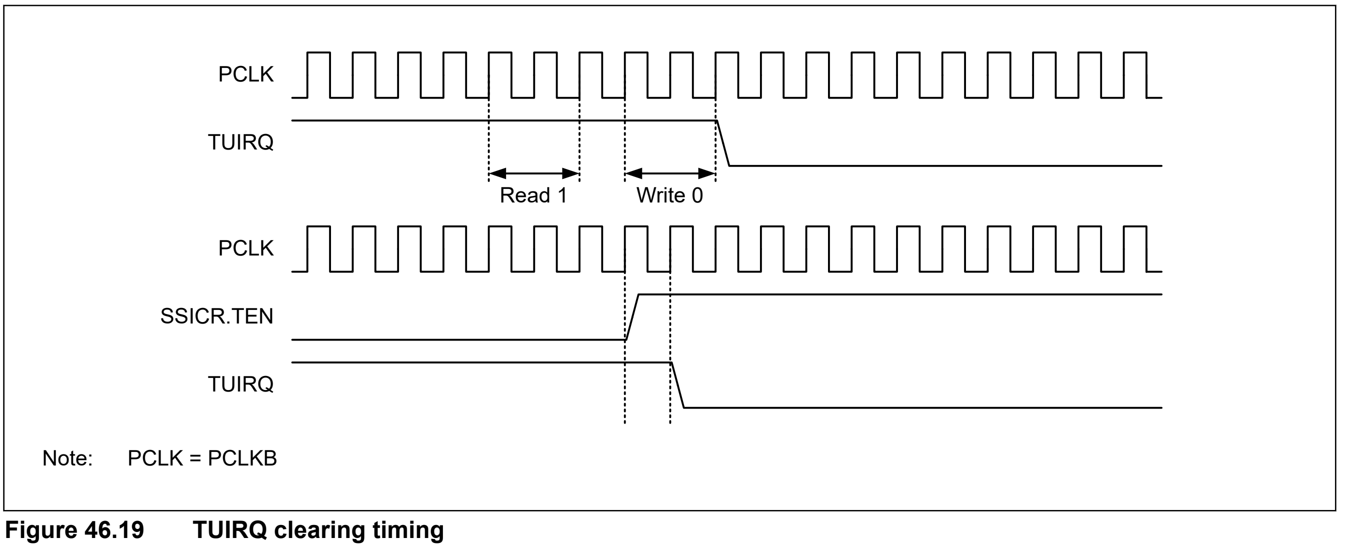


Figure 46.19 TUIRQ clearing timing

Note 1. This bit is cleared by a software reset (SSIFCR.SSIRST = 1). The software reset has priority over all the clearing conditions described above.

Note 2. After reading 1 from this bit, this bit is cleared when one of the following three conditions is met:

- A software reset (SSIFCR.SSIRST = 1) is done.
- After 1 has been read, writing of 0 is complete.
- 1 PCLKB cycle passes after 1 has been written to SSICR.TEN.

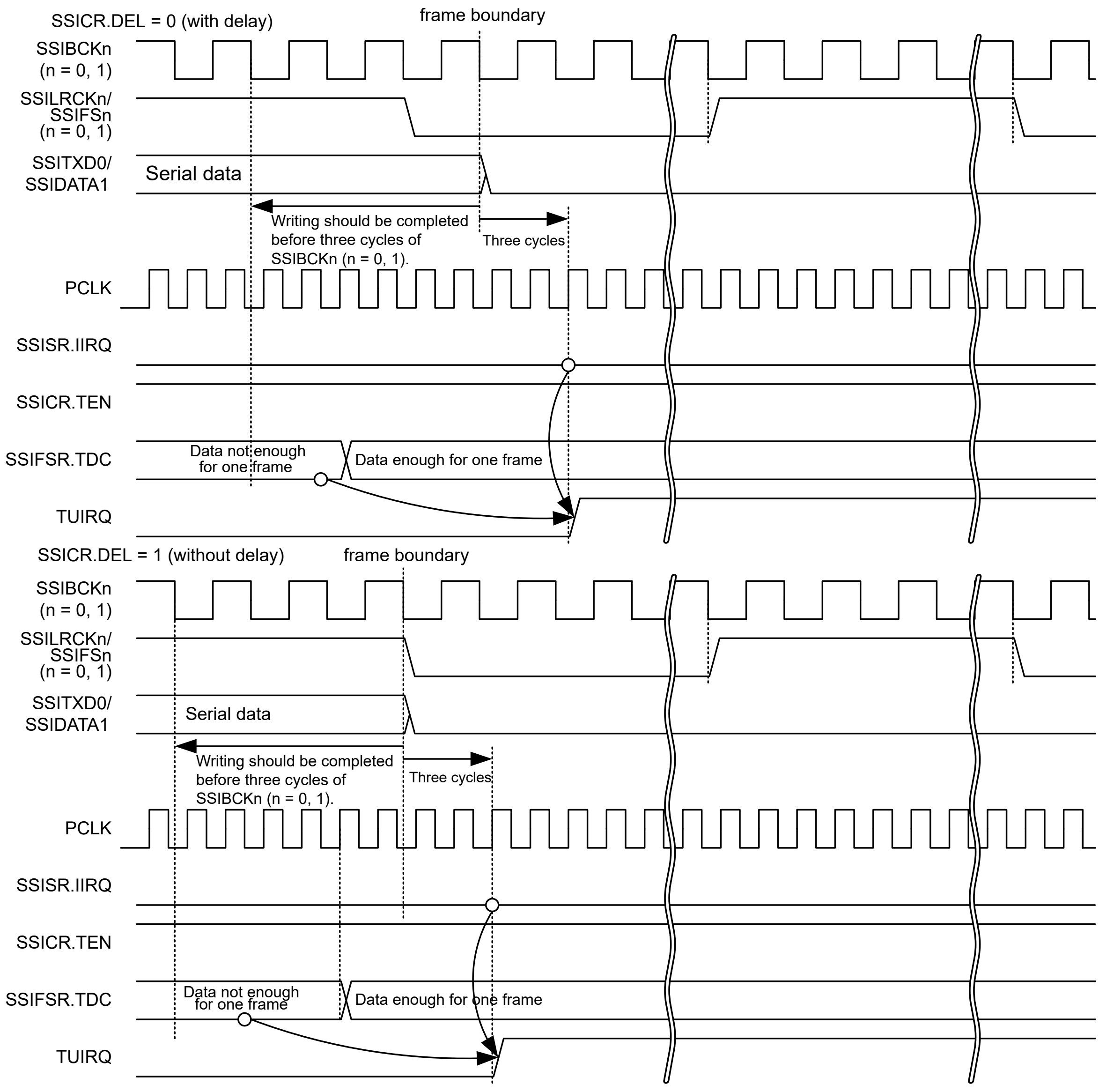
Note 3. After communication is enabled (by changing the value of SSICR.TEN bit from 0 to 1), the transmission error flags (TOIRQ and TUIRQ in the SSISR register) are cleared. If, however, the SSISR register is read continuously, the cleared status of the transmission error flags might be unable to be read.

[Setting condition]

When communication continues over a frame boundary, the transmit data required for the next frame has not been written to SSIFTDR. For details, see Figure 46.20 and Figure 46.21.

[Setting timing]

3 PCLKB cycles after the frame boundary. For details, see Figure 46.20.



Note: PCLK = PCLKB

Figure 46.20 TUIRQ setting timing (when communication continues)

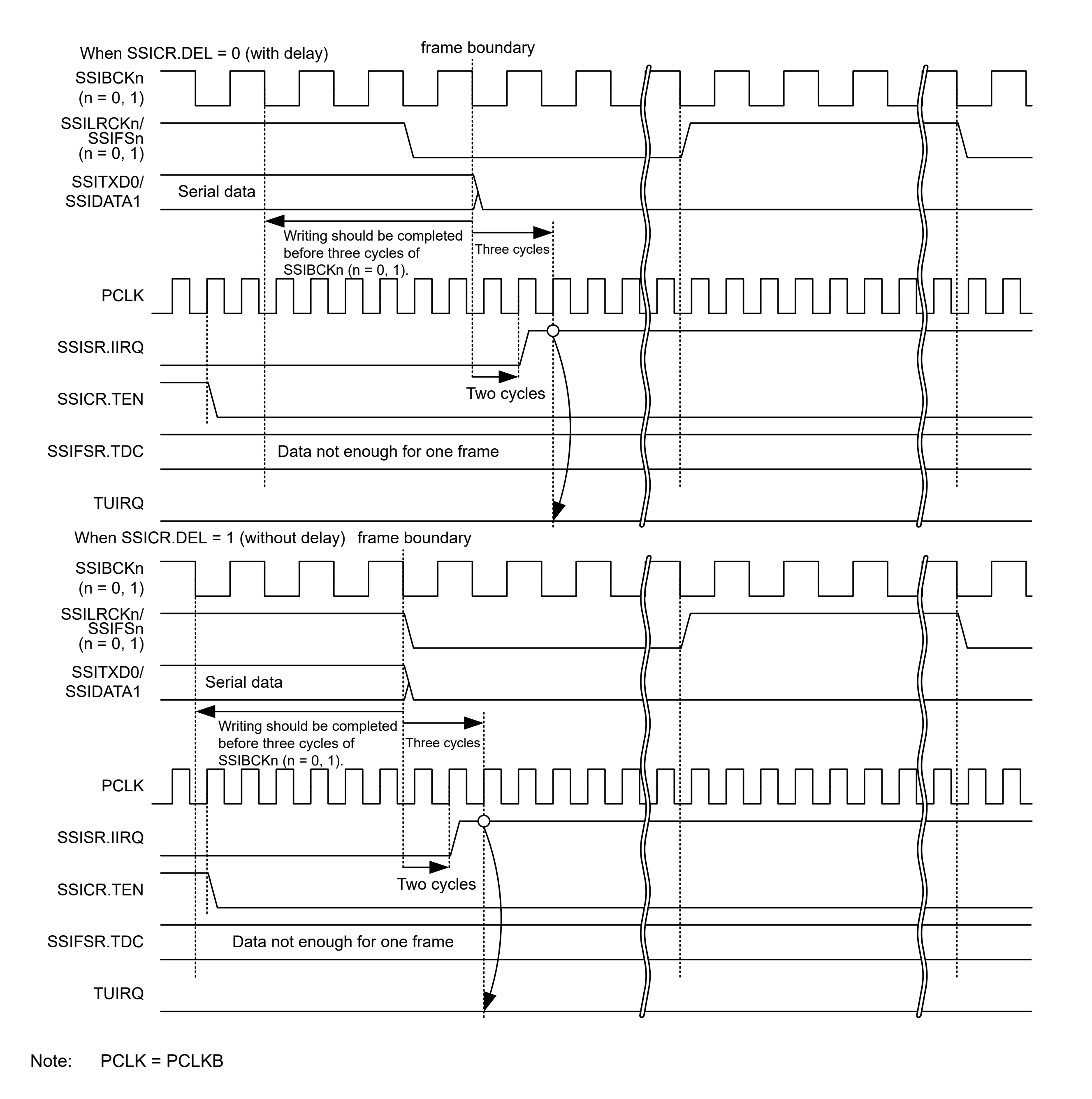


Figure 46.21 TUIRQ setting timing (when communication stops)

46.2.3 SSIFCR : FIFO Control Register

Base address: SSIEn = 0x4025_D000 + 0x0100 × n (n = 0, 1)
SSIEn_NS = 0x5025_D000 + 0x0100 × n (n = 0, 1)

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	AUCKE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSIRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	BSW	—	—	—	—	—	—	—	TIE	RIE	TFRST	RFRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFRST	Receive FIFO Data Register Reset*1 0: Clears a receive data FIFO reset condition 1: Sets a receive data FIFO reset condition	R/W
1	TFRST	Transmit FIFO Data Register Reset*1 0: Clears a transmit data FIFO reset condition 1: Sets a transmit data FIFO reset condition	R/W
2	RIE	Receive Data Full Interrupt Output Enable 0: Disables receive data full interrupts 1: Enables receive data full interrupts	R/W
3	TIE	Transmit Data Empty Interrupt Output Enable 0: Disables transmit data empty interrupts 1: Enables transmit data empty interrupts	R/W
10:4	—	These bits are read as 0. The write value should be 0.	R/W
11	BSW	Byte Swap Enable*1 0: Disables byte swap 1: Enables byte swap	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
16	SSIRST	Software Reset 0: Clears a software reset condition 1: Sets a software reset condition	R/W
30:17	—	These bits are read as 0. The write value should be 0.	R/W
31	AUCKE	AUDIO_MCK Enable in Master mode Communication*1 0: Disables supply of AUDIO_MCK 1: Enables supply of AUDIO_MCK	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Writing to these bits while SSIE is in a communication state (SSISR.IIRQ = 0) is prohibited. If the value of these bits is changed by rewriting, subsequent operation is unpredictable.

This register sets a software reset, byte swap, and enable/disable of interrupt requests.

RFRST bit (Receive FIFO Data Register Reset)

The RFRST bit sets a software reset of the receive FIFO data register (SSIFRDR). Writing 1 to this bit initializes the internal state related to SSIFRDR. The register bits subject to the software reset triggered by this bit are indicated by shading in Table 46.7. Because this bit is not automatically cleared after it has been set, write 0 to this bit to release the register bits from the software reset. After writing 0 to this bit, be sure to check that this bit is 0 before starting the next procedural step.

This bit is subject to the software reset by the SSIRST bit. Because the software reset by the SSIRST bit has priority over the reset by this bit, setting this bit is ignored when the SSIRST bit is set.

Table 46.7 Bits subject to software reset by the RFRST bit

Symbol	Address (BASE+)		+0								+1							
			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSICR	0x00	+0	—	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	—	FRM[1:0]		DWL[2:0]			SWL[2:0]		
		+2	—	MST	BCKP	LRCKP	SPDP	SDTA	PDTA	DEL	CKDV[3:0]				MUEN	—	TEN	REN
SSISR	0x04	+0	—	—	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	—	—	—	—	—	—	—	—	—
		+2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SSIFCR	0x10	+0	AUCKE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSIRST
		+2	—	—	—	—	BSW	—	—	—	—	—	—	—	TIE	RIE	TFRST	RFRST
SSIFSR	0x14	+0	—	—	TDC[5:0]						—	—	—	—	—	—	—	TDE
		+2	—	—	**RDC[5:0]**						—	—	—	—	—	—	—	**RDF**
SSIFTDR	0x18	+0	FTDR[31:16]															
		+2	FTDR[15:0]															
SSIFRDR	0x1C	+0	FRDR[31:16]															
		+2	FRDR[15:0]															
SSIOFR	0x20	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	—	—	—	BCKASTP	LRCONT	—	—	—	—	—	—	OMOD[1:0]	
SSISCR	0x24	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	TDES[4:0]					—	—	—	RDFS[4:0]				

TFRST bit (Transmit FIFO Data Register Reset)

The TFRST bit sets a software reset of the transmit FIFO data register (SSIFTDR). Writing 1 to this bit initializes the internal state related to SSIFTDR. The register bits subject to the software reset triggered by this bit are indicated by shading in Table 46.8. Because this bit is not automatically cleared after it has been set, write 0 to this bit to release the register bits from the software reset. After writing 0 to this bit, be sure to check that this bit is 0 before starting the next procedural step.

This bit is subject to the software reset by the SSIRST bit. Because the software reset by the SSIRST bit has priority over the reset by this bit, setting this bit is ignored when the SSIRST bit is set.

Table 46.8 Bits subject to software reset by the TFRST bit (1 of 2)

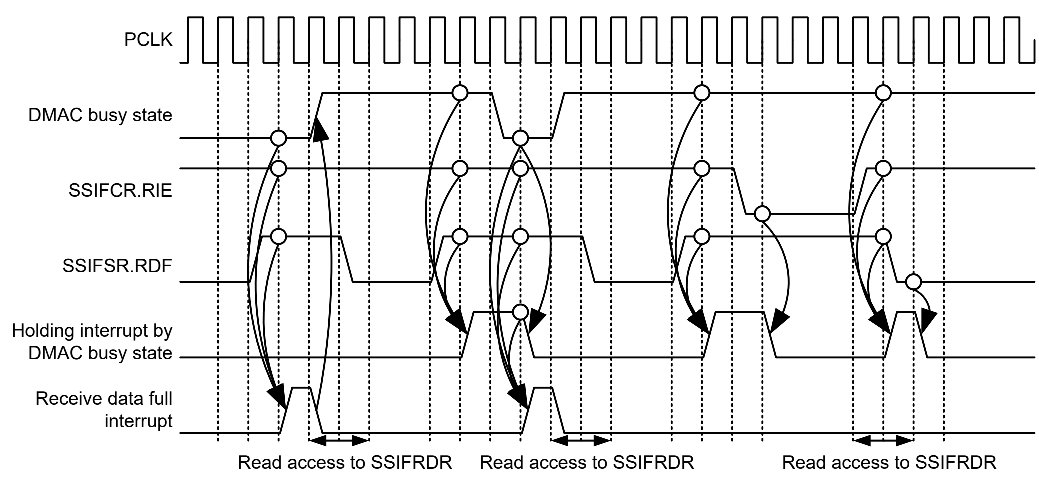
Symbol	Address (BASE+)		+0								+1							
			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSICR	0x00	+0	—	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	—	FRM[1:0]		DWL[2:0]			SWL[2:0]		
		+2	—	MST	BCKP	LRCKP	SPDP	SDTA	PDTA	DEL	CKDV[3:0]				MUEN	—	TEN	REN
SSISR	0x04	+0	—	—	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	—	—	—	—	—	—	—	—	—
		+2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 46.8 Bits subject to software reset by the TFRST bit (2 of 2)

Symbol	Address (BASE+)		+0								+1							
			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSIFCR	0x10	+0	AUC KE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSI RST
		+2	—	—	—	—	BS W	—	—	—	—	—	—	—	TIE	RIE	TFR ST	RFR ST
SSIFSR	0x14	+0	—	—	TDC[5:0]						—	—	—	—	—	—	—	TDE
		+2	—	—	RDC[5:0]						—	—	—	—	—	—	—	RDF
SSIFTDR	0x18	+0	FTDR[31:16]															
		+2	FTDR[15:0]															
SSIFRDR	0x1C	+0	FRDR[31:16]															
		+2	FRDR[15:0]															
SSIOFR	0x20	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	—	—	—	BCK AST P	LRC ONT	—	—	—	—	—	—	OMOD[1:0]	
SSISCR	0x24	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	TDES[4:0]					—	—	—	RDFS[4:0]				

RIE bit (Receive Data Full Interrupt Output Enable)

The RIE bit enables/disables output of receive data full interrupts. Use a receive data full interrupt as an interrupt to trigger data reading from the receive FIFO data register. Write 1 to this bit after specifying the setting condition for receive data full interrupt (by using the SSISCR.RDFS bit). Figure 46.22 shows the timing of generating the receive data full interrupt.



Note: PCLK = PCLKB

Figure 46.22 Timing of receive data full interrupt

TIE bit (Transmit Data Empty Interrupt Output Enable)

The TIE bit enables/disables output of transmit data empty interrupts. Use a transmit data empty interrupt as an interrupt to trigger data writing to the transmit FIFO data register. Write 1 to this bit after specifying the setting condition for transmit data empty interrupt (by using the SSISCR.TDES bit). Figure 46.23 shows the timing of generating the transmit data empty interrupt.

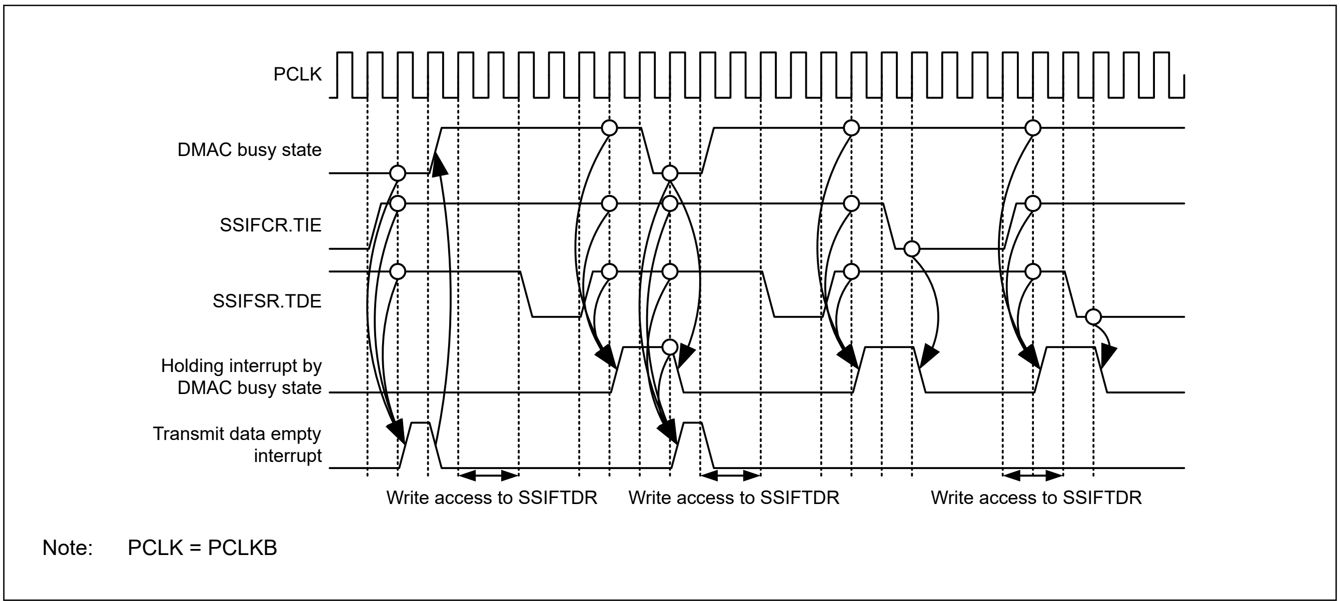


Figure 46.23 Timing of transmit data empty interrupt

BSW bit (Byte Swap Enable)

The BSW bit enables/disables byte swap of register access for the transmit FIFO data register (SSIFTDR) and the receive FIFO data register (SSIFRDR). This bit is valid only with 16-bit access or 32-bit access to SSIFTDR and SSIFRDR. For details, see Figure 46.24.

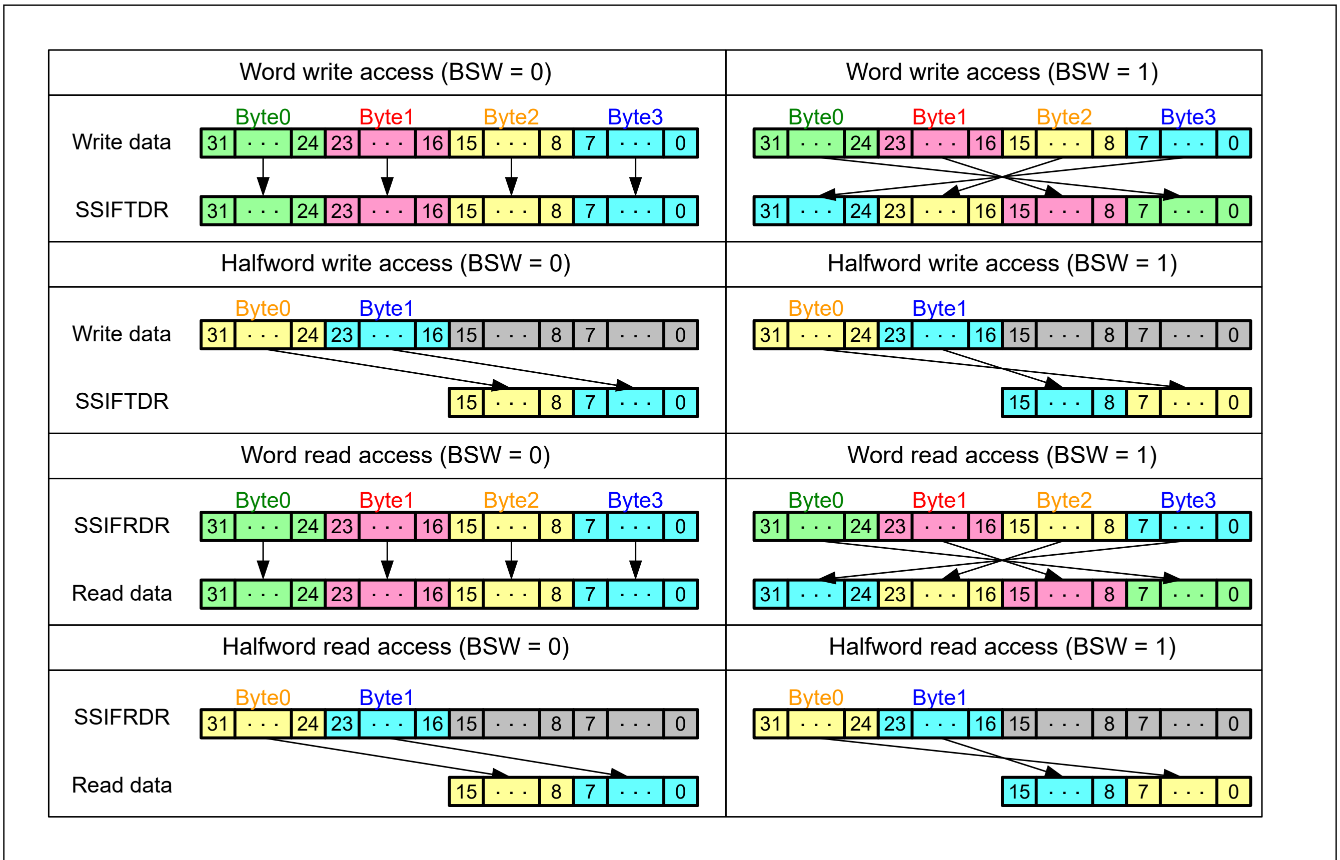


Figure 46.24 Operation example of byte swap

SSIRST bit (Software Reset)

The SSIRST bit sets a software reset of SSIE. Writing 1 to this bit initializes the internal state of SSIE. The register bits subject to the software reset triggered by this bit are indicated by shading in Table 46.9. Because this bit is not automatically cleared after it has been set, write 0 to this bit to release the register bits from the software reset. After writing 0 to this bit, be sure to check that this bit is 0 before starting the next procedural step.

To stop communication of SSIE immediately, after turning off the peripheral functions, write 1 to this bit. Initialization by a software reset is performed without any relation with the bit clock.

Table 46.9 Bits subject to software reset by the SSIRST bit

Symbol	Address (BASE+)		+0								+1							
			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSICR	0x00	+0	—	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	—	FRM[1:0]		DWL[2:0]			SWL[2:0]		
		+2	—	MST	BCKP	LRCKP	SPDP	SDTA	PDTA	DEL	CKDV[3:0]				MUEN	—	TEN	REN
SSISR	0x04	+0	—	—	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	—	—	—	—	—	—	—	—	—
		+2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SSIFCR	0x10	+0	AUCKE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSIRST
		+2	—	—	—	—	BSW	—	—	—	—	—	—	—	TIE	RIE	TFRST	RFRST
SSIFSR	0x14	+0	—	—	TDC[5:0]						—	—	—	—	—	—	—	TDE
		+2	—	—	RDC[5:0]						—	—	—	—	—	—	—	RDF
SSIFTDR	0x18	+0	FTDR[31:16]															
		+2	FTDR[15:0]															
SSIFRDR	0x1C	+0	FRDR[31:16]															
		+2	FRDR[15:0]															
SSIOFR	0x20	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	—	—	—	BCKASTP	LRCONT	—	—	—	—	—	—	OMOD[1:0]	
SSISCR	0x24	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	TDES[4:0]					—	—	—	RDFS[4:0]				

AUCKE bit (AUDIO_MCK Enable in Master mode Communication)

The AUCKE bit enables/disables supply to AUDIO_MCK while in master-mode communication (MST = 1).

Changing the value of this bit must be performed only after specifying the settings related to AUDIO_MCK (by using the CKS, MST, BCKP, and CKDV bits in the SSICR register).

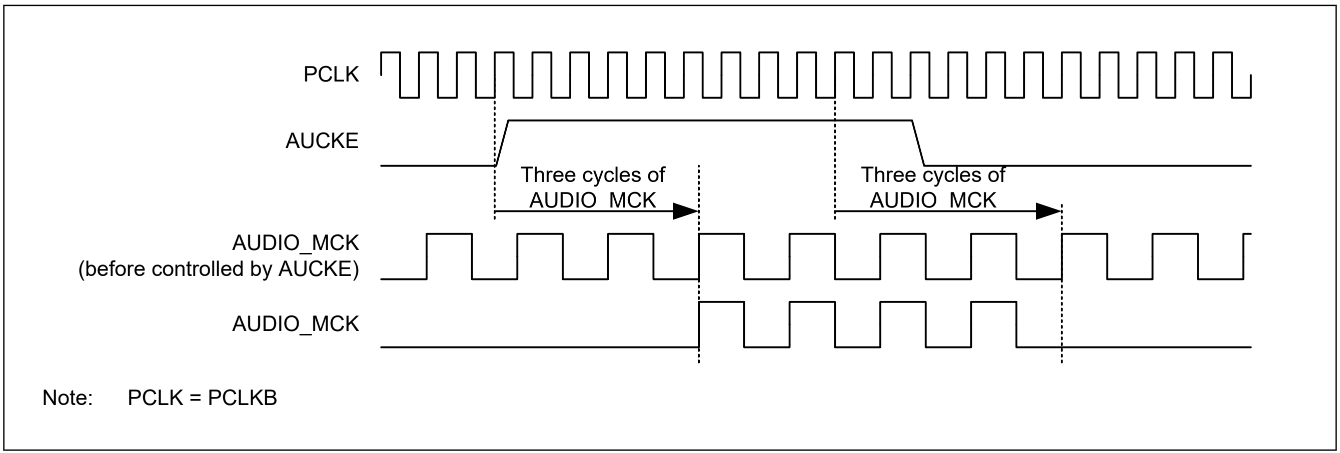


Figure 46.25 Stop/resume of AUDIO_MCK

Note: In slave-mode communication (SSICR.MST = 0), SSIE needs supply of SSIBCKn (n = 0, 1). To stop BCK on the master side, make sure that SSIE is in the idle state (SSISR.IIRQ = 1). If BCK is stopped before SSIE becomes idle, take the procedure to start communication in Figure 46.52 or wait for an idle state by taking the procedure to resume communication in Figure 46.58.

In master-mode communication (SSICR.MST = 1), SSIE operates with the audio clock (AUDIO_MCK). To stop SSIE completely, make sure that SSIE is in the idle state (SSISR.IIRQ = 1) and then write 0 to SSIFCR.AUCKE. If 0 is written to SSIFCR.AUCKE before SSIE becomes idle, take the procedure to start communication in Figure 46.52.

Figure 46.26 and Figure 46.27 show the timings of signal operation in the period from setting this bit to 1 to the output to the SSIBCKn (n = 0, 1) pin.

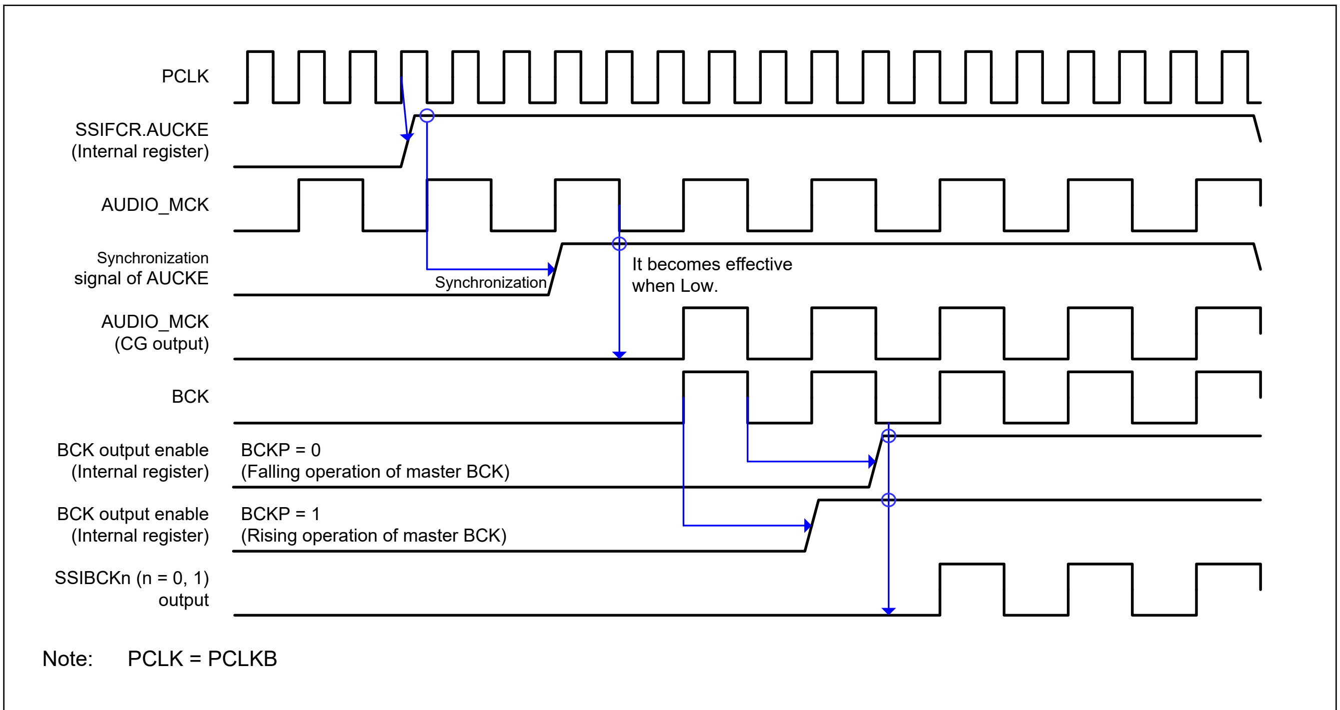
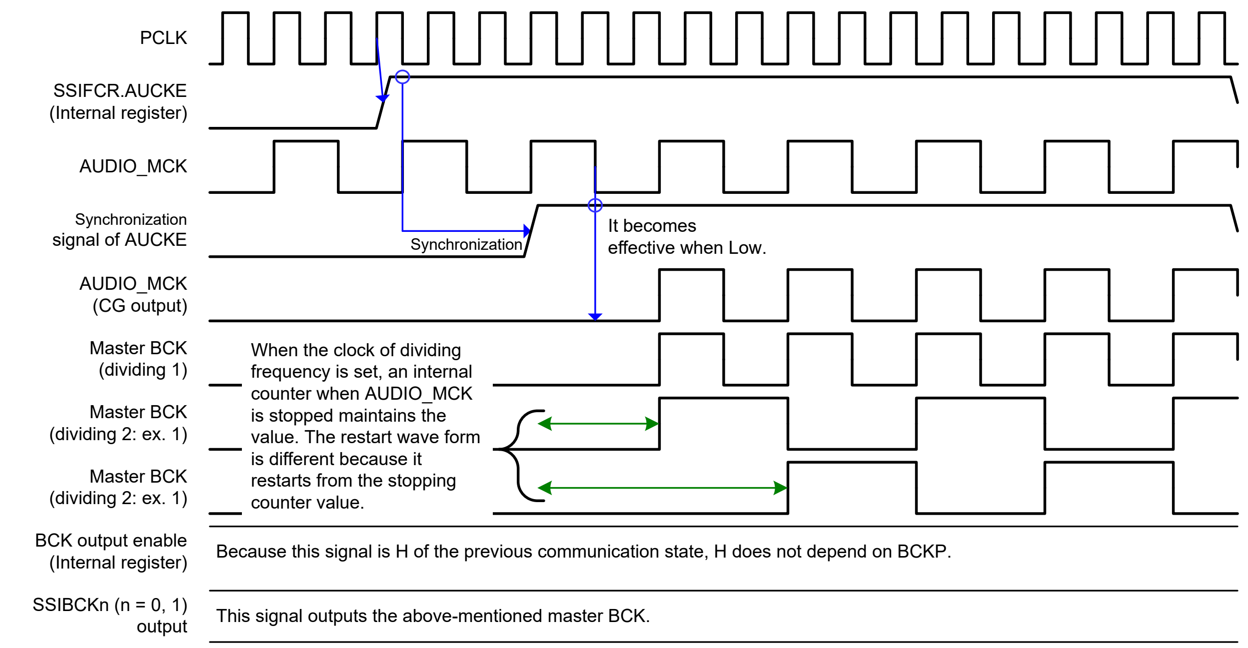


Figure 46.26 Timing diagram for the operation from system reset to start of master-mode communication



Note: PCLK = PCLKB

Figure 46.27 Timing diagram for the operation from stop of communication to start of master-mode communication

Note: If the supply of AUDIO_MCK stops, the value of the SSIBCKn (n = 0, 1) pin is held. Therefore, the SSIBCKn (n = 0, 1) signal might stop in the H (high level) state.

46.2.4 SSIFSR : FIFO Status Register

Base address: SSIEn = 0x4025_D000 + 0x0100 × n (n = 0, 1)
SSIEn_NS = 0x5025_D000 + 0x0100 × n (n = 0, 1)

Offset address: 0x14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	TDC[5:0]						—	—	—	—	—	—	—	TDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	RDC[5:0]						—	—	—	—	—	—	—	RDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RDF	Receive Data Full Flag 0: The size of received data in SSIFRDR is not more than the value of SSISCR.RDFS. 1: The size of received data in SSIFRDR is not less than the value of SSISCR.RDFS plus one.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
13:8	RDC[5:0]	Number of Receive FIFO Data Indication Flag Number of receive FIFO data indication flag	R
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
16	TDE	Transmit Data Empty Flag 0: The free space of SSIFTDR is not more than the value of SSISCR.TDES. 1: The free space of SSIFTDR is not less than the value of SSISCR.TDES plus one.	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W
29:24	TDC[5:0]	Number of Transmit FIFO Data Indication Flag Number of transmit FIFO data indication flag	R
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register is configured with status flags that indicate the status of the transmit FIFO data register and the receive FIFO data register.

RDF flag (Receive Data Full Flag)

The RDF flag indicates that the receive FIFO data register (SSIFRDR) has unread received data not less than the amount set with the SSISCR.RDFS bit plus one. This flag is set by automatic determination, but it must be cleared by register access.

[Priority order for setting and clearing]

Clearing is prioritized.

[Clearing condition]

Either of the following two:*1

1. Writing 0 to this bit after reading 1 from this bit (CPU operation)*2
2. Last access (DTC/DMAC operation) to read data from SSIFRDR by an interrupt routine using the DTC and DMAC.

[Clearing timing]

Clearing timing corresponding to the above clearing condition

1. When 0 is written to this bit after reading 1 from this bit (same as the timing in Figure 46.19)
2. After the PCLKB cycle in which the last access instruction is issued to read data from SSIFRDR by an interrupt routine using the DTC and DMAC.

[Setting condition]

SSIFTDR has free space not less than the amount set with the SSIFCR.TTRG bit plus one.

[Setting timing]

At completion of transfer from the shift register that results in SSIFRDR having data not less than the amount set with the SSISCR.RDFS bit plus one.

Note 1. These bits are cleared by a software reset (SSIFCR.SSIRST = 1) and receive FIFO data register reset (SSIFCR.RFRST = 1). Reset conditions available for these bits are the software reset and receive FIFO data register reset as well as the clearing conditions described above.

Note 2. After reading 1 from this bit, this bit is cleared when one of the following four conditions is met:

- A software reset is done (SSIFCR.SSIRST = 1).
- A receive FIFO data register reset is done (SSIFCR.RFRST = 1).
- After 1 has been read, writing of 0 is complete.
- Last access is performed to read data from SSIFRDR by an interrupt routine using the DTC and DMAC.

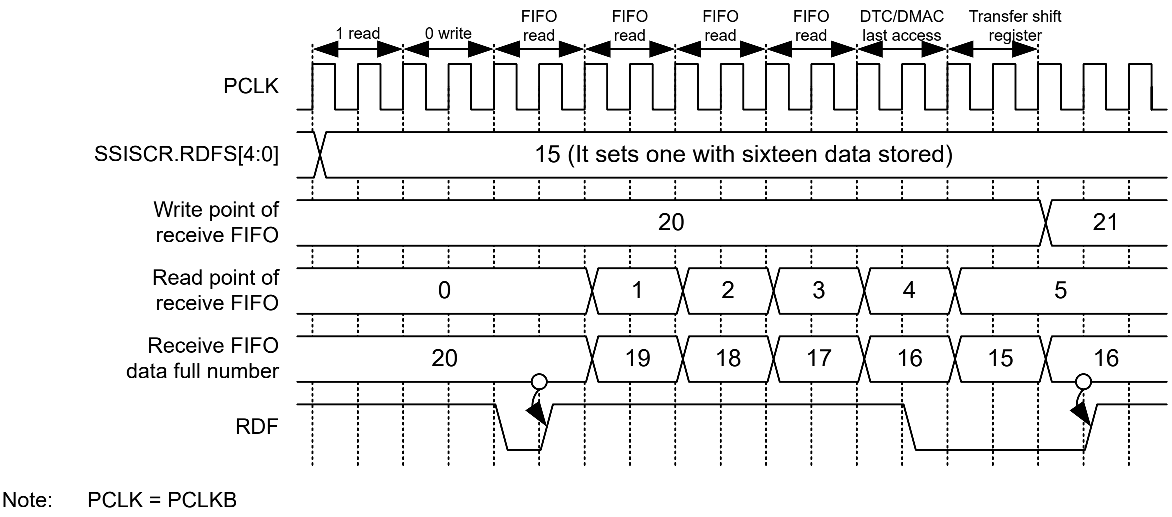


Figure 46.28 Timing diagram for setting and clearing RDF

RDC[5:0] flags (Number of Receive FIFO Data Indication Flag)

The RDC[5:0] flags indicate the number of valid data that are stored in the receive FIFO data register (SSIFRDR). With this flag as 0x00, there is no received data. With 0x20, the register is filled with received data and there is no free space.

TDE flag (Transmit Data Empty Flag)

The TDE flag indicates that the transmit FIFO data register (SSIFTDR) has free space not less than the amount set with the SSIFCR.TTRG bit plus one. This flag is set by automatic determination, but it must be cleared by register access.

[Priority order for setting and clearing]

Clearing is prioritized.*1

[Clearing condition]

Either of the following two:

1. Writing 0 to this bit after reading 1 from this bit (CPU operation)*2
2. Last access (DTC/DMAC operation) to write data to SSIFTDR by an interrupt routine using the DTC and DMAC.

[Clearing timing]

Clearing timing corresponding to the above clearing condition

1. When 0 is written to this bit after reading 1 from this bit (same as the timing in Figure 46.19)
2. Last access (DTC/DMAC operation) to write data to SSIFTDR by an interrupt routine using the DTC and DMAC.

[Setting condition]

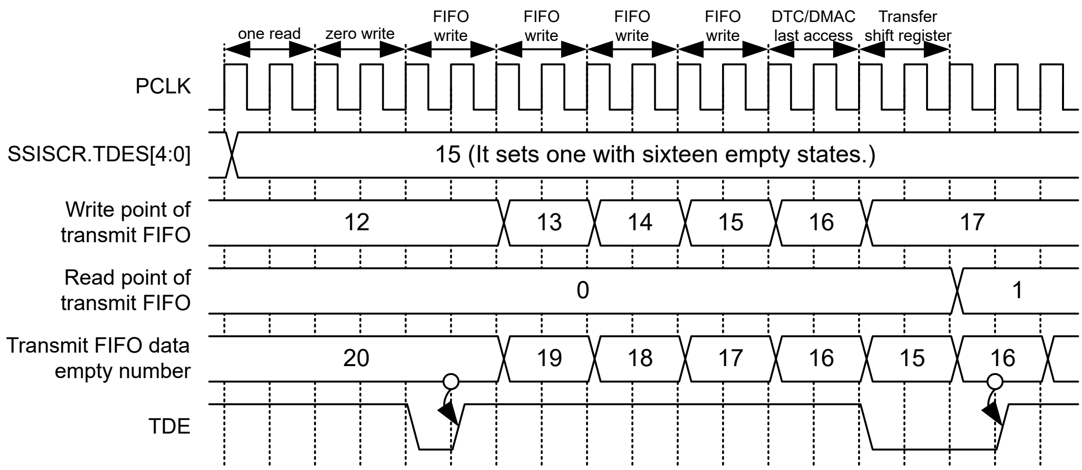
SSIFTDR has free space not less than the amount set with the SSIFCR.TTRG bit plus one.

[Setting timing]

While operating on PCLKB, SSIFTDR is found to have free space not less than “size set in the SSISCR.TDES bits + 1.”

Note 1. This bit is cleared by a software reset (SSIFCR.SSIRST = 1) and transmit FIFO data register reset (SSIFCR.TFRST = 1). The software reset and transmit FIFO data register reset have priority over all the clearing conditions described above.

Note 2. After reading 1 from this bit, this bit is cleared when one of the following four conditions is met:
- A software reset is done (SSIFCR.SSIRST = 1).
- A transmit FIFO data register reset is done (SSIFCR.TFRST = 1).
- After 1 has been read, writing of 0 is complete.
- Last access is performed to write data to SSIFTDR by an interrupt routine using the DTC and DMAC.



Note: PCLK = PCLKB

Figure 46.29 Timing diagram for setting and clearing TDE

TDC[5:0] flags (Number of Transmit FIFO Data Indication Flag)

The TDC[5:0] flags indicate the number of valid data that are stored in the transmit FIFO data register (SSIFTDR). With this flag as 0x00, there is no data to be transmitted. With 0x20, there is no space to write data.

46.2.5 SSIFTDR : Transmit FIFO Data Register

Base address: SSIEn = 0x4025_D000 + 0x0100 × n (n = 0, 1)
SSIEn_NS = 0x5025_D000 + 0x0100 × n (n = 0, 1)

Offset address: 0x18

Bit position:	31 ... 0
Bit field:	SSIFTDR[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	SSIFTDR[31:0]	Transmit FIFO Data	W

Note: S-TYPE-3, P-TYPE-3

This register stores data to be serially transmitted. 0 is returned when this register is read.

When you use this register for transmission, specify data writing to this register as the DTC/DMAC operation that is triggered by a transmit data empty interrupt. Determine the access size to this register according to the data word length to be communicated in Table 46.10.

Table 46.10 Register access restriction to FIFOs (1 of 2)

Access Size				
SSICR.DWL[2:0]	Data Word Length	Byte	Halfword	Word
000b	8	✓	—	—
001b	16	—	✓	—
010b	18	—	—	✓
011b	20	—	—	✓
100b	22	—	—	✓
101b	24	—	—	✓

Table 46.10 Register access restriction to FIFOs (2 of 2)

Access Size				
SSICR.DWL[2:0]	Data Word Length	Byte	Halfword	Word
110b	32	—	—	✓
111b	Setting prohibited	—	—	—

Figure 46.30 shows register access to the transmit FIFO data register.

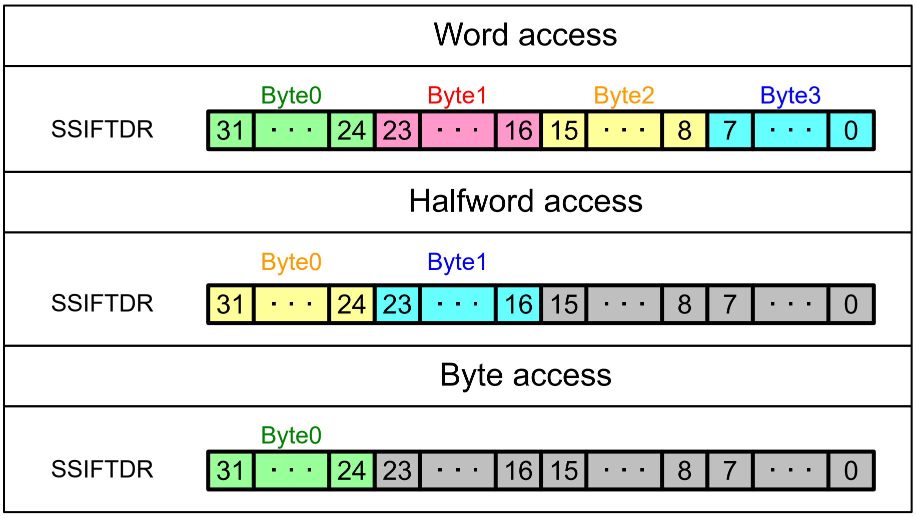
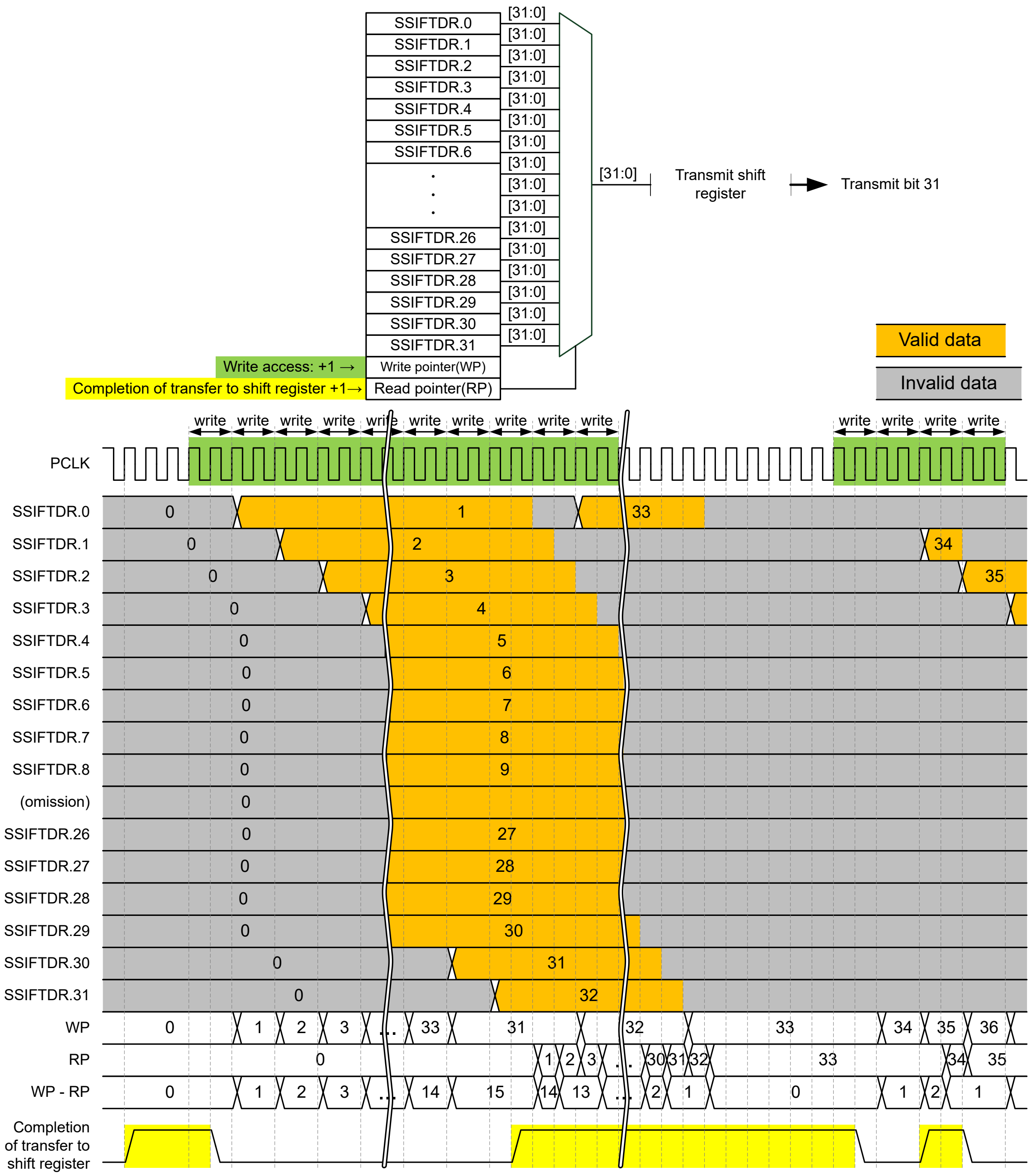


Figure 46.30 Example of register access to the transmit FIFO data register

Figure 46.31 shows the configurations and operation examples of the transmit FIFO data register and transmit shift register. The configurations are for storing data to FIFO and not related with communication.



Note: PCLK = PCLKB

Figure 46.31 Configuration of the transmit FIFO data register and transmit shift register, and FIFO operation example

46.2.6 SSIFRDR : Receive FIFO Data Register

Base address: SSIEn = 0x4025_D000 + 0x0100 × n (n = 0, 1)
SSIEn_NS = 0x5025_D000 + 0x0100 × n (n = 0, 1)

Offset address: 0x1C

Bit position:	31 … 0
Bit field:	SSIFRDR[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

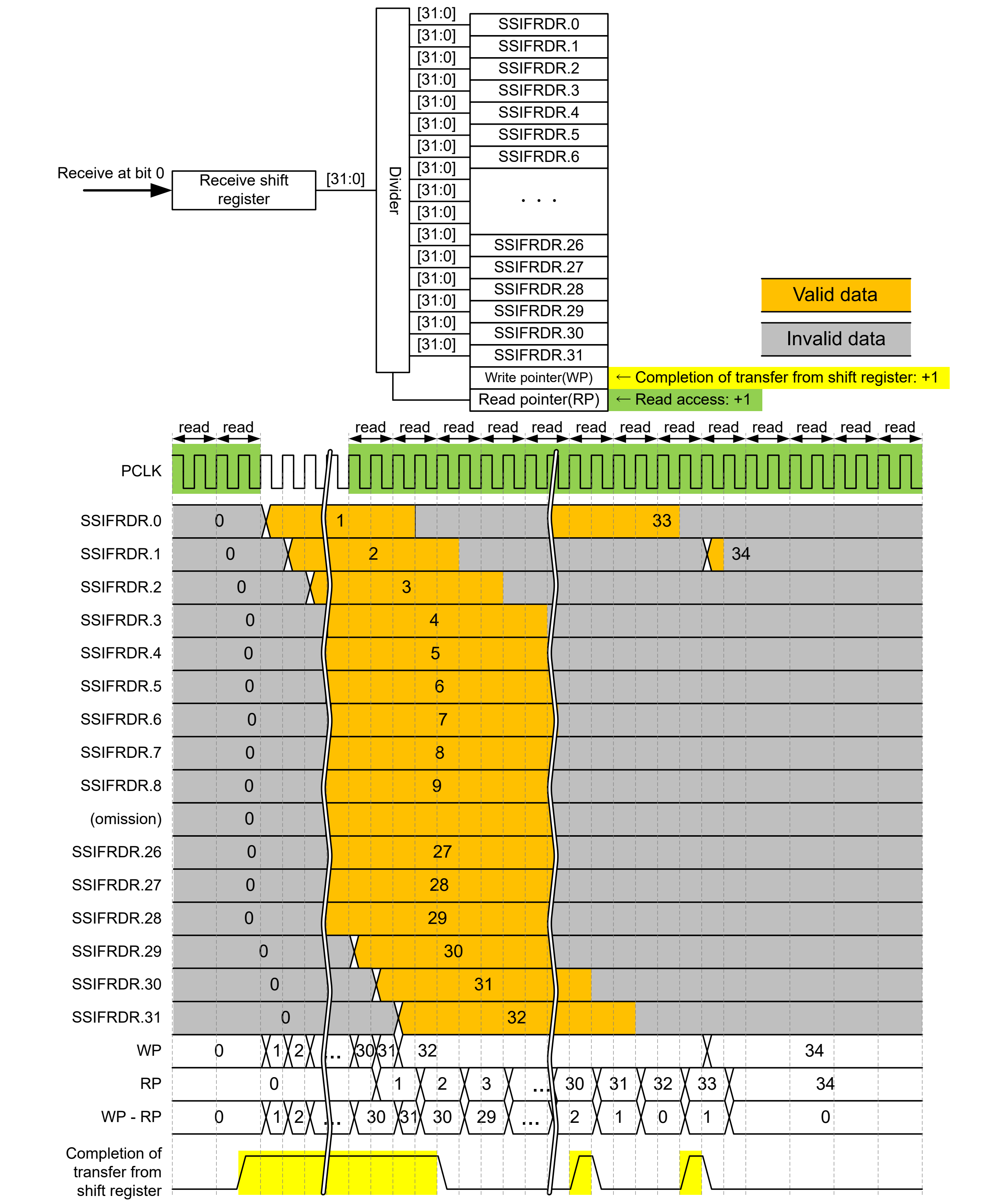
Bit	Symbol	Function	R/W
31:0	SSIFRDR[31:0]	Receive FIFO Data	R

Note: S-TYPE-3, P-TYPE-3

When you use this register for reception, specify data reading from this register as the DTC/DMAC operation that is triggered by a transmit data empty interrupt. Determine the access size to this register according to the data word length to be communicated in Table 46.10.

Register access to the receive FIFO data register is same as for the transmit FIFO data register.

Figure 46.31 shows the configurations and operation examples of the receive FIFO data register and receive shift register.



Note: PCLK = PCLKB

Figure 46.32 Configuration of the transmit FIFO data register and transmit shift register, and FIFO operation example

46.2.7 SSIOFR : Audio Format Register

Base address: SSIEn = 0x4025_D000 + 0x0100 × n (n = 0, 1)
SSIEn_NS = 0x5025_D000 + 0x0100 × n (n = 0, 1)

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	BCKASTP	LRCONT	—	—	—	—	—	—	OMOD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	OMOD[1:0]	Audio Format Select*3 *4 0 0: I²S format 0 1: TDM format 1 0: Monaural format 1 1: Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	LRCONT	Whether to Enable LRCK/FS Continuation*1 *2 0: Disables LRCK/FS continuation 1: Enables LRCK/FS continuation	R/W
9	BCKASTP	Whether to Enable Stopping BCK Output When SSIE is in Idle Status*1 *2 0: Always outputs BCK to the SSIBCKn (n = 0, 1) pin 1: Automatically controls output of BCK to the SSIBCKn (n = 0, 1) pin	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. This bit is valid only in master-mode communication (SSICR.MST = 1). The setting is invalid in slave-mode communication (SSICR.MST = 0).
Note 2. The BCKASTP and LRCONT bits must not be set to 1 together.
Note 3. While SSIE is communicating (SSISR.IIRQ = 0), writing to these bits is prohibited. If the value of these bits is changed by writing, subsequent operation is unpredictable.
Note 4. If the communication format of other-party device is compatible with a communication format of SSIE, specify and use the communication format that enables communication with the other-party device.

This register is used to set an audio format (which involves the settings of communication format, LR clock/frame synchronization continuation mode, and BCK output stop).

OMOD[1:0] bits (Audio Format Select)

The OMOD[1:0] bits set an audio format. Writing to these bits must be performed when the LR clock supply to the SSILRCKn/SSIFSn (n = 0, 1) pin is stopped. For details about the output of LR clock, see the detailed description of the LRCONT bit in section 46.2.7. SSIOFR : Audio Format Register.

LRCONT bit (Whether to Enable LRCK/FS Continuation)

The LRCONT bit enables or disables the output from SSILRCKn/SSIFSn (n = 0, 1) pin when the communication mode is master-mode communication (SSICR.MST = 1) and SSIE is in the idle state (SSISR.IIRQ = 1).

Even in the idle state, a signal can output from the SSILRCKn/SSIFSn (n = 0, 1) pin when this bit is set to 1 (to enable LR clock/frame synchronization continuation) in master mode (SSICR.MST = 1).

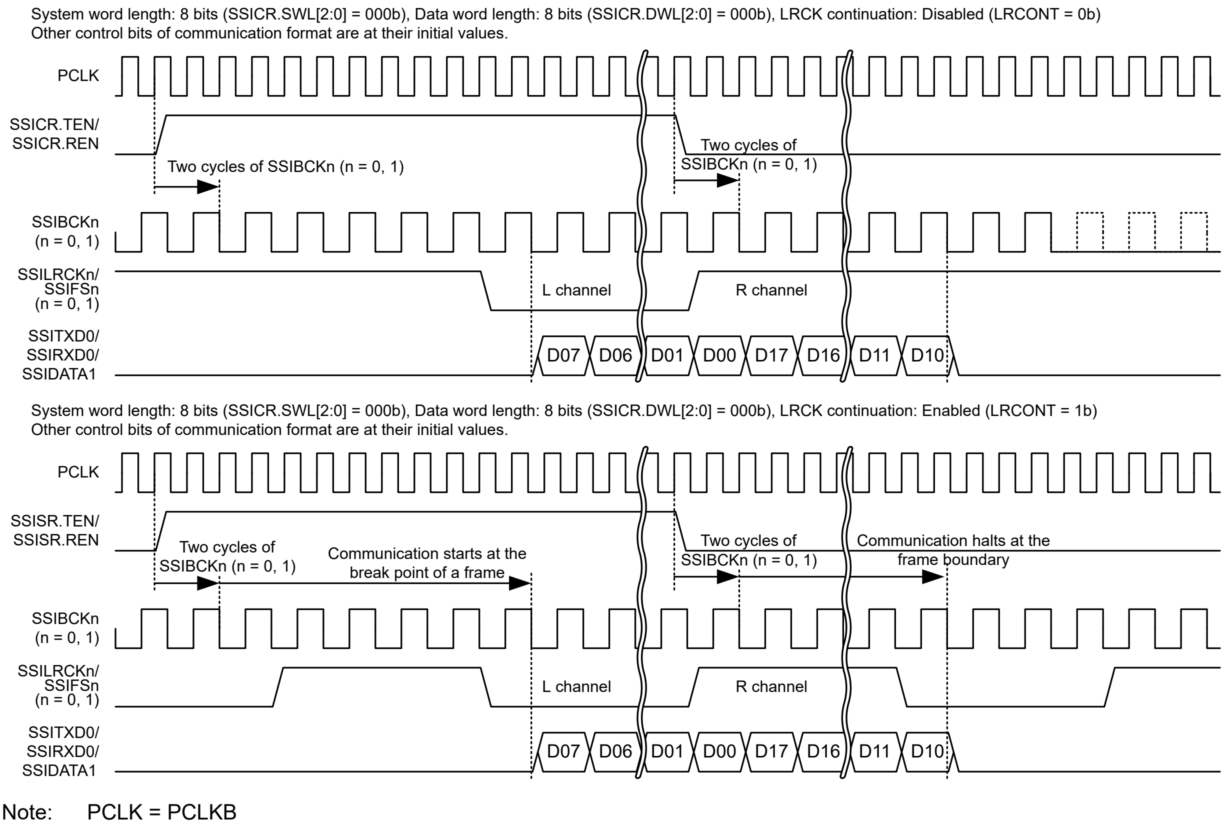


Figure 46.33 Example of LR clock/frame synchronization continuation operation

BCKASTP bit (Whether to Enable Stopping BCK Output When SSIE is in Idle Status)

The BCKASTP bit turns on or off the function to output BCK to the SSIBCKn (n = 0, 1) pin according to the communication shown in Figure 46.34 and Figure 46.35 in master-mode communication (SSICR.MST = 1).

Changing the value of this bit must be performed only after setting the communication format to be used.

This bit must be used in the following way:

Write 0 to the BCKASTP bit, and then start communication. During the communication, write 1 to the BCKASTP bit. By this operation, the bit clock output to the SSIBCKn (n = 0, 1) pin stops automatically when the communication stops. To resume the communication, set SSIE to the idle state (SSICR.IIRQ = 1), enable the supply of AUDIO_MCK (SSIFCR.AUCKE = 1), and then write 0 to the BCKASTP bit.

When the communication mode is master-mode communication (SSICR.MST = 1) and SSIE is in the idle state (SSICR.IIRQ = 1):

Table 46.11 BCKASTP bit status and SSIBCKn (n = 0, 1) pin output

BCKASTP Bit	SSIBCKn (n = 0, 1) Pin Output Status
0	Output
1	Stopped

Note: The BCKASTP bit cannot be used when the other-party device (which is a slave) requires the clock output from the SSIBCKn (n = 0, 1) pin before and during communication. In such a case, use the BCKASTP bit to stop the clock only after communication. For the timing of enabling the clock stop function, see Figure 46.34.

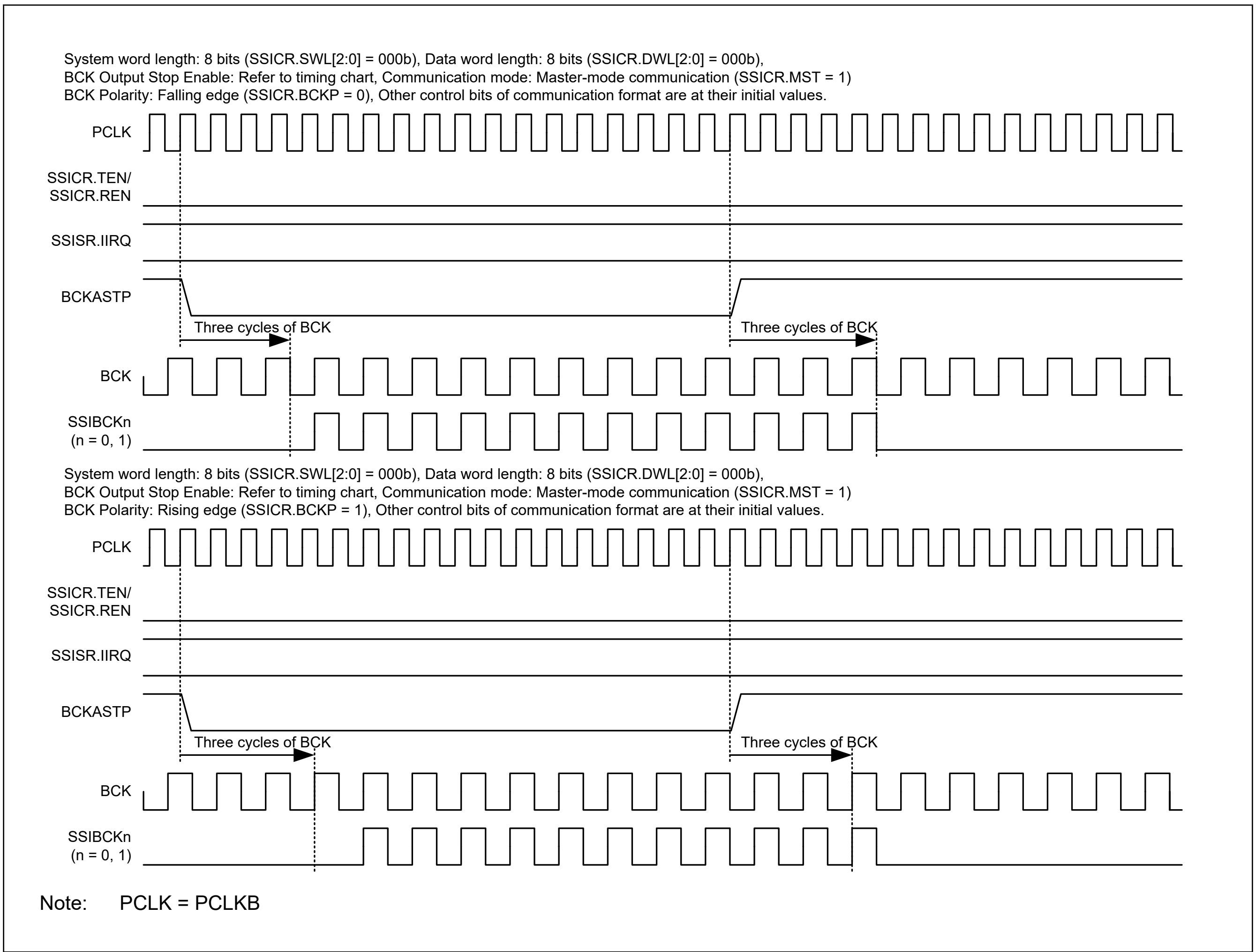


Figure 46.34 Example operation of the BCKASTP bit (idle state)

When the communication mode is master-mode communication (SSICR.MST = 1) and the BCK output stop function is enabled (BCKASTP = 1):

Details of the BCK output to the SSIBCKn (n = 0, 1) pin are as follows:

Output start timing: BCK is output in appropriate timing so that a valid edge is generated when the LR clock/frame synchronization signal shifts to a valid value.

Output stop timing: 1 to 1.5 clock cycles after a frame boundary.

For details about the timings, see the timing diagram in Figure 46.35.

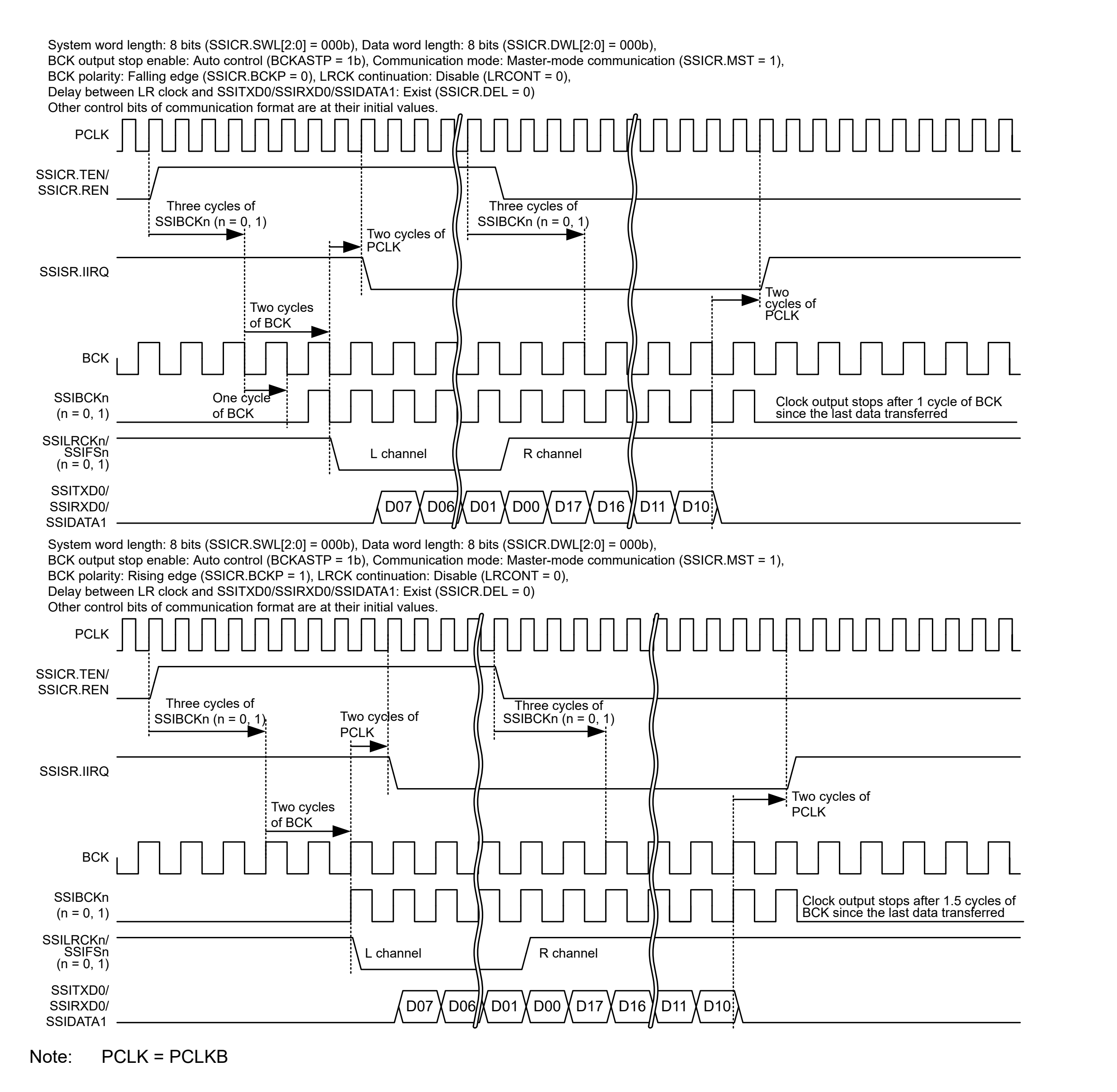


Figure 46.35 Example operation of the BCKASTP bit (communication operation with BCKASTP = 1)

46.2.8 SSISCR : Status Control Register

Base address: SSIEn = 0x4025_D000 + 0x0100 × n (n = 0, 1)
SSIEn_NS = 0x5025_D000 + 0x0100 × n (n = 0, 1)

Offset address: 0x24

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TDES[4:0]					—	—	—	RDFS[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	RDFS[4:0]	RDF Setting Condition Select*1 0x00: SSIFRDR has one stage or more data size. 0x01: SSIFRDR has two stages or more data size. ⋮ 0x1E: SSIFRDR has thirty-one stages or more data size. 0x1F: SSIFRDR has thirty-two stages or more data size.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
12:8	TDES[4:0]	TDE Setting Condition Select*1 0x00: SSIFTDR has one stage or more free space. 0x01: SSIFTDR has two stages or more free space. ⋮ 0x1E: SSIFTDR has thirty-one stages or more free space. 0x1F: SSIFTDR has thirty-two stages or more free space.	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Writing to these bits while SSIE is in a communication state (SSISR.IIRQ = 0) is prohibited. If written, the operation performed immediately after writing is not guaranteed.

RDFS[4:0] bits (RDF Setting Condition Select)

The RDFS[4:0] bits set the setting condition of the receive data full flag (RDF).

TDES[4:0] bits (TDE Setting Condition Select)

The TDES[4:0] bits set the setting condition of the transmit data empty flag (TDE).

46.3 Communication Formats

SSIE supports three communication formats. Table 46.12 shows supported communication formats.

Table 46.12 Supported communication formats

Communication Format	SSIOFR.OMOD[1:0]
I^{2}S format	00
TDM format	01
Monaural format	10

The following describes the serial data structure shared by communication formats. A serial data structure is defined by the system word length (set in SSICR.SWL[2:0]) and the data word length (set in SSICR.DWL[2:0]). If the data word length is shorter than the system word length, padding bits are transferred in the serial data. For details, see Figure 46.36.

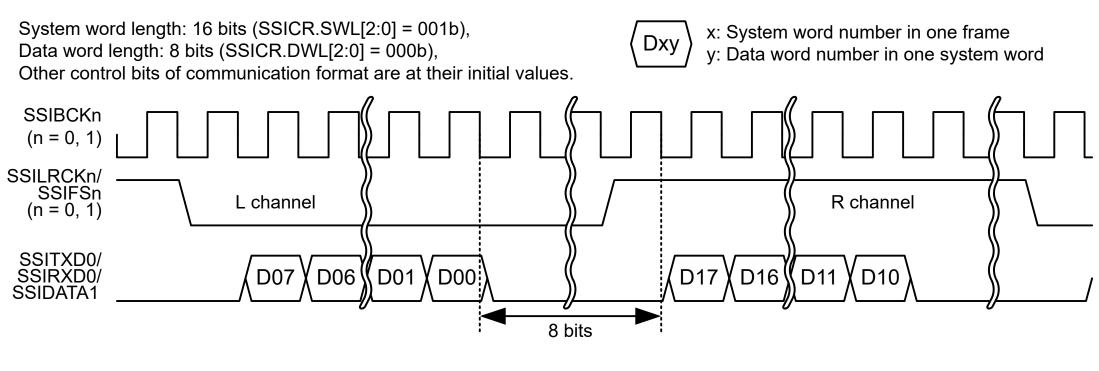


Figure 46.36 Example of padding bit transfer (I^{2}S format: system word length > data word length)

Table 46.13 lists the number of padding bits to be transferred with each combination of system word length (SSICR.SWL[2:0]) and data word length (SSICR.DWL[2:0]). “-” indicates that the setting is prohibited.

Table 46.13 Number of padding bits

	SSICR.DWL[2:0]	000b	001b	010b	011b	100b	101b	110b	111b
SSICR.SWL[2:0]	System Word Length	8	16	18	20	22	24	32	Setting prohibited
000b	8	0	—	—	—	—	—	—	—
001b	16	8	0	—	—	—	—	—	—
010b	24	16	8	6	4	2	0	—	—
011b	32	24	16	14	12	10	8	0	—
100b	48	40	32	30	28	26	24	16	—
101b	64	56	48	46	44	42	40	32	—
110b	128	120	112	110	108	106	104	96	—
111b	256	248	240	238	236	234	232	224	—

46.3.1 I²S Format

The I²S format is a communication format used for connection with I²S-compatible serial devices. With this format setting (SSIOFR.OMOD[1:0] = 00b), one frame is configured with two system words, one for the channel L and the other for channel R. The SSILRCKn/SSIFSn (n = 0, 1) signals are at a low level for the channel L and at a high level for the channel R. Set the polarity of the signals with the SSICR.LRCKP bit. Figure 46.37 shows the I²S format without padding. See Figure 46.36 for the format with padding.

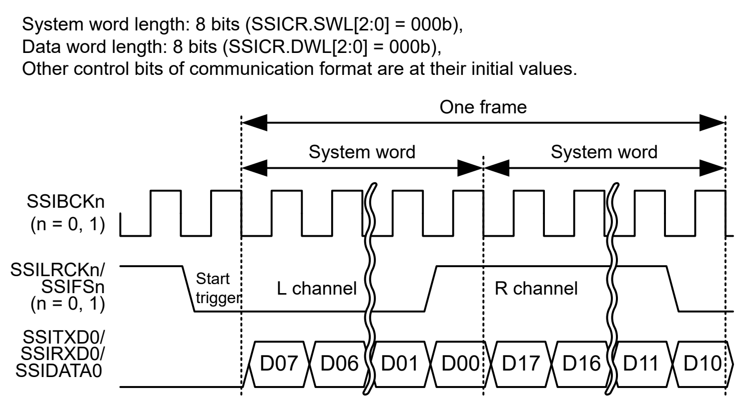


Figure 46.37 I²S format (without padding: system word length = data word length)

For the state of external pins when SSIE is in the idle state, see section 46.5.1. Idle State.

Note: SSIE has the SSILRCKn/SSIFSn (n = 0, 1) pin, which indicates the synchronization of communication. When SSIE is in slave mode (SSICR.MST = 0), the communication format SSIE uses must match that of the other-party device to communicate. SSIE uses the signal input by the SSILRCKn/SSIFSn (n = 0, 1) pin only as a trigger to start communication.

46.3.2 Monaural Format

The monaural format is a communication format used for connection with monaural-compatible serial devices. When the monaural format is specified (SSIOFR.OMOD[1:0] = 10b) for use, one frame consists of one system word. Also, a rising edge of the SSILRCKn/SSIFSn (n = 0, 1) signal indicates a communication start trigger. Figure 46.38 and Figure 46.39 respectively show the monaural formats without and with padding.

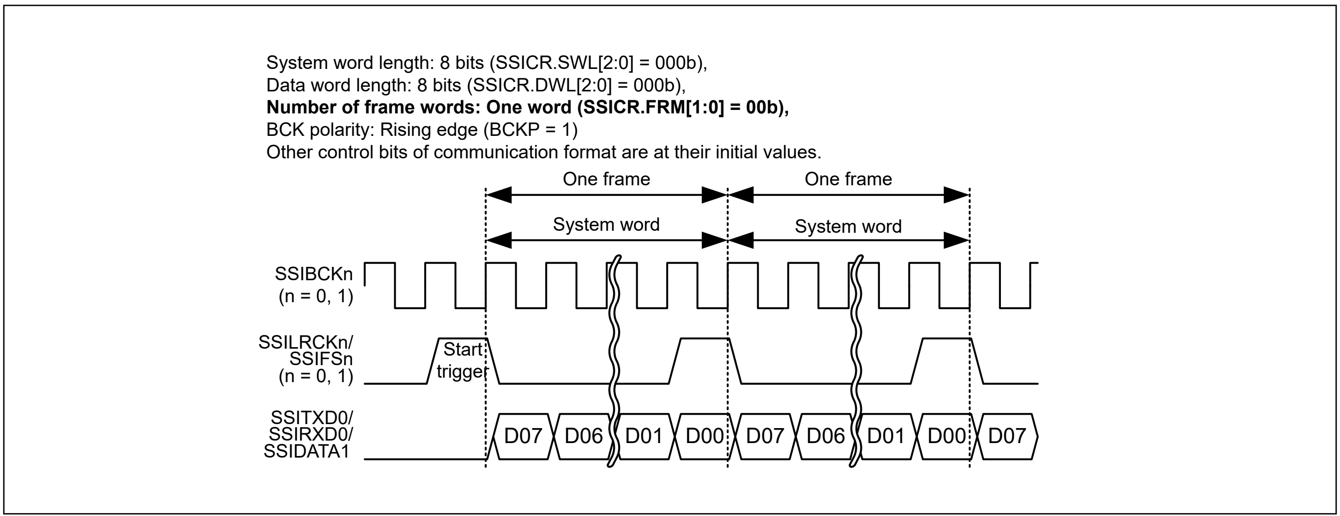


Figure 46.38 Short frame in monaural format (without padding: system word length = data word length)

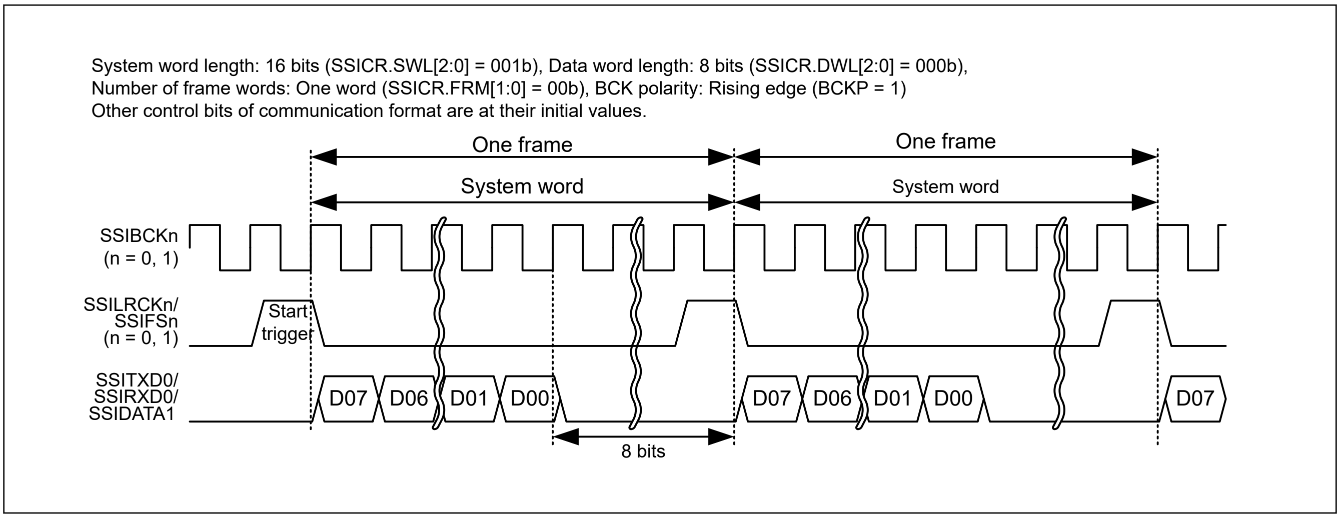


Figure 46.39 Short frame in monaural format (with padding: system word length > data word length)

The monaural formats supported by SSIE consist of short frames and long frames. See section 46.3.2.1. Short Frame and section 46.3.2.2. Long Frame for the difference between these two frames.

For the state of external pins state when SSIE is in the idle state, see section 46.5.1. Idle State.

Note: SSIE has the SSILRCKn/SSIFSn (n = 0, 1) pin, which indicates the synchronization of communication. When SSIE is in slave mode (SSICR.MST = 0), the communication format SSIE uses must match that of the other-party device to communicate. SSIE uses the signal input by the SSILRCKn/SSIFSn (n = 0, 1) pin only as a trigger to start communication.

46.3.2.1 Short Frame

When a short frame is used (SSICR.DEL = 0), the SSILRCKn/SSIFSn (n = 0, 1) signal indicating the start of serial data is set to high level only for 1 cycle of SSIBCKn (n = 0, 1). Data transfer starts at the falling edge of the signal.

46.3.2.2 Long Frame

When a long frame is used (SSICR.DEL = 1), the SSILRCKn/SSIFSn (n = 0, 1) signal indicating the start of serial data is set to high level only for 2 cycles of SSIBCKn (n = 0, 1). See Figure 46.40. Data transfer starts at the rising edge of the signal.

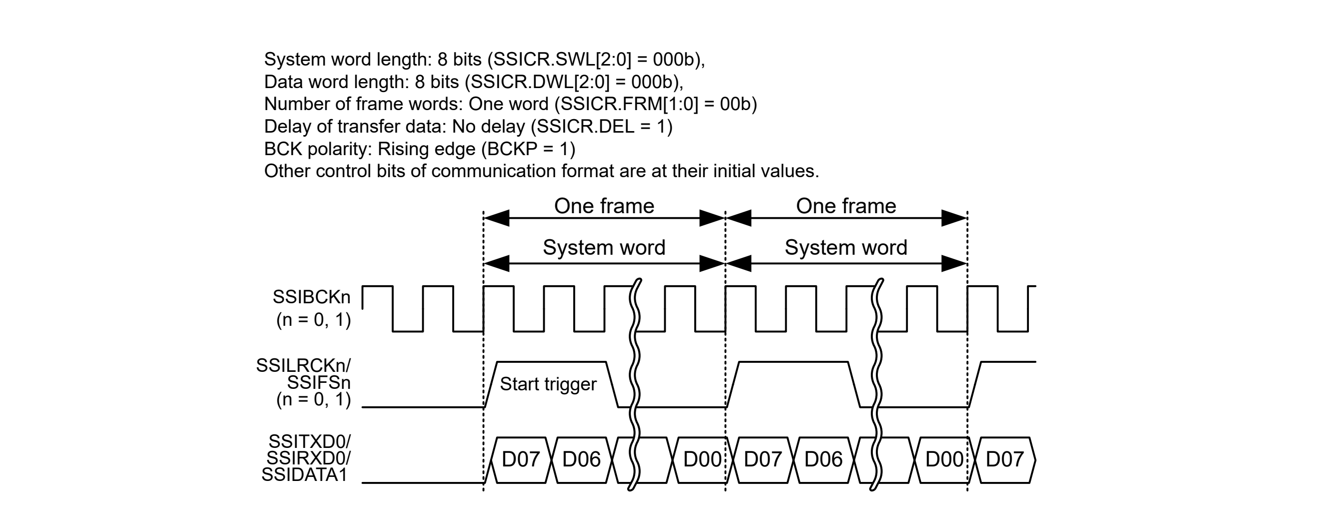


Figure 46.40 Long frame in monaural format (without padding)

46.3.3 TDM Format

The TDM format is a communication format used for connection with TDM-compatible multi-channel devices. With this format setting (SSIOFR.OMOD[1:0] = 01b), one frame is configured with four to eight system words set with the SSICR.FRM[1:0] bits. With this format, the SSILRCKn/SSIFSn (n = 0, 1) signal is at a high level for the first one system word and at a low level for the rest. The pulse generated on the SSILRCKn/SSIFSn (n = 0, 1) signal is defined as the SYNC pulse and its rising edge means a start of one frame. Figure 46.41 and Figure 46.42 respectively show the TDM formats without and with padding.

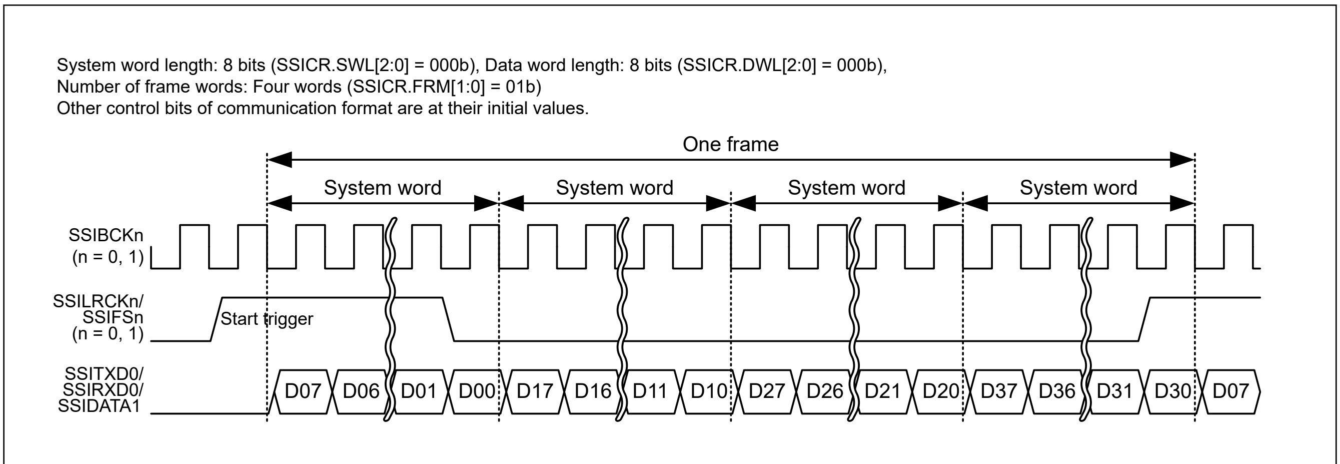


Figure 46.41 TDM format (without padding: system word length = data word length)

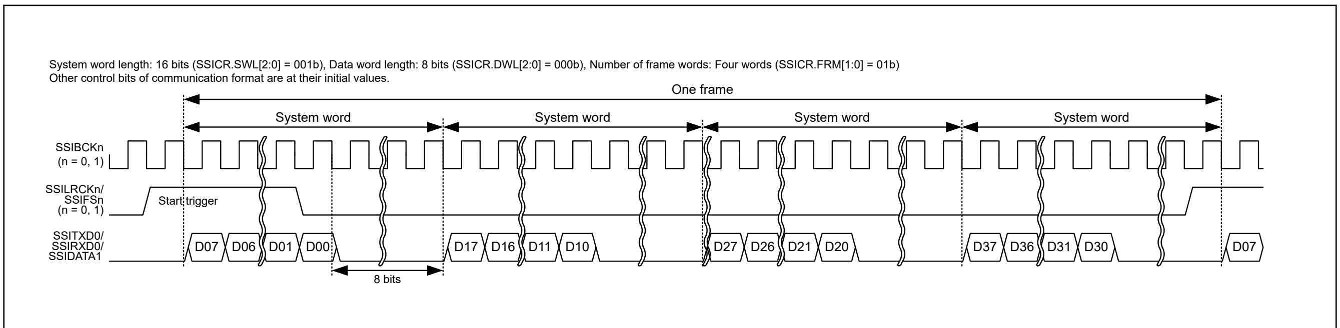


Figure 46.42 TDM format (with padding: system word length > data word length)

For the state of external pins when SSIE is in the idle state, see section 46.5.1. Idle State.

Note: SSIE has the SSILRCKn/SSIFSn (n = 0, 1) pin, which indicates the synchronization of communication. When SSIE is in slave mode (SSICR.MST = 0), the communication format SSIE uses must match that of the other-party device to communicate. SSIE uses the signal input by the SSILRCKn/SSIFSn (n = 0, 1) pin only as a trigger to start communication.

46.4 Communication Modes

SSIE supports the following communication modes. Table 46.15 lists the control bits that are not available with each communication mode. See section 46.4.1. Slave-mode Communication to section 46.4.5. Transmission and Reception for details of these communication modes.

Table 46.14 Communication modes

Communication Mode	SSICR.MST Bit	SSICR.REN Bit	SSICR.TEN Bit
Slave-mode transmission	0	0	1
Slave-mode reception	0	1	0
Slave-mode transmission and reception	0	1	1
Master-mode transmission	1	0	1
Master-mode reception	1	1	0
Master-mode transmission and reception	1	1	1

Table 46.15 Control bits that cannot be used in each communication mode

Control Bit	Slave-mode Reception	Slave-mode Transmission	Slave-mode Transmission and Reception	Master-mode Reception	Master-mode Transmission	Master-mode Transmission and Reception
SSICR.CKS	Invalid	Invalid	Invalid	Available	Available	Available
SSICR.CKDV	Invalid	Invalid	Invalid	Available	Available	Available
SSICR.MUEN	Invalid	Available	Available	Invalid	Available	Available
SSICR.TEN	Invalid	Available	Available	Invalid	Available	Available
SSICR.REN	Available	Invalid	Available	Available	Invalid	Available
SSIFCR.AUCKEN	Invalid	Invalid	Invalid	Available	Available	Available
SSIFCR.TIE	Invalid	Available	Available	Invalid	Available	Available
SSIFCR.RIE	Available	Invalid	Available	Available	Invalid	Available
SSIFCR.TFRST	Invalid	Available	Available	Invalid	Available	Available
SSIFCR.RFRST	Available	Invalid	Available	Available	Invalid	Available
SSIOFR.BCKASTP	Invalid	Invalid	Invalid	Available	Available	Available
SSIOFR.LRCONT	Invalid	Invalid	Invalid	Available	Available	Available
SSIOFR.OMOD	Available	Available	Available	Available	Available	Available
SSISCR.TDES	Invalid	Available	Available	Invalid	Available	Available
SSISCR.RDFS	Available	Invalid	Available	Available	Invalid	Available

"Invalid" means it has no effect on operation. Writing is possible.

46.4.1 Slave-mode Communication

SSIE operates in slave mode with SSICR.MST = 0. The SSIBCKn (n = 0, 1) and SSILRCKn/SSIFSn (n = 0, 1) signals to be used for serial-data communication must be supplied from an external device. If these signals do not match the communication format set for SSIE, operation is not guaranteed.

46.4.2 Master-mode Communication

SSIE operates in master mode with SSICR.MST = 1. The SSIBCKn (n = 0, 1) and SSILRCKn/SSIFSn (n = 0, 1) signals to be used for serial data communication must be internally generated from the audio clock. These signals use the format according to the setting of SSIE. If the communication format the slave device uses does not match the communication format set for SSIE, the operation is unpredictable.

46.4.3 Transmission

SSIE transmits serial data to the other-party device when the SSICR.TEN bit is 1 and the SSICR.REN bit is 0. If the communication format the other-party device uses does not match the communication format set for SSIE, the operation is unpredictable.

46.4.4 Reception

SSIE receives serial data from the other-party device when the SSICR.TEN bit is 0 and the SSICR.REN bit is 1. If the communication format the other-party device uses does not match the communication format set for SSIE, the operation is unpredictable.

46.4.5 Transmission and Reception

SSIE transmits and receives serial data to and from the other-party device when the SSICR.TEN bit is 1 and the SSICR.REN bit is 1. If the communication format the other-party device uses does not match the communication format set for SSIE, the operation is unpredictable.

46.5 Operation

SSIE has the following two main operation states Figure 46.43 shows SSIE state transition.

- Idle state (SSISR.IIRQ = 1)
- Communication state (SSISR.IIRQ = 0).

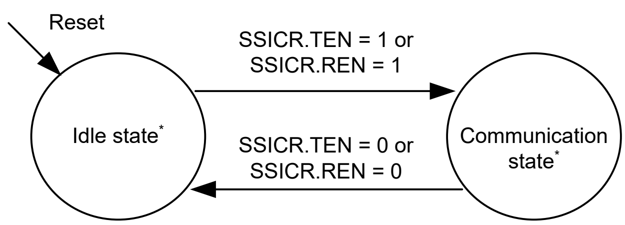


Figure 46.43 SSIE state transition

Note: See section 46.6.1. Start Communication for details of the idle state.
See section 46.6.2. Transmission for details of the communication state.

46.5.1 Idle State

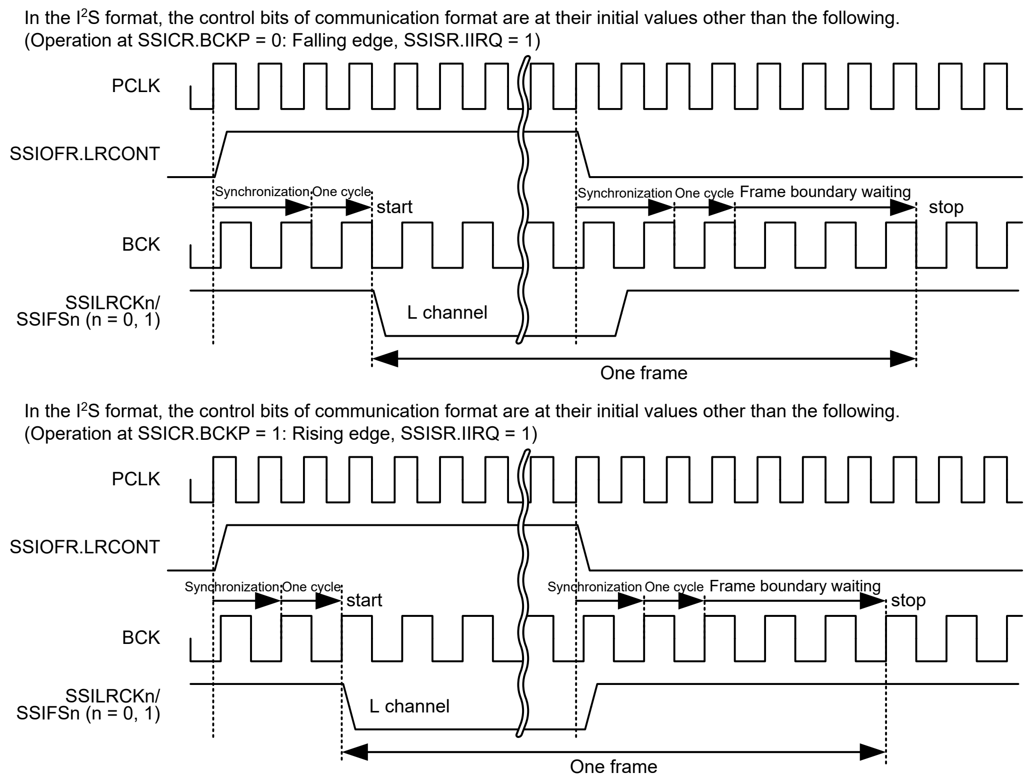
In this state, communication of SSIE is halted. If, however, the SSICR.MST bit is 1, output of the BCK and LR clock/frame synchronization signals to external pins can be controlled according to the settings of SSIOFR.BCKASTP and SSIOFR.LRCONT bits. This function is common to all formats. For details, see Table 46.16.

Table 46.16 Output from external pins in the idle state (1 of 2)

SSICR.MST	SSIOFR.BCKASTP	SSIOFR.LRCONT	Output from Pins		
			SSIBCKn (n = 0, 1)	SSILRCKn/SSIFSn (n = 0, 1)	SSITXD0/SSIDATA1
0	—	—	Stop	Stop	Stop

Table 46.16 Output from external pins in the idle state (2 of 2)

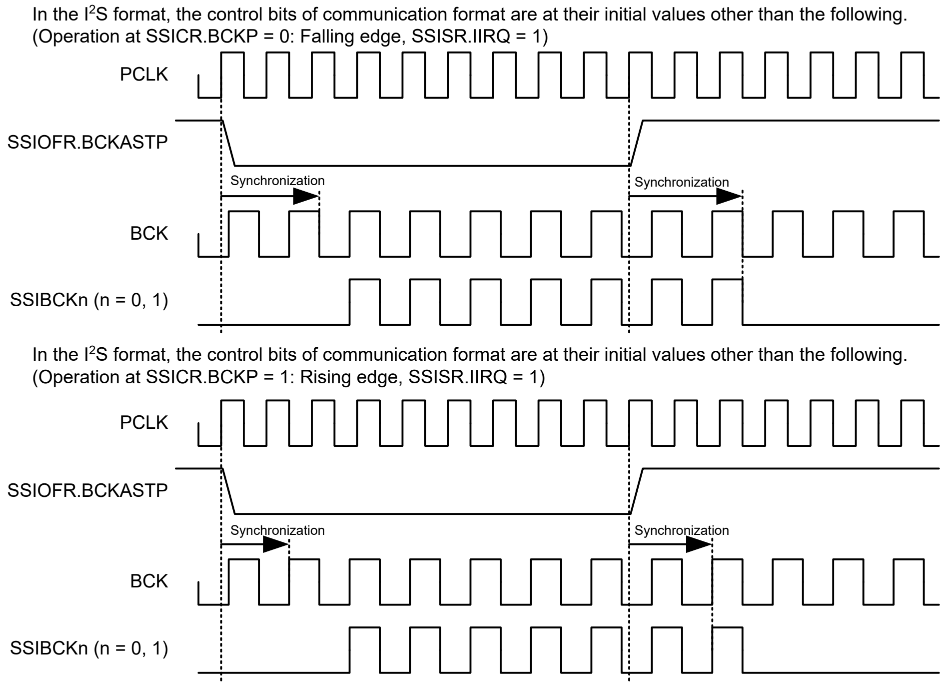
SSICR.MST	SSIOFR.BCKASTP	SSIOFR.LRCONT	Output from Pins		
			SSIBCKn (n = 0, 1)	SSILRCKn/SSIFSn (n = 0, 1)	SSITXD0/SSIDATA1
1	0	0	Supply	Stop	Stop
1	0	1	Supply	Supply	Stop
1	1	0	Stop	Stop	Stop
1	1	1	Stop	Supply	Stop



Note: PCLK = PCLKB

Figure 46.44 Example of disabling LR clock/frame synchronization continuation by SSIOFR.LRCONT

Note: To stop the output to the SSILRCKn/SSIFSn (n = 0, 1) pin with SSIOFR.LRCONT when SSIE is in the idle state in master-mode communication (SSICR.MST = 1), note the following: The output stops when the value of the SSIOFR.LRCONT bit is changed from 1 to 0. Make sure that the other-party device is not affected.



Note: PCLK = PCLKB

Figure 46.45 Example of stopping SSIBCKn (n = 0, 1) with SSIOFR.BCKASTP

Note: To stop the output to the SSIBCKn (n = 0, 1) pin with SSIOFR.BCKASTP in master-mode communication (SSICR.MST = 1) and while SSIE is in the idle state, note the following: The output stops when the value of the SSIOFR.BCKASTP bit is changed from 0 to 1. So, make sure that the other-party device is not affected.

46.5.2 Communication States

In this state, SSIE is during communication. Figure 46.46 shows transitions of communication states and Table 46.17 lists the conditions for transition. If the transition condition is not satisfied, the state does not transit.

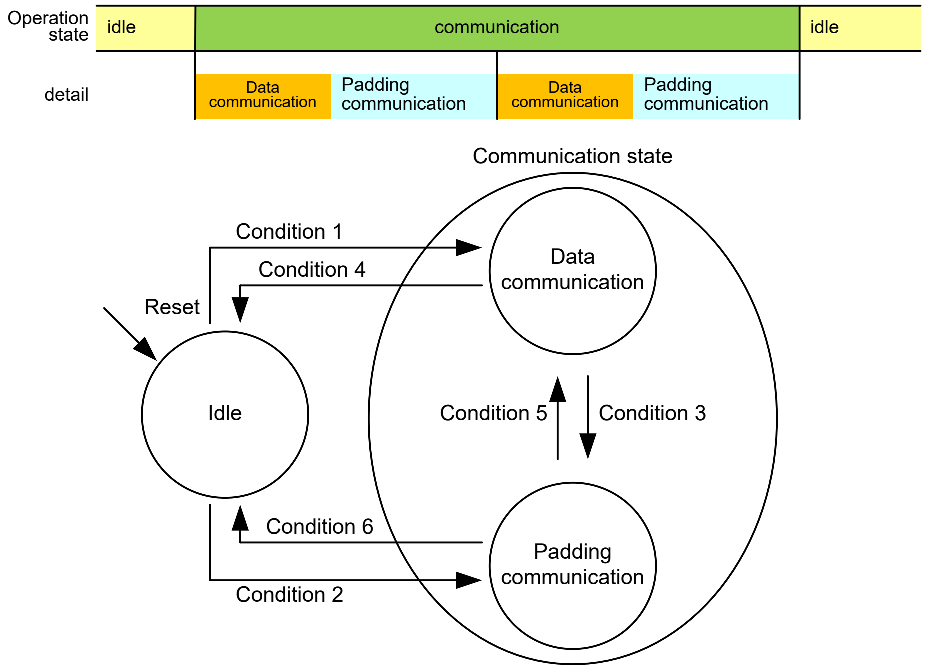


Figure 46.46 Communication state transition

Table 46.17 Condition for communication state transition

Condition Number	Condition for Transition
1	Writing SSICR.TEN = 1 or SSICR.REN = 1 while SSICR.SDTA = 0 or in the setting without padding bits.
2	Writing SSICR.TEN = 1 or SSICR.REN = 1 while SSICR.SDTA = 1 and in the setting with padding bits.
3	The following three conditions are all met: ● SSICR.TEN = 1 or SSICR.REN = 1 ● In the setting with padding bits ● The last bit of the data words has been transferred.
4	Both the following two conditions are met: ● SSICR.SDTA = 1 or without padding bits ● While SSICR.TEN = 0 and SSICR.REN = 0, the last bit of the data words in a frame has been transferred.
5	Transfer of the last padding bit is completed while SSICR.TEN = 1 or SSICR.REN = 1
6	Both the following two conditions are met: ● SSICR.SDTA = 0 and with padding bits ● While SSICR.TEN = 0 and SSICR.REN = 0, the last padding bit has been transferred.

See Table 46.13 for the setting with/without padding bits.

46.5.2.1 Data Communication State

In this state, SSIE is during communication. Data of the data word length set with the SSICR.DWL[2:0] bits is transmitted, received, or transmitted and received.

- State Transition in the Setting without Padding Bits

During communication (SSISR.IIRQ = 0), SSIE is during data communication for all the time. By disabling transmission and reception (SSICR.TEN = 0, SSICR.REN = 0), SSIE transits to the idle state. For details, see Figure 46.47 and Figure 46.48.

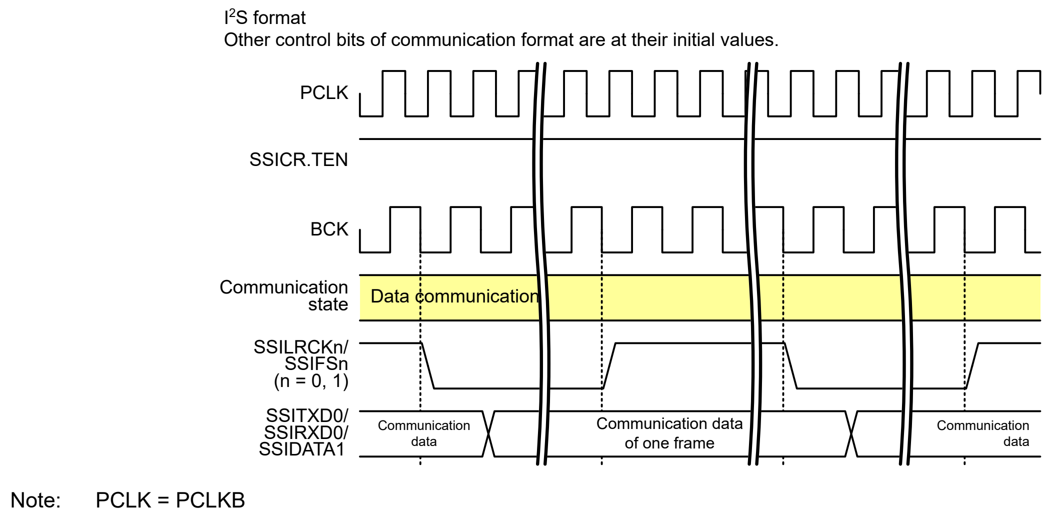


Figure 46.47 Continuation of the data communication

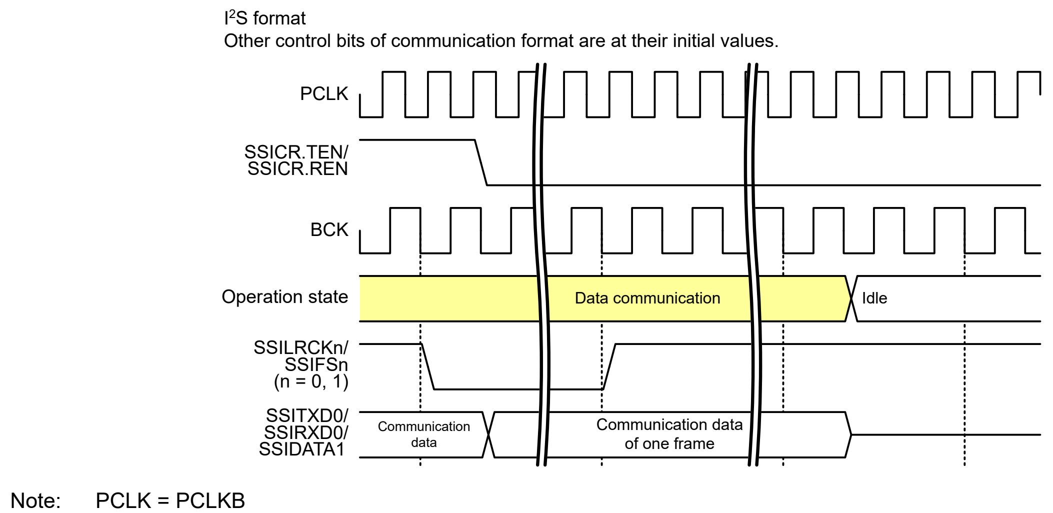
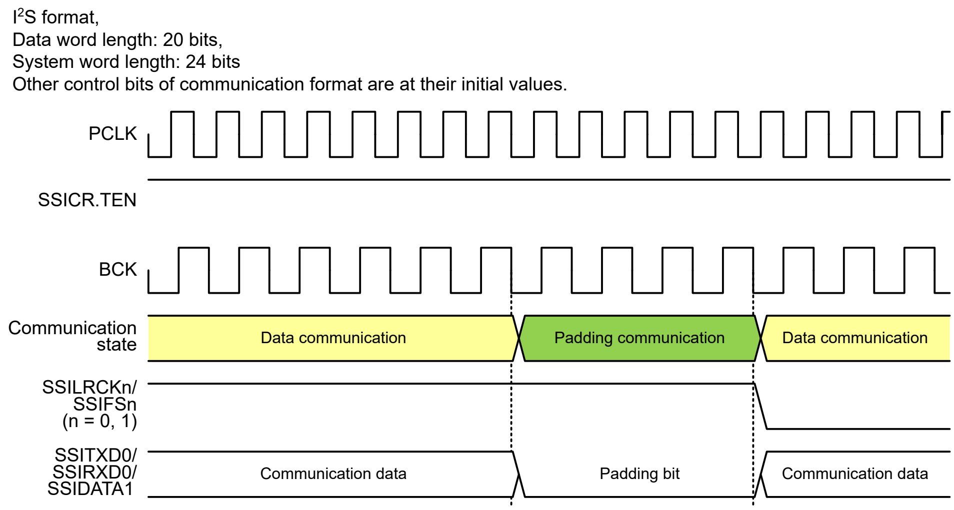


Figure 46.48 Halt from the data communication (without padding bits)

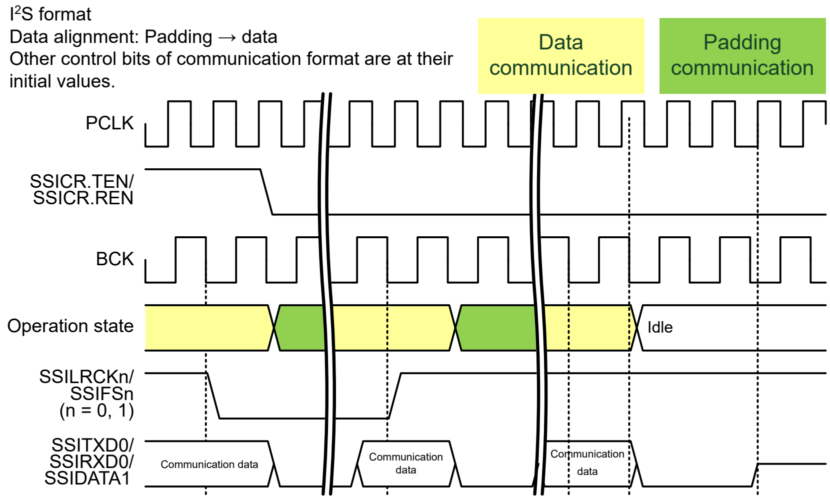
- State Transition in the Setting with Padding Bits

When SSIE ends transfer of the last bit of a data word during communication (SSISR.IIRQ = 0), SSIE transitions from the data communication state to the padding communication state in Figure 46.49. Except in the status with SSICR.SDTA = 1 and transmission and reception disabled (SSICR.TEN = 0 and SSICR.REN = 0), SSIE transitions from the data communication state to the idle state when it stops communication in Figure 46.51.



Note: PCLK = PCLKB

Figure 46.49 Transition from data communication to padding communication



Note: PCLK = PCLKB

Figure 46.50 Halt from data communication (with padding bits)

46.5.2.2 Padding Communication

In this state, SSIE is during communication. The padding bits set with the SSICR.SWL[2:0] bits and SSICR.DWL[2:0] bits are transmitted, received, or transmitted and received.

- State Transition in the Setting with Padding Bits

When SSIE ends transfer of the last padding bit during communication (SSISR.IIRQ = 0), SSIE transitions to the data communication state in Figure 46.49. If SSIE is in the status with SSICR.SDTA = 0 and transmission and reception disabled (SSICR.TEN = 0 and SSICR.REN = 0), SSIE transitions from the padding communication state to the idle state when it stops communication in Figure 46.51.

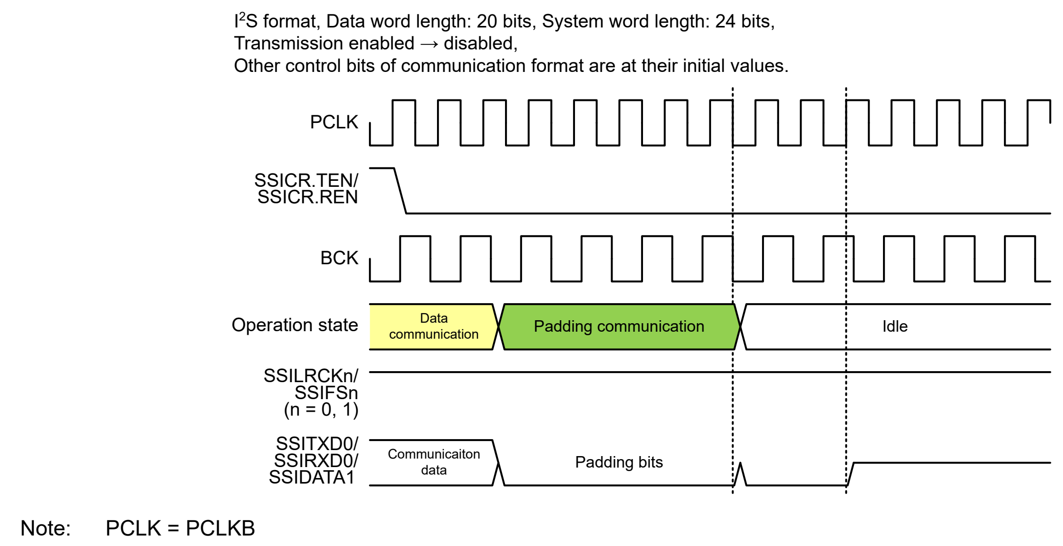


Figure 46.51 Halt from the padding communication

46.6 Communication Operation

Figure 46.52 shows the communication flow of SSIE.

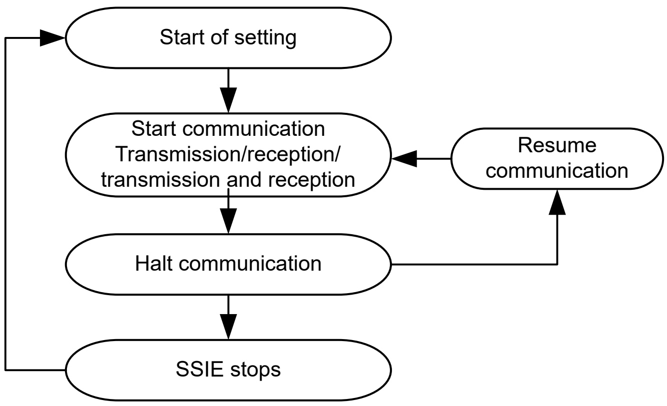
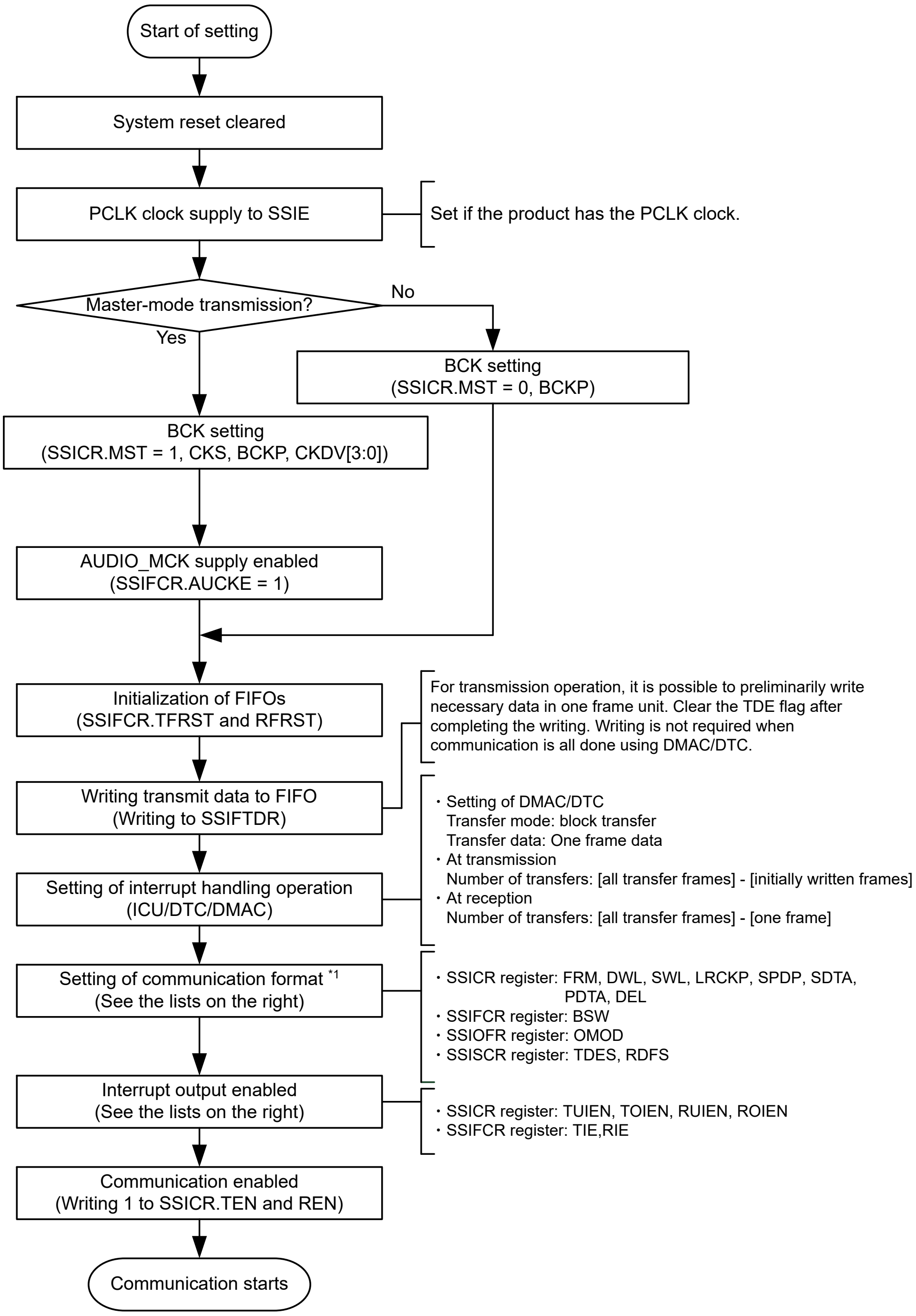


Figure 46.52 SSIE communication operation

The procedure of each operation is described from section 46.6.1. Start Communication to section 46.6.7. Resume Communication.

46.6.1 Start Communication

This section describes how to start communication of SSIE. Figure 46.53 shows the procedure to start communication. Be sure to follow the procedure. See section 46.6.2. Transmission for transmission operation and section 46.6.3. Reception for reception operation.



Note: PCLK = PCLKB
Note 1. Set SSICR.MUEN, SSIOFR.BCKASTP, and SSIOFR.LRCONT after completing the setting of the communication format.

Figure 46.53 Procedure to start communication (CPU operation procedure)

SSIE can perform continuous communication based on interrupts by the DTC/DMAC. For transmission, write 1 to SSIFCR.TIE, SSICR.TUIEN, and SSICR.TOIEN. For reception, write 1 to SSIFCR.RIE, SSICR.RUIEN, and SSICR.ROIEN.

46.6.2 Transmission

The transmission procedure in Figure 46.54 must be followed throughout a transmission operation.

After transmission is enabled (SSICR.TEN = 1 and SSICR.REN = 0), SSIE starts transmission when a start trigger is generated by SSILRCKn/SSIFSn (n = 0, 1) with the serial data for at least a frame contained in the transmit FIFO data register (SSIFTDR). SSIE outputs a transmit data empty interrupt to the DTC/DMAC according to the TDE setting condition (SSISCR.TDES) and the status of transmit data empty interrupt enable (SSIFCR.TIE) bit specified in the communication start procedure. This interrupt requests writing to the transmit FIFO data register (SSIFTDR). In the communication start procedure, specify writing to the transmit FIFO data register (SSIFTDR) as the DTC/DMAC operation in response to the transmit data empty interrupt. With this setting, SSIE can continuously transmit data not through the CPU. The transmit data empty interrupt is generated when the free space size of transmit FIFO data register reaches the value set in SSISCR.TDES. The number of times of writing must be specified in accordance with the free space size of the transmit FIFO data register indicated by the transmit data empty interrupt. If an error occurs, perform the error-handling procedure as instructed in the communication stop procedure.

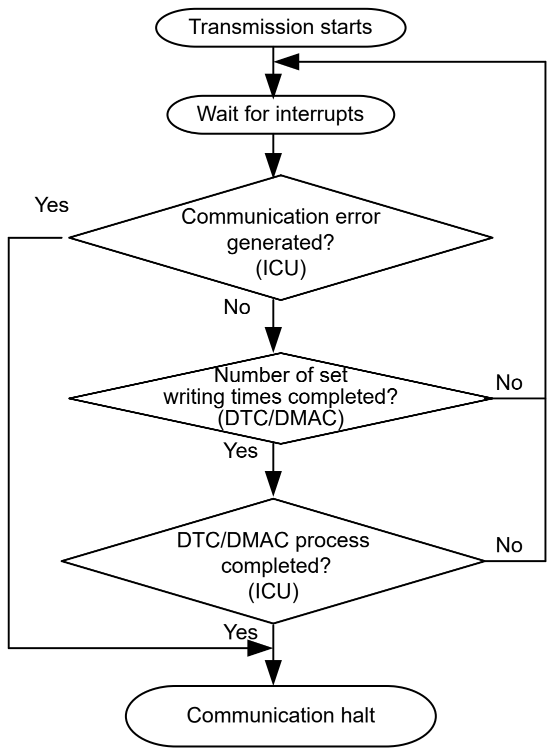


Figure 46.54 Transmission procedure

Note: The communication flow defined in SSIE uses the DTC/DMAC. If you do not use the DTC/DMAC, perform polling of the value 1 of SSIFSR.TDE to write data to SSIFTDR. The number of times of writing data to SSIFTDR by detecting the value 1 of SSIFSR.TDE must be in accordance with the free space size of the transmit FIFO data register specified by SSISCR.TDES. After as much transmit data as the free space size is written to SSIFTDR, the SSIFSR.TDE flag must be cleared. Continuous transmission is enabled by repeating data writing. If the SSIFSR.TDE flag is not cleared, the flag is not cleared automatically.

46.6.3 Reception

The reception procedure in Figure 46.55 must be followed throughout a reception operation.

After reception is enabled (SSICR.TEN = 0 and SSICR.REN = 1), SSIE starts reception when a start trigger is generated by SSILRCKn/SSIFSn (n = 0, 1). SSIE outputs a receive data full interrupt to the DTC/DMAC according to the RDF setting condition (SSISCR.RDFS) and the status of receive data full interrupt enable (SSIFCR.RIE) bit specified in the communication start procedure. This interrupt requests data reading from the receive FIFO data register (SSIFRDR). In the communication start procedure, specify reading from the receive FIFO data register (SSIFRDR) as the DTC/DMAC operation in response to the receive data full interrupt. With this setting, SSIE can continuously read data not through the CPU. The receive data full interrupt is generated when data as much as the capacity of receive FIFO data register has been stored. The number of times of reading must be specified in accordance with the data size of the receive FIFO data register indicated by the receive data full interrupt. If an error occurs, perform the error-handling procedure as instructed in the communication stop procedure.

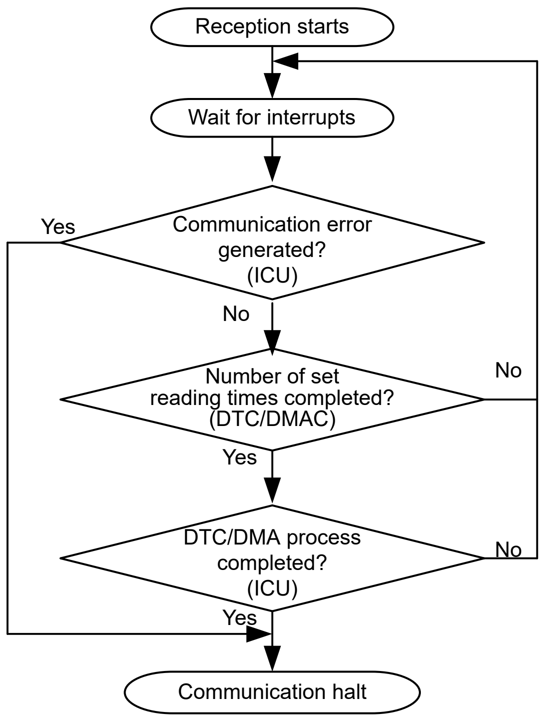


Figure 46.55 Reception procedure

Note: The communication flow defined in SSIE uses the DTC/DMAC. If you do not use the DTC/DMAC, perform polling of the value 1 of SSIFSR.RDF to read data from SSIFRDR. The number of times of reading data from SSIFRDR by detecting the value 1 of SSIFSR.RDF must be in accordance with the receive data storage capacity of the receive FIFO data register specified by SSISCR.RDFS. After received data is read from SSIFRDR, the SSIFSR.RDF flag must be cleared. Continuous reception is enabled by repeating data reading. If the SSIFSR.RDF flag is not cleared, the flag is not cleared automatically.

46.6.4 Transmission and Reception

After transmission and reception are enabled (SSICR.TEN = 1 and SSICR.REN = 1), SSIE starts transmission and reception when a start trigger is generated by SSILRCKn/SSIFSn (n = 0, 1) with the serial data for at least a frame contained in the transmit FIFO data register (SSIFTDR). SSIE can continuously transmit and receive data by performing the procedures described in section 46.6.2. Transmission and section 46.6.3. Reception, respectively. For how to stop transmission and reception, see section 46.6.5. Halt Communication.

46.6.5 Halt Communication

This section describes how to halt communication of SSIE. Figure 46.56 shows the procedure to halt communication. Be sure to follow the procedure.

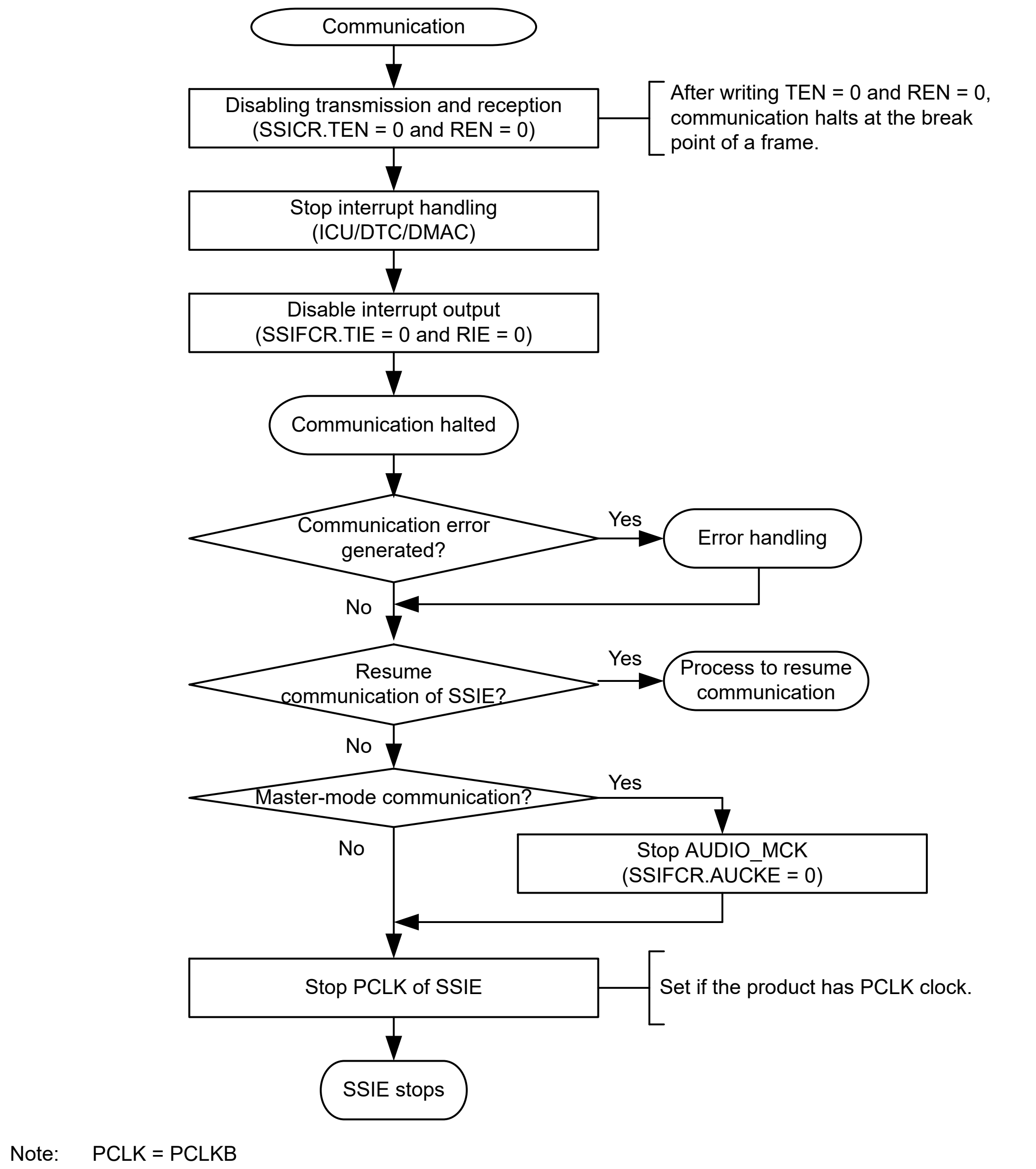


Figure 46.56 Procedure to halt communication (CPU operation procedure)

To halt the communication of SSIE, supply of the following clocks are required until the SSISR.IIRQ bit indicates an idle state.

- Input clock from the SSIBCKn (n = 0, 1) pin when SSICR.MST = 0
- AUDIO_MCK when SSICR.MST = 1
  To resume communication of SSIE in the previous setting, see section 46.6.7. Resume Communication.

Note: When communication of SSIE is halted according to the procedure to halt communication in Figure 46.56, resume communication according to the procedure to resume communication in Figure 46.58.

46.6.6 Error Handling

SSIE has the following four errors.

- Transmit underflow error
- Transmit overflow error
- Receive underflow error

- Receive overflow error.

When an underflow error or overflow error is generated, SSIE need to be restarted. Follow the procedure to halt communication in Figure 46.56 and error-handling procedure in Figure 46.57.

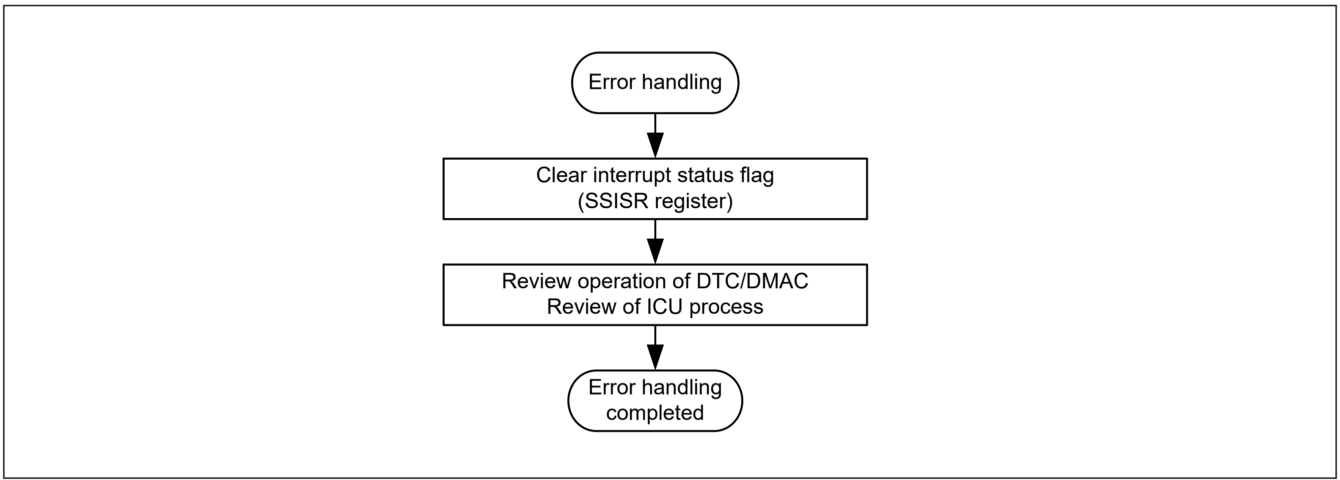


Figure 46.57 Error-handling procedure

Four error operations are described as follows. When the interrupt output enable bit of the SSICR register is enabled and error flags are set, an error interrupt is generated. See section 46.2.2. SSISR : Status Register for the setting conditions of error flags.

(1) Transmit Underflow Error

If a transmit underflow error occurs, review the number of times of writing data to the transmit FIFO data register (SSIFTDR) in response to a transmit data empty interrupt. After a transmit underflow error occurs, SSIE outputs 0s as data. To normally output the serial data written to the transmit FIFO data register (SSIFTDR) to the SSITXD0/SSIDATA1 pin, follow the procedure to halt communication in Figure 46.56 and error-handling procedure in Figure 46.57. After this error occurs, serial data is consumed as usual. If you resume communication, write the serial data from the beginning.

(2) Transmit Overflow Error

If a transmit overflow error occurs, review the number of times of writing data to the transmit FIFO data register (SSIFTDR) in response to transmit data empty interrupts. The serial data written to the transmit FIFO data register (SSIFTDR) that caused the transmit overflow error becomes invalid. This error can occur regardless of whether a transmission operation is being done. To recover from the error, follow the procedure to halt communication in Figure 46.56 and error-handling procedure in Figure 46.57. When you resume communication, deal with the invalid serial data appropriately.

(3) Receive Underflow Error

If a receive underflow error occurs, review the number of times of reading data from the receive FIFO data register (SSIFRDR) in response to receive data full interrupts. The values read from the receive FIFO data register (SSIFRDR) that caused the receive underflow error are undefined. This error can occur regardless of whether a reception operation is being done. To recover from the error, follow the procedure to halt communication in Figure 46.56 and error-handling procedure in Figure 46.57.

(4) Receive Overflow Error

If a receive overflow error occurs, review the number of times of reading data from the receive FIFO data register (SSIFRDR) in response to receive data full interrupts. The receive data that caused the receive overflow error cannot be stored in the receive FIFO data register (SSIFRDR). To recover from the error, follow the procedure to halt communication in Figure 46.56 and error-handling procedure in Figure 46.57.

46.6.7 Resume Communication

When you resume the communication using SSIE, follow the communication resume procedure in Figure 46.58. The communication resume procedure is designed on the assumption that you resume the communication stopped by the communication stop procedure without changing any settings. If you want to change clock and slave/master settings, use and follow the communication start procedure in Figure 46.53. For details about the transmission operation and reception operation after starting communication, see section 46.6.2. Transmission and section 46.6.3. Reception, respectively.

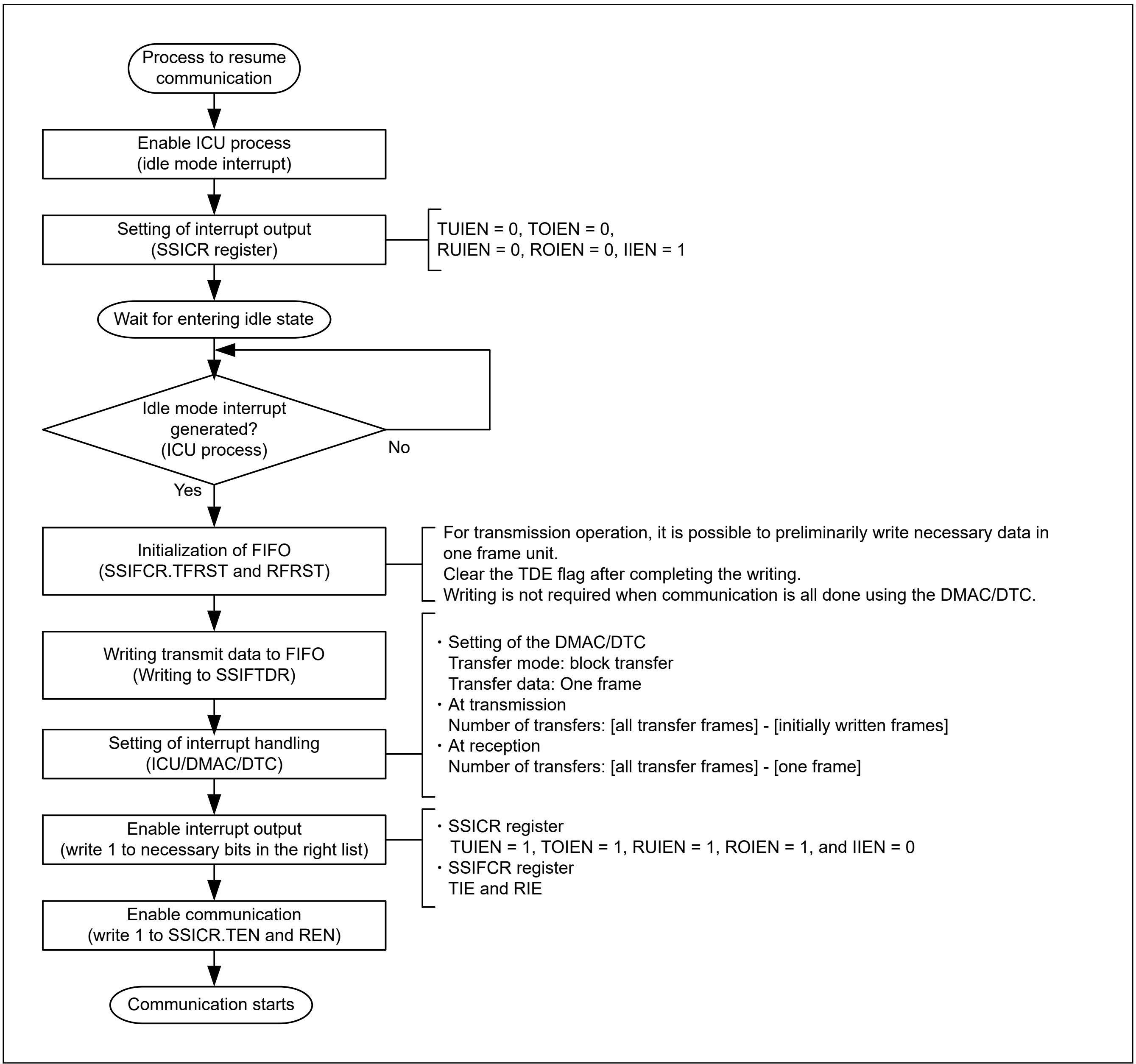


Figure 46.58 Procedure to resume communication (CPU operation procedure)

46.7 Interrupts

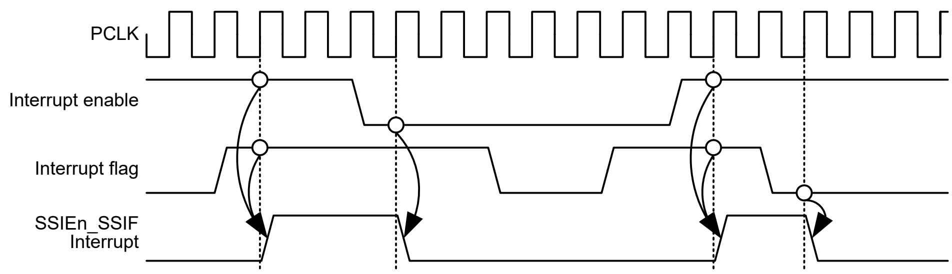
Table 46.18 lists the interrupt sources. Set enable/disable of interrupt output of each source with the TUIEN, TOIEN, RUIEN, ROIEN, and IIEN bits in the SSICR register and the TIE and RIE bits in the SSIFCR register.

Table 46.18 SSIE interrupt sources

Channel	Interrupt source	Description	Interrupt flag	DMAC/DTC activation
SSIE0	SSIE0_SSIF	• Transmit underflow interrupt • Transmit overflow interrupt • Receive underflow interrupt • Receive overflow interrupt • Idle interrupt	SSISR.TUIRQ SSISR.TOIRQ SSISR.RUIRQ SSISR.ROIRQ SSISR.IIRQ	Not possible
	SSIE0_SSIRXI	Receive data full interrupt	SSIFSR.RDF	Possible
	SSIE0_SSITXI	Transmit data empty interrupt	SSIFSR.TDE	Possible
SSIE1	SSIE1_SSIF	• Transmit underflow interrupt • Transmit overflow interrupt • Receive underflow interrupt • Receive overflow interrupt • Idle interrupt	SSISR.TUIRQ SSISR.TOIRQ SSISR.RUIRQ SSISR.ROIRQ SSISR.IIRQ	Not possible
	SSIE1_SSIRT	• Receive data full interrupt • Transmit data empty interrupt	SSIFSR.RDF SSIFSR.TDE	Possible

46.7.1 SSIEn_SSIF Interrupt (n = 0, 1)

This interrupt source combines five interrupts. Enable output of necessary interrupts before using SSIE. The five interrupts are operated by using the flags assigned to individual interrupts and interrupt output enable bits. To clear an interrupt, set the interrupt enable to 0 or clear the interrupt flag to 0.



Note: PCLK = PCLKB

Figure 46.59 Timing Diagram of the common interrupt source, SSIEn_SSIF

- Transmit underflow interrupt

As the transmit underflow interrupt, SSISR.TUIRQ is output while SSICR.TUIEN = 1. When you use SSIE for transmission, enable the output of this interrupt (SSICR.TUIRQ = 1). If this interrupt occurs, follow instructions in the procedure to halt communication in Figure 46.56 and error-handling procedure in Figure 46.57.

- Transmit overflow interrupt

As the transmit overflow interrupt, SSISR.TOIRQ is output while SSICR.TOIRQ = 1. When you use SSIE for transmission, enable the output of this interrupt (SSICR.TOIRQ = 1). If this interrupt occurs, follow instructions in the procedure to halt communication in Figure 46.56 and error-handling procedure in Figure 46.57.

- Receive underflow interrupt

As the receive underflow interrupt, SSISR.RUIRQ is output while SSICR.RUIRQ = 1. When you use SSIE for reception, enable the output of this interrupt (SSICR.RUIRQ = 1). If this interrupt occurs, follow instructions in the procedure to halt communication in Figure 46.56 and error-handling procedure in Figure 46.57.

- Receive overflow interrupt

As the receive overflow interrupt, SSISR.ROIRQ is output while SSICR.ROIRQ = 1. When you use SSIE for reception, enable the output of this interrupt (SSICR.ROIRQ = 1). If this interrupt occurs, follow instructions in the procedure to halt communication in Figure 46.56 and error-handling procedure in Figure 46.57.

- Idle mode interrupt

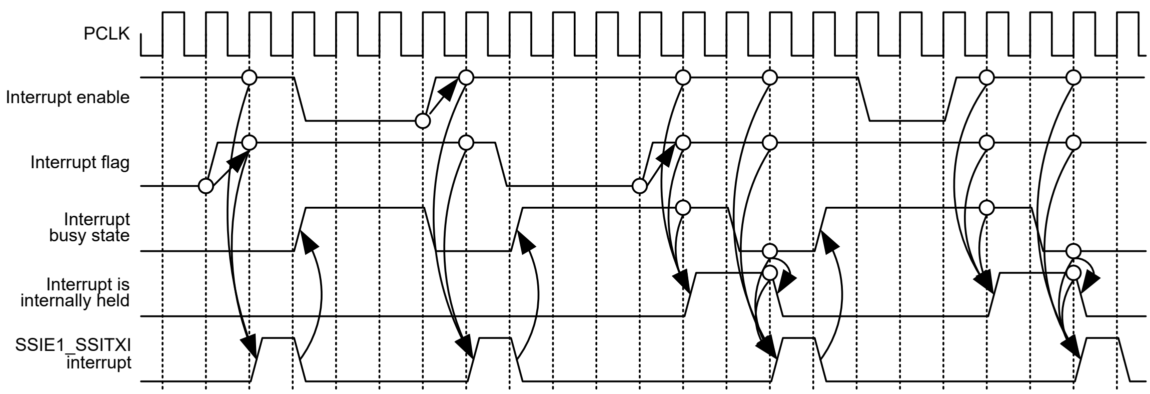
As the idle mode interrupt, SSISR.IIRQ is output while SSICR.IIEN = 1. This interrupt is used to make sure that communication has stopped fully.

46.7.2 SSIE0_SSITXI Interrupt [Full-duplex Communication]

The transmit data empty interrupt is a pulse interrupt that is output when the following condition is met:

- SSIFCR.TIE = 1 and SSIFSR.TDE = 1
  SSIE operation: When the value of SSIFSR.TDE changes from 0 to 1 while the value of SSIFCR.TIE is 1
  CPU instruction: When the value of SSIFCR.TIE changes from 0 to 1 while the value of SSIFSR.TDE is 1

This interrupt is subject to the interrupt suppression function. If an interrupt condition for this interrupt occurs when the DTC/DMAC is busy (when the DTC/DMAC cannot accept interrupts), the interrupt suppression function holds the output of this interrupt. The held interrupt will be output after the DTC/DMAC is enabled to accept interrupts. For details, see Figure 46.60.



Note: PCLK = PCLKB

Figure 46.60 SSIE1_SSITXI interrupt timing diagram

46.7.3 SSIE0_SSIRXI Interrupt [Full-duplex Communication]

The receive data full interrupt is a pulse interrupt that is output when the following condition is met:

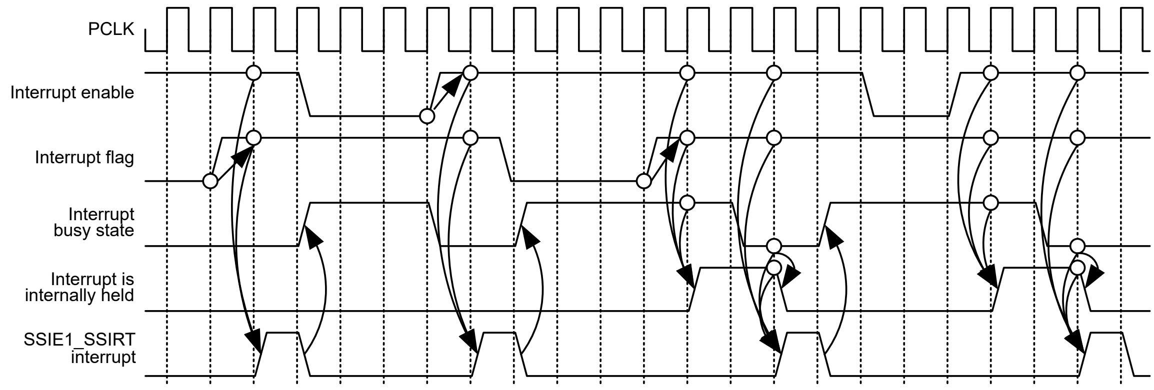
- SSIFCR.RIE = 1 and SSIFSR.RDF = 1.
  SSIE operation: When the value of SSIFSR.RDF changes from 0 to 1 while the value of SSIFCR.RIE is 1
  CPU instruction: When the value of SSIFCR.RIE changes from 0 to 1 while the value of SSIFSR.RDE is 1

This interrupt is subject to the interrupt suppression function. If an interrupt condition for this interrupt occurs when the DTC/DMAC is busy (when the DTC/DMAC cannot accept interrupts), the interrupt suppression function holds the output of this interrupt. The held interrupt will be output after the DTC/DMAC is enabled to accept interrupts. The behavior of this interrupt is the same as the behavior shown in Figure 46.60.

46.7.4 SSIE1_SSIRT Interrupt [Half-duplex Communication]

This interrupt is output by two sources, transmit data empty interrupt and receive data full interrupt. When this interrupt is generated, read the interrupt flag and specify the interrupt source.

This interrupt is subject to the interrupt suppression function. If an interrupt condition for this interrupt occurs when the DTC/DMAC is busy (when the DTC/DMAC cannot accept interrupts), the interrupt suppression function holds the output of this interrupt. The held interrupt will be output after the DTC/DMAC is enabled to accept interrupts. For details, see Figure 46.61.



Note: PCLK = PCLKB

Figure 46.61 SSIE1_SSIRT interrupt timing diagram

46.8 Software Resets

SSIE has three software reset bits to reset its states.

- SSIE software reset (SSIFCR.SSIRST)
- Transmit FIFO data register reset (SSIFCR.TFRST)
- Receive FIFO data register reset (SSIFCR.RFRST).

This section describes the procedures for the three types of software resets.

46.8.1 Software Reset Procedure

(1) SSIE Software Reset

For the SSIE software reset bit (SSIFCR.SSIRST), follow the procedure shown in Figure 46.62. After a reset, the same setting is applied when it is resumed. To change the settings of clocks and slave/master mode, follow the procedure to start communication in Figure 46.53. See section 46.6.2. Transmission and section 46.6.3. Reception respectively for transmission and reception after communication is resumed.

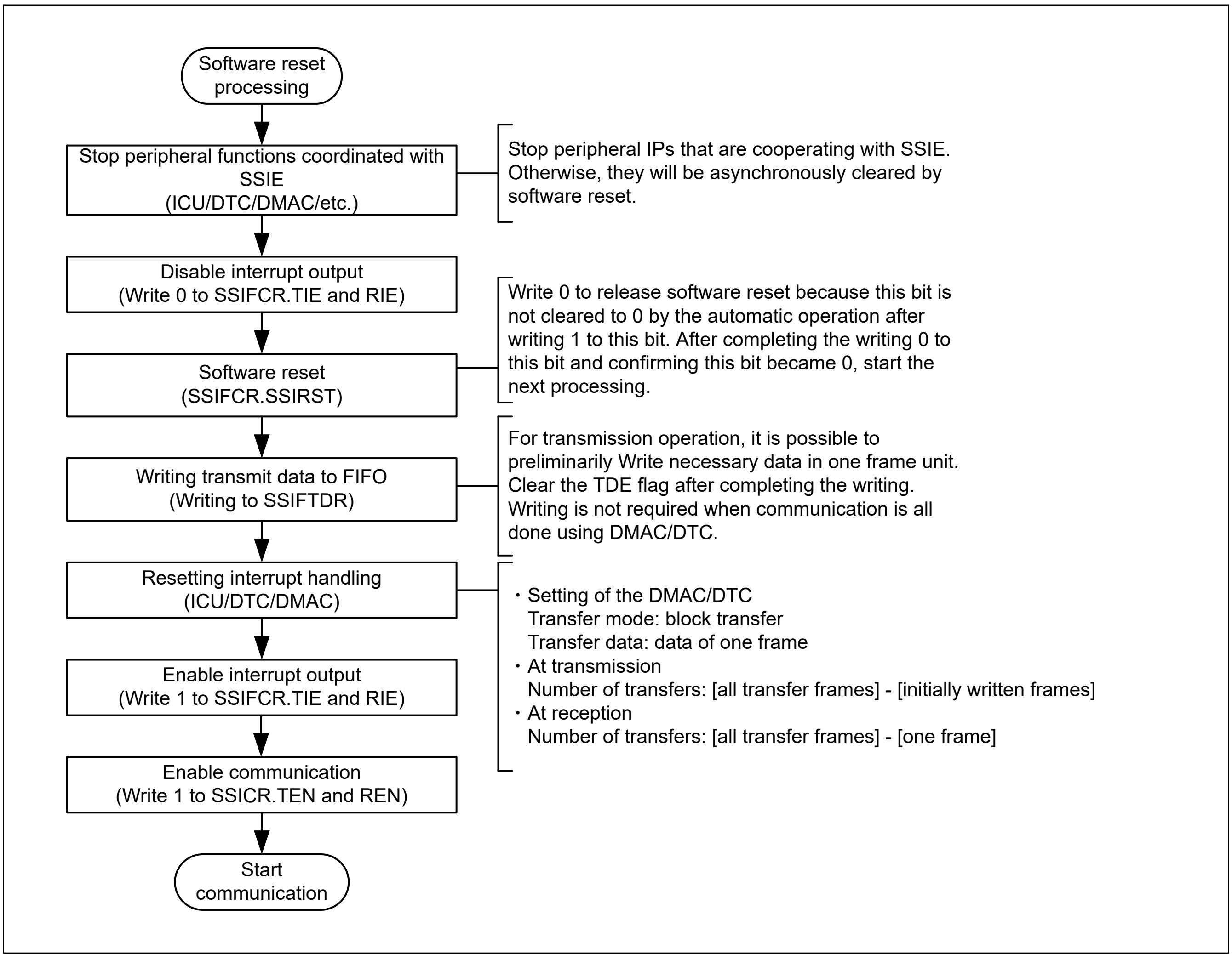


Figure 46.62 Software reset procedure (CPU operation procedure)

(2) Transmit FIFO data register reset

To perform a transmit FIFO data register reset, follow instructions in the procedure to start communication in Figure 46.53 and procedure to resume communication in Figure 46.58.

(3) Receive FIFO data register reset

To perform a receive FIFO data register reset, follow instructions in the procedure to start communication in Figure 46.53 and procedure to resume communication in Figure 46.58.

46.9 Usage Notes

46.9.1 Notes for Slave-mode Communication

46.9.1.1 SSIBCKn (n = 0, 1) Control

In slave-mode communication (SSICR.MST = 0), SSIE needs supply of SSIBCKn (n = 0, 1). To stop BCK on the master side, make sure that SSIE is in the idle state (SSISR.IIRQ = 1). If BCK is stopped before SSIE becomes idle, take the procedure to start communication in Figure 46.53 or wait for an idle state by taking the procedure to resume communication in Figure 46.58.

46.9.1.2 SSILRCKn/SSIFSn (n = 0, 1) pin

SSIE has the SSILRCKn/SSIFSn (n = 0, 1) pin, which indicates the synchronization of communication. When SSIE is in slave mode (SSICR.MST = 0), the communication format SSIE uses must match that of the other-party device to communicate. SSIE uses the signal input by the SSILRCKn/SSIFSn (n = 0, 1) pin only as a trigger to start communication.

46.9.2 Notes for Master-mode Communication

46.9.2.1 AUCKE Control

In master-mode communication (SSICR.MST = 1), SSIE operates with the audio clock (AUDIO_MCK). To stop SSIE completely, make sure that SSIE is in the idle state (SSISR.IIRQ = 1) and then write 0 to SSIFCR.AUCKE.

46.9.2.2 LRCONT Control

To stop the output to the SSILRCKn/SSIFSn (n = 0, 1) pin with SSIOFR.LRCONT when SSIE is in the idle state in master-mode communication (SSICR.MST = 1), note the following: The output stops when the value of the SSIOFR.LRCONT bit is changed from 1 to 0. Make sure that the other-party device is not affected. For details, see Figure 46.44.

46.9.2.3 BCKASTP Control

To stop the output to the SSIBCKn (n = 0, 1) pin with SSIOFR.BCKASTP in master-mode communication (SSICR.MST = 1) and while SSIE is in the idle state, note the following: The output stops when the value of the SSIOFR.BCKASTP bit is changed from 0 to 1. So, make sure that the other-party device is not affected. For details, see Figure 46.45.

The BCKASTP bit cannot be used when the other-party device (which is a slave) requires the clock output from the SSIBCKn (n = 0, 1) pin before and during communication.

46.9.3 Notes for Communication Flow

46.9.3.1 When an Error Interrupt is Generated

SSIE has the following four errors.

- Transmit underflow error
- Transmit overflow error
- Receive underflow error
- Receive overflow error

When an underflow error or overflow error is generated, SSIE need to be restarted. Follow the procedure to halt communication in Figure 46.56 and error-handling procedure in Figure 46.57.

(1) Transmit Underflow Error

If a transmit underflow error occurs, review the number of times of writing data to the transmit FIFO data register (SSIFTDR) in response to a transmit data empty interrupt. After a transmit underflow error occurs, SSIE outputs 0s as data. To normally output the serial data written to the transmit FIFO data register (SSIFTDR) to the SSITXD0/SSIDATA1 pin, follow the procedure to halt communication in Figure 46.56 and error-handling procedure in Figure 46.57. After this error occurs, serial data is consumed as usual. If you resume communication, write the serial data from the beginning.

(2) Transmit Overflow Error

If a transmit overflow error occurs, review the number of times of writing data to the Transmit FIFO Data Register (SSIFTDR) in response to transmit data empty interrupts. The serial data written to the Transmit FIFO Data Register (SSIFTDR) that caused the transmit overflow error becomes invalid. This error can occur regardless of whether a transmission operation is being done. To recover from the error, follow the procedure to halt communication in Figure 46.56 and error-handling procedure in Figure 46.57. When you resume communication, deal with the invalid serial data appropriately.

(3) Receive Underflow Error

If a receive underflow error occurs, review the number of times of reading data from the receive FIFO data register (SSIFRDR) in response to receive data full interrupts. The values read from the receive FIFO data register (SSIFRDR) that caused the receive underflow error are undefined. This error can occur regardless of whether a reception operation is being done. To recover from the error, follow the procedure to halt communication in Figure 46.56 and error-handling procedure in Figure 46.57.

(4) Receive Overflow Error

If a receive overflow error occurs, review the number of times of reading data from the receive FIFO data register (SSIFRDR) in response to receive data full interrupts. The receive data that caused the receive overflow error cannot be stored in the receive FIFO data register (SSIFRDR). To recover from the error, follow the procedure to halt communication in Figure 46.56 and error-handling procedure in Figure 46.57.

46.9.3.2 Transmit Data Empty Interrupt

The communication flow defined in SSIE uses the DTC/DMAC. If you do not use the DTC/DMAC, perform polling of the value 1 of SSIFSR.TDE to write data to SSIFTDR. The number of times of writing data to SSIFTDR by detecting the value 1 of SSIFSR.TDE must be in accordance with the free space size of the transmit FIFO data register specified by SSISCR.TDES. After as much transmit data as the free space size is written to SSIFTDR, the SSIFSR.TDE flag must be cleared. Continuous transmission is enabled by repeating data writing. If the SSIFSR.TDE flag is not cleared, the flag is not cleared automatically.

46.9.3.3 Receive Data Full Interrupt

The communication flow defined in SSIE uses the DTC/DMAC. If you do not use the DTC/DMAC, perform polling of the value 1 of SSIFSR.RDF to read data from SSIFRDR. The number of times of reading data from SSIFRDR by detecting the value 1 of SSIFSR.RDF must be in accordance with the receive data storage capacity of the receive FIFO data register specified by SSISCR.RDFS. After received data is read from SSIFRDR, the SSIFSR.RDF flag must be cleared. Continuous reception is enabled by repeating data reading. If the SSIFSR.RDF flag is not cleared, the flag is not cleared automatically.

46.9.3.4 Switching Transfer Modes

1. For state transition from transmission, reception, and transmission and reception, disable transmission and reception (SSICR.TEN = 0, SSICR.REN = 0).
2. Confirm it is in the idle state (SSISR.IIRQ = 1).
3. In the idle state, set the SSICR.TEN bit or the SSICR.REN bit again and resume transfer.

46.9.3.5 Resume Communication After Halting SSIE

When communication of SSIE is halted according to the procedure to halt communication in Figure 46.56, resume communication according to the procedure to resume communication in Figure 46.58.

46.9.4 Write Access Restriction

46.9.4.1 SSICR Register

If the TEN bit or REN bit is rewritten, make sure that the SSISR.IIRQ bit is in the desired status. If the value of the TEN or REN bit is changed by rewriting, subsequent operation is unpredictable. For example, when transmission or reception is enabled, check that SSISR.IIRQ is 0; when transmission or reception is disabled, check that SSISR.IIRQ is 1.

(1) TEN Bit and REN Bit

These bits enable/disable transmission and reception. When 1 is written to one of these bits, the corresponding communication operation starts in synchronization with a start trigger by the SSILRCKn/SSIFSn (n = 0, 1) signal. For details, see section 46.6.2. Transmission, section 46.6.3. Reception, and section 46.6.4. Transmission and Reception. When 0 is written to this bit, the current communication operation stops at the next frame boundary. To use SSIE for both transmission and reception, always write 1 to these bits together. When stopping the communication using SSIE, always disable both transmission and reception (write 0 to the TEN and REN bits).

46.9.4.2 SSISR Register

(1) Clearing TUIRQ and TOIRQ

After communication is enabled (by changing the value of SSICR.TEN bit from 0 to 1), the transmission error flags (TOIRQ and TUIRQ in the SSISR register) are cleared. If, however, the SSISR register is read continuously, the cleared status of the transmission error flags might be unable to be read.

(2) Clearing RUIRQ and ROIRQ

After communication is enabled (by changing the value of SSICR.REN bit from 0 to 1), the reception error flags (RUIRQ and ROIRQ in the SSISR register) are cleared. If, however, the SSISR register is read continuously, the cleared status of the reception error flags might be unable to be read.

46.9.4.3 Communication State

Writing to the bits with orange-shaded area in Table 46.19 is prohibited. If written, the operation performed immediately after writing is not guaranteed.

Table 46.19 Bits protected from writing during communication

Symbol	Address (BASE+)		+0								+1							
			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSICR	0x00	+0	—	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	—	FRM[1:0]		DWL[2:0]			SWL[2:0]		
		+2	—	MST	BCKP	LRCKP	SPDP	SDTA	PDTA	DEL	CKDV[3:0]				MUEN	—	TEN	REN
SSISR	0x04	+0	—	—	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	—	—	—	—	—	—	—	—	—
		+2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SSIFCR	0x10	+0	AUCKE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSIRST
		+2	—	—	—	—	BSW	—	—	—	—	—	—	—	TIE	RIE	TFRST	RFRST
SSIFSR	0x14	+0	—	—	TDC[5:0]						—	—	—	—	—	—	—	TDE
		+2	—	—	RDC[5:0]						—	—	—	—	—	—	—	RDF
SSIFTDR	0x18	+0	FTDR[31:16]															
		+2	FTDR[15:0]															
SSIFRDR	0x1C	+0	FRDR[31:16]															
		+2	FRDR[15:0]															
SSIOFR	0x20	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	—	—	—	BCKASTP	LRCONT	—	—	—	—	—	—	OMOD[1:0]	
SSISCR	0x24	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	TDES[4:0]					—	—	—	RDFS[4:0]				

46.9.5 Software Standby Mode

The SSIE must halt before transition to Software Standby mode.

Figure 46.63 shows an example flow before transition to Software Standby mode.

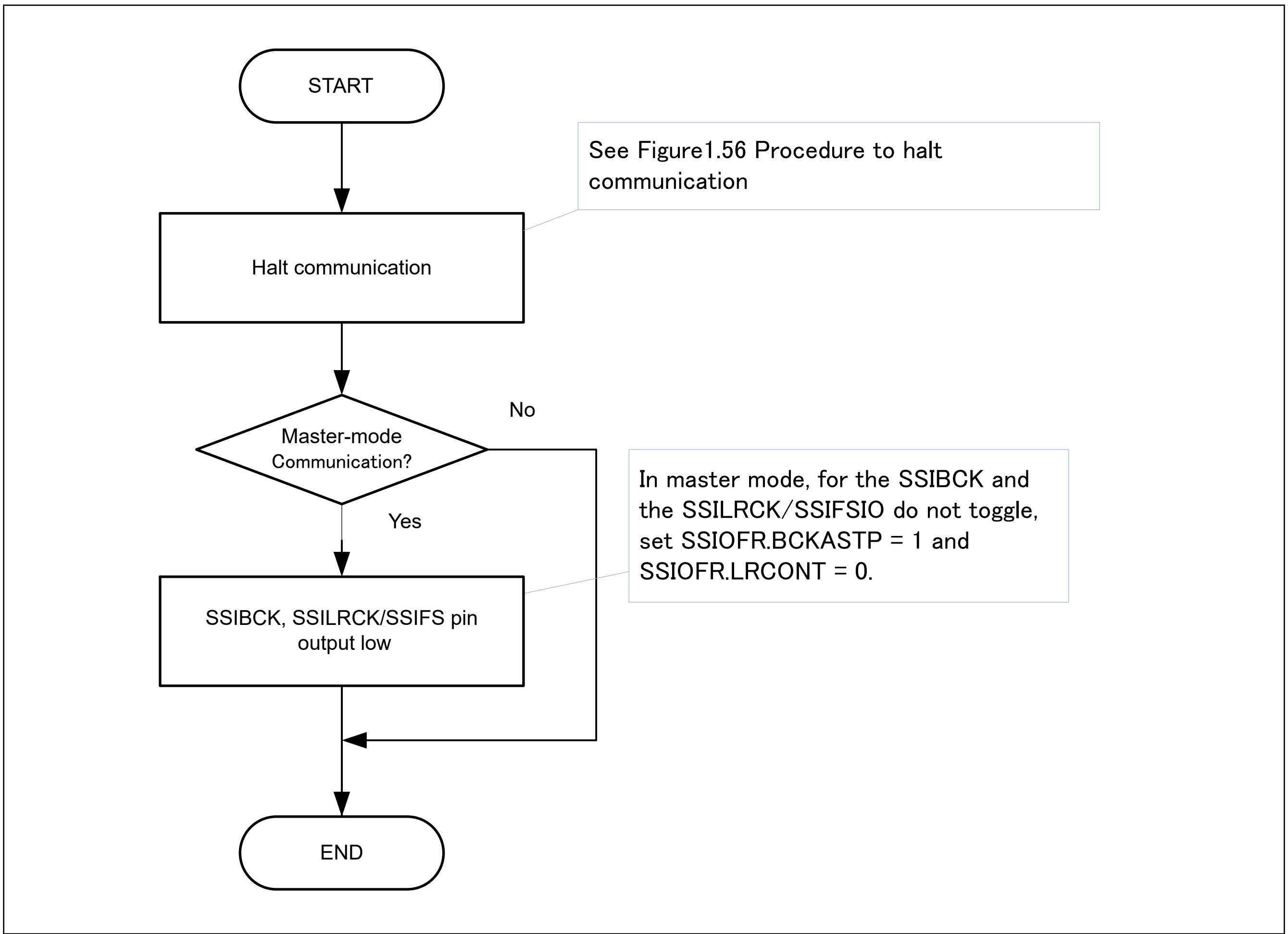


Figure 46.63 Example flow before transition to Software Standby mode

Figure 46.64 shows an example flow after canceling Software Standby mode.

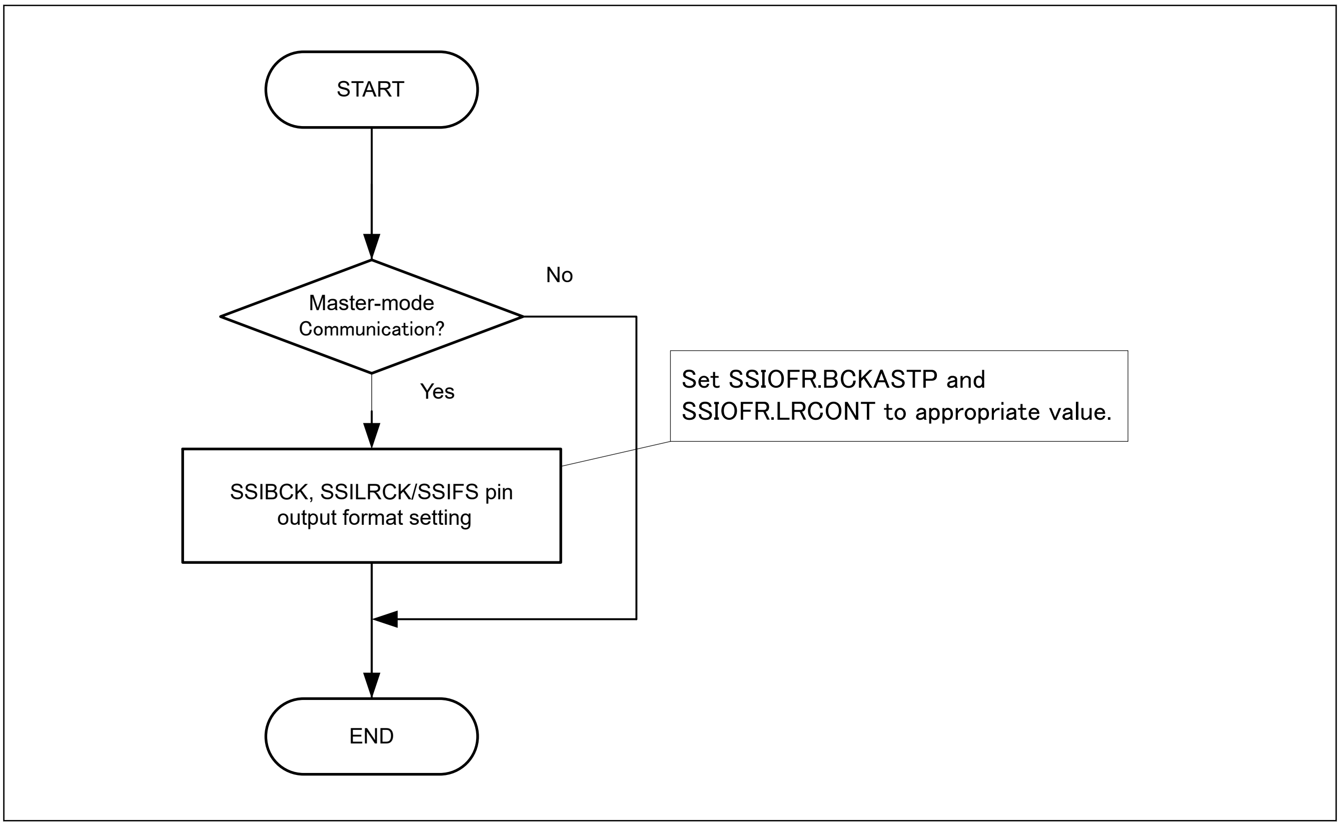


Figure 46.64 Example flow after canceling Software Standby mode

47. SD/MMC Host Interface (SDHI)

47.1 Overview

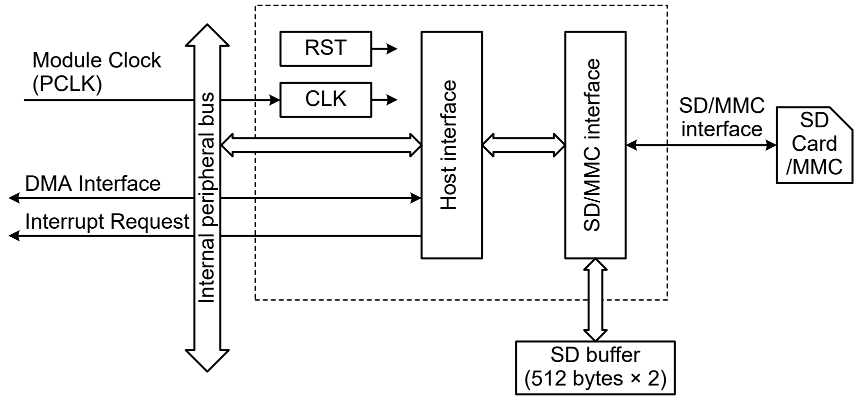
The Secure Digital Host Interface (SDHI) and MultiMediaCard (MMC) Interface provide the functionality required to connect a variety of external memory cards with the MCU. The SDHI supports both 1- and 4-bit buses for connecting different memory cards that support SD, SDHC, and SDXC formats. When developing host devices that are compliant with the SD Specifications, you must comply with the SD Host/Ancillary Product License Agreement (SD HALA).

The MMC interface supports 1-bit, 4-bit, and 8-bit MMC buses that provide eMMC 4.51 (JEDEC Standard JESD 84- B451) device access. This interface also provides backward compatibility and supports for high-speed SDR transfer modes.

Table 47.1 lists the SD/MMC Host Interface specifications and Figure 47.1 shows a block diagram.

Table 47.1 SD/MMC Host Interface specifications

Interface	Parameter	Specifications
SD	SD bus interface	● Compatible with SD memory card and SDIO card ● Transfer bus mode selectable from 4-bit wide bus mode or 1-bit default bus mode ● Compatible with SD, SDHC, and SDXC formats
	Transfer modes	Default Speed mode, High Speed mode, SDR12 or SDR25
SD and MMC shared	SDHI clock frequency	The SDHI clock is generated by dividing PCLKB by 2^n (n = 0 to 9)
	Error check functions	CRC7 (command/response), CRC16 (transfer data)
	Interrupt sources	Card access interrupt (SDHI_MMCn_ACCS), SDIO access interrupt (SDHI_MMCn_SDIO), Card detection interrupt (SDHI_MMCn_CARD) (n = 0, 1)
	DMA transfer sources	DMAC and DTC triggerable by the DMA transfer requests (SDHI_MMCn_ODMSDBREQ (n = 0, 1)) interrupt SD buffer is read and write accessible using the DMAC
	Other functions	● Card detect function ● Write protect support
MMC	MMC bus interface	Transfer bus mode selectable from 1-bit, 4-bit, or 8-bit
	Transfer modes	Backward compatible mode or high-speed SDR mode selectable
	Other functions	e.MMC device access supported
Module-stop function	Module-stop state can be set for each channel to reduce power consumption.	
TrustZone Filter	Security and Privilege attribution can be set for each channel.	



Note: PCLK = PCLKB

Figure 47.1 SD/MMC Host Interface block diagram

Table 47.2 SDHI I/O pins (n = 0, 1)

Channel	Pin name	I/O	Description
Ch n	SDnCLK	Output	SDHI clock
	SDnCMD	I/O	Command output, response input
	SDnDAT0	I/O	Data 0 (DAT0)
	SDnDAT1	I/O	Data 1 (DAT1), SDIO interrupt
	SDnDAT2	I/O	Data 2 (DAT2), SDIO Read wait
	SDnDAT3	I/O	Data 3 (DAT3), SD Card detect
	SDnDAT4	I/O	MMC Data 4 (DAT4)
	SDnDAT5	I/O	MMC Data 5 (DAT5)
	SDnDAT6	I/O	MMC Data 6 (DAT6)
	SDnDAT7	I/O	MMC Data 7 (DAT7)
	SDnCD	Input	SD card detection
	SDnWP	Input	SD card write protection

47.2 Register Descriptions

47.2.1 SD_CMD : Command Type Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMD12AT[1:0]		TRSTP	CMDRW	CMDTP	RSPTP[2:0]			ACMD[1:0]		CMDIDX[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	CMDIDX[5:0]	Command Index Field Value Select These bits configure the command index field value. The examples shown include the bit values for the ACMD[1:0] bits. 0x06: CMD6 0x12: CMD18 0x4D: ACMD13	R/W
7:6	ACMD[1:0]	Command Type Select 0 0: CMD 0 1: ACMD Others: Setting prohibited	R/W
10:8	RSPTP[2:0]	Response Type Select*1 0 0 0: Normal mode Depending on the command, the response type and transfer method are selected in the ACMD[1:0] and CMDIDX[5:0] bits. At this time, the values for bits 15 to 11 in this register are invalid. 0 1 1: Extended mode and no response 1 0 0: Extended mode and R1, R5, R6, or R7 response 1 0 1: Extended mode and R1b response 1 1 0: Extended mode and R2 response 1 1 1: Extended mode and R3 or R4 response Others: Setting prohibited	R/W

Bit	Symbol	Function	R/W
11	CMDTP	Data Transfer Select*2 0: Do not include data transfer (bc, bcr, or ac) in command 1: Include data transfer (adtc) in command	R/W
12	CMDRW	Data Transfer Direction Select*3 0: Write (SD/MMC Host Interface → SD card/MMC) 1: Read (SD/MMC Host Interface ← SD card/MMC)	R/W
13	TRSTP	Block Transfer Select*3 0: Single block transfer 1: Multiple blocks transfer	R/W
15:14	CMD12AT[1:0]	CMD12 Automatic Issue Select*4 0 0: Automatically issue CMD12 during multiblock transfer 0 1: Do not automatically issue CMD12 during multiblock transfer Others: Setting prohibited	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Some commands cannot be used in normal mode.
Note 2. The CMDTP bit is only valid when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b.
Note 3. Bits CMDRW and TRSTP are only valid when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b, and the CMDTP bit is 1.
Note 4. The CMD12AT[1:0] bits are only valid when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b, and the TRSTP bit is 1.

The command type and response type are set in the SD_CMD register. The command type and transfer mode must be set when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b. The sequence starts when a value is written to this register. See Table 47.8 and Table 47.9 for setting examples. Do not write to the SD_CMD register when the SD_INFO2.CBSY flag is 1.

47.2.2 SD_ARG : SD Command Argument Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	—	These bits specify command format[39:8] (argument).	R/W

Note: S-TYPE-3, P-TYPE-3

The SD_ARG register is used for setting the argument field value. Set the SD_ARG register before setting the SD_CMD register. The argument field value of the automatically issued CMD12 is 0x0000_0000 regardless of the SD_ARG register value.

47.2.3 SD_ARG1 : SD Command Argument Register 1

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x00C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits specify command format[39:24] (argument).	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The SD_ARG1 register is used for setting the argument field value. Set the SD_ARG1 register before setting the SD_CMD register. The argument field value of the automatically issued CMD12 is 0x0000_0000 regardless of the SD_ARG1 register value.

47.2.4 SD_STOP : Data Stop Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SEC	—	—	—	—	—	—	—	STP
Value after reset:	0	0	0	0	0	0	0	0*1	0	0	0	0	0	0	0	0*1

Bit	Symbol	Function	R/W
0	STP	Transfer Stop Data transfer stops when this bit is set to 1.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	SEC	Block Count Register Value Select*2 0: Disable SD_SECCNT register value 1: Enable SD_SECCNT register value	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The value is initialized by a reset and also on reset triggered by the SOFT_RST.SDRST flag.
Note 2. Do not rewrite this bit when the SD_INFO2.CBSY flag is 1.

The SD_STOP register stops data transfer. During a multiblock transfer sequence, the SD_SECCNT register value (number of blocks to be transferred) can be set to valid or invalid by setting the SD_STOP register.

STP bit (Transfer Stop)

When the STP bit is set to 1 during multiple block transfer, CMD12 is issued to halt the transfer through the SDHI. However, if a command sequence is halted because of a communications error or timeout, CMD12 is not issued. Although continued buffer access is possible even after STP is set to 1, the buffer access error bit (ILR or ILW) in SD_INFO2 is set accordingly.

When STP is set to 1 during transfer for single block write, the access end flag sets when SD_BUF becomes empty, and CMD12 is not issued. If SD_BUF does contain data, the access end flag sets on completion of reception of the busy state without CMD12 being issued.

When STP is set to 1 during transfer for single block read, the access end flag sets immediately after the STP bit is set, and CMD12 is not issued.

When STP is set to 1 during reception of the busy state after an R1b response, the access end flag sets on completion of reception of the busy state without CMD12 being issued.

When STP is set to 1 after a command sequence is completed, CMD12 is not issued and the access end flag does not set.

Set STP to 1 after the response end flag sets.

Set STP to 0 after the access end flag sets.

SEC bit (Block Count Register Value Select)

When SD_CMD is set in the following section to start the command sequence while the SEC bit is set to 1, CMD12 is automatically issued to stop multiblock transfer with the number of blocks set in SD_SECCNT.

CMD18 or CMD25 in normal mode (SD_CMD[10:8] = 000b)

SD_CMD[15:13] = 001b in extended mode (CMD12 is automatically issued, multiple block transfer)

When the command sequence is halted because of a communications error or timeout, CMD12 is not automatically issued.

47.2.5 SD_SECCNT : Block Count Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	—	These bits set the number of blocks to be transferred.	R/W

Note: S-TYPE-3, P-TYPE-3

When performing a multiblock transfer, SD_SECCNT is a read/write register used to set the number of blocks to be transferred. For example, when the register value is 0x0000_0001, 1 block is transferred. When the register value is 0x0000_FFFF, 65,535 blocks are transferred and when the register value is 0xFFFF_FFFF, 4,294,967,295 blocks are transferred. Do not set this register to 0x0000_0000. Do not rewrite the SD_SECCNT register when the SD_INFO2.CBSY flag is 1.

47.2.6 SD_RSPi : SD Card Response Register i (i = 10, 32, 54)

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x018 (SD_RSP10)
0x020 (SD_RSP32)
0x028 (SD_RSP54)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	—	These bits store the response from the SD card/MMC.	R

Note: S-TYPE-3, P-TYPE-3

47.2.7 SD_RSPj : SD Card Response Register j (j = 1, 3, 5)

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x01C (SD_RSP1)
0x024 (SD_RSP3)
0x02C (SD_RSP5)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits store the response from the SD card/MMC.	R
31:16	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

47.2.8 SD_RSP76 : SD Card Response Register 76

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x030

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	SD_RSP76[23:16]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SD_RSP76[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
23:0	SD_RSP76[23:0]	These bits store the response from the SD card/MMC.	R
31:24	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

47.2.9 SD_RSP7 : SD Card Response Register 7

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x034

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	SD_RSP7[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	SD_RSP7[7:0]	These bits store the response from the SD card/MMC.	R
31:8	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

SD_RSP10, SD_RSP32, SD_RSP54, SD_RSP1, SD_RSP3, SD_RSP5, SD_RSP76, and SD_RSP7 are read-only registers that store the response from the SD card/MMC. Depending on the type of response from the SD card/MMC, the SD/MMC Host Interface divides and stores the response among the four registers.

Table 47.3 lists the correspondence between the response type and its storage destination.

Table 47.3 Correspondence between response type and storage destination (1 of 2)

Response type	SD_RSP10 register	SD_RSP32 register	SD_RSP54 register	SD_RSP1 register	SD_RSP3 register	SD_RSP5 register	SD_RSP76 register	SD_RSP7 register
R1	[39:8]	—	[39:8]*1	—	—	—	—	—
R1b	[39:8]	—	[39:8]*1	—	—	—	—	—
R2	[39:8]	[71:40]	[103:72]	—	—	—	[127:104]	—
R3	[39:8]	—	—	—	—	—	—	—
R4	[39:8]	—	—	—	—	—	—	—

Table 47.3 Correspondence between response type and storage destination (2 of 2)

Response type	SD_RSP10 register	SD_RSP32 register	SD_RSP54 register	SD_RSP1 register	SD_RSP3 register	SD_RSP5 register	SD_RSP76 register	SD_RSP7 register
R5	[39:8]	—	—	—	—	—	—	—
R6	[39:8]	—	—	—	—	—	—	—
R7	[39:8]	—	—	—	—	—	—	—

Note 1. The responses for CMD18 and CMD25 are stored in registers SD_RSP10 and SD_RSP54. Therefore, even if the SD_RSP10 register is overwritten with the response for the automatically issued CMD12, the response for CMD18 or CMD25 can be confirmed by reading the SD_RSP54 register.

47.2.10 SD_INFO1 : SD Card Interrupt Flag Register 1

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x038

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SDD3MON	SDD3IN	SDD3RM	SDWPMON	—	SDCDMON	SDCDIN	SDCDRM	ACEND	—	RSPEND
Value after reset:	0	0	0	0	0	x	0	0	x	0	x	0	0	0*1	0	0*1

Bit	Symbol	Function	R/W
0	RSPEND	Response End Detection Flag 0: Response end not detected 1: Response end detected	R/(W)*2
1	—	This bit is read as 0. The write value should be 0.	R/W
2	ACEND	Access End Detection Flag 0: Access end not detected 1: Access end detected	R/(W)*2
3	SDCDRM	SDnCD Removal Flag 0: SD card/MMC removal not detected by the SDnCD pin 1: SD card/MMC removal detected by the SDnCD pin	R/(W)*2
4	SDCDIN	SDnCD Insertion Flag 0: SD card/MMC insertion not detected by the SDnCD pin 1: SD card/MMC insertion detected by the SDnCD pin	R/(W)*2
5	SDCDMON	SDnCD Pin Monitor Flag 0: SDnCD pin level is high*3 1: SDnCD pin level is low*3	R
6	—	This bit is read as 0. The write value should be 0.	R/W
7	SDWPMON	SDnWP Pin Monitor Flag 0: SDnWP pin level is high 1: SDnWP pin level is low	R
8	SDD3RM	SDnDAT3 Removal Flag 0: SD card/MMC removal not detected by the SDnDAT3 pin 1: SD card/MMC removal detected by the SDnDAT3 pin	R/(W)*2
9	SDD3IN	SDnDAT3 Insertion Flag 0: SD card/MMC insertion not detected by the SDnDAT3 pin 1: SD card/MMC insertion detected by the SDnDAT3 pin	R/(W)*2
10	SDD3MON	SDnDAT3 Pin Monitor Flag 0: SDnDAT3 pin level is low 1: SDnDAT3 pin level is high	R

Bit	Symbol	Function	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The value is initialized by a reset and also on reset triggered by the SOFT_RST.SDRST flag.
Note 2. The flag does not change even if set to 1. Writing 0 changes the flag value to 0.
Note 3. The flag changes when the pin level continues for the period set in the SD_OPTION.CTOP[3:0] bits or longer.

The SD_INFO1 register indicates the detection of a response end or access end for a command sequence. The SD_INFO1 register also indicates the detection SD card/MMC insertion/removal and the write protection status.

During a multiblock transfer sequence, if CMD12 or CMD52 (SDIO abort) is issued, the ACEND flag sets to 1, but the RSPEND flag remains set to 0.

If the command sequence is stopped because of a communication error or timeout, the ACEND flag or RSPEND flag sets to 1.

After a reset is canceled, the SDD3MON bit, SDD3IN flag, and SDD3RM flag values are changed in accordance with the status of the SDnDAT3 (n = 0, 1) pin, and their values are changed when data is being transferred in wide bus mode. These 3 bits are used only for SD card. Set flags to be cleared to 0. Set flags that are not being cleared to 1.

RSPEND flag (Response End Detection Flag)

The RSPEND flag indicates that a response end was detected.

[Setting conditions]

- When reception of the response is completed
- When transmission of a command without response is completed
- When reception of the busy state after R1b response is completed
- When reception of the response to CMD52 that was issued by setting the C52PUB bit to 1 is completed for transfer of multiple block read
- When reception of the response to CMD52 that was issued by setting the C52PUB bit to 1 is completed for transfer of multiple block write
- This bit is set when a command sequence is halted because of a communications error or timeout.

[Clearing conditions]

- When 0 is written to RSPEND
- When a command without data is issued.

Note: When a command is issued in absence of data transfer, the RSPEND flag becomes 1 after the command sequence ends.

ACEND flag (Access End Detection Flag)

The ACEND flag indicates that an access end was detected.

[Setting conditions]

- When read access to the buffer is completed for transfer of single block read
- When read access to the buffer for the last block of data is completed for transfer of multiple block read
- When read access to the buffer and reception of the response to CMD12 are completed for transfer of multiple block read with automatic issuing of CMD12
- When reception of the busy state after reception of the CRC status is completed for transfer of single block write
- When reception of the busy state after reception of the CRC status of the last block of data is completed for transfer of multiple block write
- When reception of the response busy state for CMD12 is completed for transfer of multiple block write with automatic issuing of CMD12
- When reception of the response to CMD12 that was issued by setting the STP bit to1 is completed for transfer of multiple block read

- When reception of the response busy state for CMD12 that was issued by setting the STP bit to1 is completed for transfer of multiple block write
- When reception of the response to CMD52 that was issued by setting the IOABT bit to 1 is completed for transfer of multiple block read
- When reception of the response to CMD52 that was issued by setting the IOABT bit to1 is completed for transfer of multiple block write
- This bit is set when a command sequence is halted because of a communications error or timeout.

[Clearing conditions]

- When 0 is written to ACEND
- When the access end bit is set to 1.

Note: The ACEND flag becomes 1 after the command sequence ends.

SDCDRM flag (SDnCD Removal Flag)

The SDCDRM flag indicates that SDnCD was removed.

[Setting condition]

- After a change in SDnCD from 0 to 1, Mcycle elapsed with SDnCD held at 1.

[Clearing conditions]

- When 0 is written to SDCDRM.

Note: Mcycle is set in bits [3:0] in SD_OPTION.

SDCDIN flag (SDnCD Insertion Flag)

The SDCDIN flag indicates that SDnCD was inserted.

[Setting condition]

- After a change in SDnCD from 1 to 0, Mcycle elapsed with SDnCD held at 0.

[Clearing conditions]

- When 0 is written to SDCDIN.

Note: Mcycle is set in bits [3:0] in SD_OPTION.

SDD3RM flag (SDnDAT3 Removal Flag)

The SDD3RM flag indicates that SDnDAT3 was removed.

[Setting condition]

- After a change in SDnDAT3 from 1 to 0, two cycles of PCLKB elapsed with SDnDAT3 held at 0.

[Clearing condition]

- When 0 is written to SDD3RM.

SDD3IN flag (SDnDAT3 Insertion Flag)

The SDD3IN flag indicates that SDnDAT3 was inserted.

[Setting condition]

- After a change in SDnDAT3 from 0 to 1, two cycles of PCLKB elapsed with SDnDAT3 held at 1.

[Clearing condition]

- When 0 is written to SDD3IN.

47.2.11 SD_INFO2 : SD Card Interrupt Flag Register 2

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x03C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ILA	CBSY	SD_CLK_CTRLEN	—	—	—	BWE	BRE	SDD0MON	RSPTO	ILR	ILW	DTO	ENDE	CRCE	CMDE
Value after reset:	0*1	0*1	1*1	0	x	0	0*1	0*1	x	0*1	0*1	0*1	0*1	0*1	0*1	0*1

Bit	Symbol	Function	R/W
0	CMDE	Command Error Detection Flag 0: Command error not detected 1: Command error detected	R/W*2
1	CRCE	CRC Error Detection Flag 0: CRC error not detected 1: CRC error detected	R/W*2
2	ENDE	End Bit Error Detection Flag 0: End bit error not detected 1: End bit error detected	R/W*2
3	DTO	Data Timeout Detection Flag 0: Data timeout not detected 1: Data timeout detected	R/W*2
4	ILW	SD_BUF0 Illegal Write Access Detection Flag 0: Illegal write access to the SD_BUF0 register not detected 1: Illegal write access to the SD_BUF0 register detected	R/W*2
5	ILR	SD_BUF0 Illegal Read Access Detection Flag 0: Illegal read access to the SD_BUF0 register not detected 1: Illegal read access to the SD_BUF0 register detected	R/W*2
6	RSPTO	Response Timeout Detection Flag 0: Response timeout not detected 1: Response timeout detected	R/W*2
7	SDD0MON	SDnDAT0 Pin Status Flag 0: SDnDAT0 pin is low 1: SDnDAT0 pin is high	R
8	BRE	SD_BUF0 Read Enable Flag 0: Disable read access to the SD_BUF0 register 1: Enable read access to the SD_BUF0 register	R/W*2
9	BWE	SD_BUF0 Write Enable Flag 0: Disable write access to the SD_BUF0 register 1: Enable write access to the SD_BUF0 register	R/W*2
10	—	This bit is read as 0. The write value should be 0.	R/W
11	—	The read value is undefined. The write value should be 1.	R/W
12	—	This bit is read as 0. The write value should be 0.	R/W
13	SD_CLK_CTRLEN	SD_CLK_CTRL Write Enable Flag 0: SD/MMC bus (CMD and DAT lines) is busy, so write access to the SD_CLK_CTRL.CLKEN and CLKSEL[7:0] bits is disabled 1: SD/MMC bus (CMD and DAT lines) is not busy, so write access to the SD_CLK_CTRL.CLKEN and CLKSEL[7:0] bits is enabled	R

Bit	Symbol	Function	R/W
14	CBSY	Command Sequence Status Flag 0: Command sequence complete 1: Command sequence in progress (busy)	R
15	ILA	Illegal Access Error Detection Flag 0: Illegal access error not detected 1: Illegal access error detected	R/W*2
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The value is initialized by a reset and also on reset triggered by the SOFT_RST.SDRST flag.
Note 2. The flag does not change even if set to 1. Writing 0 changes the flag value to 0.

The SD_INFO2 register indicates the status of the SD buffer and the status of the SD card/MMC. Set flags to be cleared to 0. Set flags that are not being cleared to 1.

CMDE flag (Command Error Detection Flag)

The CMDE flag indicates that a command error was detected. The command sequence stops when a command error occurs. When the SDIO_MODE.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence is not completed. Perform the error processing shown in section 47.3.12. IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Read) or section 47.3.13. IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Write), and complete the command sequence.

[Setting conditions]

- The command index of the transmitted command differs from the command index of the received response.
- The command index of a command issued within a command sequence differs from the command index of the received response.

[Clearing condition]

- When 0 is written to CMDE.

CRCE flag (CRC Error Detection Flag)

The CRCE flag indicates that a CRC error was detected. The command sequence stops when a CRC error occurs. When the SDIO_MODE.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence is not completed. Perform the error processing shown in section 47.3.12. IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Read) or section 47.3.13. IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Write), and complete the command sequence.

[Setting conditions]

- When an error occurs in the CRC status.
- When a CRC error occurs in the read data.
- When a CRC error occurs in the response.
- A CRC error in the response to a command issued within a command sequence.

[Clearing condition]

- When 0 is written to CRCE.

ENDE flag (End Bit Error Detection Flag)

The ENDE flag indicates that an end bit error was detected. The command sequence is stopped when an end bit error occurs. When the SDIO_MODE.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence is not completed. Perform the error processing shown in section 47.3.12. IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Read) or section 47.3.13. IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Write), and complete the command sequence.

[Setting conditions]

- When an error occurs in the response length (and the end bit is not detected).
- When an error occurs in the read data length (and the end bit is not detected among the valid bits).

- When an error occurs in the CRC status length (and the end bit is not detected).
- An error in the length of a response to a command issued within a command sequence, for example when the end bit is not detected.

[Clearing condition]

- When 0 is written to ENDE.

DTO flag (Data Timeout Detection Flag)

The DTO flag indicates that a data timeout was detected. The command sequence stops when a data timeout occurs.

[Setting conditions]

- After R1b response, the busy state (SDnDAT0 = 0) continues for longer than Ncycle.
- After CRC status, the busy state (SDnDAT0 = 0) continues for longer than Ncycle.
- After write data, the CRC status is not received though Ncycle has elapsed.
- After read command, read data is not received though a time longer than Ncycle has elapsed.
- After CMD12 is issued within a command sequence, the busy state (SDnDAT0 = 0) for longer than Ncycle continues.
- After the reception of read data, read data for the next block are not received though a time longer than Ncycle has elapsed.
- After release of the read wait state, read data for the next block are not received though a time longer than Ncycle has elapsed.

Note: Ncycle is set in bits [7:4] in SD_OPTION.

[Clearing condition]

- When 0 is written to DTO.

ILW flag (SD_BUF0 Illegal Write Access Detection Flag)

The ILW flag indicates that an SD_BUF0 illegal write access was detected.

[Setting conditions]

- When data is written to SD_BUF0 while it is not in the data read/write command state.
- When data is written to SD_BUF0 while SD_BUF is full.
- When data is written to SD_BUF0 while an error occurs in the CRC status or CRC status length.
- When data is written to SD_BUF0 while a busy state after the CRC status continues for longer than Ncycle.

Note: Ncycle is set in bits [7:4] in SD_OPTION.

[Clearing condition]

- When 0 is written to ILW.

ILR flag (SD_BUF0 Illegal Read Access Detection Flag)

The ILR flag indicates that an SD_BUF0 illegal read access was detected.

[Setting conditions]

- When SD_BUF is empty while SD_BUF0 is read.
- When data with a CRC error or END error is read from SD_BUF0.

[Clearing condition]

- When 0 is written to ILR.

RSPTO flag (Response Timeout Detection Flag)

The RSPTO flag indicates that a response timeout was detected. The command sequence is stopped when a response timeout occurs. When the SDIOMD.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence is not completed. Perform the error processing shown in section 47.3.12. IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Read) or section 47.3.13. IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Write), and complete the command sequence.

[Setting condition]

- When a response is not received though a time longer than 640 cycles of SD/MMC clock has elapsed (including a response to a command issued within a command sequence).

[Clearing condition]

- When 0 is written to RSPTO.

SDD0MON flag (SDnDAT0 Pin Status Flag)

The SDD0MON flag indicates the status of the SDnDAT0 pin. If the data timeout (DTO) is set but the response timeout (RSPTO) is not set after the Erase command is issued, the end of the Erase sequence (SDD0MON = 1) is confirmed by polling DAT0.

If a communication error or timeout occurs during a write sequence, the DAT0 bit might retain the value 0.

While the SD/MMC clock is stopped, the DAT0 bit retains the value before the clock is stopped.

BRE flag (SD_BUF0 Read Enable Flag)

The BRE flag indicates that SD_BUF0 is enabled for reading.

[Setting conditions]

- When data set in SD_SIZE is stored in SD_BUF0 at single block transfer.
- When data set in SD_SIZE is stored in either bank 1 or bank 2 of SD_BUF0 at multiple block transfer.

[Clearing conditions]

- When 0 is written to BRE
- Reading of a block of data from SD_BUF0 by DMA transfer

When data is read from SD_BUF0 by the CPU, clear BRE then read the amount of data specified in SD_SIZE.

Even if a CRC error or an END error occurs while block data is read, data is stored in SD_BUF0 and BRE is set.

BWE flag (SD_BUF0 Write Enable Flag)

The BWE flag indicates that SD_BUF0 is enabled for writing.

[Setting conditions]

- When SD_BUF0 is empty at single block transfer.
- When either bank 1 or bank 2 of SD_BUF0 is empty at multiple block transfer.

[Clearing conditions]

- When 0 is written to BWE.
- Writing of a block of data to SD_BUF0 by DMA transfer.

When data is written to SD_BUF0 by the CPU, clear BWE and then write the amount of data specified in SD_SIZE.

SD_CLK_CTRLEN flag (SD_CLK_CTRL Write Enable Flag)

When a command sequence is started by writing to SD_CMD, the CBSY bit is set to 1 and, at the same time, the SD_CLK_CTRLEN bit is set to 0. The SD_CLK_CTRLEN bit is set to 1 after 8 cycles of SDCLK have elapsed after the CBSY bit clears to 0 on completion of the command sequence.

ILA flag (Illegal Access Error Detection Flag)

The ILA flag indicates that an illegal access error was detected.

[Setting conditions]

- Writing of data to SD_CMD within a command sequence (CBSY = 1).

- When SD_CMD[11] = 1 (command with data transfer) and SD_CMD[7:0] = 0000 1100b (CMD12) are set in SD_CMD.

[Clearing condition]

- When 0 is written to ILA.

47.2.12 SD_INFO1_MASK : SD INFO1 Interrupt Mask Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x040

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SDD3INM	SDD3RMM	—	—	—	SDCDINM	SDCDRMM	ACENDM	—	RSPENDM
Value after reset:	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	1

Bit	Symbol	Function	R/W
0	RSPENDM	Response End Interrupt Request Mask 0: Do not mask response end interrupt request 1: Mask response end interrupt request	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	ACENDM	Access End Interrupt Request Mask 0: Do not mask access end interrupt request 1: Mask access end interrupt request	R/W
3	SDCDRMM	SDnCD Removal Interrupt Request Mask 0: Do not mask SD card/MMC removal interrupt request by the SDnCD pin 1: Mask SD card/MMC removal interrupt request by the SDnCD pin	R/W
4	SDCDINM	SDnCD Insertion Interrupt Request Mask 0: Do not mask SD card/MMC insertion interrupt request by the SDnCD pin 1: Mask SD card/MMC insertion interrupt request by the SDnCD pin	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	SDD3RMM	SDnDAT3 Removal Interrupt Request Mask 0: Do not mask SD card/MMC removal interrupt request by the SDnDAT3 pin 1: Mask SD card/MMC removal interrupt request by the SDnDAT3 pin	R/W
9	SDD3INM	SDnDAT3 Insertion Interrupt Request Mask 0: Do not mask SD card/MMC insertion interrupt request by the SDnDAT3 pin 1: Mask SD card/MMC insertion interrupt request by the SDnDAT3 pin	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The SD_INFO1_MASK register enables or disables interrupt requests from the status flags in the SD_INFO1 register. See Table 47.5, for details on the relationship between the status flags and the requested interrupt source.

47.2.13 SD_INFO2_MASK : SD INFO2 Interrupt Mask Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x044

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ILAM	—	—	—	—	—	BWEM	BREM	—	RSPTOM	ILRM	ILWM	DTOM	ENDEM	CRCEM	CMDEM
Value after reset:	1	0	0	0	1	0	1	1	0	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	CMDEM	Command Error Interrupt Request Mask 0: Do not mask command error interrupt request 1: Mask command error interrupt request	R/W
1	CRCEM	CRC Error Interrupt Request Mask 0: Do not mask CRC error interrupt request 1: Mask CRC error interrupt request	R/W
2	ENDEM	End Bit Error Interrupt Request Mask 0: Do not mask end bit detection error interrupt request 1: Mask end bit detection error interrupt request	R/W
3	DTOM	Data Timeout Interrupt Request Mask 0: Do not mask data timeout interrupt request 1: Mask data timeout interrupt request	R/W
4	ILWM	SD_BUF0 Register Illegal Write Interrupt Request Mask 0: Do not mask illegal write detection interrupt request for the SD_BUF0 register 1: Mask illegal write detection interrupt request for the SD_BUF0 register	R/W
5	ILRM	SD_BUF0 Register Illegal Read Interrupt Request Mask 0: Do not mask illegal read detection interrupt request for the SD_BUF0 register 1: Mask illegal read detection interrupt request for the SD_BUF0 register	R/W
6	RSPTOM	Response Timeout Interrupt Request Mask 0: Do not mask response timeout interrupt request 1: Mask response timeout interrupt request	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	BREM*1	BRE Interrupt Request Mask 0: Do not mask read enable interrupt request for the SD buffer 1: Mask read enable interrupt request for the SD buffer	R/W
9	BWEM*1	BWE Interrupt Request Mask 0: Do not mask write enable interrupt request for the SD_BUF0 register 1: Mask write enable interrupt request for the SD_BUF0 register	R/W
10	—	This bit is read as 0. The write value should be 0.	R/W
11	—	This bit is read as 1.The write value should be 1.	R/W
14:12	—	These bits are read as 0. The write value should be 0.	R/W
15	ILAM	Illegal Access Error Interrupt Request Mask 0: Do not mask illegal access error interrupt request 1: Mask illegal access error interrupt request	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. When the SD_INFO2_MASK.BWEM bit is 0 or the SD_INFO2_MASK.BREM bit is 0, set the SD_DMAEN.DMAEN bit to 0. When the SD_DMAEN.DMAEN bit is 1, set the SD_INFO2_MASK.BWEM bit to 1 and the SD_INFO2_MASK.BREM bit to 1.

The SD_INFO2_MASK register enables or disables interrupt requests from the status flags in the SD_INFO2 register. See Table 47.5 for details on the relationship between the status flags and the requested interrupt source.

47.2.14 SD_CLK_CTRL : SD Clock Control Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x048

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CLKCTRLEN	CLKEN	CLKSEL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0[*1]	0	0	1	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CLKSEL[7:0]	SDHI Clock Frequency Select[*2] 0xFF PCLKB 0x00: PCLKB/2 0x01: PCLKB/4 0x02: PCLKB/8 0x04: PCLKB/16 0x08: PCLKB/32 0x10: PCLKB/64 0x20: PCLKB/128 0x40: PCLKB/256 0x80: PCLKB/512 Others: Setting prohibited	R/W
8	CLKEN	SD/MMC Clock Output Control[*2] 0: Disable SD/MMC clock output (fix SDnCLK signal low) 1: Enable SD/MMC clock output	R/W
9	CLKCTRLEN	SD/MMC Clock Output Automatic Control Select 0: Disable automatic control of SD/MMC clock output 1: Enable automatic control of SD/MMC clock output	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The value is initialized by a reset and also on reset triggered by the SOFT_RST.SDRST flag.
Note 2. Bits CLKSEL[7:0] and CLKEN cannot be write accessed when the SD_INFO2.SD_CLK_CTRLEN flag is 0.

The SD_CLK_CTRL register controls the SD/MMC clock frequency settings and output. Set the CLKEN bit to 1 before writing to the SD_CMD register to start a command sequence. Do not write to the SD_CLK_CTRL register when the SD_INFO2.SD_CLK_CTRLEN flag is 0.

CLKCTRLEN bit (SD/MMC Clock Output Automatic Control Select)

The CLKCTRLEN bit enables or disables the automatic control function for SD/MMC clock output, which causes the SD/MMC clock to output only within a command sequence.

The timing with which SD/MMC clock output starts and stops is as follows:

- SD/MMC clock output starts after writing to SD_CMD
- SD/MMC clock output stops when 8 cycles of SD/MMC clock have elapsed after the end of the command sequence.

In addition, SD/MMC clock is fixed to 0 while SCLKEN of SD_CLK_CTRL is 0, regardless of the value of this bit.

47.2.15 SD_SIZE : Transfer Data Length Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x04C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	LEN[9:0]									
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	LEN[9:0]	Transfer Data Size Setting These bits specify the transfer data size.*1	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Do not rewrite these bits when the SD_INFO2.CBSY flag is 1.

The SD_SIZE register sets the transfer data size.

LEN[9:0] bits (Transfer Data Size Setting)

When using single block transfer, the transfer data size can be set in the LEN[9:0] bits from 1 byte to 512 bytes. When CMD12 is automatically issued during a multiblock transfer sequence (CMD18 and CMD25), the transfer data size can only be set to 512 bytes. When CMD12 is not automatically issued during a multiblock transfer sequence, the transfer data size can be set to 32, 64, 128, 256, or 512 bytes. However, a 32, 64, 128, or 256 bytes multiblock read transfer can only be performed during an SDIO multiblock transfer (CMD53). Do not set these bits to 0 when using a command that includes data transfer.

47.2.16 SD_OPTION : SD Card Access Control Option Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x050

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WIDTH	—	WIDTH8	—	—	—	—	TOUTMASK	TOP[3:0]				CTOP[3:0]			
Value after reset:	0*1	1	0*1	0	0	0	0	0*1	1*1	1*1	1*1	0*1	1*1	1*1	1*1	0*1

Bit	Symbol	Function	R/W
3:0	CTOP[3:0]	Card Detection Time Counter*2 0x0: PCLKB × 2^{10} 0x1: PCLKB × 2^{11} 0x2: PCLKB × 2^{12} 0x3: PCLKB × 2^{13} 0x4: PCLKB × 2^{14} 0x5: PCLKB × 2^{15} 0x6: PCLKB × 2^{16} 0x7: PCLKB × 2^{17} 0x8: PCLKB × 2^{18} 0x9: PCLKB × 2^{19} 0xA: PCLKB × 2^{20} 0xB: PCLKB × 2^{21} 0xC: PCLKB × 2^{22} 0xD: PCLKB × 2^{23} 0xE: PCLKB × 2^{24} 0xF: Setting prohibited	R/W
7:4	TOP[3:0]	Timeout Counter*2 0x0: SDHI clock × 2^{13} 0x1: SDHI clock × 2^{14} 0x2: SDHI clock × 2^{15} 0x3: SDHI clock × 2^{16} 0x4: SDHI clock × 2^{17} 0x5: SDHI clock × 2^{18} 0x6: SDHI clock × 2^{19} 0x7: SDHI clock × 2^{20} 0x8: SDHI clock × 2^{21} 0x9: SDHI clock × 2^{22} 0xA: SDHI clock × 2^{23} 0xB: SDHI clock × 2^{24} 0xC: SDHI clock × 2^{25} 0xD: SDHI clock × 2^{26} 0xE: SDHI clock × 2^{27} 0xF: Setting prohibited	R/W
8	TOUTMASK	Timeout Mask 0: Activate timeout 1: Inactivate timeout (do not set RSPTO and DTO bits of SD_INFO2 or CRCBSYTO, CRCTO, RDTO, BSYTO1, BSYTO0, RSPTO1 and RSPTO0 bits of SD_ERR_STS2) When timeout occurs because of an inactivated timeout, execute a software reset to terminate the command sequence.	R/W
12:9	—	These bits are read as 0. The write value should be 0.	R/W
13	WIDTH8	Bus Width*2 See bit 15 WIDTH bit.	R/W
14	—	This bit is read as 1. The write value should be 1.	R/W
15	WIDTH	Bus Width*2 <table><tr><th>WIDTH</th><th>WIDTH8</th><th>Bus Width</th></tr><tr><td>0</td><td>1</td><td>8-bit width</td></tr><tr><td>0</td><td>0</td><td>4-bit width</td></tr><tr><td>1</td><td>0</td><td>1-bit width</td></tr><tr><td>1</td><td>1</td><td>1-bit width</td></tr></table>For 1-byte write transfers, set 4-bit or 1-bit width. Do not set 8-bit width.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The value is initialized by a reset and also on reset triggered by the SOFT_RST.SDRST flag.

Note 2. Do not rewrite these bits when the SD_INFO2.CBSY flag is 1.

The SD bus width and timeout counter are set in the SD_OPTION register.

47.2.17 SD_ERR_STS1 : SD Error Status Register 1

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x058

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	CRCTK[2:0]			CRCTKE	RDCRCE	RSPCRCE1	RSPCRCE0	—	—	CRCLENE	RDLENE	RSPLENE1	RSPLENE0	CMDE1	CMDE0
Value after reset:	0	0*1	1*1	0*1	0*1	0*1	0*1	0*1	0	0	0*1	0*1	0*1	0*1	0*1	0*1

Bit	Symbol	Function	R/W
0	CMDE0	Command Error Flag 0 0: No error exists in command index field value of a command*2 response 1: Error exists in command index field value of a command*2 response	R
1	CMDE1	Command Error Flag 1 0: No error exists in command index field value of a command*3 response 1: Error exists in command index field value of a command*3 response (with SD_CMD.CMDIDX[5:0] setting, an error that occurs with CMD12 issue is indicated in the CMDE0 flag)	R
2	RSPLENE0	Response Length Error Flag 0 0: No error exists in command*2 response length 1: Error exists in command*2 response length	R
3	RSPLENE1	Response Length Error Flag 1 0: No error exists in command*3 response length 1: Error exists in command*3 response length (with SD_CMD.CMDIDX[5:0] setting, an error that occurs with CMD12 issue is indicated in the RSPLENE0 flag)	R
4	RDLENE	Read Data Length Error Flag 0: No read data length error occurred 1: Read data length error occurred	R
5	CRCLENE	CRC Status Token Length Error Flag 0: No CRC status token length error occurred 1: CRC status token length error occurred	R
7:6	—	These bits are read as 0.	R
8	RSPCRCE0	Response CRC Error Flag 0 0: No CRC error detected in command*2 response 1: CRC error detected in command*2 response	R
9	RSPCRCE1	Response CRC Error Flag 1 0: No CRC error detected in command*3 response (with SD_CMD.CMDIDX[5:0] setting, an error that occurs with CMD12 issue is indicated in the RSPCRCE0 flag) 1: CRC error detected in command*3 response	R
10	RDCRCE	Read Data CRC Error Flag 0: No CRC error detected in read data 1: CRC error detected in read data	R
11	CRCTKE	CRC Status Token Error Flag 0: No error detected in CRC status token 1: Error detected in CRC status token	R

Bit	Symbol	Function	R/W
14:12	CRCTK[2:0]	CRC Status Token These bits store the CRC status token value (normal value is 010b).	R
31:15	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

Note 1. The value is initialized by a reset and also on reset triggered by the SOFT_RST.SDRST flag.

Note 2. CMD other than CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

Note 3. CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

The SD_ERR_STS1 register indicates the CRC status token, CRC error, end bit error, and command error.

47.2.18 SD_ERR_STS2 : SD Error Status Register 2

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x05C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	CRCBSYTO	CRCTO	RDTO	BSYTO1	BSYTO0	RSPTO1	RSPTO0
Value after reset:	0	0	0	0	0	0	0	0	0	0*1	0*1	0*1	0*1	0*1	0*1	0*1

Bit	Symbol	Function	R/W
0	RSPTO0	Response Timeout Flag 0 0: After command*2 was issued, response was received in less than 640 cycles of the SD/MMC clock 1: After command*2 was issued, response was not received in 640 or more cycles of the SD/MMC clock	R
1	RSPTO1	Response Timeout Flag 1 0: After command*3 was issued, response was received in less than 640 cycles of the SD/MMC clock 1: After command*3 was issued, response was not received after 640 or more cycles of the SD/MMC clock (with SD_CMD.CMDIDX[5:0] setting, an error that occurs with CMD12 issue is indicated in the RSPTO0 flag)	R
2	BSYTO0	Busy Timeout Flag 0 0: After R1b response was received, SD/MMC was released from the busy state during the specified period*4 1: After R1b response was received, SD/MMC was in the busy state after the specified period*4 elapsed	R
3	BSYTO1	Busy Timeout Flag 1 0: After CMD12 was automatically issued, SD/MMC was released from the busy state during the specified period*4 1: After CMD12 was automatically issued, SD/MMC was in the busy state after the specified period*4 elapsed (with SD_CMD.CMDIDX[5:0] setting, an error that occurs with CMD12 issue is indicated in the BSYTO0 flag)	R
4	RDTO	Read Data Timeout Flag When a read command is issued, this flag sets to 1 when read data is not received after the specified period*4 elapses. When read data is received, this flag sets to 1 when the next block of read data is not received after the specified period*4 elapses. When the SD/MMC exits the read wait state, this flag sets to 1 when the next block of read data is not received after the specified period*4 elapses.	R

Bit	Symbol	Function	R/W
5	CRCTO	CRC Status Token Timeout Flag 0: After CRC data was written to the SD card/MMC, a CRC status token was received during the specified period[*4] 1: After CRC data was written to the SD card/MMC, a CRC status token was not received after the specified period[*4] elapsed	R
6	CRCBSYTO	CRC Status Token Busy Timeout Flag 0: After a CRC status token was received, the SD/MMC was released from the busy state during the specified period[*4] 1: After a CRC status token was received, the SD/MMC was in the busy state after the specified period[*4] elapsed	R
31:7	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3
Note 1. The value is initialized by a reset and also on reset triggered by the SOFT_RST.SDRST flag.
Note 2. CMD other than CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.
Note 3. CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.
Note 4. Set the SD_OPTION.TOP[3:0] bits to select the number of *n* cycles.

The SD_ERR_STS2 register indicates the timeout status.

47.2.19 SD_BUF0 : SD Buffer Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x060

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:																
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
31:0	n/a	When writing to the SD card, the write data is written to this register. When reading from the SD card, the read data is read from this register.	R/W

This register is internally connected to two 512-byte buffers.

If both buffers are not empty when executing multiple block read, the SD card/MMC clock is stopped to suspend receiving data. When one of the buffers is empty, the SD card/MMC clock is supplied to resume receiving data.

47.2.20 SDIO_MODE : SDIO Mode Control Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x068

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	C52PUB	IOABT	—	—	—	—	—	RWREQ	—	INTEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	INTEN	SDIO Interrupt Acceptance Enable*1 0: Disable SDIO interrupt acceptance 1: Enable SDIO interrupt acceptance	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	RWREQ	Read Wait Request 0: Allow SD/MMC to exit read wait state 1: Request for SD/MMC to enter read wait state	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	IOABT	SDIO Abort If this bit is set to 1 during multiblock transfer triggered by CMD53, CMD52 is immediately issued, and the command sequence is aborted.	R/W
9	C52PUB	SDIO None Abort If this bit is set to 1 during multiblock transfer triggered by CMD53, CMD52 is issued after the transfer process is complete, and the command sequence is completed.	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Do not rewrite this bit when the SD_INFO2.CBSY flag is 1.

The SDIO_MODE register controls reception of the SDIO interrupt, CMD52 issuance during multiblock transfer, and read wait request. Do not set bits C52PUB and IOABT to 1 at the same time.

RWREQ bit (Read Wait Request)

When RWREQ is set to 1 in the CMD53 (multiple block) read sequence, the block transfer enters the read wait state between blocks.

[Read wait state releasing]

- The read wait state is released, when RWREQ is cleared to 0 in the read wait state.
- When IOABT is set to 1 in the read wait state, RWREQ is automatically cleared to 0 after CMD52 is issued, and then the read wait state is released.
- When C52PUB and RWREQ are set to 1 simultaneously in the CMD53 (multiple block) read sequence, the read wait state is not automatically released. Therefore, after the CMD52 response is received, clear RWREQ. You must set RWREQ and C52PUB simultaneously.

When RWREQ is set to 1 while the last block in the CMD53 (multiple block) read sequence is transferred, the read wait state is not entered and RWREQ is automatically cleared to 0 by setting access end. Set RWREQ to 1 after the response end flag sets.

IOABT bit (SDIO Abort)

When the IOABT bit is set to 1 in a CMD53 (multiple block) sequence, the CMD53 sequence is halted and CMD52 is issued. However, if a command sequence is halted because of a communication error or timeout, CMD52 is not issued.

Although continued buffer access is possible even after IOABT is set to 1, the buffer access error bit (ILR or ILW) in SD_INFO2 is set accordingly. Set SD_ARG before setting IOABT to 1.

When IOABT is set to 1 during transfer for a single block write, the access end flag sets when SD_BUF0 becomes empty, and CMD52 is not issued. If SD_BUF0 contains data, the access end flag sets on completion of reception of the busy state without CMD52 being issued.

When IOABT is set to 1 during transfer for single block read, the access end flag sets immediately after IOABT is set, and CMD52 is not issued.

When IOABT is set to 1 during reception of the busy state after an R1b response, the access end flag sets on completion of reception of the busy state without CMD52 being issued.

When IOABT is set to 1 after a command sequence is completed, CMD52 is not issued and the access end flag does not set.

Set IOABT to 1 after the response end flag sets.

Set IOABT to 0 after the access end flag sets.

C52PUB bit (SDIO None Abort)

When the C52PUB bit is set to 1 in the CMD53 (multiple block) write sequence, CMD52 is automatically issued between blocks if SD_BUF0 becomes empty. C52PUB is automatically cleared to 0 after reception of the response to CMD52 is completed. Additionally, if C52PUB is set to 1 while the last block is being transferred, CMD52 is not issued. In this case, C52PUB is automatically cleared to 0 after the access end flag sets to 1.

When C52PUB and RWREQ are set to 1 in the CMD53 (multiple block) read sequence, the block transfer enters the read wait state between blocks and CMD52 is automatically issued. C52PUB is automatically cleared to 0 after reception of the response to CMD52 is completed. Additionally, if C52PUB is set to 1 while the last block is being transferred, CMD52 is not issued. In this case, C52PUB is automatically cleared to 0 after the access end flag sets to 1.

If C52PUB is set to 1 in the CMD53 (multiple block) read sequence, you must set RWREQ to 1 in addition to C52PUB.

Set SD_ARG before setting C52PUB to 1.

Set C52PUB to 1 after the response end flag sets.

47.2.21 SDIO_INFO1 : SDIO Interrupt Flag Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x06C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EXWT	EXPU B52	—	—	—	—	—	—	—	—	—	—	—	—	—	IOIRQ
Value after reset:	0[*1]	0[*1]	0	0	0	0	0	0	0	0	0	0	0	x	x	0[*1]

Bit	Symbol	Function	R/W
0	IOIRQ	SDIO Interrupt Status Flag 0: No SDIO interrupt detected 1: SDIO interrupt detected	R/(W)[*2]
2:1	—	The read values are undefined. The write value should be 1.	R/W
13:3	—	These bits are read as 0. The write value should be 0.	R/W
14	EXPUB52	EXPUB52 Status Flag Indicates the status of the EXPUB52.	R/(W)[*2]
15	EXWT	EXWT Status Flag Indicates the status of the EXWT.	R/(W)[*2]

Bit	Symbol	Function	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. The value is initialized by a reset and also on reset triggered by the SOFT_RST.SDRST flag.
Note 2. Only 0 can be written to clear the bit.

The SDIO_INFO1 register indicates the status of the SDIO card access. Set flags to be cleared to 0. Set flags that are not being cleared to 1.

IOIRQ flag (SDIO Interrupt Status Flag)

The IOIRQ flag indicates that an SDIO interrupt occurred.

[Setting condition]

- When SDIO interrupt from an SDIO card is received while INTEN in SDIO_MODE is set to 1.

[Clearing condition]

- When 0 is written to IOIRQ.*1

Note 1. Before clearing this bit, access the SDIO card to negate the SDIO interrupt signal from the SDIO card. If the interrupt signal is not negated, this bit can be set again.

EXPUB52 flag (EXPUB52 Status Flag)

The EXPUB52 flag indicates the EXPUB52 status.

[Setting conditions]

- While the last block in the CMD53 (multiple block) sequence is transferred, C52PUB in SDIO_MODE is set to 1.
- While C52PUB is set to 1 in the CMD53 (multiple block) write sequence, the last block is transferred.

[Clearing condition]

- When 0 is written to EXPUB52.

EXWT flag (EXWT Status Flag)

The EXWT flag indicates the EXWT status.

[Setting condition]

- While the last block in the CMD53 (multiple block) read sequence is transferred, RWREQ in SDIO_MODE is set to 1.

[Clearing condition]

- When 0 is written to EXWT.

47.2.22 SDIO_INFO1_MASK : SDIO INFO1 Interrupt Mask Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x070

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EXWTM	EXPUB52M	—	—	—	—	—	—	—	—	—	—	—	—	—	IOIRQM
Value after reset:	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	Symbol	Function	R/W
0	IOIRQM	IOIRQ Interrupt Mask Control 0: Do not mask IOIRQ interrupts 1: Mask IOIRQ interrupts	R/W
2:1	—	These bits are read as 1.The write value should be 1.	R/W
13:3	—	These bits are read as 0. The write value should be 0.	R/W
14	EXPUB52M	EXPUB52 Interrupt Request Mask Control 0: Do not mask EXPUB52 interrupt requests 1: Mask EXPUB52 interrupt requests	R/W
15	EXWTM	EXWT Interrupt Request Mask Control 0: Do not mask EXWT interrupt requests 1: Mask EXWT interrupt requests	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The SDIO_INFO1_MASK register enables or disables interrupt requests from the status flags in the SDIO_INFO1 register. See Table 47.5 for details on the relationship between the status flags and the requested interrupt source.

47.2.23 SD_DMAEN : DMA Mode Enable Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x1B0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMAEN	—
Value after reset:	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	DMAEN	DMA Transfer Enable*1 *2 0: Disable use of DMA transfer to access SD_BUF0 register 1: Enable use of DMA transfer to access SD_BUF0 register	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	—	This bit is read as 1. The write value should be 1.	R/W
11:5	—	These bits are read as 0. The write value should be 0.	R/W
12	—	This bit is read as 1. The write value should be 1.	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Do not rewrite this bit when the SD_INFO2.CBSY bit is 1.
Note 2. When the SD_INFO2_MASK.BWEM bit is 0 or the SD_INFO2_MASK.BREM bit is 0, set the SD_DMAEN.DMAEN bit to 0. When the SD_DMAEN.DMAEN bit is 1, set the SD_INFO2_MASK.BWEM bit to 1 and the SD_INFO2_MASK.BREM bit to 1.

The SD_DMAEN register enables or disables DMA transfers.

DMAEN bit (DMA Transfer Enable)

When using DMA transfer to access the SD buffer, set the DMAEN bit to 1 before setting the SD_CMD register.

47.2.24 SOFT_RST : Software Reset Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x1C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SDRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	Symbol	Function	R/W
0	SDRST	Software Reset Control 0: Reset SD/MMC Host Interface software 1: Cancel reset of SD/MMC Host Interface software	R/W
2:1	—	These bits are read as 1. The write value should be 1.	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Table 47.4 lists the bits and flags initialized by SD/MMC Host Interface software reset.

Table 47.4 Bits and flags initialized by SD/MMC Host Interface software reset

Register	Bit/flag
SD_STOP	SEC, STP
SD_INFO1	RSPEND, ACEND
SD_INFO2	CMDE, CRCE, ENDE, DTO, ILW, ILR, RSPTO, BRE, BWE, SD_CLK_CTRLEN, CBSY, ILA
SD_CLK_CTRL	CLKEN
SD_OPTION	CTOP[3:0], TOP[3:0], TOUTMASK, WIDTH8, WIDTH
SD_ERR_STS1	CMDE0, CMDE1, RSPLENE0, RSPLENE1, RDLENE, CRCLENE, RSPCRCE0, RSPCRCE1, RDCRCE, CRCTKE, CRCTK[2:0]
SD_ERR_STS2	RSPTO0, RSPTO1, BSYTO0, BSYTO1, RDTO, CRCTO, CRCBSYTO
SDIO_INFO1	IOIRQ, EXPUB52, EXWT
SDIF_MODE	NOCHKCR

47.2.25 SDIF_MODE : SD Interface Mode Setting Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x1CC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	NOCHKCR	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0*1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	NOCHKCR	CRC Check Mask CRC check mask bit for MMC test commands. Set when CRC16 or CRC status value check is not executed. 0: Enable CRC check 1: Disable CRC Check (ignore CRC16 valued when reading and ignore CRC status value when writing)	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. The value is initialized by a reset and also on reset triggered by the SOFT_RST.SDRST flag.

NOCHKCR bit (CRC Check Mask)

The NOCHKCR bit is used for MMC test commands. This bit is set when CRC16 or CRC status value check is not executed.

47.2.26 EXT_SWAP : Swap Control Register

Base address: SDHIn = 0x4025_2000 + 0x0400 × n (n = 0, 1)
SDHIn_NS = 0x5025_2000 + 0x0400 × n (n = 0, 1)

Offset address: 0x1E0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	BRSWP	BWSWP	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	BWSWP	SD_BUF0 Swap Write*1 0: Normal write operation 1: Swap the byte endian order before writing to SD_BUF0 register	R/W
7	BRSWP	SD_BUF0 Swap Read*1 0: Normal read operation 1: Swap the byte endian order before reading SD_BUF0 register	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Do not rewrite this bit when the SD_INFO2.CBSY flag is 1.

The EXT_SWAP register selects whether or not the byte endian order is swapped when accessing the SD_BUF0 register. See section 47.3.1. SD/MMC Interface for details on the differences in accessing the SD_BUF0 register based on the EXT_SWAP register value.

47.3 Operation

47.3.1 SD/MMC Interface

When data is read from the SD card/MMC, the process is as follows:

1. The SD/MMC Host Interface receives data from the SD card/MMC through the SDnDAT signal (see Figure 47.2 and Figure 47.3).
2. The received data is stored in SD_BUF of the MMC Host Interface (see Figure 47.4).

3. The data stored in SD_BUF is read from SD_BUF0 (see Figure 47.5).

When data is written to the SD card/MMC, the specified procedure is reversed.

When accessing SD_BUF0, pay attention to the transfer order in SDnDAT and the store order in SD_BUF. If required, you can change the byte endian of the data read from or written to SD_BUF0 using the SDSWAP register. See Figure 47.6.

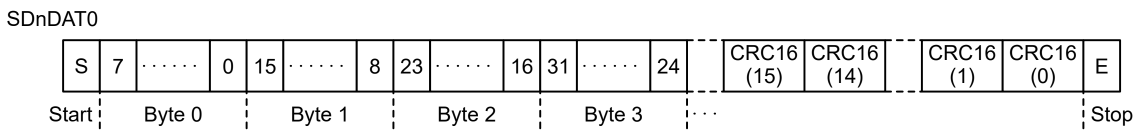


Figure 47.2 SDnDAT in 1-bit width mode

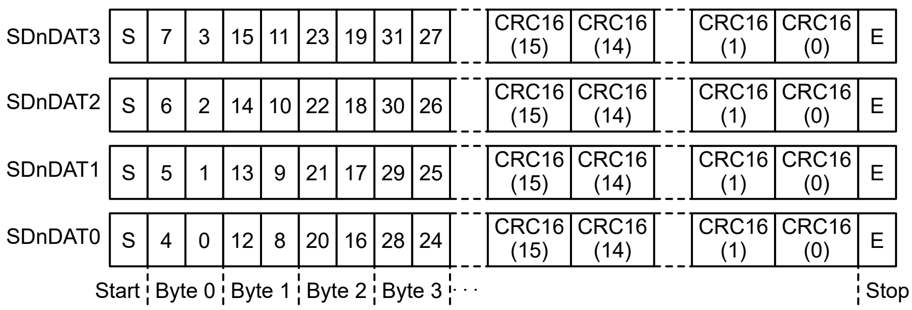


Figure 47.3 SDnDAT in 4-bit width mode

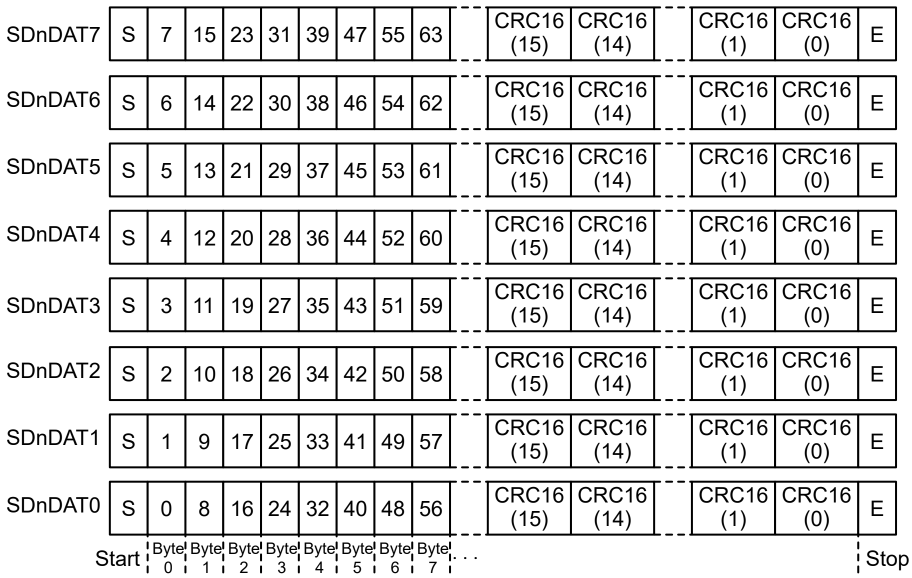


Figure 47.4 SDnDAT in 8-bit width mode

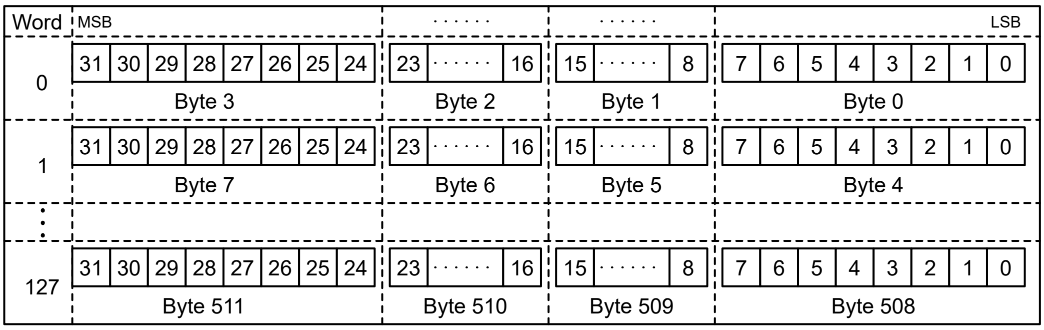


Figure 47.5 SD_BUF store data

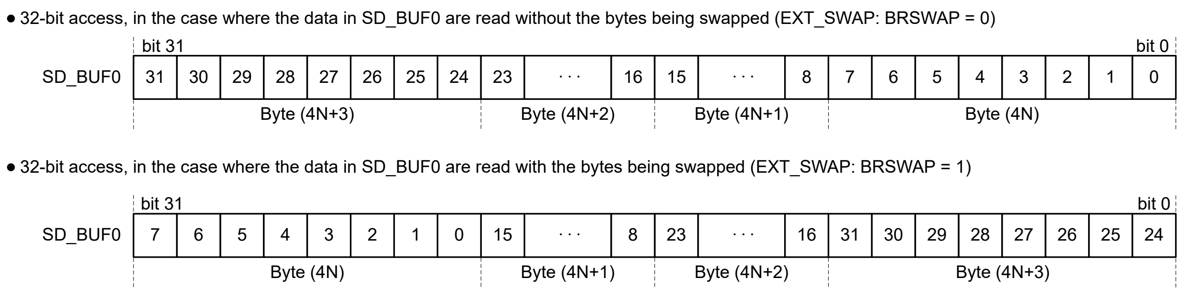


Figure 47.6 Read from SD_BUF0

47.3.2 Card Detect/Write Protect

47.3.2.1 Card Detect

The SD/MMC Host Interface has two types of card detect functions.

(1) Card detect with SDnCD (n = 0, 1)

Figure 47.7 shows the timing for card detect using SDnCD. SDnCD is connected to the card socket and pulled up on the host device. The resistance of the pull-up resistor is determined by the specification of the SD/MMC host device.

(2) Card insertion

SDnCD is pulled down when a card is inserted. At this point, if SDnCD is pulled down for the Mcycle period (set in SD_OPTION), SDCDIN in SD_INFO1 is set to 1. It is cleared by writing 0.

(3) Card removal

SDnCD is pulled up when a card is removed. At this point, if SDnCD is pulled up for the Mcycle period (set in SD_OPTION), SDCDRM in SD_INFO1 is set to 1. It is cleared by writing 0.

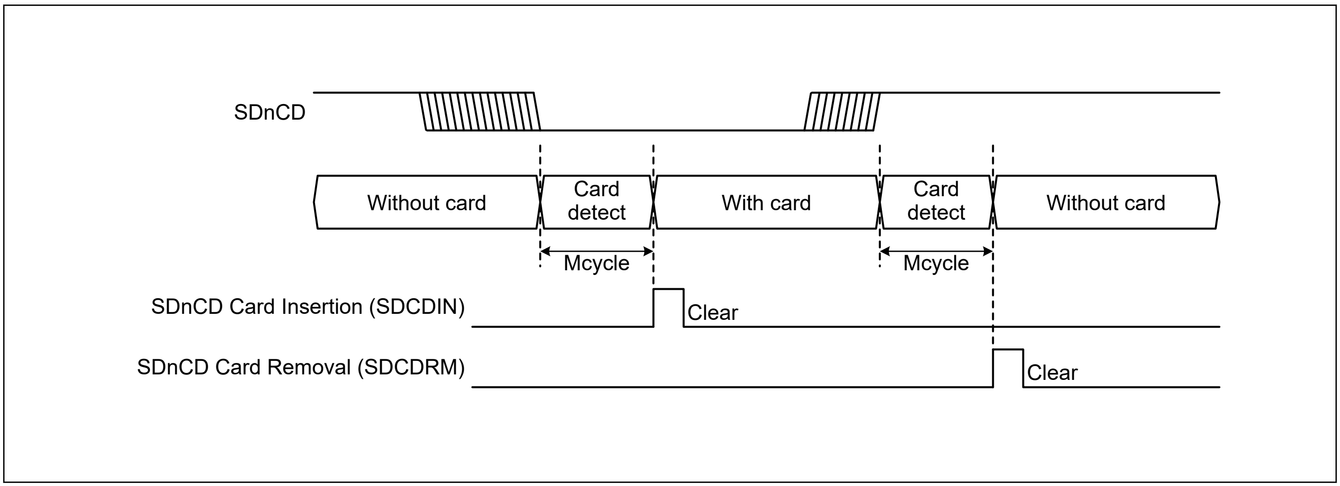


Figure 47.7 Example of card detect with SDnCD

(4) SD card detect with SDnDAT3 (n = 0, 1)

Figure 47.8 shows the timing when the SD card is detected with SDnDAT3. In addition, SDnDAT3 is pulled down by the host device, and the resistance value for pulling down is determined by the specification of the SD host device.

(5) Card insertion

When an SD card is inserted, SDnDAT3 is pulled up and SDD3IN in SD_INFO1 is set to 1. It is cleared by writing 0.

(6) Card removal

When an SD card is removed, SDnDAT3 is pulled down and SDD3RM in SD_INFO1 is set to 1. It is cleared by writing 0.

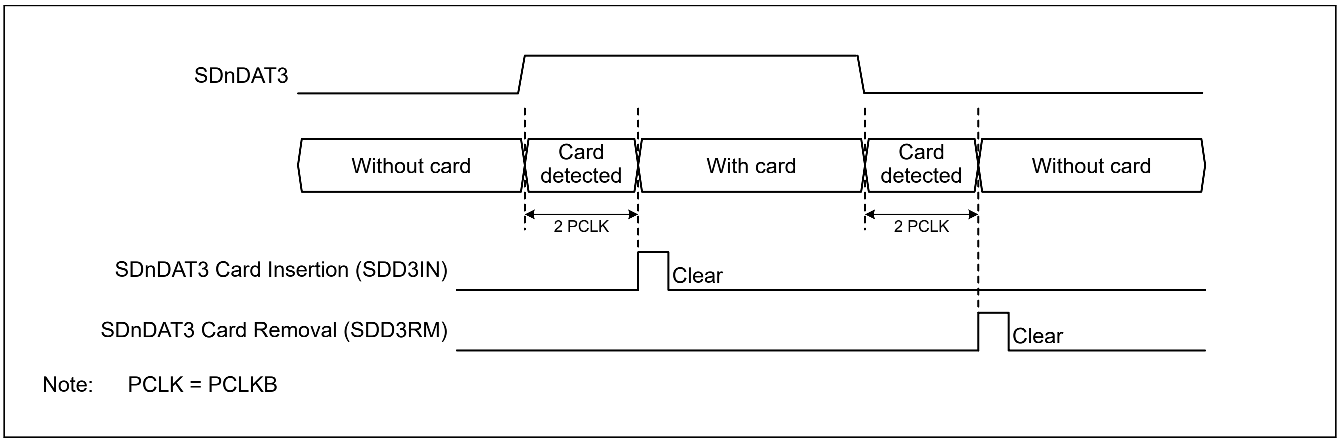


Figure 47.8 SD card detect with SDnDAT3

47.3.2.2 Write Protect

The SD/MMC Host Interface has two types of write protect functions.

(1) Write protect with SDnWP (n = 0, 1)

SDnWP is connected to the card socket and pulled up or pulled down by the card insertion. The selection of pulling up or pulling down and the resistance value is determined by the specification of the SD host device. When the SDnWP state is reflected to SDWPMON in SD_INFO1, the write protect state is set after the SD card is inserted.

(2) Write protect with command

The internal write protection of the card and the lock/unlock operation of the card are realized by the command.

47.3.3 Interrupt Request and DMA Transfer Request

47.3.3.1 Interrupts

Table 47.5 lists the SDHI interrupt sources. The SDHI requests an interrupt when:

- The status flags in registers SD_INFO1, SD_INFO2, and SDIO_INFO1 set to 1.
- The associated bits in the SD_INFO1_MASK, SD_INFO2_MASK, and SDIO_INFO1_MASK registers are 0.

When clearing the status flags in registers SD_INFO1, SD_INFO2, and SDIO_INFO1, write 0 to the status flags to be cleared and write 1 to the status flags that are not being cleared.

Table 47.5 Interrupt sources

Interrupt sources	Status flag register		Interrupt mask register		Interrupt name
	Register symbol	Bit symbol	Register symbol	Bit symbol	Ch n (n = 0, 1)
Card Access Interrupt (CACI)	SD_INFO1	ACEND	SD_INFO1_MASK	ACENDM	SDHI_MMCn_ACCS
		RSPEND		RSPENDM	
	SD_INFO2	ILA	SD_INFO2_MASK	ILAM	
		BWE		BWEM	
		BRE		BREM	
		RSPTO		RSPTOM	
		ILR		ILRM	
		ILW		ILWM	
		DTO		DTOM	
		ENDE		ENDEM	
		CRCE		CRCEM	
		CMDE		CMDEM	
SDIO Access Interrupt (SDACI)	SDIO_INFO1	EXWT	SDIO_INFO1_MASK	EXWTM	SDHI_MMCn_SDIO
		EXPUB52		EXPUB52M	
		IOIRQ		IOIRQM	
Card Detect Interrupt (CDETI)	SD_INFO1	SDD3IN	SD_INFO1_MASK	SDD3INM	SDHI_MMCn_CARD
		SDD3RM		SDD3RMM	
		SDCDIN		SDCDINM	
		SDCDRM		SDCDRMM	

47.3.3.2 DMA Transfer Requests (SDHI_MMCn_ODMSDBREQ, n = 0, 1)

The SD/MMC Host Interface has two types of DMA transfer requests.

(1) SD_BUF write DMA transfer request

- When the BWE bit in SD_INFO2 is set to 1 while the DMAEN bit in SD_DMAEN is set to 1, the SD_BUF write DMA transfer request is asserted.
- The SD_BUF write DMA transfer request is negated when the last data in one block (based on the transfer data size set in SD_SIZE) is transferred. The SD_BUF write DMA transfer request is also negated by clearing the SDRST bit in SOFT_RST to 0 or setting the STP bit in SD_STOP to 1. However, if a communications error or timeout occurs at the DMA transfer, the SD_BUF write DMA transfer request is not negated.
- The BWE bit in SD_INFO2 is cleared after transfer of the last data in one block following a request for writing to SD_BUF by DMA transfer.
- The number of DMA transfers must be n × one block. (n = integer, one block = the transfer data size set in SD_SIZE)
- When the IOABT bit in SDIO_MODE is set to 1, the SD_BUF write DMA transfer request is negated.
- The DMA transfer request is also negated by clearing the DMAEN bit to 0. However, the DMA transfer request is asserted again when the DMAEN bit is set to 1 before writing to SD_CMD.
- Because the BWE bit in SD_INFO2 is not cleared in response to setting the STP/IOABT bit, or to a communications error or timeout, clear the bit to 0 before issuing the next command. The next request to write to SD_BUF by DMA transfer is not issued while the BWE bit is set.

(2) SD_BUF read DMA transfer request

- When the BRE bit in SD_INFO2 is set to 1 while the DMAEN bit in the SD_DMAEN register is set to 1, the SD_BUF read DMA transfer request is asserted.

- The SD_BUF read DMA transfer request is negated when the last data in one block (based on the transfer data size set in SD_SIZE) is transferred. The SD_BUF read DMA transfer request is also negated by clearing the SDRST bit in SOFT_RST to 0 or setting the STP bit in SD_STOP to 1. However, if a communications error or timeout occurs at the DMA transfer, the SD_BUF read DMA transfer request is not negated.
- The BRE bit in SD_INFO2 is cleared after transfer of the last data in one block following a request to write to SD_BUF by DMA transfer.
- The number of DMA transfers must be n × one block. (n = integer, one block = the transfer data size set in SD_SIZE)
- When the IOABT bit in SDIO_MODE is set to 1, the SD_BUF read DMA transfer request is negated.
- The DMA transfer request is also negated by clearing the DMAEN bit to 0. However, the DMA transfer request is asserted again when the DMAEN bit is set to 1 before writing to SD_CMD.
- Because the BRE bit in SD_INFO2 is not cleared in response to setting the STP/IOABT bit or in response to a communications error or timeout, clear the bit to 0 before issuing the next command. The next request to write to SD_BUF by DMA transfer is not issued while the BRE bit is set.

47.3.4 Communication Errors and Timeouts

When a communication error or timeout error occurs, depending on the type of error, the associated status flag in the SD_INFO2 register sets to 1. Also, depending on the source of the error, the associated flag in the SD_ERR_STS1 or SD_ERR_STS2 register sets to 1.

The status flags in registers SD_ERR_STS1 and SD_ERR_STS2 clear to 0 by writing to the SD_CMD register, or by setting the SOFT_RST.SDRST bit to 0.

Table 47.6 Communication errors

Communication error	Interrupt flag register		Error status register		This occurs when...
	Register symbol	Bit symbol	Register symbol	Bit symbol	
End bit error	SD_INFO2	ENDE	SD_ERR_STS1	CRCLENE	The CRC status token length is in error
				RDLENE	The read data length is in error
				RSPLENE1	The response length is in error*1
				RSPLENE0	The response length is in error*2
CRC error		CRCE		CRCTKE	The CRC status token is in error
				RDCRCE	There is a CRC error in the read data
				RSPCRCE1	There is a CRC error in the response*1
				RSPCRCE0	There is a CRC error in the response*2
Command error		CMDE		CMDE1	The command index field value for the transmitted command and received response do not match*1
				CMDE0	The command index field value for the transmitted command and received response do not match*2

Note 1. CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.
Note 2. CMD other than CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

Table 47.7 Timeouts

Timeout	Interrupt flag register		Error status register		This occurs when...
	Register symbol	Bit symbol	Register symbol	Bit symbol	
Response timeout	SD_INFO2	RSPTO	SD_ERR_STS2	RSPTO1	A response is not received even after a minimum of 640 SDHI clock cycles elapse*1
				RSPTO0	A response is not received even after a minimum of 640 SDHI clock cycles elapse*2
Data timeout (excluding response timeout)		DTO		CRCBSYTO	After the CRC status token is received, the SDHI is busy for at least the period set*3
				CRCTO	After the write data is transmitted, the CRC status token is not received even after at least the period set*3 elapses
				RDTO	After the read command is issued, the read data is not received even after at least the period set*3 elapses
					After the read data is received, the next block read data is not received even after at least the period set*3 elapses
					After the SDHI exits the read wait state, the next block read data is not received even after at least the period set*3 elapses
				BSYTO1	After CMD12 is issued during the command sequence, the SDHI is busy for at least the period set*3
				BSYTO0	After the R1b response is received, the SDHI is busy for at least the period set*3 (a command other than CMD12 is issued during the command sequence)

Note 1. CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.
Note 2. CMD other than CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP bit is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.
Note 3. The period is set in the SD_OPTION.TOP[3:0] bits.

47.3.5 Command Without Data Transfer (SD/MMC)

The following legend is used for description of register read/write.

W (register name, value): Write to register

R (register name): Read from register

Figure 47.9 and Figure 47.10 show example flows.

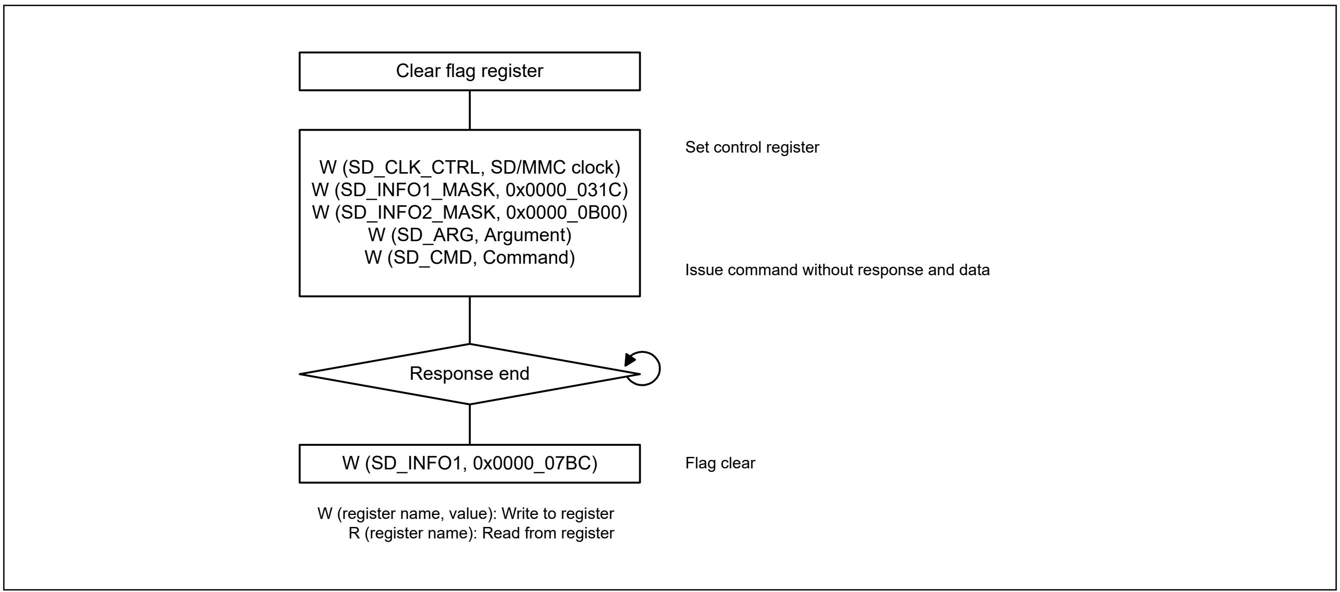


Figure 47.9 Example flow of command without response or data

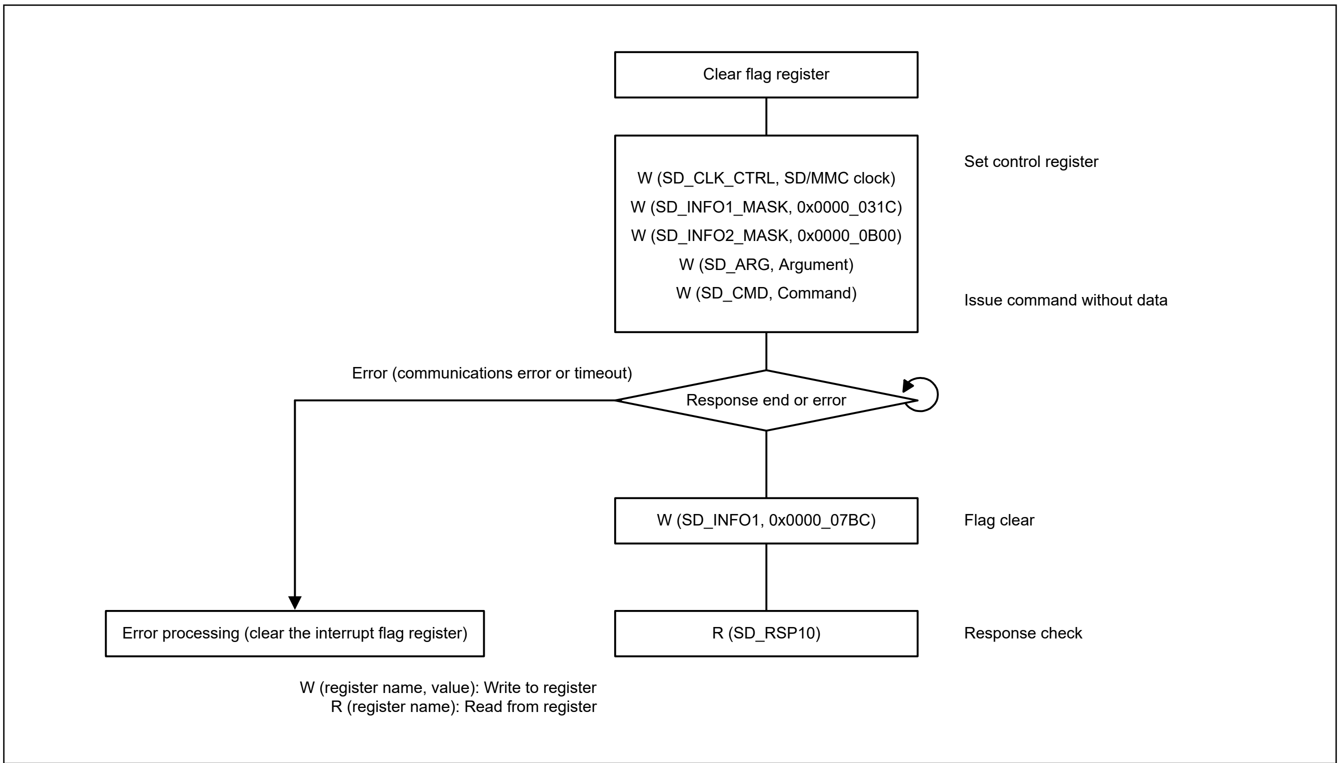


Figure 47.10 Example flow of command without data

47.3.5.1 Operation for Command Without Data Transfer

The operation is described in the following section.

(1) Command without response and data

a. Flag register clear
First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)

b. Control register set
Set the SD/MMC clock and interrupt masking. (SD_CLK_CTRL, SD_INFO1_MASK, and SD_INFO2_MASK)

c. Command issue
Set CMD argument in SD_ARG and write to SD_CMD.

Accordingly, CMD is issued, and the operation is started.

d. Flag clear
When transmission of a command is completed, RSPEND (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear RSPEND to 0.

(2) Command without data

a. Flag register clear
First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)

b. Control register set
Set the SD/MMC clock and interrupt masking. (SD_CLK_CTRL, SD_INFO1_MASK, and SD_INFO2_MASK)

c. Command issue
Set CMD argument in SD_ARG and write to the SD_CMD.
Accordingly, CMD is issued, and the operation is started.

d. Flag clear
When a response is received, RSPEND (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear RSPEND to 0.

e. Read a response from SD_RSP10. Additionally, perform error processing (clear the interrupt flag register) if a communication error or timeout occurs.

47.3.6 Single Block Read (SD/MMC)

Figure 47.11 shows an example flow of a single block read operation.

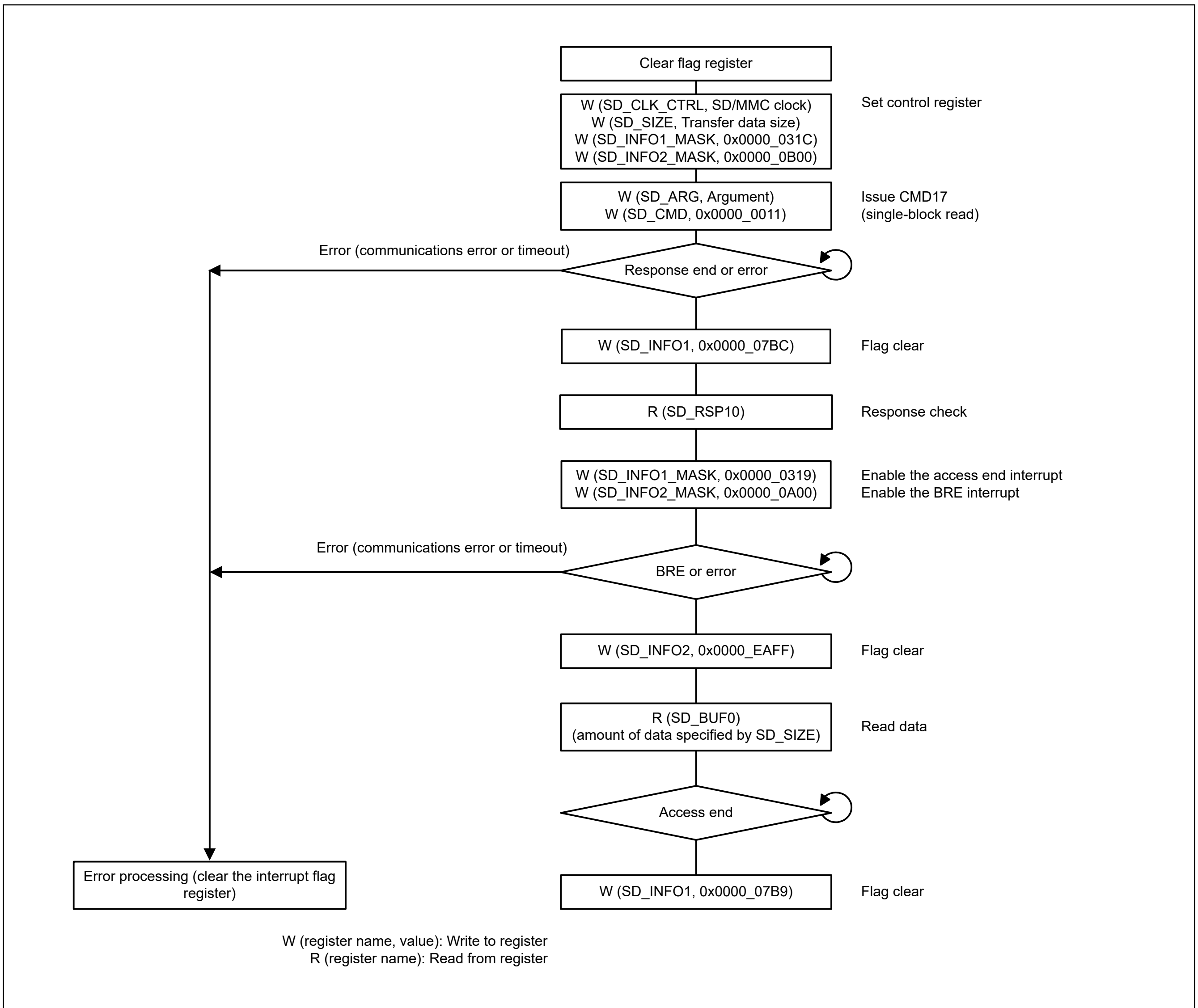


Figure 47.11 Example flow of single block read operation

47.3.6.1 Single Block Read Operation

The operation of the single block read is described as follows:

a. Flag register clear
First, clear the bits in the flag register (SD_INFO1 and SD_INFO2).

b. Control register set
Set the SD/MMC clock, transfer data size, interrupt mask (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK).

c. Command issue (CMD17)
Set CMD17 argument in SD_ARG and write 0x0000_0011 to SD_CMD. CMD17 is issued and the single block read operation is started.

d. Response check
On receiving the response, RSPEND (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear RSPEND to 0 and read the response from SD_RSP10. If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP or the IOABT bit in SDIO_MODE to 1. In addition, this causes CMD12 and CMD52 to not be issued. If the ACEND bit (access end) in SD_INFO1 is set, halting the command sequence also leads to the generation of an interrupt.

e. Data receive from SD card/MMC and data read

Write 0x0000_0319 to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0x0000_0A00 to SD_INFO2_MASK to enable the BRE interrupt. When the data received from the SD card/MMC is completed, the BRE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BRE bit to 0 and read the amount of data specified in SD_SIZE from SD_BUF0.
A communication error or timeout might be generated if data is being received while reading of SD_BUF0 is in progress.

f. Operation complete
When the data read from SD_BUF0 is completed, ACEND (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear ACEND to 0 to end the single block read operation.
Additionally, perform error processing (clear the interrupt flag register) if a communication error or timeout occurs.

47.3.7 Single Block Write (SD/MMC)

Figure 47.12 shows an example flow of a single block write operation.

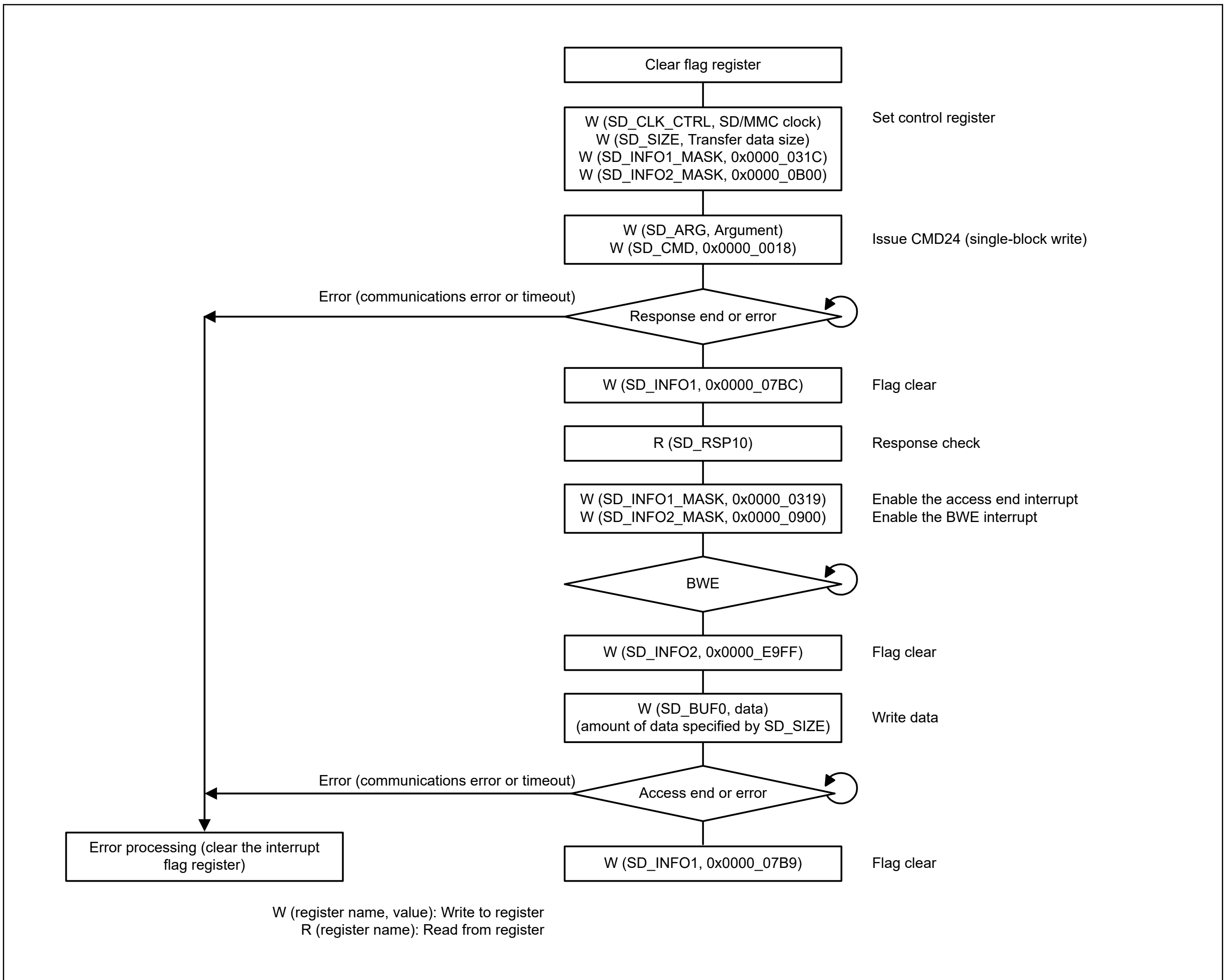


Figure 47.12 Example of single block write operation

47.3.7.1 Single Block Write Operation

The operation of the single block write is described as follows:

a. Flag register clear
First, clear the bits in the flag register (SD_INFO1 and SD_INFO2).

b. Control register set

Set the SD/MMC clock, transfer data size, interrupt mask (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK).

c. Command issue (CMD24)
Set CMD24 argument in SD_ARG and write 0x0000_0018 to SD_CMD. CMD24 is issued and the single block write operation is started.

d. Response check
On receiving the response, RSPEND (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear RSPEND to 0 and read the response from SD_RSP10. If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP or the IOABT bit in SDIO_MODE to 1. In addition, this causes CMD12 and CMD52 to not be issued. If the ACEND bit (access end) in SD_INFO is set, halting the command sequence also leads to the generation of an interrupt.

e. Data write and data transmit to SD card/MMC
Write 0x0000_0319 to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0x0000_0900 to SD_INFO2_MASK to enable the BWE interrupt. When SD_BUF0 is ready for the data to be written, the BWE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified in SD_SIZE to SD_BUF0. When the data write to SD_BUF0 is completed, data is transmitted to the SD card. Then, the CRC status and busy state are received from the SD card/MMC.
However, a communications error or timeout might be generated if data is being transmitted after writing to SD_BUF0.

f. Operation complete
When the CRC status and busy state are received from the SD card/MMC, ACEND (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the ACEND bit to 0 to end the single block write operation.
In addition, perform error processing (clear the interrupt flag register) if a communication error or timeout occurs.

47.3.8 Multiple Block Read (SD/MMC)

Figure 47.13 shows an example flow of a multiple block read operation.

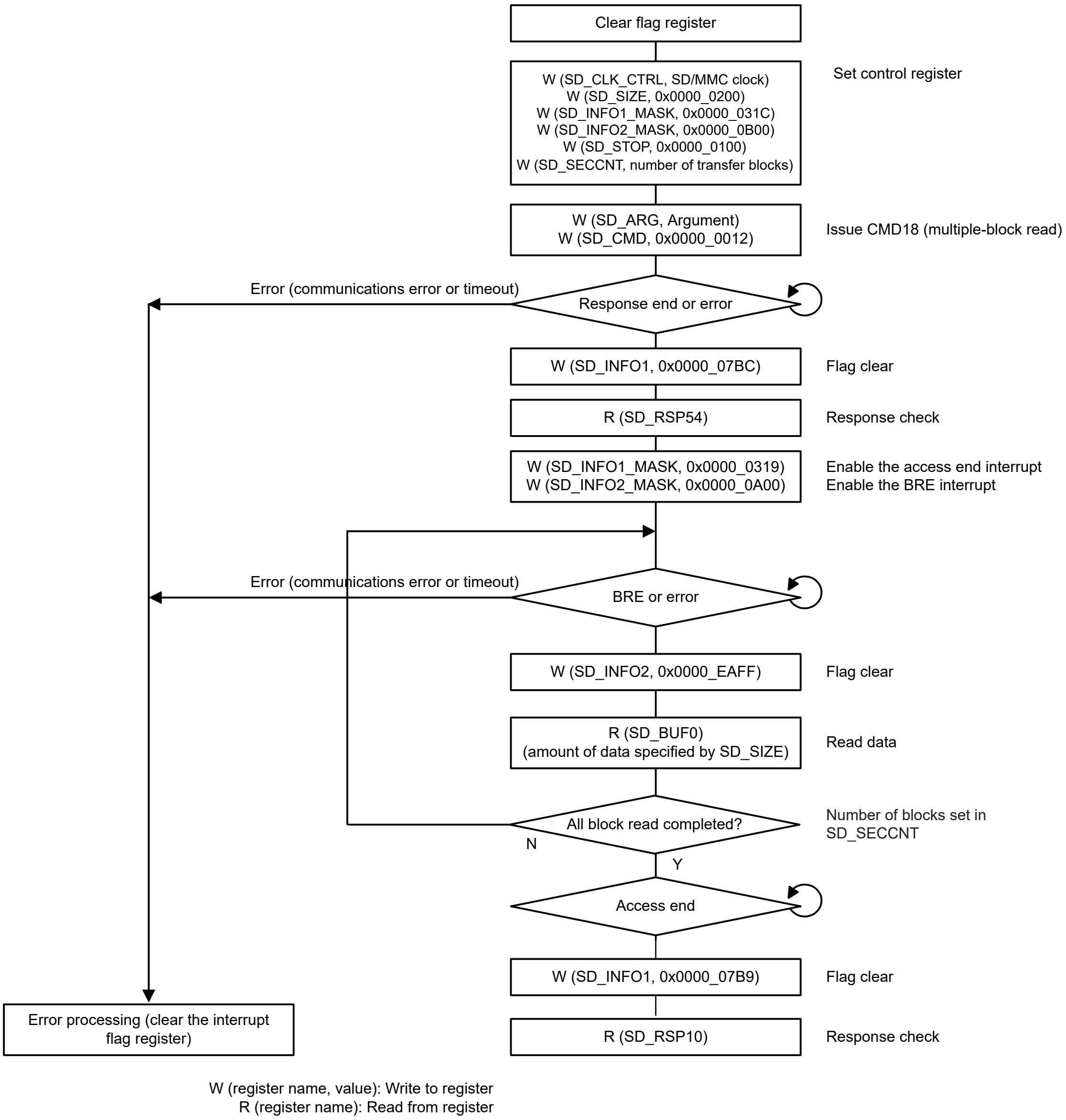


Figure 47.13 Example of multiple block read operation

47.3.8.1 Multiple Block Read Operation

The operation of the multiple block read is described as follows:

a. Flag register clear
First, clear the bits in the flag register (SD_INFO1 and SD_INFO2).

b. Control register set
Set the SD/MMC clock, transfer data size, interrupt mask (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK).
Set SEC in SD_STOP to 1, and set the number of transfer blocks in SD_SECCNT.

c. Command issue (CMD18)
Set CMD18 argument in SD_ARG and write 0x0000_0012 to SD_CMD. CMD18 is issued and the multiple block read operation is started.

d. Response check
On receiving the response, RSPEND (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear RSPEND to 0 and read the response from SD_RSP54. If the result of response decoding is an error, the command sequence can be

halted by setting the STP bit in SD_STP to 1. Setting the STP bit to 1 also causes CMD12 to be issued and the response received. If the command sequence is halted because the access end interrupt is enabled, an interrupt is generated when the ACEND bit (access end) bit in SD_INFO1 sets to 1 on completion of response reception. Clear the ACEND bit to 0 and read the response.

e. Data receive from SD card/MMC and data read
Write 0x0000_0319 to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0x0000_0A00 to SD_INFO2_MASK to enable the BRE interrupt. When one-block data received from the SD card/MMC is completed, the BRE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BRE bit to 0 and read the amount of data specified in SD_SIZE from SD_BUF0. Doing this repeats transfer of the number of blocks set in SD_SECCNT. However, a communication error or timeout might be generated if data is being received while reading of SD_BUF0 is in progress. CMD12 is automatically issued to stop multiblock transfer with the number of blocks that is set to SD_SECCNT and the response is received. At this point, CMD12 argument is automatically set to 0x0000_0000.

f. Operation complete
When all-block data read and the CMD12 response received are completed, ACEND (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear ACEND to 0 to read the response. This is the end of multiple block read operation. In addition, perform error processing (clear the interrupt flag register) if a communication error or timeout occurs.

47.3.9 Multiple Block Write (SD/MMC Using Internal Timer)

Figure 47.14 shows an example flow of a multiple block write using internal timer.

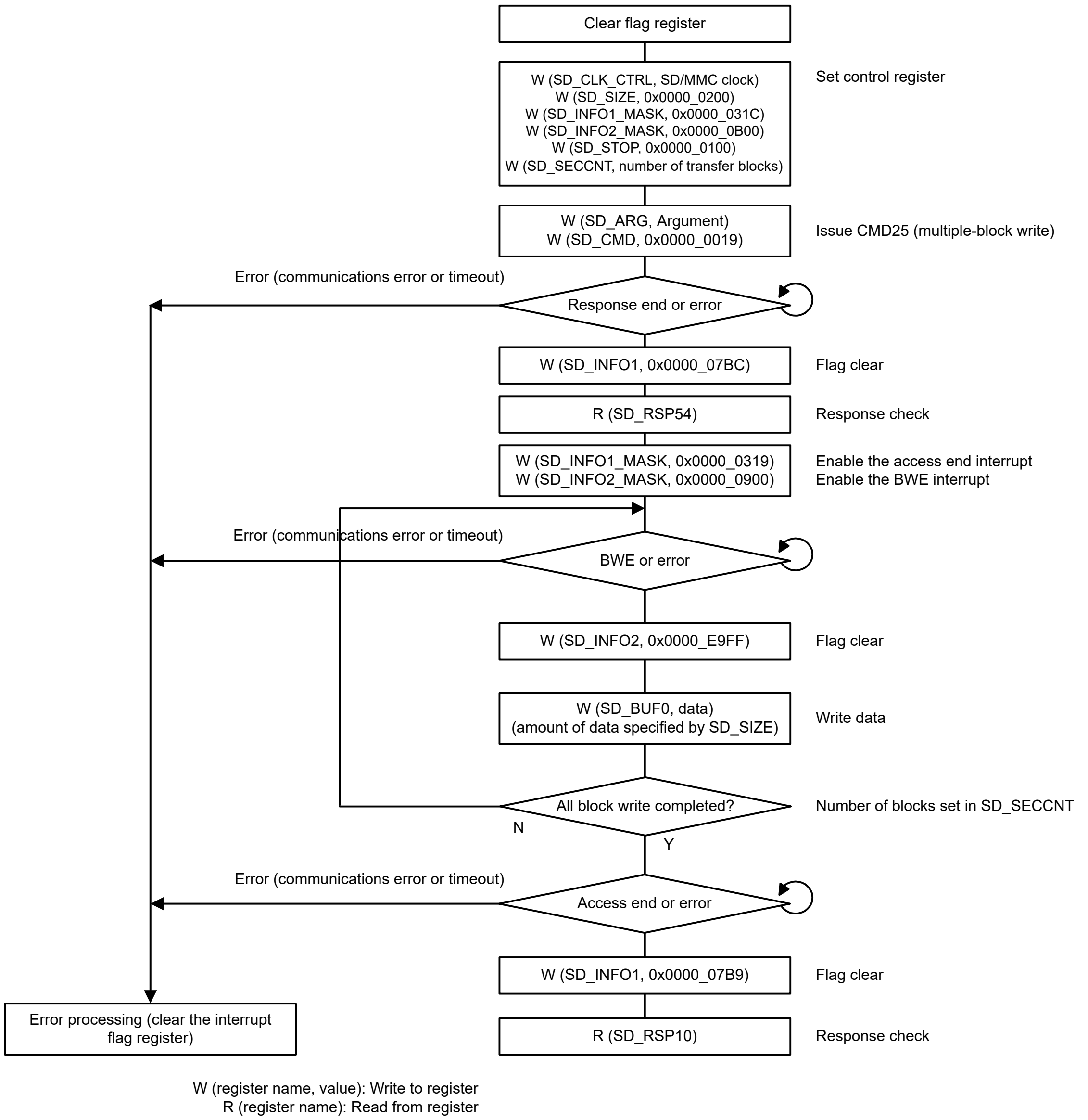


Figure 47.14 Example of multiple block write operation using internal timer

47.3.9.1 Multiple Block Write Operation Using Internal Timer

The operation of the multiple block write is described as follows:

a. Flag register clear
First, clear the bits in the flag register (SD_INFO1 and SD_INFO2).

b. Control register set
Set the SD/MMC clock, transfer data size, interrupt mask (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK).
Set the SEC bit in SD_STOP to 1, and set the number of transfer blocks in SD_SECCNT.

c. Command issue (CMD25)
Set CMD25 argument in SD_ARG and write 0x0000_0019 to SD_CMD. CMD25 is issued and the multiple block write operation is started.

d. Response check
On receiving the response, the RSPEND bit (response end) in SD_INFO1 is set to 1 to generate an interrupt.

Clear the RSPEND bit to 0 and read the response from SD_RSP54. If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP to 1. Setting the STP bit to 1 also causes CMD12 to be issued and the response received. If the command sequence is halted because the access end interrupt is enabled, an interrupt is generated by when the ACEND bit (access end) bit in SD_INFO1 sets to 1 on completion of response reception. Clear the ACEND bit to 0 and read the response.

e. Data write and data transmit to SD card/MMC
Write 0x0000_0319 to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0x0000_0900 to SD_INFO2_MASK to enable the BWE interrupt. When SD_BUF0 is ready for the data to be written, the BWE bit in the SD_INFO2 resister is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified in SD_SIZE to SD_BUF0. When the data write to SD_BUF0 is completed, data is transmitted to the SD card/MMC. The CRC status and busy state are received from the SD card/MMC. This repeats transfer of the number of blocks set in SD_SECCNT. However, a communication error or timeout might be generated if data is being received while writing to SD_BUF0 is in progress. CMD12 is automatically issued to stop multiblock transfer with the number of blocks which is set to SD_SECCNT and the response is received. At this point, CMD12 argument is automatically set to 0x0000_0000.

f. Operation complete
When all-block data transmit and the CRC status receive are completed, the ACEND bit (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the ACEND bit to 0 to read the response. This is the end of multiple block write operation. Additionally, perform error processing (clear the interrupt flag register) if a communications error or timeout occurs.

47.3.10 Multiple Block Write (MMC Using External Timer)

Figure 47.15 shows an example flow of a multiple block write using an external timer.

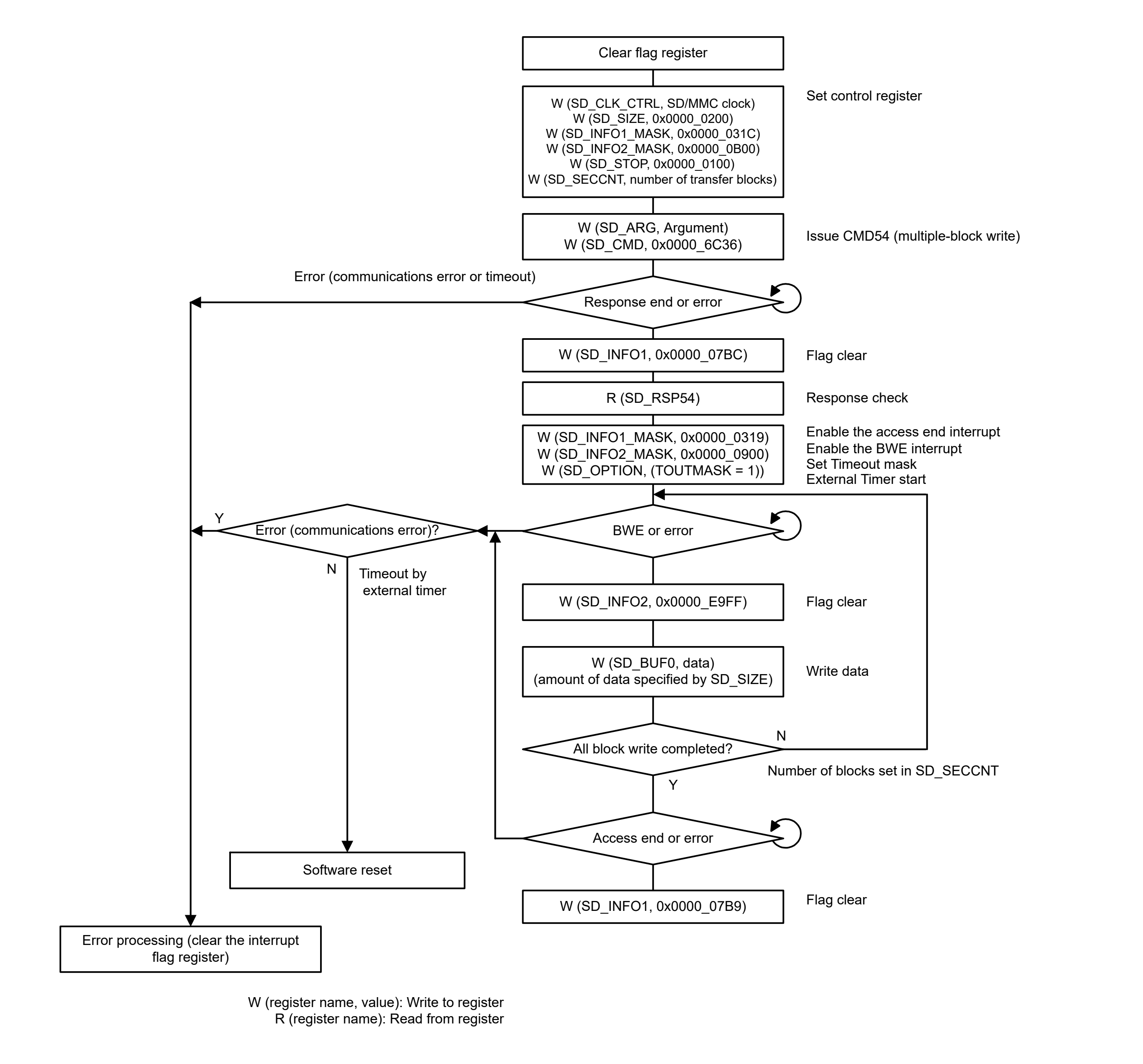


Figure 47.15 Example of multiple block write operation using external timer

47.3.10.1 Multiple Block Write Operation Using External Timer

The operation of the multiple block write is described as follows:

a. Flag register clear
First, clear the bits in the flag register (SD_INFO1 and SD_INFO2).

b. Control register set
Set the MMC clock, transfer data size, interrupt mask (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK).
Set the SEC bit in SD_STOP to 1, and set the number of transfer blocks in SD_SECCNT.

c. Command issue (CMD54)
Set CMD54 Argument in SD_ARG and write 0x0000_6C36 to SD_CMD. CMD54 is issued and the multiple block write operation is started.

d. Response check
On receiving the response, the RSPEND bit (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the RSPEND bit to 0 and read the response from SD_RSP54. If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP to 1. Setting the STP bit to 1 also causes CMD12 to be issued

and the response received. If the command sequence is halted because the access end interrupt is enabled, an interrupt is generated by when the ACEND bit (access end) bit in SD_INFO1 sets to 1 on completion of response reception. Clear the ACEND bit to 0 and read the response.

e. Data write and data transmit to MMC
Write 0x0000_0319 to SD_INFO1_MASK to enable the access end interrupt, write 0x0000_0900 to SD_INFO2_MASK to enable the BWE interrupt and set 1 to TOUTMASK of SD_OPTION to inactivate timeout. In addition, start external timer. When SD_BUF0 is ready for the data to be written, the BWE bit in the SD_INFO2 resister is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified in SD_SIZE to SD_BUF0. When the data write to SD_BUF0 is completed, data is transmitted to the MMC. The CRC status and busy state are received from the MMC. Doing this repeats transfer of the number of blocks set in SD_SECCNT. However, a communication error or timeout might be generated if data is being received while writing to SD_BUF0 is in progress.

f. Operation complete
When all-block data transmit and the CRC status receive are completed, the ACEND bit (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the ACEND bit to 0 to read the response. This is the end of multiple block write operation. Additionally, perform error processing (clear the interrupt flag register) if a communications error or timeout occurs when receiving response. Execute software reset if a timeout by external timer occurs when transmitting data.

47.3.11 IO_RW_DIRECT Command (SD: CMD52)

Figure 47.16 shows an example flow of an IO_DIRECT command (CMD52) operation.

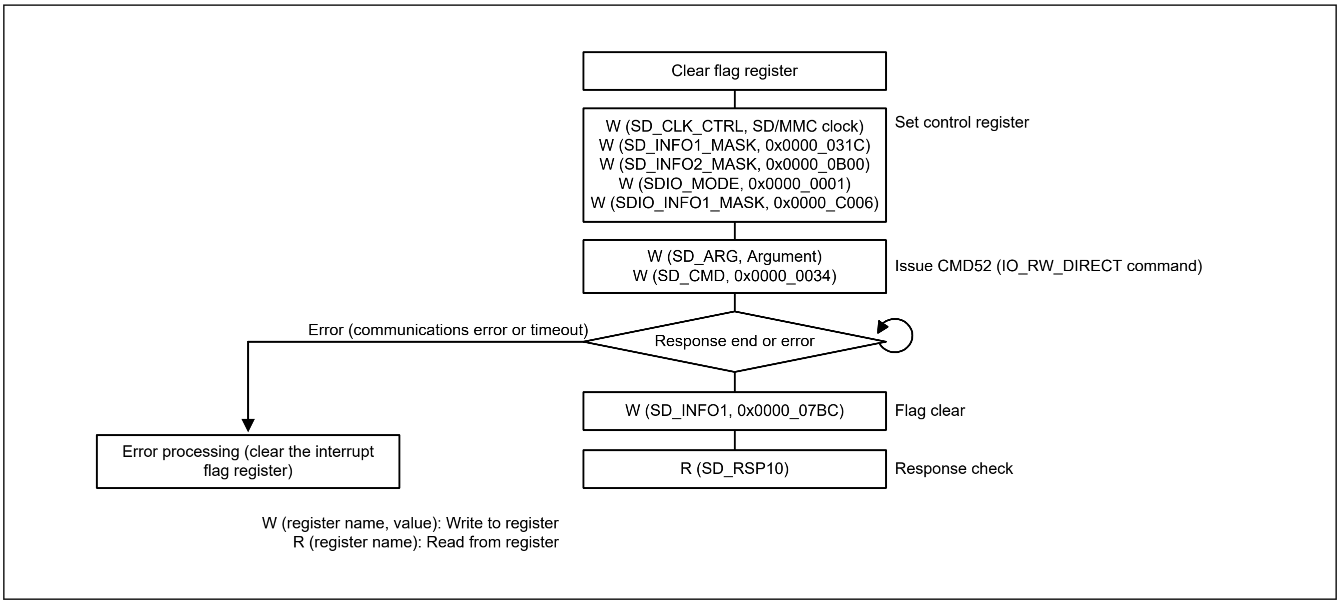


Figure 47.16 Example of IO_RW_DIRECT command (CMD52) operation

47.3.12 IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Read)

Figure 47.17 shows an example flow for a CMD53 multiple block read operation.

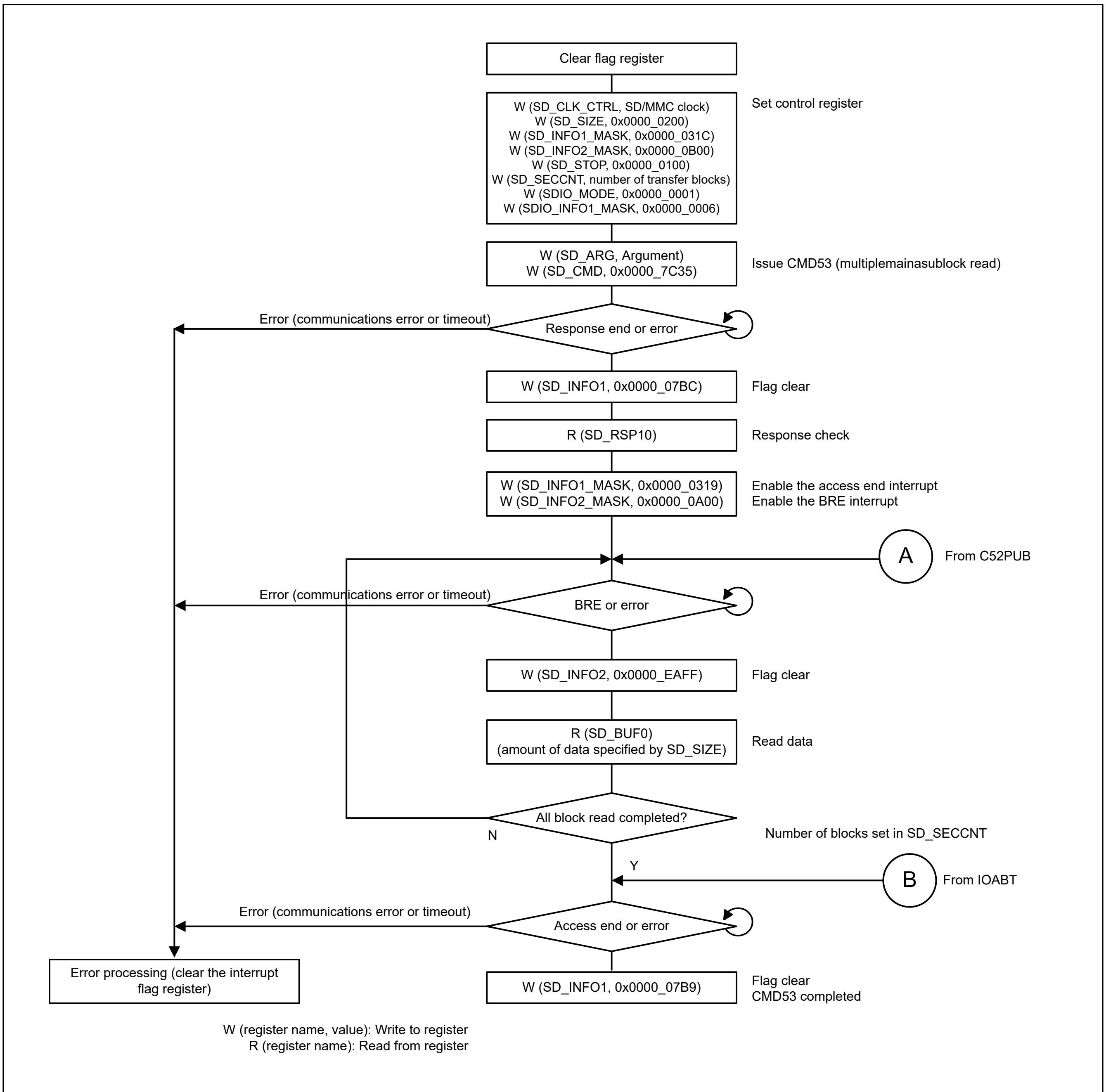


Figure 47.17 Example of IO_RW_EXTENDED command (CMD53) for multiple block read operation

Figure 47.18 shows an example flow when CMD52 (SDIO abort) is issued during a CMD53 multiple block read.

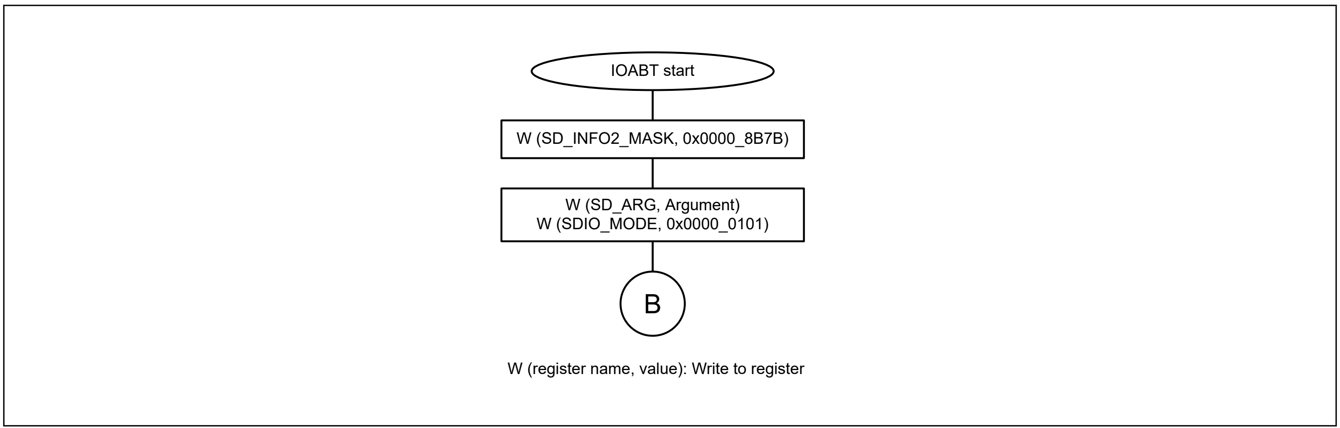


Figure 47.18 Flow when CMD52 (SDIO abort) is issued during a CMD53 multiple block read

Figure 47.19 shows an example flow when CMD52 (SDIO none abort) is issued at a CMD53 multiple block read while the SDHI is in the read wait state.

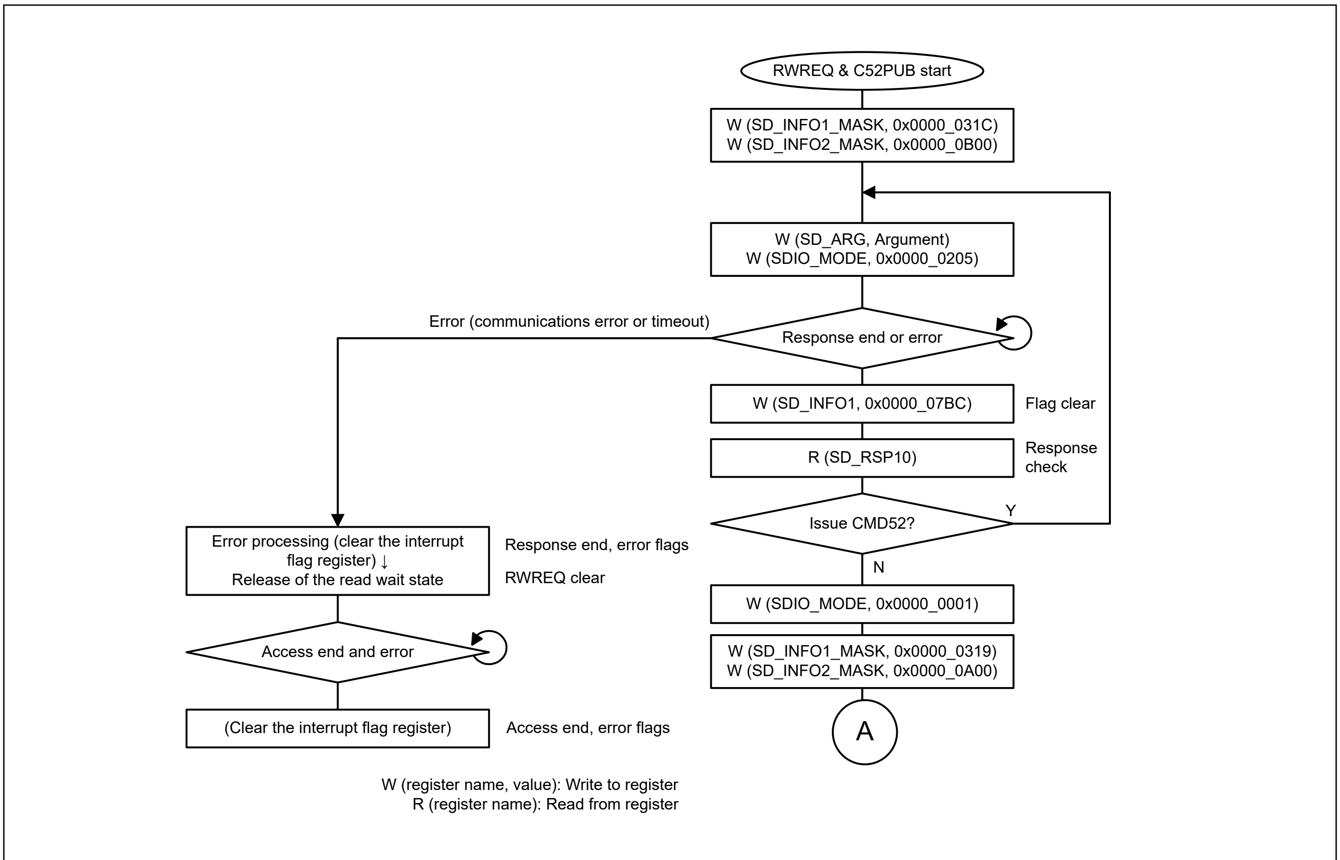


Figure 47.19 Flow when CMD52 (SDIO no abort) is issued during a CMD53 multiple block read while the SD Host Interface is in read wait state

47.3.13 IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Write)

Figure 47.20 shows an example flow for a CMD53 multiple block write.

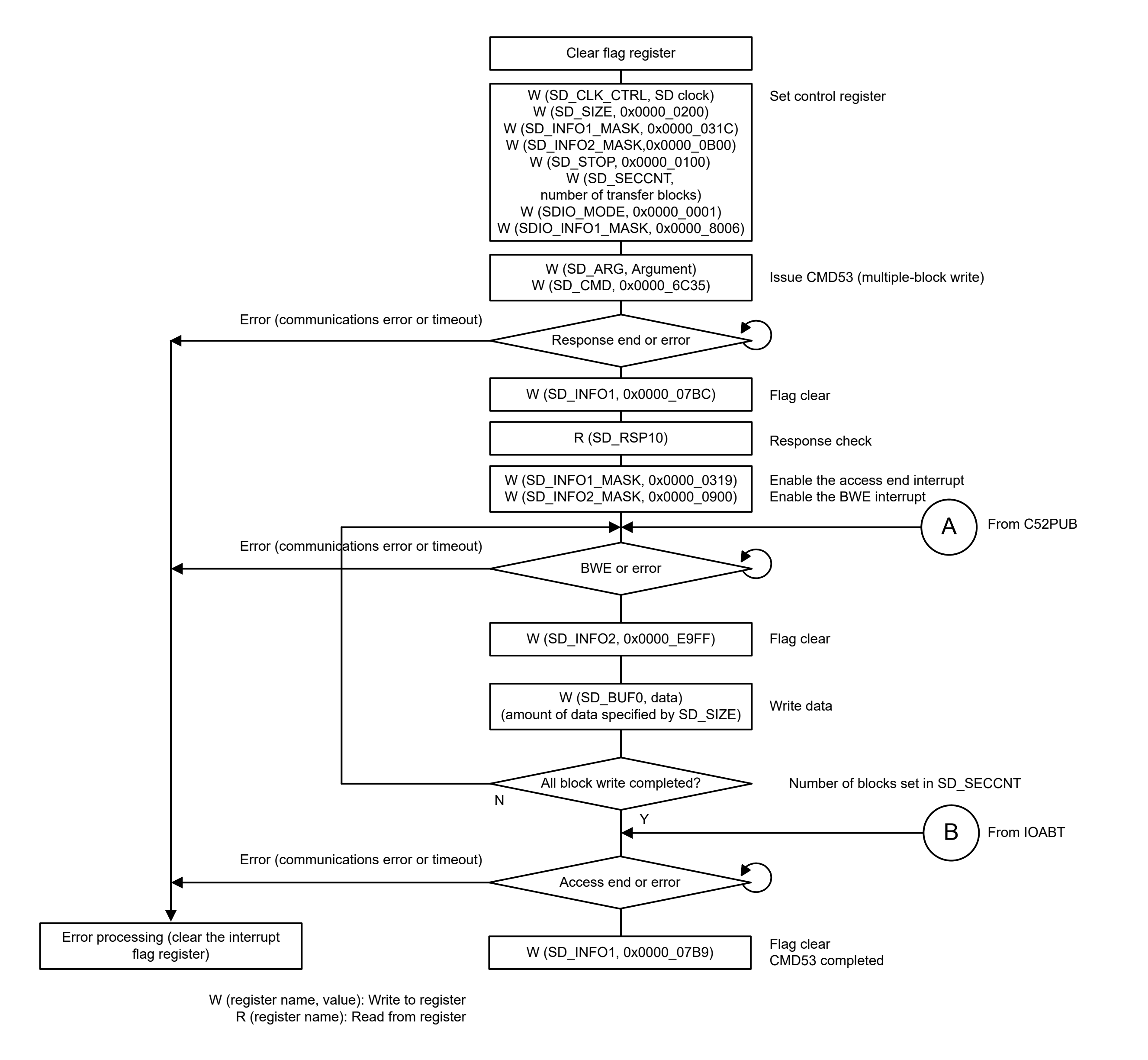


Figure 47.20 Example of IO_RW_EXTENDED command during a CMD53 multiple block write operation

Figure 47.21 shows an example flow when CMD52 (SDIO abort) is issued at CMD53 multiple block write.

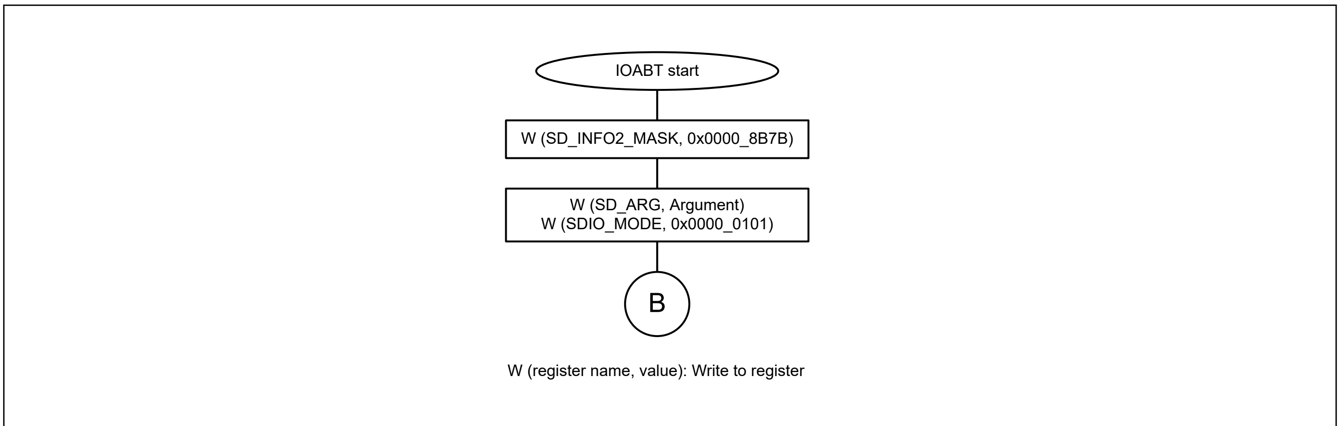


Figure 47.21 Flow when CMD52 (SDIO Abort) is issued during a CMD53 multiple block write

Figure 47.22 shows an example flow when CMD52 (SDIO none abort) is issued at CMD53 multiple block write.

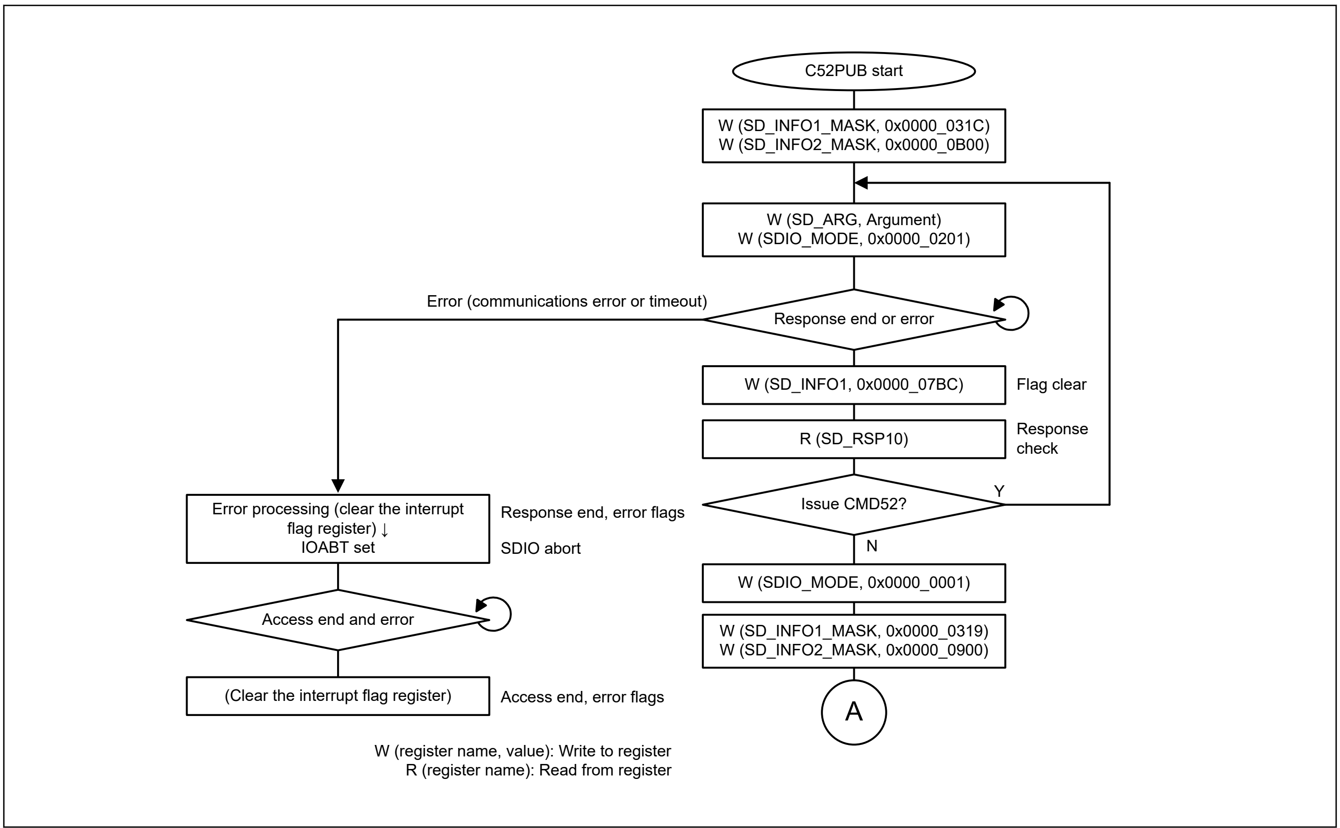


Figure 47.22 Flow when CMD52 (SDIO no abort) is issued during a CMD53 multiple block write

47.3.14 DMA Transfer (SD/MMC)

47.3.14.1 SD_BUF DMA transfer

Figure 47.23 shows an example flow for SD_BUF DMA read when CMD18 multiple block read is issued.

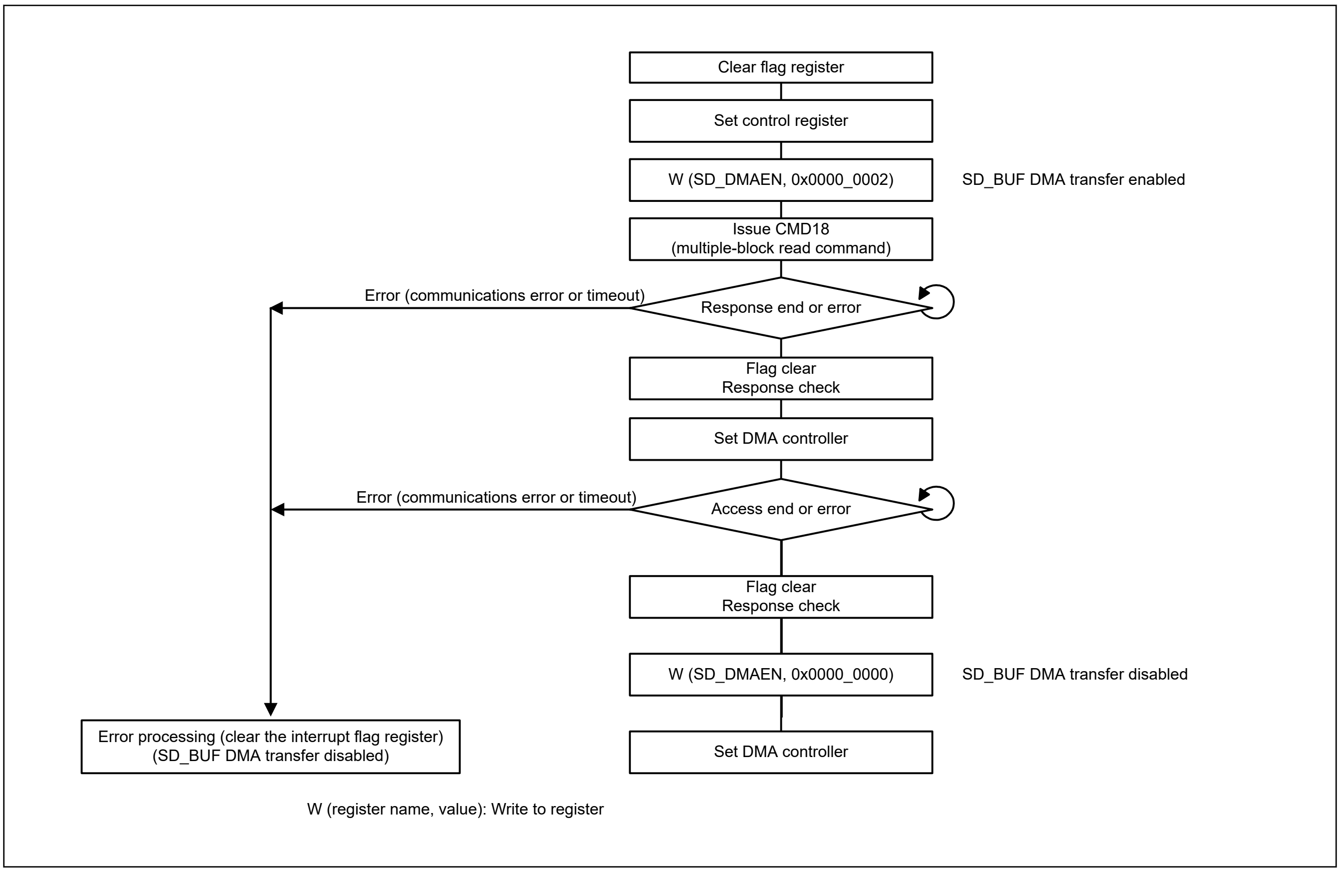


Figure 47.23 Example of SD_BUF_DMA read operation

Figure 47.24 shows an example flow for SD_BUF DMA write when CMD25 multiple block write is issued.

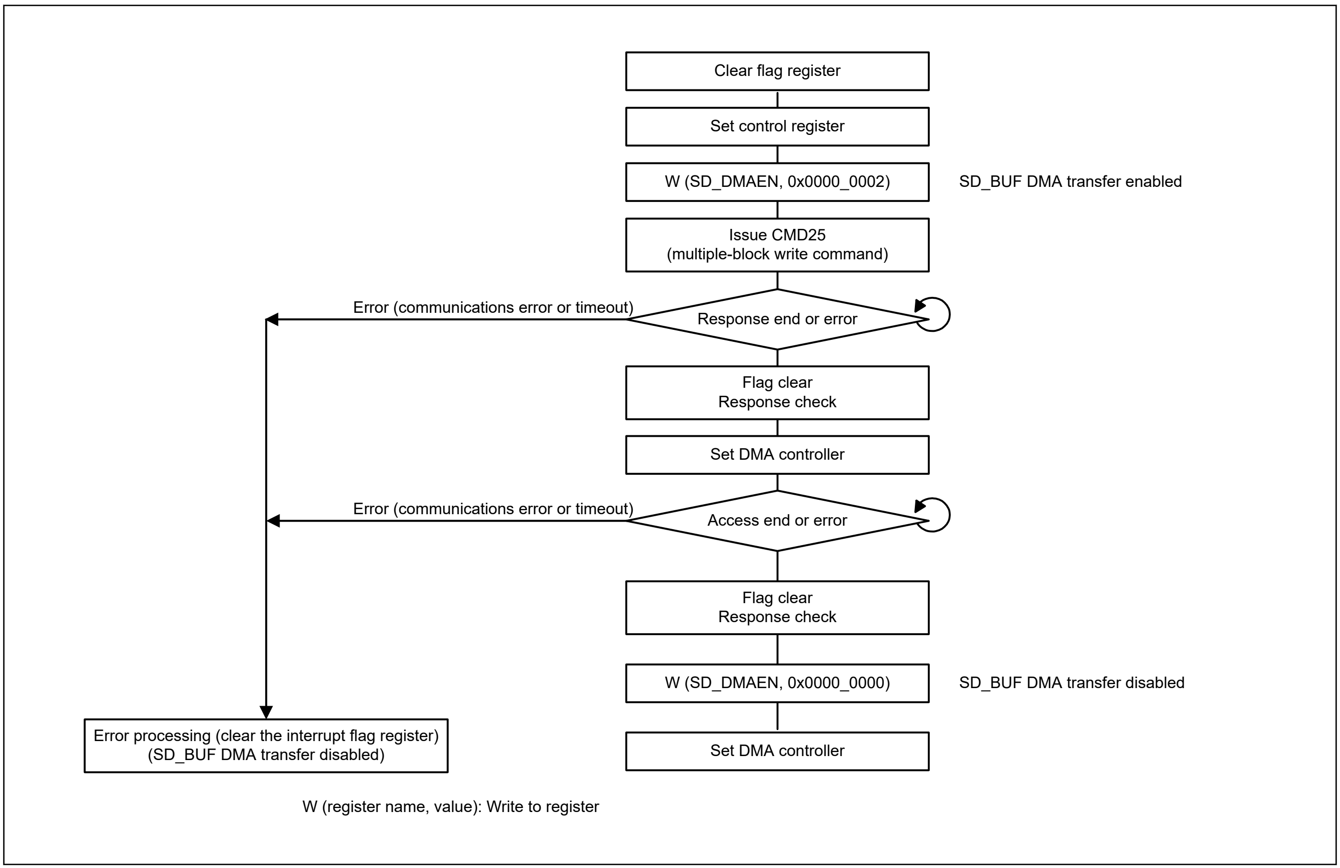


Figure 47.24 Example of SD_BUF_DMA write operation

47.3.15 Example of SD_CMD Register Setting

Table 47.8 and Table 47.9 list the SD_CMD register setting.

Table 47.8 Example SD_CMD register settings for SD (1 of 2)

Type	Command	Example SD_CMD register setting	Remark
CMD	CMD0	0x0000_0000	—
	CMD2	0x0000_0002	—
	CMD3	0x0000_0003	—
	CMD4	0x0000_0004	—
	CMD5	0x0000_0705 or 0x0000_0005	—
	CMD6	0x0000_1C06 or 0x0000_0006	—
	CMD7	0x0000_0007	When the card is placed in the deselected state, the response timeout flag sets because there is no response.
	CMD8	0x0000_0408 or 0x0000_0008	—
	CMD9	0x0000_0009	—
	CMD10	0x0000_000A	—
	CMD11	0x0000_040B or 0x0000_000B	—
	CMD12	0x0000_000C	—
	CMD13	0x0000_000D	—
	CMD15	0x0000_000F	—
	CMD16	0x0000_0010	—
	CMD17	0x0000_0011	—
	CMD18	0x0000_0012	With automatic CMD12
	CMD20	0x0000_0514 or 0x0000_0014	—
	CMD24	0x0000_0018	—
	CMD25	0x0000_0019	With automatic CMD12
	CMD27	0x0000_001B	—
	CMD28	0x0000_001C	—
	CMD29	0x0000_001D	—
	CMD30	0x0000_001E	—
	CMD32	0x0000_0020	—
	CMD33	0x0000_0021	—
	CMD38	0x0000_0026	—
	CMD42	0x0000_002A	—
	CMD52	0x0000_0434 or 0x0000_0034	—
	CMD53	0x0000_1C35	Single read
		0x0000_0C35	Single write
		0x0000_7C35	Multiple read
		0x0000_6C35	Multiple write
		0x0000_0035	The value on the left can be set for both single and multiple operations. However, the CF39 bit in SD_ARG must be set as follows. Read: 0 Write: 1
	CMD55	0x0000_0037	—
	CMD56	0x0000_0038	—

Table 47.8 Example SD_CMD register settings for SD (2 of 2)

Type	Command	Example SD_CMD register setting	Remark
ACMD	ACMD6	0x0000_0046	—
	ACMD13	0x0000_004D	—
	ACMD22	0x0000_0056	—
	ACMD23	0x0000_0057	—
	ACMD41	0x0000_0069	—
	ACMD42	0x0000_006A	—
	ACMD51	0x0000_0073	—

Table 47.9 Example SD_CMD register settings for MMC

Type	Command	Example SD_CMD register setting	Remark
CMD	CMD0	0x0000_0000	—
	CMD1	0x0000_0701	—
	CMD2	0x0000_0002	—
	CMD3	0x0000_0003	—
	CMD4	0x0000_0004	—
	CMD5	0x0000_0505	—
	CMD6	0x0000_0506	(with response busy)
		0x0000_0406	(without response busy)
	CMD7	0x0000_0007	When the card is placed in the deselected state, the response timeout flag sets because there is no response.
	CMD8	0x0000_1C08	—
	CMD9	0x0000_0009	—
	CMD10	0x0000_000A	—
	CMD12	0x0000_000C	—
	CMD13	0x0000_000D	—
	CMD14	0x0000_1C0E	Required setting: SD_IFMODE = 0x0000_0100 (CRC check is invalid)
	CMD15	0x0000_000F	—
	CMD16	0x0000_0010	—
	CMD17	0x0000_0011	—
	CMD18	0x0000_7C12	Pre-defined
	CMD19	0x0000_0C13	Required setting: SD_IFMODE = 0x0000_0100 (CRC check is invalid)
	CMD21	0x0000_1C15	DDR mode is inhibited
	CMD23	0x0000_0017	—
	CMD24	0x0000_0018	—
	CMD25	0x0000_6C19	Pre-defined
	CMD26	0x0000_0C1A	—
	CMD27	0x0000_001B	—
	CMD28	0x0000_001C	—
	CMD29	0x0000_001D	—
	CMD30	0x0000_001E	—
	CMD31	0x0000_1C1F	—
	CMD35	0x0000_0423	—
	CMD36	0x0000_0424	—
	CMD38	0x0000_0026	—
	CMD39	0x0000_0427	—
	CMD40	0x0000_0428	—
	CMD42	0x0000_002A	—
	CMD49	0x0000_0C31	—
	CMD53	0x0000_7C35	—
	CMD54	0x0000_6C36	—
	CMD55	0x0000_0037	—
	CMD56	0x0000_0038	—

47.4 Usage Notes

47.4.1 SD_BUF Illegal Write Access (SD/MMC)

When writing data to SD_BUF0 after the single block write or multi block write command is issued, the data of the size specified in SD_SIZE must be written.

If the data exceeds the size specified in SD_SIZE is written, the ERR4 bit in SD_INFO2 is set to 1. In addition, the data written to SD_BUF0 might not be transmitted and the SD_CLK_CTRLEN bit in SD_INFO2 is held at the value of 0. If this occurs, clearing the SDRST bit in SOFT_RST to 0 and then restoring its value to 1 clears the SD_CLK_CTRLEN bit to 1.

However, this does not apply to the single byte or three bytes when the SD_SIZE setting is odd, or to the fraction of bytes when the SD_SIZE setting is even (the 2 bytes that are not in a 4-byte unit), because the portion of dummy data writing is regarded as excess data and ignored.

47.4.2 Block Number Constraint for Multiple Block Read (SD)

When performing a multiple block read of one or two blocks, depending on the timing with which the SD card response register is read, the response value might not be read properly. To prevent this, do one of the following:

1. When receiving one or two blocks of data, use single block reading.
2. Read the response to CMD18 from SD_RSP54.

47.4.2.1 Mechanism of Incorrect Reading

Figure 47.25 shows the processing flows of the SDHI (hardware) operation and software operation when a multiple block read is performed on two blocks. As shown in the incorrect operation in Figure 47.25, when an interrupt is generated on reception of the CMD18 response and the timing with which the SD card response register (SD_RSP10) is read by the interrupt is delayed, the data during the CMD12 response reception or the CMD12 response might be read. This problem does not occur for multiple block reads of three or more blocks, because CMD12 is not issued until the block of data is read. The problem also does not occur for multiple block writes, because the CMD25 response is read before the block of data is sent.

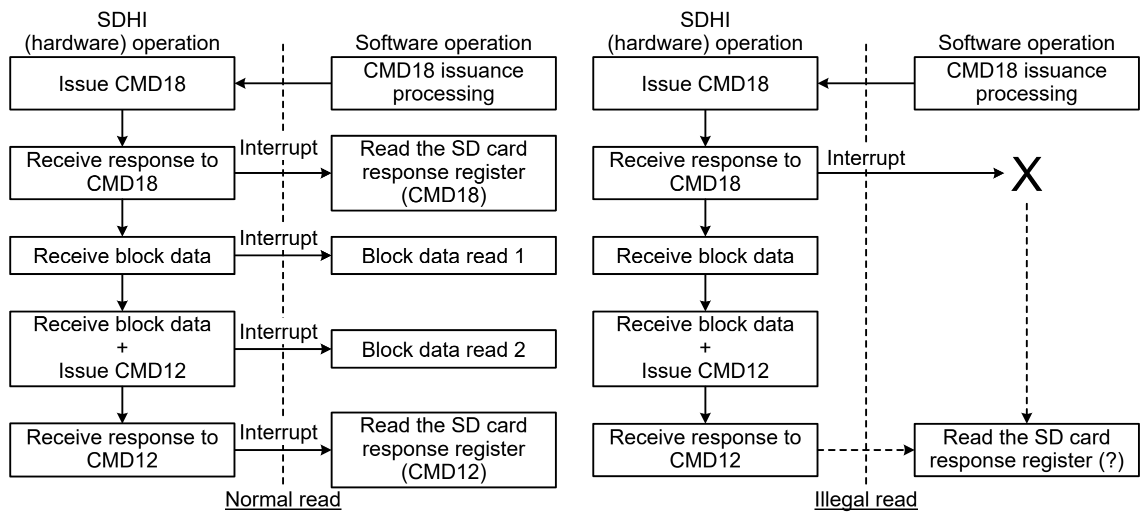


Figure 47.25 Multiple block read operation flow chart (two blocks)

47.4.3 Automatic Control of SD/MMC Clock Output (SD/MMC)

In the SD Card/MMC standard, 74 cycles of SD/MMC clock must be output before initialization of the card. For this reason, use automatic control of SD/MMC clock output after 74 cycles of SD/MMC clock are output. In addition, if automatic control of SD/MMC clock output was in use, SD/MMC clock output is stopped on completion of the sequence for a communications error or timeout. When state transitions within the SD card/MMC are necessary after completion of the sequence, release automatic control of SD/MMC clock output and restart supply of the SD/MMC clock to the SD card/MMC.

47.4.4 Control of the C52PUB Setting for Multiple Block Write (SD)

If the C52PUB bit in SDIO_MODE is set to 1 during a sequence of multiple block write because of CMD53, CMD52 is not issued until SD_BUF becomes empty. For this reason, set the C52PUB bit after suspending writing to SD_BUF by using one of the following procedures, as appropriate:

(a) When DMA transfer is not in use

1. Before setting the C52PUB bit, suspend writing to SD_BUF by making the setting in SD_INFO2 to disable BWE interrupts.
2. Set the C52PUB bit in SDIO_MODE to 1 (so that CMD52 is issued when SD_BUF becomes empty).
3. After the RSPEND interrupt processing in SD_INFO1 because the issuing of CMD52 is completed, restart writing to SD_BUF by making the setting in SD_INFO2 to enable BWE interrupts.

(b) When DMA transfer is in use

1. Every time DMA transfer of the value set in SD_SIZE × n blocks (where n = 1, 2,…) proceeds, suspend writing to SD_BUF by DMA transfer before the C52PUB bit is set.
2. Set the C52PUB bit in SDIO_MODE to 1 (so that CMD52 is issued when SD_BUF becomes empty).
3. After the RSPEND interrupt processing in SD_INFO1 because the issuing of CMD52 is completed, restart writing to SD_BUF by DMA transfer.

47.4.5 Notes on SD_CLK_CTRL Register Settings (SD/MMC)

When the SD_CLK_CTRLEN bit in SD_INFO2 is 0, SD_CLK_CTRL cannot be written to. Before writing to SD_CLK_CTRL, you must check that the SD_CLK_CTRLEN bit in SD_INFO2 is 1.

47.4.6 Specification Limitations

1. The Suspend/Resume operation of the SDIO is not supported.
2. The SPI bus is not supported. (SD/MMC)
3. The shared bus and 8-bit SD bus of the embedded SDIO are not supported.
4. Stream transfer of MMC is not supported.
5. High Priority Interrupt (HPI) of MMC is not supported.
6. Boot Operation/Alternative Boot Operation of MMC is not supported.
7. Open-ended multiple block transfer of MMC is not supported.

47.4.7 STP Bit Setting During Multiple Block Read (SD/MMC)

During execution of multiple block read with automatic CMD12 execution by setting the SEC bit in SD_STOP to 1, even if the STP bit in SD_STOP is set to 1 to forcibly stop the execution, the command sequence might not stop depending on the timing of setting the STP bit.

To avoid this, when setting the STP bit in SD_STOP to 1 during multiple block transfer, clear the SEC bit in SD_STOP to 0 at the same time. (Even when the SD_CLK_CTRLEN bit in SD_INFO2 is 0, change the SEC bit from 1 to 0.)

When the command sequence does not stop because the SEC bit was not cleared to 0, the command sequence can be stopped by clearing the SDRST bit in SOFT_RST to 0.

When forcibly terminating the CMD53 multiple block transfer through the IOABT bit in SDIO_MODE, you must leave the SEC bit in SD_STOP as 1.

47.4.8 Register Setting Notes

1. All registers in section 47.2. Register Descriptions are accessed in 32-bit access-only.
2. When setting registers, set them after the I/O Port Register setting.

47.4.9 Module-stop Function

SDHI operation can be disabled or enabled using Module Stop Control Register C (MSTPCRC). The SDHI module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Mode.

47.4.10 Software Standby Mode

Before entering Software Standby mode, confirm the SD_INFO2.CBSY bit is 0.

48. Cyclic Redundancy Check (CRC)

48.1 Overview

The Cyclic Redundancy Check (CRC) calculator generates CRC codes to detect errors in the data. The bit order of CRC calculation results can be switched for LSB-first or MSB-first communication. Additionally, various CRC-generation polynomials are available. The snoop function allows monitoring reads from and writes to specific addresses. This function is useful in applications that require CRC code to be generated automatically in certain events, such as monitoring writes to the serial transmit buffer and reads from the serial receive buffer.

Table 48.1 lists the CRC calculator specifications and Figure 48.1 shows a block diagram.

Table 48.1 CRC calculator specifications

Item	Description	
Data size	8-bit	32-bit
Data for CRC calculation*1	CRC code generated for data in 8n-bit units (where n is a natural number)	CRC code generated for data in 32n-bit units (where n is a natural number)
CRC processor unit	Operation executed on 8 bits in parallel	Operation executed on 32 bits in parallel
CRC generating polynomial	One of three generating polynomials that is selectable: [8-bit CRC] • $X^8 + X^2 + X + 1$ (CRC-8) [16-bit CRC] • $X^{16} + X^{15} + X^2 + 1$ (CRC-16) • $X^{16} + X^{12} + X^5 + 1$ (CRC-CCITT)	One of two generating polynomials that is selectable: [32-bit CRC] • $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$ (CRC-32) • $X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$ (CRC-32C)
CRC calculation switching	The bit order of CRC calculation results can be switched for LSB-first or MSB-first communication	
Module-stop function	Module-stop state can be set to reduce power consumption	
CRC snoop	Monitor reads from and writes to a certain register address	
TrustZone Filter	Security and Privilege attribution can be set	

Note 1. This function cannot divide data used in CRC calculations. Write data in 8-bit or 32-bit units.

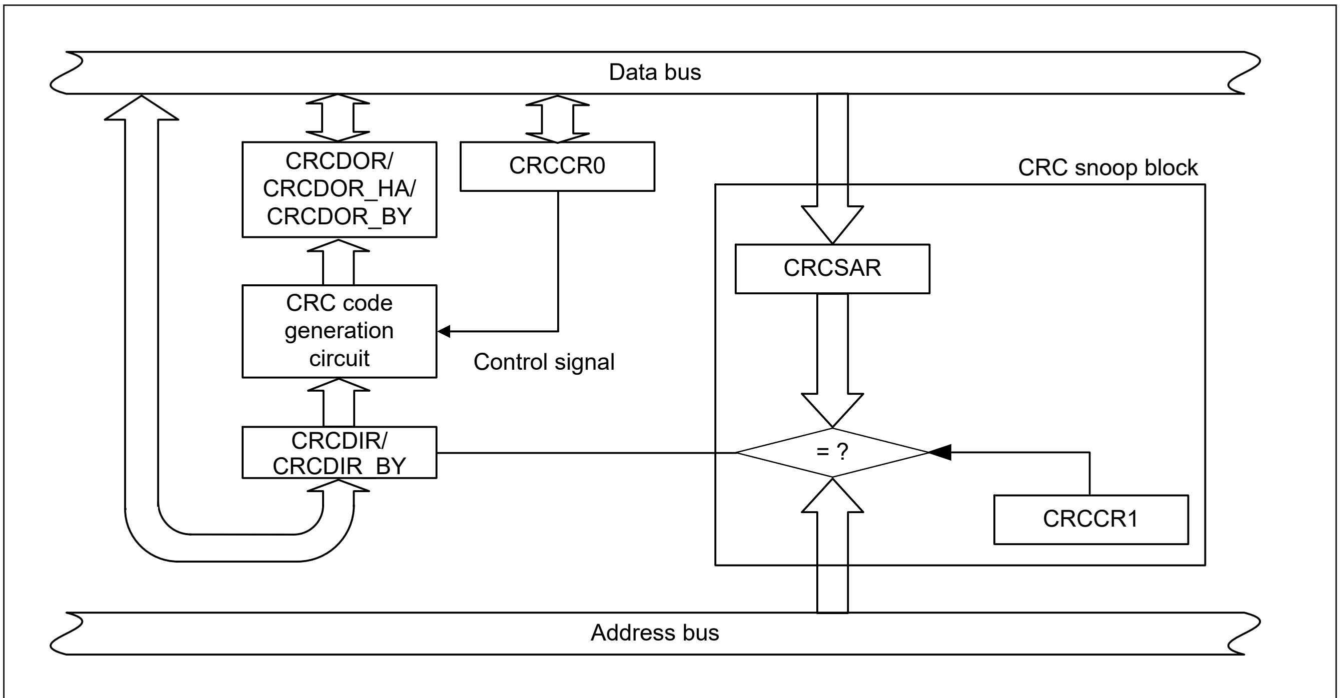


Figure 48.1 CRC calculator block diagram

48.2 Register Descriptions

48.2.1 CRCCR0 : CRC Control Register 0

Base address: CRC = 0x4031_0000
CRC_NS = 0x5031_0000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DORCLR	LMS	—	—	—	GPS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	GPS[2:0]	CRC Generating Polynomial Switching 0 0 1: 8-bit CRC-8 ($X^8 + X^2 + X + 1$) 0 1 0: 16-bit CRC-16 ($X^{16} + X^{15} + X^2 + 1$) 0 1 1: 16-bit CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$) 1 0 0: 32-bit CRC-32 ($X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$) 1 0 1: 32-bit CRC-32C ($X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$) Others: No calculation is executed	R/W
5:3	—	These bits are read as 0. The write value should be 0.	R/W
6	LMS	CRC Calculation Switching 0: Generate CRC code for LSB-first communication 1: Generate CRC code for MSB-first communication	R/W
7	DORCLR	CRCDOR/CRCDOR_HA/CRCDOR_BY Register Clear 0: No effect 1: Clear the CRCDOR/CRCDOR_HA/CRCDOR_BY register	W

Note: S-TYPE3, P-TYPE3

GPS[2:0] bits (CRC Generating Polynomial Switching)

The GPS[2:0] bits select the CRC generating polynomial.

LMS bit (CRC Calculation Switching)

The LMS bit selects the bit order of generated CRC code. Transmit the lower byte of the CRC code first for LSB-first communication and the upper byte first for MSB-first communication. For details on transmitting and receiving CRC code, see section 48.3. Operation.

DORCLR bit (CRCDOR/CRCDOR_HA/CRCDOR_BY Register Clear)

Write 1 to the DORCLR bit to set the CRCDOR/CRCDOR_HA/CRCDOR_BY register to 0x00000000. This bit is read as 0. Only 1 can be written to it.

48.2.2 CRCCR1 : CRC Control Register 1

Base address: CRC = 0x4031_0000
CRC_NS = 0x5031_0000

Offset address: 0x01

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CRCSEN	CRCSWR	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	CRCSWR	Snoop-On-Write/Read Switch 0: Snoop-on-read 1: Snoop-on-write	R/W
7	CRCSEN	Snoop Enable 0: Disabled 1: Enabled	R/W

Note: S-TYPE3, P-TYPE3

CRCSWR bit (Snoop-On-Write/Read Switch)

The CRCSWR bit selects the direction of access in the CRC snoop function.

When this bit is set to 0 (initial value), the CRC snoop operation to read a specific register is enabled. Similarly, when this bit is set to 1, the CRC snoop operation to write a specific register is enabled.

CRCSEN bit (Snoop Enable)

When the CRCSEN bit is set to 1, the CRC snoop operation is enabled. When this bit is set to 0, the CRC snoop operation is disabled.

48.2.3 CRCDIR/CRCDIR_BY : CRC Data Input Register

Base address: CRC = 0x4031_0000
CRC_NS = 0x5031_0000

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	CRC input data The CRCDIR register is a 32-bit read/write register to write data for CRC-32 or CRC-32C calculation. The CRCDIR_BY (CRCDIR[31:24]) is an 8-bit read/write register to write data for CRC-8, CRC-16, or CRC-CCITT calculation.	R/W

Note: S-TYPE3, P-TYPE3

48.2.4 CRCDOR/CRCDOR_HA/CRCDOR_BY : CRC Data Output Register

Base address: CRC = 0x4031_0000
CRC_NS = 0x5031_0000

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	CRC output data The CRCDOR register is a 32-bit read/write register for CRC-32 or CRC-32C calculation. The CRCDOR_HA (CRCDOR[31:16]) register is a 16-bit read/write register for CRC-16 or CRC-CCITT calculation. The CRCDOR_BY (CRCDOR[31:24]) register is an 8-bit read/write register for CRC-8 calculation. Because its initial value is 0x00000000, rewrite the CRCDOR/CRCDOR_HA/ CRCDOR_BY register to perform the calculations using a value other than the initial value. Data written to the CRCDIR/CRCDIR_BY register is CRC calculated and the result is stored in the CRCDOR/CRCDOR_HA/CRCDOR_BY register. If the CRC code is calculated following the transferred data and the result is 0x00000000, there is no CRC error.	R/W

Note: S-TYPE3, P-TYPE3

48.2.5 CRCSAR : Snoop Address Register

Base address: CRC = 0x4031_0000
CRC_NS = 0x5031_0000

Offset address: 0x0C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	CRCSA[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
13:0	CRCSA[13:0]	Register Snoop Address These bits store the TDR or RDR address in the SCI module to snoop	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

CRCSA[13:0] bits (Register Snoop Address)

The CRCSA[13:0] bits specify the lower address 14 bits of the register monitored by the CRC snoop operation.

Only the following addresses can be used for the CRCSA[13:0] bits:

[Secure region address]

- 0x4035_8004: SCI0.TDR, 0x4035_8000: SCI0.RDR
- 0x4035_8104: SCI1.TDR, 0x4035_8100: SCI1.RDR
- 0x4035_8204: SCI2.TDR, 0x4035_8200: SCI2.RDR
- 0x4035_8304: SCI3.TDR, 0x4035_8300: SCI3.RDR
- 0x4035_8404: SCI4.TDR, 0x4035_8400: SCI4.RDR
- 0x4035_8504: SCI5.TDR, 0x4035_8500: SCI5.RDR
- 0x4035_8604: SCI6.TDR, 0x4035_8600: SCI6.RDR
- 0x4035_8704: SCI7.TDR, 0x4035_8700: SCI7.RDR
- 0x4035_8804: SCI8.TDR, 0x4035_8800: SCI8.RDR
- 0x4035_8904: SCI9.TDR, 0x4035_8900: SCI9.RDR

[Non-secure region address]

- 0x5035_8004: SCI0_NS.TDR, 0x5035_8000: SCI0_NS.RDR
- 0x5035_8104: SCI1_NS.TDR, 0x5035_8100: SCI1_NS.RDR
- 0x5035_8204: SCI2_NS.TDR, 0x5035_8200: SCI2_NS.RDR
- 0x5035_8304: SCI3_NS.TDR, 0x5035_8300: SCI3_NS.RDR
- 0x5035_8404: SCI4_NS.TDR, 0x5035_8400: SCI4_NS.RDR

- 0x5035_8504: SCI5_NS.TDR, 0x5035_8500: SCI5_NS.RDR
- 0x5035_8604: SCI6_NS.TDR, 0x5035_8600: SCI6_NS.RDR
- 0x5035_8704: SCI7_NS.TDR, 0x5035_8700: SCI7_NS.RDR
- 0x5035_8804: SCI8_NS.TDR, 0x5035_8800: SCI8_NS.RDR
- 0x5035_8904: SCI9_NS.TDR, 0x5035_8900: SCI9_NS.RDR

48.3 Operation

48.3.1 Basic Operation

The CRC calculator generates CRC codes for use in LSB-first or MSB-first transfer.

The following examples show CRC code generation for input data (0xF0) using the 16-bit CRC-CCITT generating polynomial ($X^{16} + X^{12} + X^5 + 1$). In these examples, the value of the CRC Data Output Register (CRCDOR_HA) is cleared before CRC calculation.

When an 8-bit CRC (with the polynomial $X^8 + X^2 + X + 1$) is in use, the valid bits of the CRC code are obtained in CRCDOR_BY. When a 32-bit CRC is in use, the valid bits of the CRC code are obtained in CRCDOR.

Figure 48.2 and Figure 48.3 show the LSB-first and MSB-first data transmission examples respectively. Figure 48.4 and Figure 48.5 show the LSB-first and MSB-first data reception examples.

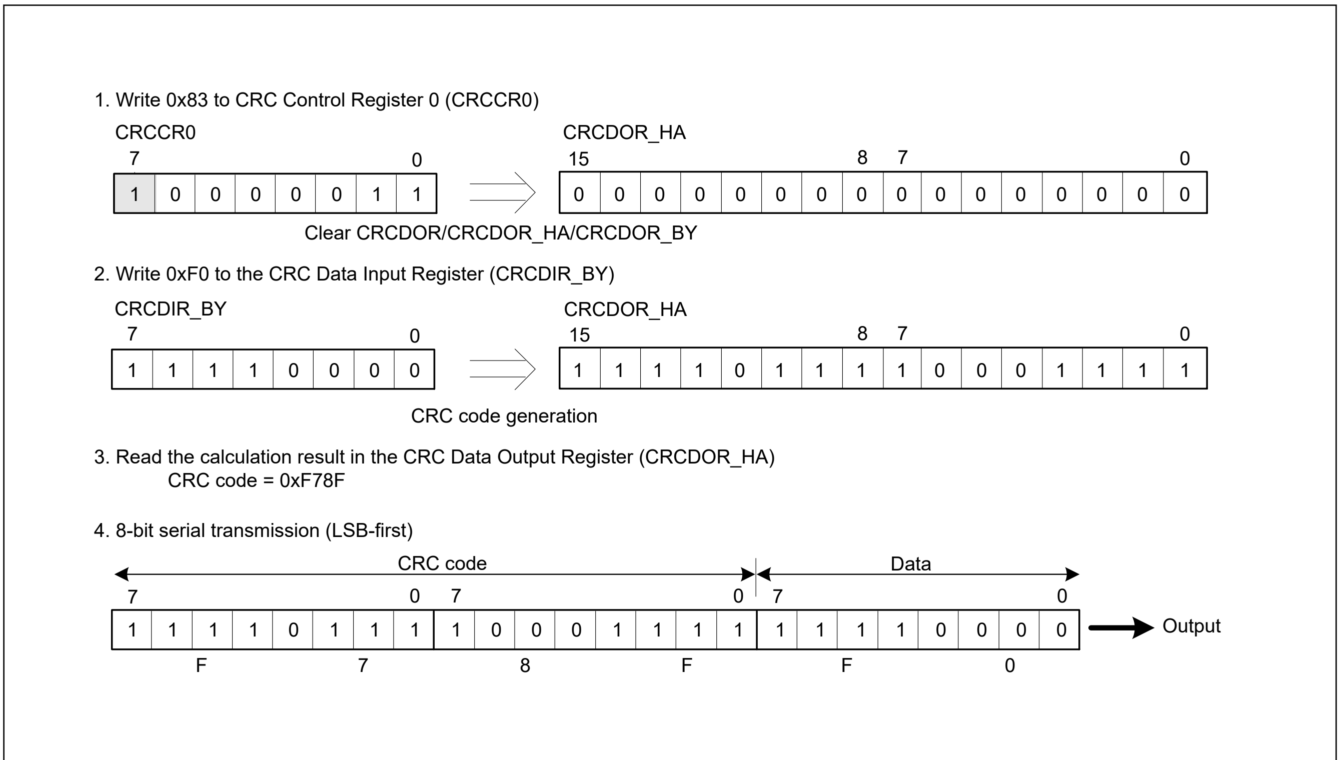


Figure 48.2 LSB-first data transmission

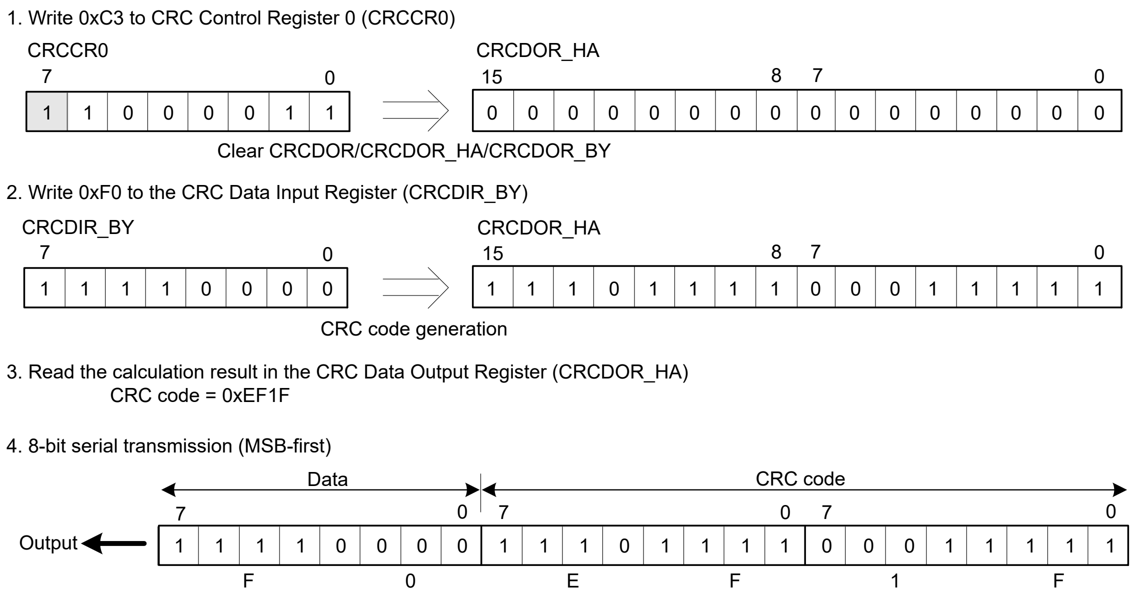


Figure 48.3 MSB-first data transmission

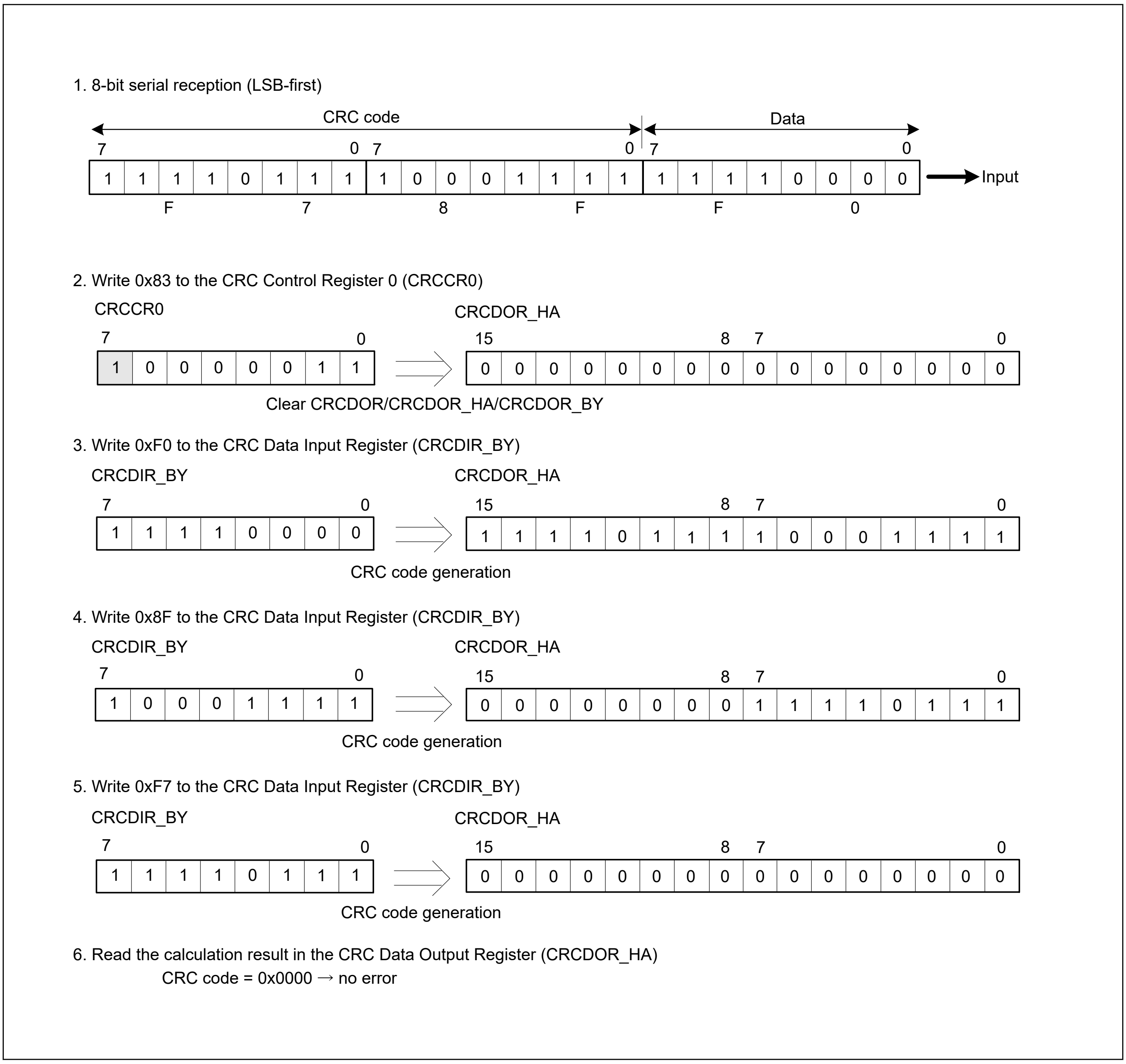


Figure 48.4 LSB-first data reception

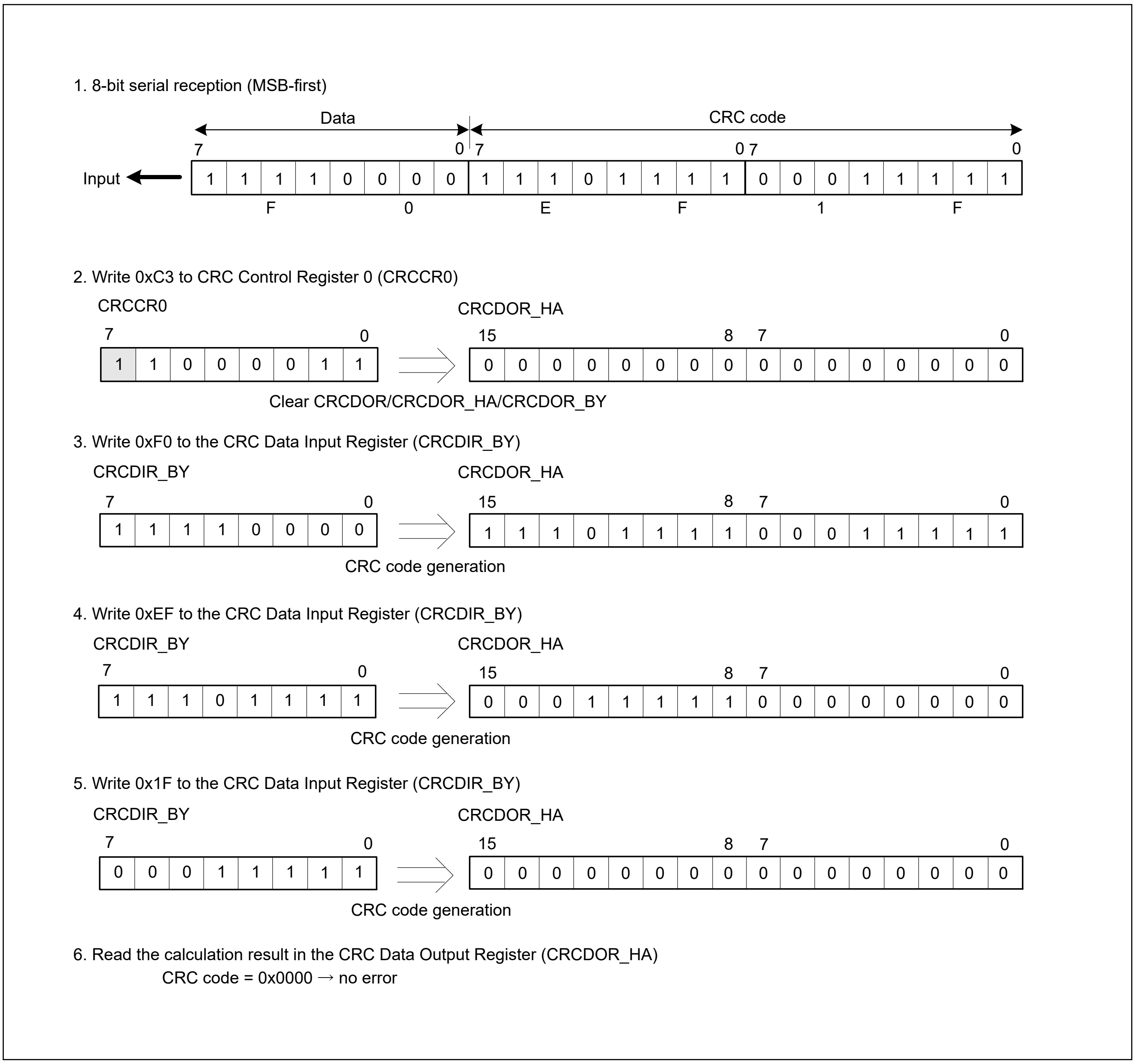


Figure 48.5 MSB-first data reception

48.3.2 CRC Snoop Function

The CRC snoop function monitors reads from and writes to a specific register and performs CRC calculation on the monitored data automatically. Because the CRC snoop function recognizes writes to and reads from a specific register address as a trigger to automatically perform CRC calculation, there is no need to write data to the CRCDIR register. All I/O registers specified in the section 48.2.5. CRCSAR : Snoop Address Register are subject to the CRC snoop. The CRC snoop is useful in monitoring writes to the SCIn.TDR (n = 0 to 9) register, and reads from the SCIn.RDR (n = 0 to 9) register.

To use this function, write the lower address 14 bits of a specific register to bits CRCSA[13] to CRCSA[0] in the CRCSAR register, and set CRCSEN bit in the CRCCR1 register to 1. Then, set the CRCSWR bit in the CRCCR1 register to 1 to enable snooping on writes to the target register, or set the CRCSWR bit in the CRCCR1 register to 0 to enable snooping on reads from the target register. It is possible that access to a target I/O register may be executed before the CRCSWR bit write completed. In this case, the data is not stored in the CRCDIR register. To avoid this issue, before accessing I/O register, read back the CRCSWR bit that was written to confirm that the write completed.

When both the CRCSEN and CRCSWR bits are set to 1, and data is written to a target register in a bus master module such as the CPU, DMAC, and DTC, the CRC calculator stores the data in the CRCDIR register and performs CRC calculation. Similarly, when the CRCSEN bit is set to 1, CRCSWR bit to 0, and data is read from a target register in a bus master module such as the CPU, DMAC, and DTC, the CRC calculator stores the data in the CRCDIR register and performs CRC calculations.

When the CRC code is generated by using CRC-8, CRC-16, and CRC-CCITT generating polynomial, the target register is accessed in 1 byte (8 bits). RDR_BY and TDRLL should be used to access RDR and TDR. Similarly, when the CRC code is generated by using CRC-32 and CRC-32C generating polynomial, the target register is accessed in 1 word (32 bits). Note that for RDR and TDR, CRC codes are generated that contain data other than RDAT and TDAT.

When CRC is marked as secure by PSARC.PSARC1 bit, the CRC snoop function is available for secure access to the specified I/O registers. When CRC is marked as non-secure by PSARC.PSARC1 bit, the CRC snoop function is available for non-secure access to the specified I/O registers.

48.4 Usage Notes

48.4.1 Settings for the Module-Stop State

The Module Stop Control Register C (MSTPCRC) can enable or disable CRC calculator operation. The CRC calculator is initially stopped after a reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Mode.

48.4.2 Note on Transmission

The transmission sequence for the CRC code differs based on whether the transmission is LSB-first or MSB-first. Figure 48.6 shows an LSB-first and MSB-first data transmission.

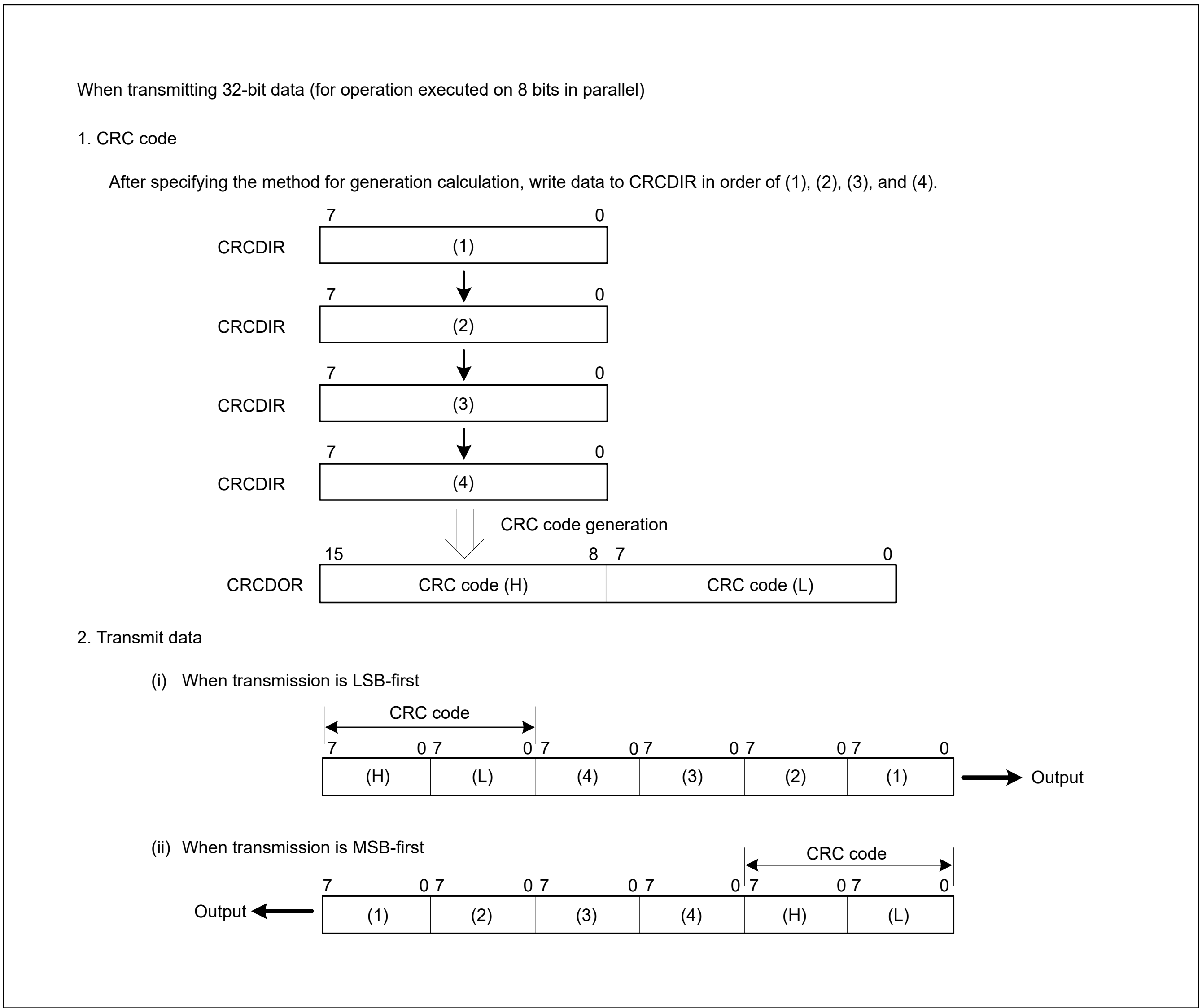


Figure 48.6 LSB-first and MSB-first data transmission

49. Pulse Density Modulation Interface (PDM-IF)

49.1 Overview

The Pulse Density Modulation Interface (PDM-IF) has maximum three channels that are connectable with external microphone which outputs the pulse density modulated signal. PDM-IF is connectable with up to three external microphones. PDM-IF can filter and convert 1-bit digital data streams that were pulse density modulated at a high sampling rate into 20-bit or 16-bit digital data at a lower sampling rate.

In this section, PCLK refers to PCLKB.

Table 49.1 lists the specifications of PDM-IF. Figure 49.1 shows the block diagrams of PDM-IF.

Table 49.1 PDM-IF specification

Item	Description
Number of channels	3 channels

Functions	• Capable of filtering 1-bit digital input data PDM_DATAn (n = 0 to 2) and converting them into 20-bit or 16-bit digital data. The bit order is little-endian.
	• IP supports stereo microphone (L/R sampling by rising/falling clock edge).
	• IP supports sound activity detector.
	• Each channel of IP includes programmable filters: 4th order sinc filter, high-pass filter (for suppression of DC bias), compensation filter (for sinc passband distortion), half-band decimation filter (for aliasing distortion).
	  – IP supports programmable and flexible decimation ratios.
	  – The sinc filter is selectable as first-, second-, third-, fourth-order.
	• IP supports DMA operation through APB.
	• Each channel of IP has an internal buffer.
	  – Buffer size as 32 stages at design time
	  – Capable of storing voice data during low power mode
	• Error detection functions can be used for debugging.

Interrupt sources	Maximum 7 sources 1. Data reception interrupt per channel (Maximum 3 interrupts) 2. Sound detection interrupt (shared between channels) 3. Error detection interrupt per channel (Maximum 3 interrupts)
Low power consumption function	• IP supports microphone low power configuration (slower clock). IP can switch to higher clock speed after sound detection recognized. • MCU can stop bus clock during low power mode.
Bus interface	• IP has an APB interface which is confirmed to APB4.
TrustZone filter	• Security and Privilege attribution can be set.

Each channel of PDM-IF consists of following blocks:

- Clock control: It generates PDMCLKn to microphone, and internal timing signal to sinc filter.
- Data input control: It receives PDMDATAn from microphone.
- Sinc filter: It converts the 1-bit digital data stream input from PDMDATAn to 34-bit signed data. And then the 34-bit signed data is clipped to 20-bit signed data.
- High-pass filter: It is a filter for suppression of DC bias.
- Compensation filter: It is a filter for sinc passband distortion.
- Low-pass (half-band decimation) filter: It is a filter for aliasing distortion.
- Moving average filter: It is a simple filter for noise reduction before sound detection.
- Sound detection: It generates the wakeup interrupt to recover from low power mode.
- Data buffer: It stores 20-bit or 16-bit data filtered by some filters.
- Short circuit detection: It detects the short circuit error which indicates consecutive 0’s or 1’s input from PDMDATAn.
- Overvoltage detection: It detects the overvoltage error.
- Error control: It generates error flags from other blocks.

The 1-bit digital data stream input from PDMDATAn pin (n = 0 to 2) is converted to 34-bit signed data by the sinc filter and then the data is clipped to 20-bit signed data. The 20-bit signed data is used for sound detection and data storing in buffer.

Sound detection block receives 20-bit signed data through high-pass filter block and moving average filter block. The block compares received data to the programmed thresholds and generates a wakeup interrupt in case that the data is above the upper threshold or below the lower threshold.

Data buffer block receives 20-bit signed data through high-pass filter, compensation filter, and low-pass filter. The block stores the 20-bit signed data or the clipped 16-bit signed data. And then the block sends the 20-bit signed data or 16-bit signed data to APB-IF once APB read access occurs.

Table 49.2 PDM I/O pins (n = 0 to 2)

Function	Pin name	I/O	Description
PDM-IF	PDMCLKn	Output	Output pins for clock
	PDMDATAn	Input	Input pins for data

Figure 49.1 shows the main components of PDM-IF.

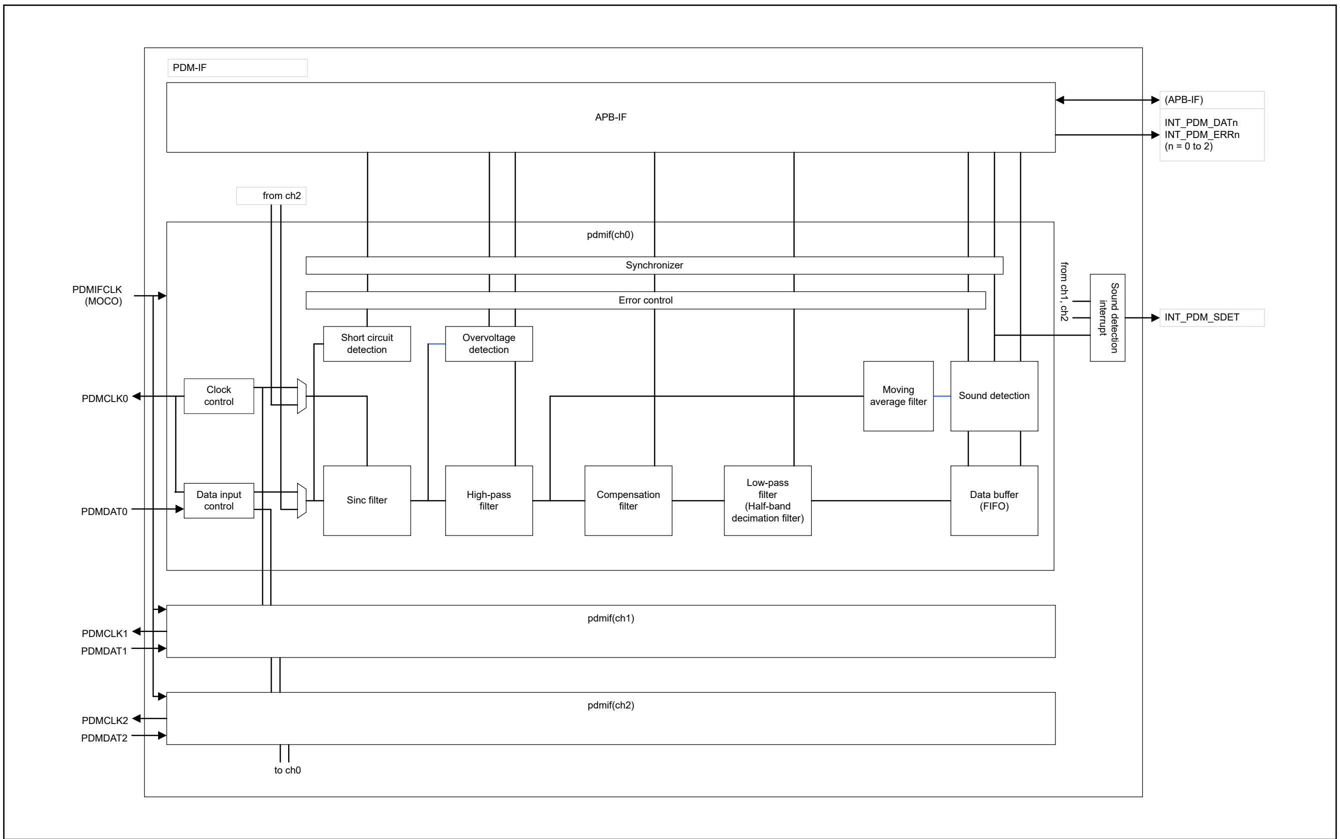


Figure 49.1 PDM-IF block diagram

49.2 Register Description

49.2.1 Register Map

Table 49.3 Register map (1 of 5)

Base address symbol	Offset address	Symbol	Register name	R/W	Initial value	Writable period*1				
						(1) Start flow	(2) Stop flow	(3) Normal processing flow	(4) Low power mode transition flow	(5) Normal mode transition flow
PDMIF	0x000	PDCSTRTR	Channel Software Start Trigger Register	W	0x00000000	✔	—	—	—	—
PDMIF	0x004	PDCSTPTR	Channel Software Stop Trigger Register	W	0x00000000	—	✔	—	—	—
PDMIF	0x008	PDCCHGTR	Channel Software Change Trigger Register	W	0x00000000	—	—	—	✔	✔
PDMIF	0x00C	PDCICR	Channel Interrupt Control Register	R/W	0x00000000	✔	✔	—	✔	✔
PDMIF	0x010	PDCSR	Channel Status Register	R	0x00000000	—	—	—	—	—
PDMIF	0x014	PDCSCR	Channel Status Clear Register	W	0x00000000	✔	✔	—	✔	✔
PDMIF	0x020	PDCSDCR	Channel Sound Detection Control Register	R/W	0x00000000	✔	✔	—	✔	✔
PDMIF	0x024	PDCDRCR	Channel Data Read Control Register	R/W	0x00000000	—	—	✔	—	—
PDMIF	0x028	PDCDCR	Channel Data Clear Register	W	0x00000000	—	—	✔*2	—	—
PDMIF	0x080	PDVR	Version Register	R	0x00000010	—	—	—	—	—
PDMIF	0x100 + 0x100 x n	PDSTRTRCHn	Software Start Trigger Register Channel n (n = 0 to 2)	W	0x00000000	✔	—	—	—	—
PDMIF	0x104 + 0x100 x n	PDSTPTRCHn	Software Stop Trigger Register Channel n (n = 0 to 2)	W	0x00000000	—	✔	—	—	—
PDMIF	0x108 + 0x100 x n	PDCHGTRCHn	Software Change Trigger Register Channel n (n = 0 to 2)	W	0x00000000	—	—	—	✔	✔
PDMIF	0x10C + 0x100 x n	PDICRCHn	Interrupt Control Register Channel n (n = 0 to 2)	R/W	0x00000000	✔	✔	—	✔	✔
PDMIF	0x110 + 0x100 x n	PDSDCRCHn	Status Detection Control Register Channel n (n = 0 to 2)	R/W	0x00000000	✔	✔	—	—	—
PDMIF	0x114 + 0x100 x n	PDSRCHn	Status Register Channel n (n = 0 to 2)	R	0x00000000	—	—	—	—	—

Table 49.3 Register map (2 of 5)

Base address symbol	Offset address	Symbol	Register name	R/W	Initial value	Writable period*1				
						(1) Start flow	(2) Stop flow	(3) Normal processing flow	(4) Low power mode transition flow	(5) Normal mode transition flow
PDMIF	0x118 + 0x100 x n	PDSCRCHn	Status Clear Register Channel n (n = 0 to 2)	W	0x00000000	✔	✔	—	✔	✔
PDMIF	0x120 + 0x100 x n	PDMDSRCHn	Mode Setting Register Channel n (n = 0 to 2)	R/W	0x00000000	✔	—	—	—	—
PDMIF	0x124 + 0x100 x n	PDSFCRCHn	Sinc filter Control Register Channel n (n = 0 to 2)	R/W	0x057C0000	✔	—	—	✔	✔
PDMIF	0x128 + 0x100 x n	PDHFCS0RCHn	High-pass filter Coefficient s(0) Register Channel n (n = 0 to 2)	R/W	0x00003F61	✔	—	—	—	—
PDMIF	0x12C + 0x100 x n	PDHFCK1RCHn	High-pass filter Coefficient k(1) Register Channel n (n = 0 to 2)	R/W	0x00003EC1	✔	—	—	—	—
PDMIF	0x130 + 0x100 x n	PDHFCH0RCHn	High-pass filter Coefficient h(0) Register Channel n (n = 0 to 2)	R/W	0x00004000	✔	—	—	—	—
PDMIF	0x134 + 0x100 x n	PDHFCH1RCHn	High-pass filter Coefficient h(1) Register Channel n (n = 0 to 2)	R/W	0x0000C000	✔	—	—	—	—
PDMIF	0x138 + 0x100 x n	PDCFCH00RCHn	Compensation filter Coefficient h(0) Register Channel n (n = 0 to 2)	R/W	0x00001FE8	✔	—	—	—	—
PDMIF	0x13C + 0x100 x n	PDCFCH01RCHn	Compensation filter Coefficient h(1) Register Channel n (n = 0 to 2)	R/W	0x00000039	✔	—	—	—	—
PDMIF	0x140 + 0x100 x n	PDCFCH02RCHn	Compensation filter Coefficient h(2) Register Channel n (n = 0 to 2)	R/W	0x0000003C	✔	—	—	—	—
PDMIF	0x144 + 0x100 x n	PDCFCH03RCHn	Compensation filter Coefficient h(3) Register Channel n (n = 0 to 2)	R/W	0x00001E56	✔	—	—	—	—
PDMIF	0x148 + 0x100 x n	PDCFCH04RCHn	Compensation filter Coefficient h(4) Register Channel n (n = 0 to 2)	R/W	0x000001DC	✔	—	—	—	—
PDMIF	0x14C + 0x100 x n	PDCFCH05RCHn	Compensation filter Coefficient h(5) Register Channel n (n = 0 to 2)	R/W	0x000006E1	✔	—	—	—	—
PDMIF	0x150 + 0x100 x n	PDCFCH06RCHn	Compensation filter Coefficient h(6) Register Channel n (n = 0 to 2)	R/W	0x000001DC	✔	—	—	—	—

Table 49.3 Register map (3 of 5)

Base address symbol	Offset address	Symbol	Register name	R/W	Initial value	Writable period*1				
						(1) Start flow	(2) Stop flow	(3) Normal processing flow	(4) Low power mode transition flow	(5) Normal mode transition flow
PDMIF	0x154 + 0x100 x n	PDCFCH07RCHn	Compensation filter Coefficient h(7) Register Channel n (n = 0 to 2)	R/W	0x00001E56	✔	—	—	—	—
PDMIF	0x158 + 0x100 x n	PDCFCH08RCHn	Compensation filter Coefficient h(8) Register Channel n (n = 0 to 2)	R/W	0x0000003C	✔	—	—	—	—
PDMIF	0x15C + 0x100 x n	PDCFCH09RCHn	Compensation filter Coefficient h(9) Register Channel n (n = 0 to 2)	R/W	0x00000039	✔	—	—	—	—
PDMIF	0x160 + 0x100 x n	PDCFCH10RCHn	Compensation filter Coefficient h(10) Register Channel n (n = 0 to 2)	R/W	0x00001FE8	✔	—	—	—	—
PDMIF	0x164 + 0x100 x n	PDLFCH010RCHn	Low-pass filter Coefficient h0(10) Register Channel n (n = 0 to 2)	R/W	0x00000400	✔	—	—	—	—
PDMIF	0x168 + 0x100 x n	PDLFCH100RCHn	Low-pass filter Coefficient h1(0) Register Channel n (n = 0 to 2)	R/W	0x00001FF8	✔	—	—	—	—
PDMIF	0x16C + 0x100 x n	PDLFCH101RCHn	Low-pass filter Coefficient h1(1) Register Channel n (n = 0 to 2)	R/W	0x0000000A	✔	—	—	—	—
PDMIF	0x170 + 0x100 x n	PDLFCH102RCHn	Low-pass filter Coefficient h1(2) Register Channel n (n = 0 to 2)	R/W	0x00001FF0	✔	—	—	—	—
PDMIF	0x174 + 0x100 x n	PDLFCH103RCHn	Low-pass filter Coefficient h1(3) Register Channel n (n = 0 to 2)	R/W	0x00000018	✔	—	—	—	—
PDMIF	0x178 + 0x100 x n	PDLFCH104RCHn	Low-pass filter Coefficient h1(4) Register Channel n (n = 0 to 2)	R/W	0x00001FDC	✔	—	—	—	—
PDMIF	0x17C + 0x100 x n	PDLFCH105RCHn	Low-pass filter Coefficient h1(5) Register Channel n (n = 0 to 2)	R/W	0x00000034	✔	—	—	—	—
PDMIF	0x180 + 0x100 x n	PDLFCH106RCHn	Low-pass filter Coefficient h1(6) Register Channel n (n = 0 to 2)	R/W	0x00001FB3	✔	—	—	—	—
PDMIF	0x184 + 0x100 x n	PDLFCH107RCHn	Low-pass filter Coefficient h1(7) Register Channel n (n = 0 to 2)	R/W	0x00000076	✔	—	—	—	—

Table 49.3 Register map (4 of 5)

Base address symbol	Offset address	Symbol	Register name	R/W	Initial value	Writable period[*1]				
						(1) Start flow	(2) Stop flow	(3) Normal processing flow	(4) Low power mode transition flow	(5) Normal mode transition flow
PDMIF	0x188 + 0x100 x n	PDLFCH108RCHn	Low-pass filter Coefficient h1(8) Register Channel n (n = 0 to 2)	R/W	0x00001F2E	✔	—	—	—	—
PDMIF	0x18C + 0x100 x n	PDLFCH109RCHn	Low-pass filter Coefficient h1(9) Register Channel n (n = 0 to 2)	R/W	0x00000289	✔	—	—	—	—
PDMIF	0x190 + 0x100 x n	PDLFCH110RCHn	Low-pass filter Coefficient h1(10) Register Channel n (n = 0 to 2)	R/W	0x00000289	✔	—	—	—	—
PDMIF	0x194 + 0x100 x n	PDLFCH111RCHn	Low-pass filter Coefficient h1(11) Register Channel n (n = 0 to 2)	R/W	0x00001F2E	✔	—	—	—	—
PDMIF	0x198 + 0x100 x n	PDLFCH112RCHn	Low-pass filter Coefficient h1(12) Register Channel n (n = 0 to 2)	R/W	0x00000076	✔	—	—	—	—
PDMIF	0x19C + 0x100 x n	PDLFCH113RCHn	Low-pass filter Coefficient h1(13) Register Channel n (n = 0 to 2)	R/W	0x00001FB3	✔	—	—	—	—
PDMIF	0x1A0 + 0x100 x n	PDLFCH114RCHn	Low-pass filter Coefficient h1(14) Register Channel n (n = 0 to 2)	R/W	0x00000034	✔	—	—	—	—
PDMIF	0x1A4 + 0x100 x n	PDLFCH115RCHn	Low-pass filter Coefficient h1(15) Register Channel n (n = 0 to 2)	R/W	0x00001FDC	✔	—	—	—	—
PDMIF	0x1A8 + 0x100 x n	PDLFCH116RCHn	Low-pass filter Coefficient h1(16) Register Channel n (n = 0 to 2)	R/W	0x00000018	✔	—	—	—	—
PDMIF	0x1AC + 0x100 x n	PDLFCH117RCHn	Low-pass filter Coefficient h1(17) Register Channel n (n = 0 to 2)	R/W	0x00001FF0	✔	—	—	—	—
PDMIF	0x1B0 + 0x100 x n	PDLFCH118RCHn	Low-pass filter Coefficient h1(18) Register Channel n (n = 0 to 2)	R/W	0x0000000A	✔	—	—	—	—
PDMIF	0x1B4 + 0x100 x n	PDLFCH119RCHn	Low-pass filter Coefficient h1(19) Register Channel n (n = 0 to 2)	R/W	0x00001FF8	✔	—	—	—	—
PDMIF	0x1B8 + 0x100 x n	PDSDLTRCHn	Sound Detection Lower Threshold Register Channel n (n = 0 to 2)	R/W	0x00000000	✔	—	—	✔	—

Table 49.3 Register map (5 of 5)

Base address symbol	Offset address	Symbol	Register name	R/W	Initial value	Writable period*1				
						(1) Start flow	(2) Stop flow	(3) Normal processing flow	(4) Low power mode transition flow	(5) Normal mode transition flow
PDMIF	0x1BC + 0x100 x n	PDSDUTRCHn	Sound Detection Upper Threshold Register Channel n (n = 0 to 2)	R/W	0x00000000	✔	—	—	✔	—
PDMIF	0x1C0 + 0x100 x n	PDDBCRCHn	Data Buffer Control Register Channel n (n = 0 to 2)	R/W	0x00000000	✔	—	—	—	—
PDMIF	0x1C4 + 0x100 x n	PDSCTSRCHn	Short Circuit Threshold Setting Register Channel n (n = 0 to 2)	R/W	0x00000000	✔	—	—	—	—
PDMIF	0x1C8 + 0x100 x n	PDOVLTRCHn	Overvoltage Lower Threshold Register Channel n (n = 0 to 2)	R/W	0x00000000	✔	—	—	—	—
PDMIF	0x1CC + 0x100 x n	PDOVUTRCHn	Overvoltage Upper Threshold Register Channel n (n = 0 to 2)	R/W	0x00000000	✔	—	—	—	—
PDMIF	0x1E0 + 0x100 x n	PDDRCRCHn	Data Read Control Register Channel n (n = 0 to 2)	R/W	0x00000000	—	—	✔	—	—
PDMIF	0x1E4 + 0x100 x n	PDDCRCHn	Data Clear Register Channel n (n = 0 to 2)	W	0x00000000	—	—	✔*2	—	—
PDMIF	0x1E8 + 0x100 x n	PDDRRCHn	Data Read Register Channel n (n = 0 to 2)	R	0x00000000	—	—	—	—	—
PDMIF	0x1EC + 0x100 x n	PDDSRCHn	Data Status Register Channel n (n = 0 to 2)	R	0x00000000	—	—	—	—	—

Note: Some registers are mirroring together between base address symbol PDM_CTL and PDMn registers. See section 49.2.67. Mirroring Register.
Note 1. See section 49.4. Setting Flow.
Note 2. The register is not shown in the Setting flow, but it is allowed to use in the flow.

49.2.2 PDCSTRTR : Channel Software Start Trigger Register

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	STRTRG2	STRTRG1	STRTRG0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STRTRG0	Channel 0 Start Trigger 0: Do nothing 1: Start channel 0	W
1	STRTRG1	Channel 1 Start Trigger 0: Do nothing 1: Start channel 1	W
2	STRTRG2	Channel 2 Start Trigger 0: Do nothing 1: Start channel 2	W
31:3	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

49.2.3 PDCSTPTR : Channel Software Stop Trigger Register

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	STPTRG2	STPTRG1	STPTRG0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STPTRG0	Channel 0 Stop Trigger 0: Do nothing 1: Stop channel 0	W
1	STPTRG1	Channel 1 Stop Trigger 0: Do nothing 1: Stop channel 1	W
2	STPTRG2	Channel 2 Stop Trigger 0: Do nothing 1: Stop channel 2	W
31:3	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

49.2.4 PDCCHGTR : Channel Software Change Trigger Register

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	CHGTRG2	CHGTRG1	CHGTRG0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CHGTRG0	Channel 0 Change Trigger 0: Do nothing 1: Change clock (PDM_CLK0) setting	W
1	CHGTRG1	Channel 1 Change Trigger 0: Do nothing 1: Change clock (PDM_CLK1) setting	W
2	CHGTRG2	Channel 2 Change Trigger 0: Do nothing 1: Change clock (PDM_CLK2) setting	W
31:3	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

49.2.5 PDCICR : Channel Interrupt Control Register

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	IEDE2	IEDE1	IEDE0	—	—	—	—	—	IDRE2	IDRE1	IDRE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	ISDE2	ISDE1	ISDE0	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	ISDE0	Channel 0 Sound Detection Interrupt Enable Bit 0: Do not allow PDM_SDET interrupt to be issued when the sound for channel 0 is detected 1: Allow PDM_SDET interrupt to be issued when the sound for channel 0 is detected	R/W
9	ISDE1	Channel 1 Sound Detection Interrupt Enable Bit 0: Do not allow PDM_SDET interrupt to be issued when the sound for channel 1 is detected 1: Allow PDM_SDET interrupt to be issued when the sound for channel 1 is detected	R/W
10	ISDE2	Channel 2 Sound Detection Interrupt Enable Bit 0: Do not allow PDM_SDET interrupt to be issued when the sound for channel 2 is detected 1: Allow PDM_SDET interrupt to be issued when the sound for channel 2 is detected	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
16	IDRE0	Channel 0 Data Reception Interrupt Enable Bit 0: Do not allow PDM_DAT0 interrupt to be issued 1: Allow PDM_DAT0 interrupt to be issued	R/W
17	IDRE1	Channel 1 Data Reception Interrupt Enable Bit 0: Do not allow PDM_DAT1 interrupt to be issued 1: Allow PDM_DAT1 interrupt to be issued	R/W
18	IDRE2	Channel 2 Data Reception Interrupt Enable Bit 0: Do not allow PDM_DAT2 interrupt to be issued 1: Allow PDM_DAT2 interrupt to be issued	R/W
23:19	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
24	IEDE0	Channel 0 Error Detection Interrupt Enable Bit 0: Do not allow PDM_ERR0 interrupt to be issued 1: Allow PDM_ERR0 interrupt to be issued	R/W
25	IEDE1	Channel 1 Error Detection Interrupt Enable Bit 0: Do not allow PDM_ERR1 interrupt to be issued 1: Allow PDM_ERR1 interrupt to be issued	R/W
26	IEDE2	Channel 2 Error Detection Interrupt Enable Bit 0: Do not allow PDM_ERR2 interrupt to be issued 1: Allow PDM_ERR2 interrupt to be issued	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

49.2.6 PDCSR : Channel Status Register

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	EDF2	EDF1	EDF0	—	—	—	—	—	DRF2	DRF1	DRF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SDF2	SDF1	SDF0	—	—	—	—	—	STATE 2	STATE 1	STATE 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STATE0	Channel 0 State 0: Channel 0 stop 1: Channel 0 in operation	R
1	STATE1	Channel 1 State 0: Channel 1 stop 1: Channel 1 in operation	R
2	STATE2	Channel 2 State 0: Channel 2 stop 1: Channel 2 in operation	R
7:3	—	These bits are read as 0.	R
8	SDF0	Channel 0 Sound Detection Flag 0: Indicates that the sound which exceeded the threshold is not detected 1: Indicates that the sound which exceeded the threshold is detected	R
9	SDF1	Channel 1 Sound Detection Flag 0: Indicates that the sound which exceeded the threshold is not detected 1: Indicates that the sound which exceeded the threshold is detected	R
10	SDF2	Channel 2 Sound Detection Flag 0: Indicates that the sound which exceeded the threshold is not detected 1: Indicates that the sound which exceeded the threshold is detected	R
15:11	—	These bits are read as 0.	R
16	DRF0	Channel 0 Data Reception Flag 0: Indicates that the number of data stored in buffer does not exceed the threshold 1: Indicates that the number of data stored in buffer exceeded the threshold	R
17	DRF1	Channel 1 Data Reception Flag 0: Indicates that the number of data stored in buffer does not exceed the threshold 1: Indicates that the number of data stored in buffer exceeded the threshold	R

Bit	Symbol	Function	R/W
18	DRF2	Channel 2 Data Reception Flag 0: Indicates that the number of data stored in buffer does not exceed the threshold 1: Indicates that the number of data stored in buffer exceeded the threshold	R
23:19	—	These bits are read as 0.	R
24	EDF0	Channel 0 Error Detection Flag 0: Indicates that errors are not detected. 1: Indicates that errors are detected. See section 49.2.17. PDSRCHn : Status Register Channel n (n = 0 to 2) for details.	R
25	EDF1	Channel 1 Error Detection Flag 0: Indicates that errors are not detected. 1: Indicates that errors are detected. See section 49.2.17. PDSRCHn : Status Register Channel n (n = 0 to 2) for details.	R
26	EDF2	Channel 2 Error Detection Flag 0: Indicates that errors are not detected. 1: Indicates that errors are detected. See section 49.2.17. PDSRCHn : Status Register Channel n (n = 0 to 2) for details.	R
31:27	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

STATEn: See section 49.3.2. Channel Start and Channel Stop for details.

[Condition to become 0]

- When 1 is written to PDCSTPTR.STPTRGn.

[Condition to become 1]

- When 1 is written to PDCSTRTR.STRTRGn.

SDFn: See section 49.3.6. Sound Detection for details.

[Condition to become 0]

- When writing 1 to a flag clear register bit (PDCSCR.SDFCn).

[Condition to become 1]

- When PDCSDCR.SDEn = 1 and a moving average filter result is higher than or equal to PDSDUTR.SDETU[19:0].
- When PDCSDCR.SDEn = 1 and a moving average filter result is lower than or equal to PDSDLTR.SDETL[19:0].

DRFn: See section 49.3.7. Data Buffer for details.

[Condition to become 0]

- When PDCDRCR.DATREn = 0 or PDDSR.DATNUM[7:0] is lower than a threshold.

[Condition to become 1]

- When PDCDRCR.DATREn = 1 and PDDSR.DATNUM[7:0] is higher than or equal to a threshold which is configured by PDDBCR.DATRITHR[2:0].

EDFn

[Condition to become 0]

- When all error flags are cleared by writing to section 49.2.18. PDSCRCHn : Status Clear Register Channel n (n = 0 to 2), this flag is also cleared.

[Condition to become 1]

- When any error occurs. See section 49.2.17. PDSRCHn : Status Register Channel n (n = 0 to 2) register for the kind of error.

49.2.7 PDCSCR : Channel Status Clear Register

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SDFC2	SDFC1	SDFC0	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	The write value should be 0.	W
8	SDFC0	Channel 0 Sound Detection Flag Clear 0: Do nothing 1: Clear PDCSR.SDF0	W
9	SDFC1	Channel 1 Sound Detection Flag Clear 0: Do nothing 1: Clear PDCSR.SDF1	W
10	SDFC2	Channel 2 Sound Detection Flag Clear 0: Do nothing 1: Clear PDCSR.SDF2	W
31:11	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

49.2.8 PDCSDCR : Channel Sound Detection Control Register

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SDE2	SDE1	SDE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SDE0	Channel 0 Sound Detection Enable Bit 0: Do not allow sound detection 1: Allow sound detection	R/W
1	SDE1	Channel 1 Sound Detection Enable Bit 0: Do not allow sound detection 1: Allow sound detection	R/W
2	SDE2	Channel 2 Sound Detection Enable Bit 0: Do not allow sound detection 1: Allow sound detection	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

49.2.9 PDCDRCR : Channel Data Read Control Register

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x24

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	DATRE2	DATRE1	DATRE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DATRE0	Channel 0 Data Read Enable Bit 0: Do not allow reading data from data buffer 1: Allow reading data from data buffer	R/W
1	DATRE1	Channel 1 Data Read Enable Bit 0: Do not allow reading data from data buffer 1: Allow reading data from data buffer	R/W
2	DATRE2	Channel 2 Data Read Enable Bit 0: Do not allow reading data from data buffer 1: Allow reading data from data buffer	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

49.2.10 PDCDCR : Channel Data Clear Register

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x28

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	DATC2	DATC1	DATC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DATC0	Channel 0 Data Clear 0: Do nothing 1: Clear data	W
1	DATC1	Channel 1 Data Clear 0: Do nothing 1: Clear data	W
2	DATC2	Channel 2 Data Clear 0: Do nothing 1: Clear data	W

Bit	Symbol	Function	R/W
31:3	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

49.2.11 PDVR : Version Register

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x80

Bit position:	31																				11											0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VER[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
11:0	VER[11:0]	PDM-IF version is shown.	R
31:12	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

49.2.12 PDSTRTRCHn : Software Start Trigger Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x100 + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STRTRG
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STRTRG	Start Trigger 0: Do nothing 1: Start the channel	W
31:1	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

49.2.13 PDSTPTRCHn : Software Stop Trigger Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x104 + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STPTRG
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STPTRG	Stop Trigger 0: Do nothing 1: Stop the channel	W
31:1	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

49.2.14 PDCHGTRCHn : Software Change Trigger Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x108 + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CHGTRG
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CHGTRG	Change Trigger 0: Do nothing 1: Change settings of PDM_CLKn and sinc filter	W
31:1	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

49.2.15 PDICRCHn : Interrupt Control Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x10C + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IEDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	IDRE	ISDE	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	ISDE	Sound Detection Interrupt Enable Bit 0: Do not allow PDM_SDET interrupt to be issued 1: Allow PDM_SDET interrupt to be issued	R/W
2	IDRE	Data Reception Interrupt Enable Bit 0: Do not allow PDM_DATn interrupt to be issued 1: Allow PDM_DATn interrupt to be issued	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W
16	IEDE	Error Detection Interrupt Enable Bit 0: Do not allow PDM_ERRn interrupt to be issued 1: Allow PDM_ERRn interrupt to be issued	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

49.2.16 PDSDCRCHn : Status Detection Control Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x110 + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	BFOWDE	—	—	—	—	—	—	—	—	OVUDE	OVLDE	SCDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SDE	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	SDE	Sound Detection Enable Bit 0: Do not allow sound detection 1: Allow sound detection	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
16	SCDE	Short Circuit Detection Enable Bit 0: Do not allow detection of the PDM_DATAn pin short circuit 1: Allow detection of the PDM_DATAn pin short circuit	R/W

Bit	Symbol	Function	R/W
17	OVLDE	Overvoltage Lower Limit Exceeded Detection Enable Bit 0: Do not allow detection of data fallen lower limit 1: Allow detection of data fallen lower limit	R/W
18	OVUDE	Overvoltage Upper Limit Exceeded Detection Enable Bit 0: Do not allow detection of data exceeded upper limit 1: Allow detection of data exceeded upper limit	R/W
26:19	—	These bits are read as 0. The write value should be 0.	R/W
27	BFOWDE	Buffer Overwriting Detection Enable Bit 0: Do not allow detection of buffer overwriting 1: Allow detection of buffer overwriting	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

49.2.17 PDSRCHn : Status Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x114 + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	BFOWDF	—	—	—	—	—	—	—	—	OVUDF	OVLDF	SCDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	DRF	SDF	STATE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STATE	State 0: Channel stop 1: Channel in operation	R
1	SDF	Sound Detection Flag 0: Indicates that the sound which exceeded the threshold is not detected 1: Indicates that the sound which exceeded the threshold is detected	R
2	DRF	Data Reception Flag 0: Indicates that the number of data stored in buffer is lower than a threshold 1: Indicates that the number of data stored in buffer is higher than or equal to a threshold	R
15:3	—	These bits are read as 0.	R
16	SCDF	Short circuit detection flag. 0: Indicates that the PDM_DATAn pin short circuit is not detected. 1: Indicates that a short circuit on the PDM_DATAn pin has been detected.	R
17	OVLDF	Overvoltage Lower Limit Exceeded Detection Flag 0: Indicates that the data is not lower than the lower limit 1: Indicates that the data has fallen the lower limit	R
18	OVUDF	Overvoltage Upper Limit Exceeded Detection Flag 0: Indicates that the data has not exceeded the upper limit 1: Indicates that the data has exceeded the upper limit	R
26:19	—	These bits are read as 0.	R
27	BFOWDF	Buffer Overwriting Detection Flag 0: Indicates that buffer overwriting does not occur 1: Indicates that buffer overwriting occurs	R
31:28	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

STATE: For details, see section 49.3.2. Channel Start and Channel Stop.

[Condition to become 0]

- When 1 is written to PDSTPTR.STPTRG.

[Condition to become 1]

- When 1 is written to PDSTRTR.STRTRG.

SDF: For details, see section 49.3.6. Sound Detection.

[Condition to become 0]

- When writing 1 to the flag clear register bit (PDSCR.SDFC).

[Condition to become 1]

- When PDSDCR.SDE = 1 and a moving average filter result is PDSDUTR.SDETU[19:0] or higher.
- When PDSDCR.SDE = 1 and a moving average filter result is PDSDLTR.SDETL[19:0] or lower.

DRF: For details, see section 49.3.7. Data Buffer.

[Condition to become 0]

- When PDDRCR.DATRE = 0 or PDDSR.DATNUM[7:0] is lower than a threshold.

[Condition to become 1]

- When PDDRCR.DATRE = 1 and PDDSR.DATNUM[7:0] is higher than or equal to a threshold which is configured by PDDBCR.DATRITHR[2:0].

SCDF: For details, see section 49.3.8. Short-circuit Detection.

[Condition to become 0]

- When writing 1 to the flag clear register bit (PDSCR.SCDFC).

[Condition to become 1]

- When PDSDCR.SCDE = 1 and 1 is continuously input to the PDM_DATAn pin, exceeding the number of times set in PDSCTSR.SCDH[12:0].
- When PDSDCR.SCDE = 1 and 0 is continuously input to the PDM_DATAn pin, exceeding the number of times set in PDSCTSR.SCDL[12:0].

OVLDF: For details, see section 49.3.9. Overvoltage Detection.

[Condition to become 0]

- When writing 1 to the flag clear register bit (PDSCR.OVLDFC).

[Condition to become 1]

- When PDSDCR.OVLDE = 1 and a clipped sinc filter result is smaller than a lower limit (PDOVLTR.OVDL[19:0]).

OVUDF: For details, see section 49.3.9. Overvoltage Detection.

[Condition to become 0]

- When writing 1 to the flag clear register bit (PDSCR.OVUDFC).

[Condition to become 1]

- When PDSDCR.OVUDE = 1 and a clipped sinc filter result is larger than an upper limit (PDOVUTR.OVDU[19:0]).

BFOWDF: For details, see section 49.3.7. Data Buffer.

[Condition to become 0]

- When writing 1 to the flag clear register bit (PDSCR.BFOWDFC).

[Condition to become 1]

- When PDSDCR.BFOWDE = 1 and data buffer is full and data is written to buffer without reading.

49.2.18 PDSCRCHn : Status Clear Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x118 + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	BFOWDFC	—	—	—	—	—	—	—	—	OVUDFC	OVLDFC	SCDFC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SDFC	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	The write value should be 0.	W
1	SDFC	Sound Detection Flag Clear 0: Do nothing 1: Clear PDSRCHn.SDF	W
15:2	—	The write value should be 0.	W
16	SCDFC	Short Circuit Detection Flag Clear 0: Do nothing 1: Clear PDSRCHn.SCDF	W
17	OVLDFC	Overvoltage Lower Limit Exceeded Detection Flag Clear 0: Do nothing 1: Clear PDSRCHn.OVLDF	W
18	OVUDFC	Overvoltage Upper Limit Exceeded Detection Flag Clear 0: Do nothing 1: Clear PDSRCHn.OVUDF	W
26:19	—	The write value should be 0.	W
27	BFOWDFC	Buffer Overwriting Detection Flag Clear 0: Do nothing 1: Clear PDSRCHn.BFOWDF	W
31:28	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

49.2.19 PDMDSRCHn : Mode Setting Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x120 + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DBIS[3:0]				—	—	SDMAMD[1:0]		—	—	—	—	—	—	LFIS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	CFIS[1:0]		—	—	HFIS[1:0]		—	SFMD[2:0]			—	—	—	INPSEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	INPSEL	Input Data Select 0: Rise-edge data of channel n. 1: Fall-edge data of channel n-1. When n = 0, fall-edge data of channel 2 is selected.	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
6:4	SFMD[2:0]	Sinc Filter Mode Setting 0 0 1: 1-order 0 1 0: 2-order 0 1 1: 3-order Others: 4-order (default)	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
9:8	HFIS[1:0]	High-pass Filter Input Shift Setting 0 0: No shift 0 1: 1-bit right shift 1 0: 2-bit right shift 1 1: 3-bit right shift	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
13:12	CFIS[1:0]	Compensation Filter Input Shift Setting 0 0: No shift 0 1: 1-bit right shift 1 0: 2-bit right shift 1 1: 3-bit right shift	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
17:16	LFIS[1:0]	Low-pass (half-band decimation) Filter Input Shift Setting 0 0: No shift 0 1: 1-bit right shift 1 0: 2-bit right shift 1 1: 3-bit right shift	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
25:24	SDMAMD[1:0]	Moving Average Mode for Sound Detection Data 0 0: 1-order (filter is skipped) (default) 0 1: 2-order 1 0: 4-order 1 1: User prohibition	R/W
27:26	—	These bits are read as 0. The write value should be 0.	R/W
31:28	DBIS[3:0]	Data Buffer Input Shift Setting S: signed bit 0x0: 20-bit mode, {1{S}, [18:0]} 0x1: 20-bit mode, {2{S}, [18:1]} 0x2: 20-bit mode, {3{S}, [18:2]} 0x3: 20-bit mode, {4{S}, [18:3]} 0x8: 16-bit mode, {S,D[18:4]} 0x9: 16-bit mode, {S,D[17:3]} 0xA: 16-bit mode, {S,D[16:2]} 0xB: 16-bit mode, {S,D[15:1]} 0xC: 16-bit mode, {S,D[14:0]} Others: Setting prohibited	R/W

Note: S-TYPE-3, P-TYPE-3

49.2.20 PDSFCRCHn : Sinc Filter Control Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x124 + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SINCRNG[4:0]					SINCDEC[7:0]								—	—	—	—	—	—	—	—	—	—	—	—	CKDIV[3:0]			
Value after reset:	0	0	0	0	0	1	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CKDIV[3:0]	PDM_CLKn Dividend Ratio to Core Clock 0x0: 2 divisions (default) 0x1: 4 divisions 0x2: 6 divisions ⋮ 0xD: 28 divisions 0xE: 30 divisions 0xF: 32 divisions	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W
23:16	SINCDEC[7:0]	Sinc Filter Decimation Ratio M indicates the decimation ratio. Setting less than M = 4 is prohibited. When the value is set, the sinc filter operation is not guaranteed. Set it to satisfy D × M > 12. If it is not satisfied, the operation of other filters is not guaranteed. (D: PDM_CLKn dividend ratio) Decimation ratio M is set as follows. M = SINCDEC + 1 0x00: M = 1 (prohibited) 0x01: M = 2 (prohibited) 0x02: M = 3 (prohibited) 0x03: M = 4 ⋮ 0x7C: M = 125 (default) ⋮ 0xFF: M = 256	R/W

Bit	Symbol	Function	R/W
28:24	SINCRNG[4:0]	Sinc Filter Output Valid Range When overflow occurs, 0x7FFFF is output. When underflow occurs, 0x80000 is output. S: Signed bit 0x00: {S, [32:14]} 0x01: {S, [31:13]} 0x02: {S, [30:12]} 0x03: {S, [29:11]} 0x04: {S, [28:10]} 0x05: {S, [27:9]} (default) 0x06: {S, [26:8]} 0x07: {S, [25:7]} 0x08: {S, [24:6]} 0x09: {S, [23:5]} 0x0A: {S, [22:4]} 0x0B: {S, [21:3]} 0x0C: {S, [20:2]} 0x0D: {S, [19:1]} 0x0E: {S, [18:0]} 0x0F: {S, [17:0], 0} 0x10: {S, [16:0], 00} 0x11: {S, [15:0], 000} 0x12: {S, [14:0], 0000} 0x13: {S, [13:0], 00000} 0x14: {S, [12:0], 000000} 0x15: {S, [11:0], 0000000} 0x16: {S, [10:0], 00000000} 0x17: {S, [9:0], 000000000} 0x18: {S, [8:0], 0000000000} 0x19: {S, [7:0], 00000000000} 0x1A: {S, [6:0], 000000000000} 0x1B: {S, [5:0], 0000000000000} 0x1C: {S, [4:0], 00000000000000} 0x1D: {S, [3:0], 000000000000000} 0x1E: {S, [2:0], 0000000000000000} 0x1F: {S, [1:0], 00000000000000000} (prohibited)	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

See Table 49.7 for the setting values of SINCDEC[7:0] and SINCRNG[4:0].

49.2.21 PDHFCS0RCHn : High-pass Filter Coefficient s(0) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x128 + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HFS[15:0]															
Value after reset*1:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	1	1	0	0	0	0	1

Bit	Symbol	Function	R/W
15:0	HFS[15:0]	High-pass Filter Coefficient s(0) 16-bit signed data (Fraction bit [13:0])	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. High-pass filter cut-off frequency = 100 Hz

49.2.22 PDHFCK1RCHn : High-pass Filter Coefficient k(1) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x12C + 0x100 × n

Bit position:	31																15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HFK[15:0]															
Value after reset[*1]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	1	0	0	0	0	0	1

Bit	Symbol	Function	R/W
15:0	HFK[15:0]	High-pass Filter Coefficient k(1) 16-bit signed data (Fraction bit [13:0])	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. High-pass filter cut-off frequency = 100 Hz

49.2.23 PDHFCH0RCHn : High-pass Filter Coefficient h(0) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x130 + 0x100 × n

Bit position:	31																15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HFH[15:0]															
Value after reset[*1]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	HFH[15:0]	High-pass Filter Coefficient h(0) 16-bit signed data (Fraction bit [13:0])	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. High-pass filter cut-off frequency = 100 Hz

49.2.24 PDHFCH1RCHn : High-pass Filter Coefficient h(1) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x134 + 0x100 × n

Bit position:	31																15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HFH[15:0]															
Value after reset[*1]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	HFH[15:0]	High-pass Filter Coefficient h(1) 16-bit signed data (Fraction bit [13:0])	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. High-pass filter cut-off frequency = 100 Hz

49.2.25 PDCFCH00RCHn : Compensation Filter Coefficient h(0) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x138 + 0x100 × n

Bit position:	31																			12												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFH[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	1	0	0	0

Bit	Symbol	Function	R/W
12:0	CFH[12:0]	Compensation Filter Coefficients h(0) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

49.2.26 PDCFCH01RCHn : Compensation Filter Coefficient h(1) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x13C + 0x100 × n

Bit position:	31																			12												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFH[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1

Bit	Symbol	Function	R/W
12:0	CFH[12:0]	Compensation Filter Coefficients h(1) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

49.2.27 PDCFCH02RCHn : Compensation Filter Coefficient h(2) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x140 + 0x100 × n

Bit position:	31																			12												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFH[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0

Bit	Symbol	Function	R/W
12:0	CFH[12:0]	Compensation Filter Coefficients h(2) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

49.2.28 PDCFCH03RCHn : Compensation Filter Coefficient h(3) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x144 + 0x100 × n

Bit position:	31																			12												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFH[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	1	0	1	1	0

Bit	Symbol	Function	R/W
12:0	CFH[12:0]	Compensation Filter Coefficients h(3) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

49.2.29 PDCFCH04RCHn : Compensation Filter Coefficient h(4) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x148 + 0x100 × n

Bit position:	31																			12												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFH[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	0

Bit	Symbol	Function	R/W
12:0	CFH[12:0]	Compensation Filter Coefficients h(4) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

49.2.30 PDCFCH05RCHn : Compensation Filter Coefficient h(5) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x14C + 0x100 × n

Bit position:	31																			12												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFH[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	0	0	0	0	1

Bit	Symbol	Function	R/W
12:0	CFH[12:0]	Compensation Filter Coefficients h(5) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

49.2.31 PDCFCH06RCHn : Compensation Filter Coefficient h(6) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x150 + 0x100 × n

Bit position:	31																			12												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFH[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	0

Bit	Symbol	Function	R/W
12:0	CFH[12:0]	Compensation Filter Coefficients h(6) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

49.2.32 PDCFCH07RCHn : Compensation Filter Coefficient h(7) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x154 + 0x100 × n

Bit position:	31																			12												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFH[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	0	1	0	1	1	0

Bit	Symbol	Function	R/W
12:0	CFH[12:0]	Compensation Filter Coefficients h(7) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

49.2.33 PDCFCH08RCHn : Compensation Filter Coefficient h(8) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x158 + 0x100 × n

Bit position:	31																			12												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFH[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0

Bit	Symbol	Function	R/W
12:0	CFH[12:0]	Compensation Filter Coefficients h(8) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

49.2.34 PDCFCH09RCHn : Compensation Filter Coefficient h(9) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x15C + 0x100 × n

Bit position:	31																			12												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFH[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1

Bit	Symbol	Function	R/W
12:0	CFH[12:0]	Compensation Filter Coefficients h(9) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

49.2.35 PDCFCH10RCHn : Compensation Filter Coefficient h(10) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x160 + 0x100 × n

Bit position:	31																			12												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFH[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	1	0	0	0

Bit	Symbol	Function	R/W
12:0	CFH[12:0]	Compensation Filter Coefficients h(10) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

49.2.36 PDLFCH010RCHn : Low-pass Filter Coefficient h0(10) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x164 + 0x100 × n

Bit position:	31																			12												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LFH0[12:0]												
Value after reset*1:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
12:0	LFH0[12:0]	Low-pass (half-band decimation) Filter Coefficient h0(10) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Low-pass filter cut-off frequency = 7 kHz

49.2.37 PDLFCH100RCHn : Low-pass Filter Coefficient h1(0) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x168 + 0x100 × n

Bit position:	31																			12												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0

Bit	Symbol	Function	R/W
12:0	LFH1[12:0]	Low-pass (half-band decimation) Filter Coefficient h1(0) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Low-pass filter cut-off frequency = 7 kHz

49.2.38 PDLFCH101RCHn : Low-pass Filter Coefficient h1(1) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x16C + 0x100 × n

Bit position:	31																			12												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

Bit	Symbol	Function	R/W
12:0	LFH1[12:0]	Low-pass (half-band decimation) Filter Coefficient h1(1) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Low-pass filter cut-off frequency = 7 kHz

49.2.39 PDLFCH102RCHn : Low-pass Filter Coefficient h1 (2) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x170 + 0x100 × n

Bit position:	31																			12												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0

Bit	Symbol	Function	R/W
12:0	LFH1[12:0]	Low-pass (half-band decimation) Filter Coefficient h1(2) 13-bit signed data (Fraction bit [10:0])	R/W

Bit	Symbol	Function	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Low-pass filter cut-off frequency = 7 kHz

49.2.40 PDLFCH103RCHn : Low-pass Filter Coefficient h1(3) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x174 + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0

Bit	Symbol	Function	R/W
12:0	LFH1[12:0]	Low-pass (half-band decimation) Filter Coefficient h1(3) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Low-pass filter cut-off frequency = 7 kHz

49.2.41 PDLFCH104RCHn : Low-pass Filter Coefficient h1(4) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x178 + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	1	1	1	0	0

Bit	Symbol	Function	R/W
12:0	LFH1[12:0]	Low-pass (half-band decimation) Filter Coefficient h1(4) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Low-pass filter cut-off frequency = 7 kHz

49.2.42 PDLFCH105RCHn : Low-pass Filter Coefficient h1(5) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x17C + 0x100 × n

Bit position: 31 … 12 … 0

Bit field: — — — — — — — — — — — — — — — — — — — | LFH1[12:0]

Value after reset*1: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0

Bit	Symbol	Function	R/W
12:0	LFH1[12:0]	Low-pass (half-band decimation) Filter Coefficient h1(5) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Low-pass filter cut-off frequency = 7 kHz

49.2.43 PDLFCH106RCHn : Low-pass Filter Coefficient h1(6) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x180 + 0x100 × n

Bit position: 31 … 12 … 0

Bit field: — — — — — — — — — — — — — — — — — — — | LFH1[12:0]

Value after reset*1: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 1 1 0 0 1 1

Bit	Symbol	Function	R/W
12:0	LFH1[12:0]	Low-pass (half-band decimation) Filter Coefficient h1(6) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Low-pass filter cut-off frequency = 7 kHz

49.2.44 PDLFCH107RCHn : Low-pass Filter Coefficient h1(7) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x184 + 0x100 × n

Bit position: 31 … 12 … 0

Bit field: — — — — — — — — — — — — — — — — — — — | LFH1[12:0]

Value after reset*1: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 1 0

Bit	Symbol	Function	R/W
12:0	LFH1[12:0]	Low-pass (half-band decimation) Filter Coefficient h1(7) 13-bit signed data (Fraction bit [10:0])	R/W

Bit	Symbol	Function	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Low-pass filter cut-off frequency = 7 kHz

49.2.45 PDLFCH108RCHn : Low-pass Filter Coefficient h1(8) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x188 + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LFH1[12:0]												
Value after reset[*1]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	0	1	1	1	0

Bit	Symbol	Function	R/W
12:0	LFH1[12:0]	Low-pass (half-band decimation) Filter Coefficient h1(8) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Low-pass filter cut-off frequency = 7 kHz

49.2.46 PDLFCH109RCHn : Low-pass Filter Coefficient h1(9) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x18C + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LFH1[12:0]												
Value after reset[*1]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1

Bit	Symbol	Function	R/W
12:0	LFH1[12:0]	Low-pass (half-band decimation) Filter Coefficient h1(9) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Low-pass filter cut-off frequency = 7 kHz

49.2.47 PDLFCH110RCHn : Low-pass Filter Coefficient h1(10) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x190 + 0x100 × n

Bit position:	31																			12												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	1

Bit	Symbol	Function	R/W
12:0	LFH1[12:0]	Low-pass (half-band decimation) Filter Coefficient h1(10) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Low-pass filter cut-off frequency = 7 kHz

49.2.48 PDLFCH111RCHn : Low-pass Filter Coefficient h1(11) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x194 + 0x100 × n

Bit position:	31																			12												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	0	1	1	1	0

Bit	Symbol	Function	R/W
12:0	LFH1[12:0]	Low-pass (half-band decimation) Filter Coefficient h1(11) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Low-pass filter cut-off frequency = 7 kHz

49.2.49 PDLFCH112RCHn : Low-pass Filter Coefficient h1(12) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x198 + 0x100 × n

Bit position:	31																			12												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0

Bit	Symbol	Function	R/W
12:0	LFH1[12:0]	Low-pass (half-band decimation) Filter Coefficient h1(12) 13-bit signed data (Fraction bit [10:0])	R/W

Bit	Symbol	Function	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Low-pass filter cut-off frequency = 7 kHz

49.2.50 PDLFCH113RCHn : Low-pass Filter Coefficient h1(13) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x19C + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LFH1[12:0]												
Value after reset[*1]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	1	1	0	0	1	1

Bit	Symbol	Function	R/W
12:0	LFH1[12:0]	Low-pass (half-band decimation) Filter Coefficient h1(13) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Low-pass filter cut-off frequency = 7 kHz

49.2.51 PDLFCH114RCHn : Low-pass Filter Coefficient h1(14) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x1A0 + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LFH1[12:0]												
Value after reset[*1]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0

Bit	Symbol	Function	R/W
12:0	LFH1[12:0]	Low-pass (half-band decimation) Filter Coefficient h1(14) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Low-pass filter cut-off frequency = 7 kHz

49.2.52 PDLFCH115RCHn : Low-pass Filter Coefficient h1(15) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x1A4 + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LFH1[12:0]												
Value after reset[*1]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	1	1	1	0	0

Bit	Symbol	Function	R/W
12:0	LFH1[12:0]	Low-pass (half-band decimation) Filter Coefficient h1(15) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Low-pass filter cut-off frequency = 7 kHz

49.2.53 PDLFCH116RCHn : Low-pass Filter Coefficient h1(16) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x1A8 + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LFH1[12:0]												
Value after reset[*1]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0

Bit	Symbol	Function	R/W
12:0	LFH1[12:0]	Low-pass (half-band decimation) Filter Coefficient h1(15) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Low-pass filter cut-off frequency = 7 kHz

49.2.54 PDLFCH117RCHn : Low-pass Filter Coefficient h1(17) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x1AC + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LFH1[12:0]												
Value after reset[*1]:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0

Bit	Symbol	Function	R/W
12:0	LFH1[12:0]	Low-pass (half-band decimation) Filter Coefficient h1(17) 3-bit signed data (Fraction bit [10:0])	R/W

Bit	Symbol	Function	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Low-pass filter cut-off frequency = 7 kHz

49.2.55 PDLFCH118RCHn : Low-pass Filter Coefficient h1(18) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x1B0 + 0x100 × n

Bit position:	31																			12												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

Bit	Symbol	Function	R/W
12:0	LFH1[12:0]	Low-pass (half-band decimation) Filter Coefficient h1(18) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Low-pass filter cut-off frequency = 7 kHz

49.2.56 PDLFCH119RCHn : Low-pass Filter Coefficient h1(19) Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x1B4 + 0x100 × n

Bit position:	31																			12												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LFH1[12:0]												
Value after reset*1:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0

Bit	Symbol	Function	R/W
12:0	LFH1[12:0]	Low-pass (half-band decimation) Filter Coefficient h1(19) 13-bit signed data (Fraction bit [10:0])	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Low-pass filter cut-off frequency = 7 kHz

49.2.57 PDSDLTRCHn : Sound Detection Lower Threshold Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x1B8 + 0x100 × n

Bit position:	31												19																			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	SDETL[19:0]																			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
19:0	SDETL[19:0]	Sound Detection Lower Limit This bit sets the lower limit value (20-bit signed data) for sound detection.	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

49.2.58 PDSDUTRCHn : Sound Detection Upper Threshold Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x1BC + 0x100 × n

Bit position:	31												19																			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	SDETU[19:0]																			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
19:0	SDETU[19:0]	Sound Detection Upper Limit This bit sets the upper limit value (20-bit signed data) for sound detection.	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

49.2.59 PDDBCRCHn : Data Buffer Control Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x1C0 + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	DATRITHR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	DATRITHR[2:0]	Data Reception Interrupt Threshold 0 0 0: Output interrupt when receiving 1 or more data 0 0 1: Output interrupt when receiving 2 or more data 0 1 0: Output interrupt when receiving 4 or more data 0 1 1: Output interrupt when receiving 8 or more data Others: Output interrupt when receiving 16 or more data	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

49.2.60 PDSCTSRCHn : Short Circuit Threshold Setting Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x1C4 + 0x100 × n

Bit position:	31			28	16			12	0
Bit field:	—	—	—	SCDH[12:0]		—	—	—	SCDL[12:0]
Value after reset:	0	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0		0	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
12:0	SCDL[12:0]	Short Circuit Detection Low Continuous Detection Count Set the upper limit for the number of times 0 are input continuously to the PDM_DATAn pin.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
28:16	SCDH[12:0]	Short Circuit Detection High Continuous Detection Count Set the upper limit for the number of times 1 are input continuously to the PDM_DATAn pin.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

49.2.61 PDOVLTRCHn : Overvoltage Lower Threshold Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x1C8 + 0x100 × n

Bit position:	31												19	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	OVDL[19:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
19:0	OVDL[19:0]	Overvoltage Detection Lower Limit These bits set the lower limit of the active voltage value (20-bit signed data) for overvoltage detection.	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

49.2.62 PDOVUTRCHn : Overvoltage Upper Threshold Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x1CC + 0x100 × n

Bit position:	31												19	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	OVDU[19:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
19:0	OVDU[19:0]	Overvoltage Detection Upper Limit These bits set the upper limit of the active voltage value (20-bit signed data) for overvoltage detection.	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

49.2.63 PDDRCRCHn : Data Read Control Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x1E0 + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DATRE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DATRE	Data Read Enable Bit 0: Do not allow reading of data from buffer 1: Allow reading of data from buffer	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

49.2.64 PDDCRCHn : Data Clear Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x1E4 + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DATC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DATC	Data Clear 0: Do nothing 1: Clear data	W
31:1	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

49.2.65 PDDRRCHn : Data Read Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x1E8 + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19–0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	DAT[19:0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
19:0	DAT[19:0]	Data 20-bit signed sound data or 16-bit signed sound data is read out. For 20-bit mode (PDMDSRCHn.DBIS[3:0] = 0xxx), the signed bit is [19]. For 16-bit mode (PDMDSRCHn.DBIS[3:0] = 1xxx), the signed bits are [19:15]. Writing is invalid.	R
31:20	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

49.2.66 PDDSRCHn : Data Status Register Channel n (n = 0 to 2)

Base address: PDMIF = 0x4025_6000
PDMIF_NS = 0x5025_6000

Offset address: 0x1EC + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DATNUM[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	DATNUM[7:0]	Number of Data Stored in Buffer When PDDRCRCHn.DATRE = 0, 0 is read out. When PDDRCRCHn.DATRE = 1, the number of data stored in buffer is read out. For buffer size that is of 32 stages, the maximum value is 31.	R
31:8	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

49.2.67 Mirroring Register

The common registers except for the Version register have mirroring registers in each channel. PDCSR.EDFn is a register which ORs all the error flags of the mirroring registers in each channel.

Use the common registers for synchronous control between channels.

Table 49.4 Mirroring register

Common register		Mirroring register in each channel	
Register name	**Bit name**	**Register name**	**Bit name**
PDCSTRTR	STRTRGn	PDSTRTR	STRTRG
PDCSTPTR	STPTRGn	PDSTPTR	STPTRG
PDCCHGTR	CHGTRGn	PDCHGTR	CHGTRG
PDCICR	ISDEn	PDICR	ISDE
	IDREn		IDRE
	IEDEn		IEDE
PDCSR	STATEn	PDSR	STATE
	SDFn		SDF
	DRFn		DRF
PDCSCR	SDFCn	PDSCR	SDFC
PDCSDCR	SDEn	PDSDCR	SDE
PDCDRCR	DATREn	PDDRCR	DATRE
PDCDCR	DATCn	PDDCR	DATC

49.3 Operation

49.3.1 Clock Specification

The frequency-divided core clock is generated and output to PDM_CLKn.

Table 49.5 shows guaranteed clock frequency and clock mode in PDM-IF. PDM clock (PDMIFCLK) and PCLK are asynchronous.

Table 49.5 Guaranteed clock frequency and clock mode

PDMIFCLK	PCLK	PDM_CLKn		
Frequency (MHz)	Frequency (MHz)	Frequency (MHz)	Supported core clock Division ratio	Setting value (PDSFCR.CKDIV[3:0])
8	1 to 62.5*1	4	2	0x0
		2	4	0x1
		1.33	6	0x2
		1	8	0x3
		0.8	10	0x4
		0.67	12	0x5
		0.57	14	0x6
		0.5	16	0x7
		0.44	18	0x8
		0.4	20	0x9
		0.36	22	0xA
		0.33	24	0xB
		0.31	26	0xC
		0.29	28	0xD
		0.27	30	0xE
		0.25	32	0xF

Note 1. PDM-IF is possible to operate at PCLK below 1 MHz but the guaranteed frequency operation range is 1 to 62.5 MHz.

49.3.1.1 Output Sampling Frequency

PDM-IF output sampling frequency is shown by the following formula:

$F_{out} = F_{clk} / (M \times 2)$

- F_{out}: PDM-IF output sampling frequency
- F_{clk}: PDM_CLKn frequency (See Table 49.5)
- M: Sinc filter decimation ratio (PDSFCRCHn.SINCDEC[7:0] + 1)

49.3.2 Channel Start and Channel Stop

The channel start and the channel stop is controlled by the register of Table 49.6.

Set the channels to be synchronized to 1 at the same time by using the common registers.

Set the channel start according to section 49.4.1. Start Flow.

Set the channel stop according to section 49.4.2. Stop Flow.

Table 49.6 Channel start trigger and channel stop trigger

Category	Register bit name	
	Common register	Channel register
Channel start	PDCSTRTR.STRTRG0	PDSTRTRCH0.STRTRG
	PDCSTRTR.STRTRG1	PDSTRTRCH1.STRTRG
	PDCSTRTR.STRTRG2	PDSTRTRCH2.STRTRG
Channel stop	PDCSTPTR.STPTRG0	PDSTPTRCH0.STPTRG
	PDCSTPTR.STPTRG1	PDSTPTRCH1.STPTRG
	PDCSTPTR.STPTRG2	PDSTPTRCH2.STPTRG

49.3.3 PDM_CLKn Frequency Change

The external microphone transitions the power mode according to input clock frequency. PDM-IF enables to transition the power mode of the microphone by changing the operating frequency of PDM_CLKn which is output to the microphone when operating. The operating frequency of PDM_CLKn can be changed using PDSFCR.CKDIV[3:0] bits. When the frequency is changed, the decimation ratio of the sinc filter must also be changed. Set PDSRCRCHn.SINCDEC[7:0] so that the sampling frequency after the decimation can be 32 kHz*1. Furthermore, when the decimation ratio is changed, the output range of the sinc filter is also changed. Set PDSRCRCHn.SINCRNG[4:0] together in accordance with the output range. See Table 49.7 for details.

After changing the configuration of these registers, software must wait for the settling time of microphone and PDM-IF channel. See section 49.4.4. Low-power Mode Transition Flow and section 49.4.5. Normal Mode Transition Flow for details.

Note 1. In case that the sampling frequency is different from 32 kHz, the characteristics of filters after the sinc filter might change.

49.3.4 Filter

49.3.4.1 Data Clipping and Shifting between Filters

Voice sample data is clipped or shifted between filters. Figure 49.2 shows this operation.

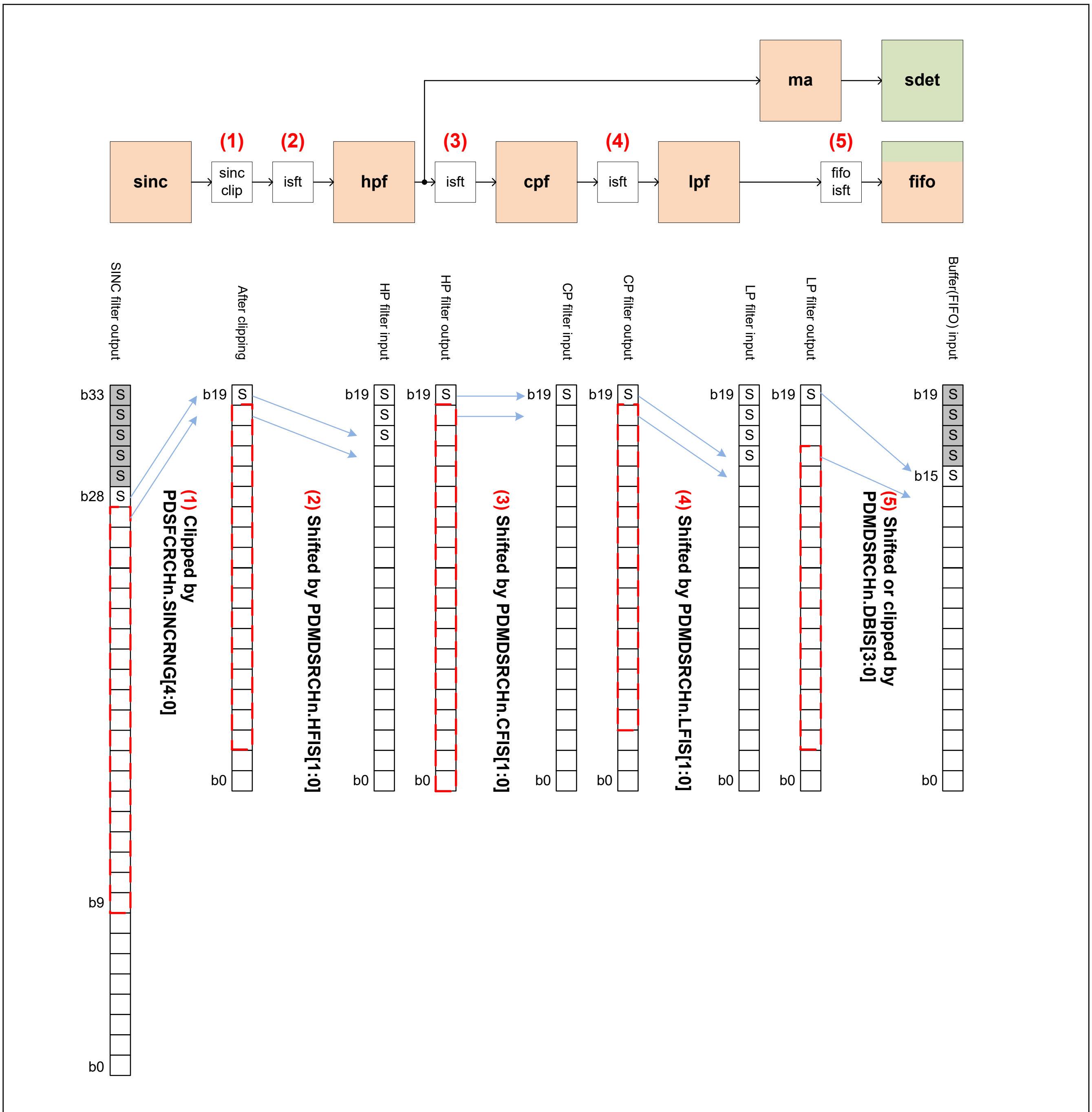


Figure 49.2 Data clipping and shifting between filters

(1) Sinc filter output clip

Table 49.7 lists the available settings of the sinc filter and data clipping.

It is controlled by PDSFCRCHn.SINCRNG[4:0].

It is prohibited to set other than those in Table 49.7.

Table 49.7 Sinc filter output clip (1 of 3)

PDMIFCLK (MHz)	Sinc filter order	PDMDSRCHn.SFMD[2:0]	PDM_CLK dividend ratio	PDSFCRCHn.CKDIV[3:0]	Sinc filter decimation ratio	PDSFCRCHn.SINCDEC[7:0]	Sinc filter output effective bits	Sinc filter output clipping (20 bits)	PDSFCRCHn. SINCRNG[4:0]	Resolution	PDM_CLKn frequency (MHz)	SINC filter output sampling frequency (kHz)	PDM-IF output sampling frequency (kHz)
8	4	0x0	2	0x0	125	0x7C	b28 to b0	b28 to b9	00101	20	4.0	32	16
			4	0x1	62	0x3D	b24 to b0	b24 to b5	01001	20	2.0	32.26	16.13
			6	0x2	42	0x29	b22 to b0	b22 to b3	01011	20	1.33	31.75	15.87
			8	0x3	31	0x1E	b20 to b0	b20 to b1	01101	20	1.0	32.26	16.13
			10	0x4	25	0x18	b19 to b0	b19 to b0	01110	20	0.8	32	16
			12	0x5	21	0x14	b18 to b0	b18 to b0, 0	01111	19	0.67	31.75	15.87
			14	0x6	18	0x11	b17 to b0	b17 to b0, 00	10000	18	0.57	31.75	15.87
			16	0x7	16	0x0F	b17 to b0	b17 to b0, 00	10000	18	0.50	31.25	15.63
			18	0x8	14	0x0D	b16 to b0	b16 to b0, 000	10001	17	0.44	31.75	15.87
			20	0x9	12	0x0B	b15 to b0	b15 to b0, 0000	10010	16	0.40	33.33	16.67
			22	0xA	11	0x0A	b14 to b0	b14 to b0, 00000	10011	15	0.36	33.06	16.53
			24	0xB	10	0x09	b14 to b0	b14 to b0, 00000	10011	15	0.33	33.33	16.67
			26	0xC	9	0x08	b13 to b0	b13 to b0, 000000	10100	14	0.31	34.19	17.09
			28	0xD	9	0x08	b13 to b0	b13 to b0, 000000	10100	14	0.29	31.75	15.87
			30	0xE	8	0x07	b13 to b0	b13 to b0, 000000	10100	14	0.27	33.33	16.67
			32	0xF	8	0x07	b13 to b0	b13 to b0, 000000	10100	14	0.25	31.25	15.63
8	3	0x3	2	0x0	125	0x7C	b21 to b0	b21 to b2	01100	20	4.0	32	16
			4	0x1	62	0x3D	b18 to b0	b18 to b0, 0	01111	19	2.0	32.26	16.13
			6	0x2	42	0x29	b17 to b0	b17 to b0, 00	10000	18	1.33	31.75	15.87
			8	0x3	31	0x1E	b15 to b0	b15 to b0, 0000	10010	16	1.0	32.26	16.13
			10	0x4	25	0x18	b14 to b0	b14 to b0, 00000	10011	15	0.8	32	16
			12	0x5	21	0x14	b14 to b0	b14 to b0, 00000	10011	15	0.67	31.75	15.87
			14	0x6	18	0x11	b13 to b0	b13 to b0, 000000	10100	14	0.57	31.75	15.87
			16	0x7	16	0x0F	b13 to b0	b13 to b0, 000000	10100	14	0.50	31.25	15.63
			18	0x8	14	0x0D	b12 to b0	b12 to b0, 0000000	10101	13	0.44	31.75	15.87
			20	0x9	12	0x0B	b11 to b0	b11 to b0, 00000000	10110	12	0.40	33.33	16.67
			22	0xA	11	0x0A	b11 to b0	b11 to b0, 00000000	10110	12	0.36	33.06	16.53
			24	0xB	10	0x09	b10 to b0	b10 to b0, 000000000	10111	11	0.33	33.33	16.67
			26	0xC	9	0x08	b10 to b0	b10 to b0, 000000000	10111	11	0.31	34.19	17.09
			28	0xD	9	0x08	b10 to b0	b10 to b0, 000000000	10111	11	0.29	31.75	15.87
			30	0xE	8	0x07	b10 to b0	b10 to b0, 000000000	10111	11	0.27	33.33	16.67
			32	0xF	8	0x07	b10 to b0	b10 to b0, 000000000	10111	11	0.25	31.25	15.63

Table 49.7 Sinc filter output clip (2 of 3)

PDMIFCLK (MHz)	Sinc filter order	PDMDSRCHn.SFMD[2:0]	PDM_CLK dividend ratio	PDSFCRCHn.CKDIV[3:0]	Sinc filter decimation ratio	PDSFCRCHn.SINCDEC[7:0]	Sinc filter output effective bits	Sinc filter output clipping (20 bits)	PDSFCRCHn. SINCRNG[4:0]	Resolution	PDM_CLKn frequency (MHz)	SINC filter output sampling frequency (kHz)	PDM-IF output sampling frequency (kHz)
8	2	0x2	2	0x0	125	0x7C	b14 to b0	b14 to b0, 00000	10011	15	4.0	32	16
			4	0x1	62	0x3D	b12 to b0	b12 to b0, 0000000	10101	13	2.0	32.26	16.13
			6	0x2	42	0x29	b11 to b0	b11 to b0, 00000000	10110	12	1.33	31.75	15.87
			8	0x3	31	0x1E	b10 to b0	b10 to b0, 000000000	10111	11	1.0	32.26	16.13
			10	0x4	25	0x18	b10 to b0	b10 to b0, 000000000	10111	11	0.8	32	16
			12	0x5	21	0x14	b9 to b0	b9 to b0, 0000000000	11000	10	0.67	31.75	15.87
			14	0x6	18	0x11	b9 to b0	b9 to b0, 0000000000	11000	10	0.57	31.75	15.87
			16	0x7	16	0x0F	b9 to b0	b9 to b0, 0000000000	11000	10	0.50	31.25	15.63
			18	0x8	14	0x0D	b8 to b0	b8 to b0, 00000000000	11001	9	0.44	31.75	15.87
			20	0x9	12	0x0B	b8 to b0	b8 to b0, 00000000000	11001	9	0.40	33.33	16.67
			22	0xA	11	0x0A	b7 to b0	b7 to b0, 000000000000	11010	8	0.36	33.06	16.53
			24	0xB	10	0x09	b7 to b0	b7 to b0, 000000000000	11010	8	0.33	33.33	16.67
			26	0xC	9	0x08	b7 to b0	b7 to b0, 000000000000	11010	8	0.31	34.19	17.09
			28	0xD	9	0x08	b7 to b0	b7 to b0, 000000000000	11010	8	0.29	31.75	15.87
			30	0xE	8	0x07	b7 to b0	b7 to b0, 000000000000	11010	8	0.27	33.33	16.67
			32	0xF	8	0x07	b7 to b0	b7 to b0, 000000000000	11010	8	0.25	31.25	15.63

Table 49.7 Sinc filter output clip (3 of 3)

PDMIFCLK (MHz)	Sinc filter order	PDMDSRCHn.SFMD[2:0]	PDM_CLK dividend ratio	PDSFCRCHn.CKDIV[3:0]	Sinc filter decimation ratio	PDSFCRCHn.SINCDEC[7:0]	Sinc filter output effective bits	Sinc filter output clipping (20 bits)	PDSFCRCHn. SINCRNG[4:0]	Resolution	PDM_CLKn frequency (MHz)	SINC filter output sampling frequency (kHz)	PDM-IF output sampling frequency (kHz)
8	1	0x1	2	0x0	125	0x7C	b7 to b0	b7 to b0, 000000000000	11010	8	4.0	32	16
			4	0x1	62	0x3D	b6 to b0	b6 to b0, 0000000000000	11011	7	2.0	32.26	16.13
			6	0x2	42	0x29	b6 to b0	b6 to b0, 0000000000000	11011	7	1.33	31.75	15.87
			8	0x3	31	0x1E	b5 to b0	b5 to b0, 00000000000000	11100	6	1.0	32.26	16.13
			10	0x4	25	0x18	b5 to b0	b5 to b0, 00000000000000	11100	6	0.8	32	16
			12	0x5	21	0x14	b5 to b0	b5 to b0, 00000000000000	11100	6	0.67	31.75	15.87
			14	0x6	18	0x11	b5 to b0	b5 to b0, 00000000000000	11100	6	0.57	31.75	15.87
			16	0x7	16	0x0F	b5 to b0	b5 to b0, 00000000000000	11100	6	0.50	31.25	15.63
			18	0x8	14	0x0D	b4 to b0	b4 to b0, 000000000000000	11101	5	0.44	31.75	15.87
			20	0x9	12	0x0B	b4 to b0	b4 to b0, 000000000000000	11101	5	0.40	33.33	16.67
			22	0xA	11	0x0A	b4 to b0	b4 to b0, 000000000000000	11101	5	0.36	33.06	16.53
			24	0xB	10	0x09	b4 to b0	b4 to b0, 000000000000000	11101	5	0.33	33.33	16.67
			26	0xC	9	0x08	b4 to b0	b4 to b0, 000000000000000	11101	5	0.31	34.19	17.09
			28	0xD	9	0x08	b4 to b0	b4 to b0, 000000000000000	11101	5	0.29	31.75	15.87
			30	0xE	8	0x07	b4 to b0	b4 to b0, 000000000000000	11101	5	0.27	33.33	16.67
			32	0xF	8	0x07	b4 to b0	b4 to b0, 000000000000000	11101	5	0.25	31.25	15.63

An overflow or an underflow never occurs as long as the combinations which are shown in Table 49.7 are set to PDMDSRCHn.SFMD[2:0], PDSFCRCHn.CKDIV[3:0], PDSFCRCHn.SINCDEC[7:0] and PDSFCRCHn.SINCRNG[4:0]. However, when other combinations are set to them, the overflow or the underflow might occur. When the overflow occurs, 0x7FFFF is output. When the underflow occurs, 0x80000 is output. Use the overvoltage detection function if you want to know if the overflow or the underflow occurs.

(2) High-pass filter input shift

The clipped sinc filter output is right-shifted and input to the high-pass filter.

It is controlled by PDMDSRCHn.HFIS[1:0].

Table 49.8 High-pass filter input shift (1 of 2)

PDMDSR.HFIS[1:0]	Function	Shifted data*1
0x0	No shift	S, b18-b0
0x1	1-bit right-shift	S, S, b18-b1

Table 49.8 High-pass filter input shift (2 of 2)

PDMDSR.HFIS[1:0]	Function	Shifted data*1
0x2	2-bit right-shift	S, S, S, b18-b2
0x3	3-bit right-shift	S, S, S, S, b18-b3

Note 1. S: Signed bit (= bit 19)

(3) Compensation filter input shift

The high-pass filter output is right-shifted and input to the compensation filter.

It is controlled by PDMDSRCHn.CFIS[1:0].

Table 49.9 Compensation filter input shift

PDMDSR.CFIS[1:0]	Function	Shifted data*1
0x0	No shift	S, b18-b0
0x1	1-bit right-shift	S, S, b18-b1
0x2	2-bit right-shift	S, S, S, b18-b2
0x3	3-bit right-shift	S, S, S, S, b18-b3

Note 1. S: Signed bit (= bit 19)

(4) Low-pass filter input shift

The compensation filter output is right-shifted and input to the low-pass (half-band decimation) filter.

It is controlled by PDMDSRCHn.LFIS[1:0].

Table 49.10 Low-pass filter input shift

PDMDSR.LFIS[1:0]	Function	Shifted data*1
0x0	No shift	S, b18-b0
0x1	1-bit right-shift	S, S, b18-b1
0x2	2-bit right-shift	S, S, S, b18-b2
0x3	3-bit right-shift	S, S, S, S, b18-b3

Note 1. S: Signed bit (= bit 19)

(5) Buffer input shift or clip

The compensation filter output is right-shifted, or 16-bit data is clipped from the compensation filter output. The shifted or clipped data is then input to buffer.

It is controlled by PDMDSRCHn.DBIS[1:0].

Table 49.11 Buffer input shift or clip (1 of 2)

PDMDSRCHn.DBIS[3:0]	Function	Valid data width	Shifted or Clipped data*1
0000	No shift	20-bit	S, b18-b0
0001	1-bit right-shift	20-bit	S, S, b18-b1
0010	2-bit right-shift	20-bit	S, S, S, b18-b2
0011	3-bit right-shift	20-bit	S, S, S, S, b18-b3
1000	Signed bit and 18-4 is clipped	16-bit	S, b18-b4
1001	Signed bit and 17-3 is clipped*2	16-bit	S, b17-b3
1010	Signed bit and 16-2 is clipped*2	16-bit	S, b16-b2

Table 49.11 Buffer input shift or clip (2 of 2)

PDMDSRCHn.DBIS[3:0]	Function	Valid data width	Shifted or Clipped data*1
1011	Signed bit and 15-1 is clipped*2	16-bit	S, b15-b1
1100	Signed bit and 14-0 is clipped*2	16-bit	S, b14-b0

Note 1. S: Signed bit (= bit 19)
Note 2. When an overflow occurs, 0x7FFF is input to the buffer.
When an underflow occurs, 0x8000 is input to the buffer.

49.3.4.2 Sinc Filter

The operating clock of the sinc filter is PDMIFCLK. Sinc filter is operated using the 1-bit stream data latched with the rise-edge or fall-edge of PDM_CLKn. For details, see section 49.3.5. Data Input Control for Stereo Sound.

Figure 49.3 shows a sinc filter block diagram.

The differentiation stage is operated by decimation clock (the frequency is 1/M of PDM_CLKn). M is the decimation ratio and is set in the PDSFCRCHn.SINCDEC[7:0] bits. The filter result is output in register every decimation clock.

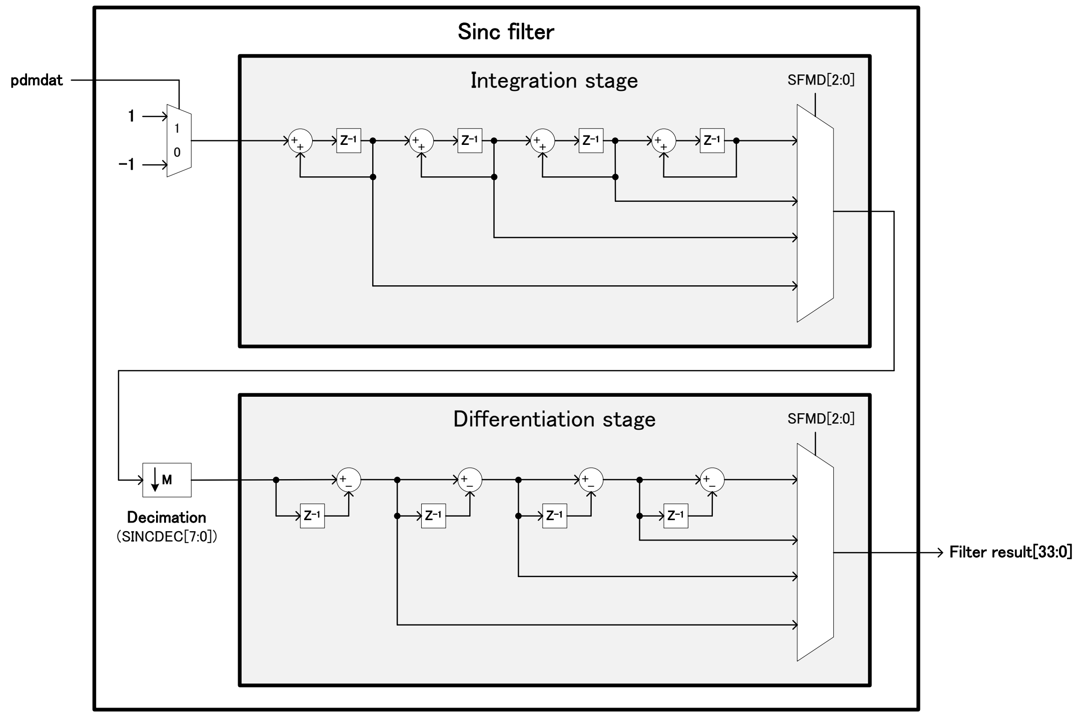


Figure 49.3 Sinc filter block diagram

49.3.4.3 High-pass Filter

The operating clock of the high-pass filter is PDMIFCLK. High-pass filter is operated using the result of the sinc filter.

Figure 49.4 shows a high-pass filter block diagram.

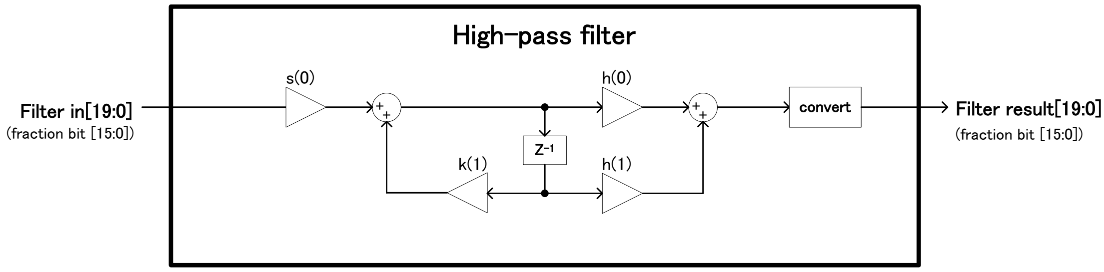


Figure 49.4 High-pass filter block diagram

Table 49.12 shows a high-pass filter specification.

Table 49.12 High-pass filter specification

Function		Specification
Input	Bit width	Signed 20-bit (fraction 16-bit)
Output	Bit width	Signed 20-bit (fraction 16-bit)
Coefficient	Bit width	Signed 16-bit (fraction 14-bit) For details, see section 49.2.21. PDHFCS0RCHn : High-pass Filter Coefficient s(0) Register Channel n (n = 0 to 2), section 49.2.22. PDHFCK1RCHn : High-pass Filter Coefficient k(1) Register Channel n (n = 0 to 2), and section 49.2.23. PDHFCH0RCHn : High-pass Filter Coefficient h(0) Register Channel n (n = 0 to 2).
Delay Z	Bit width	Signed 32-bit (fraction 18-bit)
Multiplier	Input A bit width	Signed 32-bit (fraction 18-bit)
	Input B bit width	Signed 16-bit (fraction 14-bit)
	Output bit width	Signed 47-bit (fraction 32-bit)
Adder	Input A bit width	Signed 32-bit (fraction 18-bit)
	Input B bit width	Signed 32-bit (fraction 18-bit)
	Output bit width	Signed 33-bit (fraction 18-bit)
Output conversion	Input bit width	Signed 32-bit (fraction 18-bit)
	Output bit width	Signed 20-bit (fraction 16-bit)
Excessive fraction bits processing		Excessive fraction bits are rounded down at functional unit input or final output

49.3.4.4 Compensation Filter

The operating clock of the compensation filter is PDMIFCLK. Compensation filter is operated using the result of the high-pass filter.

Figure 49.5 shows a compensation filter block diagram.

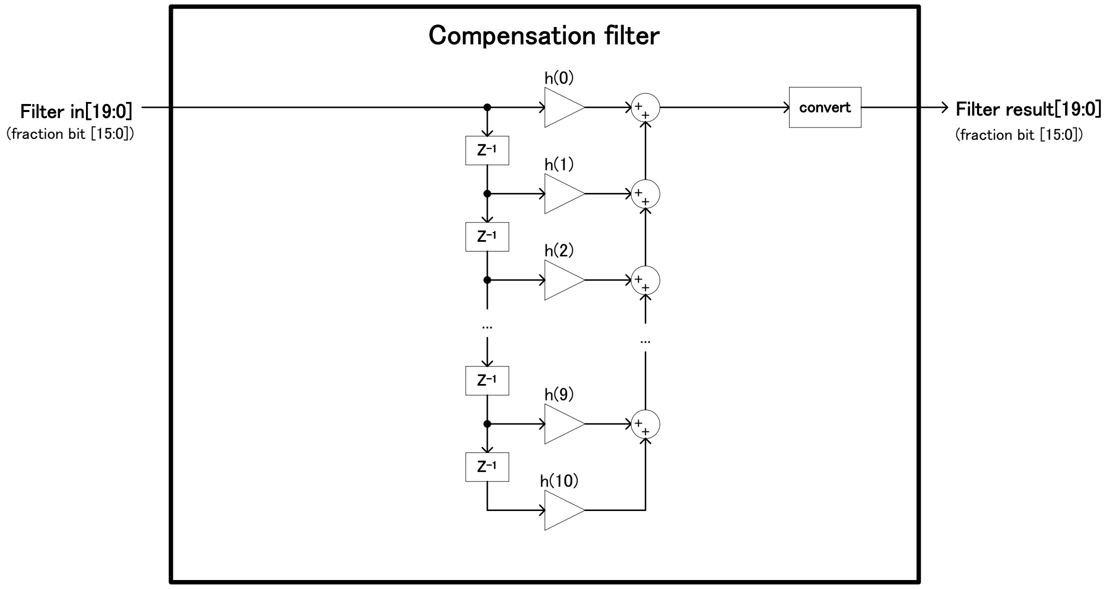


Figure 49.5 Compensation filter block diagram

Table 49.13 shows a compensation filter specification.

Table 49.13 Compensation filter specification

Function		Specification
Input	Bit width	Signed 20-bit (fraction 16-bit)
Output	Bit width	Signed 20-bit (fraction 16-bit)
Coefficient	Bit width	Signed 13-bit (fraction 11-bit) For details, see section 49.2.25. PDCFCH00RCHn : Compensation Filter Coefficient h(0) Register Channel n (n = 0 to 2).
Delay Z	Bit width	Signed 20-bit (fraction 16-bit)
Multiplier	Input A bit width	Signed 20-bit (fraction 16-bit)
	Input B bit width	Signed 13-bit (fraction 11-bit)
	Output bit width	Signed 32-bit (fraction 27-bit)
Adder	Input A bit width	Signed 22-bit (fraction 18-bit)
	Input B bit width	Signed 22-bit (fraction 18-bit)
	Output bit width	Signed 23-bit (fraction 18-bit)
Output conversion	Input bit width	Signed 22-bit (fraction 18-bit)
	Output bit width	Signed 20-bit (fraction 16-bit)
Excessive fraction bits processing		Excessive fraction bits are rounded down at functional unit input or final output

49.3.4.5 Low-pass (Half-band Decimation) Filter

The operating clock of the low-pass filter is PDMIFCLK. Low-pass filter is operated using the result of the compensation filter. Output data is decimated to a half of input data. As a result, output data frequency reduces to half of input frequency and the output data frequency is equal to the sample rate of data stored in FIFO.

Figure 49.6 shows a low-pass filter block diagram.

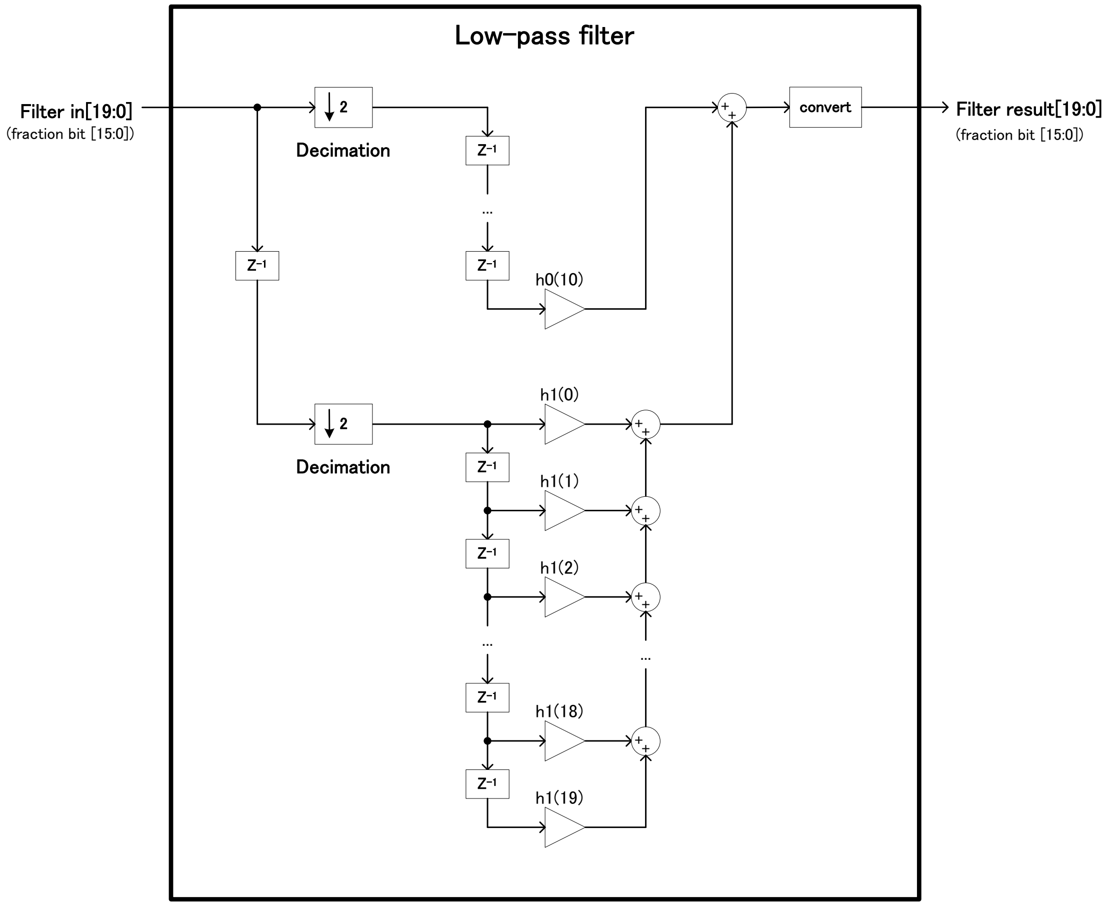


Figure 49.6 Low-pass filter block diagram

Table 49.14 shows a low-pass filter specification.

Table 49.14 Low-pass filter specification

Function		Specification
Input	Bit width	Signed 20-bit (fraction 16-bit)
Output	Bit width	Signed 20-bit (fraction 16-bit)
Coefficient	Bit width	Signed 13-bit (fraction 11-bit) For details, see section 49.2.25. PDCFCH00RCHn : Compensation Filter Coefficient h(0) Register Channel n (n = 0 to 2).
Delay Z	Bit width	Signed 20-bit (fraction 16-bit)
Multiplier	Input A bit width	Signed 20-bit (fraction 16-bit)
	Input B bit width	Signed 13-bit (fraction 11-bit)
	Output bit width	Signed 32-bit (fraction 27-bit)
Adder	Input A bit width	Signed 22-bit (fraction 18-bit)
	Input B bit width	Signed 22-bit (fraction 18-bit)
	Output bit width	Signed 23-bit (fraction 18-bit)
Output conversion	Input bit width	Signed 22-bit (fraction 18-bit)
	Output bit width	Signed 20-bit (fraction 16-bit)
Excessive fraction bits processing		Excessive fraction bits are rounded down at functional unit input or final output

49.3.4.6 Moving Average Filter

The operating clock of the moving average filter is PDMIFCLK. The moving average filter is operated using the result of the high-pass filter. The order of the moving average filter can be changed by the setting of the PDMDSRCHn.SDMAMD[1:0] bits.

Figure 49.7 shows a moving average filter block diagram.

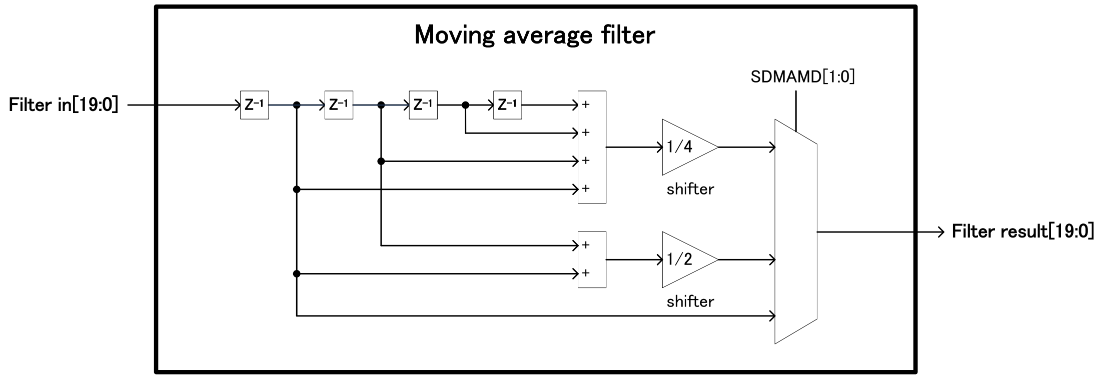


Figure 49.7 Moving average filter block diagram

Table 49.15 shows a moving average filter specification.

Table 49.15 Moving average filter specification

Function		Specification
Input	Bit width	Signed 20-bit (fraction 16-bit)
Output	Bit width	Signed 20-bit (fraction 16-bit)
Delay Z	Bit width	Signed 20-bit (fraction 16-bit)
Excessive fraction bits processing		Excessive fraction bits are rounded down at shifter

49.3.4.7 Settling Time of Channel Activation/Setting Change

Settling time of channel activation is the time after a start trigger PDCSTRTR.STRTRGn (or PDSTRTRCHn.STRTRG) is set before the filter result is stable and output. Settling time of channel setting change is the time after a change trigger PDCCHGTR.CHGTRGn (or PDCHGTRCHn.CHGTRG) is set before the filter result is stable and output. Figure 49.8 shows the settling time measurement point.

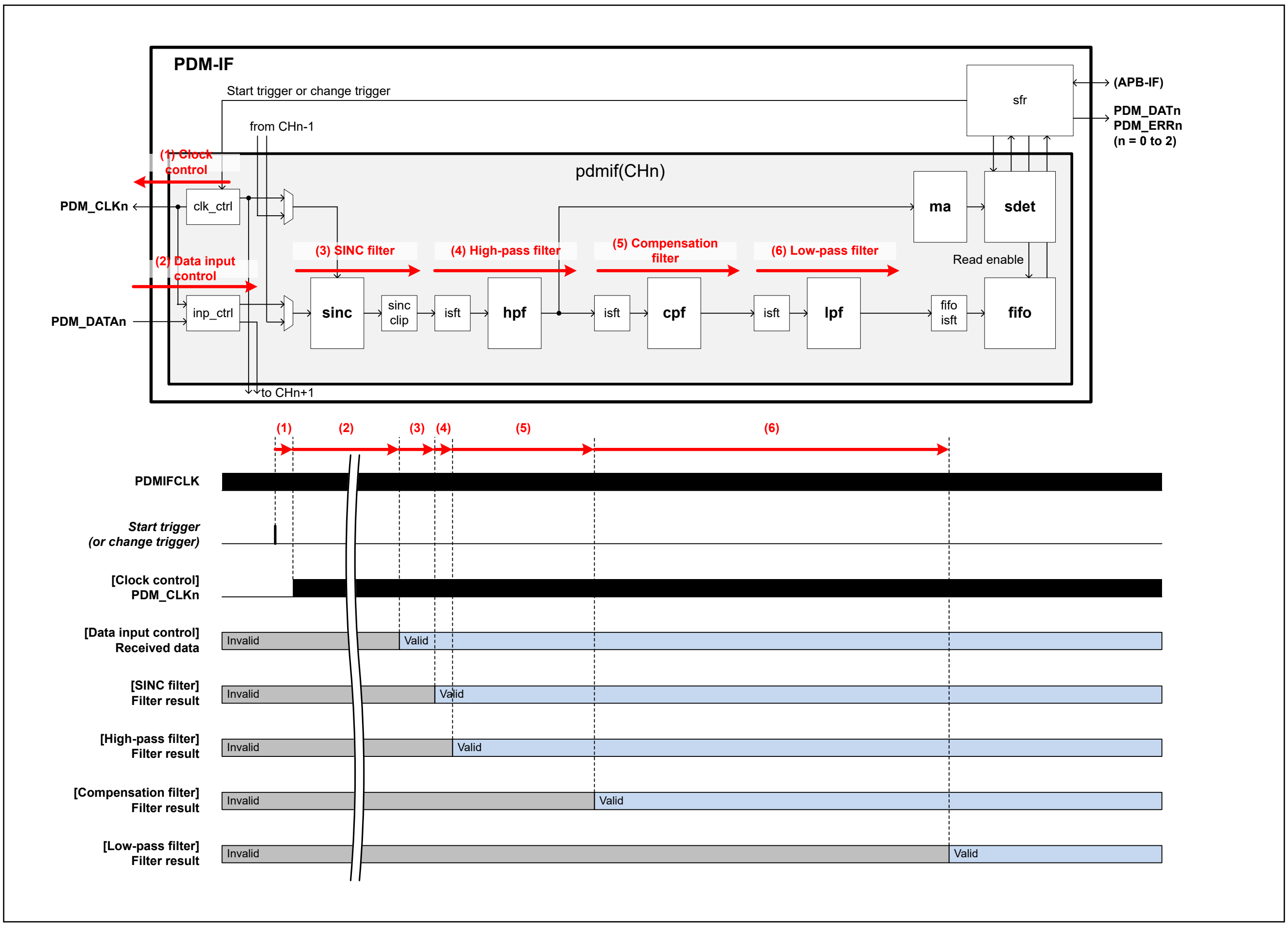


Figure 49.8 Settling time of channel activation

Settling time is calculated as Table 49.16.

Table 49.16 Settling time of channel activation/setting change

No	Function	Settling time	
		Channel activation	Channel setting change
(1)	Clock control	(4 + (D / 2)) × Cp	(3 + Do + (D / 2)) × Cp
(2)	Data input control	(D / 2) × Cp + (***Stable time for microphone***[*1])	
(3)	Sinc filter	(K × D × M) × Cp	
(4)	High-pass filter	(D × M + 5) × Cp	
(5)	Compensation filter	(10 × D × M + 12) × Cp	
(6)	Low-pass filter	(38 × D × M + 22) × Cp	

Note 1. Wakeup time from low power mode, switching time. For details, see datasheet of microphone.

Cp: PDMIFCLK period

D: PDM_CLKn’s dividend ratio

Do: PDM_CLKn’s dividend ratio (old setting)

K: Sinc filter order

M: Sinc filter decimation ratio

49.3.5 Data Input Control for Stereo Sound

The microphone which supports the stereo sound outputs the right sound synchronized with the rise-edge of PDM_CLKn and the left sound synchronized with the fall-edge of PDM_CLKn.

PDM-IF can process the right sound and the left sound using different channels.

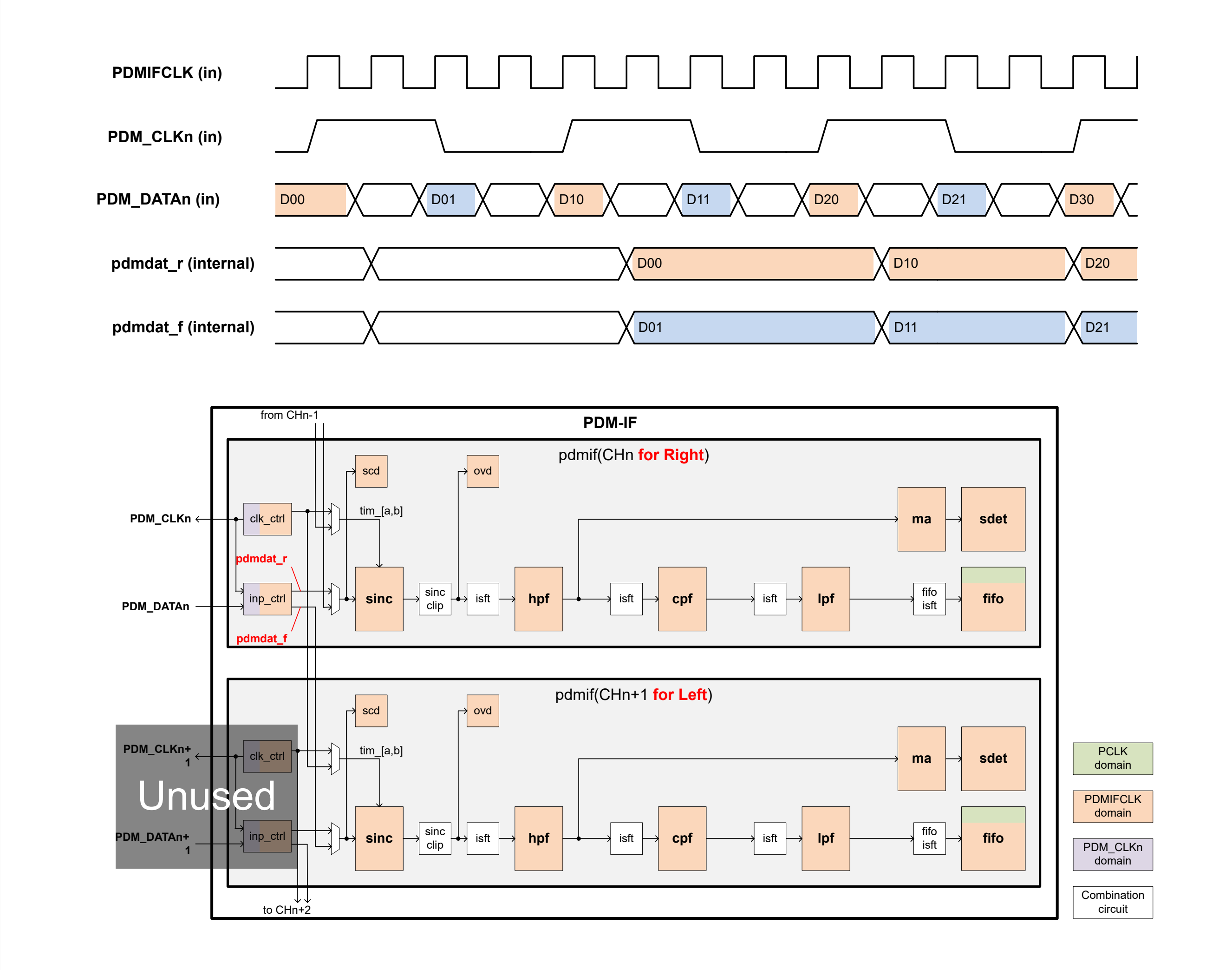


Figure 49.9 Data input control for stereo sound

Table 49.17 shows channel combinations of stereo sound processing.

Table 49.17 Channel combinations of stereo sound processing

Input	Data processing latched at rise-edge (right sound)	Data processing latched at fall-edge (left sound)
PDM_CLK0	Channel 0	Channel 1
PDM_DATA0		
PDM_CLK1	Channel 1	Channel 0
PDM_DATA1		

Table 49.18 shows an example of register settings for stereo sound processing using channel 0 and channel 1.

Table 49.18 Register settings for stereo sound processing (1 of 2)

Register	Bit	Channel 0 setting	Channel 1 setting
PDICRCHn	All bits	Channel unique settings	Channel unique settings
PDSDCRCHn	All bits	Channel unique settings	Channel unique settings

Table 49.18 Register settings for stereo sound processing (2 of 2)

Register	Bit	Channel 0 setting	Channel 1 setting
PDMDSRCHn	INPSEL	0 (use channel 0 data synchronized with rise-edge)	1 (use channel 0 data synchronized with fall-edge)
	SFMD[2:0]	Same settings for channel 0 and channel 1 (mandatory)	
	HFIS[1:0]	Channel unique settings	Channel unique settings
	HFMD	Same settings for channel 0 and channel 1 (mandatory)	
	CFIS[1:0]	Channel unique settings	Channel unique settings
	CFMD	Same settings for channel 0 and channel 1 (mandatory)	
	LFIS[1:0]	Channel unique settings	Channel unique settings
	LFMD	Same settings for channel 0 and channel 1 (mandatory)	
	SDHFMD	Same settings for channel 0 and channel 1 (mandatory)	
	SDMAMD[1:0]	Same settings for channel 0 and channel 1 (mandatory)	
	DBIS[3:0]	Channel unique settings	Channel unique settings
PDSFCRCHn	CKDIV[3:0]	Setting of the clock which is supplied to the microphone connected to channel 0	Unused
	SINCDEC[7:0]	Same settings for channel 0 and channel 1 (mandatory)	
	SINCRNG[4:0]	Same settings for channel 0 and channel 1 (mandatory)	
PDHFCS0RCHn	All bits	Channel unique settings	Channel unique settings
PDHFCK1RCHn	All bits	Channel unique settings	Channel unique settings
PDHFCH0RCHn	All bits	Channel unique settings	Channel unique settings
PDHFCH1RCHn	All bits	Channel unique settings	Channel unique settings
PDCFCHmRCHn	All bits	Channel unique settings	Channel unique settings
PDLFCH0mRCHn	All bits	Channel unique settings	Channel unique settings
PDLFCH1mRCHn	All bits	Channel unique settings	Channel unique settings
PDSDLTRCHn	All bits	Channel unique settings	Channel unique settings
PDSDUTRCHn	All bits	Channel unique settings	Channel unique settings
PDDBCRCHn	All bits	Same settings for channel 0 and channel 1 (mandatory)	
PDSCTSRCHn	All bits	Channel unique settings	Channel unique settings
PDOVLTRCHn	All bits	Channel unique settings	Channel unique settings
PDOVUTRCHn	All bits	Channel unique settings	Channel unique settings

49.3.6 Sound Detection

The operating clock of the sound detection function is PDMIFCLK. When PDSDCRCHn.SDE = 1, the function determines the sound detection using the moving average filter results.

A sound detection flag (PDSR.SDF) is set to 1 when the result is above the upper limit PDSDUTRCHn.SDETU[19:0]. The sound detection flag is also set to 1 when the result is below the lower limit PDSDLTRCHn.SDETL[19:0]. The sound detection flag is cleared to 0 when 1 is written to PDSCRCHn.SDFC.

Figure 49.10 shows the sound detection function.

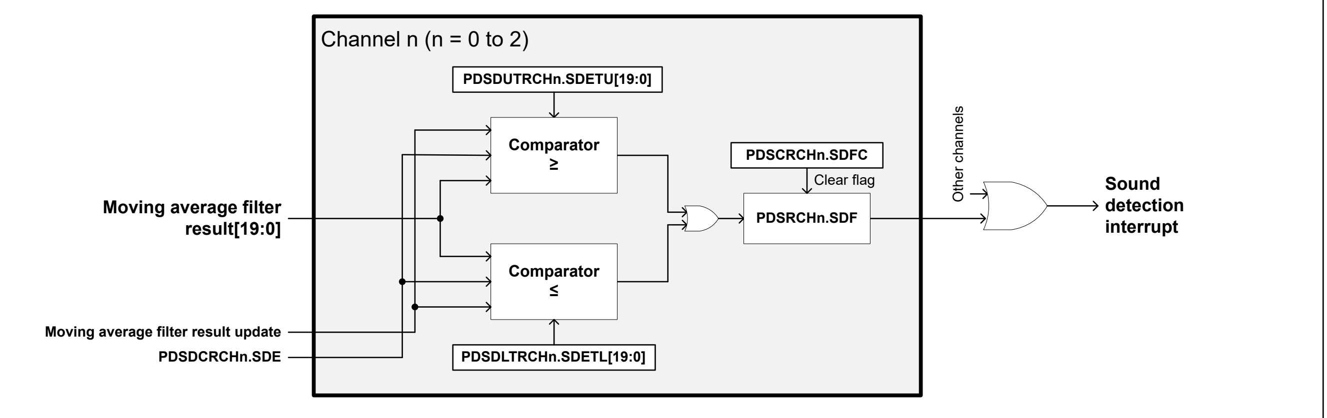


Figure 49.10 Sound detection function

49.3.7 Data Buffer

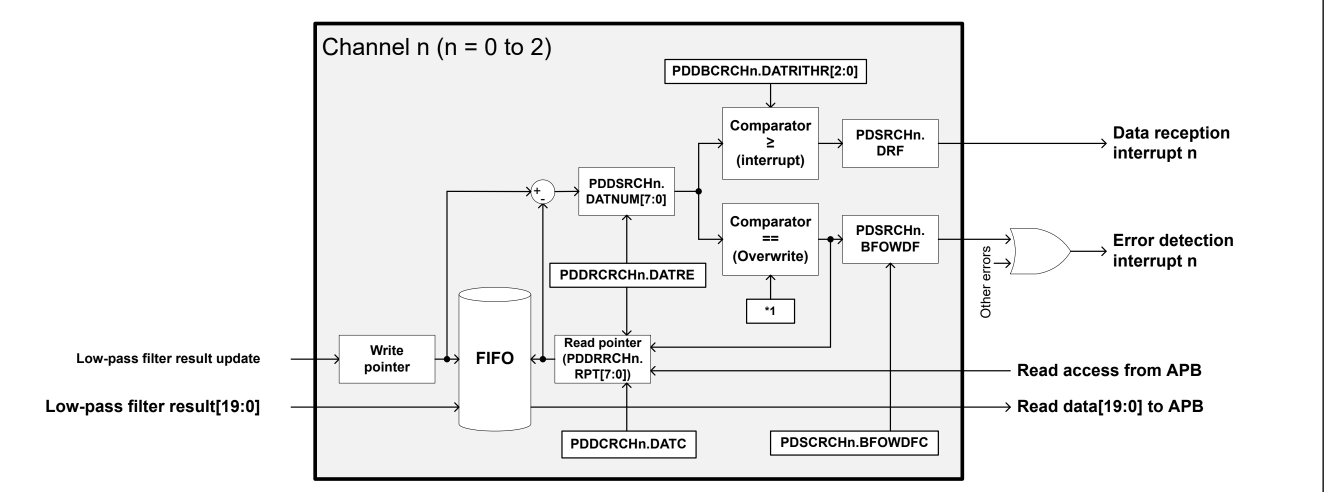
Data buffer writes sound data to FIFO synchronizing with PDMIFCLK and reads sound data from FIFO synchronizing with PCLK. Data buffer stores sound data in FIFO until it is read out.

While PDDRCRCHn.DATRE = 0 (hereinafter, DATRE), PDDSR.DATNUM[7:0] (hereinafter, DATNUM[7:0]) shows 0 and PDM-IF does not allow sound data to be read. After DATRE is changed from 0 to 1, DATNUM[7:0] shows the buffer size minus 2 and PDM-IF allows sound data to be read. While DATRE = 1, data buffer increments DATNUM[7:0] when data is written and decrements DATNUM[7:0] when data is read.

When DATNUM[7:0] is higher than or equal to a threshold which is configured by PDDBCRCHn.DATRITHR[2:0], a data reception flag PDSRCHn.DRF (hereinafter, DRF) is set to 1. When DATNUM[7:0] is lower than the threshold by reading data for instance, a data reception flag DRF is cleared to 0.

When DATNUM[7:0] is the buffer size minus 2 and data is written without reading, data buffer overwrites the oldest data with a new one and a buffer overwriting detection flag PDSRCHn.BFOWDF (hereinafter, BFOWDF) is set to 1. DATNUM[7:0] is then set to the buffer size minus 2 again after overwriting. However, data buffer overwrites but BFOWDF is not set after DATRE changes from 0 to 1 to before the first data is read.

Figure 49.11 shows the data buffer function.



Note 1. Buffer size - 1 (when buffer size is of 32 stages, this value is 31).

Figure 49.11 Data buffer function

49.3.8 Short-circuit Detection

When short-circuit detection enable (PDSDCRCHn.SCDE) is valid, the short-circuit detection can operate.

A dedicated 13-bit counter counts consecutive 0s or 1s input from pdmdat which is output by data input control block of channel n or channel n + 1.

When the number of consecutive 0s exceeds the value set in the PDSCTSRCHn.SCDL[12:0] register or the number of consecutive 1s exceeds the value set in the PDSCTSRCHn.SCDH[12:0] register, a short-circuit detection interrupt request is generated.

Figure 49.12 shows the configuration for the short-circuit detection function.

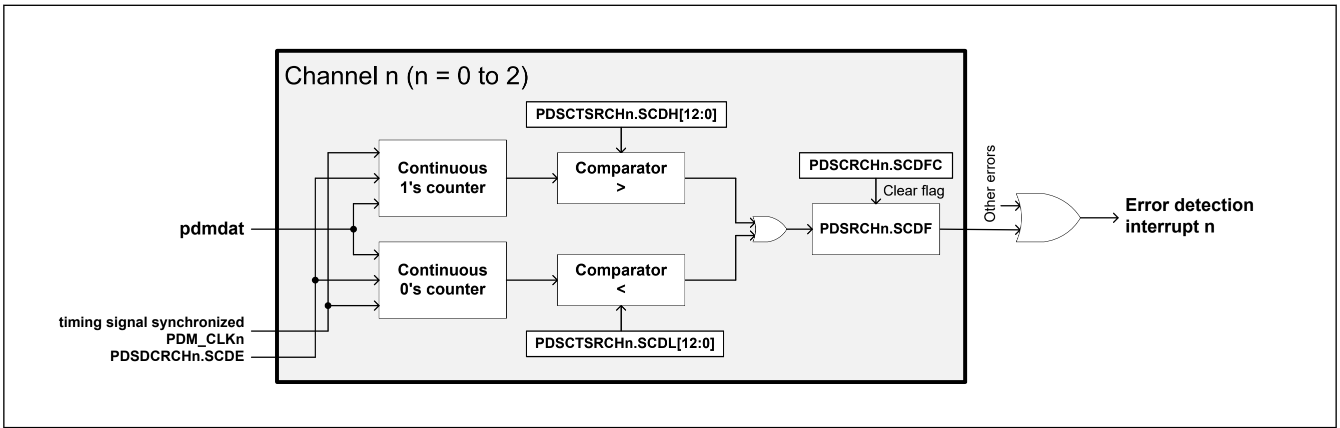


Figure 49.12 Short-circuit detection function

49.3.9 Overvoltage Detection

When the overvoltage upper limit detection enable (PDSDCRCHn.OVUDE) is valid, the overvoltage upper limit detection can operate. When the overvoltage lower limit detection enable (PDSDCRCHn.OVLDE) is valid, the overvoltage lower limit detection can operate.

The sound data is clipped as a 20-bit data from 34-bit data output from sinc filter by setting the PDSFCR.SINCRNG[4:0] bits. See Table 49.7 for detail.

An overvoltage upper limit detection flag is set to 1 when the clipped sound data is higher than PDOVUTRCHn.OVDU[19:0].

An overvoltage lower limit detection flag is set to 1 when the clipped sound data is lower than PDOVUTRCHn.OVDL[19:0].

Figure 49.13 shows the overvoltage detection function.

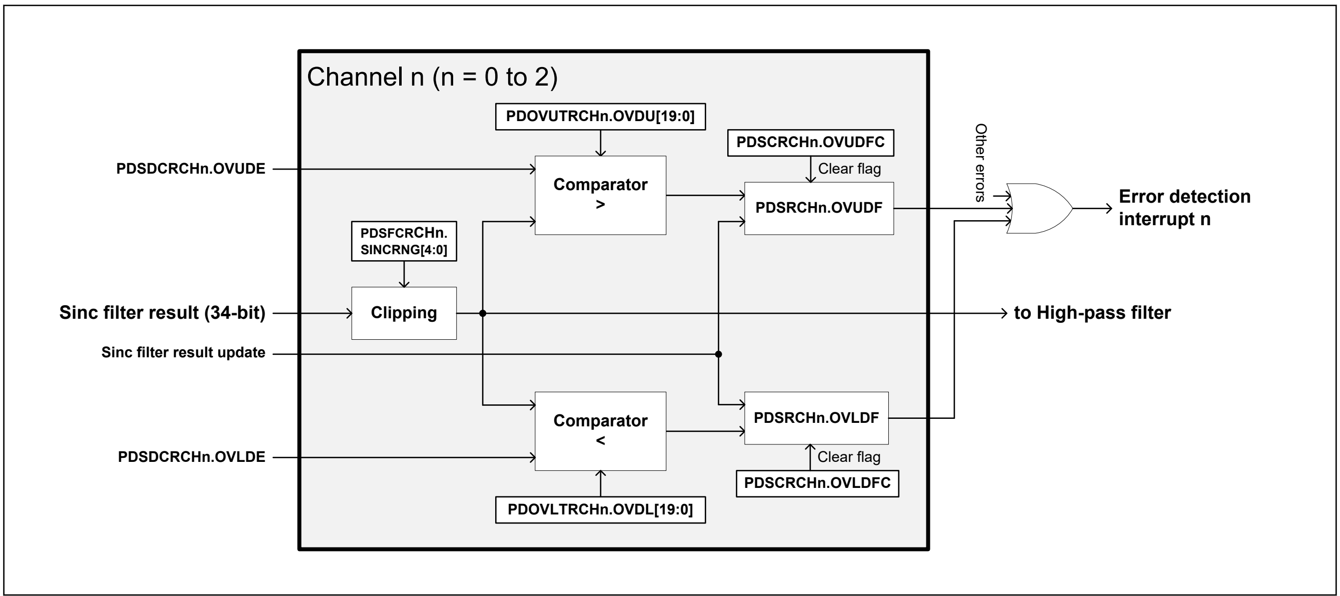


Figure 49.13 Overvoltage detection function

49.3.10 Error

Table 49.19 shows a PDM-IF error list.

Table 49.19 PDM-IF error list

No	Register name	Description	Comment	Function (Related section)
1	PDSRCHn.SCDF	Short circuit detection flag		section 49.3.8. Short-circuit Detection
2	PDSRCHn.OVLDF	Overvoltage lower limit detection flag		section 49.3.9. Overvoltage Detection
3	PDSRCHn.OVUDF	Overvoltage upper limit detection flag		section 49.3.9. Overvoltage Detection
4	PDSRCHn.BFOWDF	Buffer overwriting detection flag	For debugging	section 49.3.7. Data Buffer

No.4 errors are for debugging. Software must set registers and operate PDM-IF so that these errors do not occur. If any of these errors occurs, stop PDM-IF following section 49.4.2. Stop Flow and change the settings.

49.3.11 Interrupt Function

PDM-IF has the following interrupt function.

Table 49.20 PDM-IF interrupt list (1 of 2)

No	Signal name	Interrupt name	Number of signals	Pulse or Level	Related register bit name		Function (Related section)
					Common register	Channel register	
1	PDM_DATn (n = 0 to 2)	Data reception interrupt CHn	3	Level	PDCSR.DRFn	PDSRCHn.DRF	section 49.3.7. Data Buffer
2	PDM_SDET	Sound detection interrupt (3-channel status flags are ORed.)	1	Level	PDCSR.SDFn	PDSRCHn.SDF	section 49.3.6. Sound Detection

Table 49.20 PDM-IF interrupt list (2 of 2)

No	Signal name	Interrupt name	Number of signals	Pulse or Level	Related register bit name		Function (Related section)
					Common register	Channel register	
3	PDM_ERRn (n = 0 to 2)	Error detection interrupt CHn	3	Level	PDCSR.EDFn	PDSRCHn.SCDF	section 49.3.8. Short-circuit Detection
						PDSRCHn.OVLDF	section 49.3.9. Overvoltage Detection
						PDSRCHn.OVUDF	section 49.3.9. Overvoltage Detection
						PDSRCHn.BFOWDF	section 49.3.7. Data Buffer

49.4 Setting Flow

Figure 49.14 and Figure 49.15 show examples of the main procedure of PDM-IF.

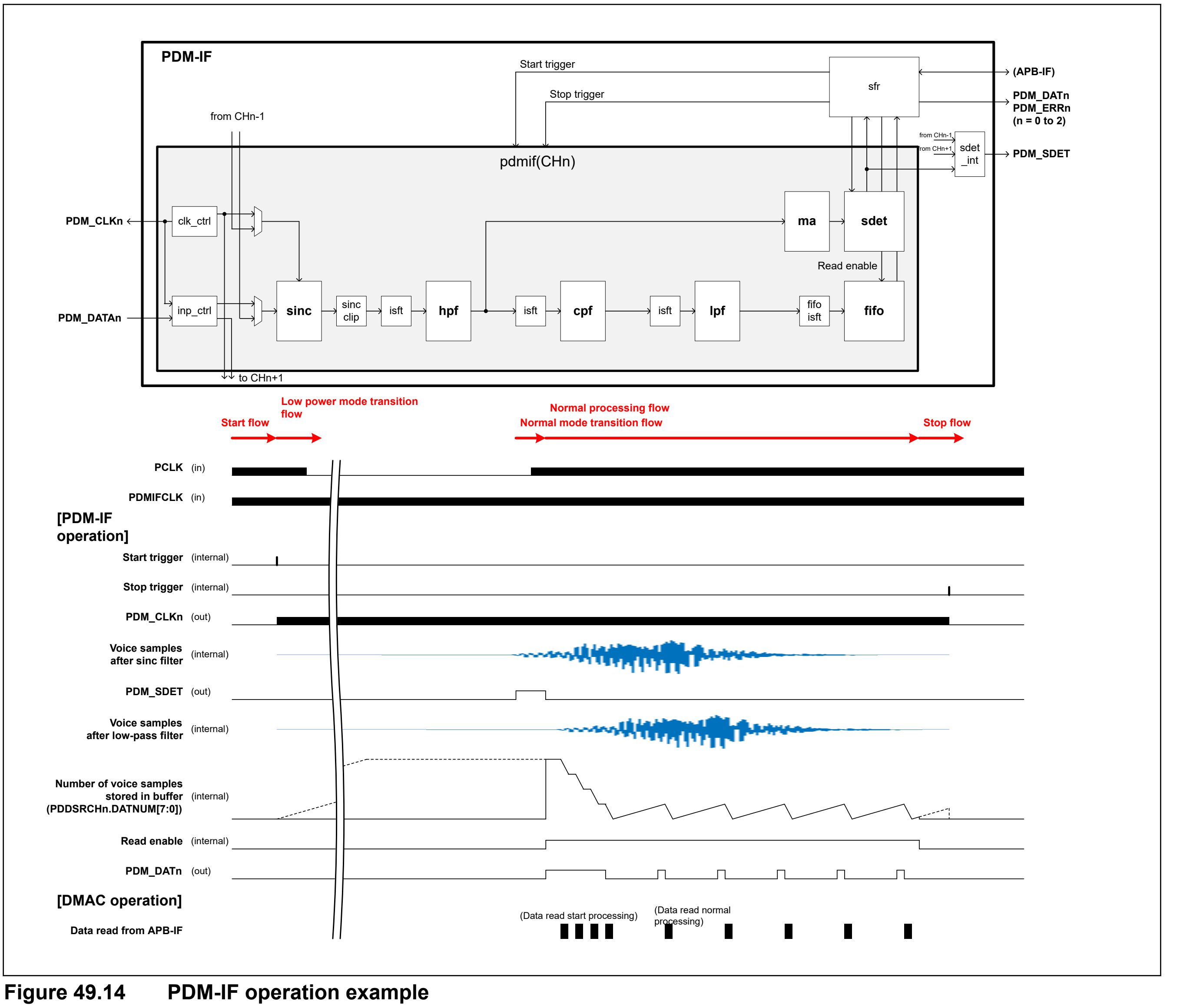


Figure 49.14 PDM-IF operation example

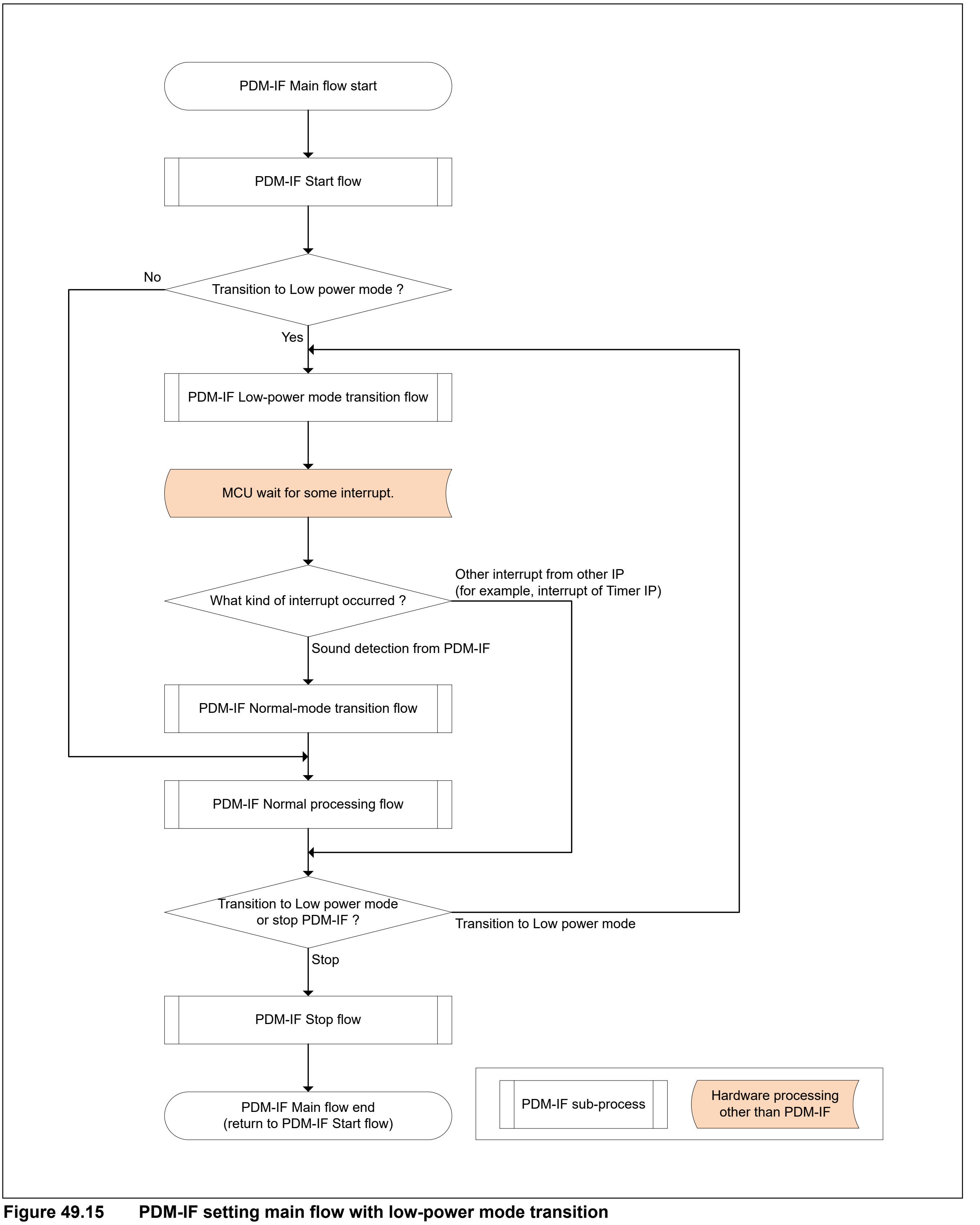


Figure 49.15 PDM-IF setting main flow with low-power mode transition

Figure 49.16 shows example of the main procedure without low-power mode transition.

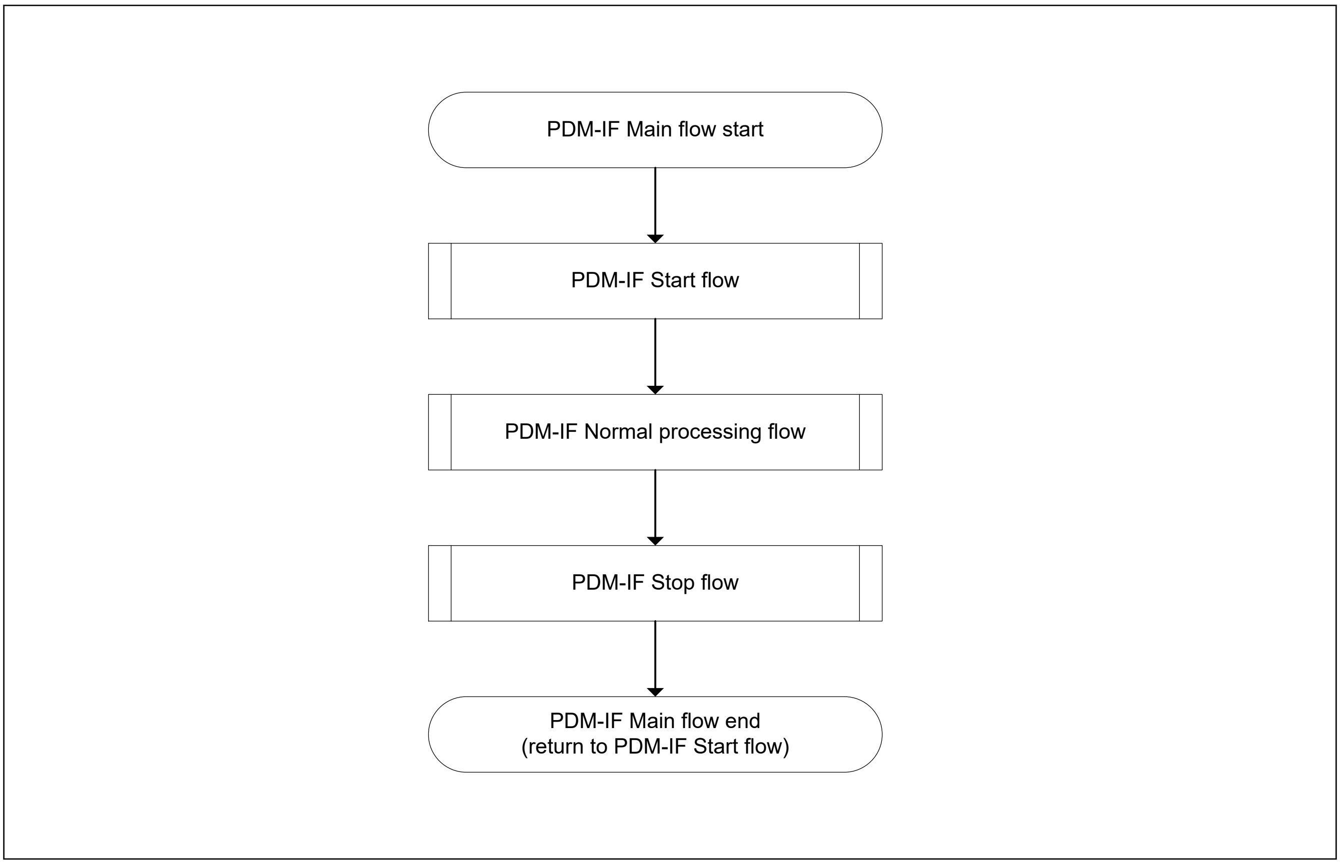
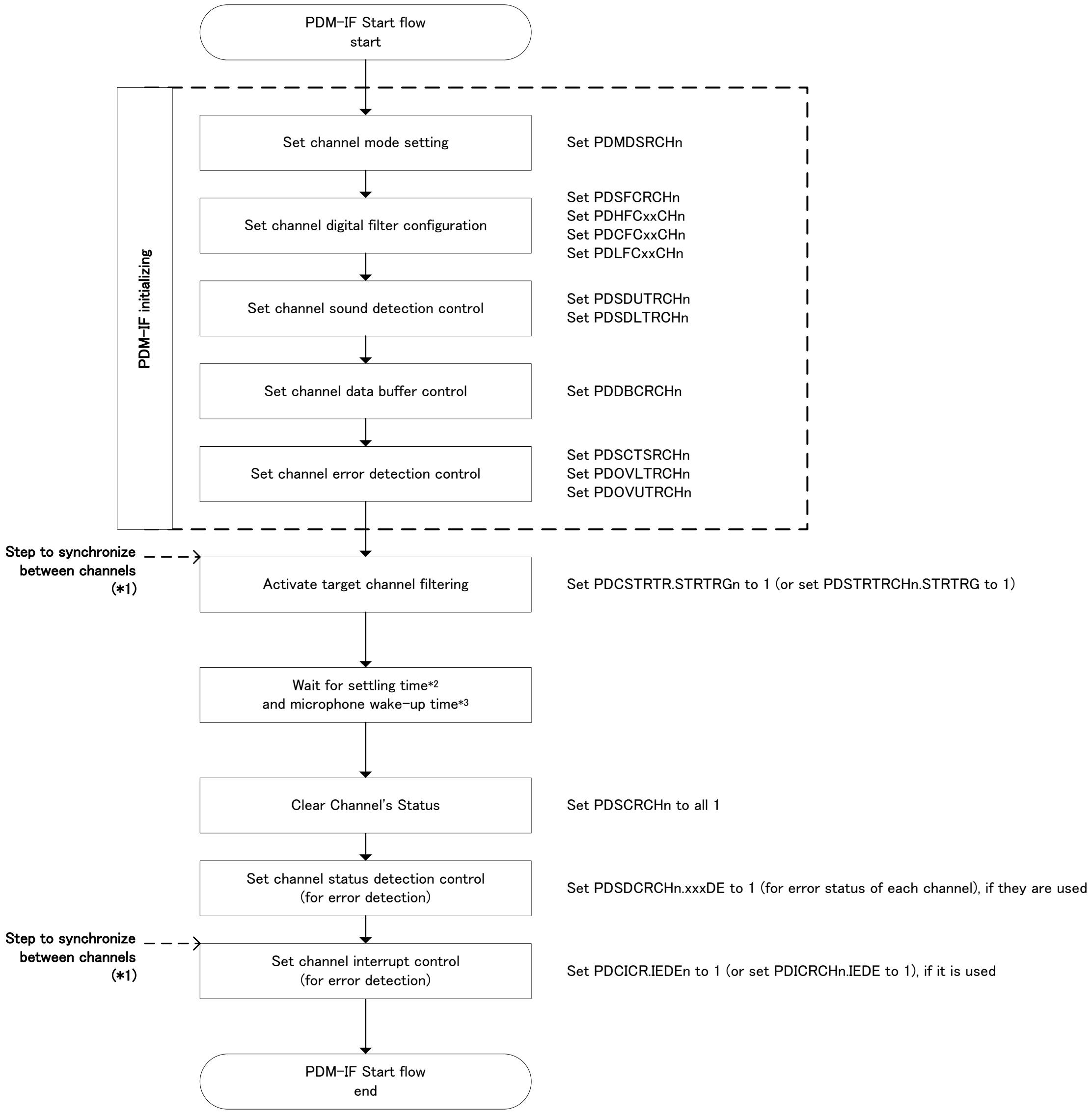


Figure 49.16 PDM-IF setting main flow without low-power mode transition

49.4.1 Start Flow

Figure 49.17 shows example of the start procedure of PDM-IF.

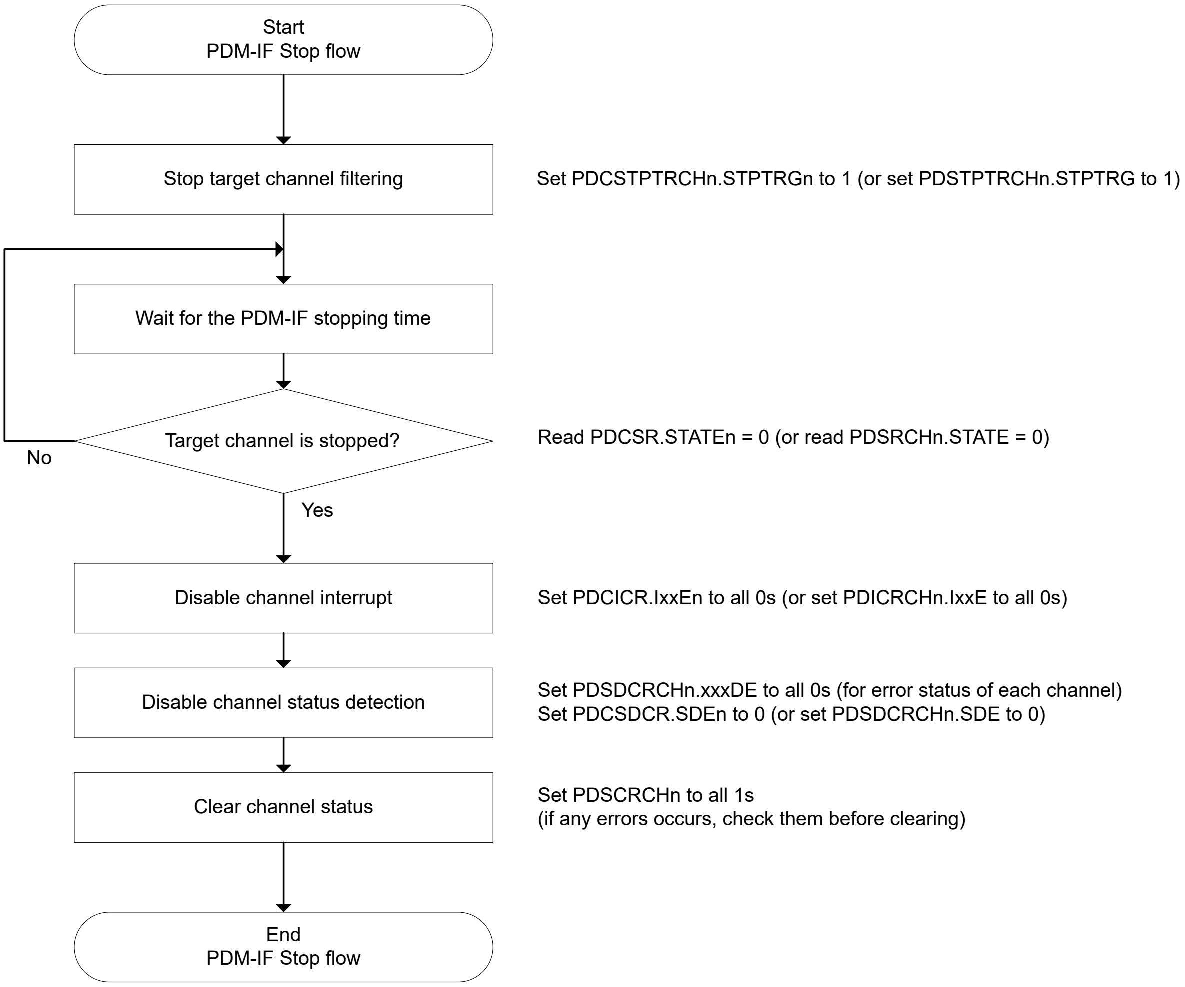


Note: xx or xxx refers to the register that matches the name pattern as similar part (ex: PDLFC) + different part (xx or xxx).
Note 1. If the channels are to be synchronized, complete the settings for each channels before this step.
Note 2. For details, see section 49.3.4.7. Settling Time of Channel Activation/Setting Change.
Note 3. Confirm the wake-up time from the connected microphone specifications.

Figure 49.17 PDM-IF start flow

49.4.2 Stop Flow

Figure 49.18 shows example of the stop procedure of PDM-IF.

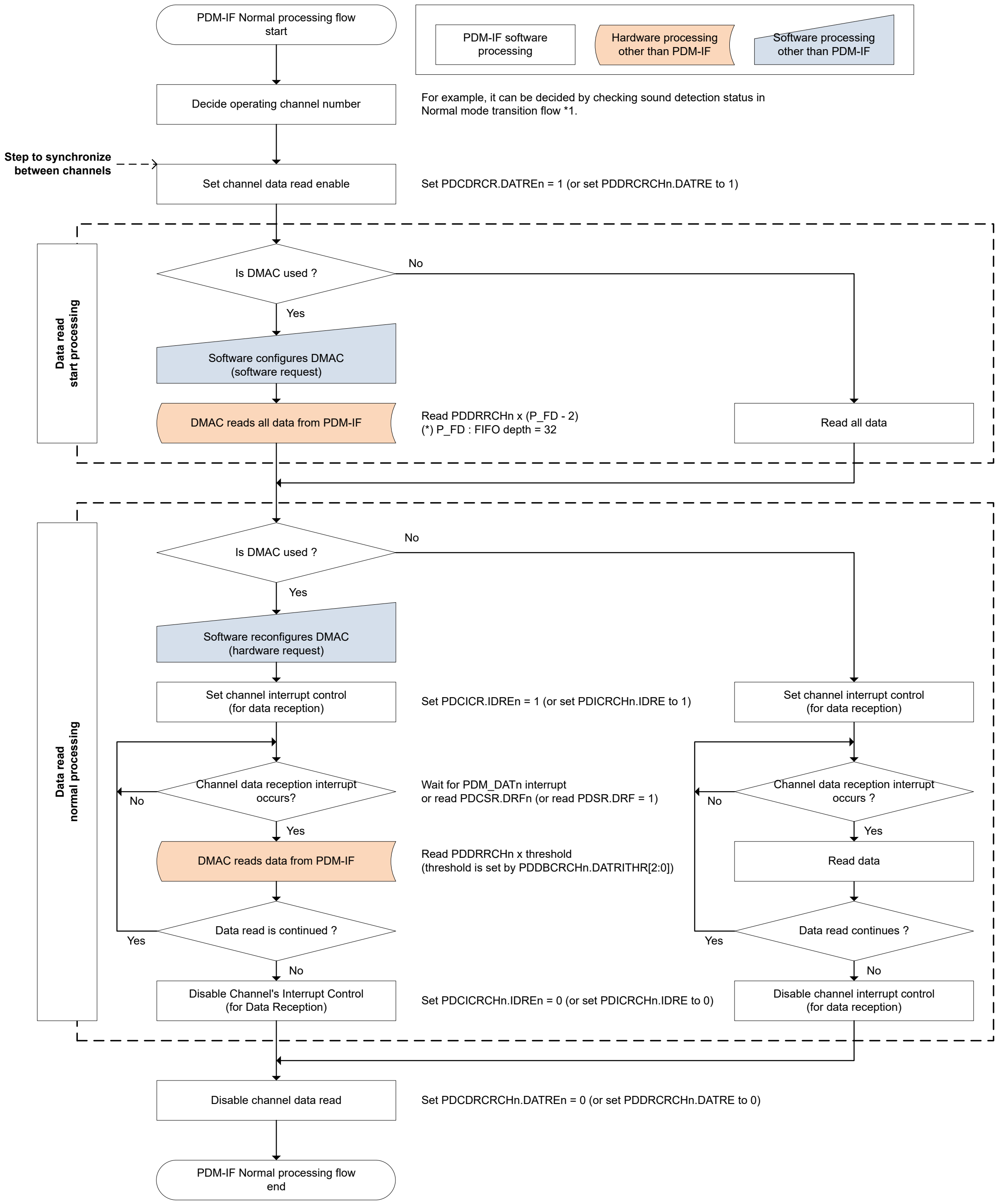


Note: xx or xxx refers to the register that matches the name pattern as different part (xx or xxx) + similar part (ex: DE).

Figure 49.18 PDM-IF stop flow

49.4.3 Normal Processing Flow

Figure 49.19 shows example of the normal processing procedure of PDM-IF.

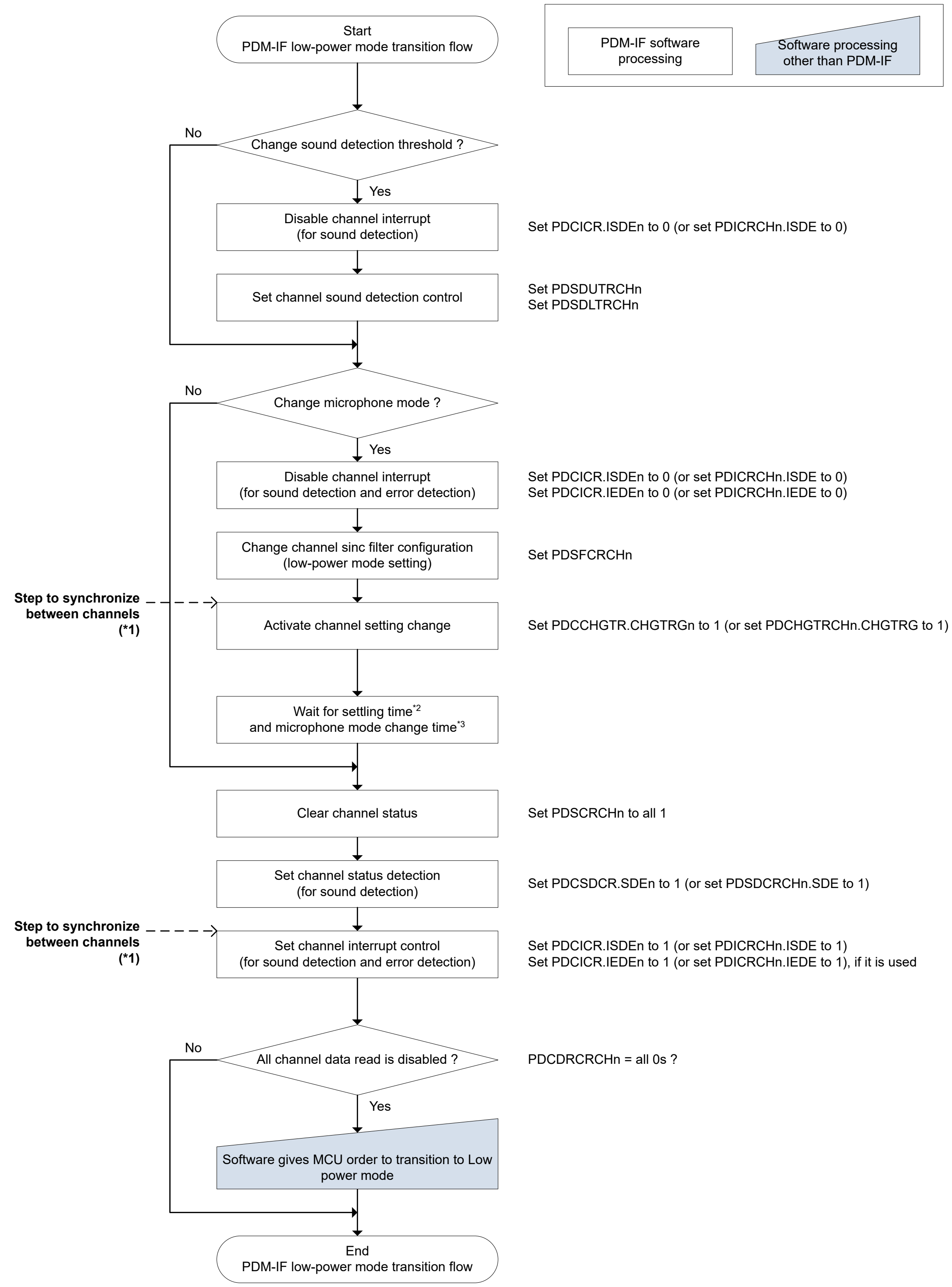


Note 1. For details, see Figure 49.21.

Figure 49.19 PDM-IF normal processing flow

49.4.4 Low-power Mode Transition Flow

Figure 49.20 shows example of the low-power mode transition procedure of PDM-IF.

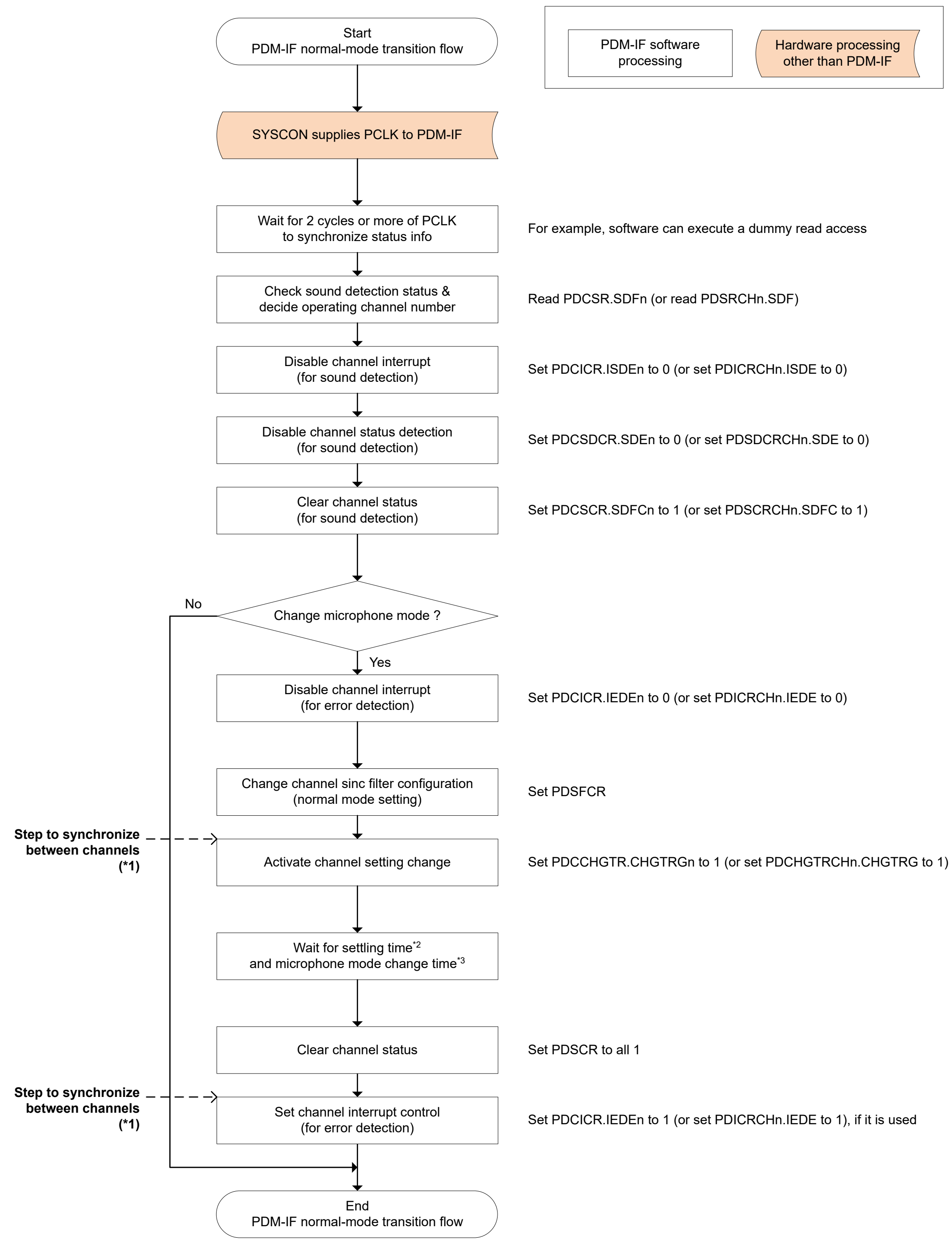


Note 1. If the channels are to be synchronized, complete the settings for each channel before this step.
Note 2. For details, see section 49.3.4.7. Settling Time of Channel Activation/Setting Change.
Note 3. Confirm the PDM microphone mode frequency and mode change time from the connected microphone specifications.

Figure 49.20 PDM-IF low-power mode transition flow

49.4.5 Normal Mode Transition Flow

Figure 49.21 shows example of normal-mode transition procedure of PDM-IF.



Note 1. If the channels are to be synchronized, complete the settings for each channel before this step.
Note 2. For details, see section 49.3.4.7. Settling Time of Channel Activation/Setting Change.
Note 3. Confirm the PDM microphone mode frequency and mode change time from the connected microphone specifications.

Figure 49.21 PDM-IF normal-mode transition flow

49.5 Usage Notes

49.5.1 Data Reception Interrupt Threshold Register Setting

A data reception interrupt PDM_DATn is issued as a level output. Therefore, if the number of samples in buffer still exceeds the data reception interrupt threshold set in PDDBCRCHn.DATRITHR[2:0] when data reading of DMA finishes, the interrupt controller might not detect the edge of PDM_DATn. The situation happens if data reading time exceeds sampling rate*1 × threshold. Set the appropriate threshold so that the interrupt controller can detect the interrupt.

In case the threshold cannot avoid the situation, read PDDBCRCHn.DATRITHR[2:0] and restart DMA by software after reading the data.

Note 1. Sampling rate: Time to spend to store a sample in buffer.

49.5.2 Frequency Characteristics of Filters with Default Settings

Figure 49.22 shows the frequency characteristics of the filters with the default settings for reference information.

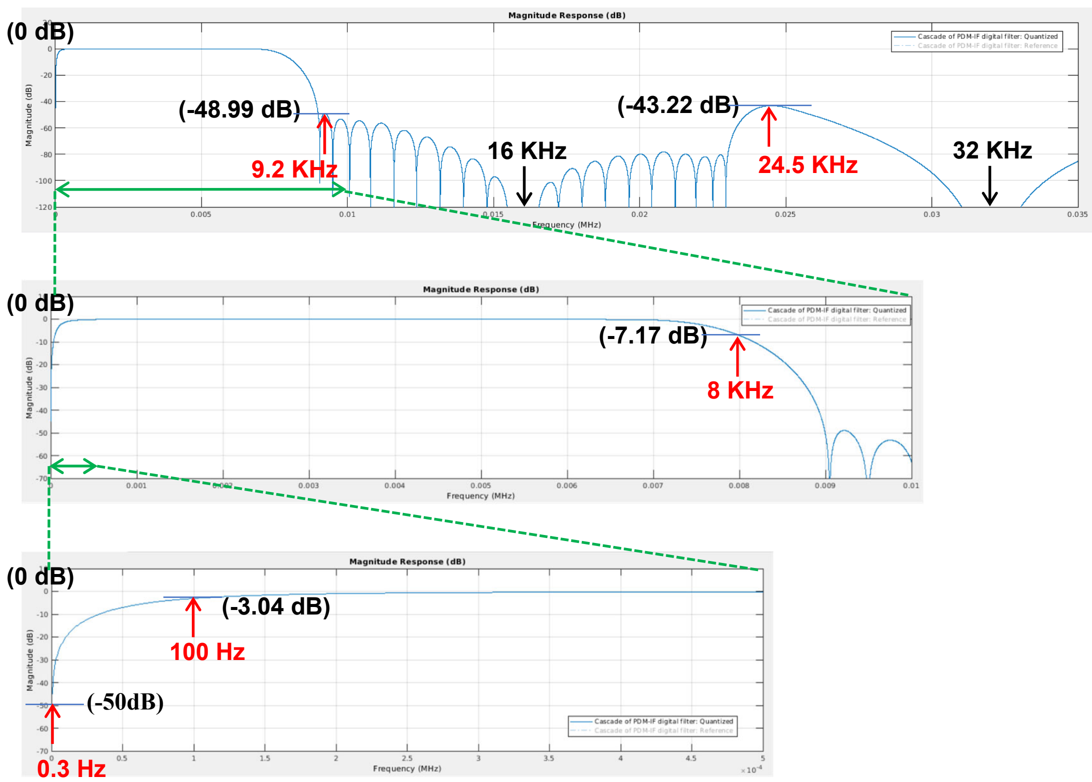


Figure 49.22 Frequency characteristics of filters with default settings

Table 49.21 Frequency characteristics of filters with default settings (1 of 2)

Frequency (Hz)		Magnitude (dB)
Cut-off frequency of LPF	24.5 kHz	-43.22
	…	…
	9.2 kHz	-48.99
	8 kHz	-7.17

Table 49.21 Frequency characteristics of filters with default settings (2 of 2)

Frequency (Hz)		Magnitude (dB)
Passband	7 kHz	-0.39
	6 kHz	-0.07
	…	…
	150 Hz	-1.63
	100 Hz	-3.04
Cut-off frequency of HPF	50 Hz	-7.02
	0.3 Hz	-50.00

50. Boundary Scan

50.1 Overview

The boundary scan function provides a serial I/O interface based on the JTAG (Joint Test Action Group), IEEE Std.1149.1, and IEEE Standard Test Access Port and Boundary Scan Architecture. Table 50.1 lists the boundary scan specifications, Figure 50.1 shows a block diagram, and Table 50.2 lists the I/O pins.

Table 50.1 Boundary scan specifications

Parameter	Specifications
Execution condition	Boundary scan must be executed when the RES pin is driven low.
Test modes	● BYPASS mode ● EXTEST mode ● SAMPLE/PRELOAD mode ● CLAMP mode ● HIGHZ mode ● IDCODE mode

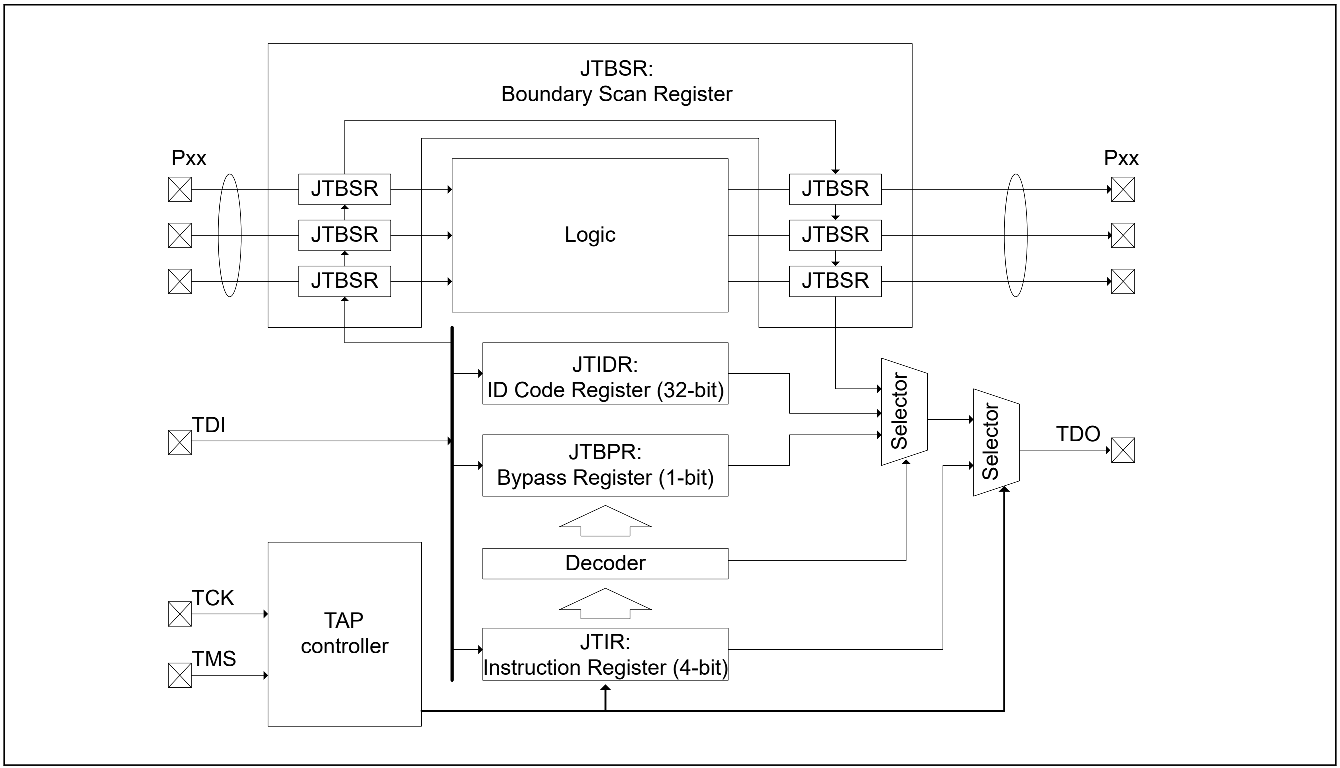


Figure 50.1 Boundary scan function block diagram

Table 50.2 Boundary scan I/O pins

Pin name	I/O	Description
TCK	Input	Test clock input pin Clock signal for boundary scan. The input clock duty cycle is 50% when the boundary scan function is used.
TMS	Input	Test mode select pin
TDI	Input	Test data input pin
TDO	Output	Test data output pin

Note: This device does not support the TRST pin for the JTAG interface.

50.2 Register Descriptions

Table 50.3 lists the boundary scan registers.

Table 50.3 Boundary scan registers

Register name	Symbol	Value after reset
Instruction Register	JTIR	0xE
ID Code Register	JTIDR	0x085D_A447
Bypass Register	JTBPR	Undefined
Boundary Scan Register	JTBSR	Undefined

Usage notes for the boundary scan registers:

- Instructions can be input to the Instruction Register (JTIR) through the TDI pin by serial transfer.
- The Bypass Register (JTBPR), which is a 1-bit register, is connected between the TDI and TDO pins in BYPASS mode.
- The Boundary Scan Register (JTBSR), which is configured according to the BSDL description, is connected between the TDI and TDO pins when test data is being shifted in.

Table 50.4 shows the availability of serial transfer for the registers.

Table 50.4 Serial transfer for registers

Register name	Serial input	Serial output
Instruction Register (JTIR)	Available	Available
ID Code Register (JTIDR)	Available	Available
Bypass Register (JTBPR)	Available	Available
Boundary Scan Register (JTBSR)	Available	Available

50.2.1 JTIR : Instruction Register

Bit position:	3	2	1	0
Bit field:	TS[3:0]			
Value after reset:	1	1	1	0

<table>
<tr><th>Bit</th><th>Symbol</th><th>Function</th><th>R/W</th></tr>
<tr><td>3:0</td><td>TS[3:0]</td><td>Test Bit Set
The command configuration for these bits
<table>
<tr><th>TS[3:0]</th><th>Instruction</th></tr>
<tr><td>0x0</td><td>EXTEST</td></tr>
<tr><td>0x1</td><td>SAMPLE/PRELOAD</td></tr>
<tr><td>0x3</td><td>IDCODE (Renesas code)</td></tr>
<tr><td>0x5</td><td>CLAMP</td></tr>
<tr><td>0x6</td><td>HIGHZ</td></tr>
<tr><td>0xF</td><td>BYPASS</td></tr>
<tr><td>Others</td><td>Reserved</td></tr>
</table></td><td>—</td></tr>
</table>

JTAG instructions can be transferred to the JTIR register by serial input from the TDI pin. The JTIR register is initialized when a power-on reset occurs, or when the TAP controller is in the Test-Logic-Reset state.

50.2.2 JTIDR : ID Code Register

Bit position: 31 … 0

Bit field: DID[31:0]

Value after reset: 0 0 0 0 1 0 0 0 0 1 0 1 1 1 0 1 1 0 1 0 0 1 0 0 0 1 0 0 0 1 1 1

Bit	Symbol	Function	R/W
31:0	DID[31:0]	Device ID These bits store the fixed value that indicates the device IDCODE (0x085D_A447).	—

The JTIDR register data is output from the TDO pin when the IDCODE instruction is executed. After a reset release, the DID[31:0] of JTIDR changes into the Arm® debug code. See the *Arm® CoreSight™ SoC-400 Technical Reference Manual* (ARM DDI 0480F).

50.2.3 JTBPR : Bypass Register

The JTBPR register is a 1-bit register and is connected between the TDI and TDO pins when the JTIR register is set to BYPASS mode. The JTBPR register cannot be read from or written to by the CPU.

50.2.4 JTBSR : Boundary Scan Register

The JTBSR register is a shift register for controlling the external input and output pins of this device, and is distributed across the pads. To apply the JTBSR register in boundary-scan testing, issue the EXTEST, SAMPLE/PRELOAD, CLAMP and HIGHZ instructions. The BSDL file describes the associations between the JTBSR register bits and the pins of this device. The value after reset is undefined.

50.3 Operation

During a reset, the JTAG ports, TCK, TMS, TDI and TDO are assigned as default pin functions. The TCK, TMS and TDI pins are pulled up by the pull-up resistors. Boundary scan testing can be executed after the setup time elapses when POR is negated and RES is driven low.

50.3.1 TAP Controller

Figure 50.2 shows the state transition diagram of the TAP controller. All transitions are controlled by the TMS signal.

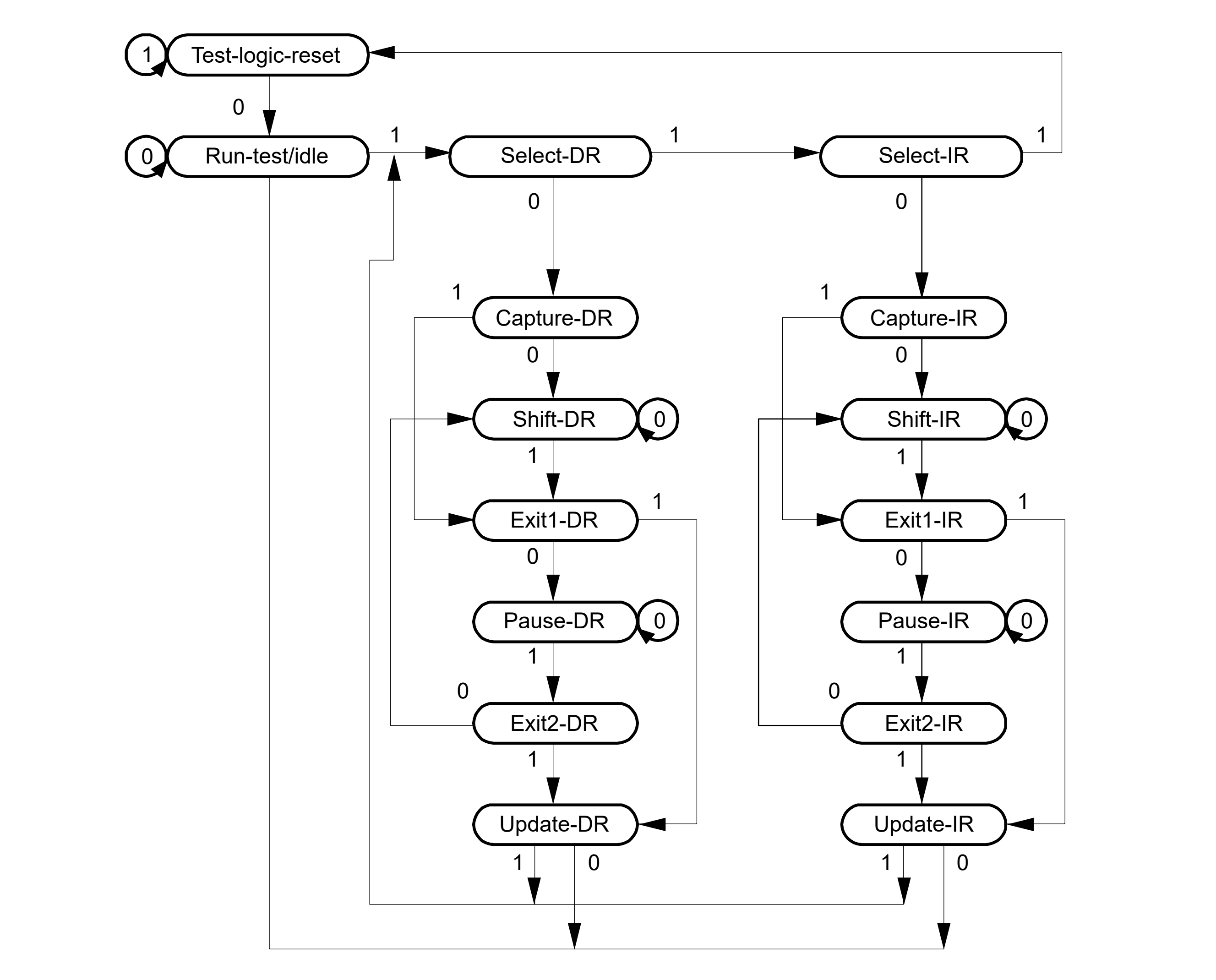


Figure 50.2 State transition diagram of TAP controller

50.3.2 Commands

(1) BYPASS

The BYPASS instruction drives the Bypass Register (JTBPR). This instruction shortens the shift path, facilitating the transfer of serial data to other LSIs on a printed circuit board at higher speeds. While this instruction is being executed, the test circuit has no effect on the system circuits.

The JTBPR register is connected between the TDI and TDO pins. Bypass operation is initiated from the Shift-DR operation. The TDO is low in the first clock cycle in the Shift-DR state. In the subsequent clock cycles, values input to the TDI pin are output from the TDO pin.

(2) EXTEST

The EXTEST instruction is used to test external circuits when this device is installed on the printed circuit board. If this instruction is executed, output pins are used to output test data (specified in the SAMPLE/PRELOAD instruction) from the Boundary Scan Register (JTBSR) to the other devices, and input pins are used to input the test result.

(3) SAMPLE/PRELOAD

The SAMPLE/PRELOAD instruction is used to input data from the internal circuits of this device to the JTBSR register, output data from the scan path, and reload the data to the scan path. While this instruction is executed, input signals are directly input to this device and output signals are also directly output to the external circuits. This device system circuit is not affected by this instruction.

In SAMPLE operation, the JTBSR register latches a snapshot of the data transferred from the input pins to the internal circuit or data transferred from the internal circuit to the output pins. The latched data is read from the scan path. The JTBSR register latches the data snapshot on the rising edge of the TCK pin in the Capture-DR state. The data snapshot is only transferred from the internal circuit to the output pins during a reset.

In PRELOAD operation, the initial value is written from the scan path to the parallel output latch of the JTBSR register prior to the EXTEST instruction execution. If EXTEST is executed without executing this PRELOAD operation, undefined values are output from the beginning to the end (transfer to the output latch) of the EXTEST sequence. (In the EXTEST instruction, output parallel latches are always output to the output pins.)

(4) IDCODE

When the IDCODE instruction is selected, the ID Code Register (JTIDR) value is output to the TDO pin in the Shift-DR state of the TAP controller. In this case, the JTIDR register value is output LSB-first. During this instruction execution, the test circuit does not affect the system circuit.

(5) CLAMP

When the CLAMP instruction is selected, output pins output the JTBSR register value that was specified in the SAMPLE/PRELOAD instruction in advance. While the CLAMP instruction is selected, the status of the JTBSR register is maintained regardless of the TAP controller state.

The JTBPR register is connected between the TDI and TDO pins, leading to the same operation as when the BYPASS instruction is selected.

(6) HIGHZ

When the HIGHZ instruction is selected, all output pins enter high-impedance state. While the HIGHZ instruction is selected, the JTBSR register is maintained regardless of the state of the TAP controller.

The JTBPR register is connected between the TDI and TDO pins, leading to the same operation as when the BYPASS instruction is selected.

50.4 Usage Notes

The boundary scan function is subject to the following constraints:

- The boundary scan must be executed when the RES pin is driven low
- Serial data input/output is in LSB order, as shown in Figure 50.3

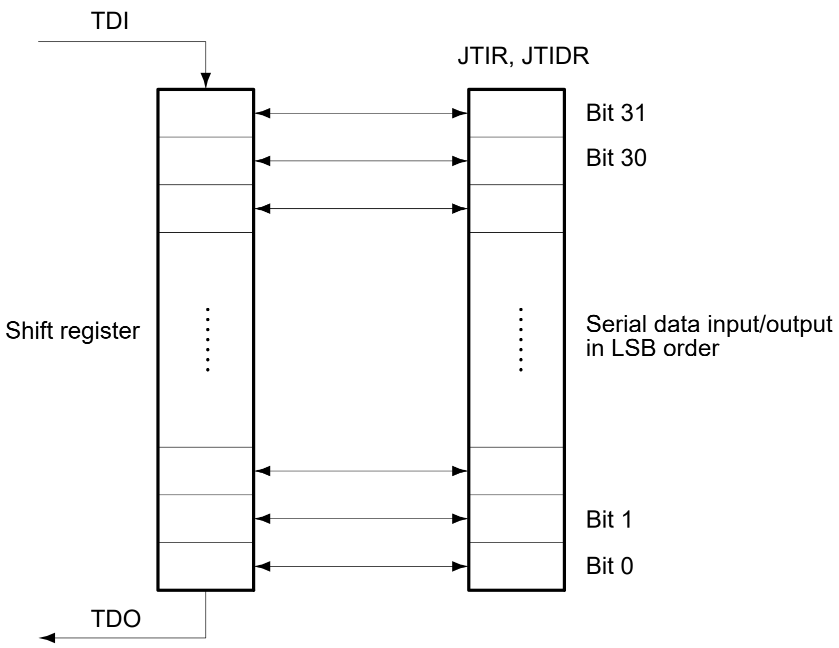


Figure 50.3 Serial data input/output

The following pins cannot be boundary-scanned for Standard Product and SiP Product:

- Power supply pins (VCC, VCC2, VCC_DCDC, VCL, VSS, VSS_DCDC, VBATT, AVCC0, AVSS0, VCC_USB, VSS_USB, VSS1_USBHS, VSS2_USBHS, VCC_USBHS, AVCC_USBHS, VSS1_USBHS, VSS2_USBHS)
- Analog reference pins (VREFH0, VREFL0, VREFH, VREFL)
- Clock pins (EXTAL, XTAL, XCIN, and XCOUT)
- Reset pin (RES)
- USBHS-dedicated pins (USBHS_DP, USBHS_DM, USBHS_RREF)
- USBFS pin (USB_DP, USB_DM)
- The boundary-scan pins (TCK, TMS, TDI, and TDO).
- The switching regulator pin (VLO)

The following pins cannot be boundary-scanned for SiP Product:

- SiP flash pins (OM_1_SCLK, OM_1_CS1, OM_1_DQS, OM_1_SIO7-0, OM1_RESET and OM_1_ECSINT1)

51. Security Features

51.1 Features

- Armv8-M TrustZone security
  - Implementation Defined Attribution Unit (IDAU) is implemented
  - Security Attribution Unit (SAU) is implemented
    - 8 regions
  - Master Security Attribution Unit (MSAU) is implemented (IDAU for master other than CPU)
  - Code MRAM
    - Up to 2 regions (secure/non-secure)
  - SiP flash (Only for SiP product)
    - Up to 2 regions (secure/non-secure)
  - CM33 TCM
    - Up to 2 regions (secure/non-secure) for C-TCM and S-TCM
  - SRAM
    - Up to 2 regions (secure/non-secure) per SRAM
  - VBATT backup registers
    - Up to 2 regions (secure/non-secure)
  - Peripheral
    - Security attributes can be set individually for each unit/channel
  - CSC*1
    - CSC region is defined as non-secure
  - SDRAM*1
    - SDRAM region is defined as non-secure
  - OPSI0*1
    - OPSI0 region is defined as non-secure
  - OPSI1*1 (only for Standard product)
    - OPSI1 region is defined as non-secure
- Privileged control
  - Access permissions for memory except VBATT backup register are controlled by MPU managed by the privileged code
  - Privilege attributes of VBATT backup register are controlled by the registers of each controller
  - Individual privileged or Unprivileged attribution for each peripheral.
- Device lifecycle management
- Three debug levels
  - AL2: Non-secure and secure debug functions are enabled and accessible from the debugger
  - AL1: Only non-secure debug functions is enabled, and the debugger can access only defined non-secure debug accessible regions
  - AL0: No debug functions are available.
- Key injection
- Secure factory programming
  - Supports image programming in ciphertext format in an untrusted factory

- Secure boot
  - Support up to four revocable Root of Trust (RoT) in OTP, which is the SHA-256 hash digest of the RoT public key.
  - Verify the integrity and authenticity of image to be programmed
  - Verify the integrity and authenticity of the executable image by the immutable (OTP) First Stage Bootloader (FSBL) prior to executing the image.
- Cryptographic accelerator
  - See section 52, Renesas Secure IP (RSIP-E50D).
- Secure pin multiplexing
  - All I/O port pins can be configured individually as secure or non-secure
  - Peripheral pin function is valid when the security attribute of peripheral and I/O port match. See section 20, I/O Ports.
- Decryption on the fly
  - Decryption on the fly (DOTF) decrypts the encrypted content stored in the external memory in real time. See section 45, Decryption On The Fly (DOTF).

Note 1. There is no TrustZone filter in the external RAM and external device area. Therefore, even if these regions are marked as secure in the SAU settings, non-secure masters except the CPU can access them.

51.2 Tamper Detection

The device is provided with I/O ports capable of detecting an external tamper attempt. In the case of a tamper event:

- Timestamping — The RTC can take a timestamp to log the event, and an interrupt can be generated. See section 26, Realtime Clock (RTC).
- Hardware Unique Key — Hardware Unique Key (HUK) is zeroized (set to zero) following detection of a tamper event. See section 59, MRAM.
- VBATT backup registers — VBATT backup registers are zeroized (set to zero) following detection of a tamper event. See section 12, Battery Backup Function.

51.3 Arm Security Features

51.3.1 Arm TrustZone Technology

Arm TrustZone technology divides the system and the application into secure and non-secure domains. A secure application can issue both secure and non-secure transactions, but a non-secure application can only issue non-secure transactions. Secure transactions can only access secure memory and resources, and non-secure transactions can only access non-secure memory and resources. Secure transactions can be issued only using secure region addresses and non-secure transactions can be issued only using non-secure region addresses. Figure 51.1shows the security attributes of transactions that can be issued by each master.

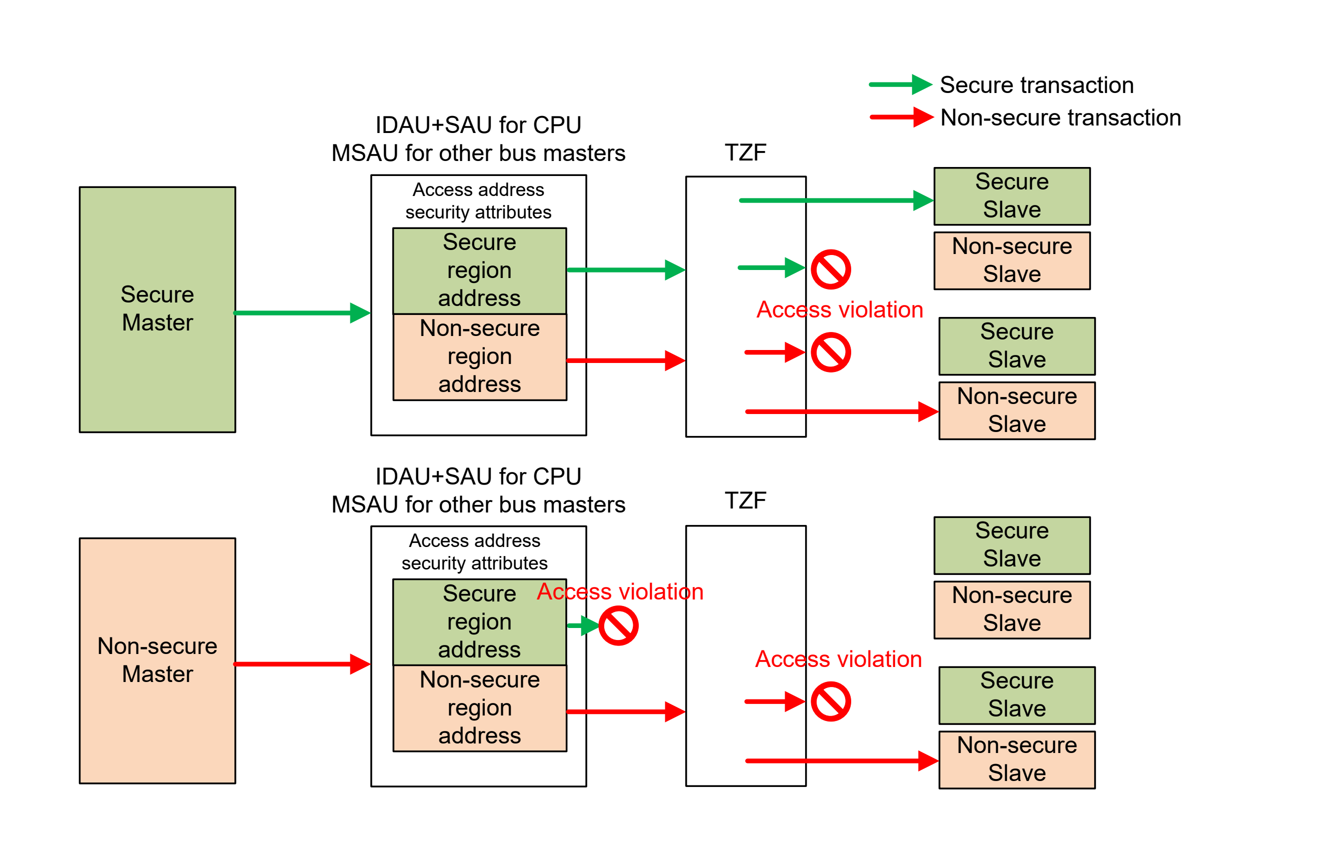


Figure 51.1 Transactions that can be issued by each master

For more details, see the *Arm® v8-M Architecture Reference Manual* and *Arm® Platform Security Architecture Trusted Base System Architecture for Arm®v6-M, Arm®v7-M and Arm®v8-M*.

51.3.2 Privileged Control

Systems and applications are divided into privileged and unprivileged domains. The CPU can limit or exclude access to some resources by executing code in privileged or unprivileged mode. Privileged mode can access both privileged and unprivileged domains, but unprivileged mode can access only the unprivileged domain.

For more details, see the *Arm® v8-M Architecture Reference Manual* and *Arm® Platform Security Architecture Trusted Base System Architecture for Arm®v6-M, Arm®v7-M and Arm®v8-M* .

51.3.3 Security Attribution

The TrustZone for Armv8-M implementation consists of the Security Attribution Unit (SAU) and Implementation Defined Attribution Unit (IDAU). The 4GB memory space is partitioned into Secure (S) and Non-Secure (NS) memory regions. The Secure memory space is further divided into two types, Non-secure Callable (NSC) and Secure.

Note:
- S = Secure addresses are used for memory and peripherals that are only accessible by secure software or secure masters.
- NSC = A special type of secure location. This type of memory is the only type in which an Armv8-M processor permits to hold a Secure Gateway (SG) instruction that enables software to transition from Non-secure to Secure state.
- NS = Non-secure addresses are used for memory and peripherals accessible by all software running on the device.

51.3.3.1 Implementation Defined Attribution Unit (IDAU)

The IDAU defines the code, SRAM and peripheral region into the secure alias region and the non-secure alias region by the address bit [28]. The secure code region and secure SRAM region are assigned the NSC security attributes. The security map defined by IDAU is fixed in hardware and cannot be changed by software.

51.3.3.2 Master Security Attribution Unit (MSAU)

MSAU is IDAU that defines system-specific security address map for masters other than CPU. The MSAU defines secure and non-secure alias regions but does not define the Non-secure Callable (NSC) and region number. Masters other than the CPU can issue a security transaction using the secure alias address defined by MSAU. However, non-secure masters are prohibited from issuing secure transactions using an address in the secure alias region. Figure 51.2 shows the defined security map of IDAU and MSAU. The security map defined by MSAU is fixed in hardware and cannot be changed by software.

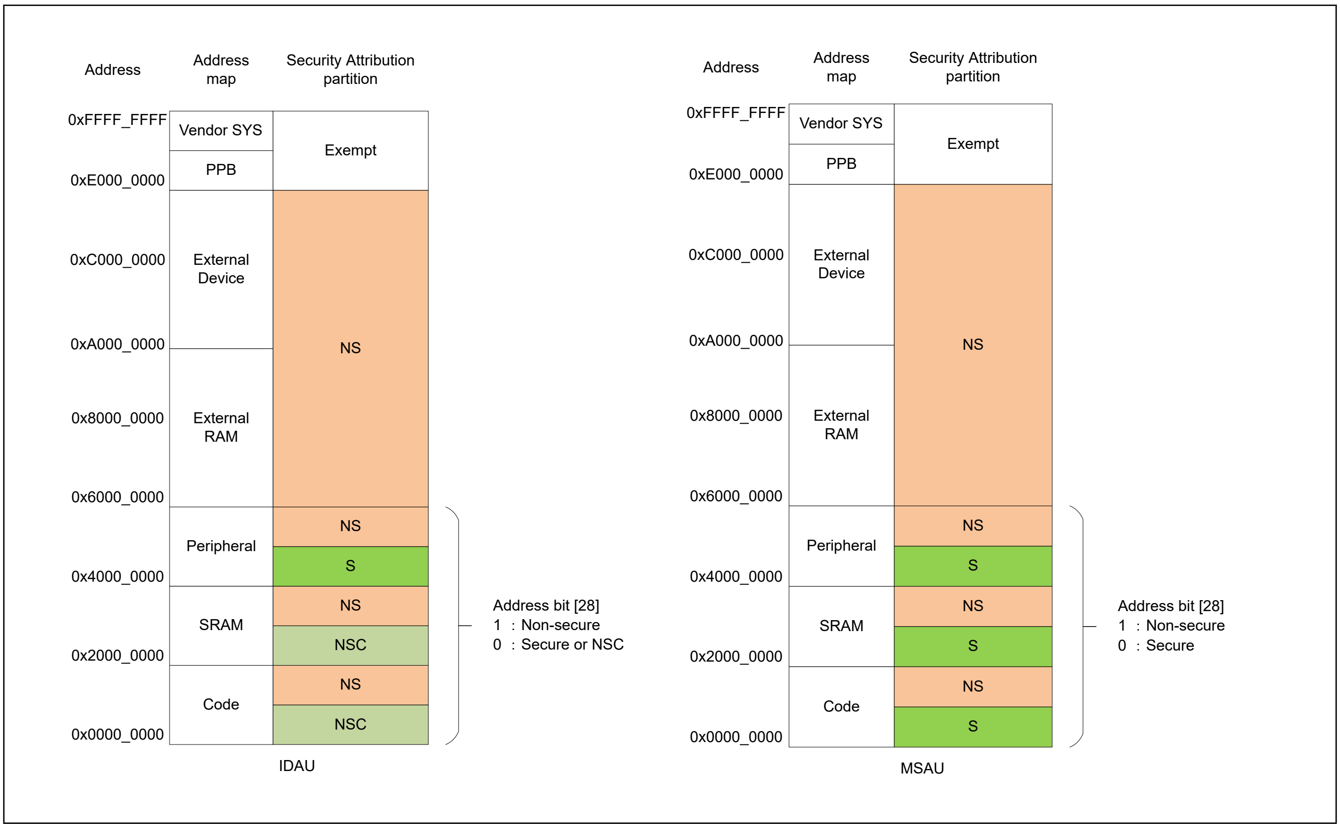


Figure 51.2 IDAU and MSAU defined security map

51.3.3.3 Secure Attribution Unit (SAU)

The SAU is a programmable unit that determines the security of an address. The SAU is programmable in the Secure state and has a programmers' model similar to the MPU. If an address maps to regions defined by both IDAU and SAU, the region of the highest security level is selected. Secure master can issue secure and non-secure transactions using the address of each security alias region. Non-secure master cannot issue secure transactions using the address of the secure alias region. Figure 51.3 shows the final determination of the address map security level.

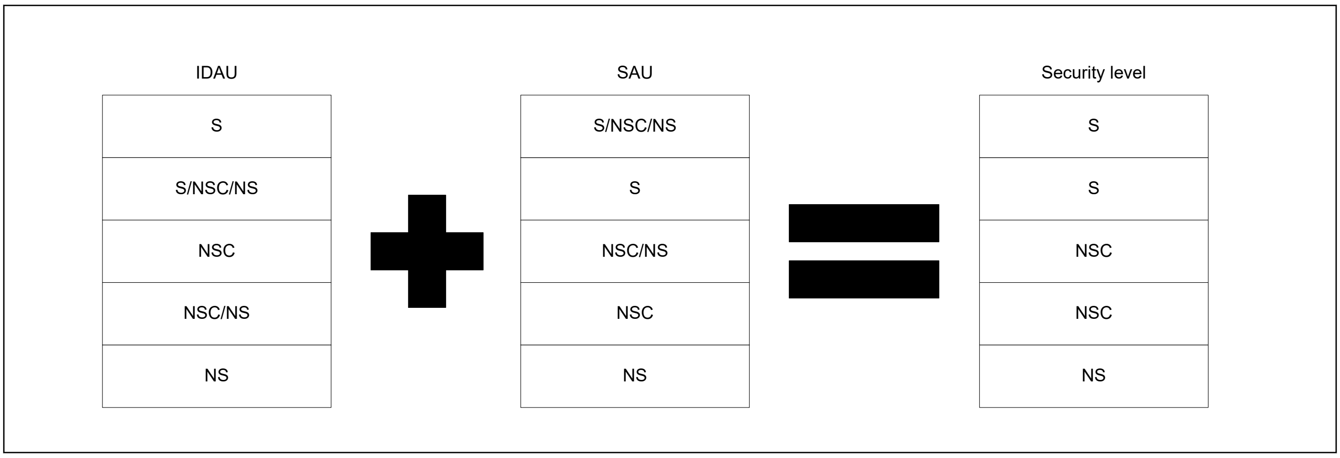


Figure 51.3 Final determination of address map security level

When using TrustZone to perform secure and non-secure region separation, SAU MUST be set according to the following.

The regions set as NS attribute in IDAU MUST be set to NS in the SAU as well. The regions set to NS attribute in IDAU are:

0x1000_0000 to 0x1FFF_FFFF (SAU Region 1 in Figure 51.4)

0x3000_0000 to 0x3FFF_FFFF (SAU Region 3 in Figure 51.4)

0x5000_0000 to 0xDFFF_FFFF (SAU Region 3 in Figure 51.4)

At least one NSC region MUST be created within any region defined as NSC by the IDAU. The regions set to NSC attribute by the IDAU are:

0x0000_0000 to 0x0FFF_FFFF (SAU Region 0 in Figure 51.4)

0x2000_0000 to 0x2FFF_FFFF (SAU Region 2 in Figure 51.4)

If you do not wish to define any isolation by means of using TrustZone, do not change SAU_CTRL.ALLNS = 0 and SAU_CTRL.ENABLE = 0 (initial value).

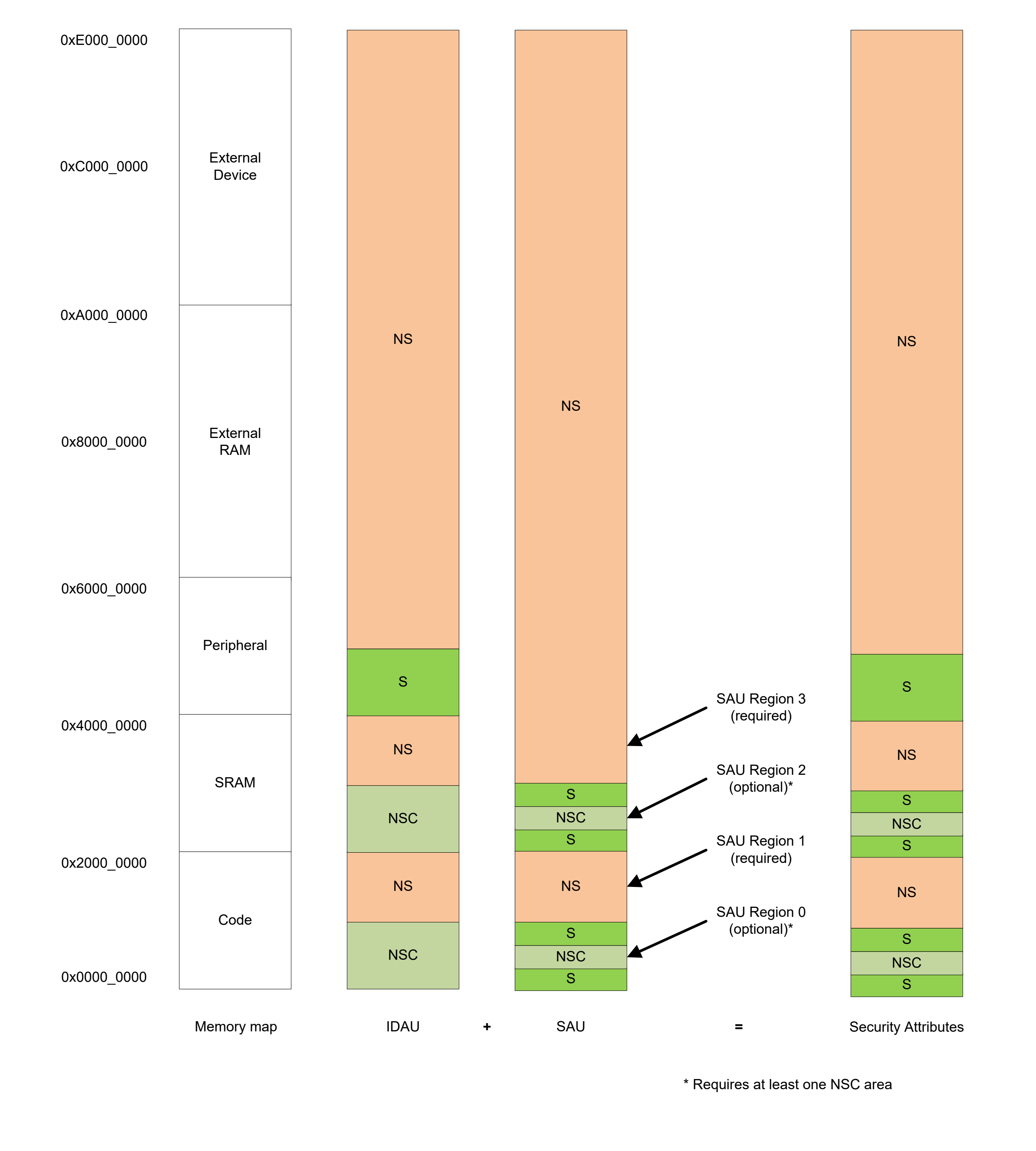


Figure 51.4 SAU settings and resulting security attributes

51.3.3.4 Region Number

The SAU and IDAU also define region numbers for each of the memory regions and security attributes. This region number is used by software to determine if a contiguous range of memory shares common security attributes. Figure 51.5 shows the defined region number of IDAU.

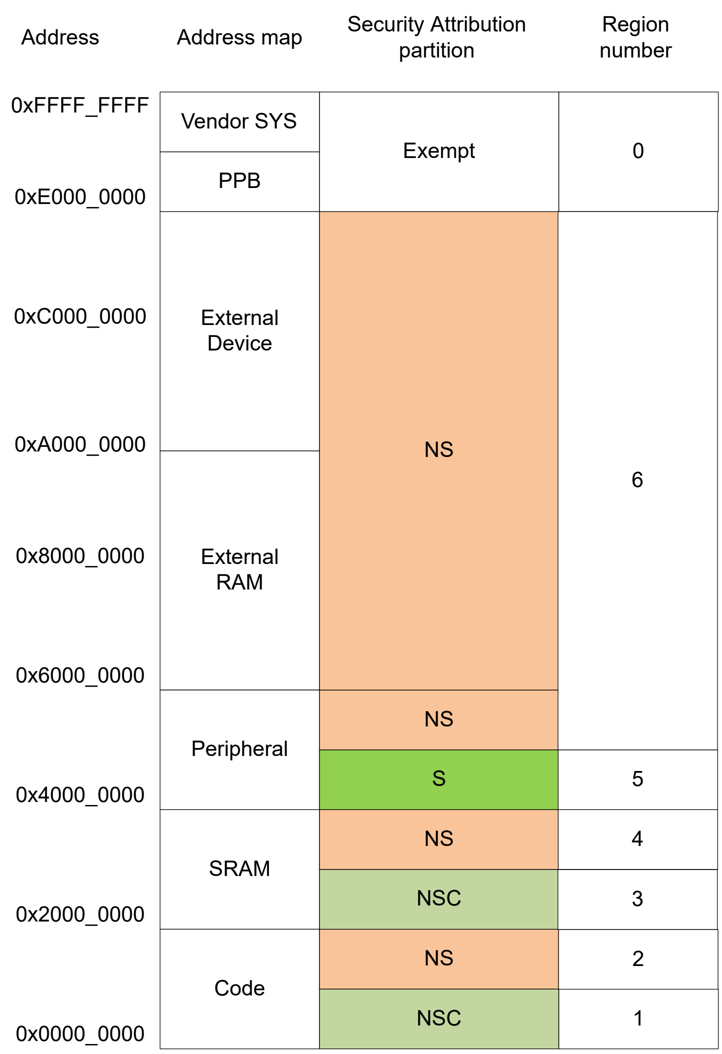


Figure 51.5 IDAU defined region numbers

51.3.3.5 Memory Security Attribution of TrustZone Filter

The memories are divided into S and NS regions. The memory security attribution of code MRAM and SiP flash (when equipped) is stored into nonvolatile memory by a boot firmware command when the device lifecycle is in the OEM state and the authentication level is AL2. These memory security attributions are applied before application execution. They cannot be updated by the application but are readable using CMSAMON or SFSAMON registers. The memory security attributions of CM33 TCM, SRAM and VBATT backup registers are set by a dedicated security attribution register writable only by secure access.

The code MRAM can be divided in up to two regions. The SiP flash can be divided in up to two regions. Programming or erasing of SiP flash should be managed by secure user to prevent non-secure user from programming or erasing the secure area. The CM33 TCM, SRAM and VBATT backup registers can be divided in up to two regions.. Figure 51.6 shows the memory mapping. Table 51.1 shows size of the memory region and Table 51.2 shows access permission of the memory region.

SRAM ECC region can be used as data region when ECC function is disabled. The security attribution of this region is set by the dedicated security attribute register. See section 58, SRAM for the details.

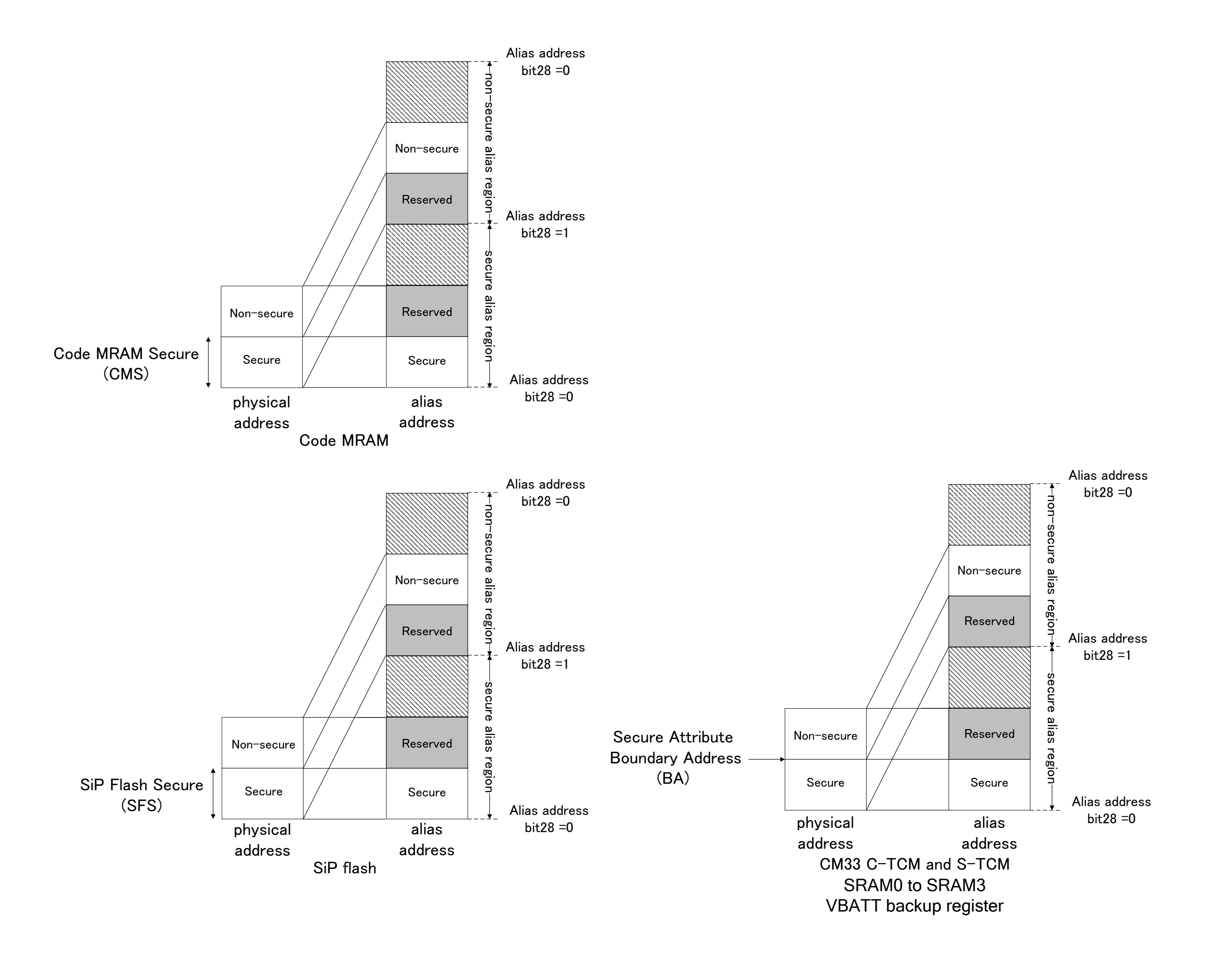


Figure 51.6 Memory mapping

Table 51.1 Size of memory region (1 of 2)

Memory region	Start address	Size
CM33 C-TCM secure	0x0000_0000*1	BA (8 KB)
CM33 C-TCM non-secure	0x1000_0000 + BA (8 KB)*1	C-TCM size - BA (8 KB)
Code MRAM secure	0x0200_0000	CMS size (32 KB)
Code MRAM non-secure	0x1200_0000 + CMS size (32 KB)	Code MRAM size - CMS size (32 KB)
SiP flash secure	0x0800_0000	SFS size (32 KB)
SiP flash non-secure	0x1800_0000 + SFS size (32 KB)	SiP flash size - SFS size (32 KB)
CM33 S-TCM secure	0x2000_0000*1	BA (8 KB)
CM33 S-TCM non-secure	0x3000_0000 + BA (8 KB)*1	S-TCM size - BA (8 KB)
SRAM0 secure	0x2200_0000	BA (8 KB)
SRAM0 non-secure	0x3200_0000 + BA (8 KB)	SRAM0 size - BA (8 KB)
SRAM1 secure	0x2208_0000	BA (8 KB)
SRAM1 non-secure	0x3208_0000 + BA (8 KB)	SRAM1 size - BA (8 KB)
SRAM2 secure	0x2210_0000	BA (8 KB)
SRAM2 non-secure	0x3210_0000 + BA (8 KB)	SRAM2 size - BA (8 KB)
SRAM3 secure	0x2218 0000	BA (8 KB)

Table 51.1 Size of memory region (2 of 2)

Memory region	Start address	Size
SRAM3 non-secure	0x3218 0000 + BA (8 KB)	SRAM3 size - BA (8KB)
VBATT backup register secure	0x4001_ED00	BA (32 bytes)
VBATT backup register non-secure	0x5001_ED00 + BA (32 bytes)	Backup register size - BA (32 bytes)

Note: The number in parentheses indicates the smallest unit that can be set by the user.
Note: BA is the setting value of the security attribution boundary address register for each memory region.
Note 1. Address of CPU1. For address of other bus masters, see Table 2.37.

Table 51.2 Access permission of the memory region

Memory region	Secure transaction	Non-secure transaction
Each memory region configured as S or NSC	Allowed	Write ignored/Read ignored TrustZone access error is generated
Each memory region configured as NS	Write ignored/Read ignored TrustZone access error is generated	Allowed

51.3.3.6 CPU Security Attribution

Cortex-M33 security extension can be disabled by the boot firmware command only when cortex-M33 is configured as the secondary CPU in dual core product. Disabling the Cortex-M33 security extension is a permanent setting.

51.3.3.7 Peripheral Security and Privilege Attribution of TrustZone Filter

Each peripheral can be configured as secure or non-secure and privileged or unprivileged. Peripherals are divided into two types.

Type1 peripheral has one security and privilege attribution and access to all registers is controlled by one security and privilege attribution. Type1 peripheral security and privilege attribution is set to the PSARx (x = B to E) and the PPARx (x = B to E) registers by the secure and privileged application.

Type2 peripheral has the security and the privilege attribution for each register or for each bit and access to each register or bit field is controlled according to these security and privilege attributions. Type2 peripheral security and privilege attribution is set to the Security Attribution register and the Privilege Attribution register in each module by the secure and privileged application. For details on the Security Attribution register and the Privilege Attribution register, see each section.

Table 51.3 shows the classification of peripheral type.

Table 51.3 Classification of peripheral type

Type1	Type2
SCI, SPI, OSPI, DOTF, USBHS, USBFS, IIC, I3C, RSIP-E50D, ESWM, CANFD, PDMIF, CEU, DOC, SDHI, SSIE, CRC, CAC, ACMPHS, TSN, ADC16H, DAC12, POEG, PDG, AGT, GPT, ULPT, RTC, IWDT, and WDT	System control (Resets, PVD, Clock Generation Circuit, Low Power Modes, Battery Backup Function, Register Write Protection Function), MRAM controller, SRAM controller, CPU cache, DMAC, DTC, ICU, MPU, BUS, Security setting, ELC, and I/O ports

Table 51.4 shows the access permission of type1 peripherals. The access permission of type2 peripherals is different by peripherals. Each register description of type2 peripherals includes the note of S-TYPE and P-TYPE indicating the specification of the access permission. For details on S-TYPE and P-TYPE, see section Appendix 4, Notes for Register R/W.

Table 51.4 Access permission of type1 peripherals (1 of 2)

	Secure and Privileged access	Secure and Unprivileged access	Non-secure and Privileged access	Non-secure and Unprivileged access
Peripheral configured as Secure and Privileged	Allowed	Write ignored/Read ignored TrustZone Access error is generated	Write ignored/Read ignored TrustZone Access error is generated	Write ignored/Read ignored TrustZone Access error is generated
Peripheral configured as Secure and Unprivileged	Allowed	Allowed	Write ignored/Read ignored TrustZone Access error is generated	Write ignored/Read ignored TrustZone Access error is generated

Table 51.4 Access permission of type1 peripherals (2 of 2)

	Secure and Privileged access	Secure and Unprivileged access	Non-secure and Privileged access	Non-secure and Unprivileged access
Peripheral configured as Non-secure and Privileged	Write ignored/Read ignored TrustZone Access error is generated	Write ignored/Read ignored TrustZone Access error is generated	Allowed	Write ignored/Read ignored TrustZone Access error is generated
Peripheral configured as Non-secure and Unprivileged	Write ignored/Read ignored TrustZone Access error is generated	Write ignored/Read ignored TrustZone Access error is generated	Allowed	Allowed

51.3.4 TrustZone Access Error

Table 51.5 shows the behavior when TrustZone access error occurs. The behavior varies depending on the master.

Table 51.5 Behavior of TrustZone access error

DAP	CPU	DMAC/DTC	ESWM	CEU
• Only error response is returned*1	• IDAU/SAU detects SecureFault exception • TrustZone filter detects BusFault exception • Can issue IRQ or reset*2	• Stop transfer • Issue IRQ or reset*2 • Generate interrupt (DMAn_TRANSERR)	• Issue IRQ or reset*2 • Generate interrupt (ETHER_GWEI)*3	• Issue IRQ or reset*2

Note: This behavior is not applicable for bufferable write access. For more information on when a bufferable write access error is detected, see section 15, Buses.

Note 1. When a TrustZone access error occurs by a debugger access, exception, NMI, or reset does not occur. Only the error response is returned.

Note 2. The operation after error detection is selected by the BUSOAD.SLERROAD bit.

Note 3. AXI error status in GWEIS0.AES bit is set. The interrupt occurs when interrupt is enabled in GWEIE0.AEE bit.

51.4 Device Lifecycle Management

Device Lifecycle Management (DLM) identifies the current development/production/deployment phase of the device and controls the capabilities of the debug function, the serial programming interface, and Renesas test mode. Figure 51.7 shows the available device lifecycle states and Table 51.6 shows the lifecycle state definitions and capabilities in each state.

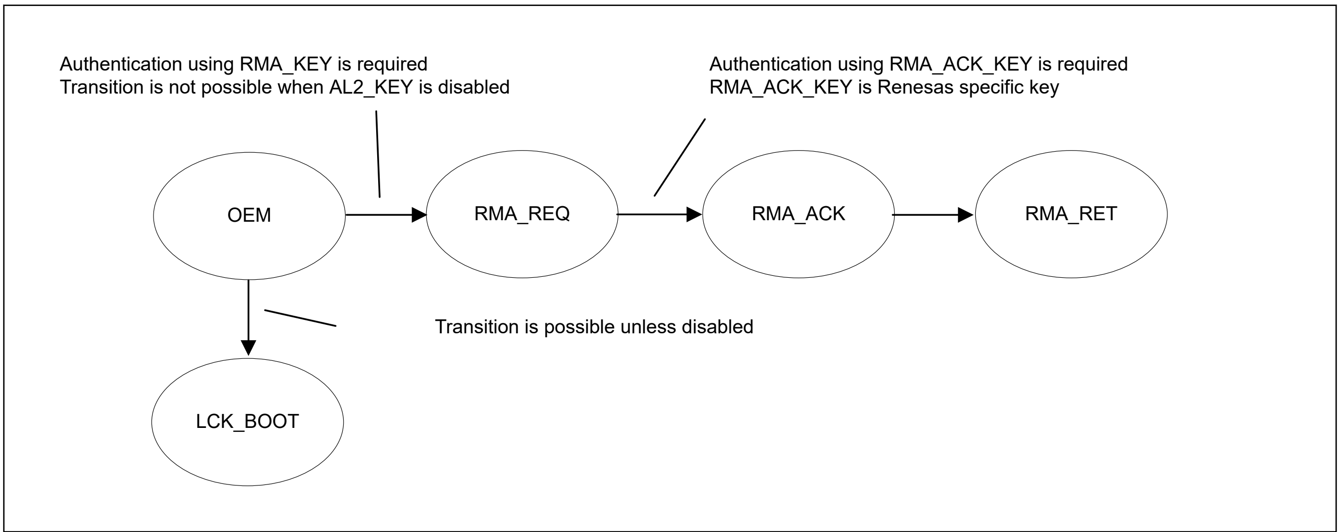


Figure 51.7 Device lifecycle states

Table 51.6 Lifecycle state definition and capabilities in each state

Lifecycle	Definition	Protection level	Debug function	Serial programming	Renesas test mode
OEM	"Original Equipment Manufacturer" The device is owned by the customer.	PL2 or PL1 or PL0	Depend on the authentication level		Not available
LCK_BOOT	"LoCKed BOOT interface" The debug interface and the serial programming interface are permanently disabled.	PL0	Not available	Not available	Not available
RMA_REQ	"Return Material Authorization REQuest" Request for RMA. The customer must send the device to Renesas in this state.	PL0	Not available	Available Cannot access code MRAM/SiP flash area	Not available
RMA_ACK	"Return Material Authorization ACKnowledged" Failure analysis in Renesas	PL2	Available in the secure and non-secure debug	Available Cannot access code MRAM/SiP flash area	Available
RMA_RET	"Return Material Authorization RETurn" The device is back to the customer. The device does not boot.	PL0	Not available	Not available	Not available

51.4.1 Changing the Lifecycle State

Use the boot firmware commands to change the device lifecycle state. See the boot firmware application note for details of the command. The lifecycle state cannot be updated by an application, but the current lifecycle state can be read through the DLMMON register.

As shown in Figure 51.7, each transition is one-way and the state cannot be regressed.

Transition from OEM to RMA_REQ requires key authentication using the RMA_KEY which must have been previously injected by the customer. The key length of the RMA_KEY is 128 bits. Inject the RMA_KEY as shown in section 51.5. Secure Key Injection. RMA_KEY can be injected in AL2.

The key authentication uses a challenge and response authentication or authentication using the MCU's unique ID. The response (challenge and response authentication) or the authentication code (using the MCU's unique ID) can be calculated as follows:

Response = AES-128 CMAC (RMA_KEY, 128-bit challenge)

Authentication code = AES-128 CMAC (RMA_KEY, 128-bit MCU unique ID)

The contents of the MRAM and SiP flash except permanently locked blocks or registers are erased when transitioning to RMA_REQ. The contents of the permanently locked blocks or registers can be read by Renesas at failure analysis. An MRAM block can be permanently locked by setting the PBPS/PBPS_SEC registers to permanently disable writing of the block. The option-setting memory can be permanently locked by the POFSPS register, permanently disabled writing of the register. Transition to RMA_REQ is not possible if the AL2_KEY is disabled. The MCU does not respond after changing the device lifecycle state to RMA_REQ. To continue to use boot firmware commands, you must enter boot mode again after a reset. See the *Boot Firmware Application Note* for details.

Transition from OEM to LCK_BOOT is possible unless that transition has been explicitly disabled. Use the parameter setting command in AL2 or AL1 to prohibit the transition to LCK_BOOT. The LCK_BOOT transition prohibition is a permanent setting and cannot be undone. The debug interface and serial programming interface are permanently disabled in LCK_BOOT.

51.4.2 Protection and Authentication Level

The protection level (PL) and the authentication level (AL) determine the availability of the debug function and the serial programming interface. PL and AL are fixed except in the OEM state. For PL and for AL, three levels can be set for each in the OEM state. AL indicates a temporary authentication status and is initialized to PL after an MCU power-on reset. Both PL and AL can be changed only by the boot firmware and cannot be changed by the application. Figure 51.8 shows the available Protection Level and Authentication Level states and transitions. Table 51.7 shows the availability of the debug function and the serial programming interface in each level.

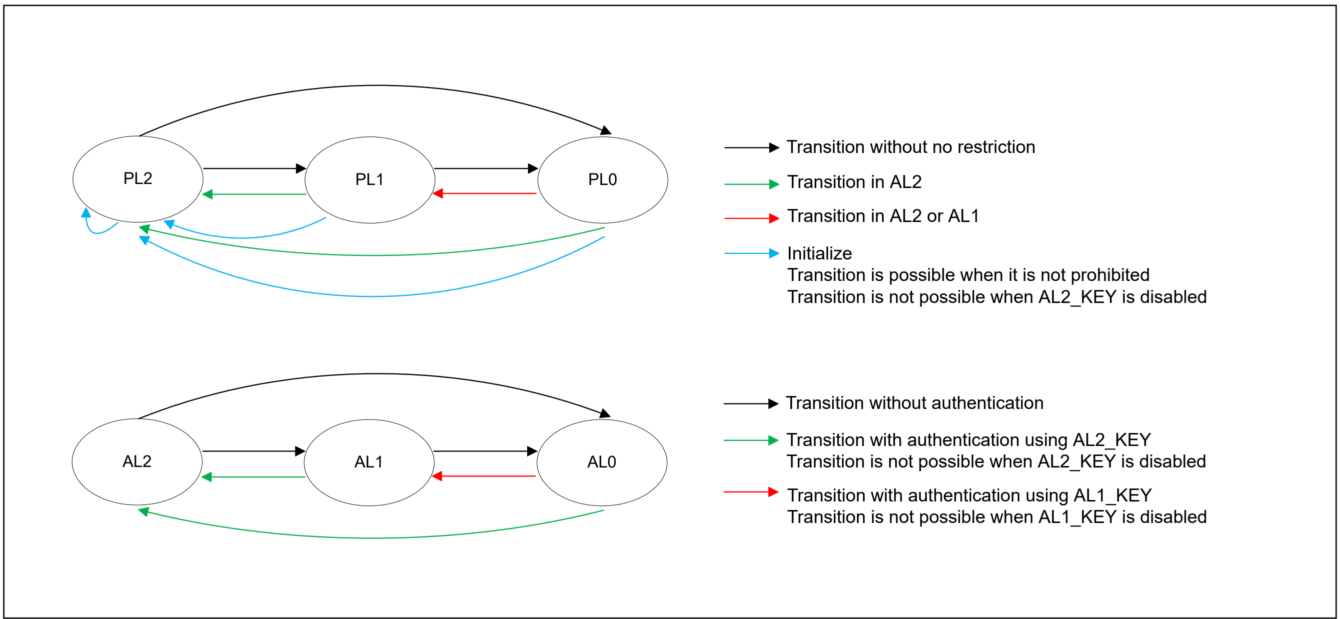


Figure 51.8 PL and AL states and transitions

Table 51.7 Availability of the debug function and the serial programming interface in each authentication level

AL	Debug function	Serial programming interface
AL2	Non-secure and Secure debug functions are enabled and accessible from the debugger	All functions are available
AL1	Only non-secure debug function is enabled, and debugger can access only defined non-secure debug accessible regions	Available but cannot program, erase, or read secure code MRAM or SiP flash area
AL0	No debug functions are available	Available but cannot access code MRAM or SiP flash area

Use the boot firmware commands to change PL and AL. See the boot firmware application note for details of the command.

Changing to a lower PL can be done with no restriction. Changing to a higher PL requires that the MCU be at that AL. For example, when the current AL is AL1, changing to PL1 is possible but changing to PL2 is impossible. PL can be reset by the Initialize command unless the command itself is disabled. After the Initialize command is executed, PL is set back to PL2 and the contents on the MRAM and SiP flash memory are erased. Initialize command does not execute when OTP area of option-setting memory is not blank. Initialize command can be disabled permanently in all AL states by parameter setting command to prevent memory erasure without authorization. Initialize command is disabled also when AL2_KEY is disabled. MCU does not respond after executing the Initialize command. To continue to use the serial programming commands, reenter the boot mode after a reset. See the *Boot Firmware Application Note* for the detail.

Changing to a lower AL can be done without authentication. Changing to a higher AL requires key authentication using AL2_KEY or AL1_KEY, as appropriate. These keys are 128-bit keys. Injection of AL2_KEY or AL1_KEY is performed as shown in section 51.5. Secure Key Injection. AL2_KEY can be injected in AL2, AL1_KEY can be injected in AL2 or AL1. The key authentication uses a challenge and response authentication. The response can be calculated as follows:

Response = AES-128 CMAC (KEY, 128 bits challenge)

AL2_KEY can be disabled permanently in AL2 with the parameter setting command.

AL1_KEY can be disabled permanently in AL2 or AL1 by parameter setting command.

51.4.3 Serial Programming

Whether a serial programmer can be connected and the range of memory that can be accessed depends on the device lifecycle state and AL as shown in Table 51.6 and Table 51.7. In addition, the accepted serial programming commands differ depending on the device lifecycle state and AL. See the *Boot Firmware Application Note* for details of the commands.

51.4.4 Device Lifecycle State and PL Change Example

The following section describes a typical device lifecycle state and PL change example.

- Secure developer
  - Set the memory security attribution of the code MRAM and SiP flash using the boot firmware command
  - Program and debug the secure application
  - If the AL2_KEY and RMA_KEY are required, inject them using the boot firmware commands
  - Inject application AES, ChaCha20-Poly1305, RSA, ECC, HMAC and EdDSA keys listed in Table 51.8 (if required)
  - Prepare the MCU for the non-secure developer
    - If the non-secure developer is not permitted to use the Initialize command, disable it using the boot firmware command.
    - If the AL2_KEY is not used and the non-secure developer is not permitted to use the Initialize function, disable the AL2_KEY using the boot firmware command.
    - Change the PL from PL2 to PL1 using the boot firmware command
- Non-secure developer
  - Program and debug the non-secure application.
  - If the AL1_KEY is required, inject it using the boot firmware command.
  - Inject application AES, ChaCha20-Poly1305, RSA, ECC, HMAC and EdDSA keys listed in Table 51.8 (if required)
  - Prepare the MCU for end-product deployment
    - Disable the Initialize command using the boot firmware command (if required)
    - If the AL1_KEY is not used, disable the AL1_KEY using the boot firmware command
    - Change the PL from PL1 to PL0 using the boot firmware command

51.4.5 Failure Analysis

If the customer requests failure analysis by Renesas, it is necessary to send the device after changing the device lifecycle state to RMA_REQ. If the device lifecycle state is not RMA_REQ, Renesas cannot perform the failure analysis. After failure analysis, Renesas changes the lifecycle to RMA_RET and returns the device to the customer.

Note: RMA_KEY is required to change the device lifecycle state to RMA_REQ or the MCU's Unique ID must be used as part of an authentication code. See section 51.4.1. Changing the Lifecycle State for details.

51.5 Secure Key Injection

To inject a user key into the MCU, perform the following steps.

Renesas provides the Security Key Management Tool, available on the Renesas website, to assist with key injection preparation.

1. Create a 256-bit installation key.
   This key is called the User Factory Programming Key (UFPK) and is used to wrap a user key
2. Get the wrapped version (W-UFPK) of the UFPK through the Renesas Key Wrapping Service.
3. Wrap the user key using the UFPK.
4. Send the W-UFPK and the wrapped user key to the MCU using a boot firmware command.
   The user key is unwrapped, wrapped with the MCU's hardware unique key, and stored in nonvolatile memory. DLM keys are stored in unmapped flash. Application keys are stored at the address specified with the key injection command.

Figure 51.9 shows an example of key injection and Table 51.8 shows the keys that can be injected.

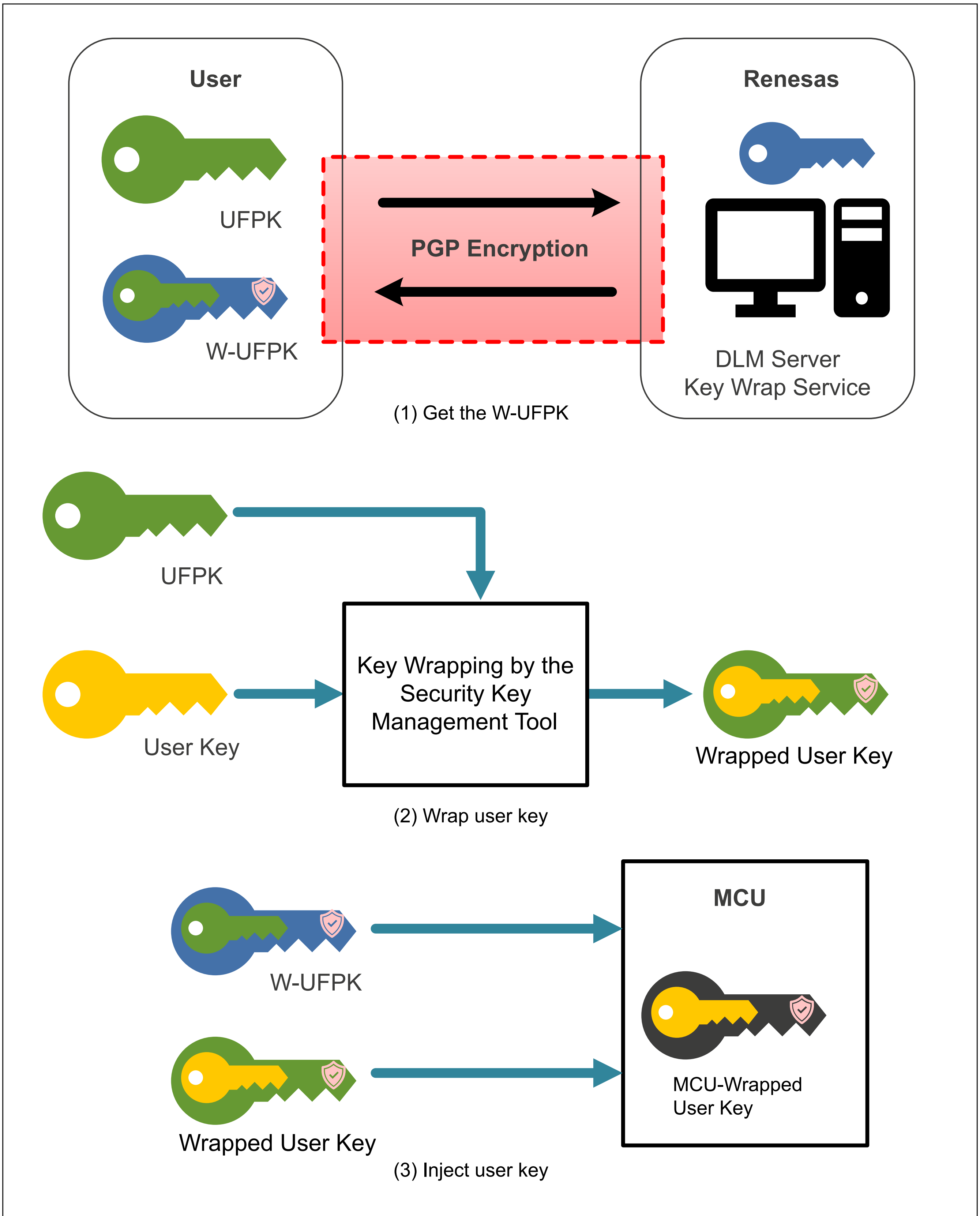


Figure 51.9 Key injection

Table 51.8 Keys that can be injected

DLM transition	AL transition	AES	ChaCha20-Poly-1305	RSA	ECC	HMAC	EdDSA	Secure boot	Other
RMA_KEY	AL2_KEY, AL1_KEY	AES-128, AES-192, AES-256, AES-XTS-128, AES-XTS-256	ChaCha20-Poly-1305	RSA-1024, RSA-2048, RSA-3072, RSA-4096 (Public and Private) RSA-2048 Public Key for TLS	ECC P-192, ECC P-224, ECC P-256, ECC P-384, ECC P-521 (Public and Private) ECC P256r1, ECC P384r1, ECC P512r1, ECC secp256k1 (Public and Private)	HMAC-SHA224, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512, HMAC-SHA512/224, HMAC-SHA512/256, HMAC-SHA3-224, HMAC-SHA3-256, HMAC-SHA3-384, HMAC-SHA3-512	Ed25519 (Public and Private)	Up to 4 OEM_ROOT_PKs	Key-Update Key

51.6 Secure Factory Programming

In addition to secure key injection support for injecting user keys (DLM, debug authentication, application, and secure boot keys), the MCU supports the programming of a firmware image in ciphertext format to prevent assets from being leaked during production programming. This enables secure factory programming in a non-secure environment. Secure factory programming is supported by the boot firmware. The maximum capacity that can be programmed with the secure factory programming is almost 16 MB. See the security key management tool user's manual or the boot firmware application note for details of the capacity limitation. Figure 51.10 shows an example of secure factory programming of an encrypted firmware image. The customer wraps the Image Encryption Key with the UFPK and encrypts the image with the Image Encryption Key using AES128-CCM. When the customer sends the W-UFPK, the wrapped Image Encryption Key, and encrypted image to the MCU through the serial programming interface, the MCU decrypts and programs the firmware image.

To further support secure factory programming in a non-secure environment, the DLM state, Protection Level, and Authentication Keys can all be set via a single boot firmware command. Points to note about this boot firmware command include:

- Encrypted firmware programming can be performed only when the MCU is in the OEM state.
- This command changes the Protection Level of the MCU. The initial PL must be PL2. The final PL must be PL0.
- If the DLM state remains in the OEM state, AL2 key must be injected. AL1 key can optionally also be injected.
- The MCU can be transitioned to the LCK_BOOT state. In this case, AL keys cannot be injected.
- AL keys must be wrapped with the same UFPK as the Image Encryption Key.
- This command erases all code MRAM area except option-setting memory before programming encrypted firmware image. If there are any permanently locked blocks, this command cannot be executed.
- Prohibit to execute this command if the current register setting or write value to the registers related to startup area selection are other than the following:
  - SAS.BTFLG = 1b
- All option-setting memory values must be included in the encrypted firmware image, including those not changed from their default values. However, if the following areas are write-protected, the write data in these areas should not be included in the image. If included, this command terminates with an error.
  - Configuration setting area protected by POFSPS register.

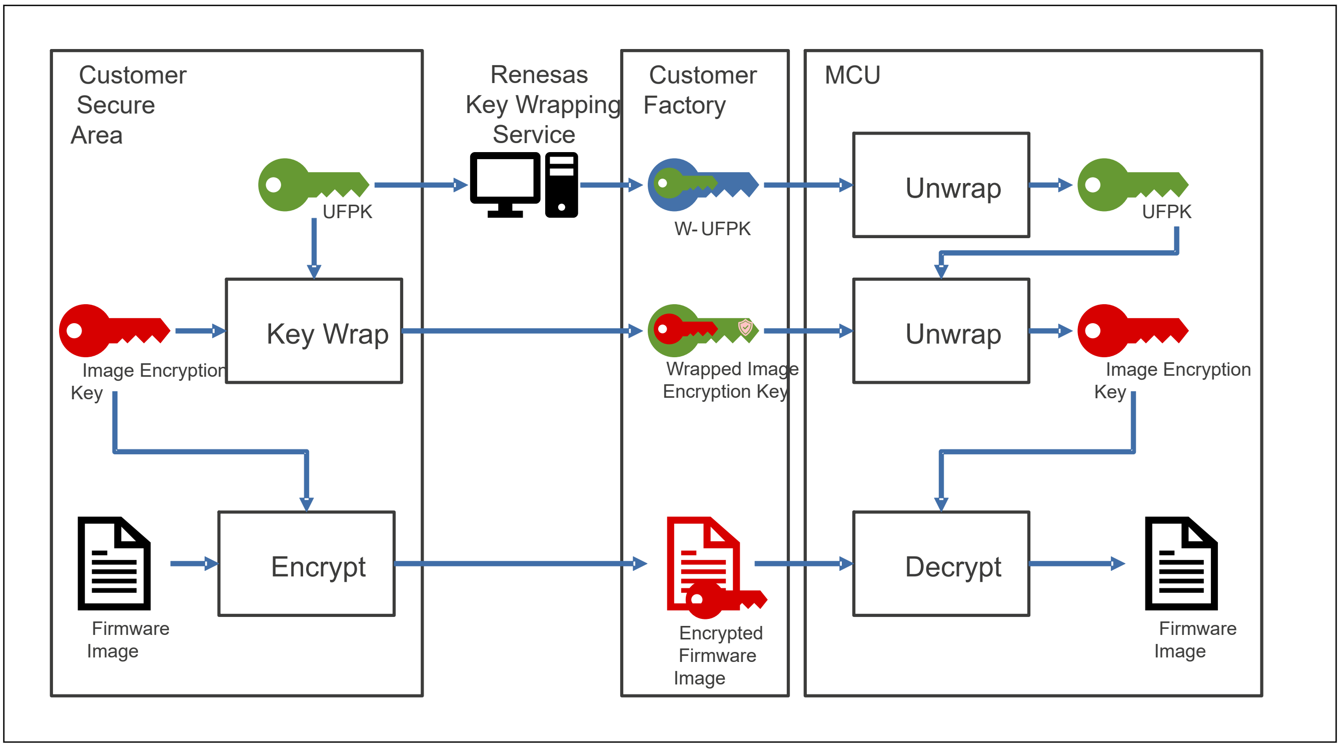


Figure 51.10 Secure factory encrypted image programming

51.7 Secure Boot

Secure boot verifies the integrity and authenticity of an OEM boot loader (called OEM_BL) starting at the beginning of application executable memory.

The OEM_BL is verified when it is initially programmed and prior to execution.

OEM_BL is verified when it is programmed using the boot firmware. Two certificates are used for validation. One is the key certificate, which is used to certify the OEM_BL verification key (called the OEM_BL_PK, the public part of the OEM_BL verification key pair). The key certificate is signed by the secret part of OEM_BL verification root key pair (called OEM_ROOT_SK). The public part of OEM_BL verification root key pair (called OEM_ROOT_PK) is registered with the MCU as the root of trust key using secure key injection as described in section 51.5. Secure Key Injection. The OEM_ROOT_SK and OEM_ROOT_PK are a 256-bit ECC (secp256r1 curve) key pair. The SHA2-256 hash of the OEM_ROOT_PK is stored during the secure key injection process. Up to four OEM_ROOT_PKs can be registered. These OEM_ROOT_PKs can be revoked if they are no longer needed. The validation process succeeds if the non-revoked first OEM_ROOT_PK with the biggest n value matches.

The other certificate for validation during programming is the code certificate, which is used to certify the OEM_BL. The code certificate is signed by the secret part of OEM_BL verification key pair (called OEM_BL_SK). After successful validation of the OEM_BL, the HMAC value of OEM_BL and the code certificate (called OEM_BL_digest) is generated and programmed to MRAM. HMAC SHA2-256 is used to generate the HMAC value. A derived key from the HUK is used as the HMAC key, so the HMAC value is unique in each MCU.

In single-chip mode, the immutable first stage boot loader (called FSBL) in the on-chip OTP is executed after reset when the FSBL is enabled in the FSBLCTRL0 register. When secure boot is selected in FSBLCTRL1 register, the FSBL generates the HMAC value of the OEM_BL and the code certificate, and compares it to the expected HMAC value. When the HMAC value is the same, the FSBL jumps to the OEM_BL. When CRC boot is selected in the FSBLCTRL1 register, the FSBL calculates the CRC of the OEM_BL and compares it to the expected CRC value, which is located in the code certificate. If the CRC values match, the FSBL jumps to the OEM_BL.

The code certificate must be programmed into MRAM, and its location in MRAM must be programmed into the SACC0n register or SACC1n register. The SACC0n register specifies the start address of the code certificate when BTFLG = 1. The SACC1n register specifies the start address of the code certificate when BTFLG = 0. Because SACC0n and SACC1n registers are OTP and cannot be erased, there are four areas each for updating the code certificate start address. FSBL reads the values in order from the register with the largest n value, and uses the first value that is not all 1 as the start address of the code certificate. Therefore, use registers in order from the smallest n value.

When measurement reporting is enabled in the FSBLCTRL1 register, the FSBL stores the measurement report to the SRAM address specified by the SAMR register. When reading the measurement report, disable the ECC function by setting SRAMCRn.ECCMOD[1:0] = 00b.

FSBL execution can be skipped after a software reset or deep software standby reset, as configured by the FSBLCTRL0 register.

If the generated HMAC or CRC value is different from the expected value, the high level is output to the port set by the FSBLCTRL2 register and the MCU goes to CPU Sleep mode.

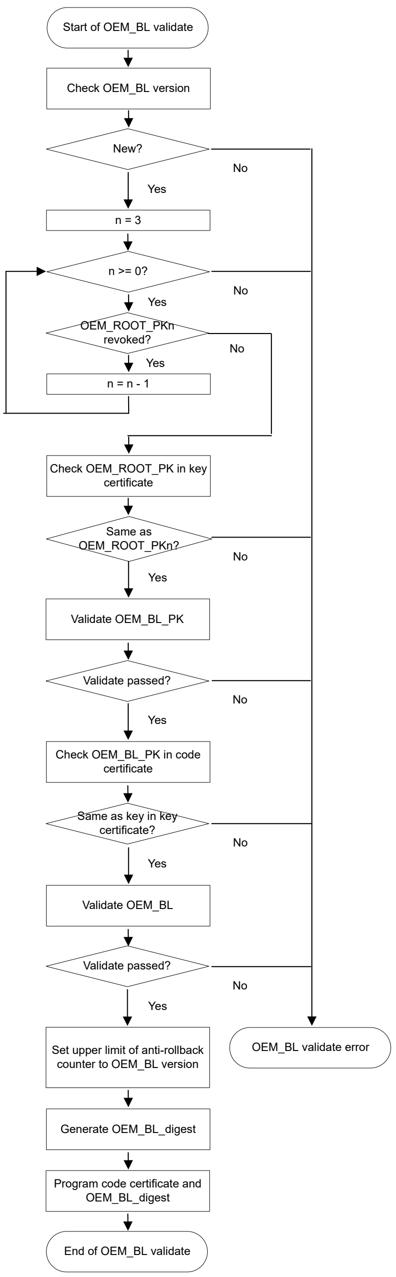


Figure 51.11 OEM_BL validation flow in serial programming mode

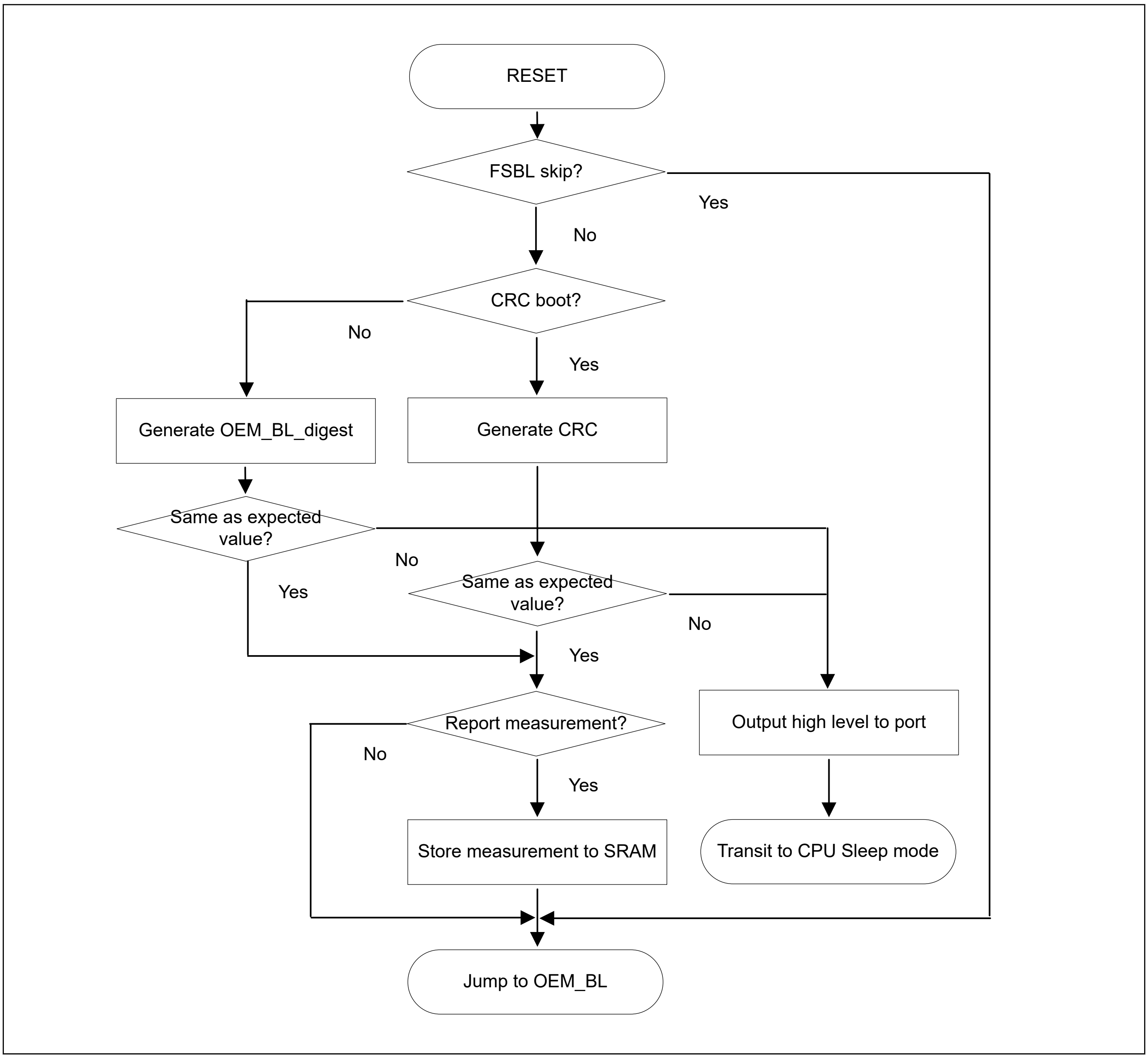


Figure 51.12 FSBL flow in single-chip mode

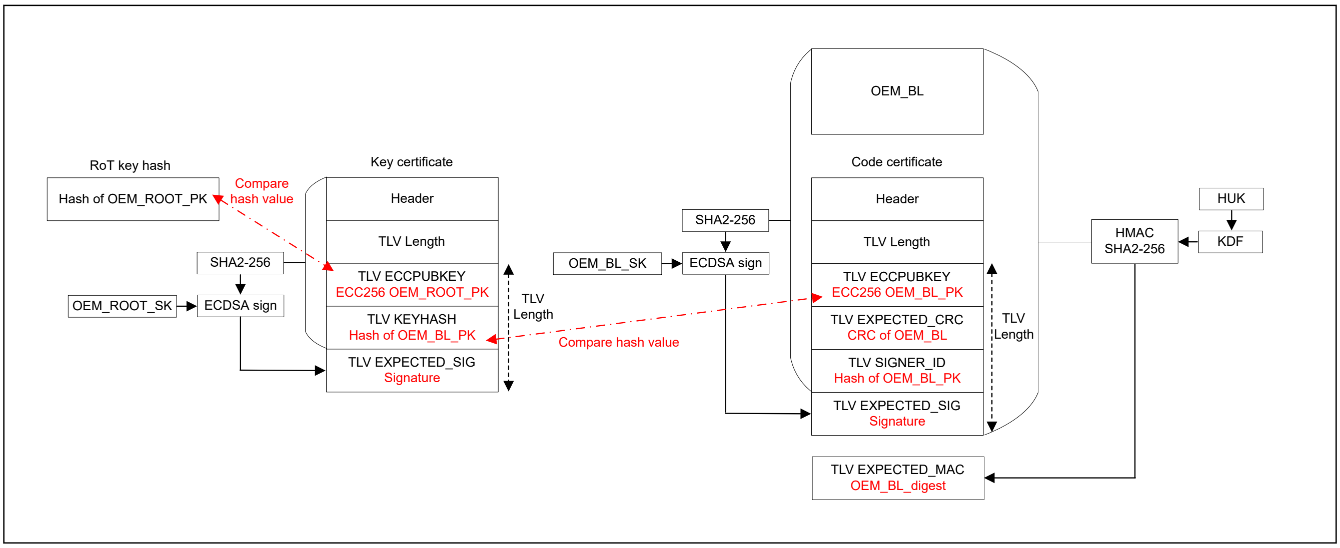


Figure 51.13 Validation structure and certificate format

Table 51.9 Detail of key certificate

Field		Size [byte]	Description
Header	Magic	4	Set to 0x6B657963
	Manifest version	4	Set to 0x00010000
	Flags	4	Reserved for future use
	Reserved	20	Reserved for future use
TLV Length		4	Length in bytes of TLV field
TLV ECCPUBKEY	Type & Length	4	Set to 0x00088010
	Value	64	ECC P-256 OEM_ROOT_PK
TLV KEYHASH	Type & Length	4	Set to 0x10144008
	Value	32	SHA2-256 hash value of OEM_BL_PK
TLV EXPECTED_SIG	Type & Length	4	Set to 0x20088410
	Value	64	Signature

Table 51.10 Detail of code certificate (1 of 2)

Field		Size [byte]	Description
Header	Magic	4	Set to 0x636F6463
	Manifest version	4	Set to 0x00010000
	Flags	4	Set to 0x00000000
	Load Addr	4	Set to 0x00000000
	Dest Addr	4	Set to 0x00000000
	Image size	4	Size in bytes of OEM_BL set in multiples of 16, minimum size is 64 bytes
	Image version	4	Version number of OEM_BL. Can be specified from 1 to 64
	Build number	4	Set to 0x00000000
TLV length		4	Length in bytes of TLV field
TLV ECCPUBKEY	Type & Length	4	Set to 0x01088010
	Value	64	ECC P-256 OEM_BL_PK
TLV EXPECTED_CRC	Type & Length	4	Set to 0x40000001
	Value	4	CRC32 of OEM_BL

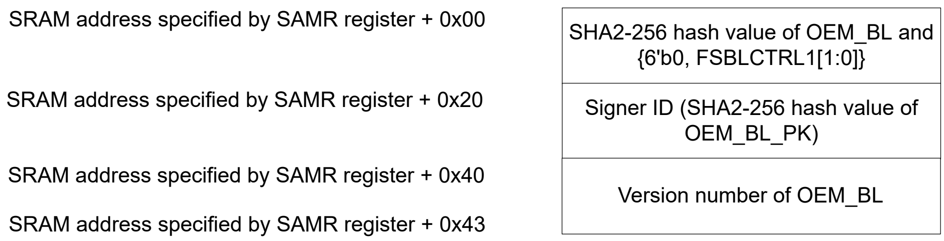
Table 51.10 Detail of code certificate (2 of 2)

Field		Size [byte]	Description
TLV SIGNER_ID	Type & Length	4	Set to 0x10144008
	Value	32	SHA2-256 hash value of OEM_BL_PK
TLV EXPECTED_SIG	Type & Length	4	Set to 0x25088410
	Value	64	Signature of [Code Certificate | OEM_BL], signed by the OEM_BL_SK When generating the signature, enter the code certificate and OEM_BL in the hash function in this order.

Table 51.11 Detail of OEM_BL_digest

Field		Size [byte]	Description
TLV EXPECTED_MAC	Type & Length	4	Fixed value 0x30184008
	Value	32	Unique OEM_BL_digest in each MCU

Note: OEM_BL_digest should be programmed in the area contiguous with the code certificate.

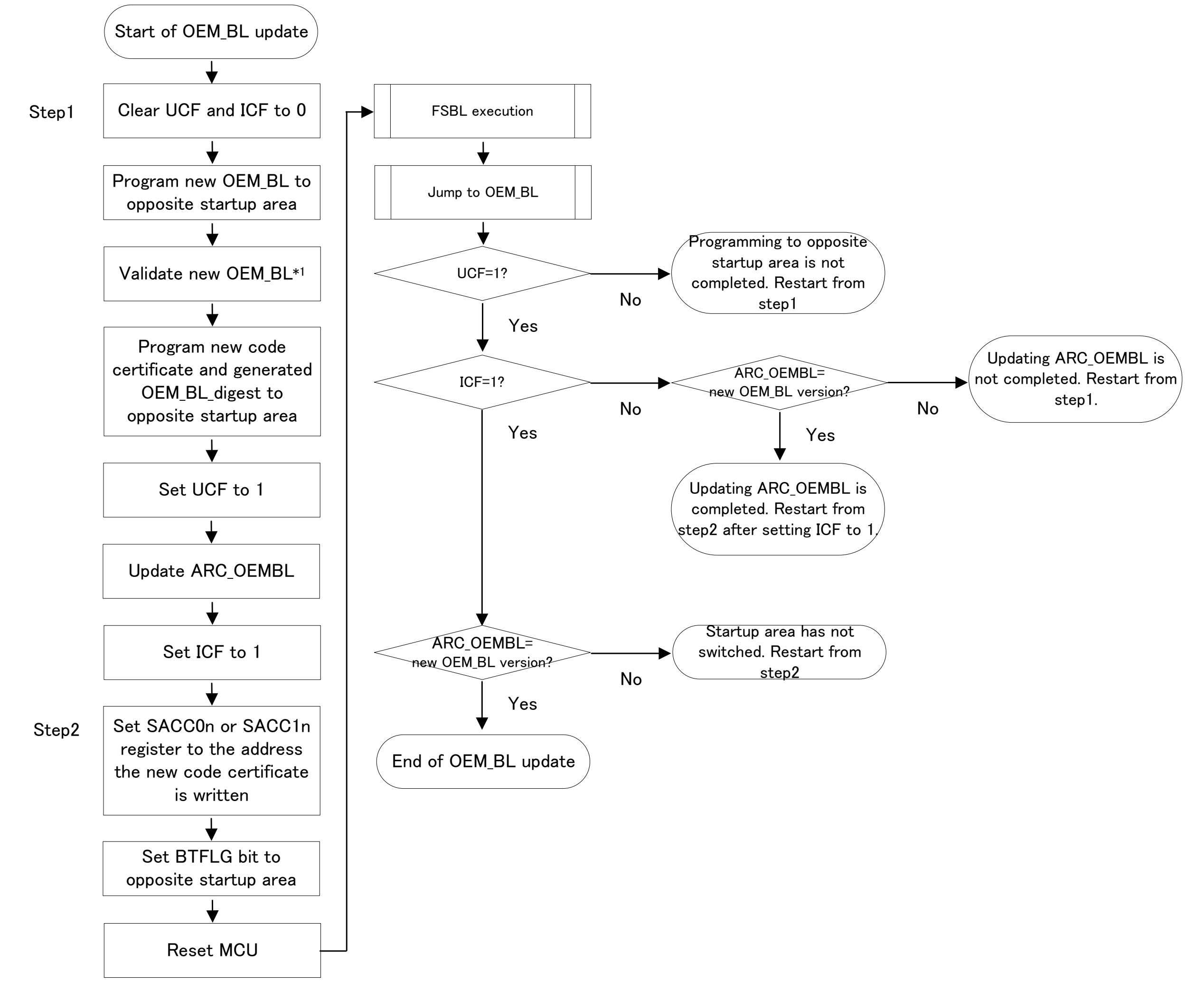


Note: OEM_BL and FSBLCTRL[7:0] are entered into the hash function in this order.

Figure 51.14 Format and location of the measurement report

OEM_BL can be updated by application code. In this case, the application code must validate the updated bootloader, generate the OEM_BL_digest, and store it immediately after the new code certificate, as per the flow described by Figure 51.11.Figure 51.15 show the OEM_BL update flow.

The implementation of the OEM_BL is user defined and application specific. However, it is recommended to implement status flags such as Update Complete Flag (UCF) and Increment Complete Flag (ICF) in the code MRAM as a countermeasure against update errors due to unexpected power failure. If an update error occurs, it can be recovered as shown in the following figures. If an update error occurs, the current OEM_BL is executed after FSBL execution. Note that the update error determination program is required for both the current OEM_BL and the new OEM_BL.



Note 1. Validate new OEM_BL the same way as Figure 51.11.

Figure 51.15 OEM_BL update flow

51.8 Register Description

51.8.1 PSARB : Peripheral Security Attribution Register B

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PSARB31	PSARB30	PSARB29	PSARB28	PSARB27	PSARB26	PSARB25	PSARB24	PSARB23	PSARB22	—	—	PSARB19	PSARB18	PSARB17	PSARB16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	PSARB12	PSARB11	—	PSARB9	PSARB8	PSARB7	—	—	PSARB4	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	PSARB4	I3C Bus Interface Security Attribution Target module: I3C and the MSTPCRB.MSTPB4 bit 0: Secure 1: Non-secure	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	PSARB7	I2C Bus Interface 2 Security Attribution Target module: IIC2 and the MSTPCRB.MSTPB7 bit 0: Secure 1: NonSecure	R/W
8	PSARB8	I2C Bus Interface 1 Security Attribution Target module: IIC1 and the MSTPCRB.MSTPB8 bit 0: Secure 1: Non-secure	R/W
9	PSARB9	I2C Bus Interface 0 Security Attribution Target module: IIC0 and the MSTPCRB.MSTPB9 bit 0: Secure 1: Non-secure	R/W
10	—	This bit is read as 0. The write value should be 0.	R/W
11	PSARB11	Universal Serial Bus 2.0 FS Interface 0 Security Attribution Target module: USBFS0 and the MSTPCRB.MSTPB11 bit 0: Secure 1: Non-secure	R/W
12	PSARB12	Universal Serial Bus 2.0 HS Interface Security Attribution Target module: USBHS and the MSTPCRB.MSTPB12 bit 0: Secure 1: Non-secure	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
16	PSARB16	Octa Memory Controller 0 Security Attribution Target module: OSPI0 (+DOTF0) and the MSTPCRB.MSTPB16 bit 0: Secure 1: Non-secure	R/W
17	PSARB17	Octa Memory Controller 1 Security Attribution Target module: OSPI1 (+DOTF1) and the MSTPCRB.MSTPB17 bit 0: Secure 1: Non-secure	R/W
18	PSARB18	Serial Peripheral Interface 1 Security Attribution Target module: RSPI1 and the MSTPCRB.MSTPB18 bit 0: Secure 1: Non-secure	R/W
19	PSARB19	Serial Peripheral Interface 0 Security Attribution Target module: RSPI0 and the MSTPCRB.MSTPB19 bit 0: Secure 1: Non-secure	R/W
21:20	—	These bits are read as 0. The write value should be 0.	R/W
22	PSARB22	Serial Communication Interface 9 Security Attribution Target module: SCI9 and the MSTPCRB.MSTPB22 bit 0: Secure 1: Non-secure	R/W
23	PSARB23	Serial Communication Interface 8 Security Attribution Target module: SCI8 and the MSTPCRB.MSTPB23 bit 0: Secure 1: Non-secure	R/W
24	PSARB24	Serial Communication Interface 7 Security Attribution Target module: SCI7 and the MSTPCRB.MSTPB24 bit 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
25	PSARB25	Serial Communication Interface 6 Security Attribution Target module: SCI6 and the MSTPCRB.MSTPB25 bit 0: Secure 1: Non-secure	R/W
26	PSARB26	Serial Communication Interface 5 Security Attribution Target module: SCI5 and the MSTPCRB.MSTPB26 bit 0: Secure 1: Non-secure	R/W
27	PSARB27	Serial Communication Interface 4 Security Attribution Target module: SCI4 and the MSTPCRB.MSTPB27 bit 0: Secure 1: Non-secure	R/W
28	PSARB28	Serial Communication Interface 3 Security Attribution Target module: SCI3 and the MSTPCRB.MSTPB28 bit 0: Secure 1: Non-secure	R/W
29	PSARB29	Serial Communication Interface 2 Security Attribution Target module: SCI2 and the MSTPCRB.MSTPB29 bit 0: Secure 1: Non-secure	R/W
30	PSARB30	Serial Communication Interface 1 Security Attribution Target module: SCI1 and the MSTPCRB.MSTPB30 bit 0: Secure 1: Non-secure	R/W
31	PSARB31	Serial Communication Interface 0 Security Attribution Target module: SCI0 and the MSTPCRB.MSTPB31 bit 0: Secure 1: Non-secure	R/W

Note: Access category: S-TYPE-1, P-TYPE-1.

The PSARB register specifies the security attribution for each module and the corresponding bit in Module Stop Control Register.

51.8.2 PSARC : Peripheral Security Attribution Register C

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PSARC31	PSARC30	—	PSARC28	PSARC27	PSARC26	—	PSARC24	—	—	—	—	—	—	—	PSARC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	PSARC13	PSARC12	PSARC11	—	—	PSARC8	PSARC7	—	—	—	—	—	PSARC1	PSARC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PSARC0	Clock Frequency Accuracy Measurement Circuit Security Attribution. Target module: CAC and the MSTPCRC.MSTPC0 bit 0: Secure 1: Non-secure	R/W
1	PSARC1	Cyclic Redundancy Check Calculator Security Attribution Target module: CRC and the MSTPCRC.MSTPC1 bit 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
6:2	—	These bits are read as 0. The write value should be 0.	R/W
7	PSARC7	Serial Sound Interface Enhanced (channel 1) Security Attribution Target module: SSIE1 and the MSTPCRC.MSTPC7 bit 0: Secure 1: Non-secure	R/W
8	PSARC8	Serial Sound Interface Enhanced (channel 0) Security Attribution Target module: SSIE0 and the MSTPCRC.MSTPC8 bit 0: Secure 1: Non-secure	R/W
10:9	—	These bits are read as 0. The write value should be 0.	R/W
11	PSARC11	Secure Digital Host IF 1 Security Attribution Target module: SDHI1 and the MSTPCRC.MSTPC11 bit 0: Secure 1: Non-secure	R/W
12	PSARC12	Secure Digital Host IF 0 Security Attribution Target module: SDHI0 and the MSTPCRC.MSTPC12 bit 0: Secure 1: Non-secure	R/W
13	PSARC13	Data Operation Circuit Security Attribution Target module: DOC and the MSTPCRC.MSTPC13 bit 0: Secure 1: Non-secure	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	PSARC16	CEU Security Attribution Target module: CEU and the MSTPCRC.MSTPC16 bit 0: Secure 1: Non-secure	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W
24	PSARC24	Pulse Density Modulation Interface Security Attribution Target module: PDMIF and the MSTPCRC.MSTPC24 bit 0: Secure 1: Non-secure	R/W
25	—	This bit is read as 0. The write value should be 0.	R/W
26	PSARC26	Controller Area Network with Flexible Data-Rate 1 Security Attribution Target module: CANFD1 and the MSTPCRC.MSTPC26 bit 0: Secure 1: Non-secure	R/W
27	PSARC27	Controller Area Network with Flexible Data-Rate 0 Security Attribution Target module: CANFD0 and the MSTPCRC.MSTPC27 bit 0: Secure 1: Non-secure	R/W
28	PSARC28	Ether-PHY clock Security Attribution Target module: ETHPHYCLK and the MSTPCRC.MSTPC28 bit 0: Secure 1: Non-secure	R/W
29	—	This bit is read as 0. The write value should be 0.	R/W
30	PSARC30	Layer 3 Ethernet Switch Module Security Attribution Target module: ESWM and the MSTPCRC.MSTPC30 bit 0: Secure 1: Non-secure	R/W
31	PSARC31	RSIP-E50D Security Attribution Target module: RSIP-E50D and the MSTPCRC.MSTPC31 bit 0: Secure 1: Non-secure	R/W

Note: Access category: S-TYPE-1, P-TYPE-1.

The PSARC register specifies the security attribution for each module and the corresponding bit in Module Stop Control

Register.

51.8.3 PSARD : Peripheral Security Attribution Register D

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	PSARD28	PSARD27	PSARD26	PSARD25	—	—	PSARD22	PSARD21	PSARD20	PSARD19	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	PSARD14	PSARD13	PSARD12	PSARD11	—	—	—	—	PSARD6	PSARD5	PSARD4	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	PSARD4	Asynchronous General Purpose Timer 1 Security Attribution Target module: AGT1 and the MSTPCRD.MSTPD4 bit 0: Secure 1: Non-secure	R/W
5	PSARD5	Asynchronous General Purpose Timer 0 Security Attribution Target module: AGT0 and the MSTPCRD.MSTPD5 bit 0: Secure 1: Non-secure	R/W
6	PSARD6	PWM Delay Generation Circuit Security Attribution Target module: PDG and the MSTPCRD.MSTPD6 bit 0: Secure 1: Non-secure	R/W
10:7	—	These bits are read as 0. The write value should be 0.	R/W
11	PSARD11	Port Output Enable for GPT Group 3 Security Attribution Target module: PGI3 and the MSTPCRD.MSTPD11 bit 0: Secure 1: Non-secure	R/W
12	PSARD12	Port Output Enable for GPT Group 2 Security Attribution Target module: PGI2 and the MSTPCRD.MSTPD12 bit 0: Secure 1: Non-secure	R/W
13	PSARD13	Port Output Enable for GPT Group 1 Security Attribution Target module: PGI1 and the MSTPCRD.MSTPD13 bit 0: Secure 1: Non-secure	R/W
14	PSARD14	Port Output Enable for GPT Group 0 Security Attribution Target module: PGI0 and the MSTPCRD.MSTPD14 bit 0: Secure 1: Non-secure	R/W
18:15	—	These bits are read as 0. The write value should be 0.	R/W
19	PSARD19	12-Bit D/A Converter 1 Security Attribution Target module: DAC121 and the MSTPCRD.MSTPD19 bit 0: Secure 1: Non-secure	R/W
20	PSARD20	12-Bit D/A Converter 0 Security Attribution Target module: DAC120 and the MSTPCRD.MSTPD20 bit 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
21	PSARD21	16-Bit A/D Converter Security Attribution Target module: ADC16H and the MSTPCRD.MSTPD21 bit 0: Secure 1: Non-secure	R/W
22	PSARD22	Temperature Sensor Security Attribution Target module: TSN and the MSTPCRD.MSTPD22 bit 0: Secure 1: Non-secure	R/W
24:23	—	These bits are read as 0. The write value should be 0.	R/W
25	PSARD25	High speed analog Comparator 3 Security Attribution Target module: ACMPHS3 and the MSTPCRD.MSTPD25 bit 0: Secure 1: Non-secure	R/W
26	PSARD26	High speed analog Comparator 2 Security Attribution Target module: ACMPHS2 and the MSTPCRD.MSTPD26 bit 0: Secure 1: Non-secure	R/W
27	PSARD27	High Speed analog Comparator 1 Security Attribution Target module: ACMPHS1 and the MSTPCRD.MSTPD27 bit 0: Secure 1: Non-secure	R/W
28	PSARD28	High Speed Analog Comparator 0 Security Attribution Target module: ACMPHS0 and the MSTPCRD.MSTPD28 bit 0: Secure 1: Non-secure	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note: Access category: S-TYPE-1, P-TYPE-1.

The PSARD register specifies the security attribution for each module and the corresponding bit in Module Stop Control Register.

51.8.4 PSARE : Peripheral Security Attribution Register E

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PSARE31	PSARE30	PSARE29	PSARE28	PSARE27	—	—	—	—	—	PSARE21	PSARE20	PSARE19	PSARE18	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PSARE15	—	—	—	—	—	PSARE9	PSARE8	—	—	—	—	PSARE3	PSARE2	PSARE1	PSARE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PSARE0	WDT1 Security Attribution Target module: WDT1 0: Secure 1: Non-secure	R/W
1	PSARE1	WDT0 Security Attribution Target module: WDT0 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
2	PSARE2	IWDT Security Attribution Target module: IWDT 0: Secure 1: Non-secure	R/W
3	PSARE3	Real Time Clock Security Attribution Target module: RTC 0: Secure 1: Non-secure	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	PSARE8	ULPT1 Security Attribution Target module: ULPT1 and the MSTPCRE.MSTPE8 bit 0: Secure 1: Non-secure	R/W
9	PSARE9	ULPT0 Security Attribution Target module: ULPT0 and the MSTPCRE.MSTPE9 bit 0: Secure 1: Non-secure	R/W
14:10	—	These bits are read as 0. The write value should be 0.	R/W
15	PSARE15	GPT common Security Attribution Target module: GPT GTCLKCR register 0: Secure 1: Non-secure	R/W
17:16	—	These bits are read as 0. The write value should be 0.	R/W
18	PSARE18	General PWM Timer Channel 13 Security Attribution Target module: GPT13 and the MSTPCRE.MSTPE18 bit 0: Secure 1: Non-secure	R/W
19	PSARE19	General PWM Timer Channel 12 Security Attribution Target module: GPT12 and the MSTPCRE.MSTPE19 bit 0: Secure 1: Non-secure	R/W
20	PSARE20	General PWM Timer Channel 11 Security Attribution Target module: GPT11 and the MSTPCRE.MSTPE20 bit 0: Secure 1: Non-secure	R/W
21	PSARE21	General PWM Timer Channel 10 Security Attribution Target module: GPT10 and the MSTPCRE.MSTPE21 bit 0: Secure 1: Non-secure	R/W
26:22	—	These bits are read as 0. The write value should be 0.	R/W
27	PSARE27	General PWM Timer Channel 4 Security Attribution Target module: GPT4 and the MSTPCRE.MSTPE27 bit 0: Secure 1: Non-secure	R/W
28	PSARE28	General PWM Timer Channel 3 Security Attribution Target module: GPT3 and the MSTPCRE.MSTPE28 bit 0: Secure 1: Non-secure	R/W
29	PSARE29	General PWM Timer Channel 2 Security Attribution Target module: GPT2 and the MSTPCRE.MSTPE29 bit 0: Secure 1: Non-secure	R/W
30	PSARE30	General PWM Timer Channel 1 Security Attribution Target module: GPT1 and the MSTPCRE.MSTPE30 bit 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
31	PSARE31	General PWM Timer Channel 0 Security Attribution Target module: GPT0, GPT_OPS and the MSTPCRE.MSTPE31 bit 0: Secure 1: Non-secure	R/W

Note: Access category: S-TYPE-1, P-TYPE-1.

The PSARE register specifies the security attribution for each module and the corresponding bit in Module Stop Control Register.

51.8.5 MSSAR : Module Stop Security Attribution Register

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MSSAR31	—	—	—	—	—	—	—	MSSAR23	MSSAR22	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MSSAR3	MSSAR2	MSSAR1	MSSAR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MSSAR0	SRAM0 Clock Stop Security Attribution Target module: the MSTPCRA.MSTPA0 bit 0: Secure 1: Non-secure	R/W
1	MSSAR1	SRAM1 Clock Stop Security Attribution Target module: the MSTPCRA.MSTPA1 bit 0: Secure 1: Non-secure	R/W
2	MSSAR2	SRAM2 Clock Stop Security Attribution Target module: the MSTPCRA.MSTPA2 bit 0: Secure 1: Non-secure	R/W
3	MSSAR3	SRAM3 Clock Stop Security Attribution Target module: the MSTPCRA.MSTPA3 bit 0: Secure 1: Non-secure	R/W
21:4	—	These bits are read as 0. The write value should be 0.	R/W
22	MSSAR22	DMAC0/DTC0 Clock Stop Security Attribution Target module: the MSTPCRA.MSTPA22 bit 0: Secure 1: Non-secure	R/W
23	MSSAR23	DMAC1/DTC1 Clock Stop Security Attribution Target module: the MSTPCRA.MSTPA23 bit 0: Secure 1: Non-secure	R/W
30:24	—	These bits are read as 0. The write value should be 0.	R/W
31	MSSAR31	ELC Clock Stop Security Attribution Target module: the MSTPCRC.MSTPC14 bit 0: Secure 1: Non-secure	R/W

Note: Access category: S-TYPE-1, P-TYPE-1.

The MSSAR register specifies the security attribution for the corresponding bit in Module Stop Control Register.

51.8.6 PPARB : Peripheral Privilege Attribution Register B

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x1C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PPARB31	PPARB30	PPARB29	PPARB28	PPARB27	PPARB26	PPARB25	PPARB24	PPARB23	PPARB22	—	—	PPARB19	PPARB18	PPARB17	PPARB16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	PPARB12	PPARB11	—	PPARB9	PPARB8	PPARB7	—	—	PPARB4	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 1. The write value should be 1.	R/W
4	PPARB4	I3C Bus Interface Privilege Attribution Target module: I3C and the MSTPCRB.MSTPB4 bit 0: Privileged 1: Unprivileged	R/W
6:5	—	These bits are read as 1. The write value should be 1.	R/W
7	PPARB7	I2C Bus Interface 2 Privilege Attribution Target module: IIC2 and the MSTPCRB.MSTPB7 bit 0: Privileged 1: Unprivileged	R/W
8	PPARB8	I2C Bus Interface 1 Privilege Attribution Target module: IIC1 and the MSTPCRB.MSTPB8 bit 0: Privileged 1: Unprivileged	R/W
9	PPARB9	I2C Bus Interface 0 Privilege Attribution Target module: IIC0 and the MSTPCRB.MSTPB9 bit 0: Privileged 1: Unprivileged	R/W
10	—	This bit is read as 1. The write value should be 1.	R/W
11	PPARB11	Universal Serial Bus 2.0 FS Interface 0 Privilege Attribution Target module: USBFS0 and the MSTPCRB.MSTPB11 bit 0: Privileged 1: Unprivileged	R/W
12	PPARB12	Universal Serial Bus 2.0 HS Interface 0 Privilege Attribution Target module: USBHS and the MSTPCRB.MSTPB12 bit 0: Privileged 1: Unprivileged	R/W
15:13	—	These bits are read as 1. The write value should be 1.	R/W
16	PPARB16	Octa Memory Controller 0 Privilege Attribution Target module: OSPI0 (+DOTF0) and the MSTPCRB.MSTPB16 bit 0: Privileged 1: Unprivileged	R/W
17	PPARB17	Octa Memory Controller 1 Privilege Attribution Target module: OSPI1 (+DOTF1) and the MSTPCRB.MSTPB17 bit 0: Privileged 1: Unprivileged	R/W

Bit	Symbol	Function	R/W
18	PPARB18	Serial Peripheral Interface 1 Privilege Attribution Target module: RSPI1 and the MSTPCRB.MSTPB18 bit 0: Privileged 1: Unprivileged	R/W
19	PPARB19	Serial Peripheral Interface 0 Privilege Attribution Target module: RSPI0 and the MSTPCRB.MSTPB19 bit 0: Privileged 1: Unprivileged	R/W
21:20	—	These bits are read as 1. The write value should be 1.	R/W
22	PPARB22	Serial Communication Interface 9 Privilege Attribution Target module: SCI9 and the MSTPCRB.MSTPB22 bit 0: Privileged 1: Unprivileged	R/W
23	PPARB23	Serial Communication Interface 8 Privilege Attribution Target module: SCI8 and the MSTPCRB.MSTPB23 bit 0: Privileged 1: Unprivileged	R/W
24	PPARB24	Serial Communication Interface 7 Privilege Attribution Target module: SCI7 and the MSTPCRB.MSTPB24 bit 0: Privileged 1: Unprivileged	R/W
25	PPARB25	Serial Communication Interface 6 Privilege Attribution Target module: SCI6 and the MSTPCRB.MSTPB25 bit 0: Privileged 1: Unprivileged	R/W
26	PPARB26	Serial Communication Interface 5 Privilege Attribution Target module: SCI5 and the MSTPCRB.MSTPB26 bit 0: Privileged 1: Unprivileged	R/W
27	PPARB27	Serial Communication Interface 4 Privilege Attribution Target module: SCI4 and the MSTPCRB.MSTPB27 bit 0: Privileged 1: Unprivileged	R/W
28	PPARB28	Serial Communication Interface 3 Privilege Attribution Target module: SCI3 and the MSTPCRB.MSTPB28 bit 0: Privileged 1: Unprivileged	R/W
29	PPARB29	Serial Communication Interface 2 Privilege Attribution Target module: SCI2 and the MSTPCRB.MSTPB29 bit 0: Privileged 1: Unprivileged	R/W
30	PPARB30	Serial Communication Interface 1 Privilege Attribution Target module: SCI1 and the MSTPCRB.MSTPB30 bit 0: Privileged 1: Unprivileged	R/W
31	PPARB31	Serial Communication Interface 0 Privilege Attribution Target module: SCI0 and the MSTPCRB.MSTPB31 bit 0: Privileged 1: Unprivileged	R/W

Note: Access category: S-TYPE-2, P-TYPE-1.

The PPARB register specifies the privilege attribution for each module and the corresponding bit in Module Stop Control Register.

51.8.7 PPARC : Peripheral Privilege Attribution Register C

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PPARC31	PPARC30	—	PPARC28	PPARC27	PPARC26	—	PPARC24	—	—	—	—	—	—	—	PPARC16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	PPARC13	PPARC12	PPARC11	—	—	PPARC8	PPARC7	—	—	—	—	—	PPARC1	PPARC0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	PPARC0	Clock Frequency Accuracy Measurement Circuit Privilege Attribution Target module: CAC and the MSTPCRC.MSTPC0 bit 0: Privileged 1: Unprivileged	R/W
1	PPARC1	Cyclic Redundancy Check Calculator Privilege Attribution Target module: CRC and the MSTPCRC.MSTPC1 bit 0: Privileged 1: Unprivileged	R/W
6:2	—	These bits are read as 1. The write value should be 1.	R/W
7	PPARC7	Serial Sound Interface Enhanced (Channel 1) Privilege Attribution Target module: SSIE1 and the MSTPCRC.MSTPC7 bit 0: Privileged 1: Unprivileged	R/W
8	PPARC8	Serial Sound Interface Enhanced (Channel 0) Privilege Attribution Target module: SSIE0 and the MSTPCRC.MSTPC8 bit 0: Privileged 1: Unprivileged	R/W
10:9	—	These bits are read as 1. The write value should be 1.	R/W
11	PPARC11	Secure Digital Host IF 1 Privilege Attribution Target module: SDHI1 and the MSTPCRC.MSTPC11 bit 0: Privileged 1: Unprivileged	R/W
12	PPARC12	Secure Digital Host IF 0 Privilege Attribution Target module: SDHI0 and the MSTPCRC.MSTPC12 bit 0: Privileged 1: Unprivileged	R/W
13	PPARC13	Data Operation Circuit Privilege Attribution Target module: DOC and the MSTPCRC.MSTPC13 bit 0: Privileged 1: Unprivileged	R/W
15:14	—	These bits are read as 1. The write value should be 1.	
16	PPARC16	CEU Privilege Attribution Target module: CEU and the MSTPCRC.MSTPC16 bit 0: Privileged 1: Unprivileged	R/W
23:17	—	These bits are read as 1. The write value should be 1.	R/W
24	PPARC24	Pulse Density Modulation Interface Privilege Attribution Target module: PDMIF and the MSTPCRC.MSTPC24 bit 0: Privileged 1: Unprivileged	R/W
25	—	This bit is read as 1. The write value should be 1.	R/W

Bit	Symbol	Function	R/W
26	PPARC26	Controller Area Network with Flexible Data-Rate 1 Privilege Attribution Target module: CANFD1 and the MSTPCRC.MSTPC26 bit 0: Privileged 1: Unprivileged	R/W
27	PPARC27	Controller Area Network with Flexible Data-Rate 0 Privilege Attribution Target module: CANFD0 and the MSTPCRC.MSTPC27 bit 0: Privileged 1: Unprivileged	R/W
28	PPARC28	Ether-PHY clock Privilege Attribution Target module: ETHPHYCLK and the MSTPCRC.MSTPC28 bit 0: Privileged 1: Unprivileged	R/W
29	—	This bit is read as 1. The write value should be 1.	R/W
30	PPARC30	Layer 3 Ethernet Switch Module Privilege Attribution Target module: ESWM and the MSTPCRC.MSTPC30 bit 0: Privileged 1: Unprivileged	R/W
31	PPARC31	RSIP-E50D Privilege Attribution Target module: RSIP-E50D and the MSTPCRC.MSTPC31 bit 0: Privileged 1: Unprivileged	R/W

Note: Access category: S-TYPE-2, P-TYPE-1.

The PPARC register specifies the privilege attribution for each module and the corresponding bit in Module Stop Control Register.

51.8.8 PPARD : Peripheral Privilege Attribution Register D

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x24

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	PPARD28	PPARD27	PPARD26	PPARD25	—	—	PPARD22	PPARD21	PPARD20	PPARD19	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	PPARD14	PPARD13	PPARD12	PPARD11	—	—	—	—	PPARD6	PPARD5	PPARD4	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 1. The write value should be 1.	R/W
4	PPARD4	Asynchronous General Purpose Timer 1 Privilege Attribution Target module: AGT1 and the MSTPCRD.MSTPD4 bit 0: Privileged 1: Unprivileged	R/W
5	PPARD5	Asynchronous General Purpose Timer 0 Privilege Attribution Target module: AGT0 and the MSTPCRD.MSTPD5 bit 0: Privileged 1: Unprivileged	R/W
6	PPARD6	PWM Delay Generation Circuit Privilege Attribution Target module: PDG and the MSTPCRD.MSTPD6 bit 0: Privileged 1: Unprivileged	R/W
10:7	—	These bits are read as 1. The write value should be 1.	R/W

Bit	Symbol	Function	R/W
11	PPARD11	Port Output Enable for GPT Group 3 Privilege Attribution Target module: PGI3 and the MSTPCRD.MSTPD11 bit 0: Privileged 1: Unprivileged	R/W
12	PPARD12	Port Output Enable for GPT Group 2 Privilege Attribution Target module: PGI2 and the MSTPCRD.MSTPD12 bit 0: Privileged 1: Unprivileged	R/W
13	PPARD13	Port Output Enable for GPT Group 1 Privilege Attribution Target module: PGI1 and the MSTPCRD.MSTPD13 bit 0: Privileged 1: Unprivileged	R/W
14	PPARD14	Port Output Enable for GPT Group 0 Privilege Attribution Target module: PGI0 and the MSTPCRD.MSTPD14 bit 0: Privileged 1: Unprivileged	R/W
18:15	—	These bits are read as 1. The write value should be 1.	R/W
19	PPARD19	12-Bit D/A Converter 1 Privilege Attribution Target module: DAC121 and the MSTPCRD.MSTPD19 bit 0: Privileged 1: Unprivileged	R/W
20	PPARD20	12-Bit D/A Converter 0 Privilege Attribution Target module: DAC120 and the MSTPCRD.MSTPD20 bit 0: Privileged 1: Unprivileged	R/W
21	PPARD21	16-Bit A/D Converter Privilege Attribution Target module: ADC16H and the MSTPCRD.MSTPD21 bit 0: Privileged 1: Unprivileged	R/W
22	PPARD22	Temperature Sensor Privilege Attribution Target module: TSN and the MSTPCRD.MSTPD22 bit 0: Privileged 1: Unprivileged	R/W
24:23	—	These bits are read as 1. The write value should be 1.	R/W
25	PPARD25	High speed analog Comparator 3 Privilege Attribution Target module: ACMPHS3 and the MSTPCRD.MSTPD25 bit 0: Privileged 1: Unprivileged	R/W
26	PPARD26	High speed analog Comparator 2 Privilege Attribution Target module: ACMPHS2 and the MSTPCRD.MSTPD26 bit 0: Privileged 1: Unprivileged	R/W
27	PPARD27	High speed analog Comparator 1 Privilege Attribution Target module: ACMPHS1 and the MSTPCRD.MSTPD27 bit 0: Privileged 1: Unprivileged	R/W
28	PPARD28	High speed analog Comparator 0 Privilege Attribution Target module: ACMPHS0 and the MSTPCRD.MSTPD28 bit 0: Privileged 1: Unprivileged	R/W
31:29	—	These bits are read as 1. The write value should be 1.	R/W

Note: Access category: S-TYPE-2, P-TYPE-1.

The PPARD register specifies the privilege attribution for each module and the corresponding bit in Module Stop Control Register.

51.8.9 PPARE : Peripheral Privilege Attribution Register E

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x28

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PPARE31	PPARE30	PPARE29	PPARE28	PPARE27	—	—	—	—	—	PPARE21	PPARE20	PPARE19	PPARE18	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PPARE15	—	—	—	—	—	PPARE9	PPARE8	—	—	—	—	PPARE3	PPARE2	PPARE1	PPARE0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	PPARE0	WDT1 Privilege Attribution Target module: WDT1 0: Privileged 1: Unprivileged	R/W
1	PPARE1	WDT0 Privilege Attribution Target module: WDT0 0: Privileged 1: Unprivileged	R/W
2	PPARE2	IWDT Privilege Attribution Target module: IWDT 0: Privileged 1: Unprivileged	R/W
3	PPARE3	Real Time Clock Privilege Attribution Target module: RTC 0: Privileged 1: Unprivileged	R/W
7:4	—	These bits are read as 1. The write value should be 1.	R/W
8	PPARE8	ULPT1 Privilege Attribution Target module: ULPT1 and the MSTPCRE.MSTPE8 bit 0: Privileged 1: Unprivileged	R/W
9	PPARE9	ULPT0 Privilege Attribution Target module: ULPT0 and the MSTPCRE.MSTPE9 bit 0: Privileged 1: Unprivileged	R/W
14:10	—	These bits are read as 1. The write value should be 1.	R/W
15	PPARE15	GPT common Privilege Attribution Target module: GPT GTCLKCR register 0: Privileged 1: Unprivileged	R/W
17:16	—	These bits are read as 1. The write value should be 1.	R/W
18	PPARE18	General PWM Timer Channel 13 Privilege Attribution Target module: GPT13 and the MSTPCRE.MSTPE18 bit 0: Privileged 1: Unprivileged	R/W
19	PPARE19	General PWM Timer Channel 12 Privilege Attribution Target module: GPT12 and the MSTPCRE.MSTPE19 bit 0: Privileged 1: Unprivileged	R/W

Bit	Symbol	Function	R/W
20	PPARE20	General PWM Timer Channel 11 Privilege Attribution Target module: GPT11 and the MSTPCRE.MSTPE20 bit 0: Privileged 1: Unprivileged	R/W
21	PPARE21	General PWM Timer Channel 10 Privilege Attribution Target module: GPT10 and the MSTPCRE.MSTPE21 bit 0: Privileged 1: Unprivileged	R/W
26:22	—	These bits are read as 1. The write value should be 1.	R/W
27	PPARE27	General PWM Timer Channel 4 Privilege Attribution Target module: GPT4 and the MSTPCRE.MSTPE27 bit 0: Privileged 1: Unprivileged	R/W
28	PPARE28	General PWM Timer Channel 3 Privilege Attribution Target module: GPT3 and the MSTPCRE.MSTPE28 bit 0: Privileged 1: Unprivileged	R/W
29	PPARE29	General PWM Timer Channel 2 Privilege Attribution Target module: GPT2 and the MSTPCRE.MSTPE29 bit 0: Privileged 1: Unprivileged	R/W
30	PPARE30	General PWM Timer Channel 1 Privilege Attribution Target module: GPT1 and the MSTPCRE.MSTPE30 bit 0: Privileged 1: Unprivileged	R/W
31	PPARE31	General PWM Timer Channel 0 Privilege Attribution Target module: GPT0 and the MSTPCRE.MSTPE31 bit 0: Privileged 1: Unprivileged	R/W

Note: Access category: S-TYPE-2, P-TYPE-1.

The PPARE register specifies the privilege attribution for each module and the corresponding bit in Module Stop Control Register.

51.8.10 MSPAR : Module Stop Privilege Attribution Register

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x2C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MSPAR31	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
30:0	—	These bits are read as 1. The write value should be 1.	R/W
31	MSPAR31	ELC Clock Stop Privilege Attribution Target module: the MSTPCRC.MSTPC14 bit 0: Privileged 1: Unprivileged	R/W

Note: Access category: S-TYPE-2, P-TYPE-1.

The MSPAR register specifies the privilege attribution for the corresponding bit in the Module Stop Control Register.

51.8.11 CMSAMON : Code MRAM Security Attribution Monitor Register

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x30

Bit position:	31								23								15															0
Bit field:	—	—	—	—	—	—	—	—	CMS[8:0]									—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	*1	*1	*1	*1	*1	*1	*1	*1	*1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
14:0	—	These bits are read as 0.	R
23:15	CMS[8:0]	Code MRAM Secure Area Indicate the area of secure region for code MRAM.	R
31:24	—	These bits are read as 0.	R

Note: Access category: S-TYPE-5, P-TYPE-5.
Note 1. The value in a blank product is 0x1FF. It is set to the value written by your application.

51.8.12 SFSAMON : SiP Flash Security Attribution Monitor Register

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x3C

Bit position:	31								23								15															0
Bit field:	—	—	—	—	—	—	—	—	SFS[8:0]									—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	*1	*1	*1	*1	*1	*1	*1	*1	*1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
14:0	—	These bits are read as 0.	R
23:15	SFS[8:0]	SiP Flash Secure Area Indicate the area of secure region for SiP flash.	R
31:24	—	These bits are read as 0.	R

Note: Access category: S-TYPE-5, P-TYPE-5.
Note 1. The value in a blank product is 0x1FF. It is set to the value written by your application.

51.8.13 DLMMON : Device Lifecycle Management State Monitor Register

Base address: PSCU = 0x4020_4000
PSCU_NS = 0x5020_4000

Offset address: 0x38

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	DLMMON[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1

Bit	Symbol	Function	R/W
3:0	DLMMON[3:0]	Device Lifecycle Management State Monitor Indicate DLM status value. 0x4: OEM 0x6: LCK_BOOT 0x7: RMA_REQ 0x8: RMA_ACK 0x9: RMA_RET Others: Reserved	R
31:4	—	These bits are read as 0.	R

Note: Access category: S-TYPE-5, P-TYPE-5.
Note 1. The value in a blank product is 0100b. These bits depend on DLM status.

51.8.14 MSAOAD : Master Security Attribution Operation After Detection Register

Base address: BUS = 0x4000_3000

Offset address: 0x1010

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	OAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OAD	Operation after detection 0: IRQ 1: Reset	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits enable or disable writing of the OAD bit.	W

Note: Access category: S-TYPE-6, P-TYPE-2.

OAD bit (Operation after detection)

The OAD bit specifies operation when the access violation is detected. When OAD = 0, error response is returned to the bus master, and IRQ request is generated if BUSIRQEN<Master Name>. EN bit is 1. Interrupts due to bus errors can be used as NMI if NMIER.BUSEN bit is 1. When OAD = 1, reset request is generated. When writing to the OAD bit, write 0xA5 simultaneously to the KEY[7:0] bits using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the OAD bit. When writing to the OAD bit, write 0xA5 simultaneously to the KEY[7:0] bits. If this value is not written to the KEY[7:0] bits, the OAD bit is not updated.

51.8.15 MSAPT : Master Security Attribution Protect Register

Base address: BUS = 0x4000_3000

Offset address: 0x1014

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: MSAOAD register writing is possible. 1: MSAOAD register writing is protected. Read is possible.	R/W

Bit	Symbol	Function	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits enable or disable writing of the PROTECT bit.	W

Note: Access category: S-TYPE-6, P-TYPE-2.

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writes to the associated registers to be protected.

MSAPT.PROTECT controls the MSAOAD register.

When the PROTECT bit is set simultaneously, write 0xA5 to the KEY[7:0] bits using half word access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits. If this value is not written to the KEY[7:0] bits, the PROTECT bit is not updated.

51.9 Usage Note

51.9.1 Security or Privilege Bit Write Timing

When writing a security or privilege attribution bit, confirm that writing is complete by reading the security or privilege register until it matches the value written. The protection is not effective until writing to the security or privilege register is complete.

52. Renesas Secure IP (RSIP-E50D)

52.1 Overview

This security engine consists of an isolated subsystem comprising an access management circuit, storage area, encryption/ decryption circuit, and random number generation circuit. In combination with the RSIP library, the security engine can prevent eavesdropping (to maintain confidentiality), falsification of information (to ensure integrity), and impersonation (to verify authenticity).

Because key information required for encryption and decryption is stored only in the security engine and all accesses from the outside can be blocked, the security engine enables building a more robust security system.

Table 52.1 lists the security engine specifications. Figure 52.1 shows the security engine block diagram.

Table 52.1 Security engine specifications (1 of 2)

Parameter	Specifications
Access control	Access management circuit ● In case of irregular access to the security engine due to a tampered program or CPU runaway, this circuit blocks all subsequent accesses and stops data output from the security engine. Interrupt RSIP_TADI is generated.
Symmetric cryptography	AES: Compliant with NIST FIPS PUB 197 ● Key length: 128, 192, or 256 bits ● Data block size: 128 bits ● AES supports the following block cipher mode: – ECB, CBC, CTR: Compliant with NIST SP 800-38A – CCM: Compliant with NIST SP 800-38C – GCM: Compliant with NIST SP 800-38D – XTS: Compliant with IEEE 1619-2007 ● AES supports the following authentication algorithms: – CMAC: Compliant with NIST SP 800-38B – GMAC: Compliant with NIST SP 800-38D ChaCha20-Poly1305: Compliant with RFC7539 ● Key length: 256 bits ● Block size: 512 bits ● Support the following algorithms: – ChaCha20 – ChaCha20-Poly1305
Random number generation	128-bit true random number generation circuit
Asymmetric cryptography	RSA ● Maximum number of operable bits: 4224 bits ● Supported key sizes: 1024 bits, 2048 bits, 3072 bits, and 4096 bits ● Signature generation, signature verification, public-key encryption, private-key decryption. ECC ● Maximum number of operable bits: 576 bits ● Support for curve – NIST P-192, P-224, P-256, P-384, and P-521 – Brain pool P256r1, P384r1, and P512r1 – Ed25519 – secp256k1 ● Signature generation, signature verification, key generation.
Message digest computation	HASH ● Block size: 512 bits (SHA-224, SHA-256) 1024 bits (SHA-512/224, SHA-512/256, SHA-384, SHA-512) 1152 bits (SHA3-224), 1088 bits (SHA3-256), 832 bits (SHA3-384), 576 bits (SHA3-512), 1344 bits (SHAKE128), 1088 bits (SHAKE256) ● Key size: 512 bits or less ● HASH supports the following secure hash algorithms: – SHA-224, SHA-256, SHA-512/224, SHA-512/256, SHA-384, and SHA-512: Compliant with FIPS PUB 180-4 – SHA3-224, SHA3-256, SHA3-384, SHA3-512, SHAKE128, and SHAKE256: Compliant with FIPS PUB 202 ● HASH supports the following message authentication algorithm: – HMAC: Compliant with FIPS PUB 198

Table 52.1 Security engine specifications (2 of 2)

Parameter	Specifications
Hardware Unique Key	● A read-only, 256-bit Hardware Unique Key (HUK) is exclusively accessible by the security engine access management circuit through a dedicated bus. ● Key derivation functions (KDFs) combine the Hardware Unique Key with the key generation information. The derived keys implement the key wrapping for user key secure storage. ● The HUK uniqueness prevents the illicit cloning and copying of keys to another MCU of the MCU group. ● The HUK itself is stored in wrapped (encrypted, non-plain) format in an isolated memory area. Therefore, it is protected from illicit access and copy.
Application Key Management	● Wrapped keys are only valid within the security engine.
Unique ID	● A read-only, 128-bit ID unique to an MCU (Unique ID) is accessible from the access management circuit. ● Key derivation functions (KDFs) combine the Unique ID with the key generation information. Such derived keys are used to unwrap the HUK within the security engine boundary.
OEM boot loader version	● The security engine outputs the current OEM boot loader version (OEM BL Ver.) after successful validation of a firmware update. ● MRAM sequencer supports Anti-Rollback Counter Protection using the OEM boot loader version, which can be validated using the security engine.
Decryption-on-the-fly	The security engine outputs the key data of the decryption-on-the-fly IP (DOTF) through a dedicated bus. The key can be set separately for DOTF0 and DOTF1.
Tamper Resistance	Countermeasures available for side-channel attacks, including SPA/DPA and timing attacks.
Module-stop function	Module-stop state can be set to reduce power consumption.
TrustZone Filter	Security and Privilege attribution can be set.

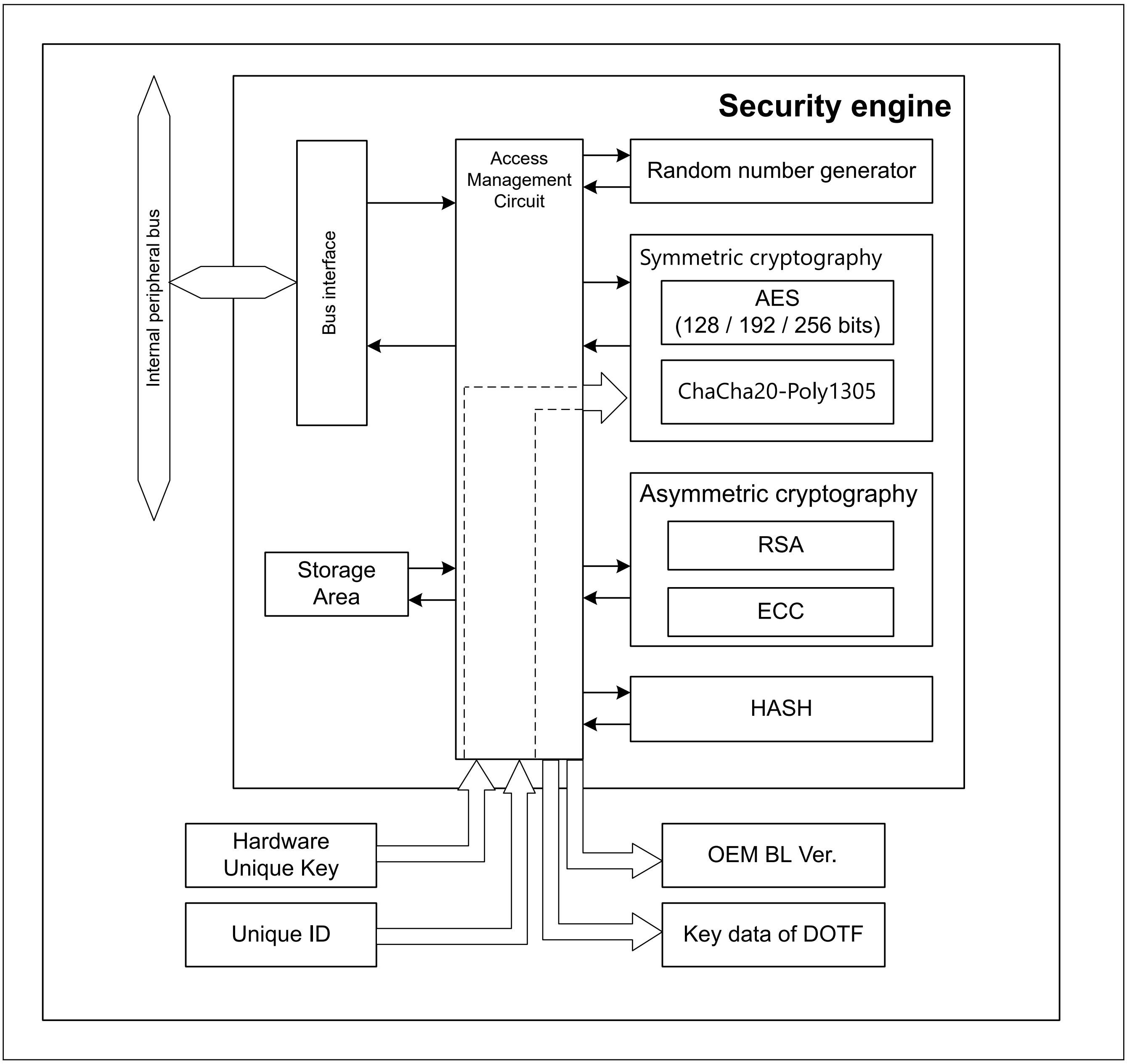


Figure 52.1 Security engine block diagram

52.2 Operation

52.2.1 Symmetric Cryptography

Figure 52.2 shows conceptual block diagram of the symmetric cryptography installed in the security engine.

The symmetric cryptography uses the input key information and converts the plaintext data to ciphertext or ciphertext data to plaintext through the hardware.

The encryption/decryption process can be completed without exposing the key data and the intermediate data of the process outside of the security engine. This process is performed by the encryption/decryption circuit and storage area of the security engine internally.

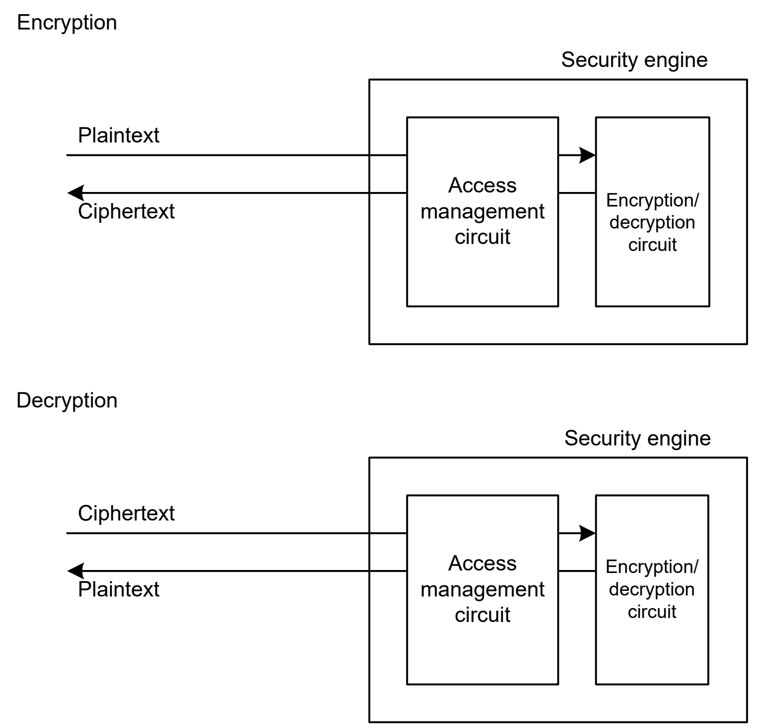


Figure 52.2 Conceptual diagram of the symmetric cryptography

52.2.2 Asymmetric Cryptography

Figure 52.3 shows conceptual block diagram of the asymmetric cryptography installed in the security engine.

The asymmetric cryptography uses the input key information and converts the plaintext data to ciphertext or ciphertext data to plaintext or generates/verifies the signature.

The encryption/decryption or signature generation/verification process can be completed without exposing the key data and the process's intermediate data outside of the security engine. This process is performed by the cryptographic circuit and storage area of the security engine internally.

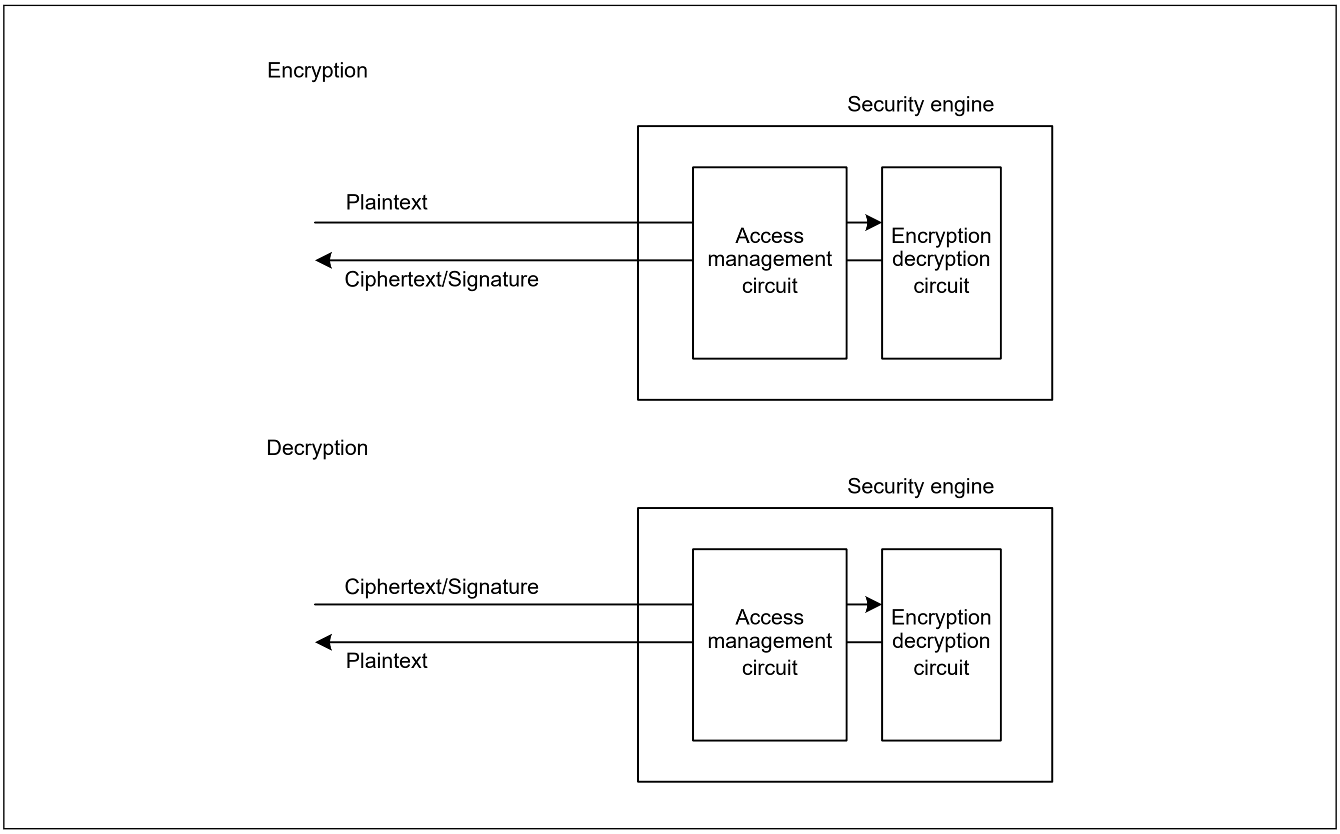


Figure 52.3 Conceptual diagram of the asymmetric cryptography

52.2.3 Hash Generator

Figure 52.4 shows conceptual block diagram of the hash generator installed in the security engine.

The hash generator generates uses the input key information (if required for the hash calculation) and generates the message digest or tag.

The calculation process can be completed without exposing any key data and the intermediate data of the process outside of the security engine. This process is performed by the hash circuit and storage area of the security engine internally.

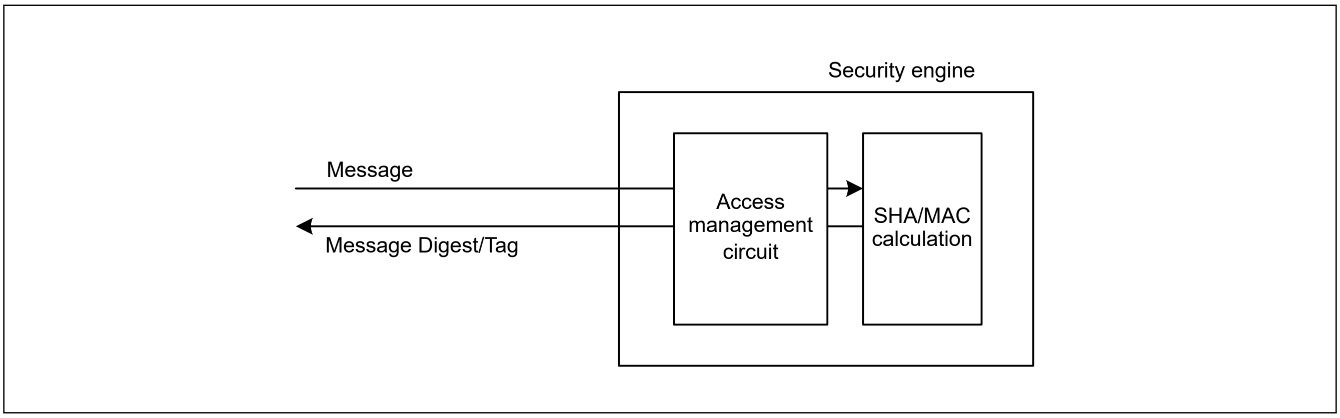


Figure 52.4 Conceptual diagram of the hash generator

52.2.4 Encryption and Decryption

Follow the procedure below to encrypt and decrypt the data:

1. Input the key information to the security engine.
2. Input the target data to the security engine. Plaintext data is converted to ciphertext and ciphertext data to plaintext.
3. Read the converted data.

The encryption/decryption circuit has input and output buffers, and can perform encryption/decryption in parallel with data input/output. Figure 52.5 shows the encryption engine timing.

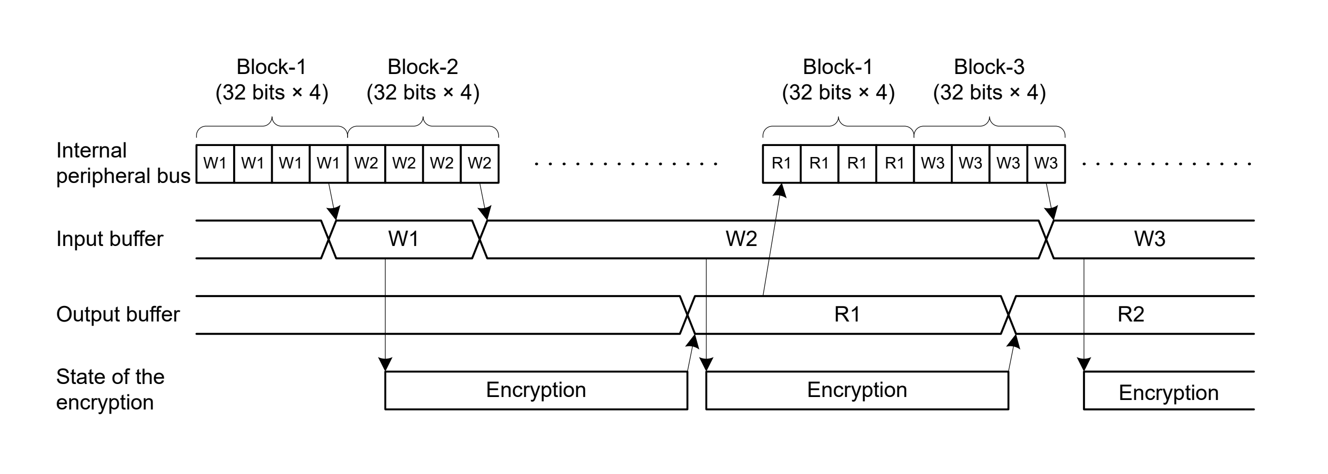


Figure 52.5 Encryption and decryption timing (AES)

52.3 Usage Notes

52.3.1 Software Standby Mode

When Software Standby mode is entered while the security engine is in process, proper processing cannot be resumed after cancelling Software Standby mode. Software Standby mode should therefore be entered while the security engine is not running.

52.3.2 Module-Stop Function Setting

The security engine operation can be disabled or enabled using Module Stop Control Register C (MSTPCRC). The security engine module is initially stopped after reset. Releasing the module-stop state enables access to the registers.

52.3.3 SPA/DPA Protections

SPA/DPA protections can optionally be enabled for encryption/decryption processing.

52.3.4 Restrictions on Open Process

While open process of the API of security engine (R_RSIP_Open) is running, the following restriction applies:

- The CPUCLK0 frequency and the ICLK frequency must be the same.

53. 16-bit A/D Converter (ADC16H)

This is the ADC_B version of the ADC16H peripheral module. ADC_B is referred to as ADC16H in this chapter.

53.1 Overview

This MCU contains two units of Noise-Shaping SAR-type A/D converters (ADC16H) that is the hybrid architecture with features of SAR-type and Delta-Sigma modulation-type. The A/D converter unit 0 (ADC0) can select up to 15 channels of analog inputs. The A/D converter unit 1 (ADC1) can select up to 15 channels of analog inputs. The temperature sensor, internal reference voltage, VBATT 1/6 voltage monitor output, and D/A converters can be A/D-converted by A/D converter unit 0 or unit 1. A/D conversion data can be selected from 16-bit, 14-bit, 12-bit, and 10-bit length data formats.

The ADC16H has the following features:

- Resolution: Up to 16-bit
- Fast conversion: Up to 6.25 Msps (0.16 µs per channel) (when A/D conversion clock ADCLK = 50 MHz)
- Input channels: Up to 23 analog input channels
- Support single-ended input or differential inputs
- Self-calibration function
- Built-in channel-dedicated sample-and-hold circuit (S&H)

Table 53.1 shows the ADC16H specifications, Table 53.2 shows the correspondence between ADC16H functions and ADC16H operation mode, and Table 53.3 shows the ADC16H I/O pins.

Figure 53.1 shows the block diagram of ADC16H. Figure 53.2 shows the analog channel structure for A/D converter unit 0. Figure 53.3 shows the analog channel structure for A/D converter unit 1.

Table 53.5 shows the ADC16H channel configuration of the extended analog function.

Table 53.1 ADC16H specifications (1 of 3)

Item	Description
Number of units	Two units (unit 0 and unit 1)
Input channels	● Support single-ended input or differential inputs ● Up to 23 analog input channels – A/D converter unit 0: Up to 15 analog input channels at Single-ended input, 6 analog input channels (3 pairs) of which support Differential inputs. – A/D converter unit 1: Up to 15 analog input channels at Single-ended input, 6 analog input channels (3 pairs) of which support Differential inputs. – 7 analog input channels are shared by A/D converter unit 0 and unit 1
Extended analog function	Self-diagnosis, temperature sensor, internal reference voltage, D/A converter (DA0, DA1), VBATT 1/6 voltage monitor output
A/D conversion method	● Successive approximation register method in SAR mode ● Noise-Shaping Successive approximation register method in Oversampling mode and Hybrid mode
Resolution of A/D converter	● Up to 12-bits in SAR mode ● Up to 16-bits in Oversampling mode or Hybrid mode
Conversion time	0.16 µs per channel (when A/D conversion clock ADCLK = 50 MHz).
A/D conversion clock	The A/D conversion clock (ADCLK) can be set by selecting the clock source and the division ratio as follows: ● The clock source: the ADC clock ADCCLK*1, the peripheral module clock PCLKA*1, the GPT clock GPTCLK*1. ● The division ratio: 1/2/3/4/5 A/D conversion clock (ADCLK) can operate between 25 MHz at a minimum and 60 MHz at a maximum.
A/D conversion data	● A/D conversion results are stored in data register or FIFO. ● A/D conversion results are available in 16-, 14-, 12-, and 10-bit data formats.

Table 53.1 ADC16H specifications (2 of 3)

<table>
<tr><th>Item</th><th>Description</th></tr>
<tr><td>Operation mode</td><td>
• SAR mode
– A/D converter samples the signal source once, and convert by Successive Approximation Register method
– Fast A/D conversion
– Support up to 8 channels per one scan group
– Support single-ended input and differential input
• Oversampling mode
– A/D converter oversamples the signal source, and converts analog-to-digital by Noise Shaping Successive Approximation Register method
– High-accuracy A/D conversion
– Support up to 8 channels per one scan group
– Support single-ended input and differential input
• Hybrid mode
– A/D converter oversamples the signal source, and converts analog-to-digital by Noise Shaping Successive Approximation Register method
– High-accuracy A/D conversion
– High data rate in continuous scanning operation
– Support up to 4 channels per one scan group
– Support single-ended input and differential input
</td></tr>
<tr><td>Scan mode</td><td>
• Single scan mode
– Assign any selected analog input or analog channel of the extended analog function to any scan group, and convert the selected analog input only once per scan group for A/D conversion.
– By selecting the scan start conditions for each scan group individually, A/D conversion for each scan group can be started at different times.
• Continuous scan mode
– Assign any selected analog input or analog channel of the extended analog function to any scan group and repeat A/D conversion in scan group units.
• Background continuous scan mode
– Assign any selected analog input or analog channel of the extended analog function to any scan group and repeat A/D conversion in scan group units.
– The A/D conversion is performed in the background and, if the A/D conversion start trigger is entered, the A/D conversion data at that point is acquired.
• One-channel continuous scan mode
– Repeat A/D conversion of a single analog input or analog channel of an extended analog function.
– If the A/D conversion start trigger is entered, the channel to be converted can be switched.
– This conversion mode is suitable for use cases where a change to the A/D conversion channel is required according to an external factor.
• Fixed-channel continuous scan mode
– First scan a fixed number of fixed channels, then scan the non-fixed channels assigned to the scan group in sequence, and repeat these operations.
– Enabling adjustment of the A/D conversion frequency and A/D converts more channels than in normal Hybrid mode.
</td></tr>
<tr><td>Combination of operation mode and scan mode</td><td>
<table>
<tr><th rowspan="2">Scan mode</th><th colspan="3">Operation mode of A/D converter</th></tr>
<tr><th>SAR mode</th><th>Oversampling mode</th><th>Hybrid mode</th></tr>
<tr><td>Single scan mode</td><td>✓</td><td>✓</td><td>✓</td></tr>
<tr><td>Continuous scan mode</td><td>✓</td><td>✓</td><td>✓</td></tr>
<tr><td>Background continuous scan mode</td><td>—</td><td>—</td><td>✓</td></tr>
<tr><td>One-channel continuous scan mode</td><td>—</td><td>✓</td><td>—</td></tr>
<tr><td>Fixed-channel continuous scan mode</td><td>—</td><td>—</td><td>✓</td></tr>
</table>
Note: ✓: available, —: not available
</td></tr>
<tr><td>Conditions for A/D conversion start</td><td>
• Software trigger (for simultaneous activation of scan group: max. 9 triggers)
• Software trigger (for individual scan group activation: up to 9 triggers)
• Trigger from Event Link Controller: 6 triggers
• Triggers from GPT: 28 triggers
• External trigger input: 2 triggers (ADTRGn input (n = 0, 1))
</td></tr>
</table>

Table 53.1 ADC16H specifications (3 of 3)

Item	Description
Functions	● Virtual Channel function (33 virtual channels) ● Scan Group function (up to 9 scan groups) ● Variable sampling time (select from 16 tables per virtual channel) ● Channel-dedicated sample-and-hold circuit (SH) (3 SH units for A/D converter unit 0, 3 SH units for A/D converter unit 1) ● Self-diagnosis function for A/D converter ● Selectable A/D-converted value addition mode or average mode ● Analog input disconnection detection assist function (discharge function/precharge function) ● Selectable the data format from 16- / 14- / 12- / 10-bit. ● Selectable the signed-data or the unsigned-data format ● Limiter Clip function (up to 8 tables) ● Compare Match function (up to 8 tables) ● Self-calibration function ● User's Gain adjustment function ● User's Offset adjustment function ● Built-in FIFO (8 stages per each scan group) ● Built-in digital filter (support only Oversampling mode and Hybrid mode) ● Multiple A/D converters unit-to-unit synchronous operation function
Interrupt sources	● A/D scan end interrupt – Generates the interrupt requests and the ELC events at the end of A/D scan operation for the scan group i (ADC_ADIi (i = 0 to 8)). The interrupt requests are independent for each scan group. ● FIFO data read request interrupt – Generates the interrupt requests when the number of vacant stages of FIFO for the scan group i becomes less than or equal to the specified value (ADC_FIFOREQi (i = 0 to 8)). The interrupt requests are independent for each scan group. ● FIFO data overflow interrupt – Generates the interrupt request when the overflow occurs in any of the FIFO for the scan groups 0 to 8 (ADC_FIFOOVF). ● Limiter clip interrupt – Generates the interrupt request when the limiter clip using the limiter tables 0 to 7 is occurred for A/D conversion results (ADC_LIMCLPI). ● Compare match interrupt – Generates the interrupt requests when the compare match using the compare match tables j are occurred for A/D conversion results (ADC_CMPIj (j = 0 to 3)). The interrupt requests are independent for each compare match table. ● Composite compare match interrupt – Generates the interrupt requests and the ELC events when the compare match of the composite condition using compare match tables 0 to 7 occurs (ADC_CCMPMm (m = 0, 1), ADC_CCMPUMm (m = 0, 1)). ● A/D converter error interrupt – Generates the interrupt requests when the operational error is detected in A/D converter Unit j (ADC_ERRj (j = 0, 1)). ● A/D conversion overflow interrupt – Generates the interrupt request when the A/D conversion result overflow is occurred (ADC_RESOVFj (j = 0, 1)). The interrupt requests are independent for A/D converter unit j. ● A/D converter calibration end interrupt – Generates the interrupt requests at the end of calibration operation of A/D converter unit j (ADC_CALENDj (j = 0, 1)).
ELC interface	● Trigger Input – Scan can be started by the trigger from the ELC. ● Event Generation – An event is generated at the end of each scan operation for the scan group 0 to 8. – An event is generated when Composite Compare Match is occurred.
Reference voltage	● Unit 0: – VREFH0 is the analog reference voltage. – VREFL0 is the analog reference ground. ● Unit 1: – VREFH is the analog reference voltage. – VREFL is the analog reference ground.
Module-stop function	Module-stop state can be set to reduce power consumption.*2

Note 1. For details, see section 9, Clock Generation Circuit.
Note 2. For details, see section 11, Low Power Mode.

Table 53.2 Association between ADC16H functions and ADC16H operation mode (1 of 2)

Function		SAR mode		Oversampling mode			Hybrid mode			
		Single scan mode	Continuous scan mode	Single scan mode	Continuous scan mode	One-channel continuous scan mode	Single scan mode	Continuous scan mode	Background continuous scan mode	Fixed-channel continuous scan mode
Analog input	Single-ended input	✓	✓	✓	✓	—	✓	✓	✓	—
	Differential input	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Channel-dedicated sample-and-hold circuit	✓	—	—	—	—	✓	✓	✓	✓*3
Diagnosis function/extended analog function	Disconnection detection assist function	✓	✓	✓	✓	✓	✓	✓	—	—
	Self-diagnosis	✓	✓	✓	—	—	✓	—	—	—
	Disconnection detection assist function of channel-dedicated sample-and-hold circuit	✓	—	—	—	—	—	—	—	—
	Self-diagnosis of channel-dedicated sample-and-hold circuit	✓	—	—	—	—	—	—	—	—
	Internal reference voltage	✓	✓	✓	✓	✓	✓	✓	✓	✓
	VBATT 1/6 voltage monitor output	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Temperature sensor	✓	✓	✓	✓	✓	✓	✓	✓	✓
	D/A converter (DA0, DA1)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Scanning operation	Group priority operation	✓*2	✓*2	—	—	—	—	—	—	—
	Multiple A/D converter synchronous operation	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 53.2 Association between ADC16H functions and ADC16H operation mode (2 of 2)

Function		SAR mode		Oversampling mode			Hybrid mode			
		Single scan mode	Continuous scan mode	Single scan mode	Continuous scan mode	One-channel continuous scan mode	Single scan mode	Continuous scan mode	Background continuous scan mode	Fixed-channel continuous scan mode
Digital calculation	Digital filter function	—	—	✓*1	✓*1	✓*1	✓*1	✓*1	✓*1	✓*1
	User’s gain/ offset adjustment function	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Addition/average function	✓	✓	✓	✓	✓	✓	✓	✓	—
	Limiter clip function	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Compare match function	✓	✓	✓	✓	✓	✓	✓	✓	✓
FIFO function		✓	✓	✓	✓	✓	✓	✓	✓	✓

Note 1. Use of the digital filter function is required when in Oversampling mode and Hybrid mode.
Note 2. When using the channel-dedicated sample-and-hold circuit in group priority operation, it is available only for the highest priority scan group.
Note 3. Only fixed channels are supported when the number of fixed channels is set to 3 CH and all fixed channels are set to channel-dedicated sample-and-hold circuit. Non-fixed channels cannot use channel-dedicated sample-and-hold circuit.

Table 53.3 ADC16H I/O pins

Pin name	I/O	Function
AVCC0	Input	Analog block power supply pin for unit 0 and unit 1
AVSS0	Input	Analog block power supply ground pin for unit 0 and unit 1
VREFH0	Input	Analog reference voltage supply pin for unit 0
VREFL0	Input	Analog reference ground pin for unit 0
VREFH	Input	Analog reference voltage supply pin for unit 1
VREFL	Input	Analog reference ground pin for unit 1
AN000 to AN022	Input	Analog input pins
SHIN0P to SHIN2P, SHIN4P to SHIN6P	Input	Analog input pins for channel-dedicated sample-and-hold circuit ● For single-ended input: Connect to signal source ● For differential input: Functions as the non-inverting inputs (+) terminal for differential input
SHIN0N to SHIN2N, SHIN4N to SHIN6N	Input	Analog input pins for channel-dedicated sample-and-hold circuit ● For single-ended input: Connect to signal source ● For differential input: Functions as the inverting inputs (-) terminal for differential input
ADTRG0, ADTRG1	Input	External trigger input pin for starting A/D conversion, active-low.
ADST0, ADST1	Output	A/D conversion start
AD0FLAG1, AD1FLAG1	Output	A/D conversion end
ADSYNC	Output	Synchronization signal between units

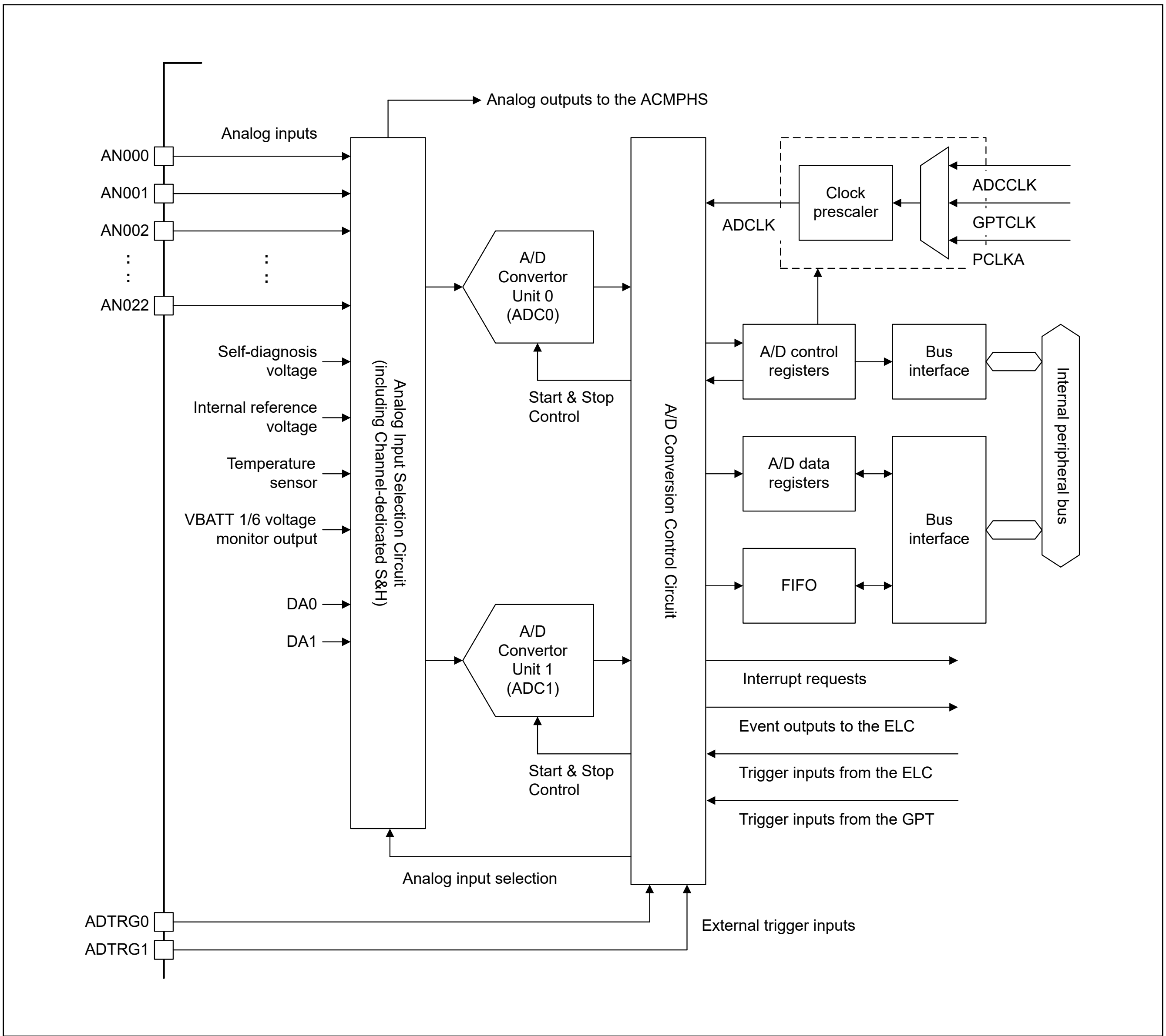


Figure 53.1 Block diagram of ADC16H

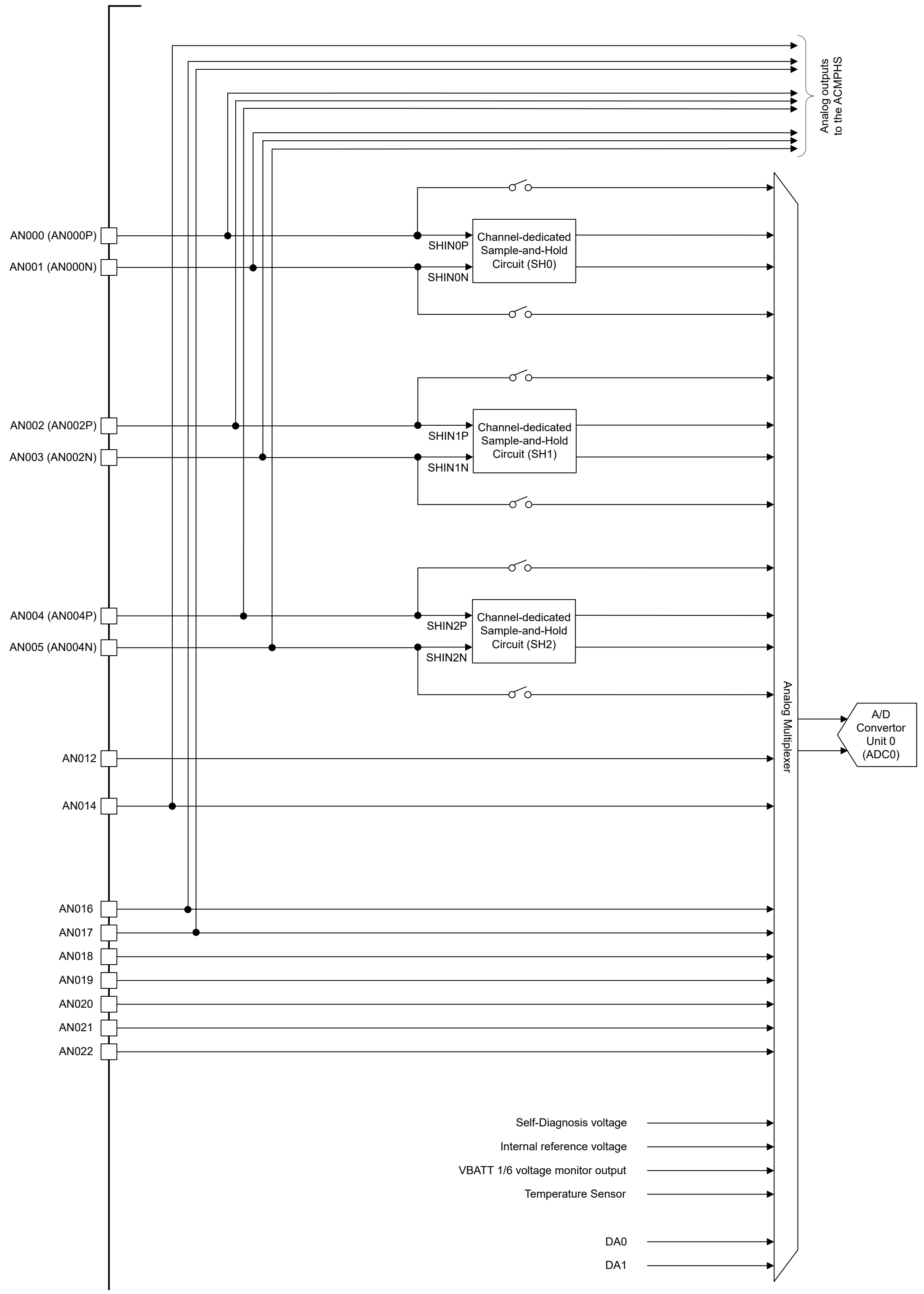


Figure 53.2 Analog channel structure for A/D converter unit 0

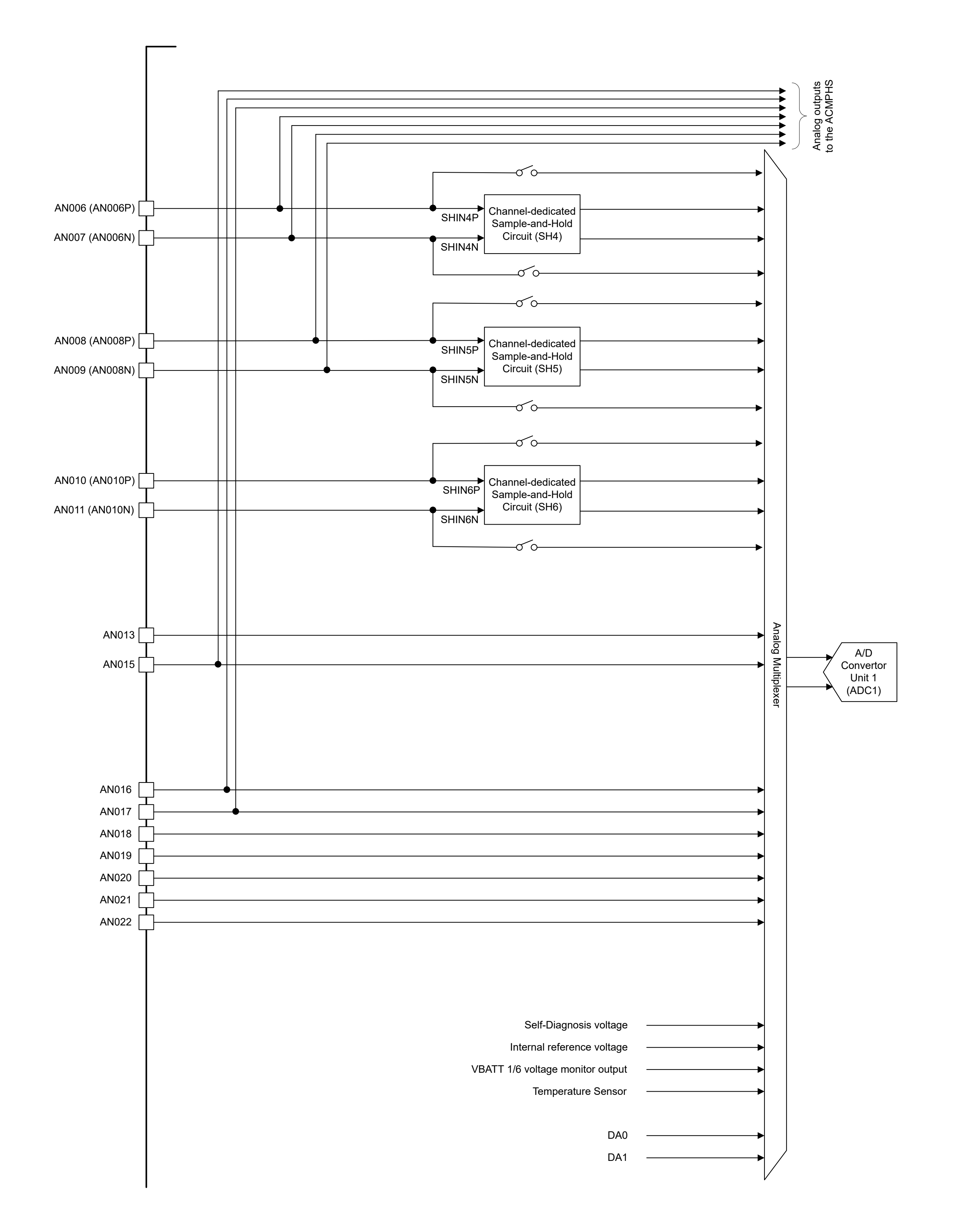


Figure 53.3 Analog channel structure for A/D converter unit 1

Table 53.4 shows the ADC16H channel configuration of the analog inputs.

Table 53.4 ADC16H channel configuration of the analog inputs

Analog channel number	Source of analog signal		A/D converter unit 0		A/D converter unit 1	
	Analog input	Channel-dedicated Sample-and-hold Circuit (S&H)[*2]	Single-ended input	Differential input[*3]	Single-ended input	Differential input[*3]
0	AN000 (AN000P)	SHIN0	✓	✓	—	—
1	AN001 (AN000N)		✓		—	
2	AN002 (AN002P)	SHIN1	✓	✓	—	—
3	AN003 (AN002N)		✓		—	
4	AN004 (AN004P)	SHIN2	✓	✓	—	—
5	AN005 (AN004N)		✓		—	
6	AN006 (AN006P)	SHIN4	—	—	✓	✓
7	AN007 (AN006N)		—		✓	
8	AN008 (AN008P)	SHIN5	—	—	✓	✓
9	AN009 (AN008N)		—		✓	
10	AN010 (AN010P)	SHIN6	—	—	✓	✓
11	AN011 (AN010N)		—		✓	
12	AN012	—	✓	—	—	—
13	AN013	—	—	—	✓	—
14	AN014	—	✓	—	—	—
15	AN015	—	—	—	✓	—
16	AN016[*1]	—	✓	—	✓	—
17	AN017[*1]	—	✓	—	✓	—
18	AN018[*1]	—	✓	—	✓	—
19	AN019[*1]	—	✓	—	✓	—
20	AN020[*1]	—	✓	—	✓	—
21	AN021[*1]	—	✓	—	✓	—
22	AN022[*1]	—	✓	—	✓	—

Note 1. Do not perform A/D conversions of the same signal source from both A/D converter unit 0 and unit 1.
Note 2. When the channel-dedicated sample-and-hold circuit is used in single-ended input, the SHINxP (non-inverting input (+)) channel and the SHINxN (inverting input (−)) channel must not be assigned to the same scan group (x = 0 to 2, 4 to 6).
Note 3. When performing A/D conversion with differential-input, set the even-numbered channel as the A/D conversion target. For differential input, the even-numbered channels function as non-inverting inputs (+), and the odd-numbered channels function as inverting inputs (−).

Table 53.5 ADC16H channel configuration of the extended analog function (1 of 2)

Analog channel number	Source of analog signal	A/D converter unit 0		A/D converter unit 1	
		Single-ended input	Differential input	Single-ended input	Differential input
96	Self-diagnosis A/D unit 0[*2]	—	✓	—	—
97	Self-diagnosis A/D unit 1[*2]	—	—	—	✓
100	Temperature sensor[*1]	✓	—	✓	—
101	Internal reference voltage[*1]	✓	—	✓	—
102	VBATT 1/6 voltage monitor output[*1]	✓	—	✓	—
104	D/A converter channel 0[*1]	✓	—	✓	—
105	D/A converter channel 1[*1]	✓	—	✓	—

Table 53.5 ADC16H channel configuration of the extended analog function (2 of 2)

Analog channel number	Source of analog signal	A/D converter unit 0		A/D converter unit 1	
		Single-ended input	Differential input	Single-ended input	Differential input
112	Self-diagnosis S&H unit 0*2	—	✓	—	—
113	Self-diagnosis S&H unit 1*2	—	✓	—	—
114	Self-diagnosis S&H unit 2*2	—	✓	—	—
116	Self-diagnosis S&H unit 4*2	—	—	—	✓
117	Self-diagnosis S&H unit 5*2	—	—	—	✓
118	Self-diagnosis S&H unit 6*2	—	—	—	✓

Note 1. Do not perform A/D conversions of the same signal source from both A/D converter unit 0 and unit 1.
Note 2. The self-diagnosis function is only supported with differential input.

53.2 Register Descriptions

53.2.1 System

53.2.1.1 ADCLKENR : A/D Conversion Clock Enable Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CLKEN	ADCLK Operating Enable 0: Stop ADCLK 1: Supply ADCLK	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The ADCLKENR controls the supply of the A/D conversion clock (ADCLK).

53.2.1.2 ADCLKSR : A/D Conversion Clock Status Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKSR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CLKSR	ADCLK status 0: ADCLK is stopped 1: ADCLK is in supply	R
31:1	—	These bits are read as 0.	R

The ADCLKSR register indicates the supply status of the A/D conversion clock (ADCLK).

53.2.1.3 ADCLKCR : A/D Conversion Clock Control Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	DIVR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CLKSEL[1:0]	ADCLK Clock Source Select 0 0: ADC Clock (ADCCLK) 0 1: GPT Clock (GPTCLK) 1 0: Peripheral Module Clock A (PCLKA) 1 1: Setting prohibited	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
18:16	DIVR[2:0]	Clock Division Ratio Select 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/3 0 1 1: 1/4 1 0 0: 1/5 Others: Setting prohibited	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

The ADCLKCR register selects the frequency of A/D conversion clock (ADCLK). The frequency of ADCLK should be set within the range specified in the section 62, Electrical Characteristics.

CLKSEL[1:0] bits (ADCLK Clock Source Select)

CLKSEL[1:0] bits select the clock source of A/D conversion clock (ADCLK).

DIVR[2:0] bits (Clock Division Ratio Select)

DIVR[2:0] bits select the division ratio for the clock source selected by CLKSEL[1:0] bits. The clock divided by the clock source selected by CLKSEL[1:0] bits at the ratio set by DIVR[2:0] bits becomes the A/D conversion operating clock (ADCLK).

53.2.1.4 ADSYCR : A/D Converter Synchronous Operation Control Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x00C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADSYDIS1	ADSYDIS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	ADSYCYC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Function	R/W
7:0	ADSYCYC[7:0]	Synchronous Operation Period Cycle These bits should be set in the range from 5 to 255. The settings of 0 to 4 are prohibited.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
16	ADSYDIS0	ADC0 Synchronous Operation Select 0: Enable ADC0 synchronous operation 1: Disable ADC0 synchronous operation	R/W
17	ADSYDIS1	ADC1 Synchronous Operation Select 0: Enable ADC1 synchronous operation 1: Disable ADC1 synchronous operation	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

The ADSYCR register controls the synchronous operation function.

ADSYCYC[7:0] bits (Synchronous Operation Period Cycle)

These bits are selected the synchronous operation period cycles.

These bits set in the range from 5 to 255 regardless of whether the synchronous operation function is used or not. The settings from 0 to 4 are prohibited.

ADSYDIS0 bit (ADC0 Synchronous Operation Select)

ADSYDIS0 bit selects enable/disable the synchronous operation of ADC0. If the synchronous operation is enabled, the ADC0 operates synchronously with other A/D converter that has synchronous operation enabled.

ADSYDIS1 bit (ADC1 Synchronous Operation Select)

ADSYDIS1 bit selects enables/disables to the synchronous operation of ADC1. If the synchronous operation is enabled, the ADC1 operates synchronously with other A/D converter that has synchronous operation enabled.

53.2.1.5 ADUSLPCR0 : A/D Converter Unit Sleep Control Register 0

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADUSLP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ADUSLP	Sleep setting of A/D converter unit 0 0: Set A/D converter to Normal operation mode 1: Set A/D converter to Sleep operation mode	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

This register is used to control the A/D converter unit 0 Sleep mode.

53.2.1.6 ADUSLPCR1 : A/D Converter Unit Sleep Control Register 1

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADUSLP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ADUSLP	Sleep setting of A/D converter unit 1 0: Set A/D converter to Normal operation mode 1: Set A/D converter to Sleep operation mode	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

This register is used to control the A/D converter unit 1 Sleep mode.

53.2.1.7 ADERINTCR : A/D Conversion Error Interrupt Enable Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x020

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADEIE1	ADEIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ADEIE0	ADC0 A/D Conversion Error Interrupt Enable 0: Disable A/D conversion error interrupt for ADC0 1: Enable A/D conversion error interrupt for ADC0	R/W
1	ADEIE1	ADC1 A/D Conversion Error Interrupt Enable 0: Disable A/D conversion error interrupt for ADC1 1: Enable A/D conversion error interrupt for ADC1	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

The ADERINTCR register controls the enable/disable to the A/D conversion error interrupt.

53.2.1.8 ADOVFINTCR : A/D Conversion Overflow Interrupt Enable Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x024

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADOVFIE1	ADOVFIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ADOVFIE0	ADC0 A/D Conversion Overflow Interrupt Enable 0: Disable A/D conversion overflow interrupt for ADC0 1: Enable A/D conversion overflow interrupt for ADC0	R/W
1	ADOVFIE1	ADC1 A/D Conversion Overflow Interrupt Enable 0: Disable A/D conversion overflow interrupt for ADC1 1: Enable A/D conversion overflow interrupt for ADC1	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

The ADOVFINTCR register controls the enable/disable to the A/D conversion overflow interrupt.

53.2.1.9 ADCALINTCR : Calibration Interrupt Enable Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x028

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CALENDIE1	CALENDIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
16	CALENDIE0	ADC0 Calibration End Interrupt Enable 0: Disable Calibration End Interrupt for ADC0 1: Enable Calibration End Interrupt for ADC0	R/W
17	CALENDIE1	ADC1 Calibration End Interrupt Enable 0: Disable Calibration End Interrupt for ADC1 1: Enable Calibration End Interrupt for ADC1	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

The ADCALINTCR register controls the enable/disable to the Calibration End Interrupt.

53.2.1.10 ADMDR : A/D Converter Mode Selection Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x040

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADMD1[3:0]				—	—	—	—	ADMD0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	ADMD0[3:0]	ADC0 Mode Selection 0x0: SAR mode – Single scan mode 0x1: SAR mode – Continuous scan mode 0x4: Oversampling mode – Single scan mode 0x5: Oversampling mode – Continuous scan mode 0x6: Oversampling mode – One-channel continuous scan mode 0x8: Hybrid mode – Single scan mode 0x9: Hybrid mode – Continuous scan mode 0xA: Hybrid mode – Background continuous scan mode 0xB: Hybrid mode – Fixed-channel continuous scan mode Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
11:8	ADMD1[3:0]	ADC1 Mode Selection 0x0: SAR mode – Single scan mode 0x1: SAR mode – Continuous scan mode 0x4: Oversampling mode – Single scan mode 0x5: Oversampling mode – Continuous scan mode 0x6: Oversampling mode – One-channel continuous scan mode 0x8: Hybrid mode – Single scan mode 0x9: Hybrid mode – Continuous scan mode 0xA: Hybrid mode – Background continuous scan mode 0xB: Hybrid mode – Fixed-channel continuous scan mode Others: Setting prohibited	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

The ADMDR register selects the operation mode and the scan mode for each A/D converter.

53.2.2 Scan Group

53.2.2.1 ADGSPCR : A/D Group Scan Priority Control Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x044

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	GRP1	LGRRS1	RSCN1	PGS1	—	—	—	—	GRP0	LGRRS0	RSCN0	PGS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PGS0	ADC0 Group Priority Control Setting 0: ADC0 operation is without group priority control 1: ADC0 operation is with group priority control in SAR mode. Setting prohibited when other than SAR mode.	R/W
1	RSCN0	ADC0 Group Priority Control Setting 2 0: Set when PGS0 is set to 0. 1: Set when PGS0 is set to 1.	R/W
2	LGRRS0	ADC0 Group Priority Control Setting 3 0: Set when PGS0 is set to 0. 1: Set when PGS0 is set to 1.	R/W
3	GRP0	ADC0 Group Priority Control Setting 4 0: Set the following case: ● When PGS0 is set to 0 ● When PGS0 is set to 1 and ADC0 is SAR mode – Single scan mode. 1: Set when PGS0 is set to 1 and ADC0 is SAR mode – Continuous scan mode.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	PGS1	ADC1 Group Priority Control Setting 0: ADC1 operation is without group priority control 1: ADC1 operation is with group priority control in SAR mode. Setting prohibited when other than SAR mode.	R/W
9	RSCN1	ADC1 Group Priority Control Setting 2 0: Set when PGS1 is set to 0. 1: Set when PGS1 is set to 1.	R/W

Bit	Symbol	Function	R/W
10	LGRRS1	ADC1 Group Priority Control Setting 3 0: Set when PGS1 is set to 0. 1: Set when PGS1 is set to 1.	R/W
11	GRP1	ADC1 Group Priority Control Setting 4 0: Set the following case: ● When PGS1 is set to 0 ● When PGS1 is set to 1 and ADC1 is SAR mode – Single scan mode. 1: Set when PGS1 is set to 1 and ADC1 is SAR mode – Continuous scan mode.	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

The ADGSPCR register controls the group priority operation to each A/D converter. For details on setting this register, see section 53.3.18. Group Priority Operation.

53.2.2.2 ADSGER : Scan Group Enable Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x048

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SGRE8	SGRE7	SGRE6	SGRE5	SGRE4	SGRE3	SGRE2	SGRE1	SGRE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	SGRE0 to SGRE8	Scan Group n Enable The suffix number of each bit symbol corresponds to the scan group number n. 0: Disable the scan group n 1: Enable the scan group n	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

The ADSGER register selects the enable/disable to each scan group.

SGREn bit (Scan Group n Enable) (n = 0 to 8)

The SGREn bit selects the enable/disable to the scan group n. When SGREn is set to 1, A/D conversion with the scan group n is available. When SGREn is set to 0, A/D conversion with the scan group n is not available, even if the trigger corresponding to the scan group n is input.

53.2.2.3 ADSGCR0 : Scan Group Control Register 0

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x04C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	SGADS3[1:0]		—	—	—	—	—	—	SGADS2[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SGADS1[1:0]		—	—	—	—	—	—	SGADS0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	SGADS0[1:0]	Scan Group 0 A/D Converter Selection 0 0: Select ADC0 0 1: Select ADC1 Others: Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	SGADS1[1:0]	Scan Group 1 A/D Converter Selection 0 0: Select ADC0 0 1: Select ADC1 Others: Setting prohibited	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
17:16	SGADS2[1:0]	Scan Group 2 A/D Converter Selection 0 0: Select ADC0 0 1: Select ADC1 Others: Setting prohibited	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
25:24	SGADS3[1:0]	Scan Group 3 A/D Converter Selection 0 0: Select ADC0 0 1: Select ADC1 Others: Setting prohibited	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADSGCR0 register selects which A/D converter is used for each scan group.

53.2.2.4 ADSGCR1 : Scan Group Control Register 1

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x050

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	SGADS7[1:0]		—	—	—	—	—	—	SGADS6[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SGADS5[1:0]		—	—	—	—	—	—	SGADS4[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	SGADS4[1:0]	Scan Group 4 A/D Converter Selection 0 0: Select ADC0 0 1: Select ADC1 Others: Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	SGADS5[1:0]	Scan Group 5 A/D Converter Selection 0 0: Select ADC0 0 1: Select ADC1 Others: Setting prohibited	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
17:16	SGADS6[1:0]	Scan Group 6 A/D Converter Selection 0 0: Select ADC0 0 1: Select ADC1 Others: Setting prohibited	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
25:24	SGADS7[1:0]	Scan Group 7 A/D Converter Selection 0 0: Select ADC0 0 1: Select ADC1 Others: Setting prohibited	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADSGCR1 register selects which A/D converter is used for each scan group.

53.2.2.5 ADSGCR2 : Scan Group Control Register 2

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x054

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SGADS8[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	SGADS8[1:0]	Scan Group 8 A/D Converter Selection 0 0: Select ADC0 0 1: Select ADC1 Others: Setting prohibited	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

The ADSGCR2 register selects which A/D converter is used for each scan group.

53.2.2.6 ADINTCR : Scan End Interrupt Enable Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x05C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ADIE8	ADIE7	ADIE6	ADIE5	ADIE4	ADIE3	ADIE2	ADIE1	ADIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	ADIE0 to ADIE8	Scan Group n Scan End Interrupt Enable The suffix number of each bit symbol corresponds to the scan group number n. 0: Disable scan end interrupt 1: Enable scan end interrupt	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

The ADINTCR register enables/disables the scan end interrupt.

53.2.2.7 ADTRGEXTn : External Trigger Enable Register n (n = 0 to 8)

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x0C0 + 0x10 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TRGEXT1	TRGEXT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TRGEXT0	External Trigger Input 0 (ADTRG0) Enable 0: Disable ADTRG0 1: Enable ADTRG0	R/W
1	TRGEXT1	External Trigger Input 1 (ADTRG1) Enable 0: Disable ADTRG1 1: Enable ADTRG1	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

The ADTRGEXTn register enables/disables the External Trigger input 0 and 1 (ADTRG0 and ADTRG1) as the starting conditions for A/D conversion of scan group n.

53.2.2.8 ADTRGELCn : ELC Trigger Enable Register n (n = 0 to 8)

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x0C4 + 0x10 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	TRGELC5	TRGELC4	TRGELC3	TRGELC2	TRGELC1	TRGELC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	TRGELC0 to TRGELC5	ELC Trigger m Enable The suffix number of each bit symbol corresponds to the ELC trigger number m. 0: Disable ELC Trigger m 1: Enable ELC Trigger m	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

The ADTRGELCn register enables/disables the ELC Trigger m as the starting conditions for A/D conversion of scan group n.

53.2.2.9 ADTRGGPTn : GPT Trigger Enable Register n (n = 0 to 8)

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x0C8 + 0x10 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	TRGGPTB13	TRGGPTB12	TRGGPTB11	TRGGPTB10	TRGGPTB9	TRGGPTB8	TRGGPTB7	TRGGPTB6	TRGGPTB5	TRGGPTB4	TRGGPTB3	TRGGPTB2	TRGGPTB1	TRGGPTB0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	TRGGPTA13	TRGGPTA12	TRGGPTA11	TRGGPTA10	TRGGPTA9	TRGGPTA8	TRGGPTA7	TRGGPTA6	TRGGPTA5	TRGGPTA4	TRGGPTA3	TRGGPTA2	TRGGPTA1	TRGGPTA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
13:0	TRGGPTA0 to TRGGPTA13	GPT channel m A/D Conversion Starting Request A Enable The suffix number of each bit symbol corresponds to the GPT channel number m. 0: Disable the A/D conversion stating request A from GPT channel m 1: Enable the A/D conversion stating request A from GPT channel m	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
29:16	TRGGPTB0 to TRGGPTB13	GPT channel m A/D Conversion Starting Request B Enable The suffix number of each bit symbol corresponds to the GPT channel number m. 0: Disable the A/D conversion stating request B from GPT channel m 1: Enable the A/D conversion stating request B from GPT channel m	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

The ADTRGGPTn register enables/disables the A/D conversion starting request A/B from GPT channel m as the starting conditions for A/D conversion of scan group n.

53.2.2.10 ADTRGENR : A/D Conversion Start Trigger Enable Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xC08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	STTRGEN8	STTRGEN7	STTRGEN6	STTRGEN5	STTRGEN4	STTRGEN3	STTRGEN2	STTRGEN1	STTRGEN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	STTRGEN0 to STTRGEN8	Scan Group n A/D Conversion Start Trigger Enable The suffix number of each bit symbol corresponds to the scan group number n. 0: Disable the A/D conversion start trigger 1: Enable the A/D conversion start trigger	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

The ADTRGENR register enables/disables the trigger from the peripheral modules for starting the A/D conversion of Scan group n. The triggers for each scan group are selected in the ADTRGEXTn, ADTRGELCn, and ADTRGGPTn registers.

STTRGENn bit (Scan Group n A/D Conversion Start Trigger Enable)

The STTRGENn bit selects whether the trigger inputs from the peripheral module, which are selected in the ADTRGEXTn, ADTRGELCn, and ADTRGGPTn registers, is used for the A/D conversion start condition.

When the STTRGENn bit is set to 1, the A/D conversion of scan group n can be started by trigger inputs from the peripheral module.

When the STTRGENn bit is set to 0, the A/D conversion of scan group n cannot be started by trigger inputs from the peripheral module. In this setting, the A/D conversion of scan group n can be only started by writing to the ADSYSTR or ADSTRn registers.

53.2.2.11 ADTRGDLR0 : A/D Conversion Start Trigger Delay Register 0

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x1C0

Bit position:	31								23							16									7							0
Bit field:	—	—	—	—	—	—	—	—	TRGDLY1[7:0]								—	—	—	—	—	—	—	—	TRGDLY0[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TRGDLY0[7:0]	Scan Group 0 Trigger Input Delay Configuration	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
23:16	TRGDLY1[7:0]	Scan Group 1 Trigger Input Delay Configuration	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

The ADTRGDLR0 register controls the input delay added to the External trigger inputs, ELC triggers and GPT triggers which are the starting conditions for the A/D conversion of each scan group. The input delay value is given as the register setting value × the cycle of the A/D conversion clock (ADCLK).

53.2.2.12 ADTRGDLR1 : A/D Conversion Start Trigger Delay Register 1

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x1C4

Bit position:	31								23 16									7 0
Bit field:	—	—	—	—	—	—	—	—	TRGDLY3[7:0]	—	—	—	—	—	—	—	—	TRGDLY2[7:0]
Value after reset:	0	0	0	0	0	0	0	0	0 0 0 0 0 0 0 0	0	0	0	0	0	0	0	0	0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
7:0	TRGDLY2[7:0]	Scan Group 2 Trigger Input Delay Configuration	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
23:16	TRGDLY3[7:0]	Scan Group 3 Trigger Input Delay Configuration	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

The ADTRGDLR1 register controls the input delay added to the External trigger inputs, ELC triggers and GPT triggers which are the starting conditions for the A/D conversion of each scan group. The input delay value is given as the register setting value × the cycle of the A/D conversion clock (ADCLK).

53.2.2.13 ADTRGDLR2 : A/D Conversion Start Trigger Delay Register 2

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x1C8

Bit position:	31								23 16									7 0
Bit field:	—	—	—	—	—	—	—	—	TRGDLY5[7:0]	—	—	—	—	—	—	—	—	TRGDLY4[7:0]
Value after reset:	0	0	0	0	0	0	0	0	0 0 0 0 0 0 0 0	0	0	0	0	0	0	0	0	0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
7:0	TRGDLY4[7:0]	Scan Group 4 Trigger Input Delay Configuration	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
23:16	TRGDLY5[7:0]	Scan Group 5 Trigger Input Delay Configuration	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

The ADTRGDLR2 register controls the input delay added to the External trigger inputs, ELC triggers and GPT triggers which are the starting conditions for the A/D conversion of each scan group. The input delay value is given as the register setting value × the cycle of the A/D conversion clock (ADCLK).

53.2.2.14 ADTRGDLR3 : A/D Conversion Start Trigger Delay Register 3

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x1CC

Bit position:	31								23 16									7 0
Bit field:	—	—	—	—	—	—	—	—	TRGDLY7[7:0]	—	—	—	—	—	—	—	—	TRGDLY6[7:0]
Value after reset:	0	0	0	0	0	0	0	0	0 0 0 0 0 0 0 0	0	0	0	0	0	0	0	0	0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
7:0	TRGDLY6[7:0]	Scan Group 6 Trigger Input Delay Configuration	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
23:16	TRGDLY7[7:0]	Scan Group 7 Trigger Input Delay Configuration	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

The ADTRGDLR3 register controls the input delay added to the External trigger inputs, ELC triggers and GPT triggers which are the starting conditions for the A/D conversion of each scan group. The input delay value is given as the register setting value × the cycle of the A/D conversion clock (ADCLK).

53.2.2.15 ADTRGDLR4 : A/D Conversion Start Trigger Delay Register 4

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x1D0

Bit position:	31																								7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TRGDLY8[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TRGDLY8[7:0]	Scan Group 8 Trigger Input Delay Configuration	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

The ADTRGDLR4 register controls the input delay added to the External trigger inputs, ELC triggers and GPT triggers which are the starting conditions for the A/D conversion of each scan group. The input delay value is given as the register setting value × the cycle of the A/D conversion clock (ADCLK).

53.2.2.16 ADSWNR0 : A/D conversion times per scan Register 0

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x060

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	SWNOFIX[2:0]			—	—	—	—	—	—	SWFIX[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	BCSSTSL[3:0]				—	—	—	—	—	SWNUM[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SWNUM[2:0]	Sweep Conversion Channel Number per scan Specifies the scan period of fixed channel scan mode. 0x0: 1 times 0x1: 2 times ⋮ 0x7: 8 times	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R

Bit	Symbol	Function	R/W
11:8	BCSSTSL[3:0]	Blank Channel Sampling State Table Select Sampling state table selection bit for blank channel. Select the analog input sampling time setting table. The sampling state value of the selected table is used. 0x0: Select sampling state table 0 (ADSSTR0.SST0[9:0]) 0x1: Select sampling state table 1 (ADSSTR0.SST1[9:0]) 0x2: Select sampling state table 2 (ADSSTR1.SST2[9:0]) 0x3: Select sampling state table 3 (ADSSTR1.SST3[9:0]) ⋮ 0xF: Select sampling state table 15 (ADSSTR7.SST15[9:0])	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R
17:16	SWFIX[1:0]	Fixed channel Number Specifies the fixed channel number. 0x0: 1 fixed channel 0x1: 2 fixed channels 0x2: 3 fixed channels 0x3: Setting prohibited	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R
26:24	SWNOFIX[2:0]	Non-Fixed Channel Number Specifies the non-fixed channel number. 0x1: 1 channels Others: Setting prohibited	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R

The ADSWNR0 register sets the A/D conversion period for the fixed channel scan of A/D converter Unit 0.

SWNUM[2:0] bits (Sweep Conversion Channel Number per scan)

The SWNUM[2:0] bits specify the AD conversion cycle for fixed channel scan of A/D converter Unit 0.

The A/D conversion cycle is the value set by SWNUM[2:0] + 1.

BCSSTSL[3:0] bits (Blank Channel Sampling State Table Select)

The BCSSTSL[3:0] bits set the number of sampling states on blank channels that do not get the A/D conversion data.

SWFIX[1:0] bits (Fixed channel Number)

The SWFIX[1:0] bits set the number of fixed channels that are converted channels sweeping through each scan.

SWNOFIX[2:0] bits (Non-Fixed Channel Number)

The SWNOFIX[2:0] bits set the number of non-fixed channels that are converted while switching channels in each scan.

Note: This register is valid only when fixed channel continuous scan is set (ADMD0[3:0] = 0xB).

53.2.2.17 ADSWNR1 : A/D conversion times per scan Register 1

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x064

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	SWNOFIX[2:0]			—	—	—	—	—	—	SWFIX[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	BCSSTSL[3:0]				—	—	—	—	—	SWNUM[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SWNUM[2:0]	Sweep Conversion Channel Number per scan Specifies the scan period of fixed channel scan mode. 0x0: 1 times 0x1: 2 times : 0x7: 8 times	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R
11:8	BCSSTSL[3:0]	Blank Channel Sampling State Table Select Sampling state table selection bit for blank channel. Select the analog input sampling time setting table. The sampling state value of the selected table is used. 0x0: Select sampling state table 0 (ADSSTR0.SST0[9:0]) 0x1: Select sampling state table 1 (ADSSTR0.SST1[9:0]) 0x2: Select sampling state table 2 (ADSSTR1.SST2[9:0]) 0x3: Select sampling state table 3 (ADSSTR1.SST3[9:0]) : 0xF: Select sampling state table 15 (ADSSTR7.SST15[9:0])	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R
17:16	SWFIX[1:0]	Fixed channel Number Specifies the fixed channel number. 0x0: 1 fixed channel 0x1: 2 fixed channels 0x2: 3 fixed channels 0x3: Setting prohibited	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R
26:24	SWNOFIX[2:0]	Non-Fixed Channel Number Specifies the non-fixed channel number. 0x1: 1 channels Others: Setting prohibited	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R

The ADSWNR1 register sets the A/D conversion period for the fixed channel scan of A/D converter Unit 1.

SWNUM[2:0] bits (Sweep Conversion Channel Number per scan)

The SWNUM[2:0] bits specify the AD conversion cycle for fixed channel scan of A/D converter Unit 1.

The A/D conversion cycle is the value set by SWNUM[2:0] + 1.

BCSSTSL[3:0] bits (Blank Channel Sampling State Table Select)

The BCSSTSL[3:0] bits set the number of sampling states on blank channels that do not get the A/D conversion data.

SWFIX[1:0] bits (Fixed channel Number)

The SWFIX[1:0] bits set the number of fixed channels that are converted channels sweeping through each scan.

SWNOFIX[2:0] bits (Non-Fixed Channel Number)

The SWNOFIX[2:0] bits set the number of non-fixed channels that are converted while switching channels in each scan.

Note: This register is valid only when fixed channel continuous scan is set (ADMD1[3:0] = 0xB).

53.2.2.18 ADDECCR : One-channel Continuous Scan Mode Decimation Control Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x080

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DCIM1[4:0]					—	—	—	DCIM0[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	DCIM0[4:0]	One-channel continuous scan mode output decimation setting for A/D converter unit 0 0x0: No decimation 0x3: Output data once for every 4 conversions (thinned 3 times) 0x7: Output data once for every 8 conversions (thinned out 7 times) Others: Setting prohibited	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
12:8	DCIM1[4:0]	One-channel continuous scan mode output decimation setting for A/D converter unit 1 0x0: No decimation 0x3: Output data once for every 4 conversions (thinned 3 times) 0x7: Output data once for every 8 conversions (thinned out 7 times) Others: Setting prohibited	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

ADDECCR register controls the decimation of data output in the one-channel continuous scan mode.

53.2.2.19 ADACMDR : A/D Converter Accuracy Mode Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x084

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ADHACMD1	—	—	—	—	—	—	—	ADHACMD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ADHACMD0	A/D converter unit 0 high accuracy mode setting 0: High accuracy mode of A/D converter is disabled 1: High accuracy mode of A/D converter is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	ADHACMD1	A/D converter unit 1 high accuracy mode setting 0: High accuracy mode of A/D converter is disabled 1: High accuracy mode of A/D converter is enabled	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

The ADACMDR register controls the accuracy mode of the A/D converter.

ADHACMDx (x = 0, 1) bit (A/D Converter Accuracy Mode Setting Register)

The ADHACMDx (x = 0, 1) bit selects the accuracy mode of the A/D converter.

Setting this bit to 1 enables high accuracy mode of the A/D converter. Enabling the high accuracy mode provides higher accuracy A/D conversion results than when the high accuracy mode is disabled. For details on analog characteristics, see section 62, Electrical Characteristics.

When setting this bit to 1 (enable high accuracy mode), some register setting restrictions apply to this function. For details, see .

53.2.3 Virtual Channel

53.2.3.1 ADCHCRn : A/D Conversion Channel Configuration Register n (n = 0 to 32)

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x600 + 0x10 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	SSTSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AINMD	CNVCS[6:0]							—	—	—	SGSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	SGSEL[4:0]	Scan Group Selection 0x00: Not select any scan group 0x01: Select the scan group 0 0x02: Select the scan group 1 0x03: Select the scan group 2 0x04: Select the scan group 3 ⋮ 0x09: Select the scan group 8 Others: Setting prohibited	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
14:8	CNVCS[6:0]	A/D Conversion Channel Selection 0x00: AN000 0x01: AN001 ⋮ 0x16: AN022 0x60: Self-diagnosis A/D unit 0*1 0x61: Self-diagnosis A/D unit 1*1 0x64: Temperature sensor 0x65: Internal reference voltage 0x66: VBATT 1/6 voltage monitor output 0x68: D/A converter channel 0 0x69: D/A converter channel 1 0x70: Self-diagnosis Sample-and-hold Circuit Unit 0*1 0x71: Self-diagnosis Sample-and-hold Circuit Unit 1*1 0x72: Self-diagnosis Sample-and-hold Circuit Unit 2*1 0x74: Self-diagnosis Sample-and-hold Circuit Unit 4*1 0x75: Self-diagnosis Sample-and-hold Circuit Unit 5*1 0x76: Self-diagnosis Sample-and-hold Circuit Unit 6*1 Others: Setting prohibited	R/W

Bit	Symbol	Function	R/W
15	AINMD	Analog Input mode selection[*1,*2] 0: Single-ended input 1: Differential input	R/W
19:16	SSTSEL[3:0]	Sampling State Table Selection 0x0: Select the sampling state table 0 0x1: Select the sampling state table 1 0x2: Select the sampling state table 2 0x3: Select the sampling state table 3 ⋮ 0xF: Select the sampling state table 15	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Select Differential input when self-diagnosis is selected.
Note 2. When using channel-dedicated sample-and-hold circuit, ADSHCRm.SHMDn (m = 0, 1, n = 0 to 2, 4 to 6) and corresponding ADCHCRn.AINMD (n = 0 to 32) setting must be matched.

The ADCHCRn register selects the Virtual channel configuration for A/D conversion.

SGSEL[4:0] bits (Scan Group Selection)

The SGSEL[4:0] bits select which scan group to assign the selected channel in CNVCS[6:0]. If SGSEL[4:0] is set to 0x00, the channel selected by CNVCS[6:0] is not available for A/D conversion. If SGSEL[4:0] is set to other than 0x00, the channel selected by CNVCS[6:0] is available for A/D conversion in the scan operation of the scan group corresponding to the setting value.

Up to 8 A/D conversion channels can be assigned per scan group in the SAR mode or Oversampling mode; Up to 4 A/D conversion channels can be assigned per scan group in the Hybrid mode. Do not assign more than specified number of A/D conversion channels per scan group.

CNVCS[6:0] bits (A/D Conversion Channel Selection)

The CNVCS[6:0] bits select the analog signal source for the A/D conversion. The CNVCS[6:0] bits should be set for the channel number corresponding to the Analog Input or the Extended Analog function. For A/D conversion of the analog signal source in Differential input mode, the CNVCS[6:0] bits should be set the even-numbered channel; When the odd-numbered channel is set in Differential input mode, A/D conversion is not guaranteed.

To A/D convert the channel selected by the CNVCS[6:0] bits, it should be assigned to one of the scan groups by the SGSEL[4:0] bits.

AINMD bits (Analog Input mode selection)

The AINMD bit selects the input method for A/D conversion of the analog signal source selected by the CNVCS[6:0] bits. When the AINMD bit is set to 0, A/D conversion is performed in single-ended input. When the AINMD bit is set to 1, A/D conversion is performed in differential input.

If the self-diagnosis channel is selected with the CNVCS[6:0] bits, a differential input should be selected.

SSTSEL[3:0] bits (Sampling State Table Selection)

The SSTSEL[3:0] bits select the sampling state table for A/D conversion of the analog signal source selected by the CNVCS[6:0] bits. The number of sampling states is the value set in the selected sampling state table.

53.2.3.2 ADDOPCRAn : A/D Conversion Data Operation Control A Register n (n = 0 to 32)

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x604 + 0x10 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	OFSETSEL[3:0]				—	—	—	—	GAINSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	DFSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	DFSEL[2:0]	Digital Filter Selection[*1 *2] 0x0: Not use the digital filter 0x1: Use the 1st digital filter 0x2: Use the 2nd digital filter 0x3: Use the 3rd digital filter 0x4: Use the 4th digital filter Others: Setting prohibited	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W
19:16	GAINSEL[3:0]	User Gain Table Selection 0x0: Not use the User Gain Table 0x1: Use the User Gain Table 0 0x2: Use the User Gain Table 1 0x3: Use the User Gain Table 2 ⋮ 0x8: Use the User Gain Table 7 Others: Setting prohibited	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
27:24	OFSETSEL[3:0]	User Offset Table Selection 0x0: Not use the user offset table 0x1: Use the user offset table 0 0x2: Use the user offset table 1 0x3: Use the user offset table 2 ⋮ 0x8: Use the user offset table 7 Others: Setting prohibited	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Whenever A/D conversion is performed in Oversampling mode or Hybrid mode, the digital filter must be used.
Note 2. Whenever A/D conversion is performed in SAR mode, the digital filter must not be used.

The ADDOPCRAn register is one of the registers that selects the data calculation function for the A/D conversion data of virtual channel n.

DFSEL[2:0] bits (Digital Filter Selection)

The DFSEL[2:0] bits select the digital filter for inputting A/D converted data of virtual channel n. When 000b is set in the DFSEL[2:0] bits, the digital filter is not used. When 001b, 010b, 011b, or 100b is set in the DFSEL[2:0] bits, the digital filter corresponding to the setting of the DFSEL[2:0] bits is selected. The characteristics of the digital filters are specified by ADDFSRm (m = 0, 1) register.

For SAR mode, select 000b for the DFSEL[2:0] bits. For Oversampling mode or Hybrid mode, set the DFSEL[2:0] bits to 001b, 010b, 011b, or 100b.

GAINSEL[3:0] bits (User Gain Table Selection)

The GAINSEL[3:0] bits select the User Gain Table for adjusting the A/D conversion data of virtual channel n.

When the GAINSEL[3:0] bits are set to 0x0, no gain adjustment is performed. When the GAINSEL[3:0] bits are set to any of 0x1 to 0x8, the gain adjustment is performed according to the User Gain Table selected by the GAINSEL[3:0] bits.

OFSETSEL[3:0] bits (User Offset Table Selection)

The OFSETSEL[3:0] bits select the user offset table for adjusting the A/D conversion data of virtual channel n. When the OFSETSEL[3:0] bits are set to 0x0, no offset adjustment is performed. When the OFSETSEL[3:0] bits are set to any of 0x1 to 0x8, the offset adjustment is performed according to the User Offset Table selected by the OFSETSEL[3:0] bits.

53.2.3.3 ADDOPCRBn : A/D Conversion Data Operation Control B Register n (n = 0 to 32)

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x608 + 0x10 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	CMPTBLE7	CMPTBLE6	CMPTBLE5	CMPTBLE4	CMPTBLE3	CMPTBLE2	CMPTBLE1	CMPTBLE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	ADC[3:0]				—	—	—	—	—	—	AVEMD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	AVEMD[1:0]	Addition/Averaging Mode Selection 0 0: Not use Addition/Averaging mode 0 1: Addition mode 1 0: Averaging mode 1 1: Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
11:8	ADC[3:0]	Addition/Averaging Times Selection 0x0: 1-time conversion (no addition, same as normal conversion) 0x1: 2-time conversion (1 addition) 0x3: 4-time conversion (3 additions) 0x4: 8-time conversion (7 additions) 0x5: 16-time conversion (15 additions) 0x6: 32-time conversion (31 additions) 0x7: 64-time conversion (63 additions) 0x8: 128-time conversion (127 additions) 0x9: 256-time conversion (255 additions) 0xA: 512-time conversion (511 additions) 0xB: 1024-time conversion (1023 additions) Others: Setting prohibited	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
23:16	CMPTBLE0 to CMPTBLE7	Compare Match Enable The suffix number of each bit symbol corresponds to the compare match table number m. 0: Disable the compare match with the compare match table m 1: Enable the compare match with the compare match table m	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

The ADDOPCRBn register is one of the registers that selects the data calculation function for the A/D conversion data of virtual channel n.

AVEMD[1:0] bits (Addition/Averaging Mode Selection)

The AVEMD[1:0] bits select the addition mode or the averaging mode for the A/D conversion of virtual channel n. When the AVEMD[1:0] bits are set to 00b, the addition/averaging operation is not performed. When the AVEMD[1:0] bits are set to 01b or 10b, the addition/averaging operation is performed according to the setting value of the ADC[3:0] bits.

ADC[3:0] bits (Addition/Averaging Times Selection)

The ADC[3:0] bits select the number of times of addition/averaging for the A/D conversion of virtual channel n. If the addition mode or the averaging mode is selected in the AVEMD[1:0] bits, the A/D conversion is performed a number of times according to the setting value of the ADC[3:0] bits.

CMPTBLEm bits (Compare Match Enable) (m = 0 to 7)

The CMPTBLEm bits enable/disable the compare match with the compare match table m for the A/D conversion data of virtual channel n.

53.2.3.4 ADDOPCRCn : A/D Conversion Data Operation Control C Register n (n = 0 to 32)

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x60C + 0x10 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	SIGNSEL	—	—	ADPRC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	LIMTBLS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	LIMTBLS[3:0]	Limiter Clip Table Selection 0x0: Not use the Limiter Clip Table 0x1: Use the Limiter Clip Table 0 0x2: Use the Limiter Clip Table 1 0x3: Use the Limiter Clip Table 2 ⋮ 0x8: Use the Limiter Clip Table 7 Others: Setting prohibited	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W
17:16	ADPRC[1:0]	A/D Conversion Data Format Selection 0 0: Store the A/D conversion result as 16-bit data format 0 1: Store the A/D conversion result as 14-bit data format 1 0: Store the A/D conversion result as 12-bit data format 1 1: Store the A/D conversion result as 10-bit data format	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
20	SIGNSEL	A/D Conversion Data Signed/Un-signed Selection 0: Signed data format 1: Un-signed data format	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

The ADDOPCRCn register is one of the registers that selects the data calculation function for the A/D conversion data of virtual channel n.

LIMTBLS[3:0] bits (Limiter Clip Table Selection)

The LIMTBLS[3:0] bits select the limiter clip table for clipping the A/D conversion data of virtual channel n. When the LIMTBLS[3:0] bits are set to 0x0, the A/D conversion data is not clipped. When the LIMTBLS[3:0] bits are set to any of 0x1 to 0x8, the A/D conversion data is clipped according to the limiter clip table selected by the LIMTBLS[3:0] bits.

ADPRC[1:0] bits (A/D Conversion Data Format Selection)

The ADPRC[1:0] bits select the data length of A/D conversion data. The A/D conversion data of virtual channel n is stored to the A/D Data Register or the FIFO with selected data length.

SIGNSEL bit (A/D Conversion Data Signed/Un-signed Selection)

The SIGNSEL bit selects the Signed/Unsigned data format of the A/D conversion data. The A/D conversion data of the virtual channel n is stored to the A/D Data Register or the FIFO with selected data format.

Select the Signed data format (SIGNSEL = 0) whenever the A/D conversion of the self-diagnosis channel is performed. Select the Unsigned data format (SIGNSEL = 1) whenever the A/D conversion in SAR mode, unless to convert the self-diagnosis channel, is performed. The signed data format or the unsigned data format is available for the A/D conversion in Oversampling mode or Hybrid mode unless converting to the self-diagnosis channel.

53.2.4 A/D Conversion Configuration

53.2.4.1 ADSGDCRn : Scan Group Diagnosis Function Control Register n (n = 0 to 8)

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x200 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ADNDIS[7:0]								—	—	ADDISN	ADDISP	—	—	—	ADDISEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	DIAGVAL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	DIAGVAL[2:0]	Self-diagnosis Mode Selection 0 0 0: Set when any self-diagnosis channel is not included. Setting prohibited when any self-diagnosis channels are included. 1 0 0: Self-diagnosis mode 1 1 0 1: Self-diagnosis mode 2 1 1 0: Self-diagnosis mode 3 Others: Setting prohibited	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W
16	ADDISEN	Disconnection Detection Assist Enable 0: Disable the disconnection detection assist function 1: Enable the disconnection detection assist function	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	ADDISP	Disconnection Detection Assist Mode Selection (for the even-numbered analog channel) 0: Discharge 1: Precharge	R/W
21	ADDISN	Disconnection Detection Assist Mode Selection (for the odd-numbered analog channel) 0: Discharge 1: Precharge	R/W

Bit	Symbol	Function	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
31:24	ADNDIS[7:0]	Disconnection Detection Assist Period 0x0: Setting prohibited when the disconnection detection assist function is enabled 0x1: Setting prohibited 0x2: Setting prohibited 0x3: Setting prohibited Others: The number of states for the discharge or precharge period. The number of states actually valid in the operation is the value of the register setting minus one.	R/W

The ADSGDCRn register controls the diagnosis function at the scanning operation of scan group n.

DIAGVAL[2:0] bits (Self-diagnosis Mode Selection)

The DIAGVAL[2:0] bits select the self-diagnosis mode for A/D converter.

The DIAGVAL[2:0] bits must be set to 000b when the scan group n does not include any virtual channels that selected the self-diagnosis channel. The DIAGVAL[2:0] bits must be set to 100b, 101b or 110b when the scan group n is including the virtual channel that selected the self-diagnosis channel.

ADDISEN bit (Disconnection Detection Assist Enable)

The ADDISEN bit enables/disables the disconnection detection assist function. When the ADDISEN bit is set to 0, the disconnection detection assist operation is not performed. When the ADDISEN bit is set to 1, the disconnection detection assist operation according to the setting of the ADDISP bit and the ADDISN bit is performed at the scanning operation of scan group n.

ADDISP bit / ADDISN bit (Disconnection Detection Assist Mode Selection)

The ADDISP bit and the ADDISN bit select the discharge/precharge operation for the disconnection detection.

The ADDISP bit selects the discharge/precharge operation for the even-numbered analog channel.

The ADDISN bit selects the discharge/precharge operation for the odd-numbered analog channel.

ADNDIS[7:0] bits (Disconnection Detection Assist Period)

The ADNDIS[7:0] bits select the discharge/precharge period. When the disconnection detection assist function is disabled (ADDISEN = 0), the ADNDIS[7:0] bits should be set to 0x0. When the disconnection detection assist function is enabled (ADDISEN = 1), the ADNDIS[7:0] bits should be set the value within 0x3 to 0xFF. The number of states actually valid in the operation is the value of the register setting minus one. For example, if ADNDIS[7:0] = 0x05 is set, the discharge/precharge period is 4 states.

53.2.4.2 ADSSTR0 : Sampling State Table Register 0

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x240

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SST1[9:0]										—	—	—	—	—	—	SST0[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
9:0	SST0[9:0]	Sampling State Table 0 These bits set the sampling time in the range from 2 to 1023 states.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	SST1[9:0]	Sampling State Table 1 These bits set the sampling time in the range from 2 to 1023 states.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADSSTR0 register selects the sampling time at the A/D conversion that is using each sampling state table.

53.2.4.3 ADSSTR1 : Sampling State Table Register 1

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x244

Bit position:	31						25									16							9									0
Bit field:	—	—	—	—	—	—	SST3[9:0]										—	—	—	—	—	—	SST2[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
9:0	SST2[9:0]	Sampling State Table 2 These bits set the sampling time in the range from 2 to 1023 states.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	SST3[9:0]	Sampling State Table 3 These bits set the sampling time in the range from 2 to 1023 states.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADSSTR1 register selects the sampling time at the A/D conversion that is using each sampling state table.

53.2.4.4 ADSSTR2 : Sampling State Table Register 2

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x248

Bit position:	31						25									16							9									0
Bit field:	—	—	—	—	—	—	SST5[9:0]										—	—	—	—	—	—	SST4[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
9:0	SST4[9:0]	Sampling State Table 4 These bits set the sampling time in the range from 2 to 1023 states.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	SST5[9:0]	Sampling State Table 5 These bits set the sampling time in the range from 2 to 1023 states.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADSSTR2 register selects the sampling time at the A/D conversion that is using each sampling state table.

53.2.4.5 ADSSTR3 : Sampling State Table Register 3

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x24C

Bit position:	31						25									16							9									0
Bit field:	—	—	—	—	—	—	SST7[9:0]										—	—	—	—	—	—	SST6[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
9:0	SST6[9:0]	Sampling State Table 6 These bits set the sampling time in the range from 2 to 1023 states.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	SST7[9:0]	Sampling State Table 7 These bits set the sampling time in the range from 2 to 1023 states.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADSSTR3 register selects the sampling time at the A/D conversion that is using each sampling state table.

53.2.4.6 ADSSTR4 : Sampling State Table Register 4

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x250

Bit position:	31	30	29	28	27	26	25–16	15	14	13	12	11	10	9–0
Bit field:	—	—	—	—	—	—	SST9[9:0]	—	—	—	—	—	—	SST8[9:0]
Value after reset:	0	0	0	0	0	0	0 0 0 0 0 0 0 0 1 0	0	0	0	0	0	0	0 0 0 0 0 0 0 0 1 0

Bit	Symbol	Function	R/W
9:0	SST8[9:0]	Sampling State Table 8 These bits set the sampling time in the range from 2 to 1023 states.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	SST9[9:0]	Sampling State Table 9 These bits set the sampling time in the range from 2 to 1023 states.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADSSTR4 register selects the sampling time at the A/D conversion that is using each sampling state table.

53.2.4.7 ADSSTR5 : Sampling State Table Register 5

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x254

Bit position:	31	30	29	28	27	26	25–16	15	14	13	12	11	10	9–0
Bit field:	—	—	—	—	—	—	SST11[9:0]	—	—	—	—	—	—	SST10[9:0]
Value after reset:	0	0	0	0	0	0	0 0 0 0 0 0 0 0 1 0	0	0	0	0	0	0	0 0 0 0 0 0 0 0 1 0

Bit	Symbol	Function	R/W
9:0	SST10[9:0]	Sampling State Table 10 These bits set the sampling time in the range from 2 to 1023 states.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	SST11[9:0]	Sampling State Table 11 These bits set the sampling time in the range from 2 to 1023 states.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADSSTR5 register selects the sampling time at the A/D conversion that is using each sampling state table.

53.2.4.8 ADSSTR6 : Sampling State Table Register 6

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x258

Bit position:	31						25									16							9									0
Bit field:	—	—	—	—	—	—	SST13[9:0]										—	—	—	—	—	—	SST12[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
9:0	SST12[9:0]	Sampling State Table 12 These bits set the sampling time in the range from 2 to 1023 states.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	SST13[9:0]	Sampling State Table 13 These bits set the sampling time in the range from 2 to 1023 states.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADSSTR6 register selects the sampling time at the A/D conversion that is using each sampling state table.

53.2.4.9 ADSSTR7 : Sampling State Table Register 7

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x25C

Bit position:	31						25									16							9									0
Bit field:	—	—	—	—	—	—	SST15[9:0]										—	—	—	—	—	—	SST14[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
9:0	SST14[9:0]	Sampling State Table 14 These bits set the sampling time in the range from 2 to 1023 states.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
25:16	SST15[9:0]	Sampling State Table 15 These bits set the sampling time in the range from 2 to 1023 states.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADSSTR7 register selects the sampling time at the A/D conversion that is using each sampling state table.

53.2.4.10 ADCNVSTR : A/D Conversion State Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x260

Bit position:	31																		13					8			5					0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CST1[5:0]						—	—	CST0[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
5:0	CST0[5:0]	A/D Converter Unit 0 (ADC0) A/D successive approximation time configuration These bits set the successive approximation time in the range from 3 to 63 states.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
13:8	CST1[5:0]	A/D Converter Unit 1 (ADC1) A/D successive approximation time configuration These bits set the successive approximation time in the range from 3 to 63 states.	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

The ADCNVSTR register specifies the A/D successive approximation time, which required by the A/D converter Unit 0 or Unit 1 to convert the sampled analog input to the digital value, in the number of cycles based on A/D conversion clock (ADCLK). The A/D successive approximation time does not include the sampling time of the analog input.

The A/D successive approximation time must be set to meet the value specified in section 62, Electrical Characteristics. If the successive approximation time is set to a value over the range specified for the electrical characteristics, the A/D conversion result is not guaranteed. In addition, the A/D successive approximation time for each A/D converter should be set to be the same. (Set the same value to CST0[5:0] bits and CST1[5:0] bits).

53.2.5 S&H and Others

53.2.5.1 ADSHCR0 : Channel-dedicated Sample-and-hold Circuit Control Register 0

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x280

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SHMD2	SHMD1	SHMD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SHEN2	SHEN1	SHEN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SHEN0	Channel-dedicated Sample-and-hold Circuit Unit 0 Select 0: Bypass the circuit unit 0 1: Use the circuit unit 0	R/W
1	SHEN1	Channel-dedicated Sample-and-hold Circuit Unit 1 Select 0: Bypass the circuit unit 1 1: Use the circuit unit 1	R/W
2	SHEN2	Channel-dedicated Sample-and-hold Circuit Unit 2 Select 0: Bypass the circuit unit 2 1: Use the circuit unit 2	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W
16	SHMD0[*1]	Channel-dedicated Sample-and-hold Circuit Unit 0 Input Mode Select 0: Single-ended Input mode 1: Differential Input mode	R/W
17	SHMD1[*1]	Channel-dedicated Sample-and-hold Circuit Unit 1 Input Mode Select 0: Single-ended Input mode 1: Differential Input mode	R/W
18	SHMD2[*1]	Channel-dedicated Sample-and-hold Circuit Unit 2 Input Mode Select 0: Single-ended Input mode 1: Differential Input mode	R/W

Bit	Symbol	Function	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. When using channel-dedicated sample-and-hold circuit, ADSHCRm.SHMDn (m = 0, 1, n = 0 to 2, 4 to 6) and corresponding ADCHCRn.AINMD (n = 0 to 32) setting must be matched.

The ADSHCR0 register controls the channel-dedicated sample-and-hold circuits (unit 0, unit 1 and unit 2) connected to the A/D converter unit 0 (ADC0).

SHENn bit (Channel-dedicated Sample-and-hold Circuit Bypass Select) (n = 0 to 2)

The SHENn bit selects whether to use or bypass the channel-dedicated sample-and-hold circuits unit n.

When the SHENn bit is set to 0, the channel-dedicated sample-and-hold circuits unit n is disabled. If A/D conversion of analog input channel which is connected to the channel-dedicated sample-and-hold circuit unit n is performed, the channel-dedicated sample-and-hold circuit is not used and is bypassed.

When the SHENn bit is set to 1, the channel-dedicated sample-and-hold circuits unit n is enabled. In this case, A/D conversion of analog input channel to which the channel-dedicated sample-and-hold circuit unit n is performed with the channel-dedicated sample-and-hold circuit used.

SHMDn bit (Channel-dedicated Sample-and-hold Circuit Input Mode Select) (n = 0 to 2)

The SHMDn bit selects the input mode of the channel-dedicated sample-and-hold circuits unit n.

When the SHMDn bit is set to 0, the channel-dedicated sample-and-hold circuits unit n is in Single-ended input mode.

When the SHMDn bit is set to 1, the channel-dedicated sample-and-hold circuits unit n is in Differential input mode.

53.2.5.2 ADSHCR1 : Channel-dedicated Sample-and-hold Circuit Control Register 1

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x28C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SHMD6	SHMD5	SHMD4
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SHEN6	SHEN5	SHEN4
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SHEN4	Channel-dedicated Sample-and-hold Circuit Unit 4 Select 0: Bypass the circuit unit 4 1: Use the circuit unit 4	R/W
1	SHEN5	Channel-dedicated Sample-and-hold Circuit Unit 5 Select 0: Bypass the circuit unit 5 1: Use the circuit unit 5	R/W
2	SHEN6	Channel-dedicated Sample-and-hold Circuit Unit 6 Select 0: Bypass the circuit unit 6 1: Use the circuit unit 6	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W
16	SHMD4*1	Channel-dedicated Sample-and-hold Circuit Unit 4 Input Mode Select 0: Single-ended Input mode 1: Differential Input mode	R/W
17	SHMD5*1	Channel-dedicated Sample-and-hold Circuit Unit 5 Input Mode Select 0: Single-ended Input mode 1: Differential Input mode	R/W

Bit	Symbol	Function	R/W
18	SHMD6[*1]	Channel-dedicated Sample-and-hold Circuit Unit 6 Input Mode Select 0: Single-ended Input mode 1: Differential Input mode	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. When using channel-dedicated sample-and-hold circuit, ADSHCRm.SHMDn (m = 0, 1, n = 0 to 2, 4 to 6) and corresponding ADCHCRn.AINMD (n = 0 to 32) setting must be matched.

The ADSHCR1 register controls the channel-dedicated sample-and-hold circuits (unit 4, unit 5 and unit 6) connected to the A/D converter unit 1 (ADC1).

SHENn bit (Channel-dedicated Sample-and-hold Circuit Bypass Select) (n = 4 to 6)

The SHENn bit selects whether to use or bypass the channel-dedicated sample-and-hold circuits unit n.

When the SHENn bit is set to 0, the channel-dedicated sample-and-hold circuits unit n is disabled. If A/D conversion of analog input channel which is connected to the channel-dedicated sample-and-hold circuit unit n is performed, the channel-dedicated sample-and-hold circuit is not used and is bypassed.

When the SHENn bit is set to 1, the channel-dedicated sample-and-hold circuits unit n is enabled. In this case, A/D conversion of analog input channel to which the channel-dedicated sample-and-hold circuit unit n is performed with the channel-dedicated sample-and-hold circuit used.

SHMDn bit (Channel-dedicated Sample-and-hold Circuit Input Mode Select) (n = 4 to 6)

The SHMDn bit selects the input mode of the channel-dedicated sample-and-hold circuits unit n.

When the SHMDn bit is set to 0, the channel-dedicated sample-and-hold circuits unit n is in Single-ended input mode.

When the SHMDn bit is set to 1, the channel-dedicated sample-and-hold circuits unit n is in Differential input mode.

53.2.5.3 ADSHDCR0 : Channel-dedicated Sample-and-hold Circuit Diagnosis Control Register 0

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x284

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	SH0DIS	—	—	—	SH0DISEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SH0DISEN	A/D converter unit 0 channel-dedicated sample-and-hold circuit disconnection detection assist enable 0: Disconnection detection assist is disabled 1: Disconnection detection assist is enabled	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	SH0DIS	A/D converter unit 0 channel-dedicated sample-and-hold circuit disconnection detection assist selection 0: Discharge 1: Precharge	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: For more details about restrictions, see section 53.3.16.6. Restrictions on Channel-dedicated Sample-and-hold Circuit.

The ADSHDCR0 register controls settings of the diagnosis function for the channel-dedicated sample-and-hold circuits (unit 0, unit 1, unit 2) connected to the A/D converter unit 0 (ADC0).

53.2.5.4 ADSHDCR1 : Channel-dedicated Sample-and-hold Circuit Diagnosis Control Register 1

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x290

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	SH1DIS	—	—	—	SH1DISEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SH1DISEN	A/D converter unit 1 channel-dedicated sample-and-hold circuit disconnection detection assist enable 0: Disconnection detection assist is disabled 1: Disconnection detection assist is enabled	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	SH1DIS	A/D converter unit 1 channel-dedicated sample-and-hold circuit disconnection detection assist selection 0: Discharge 1: Precharge	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: For more details about restrictions, see section 53.3.16.6. Restrictions on Channel-dedicated Sample-and-hold Circuit.

The ADSHDCR1 register controls settings of the diagnosis function for the channel-dedicated sample-and-hold circuits (unit 4, unit 5, unit 6) connected to the A/D converter unit 1 (ADC1).

53.2.5.5 ADSHSTR0 : Channel-dedicated Sample-and-hold Circuit State Register 0

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x288

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SHHST[2:0]			—	—	—	—	—	—	—	—	SHSST[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
7:0	SHSST[7:0]	Channel-dedicated Sample-and-hold Circuit Unit 0 to 2 Sampling Time Setting These bits set the sampling time in the range from 4 to 255 states.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
18:16	SHHST[2:0]	Channel-dedicated Sample-and-hold Circuit Unit 0 to 2 Hold Mode Switching Time Setting These bits set the hold mode switching time in the range from 2 to 7 states.	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

The ADSHSTR0 register specifies the sampling time and the hold mode switching time for the channel-dedicated sample-and-hold circuits (unit 0, unit 1, and unit 2) connected to the A/D converter unit 0 (ADC0). The value set in this register should be set to meet the value specified in section 62, Electrical Characteristics.

53.2.5.6 ADSHSTR1 : Channel-dedicated Sample-and-hold Circuit State Register 1

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x294

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SHHST[2:0]			—	—	—	—	—	—	—	—	SHSST[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
7:0	SHSST[7:0]	Channel-dedicated Sample-and-hold Circuit Unit 4 to 6 Sampling Time Setting These bits set the sampling time in the range from 4 to 255 states.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
18:16	SHHST[2:0]	Channel-dedicated Sample-and-hold Circuit Unit 4 to 6 Hold Mode Switching Time Setting These bits set the hold mode switching time in the range from 2 to 7 states.	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

The ADSHSTR1 register specifies the sampling time and the hold mode switching time for the channel-dedicated sample-and-hold circuits (unit 4, unit 5, and unit 6) connected to the A/D converter unit 1 (ADC1). The value set in this register should be set to meet the value specified in section 62, Electrical Characteristics.

53.2.5.7 ADSHSBPCR : Channel-dedicated Sample-and-hold Circuit Single-ended Input Mode Bypass Control Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x310

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	SHSBN6	SHSBN5	SHSBN4	—	SHSBN2	SHSBN1	SHSBN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	SHSBP6	SHSBP5	SHSBP4	—	SHSBP2	SHSBP1	SHSBP0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SHSBP0	Channel-dedicated sample-and-hold circuit unit 0 single-ended input bypass selection (even-numbered analog channel) Can be written only when the target channel-dedicated sample-and-hold circuit setting is ADSHCR0.SHEN0 = 0. 0: Use the sample-and-hold circuit 1: Bypass the sample-and-hold circuit	R/W

Bit	Symbol	Function	R/W
1	SHSBP1	Channel-dedicated sample-and-hold circuit unit 1 single-ended input bypass selection (even-numbered analog channel) Can be written only when the target channel-dedicated sample-and-hold circuit setting is ADSHCR0.SHEN1 = 0. 0: Use the sample-and-hold circuit 1: Bypass the sample-and-hold circuit	R/W
2	SHSBP2	Channel-dedicated sample-and-hold circuit unit 2 single-ended input bypass selection (even-numbered analog channel) Can be written only when the target channel-dedicated sample-and-hold circuit setting is ADSHCR0.SHEN2 = 0. 0: Use the sample-and-hold circuit 1: Bypass the sample-and-hold circuit	R/W
3	—	This bit is read as 0. The write value should be 0.	R
4	SHSBP4	Channel-dedicated sample-and-hold circuit unit 4 single-ended input bypass selection (even-numbered analog channel) Can be written only when the target channel-dedicated sample-and-hold circuit setting is ADSHCR1.SHEN4 = 0. 0: Use the sample-and-hold circuit 1: Bypass the sample-and-hold circuit	R/W
5	SHSBP5	Channel-dedicated sample-and-hold circuit unit 5 single-ended input bypass selection (even-numbered analog channel) Can be written only when the target channel-dedicated sample-and-hold circuit setting is ADSHCR1.SHEN5 = 0. 0: Use the sample-and-hold circuit 1: Bypass the sample-and-hold circuit	R/W
6	SHSBP6	Channel-dedicated sample-and-hold circuit unit 6 single-ended input bypass selection (even-numbered analog channel) Can be written only when the target channel-dedicated sample-and-hold circuit setting is ADSHCR1.SHEN6 = 0. 0: Use the sample-and-hold circuit 1: Bypass the sample-and-hold circuit	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R
16	SHSBN0	Channel-dedicated sample-and-hold circuit unit 0 single-ended input bypass selection (odd analog channel) Can be written only when the target channel-dedicated sample-and-hold circuit setting is ADSHCR0.SHEN0 = 0. 0: Use the sample-and-hold circuit 1: Bypass the sample-and-hold circuit	R/W
17	SHSBN1	Channel-dedicated sample-and-hold circuit unit 1 single-ended input bypass selection (odd analog channel) Can be written only when the target channel-dedicated sample-and-hold circuit setting is ADSHCR0.SHEN1 = 0. 0: Use the sample-and-hold circuit 1: Bypass the sample-and-hold circuit	R/W
18	SHSBN2	Channel-dedicated sample-and-hold circuit unit 2 single-ended input bypass selection (odd analog channel) Can be written only when the target channel-dedicated sample-and-hold circuit setting is ADSHCR0.SHEN2 = 0. 0: Use the sample-and-hold circuit 1: Bypass the sample-and-hold circuit	R/W
19	—	This bit is read as 0. The write value should be 0.	R
20	SHSBN4	Channel-dedicated sample-and-hold circuit unit 4 single-ended input bypass selection (odd analog channel) Can be written only when the target channel-dedicated sample-and-hold circuit setting is ADSHCR1.SHEN4 = 0. 0: Use the sample-and-hold circuit 1: Bypass the sample-and-hold circuit	R/W

Bit	Symbol	Function	R/W
21	SHSBN5	Channel-dedicated sample-and-hold circuit unit 5 single-ended input bypass selection (odd analog channel) Can be written only when the target channel-dedicated sample-and-hold circuit setting is ADSHCR1.SHEN5 = 0. 0: Use the sample-and-hold circuit 1: Bypass the sample-and-hold circuit	R/W
22	SHSBN6	Channel-dedicated sample-and-hold circuit unit 6 single-ended input bypass selection (odd analog channel) Can be written only when the target channel-dedicated sample-and-hold circuit setting is ADSHCR1.SHEN6 = 0. 0: Use the sample-and-hold circuit 1: Bypass the sample-and-hold circuit	R/W
31:23	—	These bits are read as 0. The write value should be 0.	R/W

Note: Write access is permitted only when ADSHCRx.SHENn (x = 0, 1, n = 0, 1, 2, 4, 5, 6) of the target sample-and-hold circuit is set to 0. If ADSHCRx.SHENn is set to 1, writing 1 to this bit is ignored.

ADSHSBPCR register controls the channel-dedicated sample-and-hold circuit single-ended bypass mode.

SHSBPn (Channel-dedicated sample-and-hold circuit unit n single-end input bypass control for even-numbered channel) (n = 0 to 2, 4 to 6)

SHSBNn (Channel-dedicated sample-and-hold circuit unit n single-end input bypass control for odd-numbered channel) (n = 0 to 2, 4 to 6)

The SHSBPn bit and the SHSBNn bit control the analog input for A/D conversion when the channel-dedicated sample-and-hold circuit is enabled.

When the bit is set to 0, the analog output held by the channel-dedicated sample-and-hold circuit is A/D-converted.

When the bit is set to 1, the analog input of the channel-dedicated sample-and-hold circuit is A/D-converted. In this case, the analog output held by the channel-dedicated sample-and-hold circuit is ignored on the A/D conversion.

When the channel-dedicated sample-and-hold circuit is used in single-ended input mode, the analog input of unused channel of the channel-dedicated sample-and-hold circuit can be A/D-converted without going through the channel-dedicated sample-and-hold circuit.

53.2.5.8 ADSHDBPCR : Channel-dedicated Sample-and-hold Circuit Differential Input Mode Bypass Control Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x314

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SHDBP8	SHDBP7	SHDBP6	SHDBP5	SHDBP4	SHDBP3	SHDBP2	SHDBP1	SHDBP0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SHDBP0	Scan group 0 channel-dedicated sample-and-hold circuit differential input bypass select 0: Use the sample-and-hold circuit 1: Bypass the sample-and-hold circuit	R/W
1	SHDBP1	Scan group 1 channel-dedicated sample-and-hold circuit differential input bypass select 0: Use the sample-and-hold circuit 1: Bypass the sample-and-hold circuit	R/W

Bit	Symbol	Function	R/W
2	SHDBP2	Scan group 2 channel-dedicated sample-and-hold circuit differential input bypass select 0: Use the sample-and-hold circuit 1: Bypass the sample-and-hold circuit	R/W
3	SHDBP3	Scan group 3 channel-dedicated sample-and-hold circuit differential input bypass select 0: Use the sample-and-hold circuit 1: Bypass the sample-and-hold circuit	R/W
4	SHDBP4	Scan group 4 channel-dedicated sample-and-hold circuit differential input bypass select 0: Use the sample-and-hold circuit 1: Bypass the sample-and-hold circuit	R/W
5	SHDBP5	Scan group 5 channel-dedicated sample-and-hold circuit differential input bypass select 0: Use the sample-and-hold circuit 1: Bypass the sample-and-hold circuit	R/W
6	SHDBP6	Scan group 6 channel-dedicated sample-and-hold circuit differential input bypass select 0: Use the sample-and-hold circuit 1: Bypass the sample-and-hold circuit	R/W
7	SHDBP7	Scan group 7 channel-dedicated sample-and-hold circuit differential input bypass select 0: Use the sample-and-hold circuit 1: Bypass the sample-and-hold circuit	R/W
8	SHDBP8	Scan group 8 channel-dedicated sample-and-hold circuit differential input bypass select 0: Use the sample-and-hold circuit 1: Bypass the sample-and-hold circuit	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

ADSHDBPCR register controls the channel-dedicated sample-and-hold circuit differential input bypass mode.

SHDBPn (Channel-dedicated sample-and-hold circuit differential input bypass control for scan group n) (n = 0 to 8)

When the bit is set to 0, the analog output held by the channel-dedicated sample-and-hold circuit specified in the scan group is A/D-converted.

When the bit is set to 1, the analog input of the channel-dedicated sample-and-hold circuit specified in the scan group is A/D-converted. In this case, the analog output held by the channel-dedicated sample-and-hold circuit is ignored on the A/D conversion.

In the case that the disconnection detection assist function is needed when the channel-dedicated sample-and-hold circuits are used in differential input mode, the input mode must be changed to single-ended mode. However, if the input mode is changed to use the disconnection detection assist function built in the channel-dedicated sample-and-hold circuit, calibration of the channel-dedicated sample-and-hold circuit is required.

This function allows bypassing the channel-dedicated sample-and-hold circuits to use the disconnection detection assist function without recalibration of the channel-dedicated sample-and-hold circuits.

53.2.5.9 ADSHSDCR0 : Channel-dedicated Sample-and-hold Circuit Self-diagnosis Control Register 0

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x318

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	SHDIAG7[2:0]			—	SHDIAG6[2:0]			—	SHDIAG5[2:0]			—	SHDIAG4[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	SHDIAG3[2:0]			—	SHDIAG2[2:0]			—	SHDIAG1[2:0]			—	SHDIAG0[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SHDIAG0[2:0]	Scan group 0 channel-dedicated sample-and-hold circuit self-diagnostic voltage selection 000: Normal operation 001: Self-diagnostic voltage 0 V 010: Self-diagnostic voltage −VREFH 011: Self-diagnostic voltage +VREFH 100: Self-diagnostic voltage 0 V 101: Self-diagnostic voltage −VREFH/2 110: Self-diagnostic voltage +VREFH/2 111: Setting prohibited	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	SHDIAG1[2:0]	Scan group 1 channel-dedicated sample-and-hold circuit self-diagnostic voltage selection 000: Normal operation 001: Self-diagnostic voltage 0 V 010: Self-diagnostic voltage −VREFH 011: Self-diagnostic voltage +VREFH 100: Self-diagnostic voltage 0 V 101: Self-diagnostic voltage −VREFH/2 110: Self-diagnostic voltage +VREFH/2 111: Setting prohibited	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
10:8	SHDIAG2[2:0]	Scan group 2 channel-dedicated sample-and-hold circuit self-diagnostic voltage selection 000: Normal operation 001: Self-diagnostic voltage 0 V 010: Self-diagnostic voltage −VREFH 011: Self-diagnostic voltage +VREFH 100: Self-diagnostic voltage 0 V 101: Self-diagnostic voltage −VREFH/2 110: Self-diagnostic voltage +VREFH/2 111: Setting prohibited	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
14:12	SHDIAG3[2:0]	Scan group 3 channel-dedicated sample-and-hold circuit self-diagnostic voltage selection 000: Normal operation 001: Self-diagnostic voltage 0 V 010: Self-diagnostic voltage −VREFH 011: Self-diagnostic voltage +VREFH 100: Self-diagnostic voltage 0 V 101: Self-diagnostic voltage −VREFH/2 110: Self-diagnostic voltage +VREFH/2 111: Setting prohibited	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
18:16	SHDIAG4[2:0]	Scan group 4 channel-dedicated sample-and-hold circuit self-diagnostic voltage selection 000: Normal operation 001: Self-diagnostic voltage 0 V 010: Self-diagnostic voltage −VREFH 011: Self-diagnostic voltage +VREFH 100: Self-diagnostic voltage 0 V 101: Self-diagnostic voltage −VREFH/2 110: Self-diagnostic voltage +VREFH/2 111: Setting prohibited	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
22:20	SHDIAG5[2:0]	Scan group 5 channel-dedicated sample-and-hold circuit self-diagnostic voltage selection 000: Normal operation 001: Self-diagnostic voltage 0 V 010: Self-diagnostic voltage −VREFH 011: Self-diagnostic voltage +VREFH 100: Self-diagnostic voltage 0 V 101: Self-diagnostic voltage −VREFH/2 110: Self-diagnostic voltage +VREFH/2 111: Setting prohibited	R/W

Bit	Symbol	Function	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
26:24	SHDIAG6[2:0]	Scan group 6 channel-dedicated sample-and-hold circuit self-diagnostic voltage selection 000: Normal operation 001: Self-diagnostic voltage 0 V 010: Self-diagnostic voltage −VREFH 011: Self-diagnostic voltage +VREFH 100: Self-diagnostic voltage 0 V 101: Self-diagnostic voltage −VREFH/2 110: Self-diagnostic voltage +VREFH/2 111: Setting prohibited	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
30:28	SHDIAG7[2:0]	Scan group 7 channel-dedicated sample-and-hold circuit self-diagnostic voltage selection 000: Normal operation 001: Self-diagnostic voltage 0 V 010: Self-diagnostic voltage −VREFH 011: Self-diagnostic voltage +VREFH 100: Self-diagnostic voltage 0 V 101: Self-diagnostic voltage −VREFH/2 110: Self-diagnostic voltage +VREFH/2 111: Setting prohibited	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: For more details about restrictions, see section 53.3.16.6. Restrictions on Channel-dedicated Sample-and-hold Circuit.

The ADSHSDCR0 register controls the diagnostic function of the channel-dedicated sample-and-hold circuit.

53.2.5.10 ADSHSDCR1 : Channel-dedicated Sample-and-hold Circuit Self-diagnosis Control Register 1

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x31C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SHDIAG8[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SHDIAG8[2:0]	Scan group 8 channel-dedicated sample-and-hold circuit self-diagnostic voltage selection 000: Normal operation 001: Self-diagnostic voltage 0 V 010: Self-diagnostic voltage −VREFH 011: Self-diagnostic voltage +VREFH 100: Self-diagnostic voltage 0 V 101: Self-diagnostic voltage −VREFH/2 110: Self-diagnostic voltage +VREFH/2 111: Setting prohibited	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: For more details about restrictions, see section 53.3.16.6. Restrictions on Channel-dedicated Sample-and-hold Circuit.

The ADSHSDCR1 register controls the diagnostic function of the channel-dedicated sample-and-hold circuit.

53.2.5.11 ADSHCSCR : Channel-Dedicated Sample & Hold Circuit Constant Sampling Control Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xC04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SHCSST1	—	—	—	—	—	—	—	SHCSST0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SHCSST0	Channel-dedicated sample-and-hold circuit constant sampling control bit for A/D converter unit 0 (ADC0) When SHCSST0 is enabled, channel-dedicated sample-and-hold circuit sampling starts. 0: Disable channel-dedicated sample-and-hold circuit constant sampling 1: Enable channel-dedicated sample-and-hold circuit constant sampling	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	SHCSST1	Channel-dedicated sample-and-hold circuit constant sampling control bit for A/D converter unit 1 (ADC1) When SHCSST1 is enabled, channel-dedicated sample-and-hold circuit sampling starts. 0: Disable channel-dedicated sample-and-hold circuit constant sampling 1: Enable channel-dedicated sample-and-hold circuit constant sampling	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: This register must be used after calibration.
Note: Using this register is prohibited when channel-dedicated sample-and-hold circuit is not used.
Note: For more details about restrictions, see section 53.3.16.6. Restrictions on Channel-dedicated Sample-and-hold Circuit.

The ADSHCSCR register controls the constant sampling function of channel-dedicated sample-and-hold circuit.

53.2.5.12 ADREFCR : Internal Reference Voltage Monitor Enable Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x320

Bit position:	31																															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	VDE	Internal Reference Voltage A/D Conversion Select 0: Disable A/D conversion of internal reference voltage 1: Enable A/D conversion of internal reference voltage	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The ADREFCR register controls the function of the internal reference voltage circuit. When the A/D conversion of the internal reference voltage is performed, the VDE bit must be set to 1.

53.2.6 Digital Filter

53.2.6.1 ADDFSRn : A/D Converter Digital Filter Selection Register n (n = 0, 1)

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x340 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	DFSEL3[1:0]		—	—	—	—	—	—	DFSEL2[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	DFSEL1[1:0]		—	—	—	—	—	—	DFSEL0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	DFSEL0[1:0]	A/D Converter Unit n the 1st Digital Filter Characteristic Selection 0 1: Sinc filter Others: Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	DFSEL1[1:0]	A/D Converter Unit n the 2nd Digital Filter Characteristic Selection 0 1: Sinc filter Others: Setting prohibited	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
17:16	DFSEL2[1:0]	A/D Converter Unit n the 3rd Digital Filter Characteristic Selection 0 1: Sinc filter Others: Setting prohibited	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
25:24	DFSEL3[1:0]	A/D Converter Unit n the 4th Digital Filter Characteristic Selection 0 1: Sinc filter Others: Setting prohibited	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADDFSRn register selects the characteristics of the digital filter for the A/D converter unit n. The selection of the digital filter to be used at A/D conversion is specified by the ADDOPCRAm register (m = 0 to 32). If the digital filter function is used with prohibited values set to the ADDFSRn.DFSELx[1:0] (x = 0 to 3) bits, the A/D conversion characteristics and A/D conversion results are not guaranteed.

53.2.7 Self-calibration

53.2.7.1 ADCALSTCR : A/D Converter Self-calibration State Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x264

Bit position:	31										21					16							9									0
Bit field:	—	—	—	—	—	—	—	—	—	—	CALADCST[5:0]						—	—	—	—	—	—	CALADSST[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
9:0	CALADSST[9:0]	A/D Converter Self-calibration Sampling Time Configuration These bits set the sampling time in the range from 2 to 1023 states.	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
21:16	CALADCST[5:0]	A/D Converter Self-calibration Successive Approximation Time Configuration. These bits set the successive approximation time in the range from 3 to 63 states.	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

The ADCALSTCR register specifies the sampling and the A/D successive approximation times at the self-calibration operation for all A/D converter (ADC0, ADC1). The sampling time and successive approximation time should be set by the number of clock cycles based on the A/D conversion clock (ADCLK) to meet the values specified in section 62, Electrical Characteristics.

53.2.7.2 ADCALSHCR : Channel-dedicated Sample-and-hold Circuit Self-calibration State Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x2B0

Bit position: 31 18 16 7 0

Bit field: — — — — — — — — — — — — — CALSHHST[2:0] — — — — — — — — CALSHSST[7:0]

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0

Bit	Symbol	Function	R/W
7:0	CALSHSST[7:0]	Channel-dedicated Sample-and-hold Circuit Self-calibration Sampling Time Configuration These bits set the sampling time in the range from 4 to 255 states.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
18:16	CALSHHST[2:0]	Channel-dedicated Sample-and-hold Circuit Self-calibration Hold Mode Switching Time Configuration These bits set the hold mode switching time in the range from 2 to 7 states.	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

The ADCALSHCR register specifies the sampling time and the hold mode switching time at the self-calibration operation for all channel-dedicated sample-and-hold circuits (SH0 to SH2, SH4 to SH6). The sampling time and the hold mode switching time should be set by the number of clock cycles based on the A/D conversion clock (ADCLK) to meet the values specified in section 62, Electrical Characteristics.

53.2.7.3 ADUOFTRn : User Offset Table Register n (n = 0 to 7)

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x360 + 0x04 × n

Bit position: 31 15 0

Bit field: — — — — — — — — — — — — — — — — UOFSET[15:0]

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	UOFSET[15:0]	User Offset Table n 0x7FFF: + 32767 0x7FFE: + 32766 ⋮ 0x0002: + 2 0x0001: + 1 0x0000: 0 (without user offset) 0xFFFF: - 1 0xFFFE: - 2 ⋮ 0x8001: - 32767 0x8000: - 32768	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The ADUOFTRn register specifies the offset value in the User Offset Table n.

53.2.7.4 ADUGTRn : User Gain Table Register n (n = 0 to 7)

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x380 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15–0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UGAIN[15:0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	UGAIN[15:0]	User Gain Table n These bits set the gain value to be multiplied by the A/D conversion result. UGAIN[15:14]: Integer part of gain UGAIN[13:0]: Fractional part of gain	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The ADUGTRn register specifies the gain value in the user gain table n.

53.2.8 Limiter Clip Function

53.2.8.1 ADLIMINTCR : Limiter Clip Interrupt Enable Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x3A0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	LIMIE8	LIMIE7	LIMIE6	LIMIE5	LIMIE4	LIMIE3	LIMIE2	LIMIE1	LIMIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	LIMIE0 to LIMIE8	Limiter Clip Interrupt n Enable bit The suffix number of each bit symbol corresponds to the limiter clip interrupt number n. 0: Disable the limiter clip interrupt n 1: Enable the limiter clip interrupt n	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

The ADLIMINTCR register enables/disables the limiter clip interrupt n.

53.2.8.2 ADLIMTRn : Limiter Clip Table Register n (n = 0 to 7)

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x3A4 + 0x04 × n

Bit position:	31–16	15–0
Bit field:	LIMU[15:0]	LIML[15:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	LIML[15:0]	Limiter clip table n : Lower-side limit value	R/W
31:16	LIMU[15:0]	Limiter clip table n : Upper-side limit value	R/W

The ADLIMTRn register specifies the lower- and upper-side limit value of the limiter clip table n.

53.2.8.3 ADLIMGRSR : Limiter Clip Scan Group Status Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xD28

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	LIMGRF8	LIMGRF7	LIMGRF6	LIMGRF5	LIMGRF4	LIMGRF3	LIMGRF2	LIMGRF1	LIMGRF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	LIMGRF0 to LIMGRF8	Scan Group n Limiter Clip Flag The suffix number of each bit symbol corresponds to the scan group number n. 0: Limiter clip for scan group n is not detected 1: Limiter clip for scan group n is detected	R
31:9	—	These bits are read as 0.	R

The ADLIMGRSR register indicates whether the limiter clip occurred in the scanning operation for scan group n. Each flag can be cleared in the ADLIMGRSCR.

53.2.8.4 ADLIMCHSR0 : Limiter Clip Channel Status Register 0

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xD2C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	LIMCHF22	LIMCHF21	LIMCHF20	LIMCHF19	LIMCHF18	LIMCHF17	LIMCHF16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	LIMCHF15	LIMCHF14	LIMCHF13	LIMCHF12	LIMCHF11	LIMCHF10	LIMCHF9	LIMCHF8	LIMCHF7	LIMCHF6	LIMCHF5	LIMCHF4	LIMCHF3	LIMCHF2	LIMCHF1	LIMCHF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
22:0	LIMCHF0 to LIMCHF22	Analog Channel n: Limiter Clip Flag The suffix number of each bit symbol corresponds to the analog channel number n. 0: Limiter clip is not detected 1: Limiter clip is detected	R
31:23	—	These bits are read as 0.	R

The ADLIMCHSR0 register indicates whether the limiter clip occurred when the A/D conversion for the analog channel n is performed. Each flag can be cleared in ADLIMCHSCR0.

53.2.8.5 ADLIMEXSR : Extended Analog Limiter Clip Status Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xD38

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	LIMEXF22	LIMEXF21	LIMEXF20	—	LIMEXF18	LIMEXF17	LIMEXF16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	LIMEXF9	LIMEXF8	—	LIMEXF6	LIMEXF5	LIMEXF4	—	—	LIMEXF1	LIMEXF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LIMEXF0	Self-diagnosis Channel for A/D unit 0: Limiter Clip Flag 0: Limiter clip is not detected 1: Limiter clip is detected	R
1	LIMEXF1	Temperature Sensor Channel for A/D unit 1: Limiter Clip Flag 0: Limiter clip is not detected 1: Limiter clip is detected	R
3:2	—	These bits are read as 0.	R
4	LIMEXF4	Temperature Sensor Channel: Limiter Clip Flag 0: Limiter clip is not detected 1: Limiter clip is detected	R
5	LIMEXF5	Internal Reference Voltage Channel: Limiter Clip Flag 0: Limiter clip is not detected 1: Limiter clip is detected	R

Bit	Symbol	Function	R/W
6	LIMEXF6	VBATT 1/6 voltage monitor output Channel: Limiter Clip Flag 0: Limiter clip is not detected 1: Limiter clip is detected	R
7	—	This bit is read as 0.	R
8	LIMEXF8	D/A Converter 0 Channel: Limiter Clip Flag 0: Limiter clip is not detected 1: Limiter clip is detected	R
9	LIMEXF9	D/A Converter 1 Channel: Limiter Clip Flag 0: Limiter clip is not detected 1: Limiter clip is detected	R
15:10	—	These bits are read as 0.	R
16	LIMEXF16	Self-diagnosis Channel for Sample-and-hold circuit unit0: Limiter Clip Flag 0: Limiter clip is not detected 1: Limiter clip is detected	R
17	LIMEXF17	Self-diagnosis Channel for Sample-and-hold circuit unit1: Limiter Clip Flag 0: Limiter clip is not detected 1: Limiter clip is detected	R
18	LIMEXF18	Self-diagnosis Channel for Sample-and-hold circuit unit2: Limiter Clip Flag 0: Limiter clip is not detected 1: Limiter clip is detected	R
19	—	This bit is read as 0.	R
20	LIMEXF20	Self-diagnosis Channel for Sample-and-hold circuit unit4: Limiter Clip Flag 0: Limiter clip is not detected 1: Limiter clip is detected	R
21	LIMEXF21	Self-diagnosis Channel for Sample-and-hold circuit unit5: Limiter Clip Flag 0: Limiter clip is not detected 1: Limiter clip is detected	R
22	LIMEXF22	Self-diagnosis Channel for Sample-and-hold circuit unit6: Limiter Clip Flag 0: Limiter clip is not detected 1: Limiter clip is detected	R
31:23	—	These bits are read as 0.	R

The ADLIMEXSR register indicates whether the limiter clip occurred when the A/D conversion for the extended analog function is performed. Each flag can be cleared in the ADLIMEXSCR.

53.2.8.6 ADLIMGRSCR : Limiter Clip Scan Group Status Clear Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xD3C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	LIMGRC8	LIMGRC7	LIMGRC6	LIMGRC5	LIMGRC4	LIMGRC3	LIMGRC2	LIMGRC1	LIMGRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	LIMGRC0 to LIMGRC8	Scan Group n Limiter Clip Flag Clear The suffix number of each bit symbol corresponds to the scan group number n. 0: No effect 1: ADLIMGRSR.LIMGRFn is cleared	W
31:9	—	The write value should be 0.	W

The ADLIMGRSCR register clears the limiter clip flag for scan group n (ADLIMGRSR.LIMGRFn).

53.2.8.7 ADLIMCHSCR0 : Limiter Clip Channel Status Clear Register 0

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xD40

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	LIMCHC22	LIMCHC21	LIMCHC20	LIMCHC19	LIMCHC18	LIMCHC17	LIMCHC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	LIMCHC15	LIMCHC14	LIMCHC13	LIMCHC12	LIMCHC11	LIMCHC10	LIMCHC9	LIMCHC8	LIMCHC7	LIMCHC6	LIMCHC5	LIMCHC4	LIMCHC3	LIMCHC2	LIMCHC1	LIMCHC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
22:0	LIMCHC0 to LIMCHC22	Analog Channel n Limiter Clip Flag Clear bit The suffix number of each bit symbol corresponds to the analog channel number n. 0: No effect 1: ADLIMCHSR0.LIMCHFn is cleared	W
31:23	—	The write value should be 0.	W

The ADLIMCHSCR0 register clears the limiter clip flag for the analog channel n (ADLIMCHSR0.LIMCHFn).

53.2.8.8 ADLIMEXSCR : Extended Analog Limiter Clip Status Clear Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xD4C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	LIMEXC22	LIMEXC21	LIMEXC20	—	LIMEXC18	LIMEXC17	LIMEXC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	LIMEXC9	LIMEXC8	—	LIMEXC6	LIMEXC5	LIMEXC4	—	—	LIMEXC1	LIMEXC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LIMEXC0	Self-diagnosis channel for A/D unit 0: Limiter Clip Flag Clear 0: No effect 1: ADLIMEXSR.LIMEXC0 is cleared	W
1	LIMEXC1	Self-diagnosis channel for A/D unit 1: Limiter Clip Flag Clear 0: No effect 1: ADLIMEXSR.LIMEXC1 is cleared	W

Bit	Symbol	Function	R/W
3:2	—	The write value should be 0.	W
4	LIMEXC4	Temperature sensor channel: Limiter Clip Flag Clear 0: No effect 1: ADLIMEXSR.LIMEXC4 is cleared	W
5	LIMEXC5	Internal reference voltage channel: Limiter Clip Flag Clear 0: No effect 1: ADLIMEXSR.LIMEXC5 is cleared	W
6	LIMEXC6	VBATT 1/6 voltage monitor output channel: Limiter Clip Flag Clear 0: No effect 1: ADLIMEXSR.LIMEXC6 is cleared	W
7	—	The write value should be 0.	W
8	LIMEXC8	D/A converter 0 channel: Limiter Clip Flag Clear 0: No effect 1: ADLIMEXSR.LIMEXC8 is cleared	W
9	LIMEXC9	D/A converter 1 channel: Limiter Clip Flag Clear 0: No effect 1: ADLIMEXSR.LIMEXC9 is cleared	W
15:10	—	The write value should be 0.	W
16	LIMEXC16	Self-diagnosis channel for sample-and-hold circuit unit 0: Limiter Clip Flag Clear 0: No effect 1: ADLIMEXSR.LIMEXC16 is cleared	W
17	LIMEXC17	Self-diagnosis channel for sample-and-hold circuit unit 1: Limiter Clip Flag Clear 0: No effect 1: ADLIMEXSR.LIMEXC17 is cleared	W
18	LIMEXC18	Self-diagnosis channel for sample-and-hold circuit unit 2: Limiter Clip Flag Clear 0: No effect 1: ADLIMEXSR.LIMEXC18 is cleared	W
19	—	The write value should be 0.	W
20	LIMEXC20	Self-diagnosis channel for sample-and-hold circuit unit 4: Limiter Clip Flag Clear 0: No effect 1: ADLIMEXSR.LIMEXC20 is cleared	W
21	LIMEXC21	Self-diagnosis channel for sample-and-hold circuit unit 5: Limiter Clip Flag Clear 0: No effect 1: ADLIMEXSR.LIMEXC21 is cleared	W
22	LIMEXC22	Self-diagnosis channel for sample-and-hold circuit unit 6: Limiter Clip Flag Clear 0: No effect 1: ADLIMEXSR.LIMEXC22 is cleared	W
31:23	—	The write value should be 0.	W

The ADLIMEXSCR register clears the limiter clip flag for the extended analog function.

53.2.9 Compare Match Function

53.2.9.1 ADCMPENR : Compare Match Enable Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x400

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	CMPEN7	CMPEN6	CMPEN5	CMPEN4	CMPEN3	CMPEN2	CMPEN1	CMPEN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CMPEN0 to CMPEN7	Compare Match n Enable The suffix number of each bit symbol corresponds to the compare match number n. 0: Disable the compare match n 1: Enable the compare match n	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

The ADCMPENR register enables/disables the compare match n.

53.2.9.2 ADCMPINTCR : Compare Match Interrupt Enable Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x404

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	CMPIE3	CMPIE2	CMPIE1	CMPIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CMPIE0 to CMPIE3	Compare Match Interrupt n Enable The suffix number of each bit symbol corresponds to the compare match interrupt number n. 0: Disable the compare match interrupt n 1: Enable the compare match interrupt n	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

The ADCMPINTCR register enables/disables the compare match interrupt n.

53.2.9.3 ADCCMPCRn : Composite Compare Match Configuration Register n (n = 0, 1)

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x408 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	CCMPTBL7	CCMPTBL6	CCMPTBL5	CCMPTBL4	CCMPTBL3	CCMPTBL2	CCMPTBL1	CCMPTBL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CCMPCND[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CCMPCND[1:0]	Composite Compare Match Condition Selection 0 0: Logical disjunction (OR) conditions Others: Setting prohibited	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
23:16	CCMPTBL0 to CCMPTBL7	Composite Compare Match Condition Table Selection 0: Not use the compare match table m 1: Use the compare match table m	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

The ADCCMPCRn register configures the conditions for the composite compare match interrupt n.

CCMPCND[1:0] bits (Composite Compare Match Condition Selection)

The CCMPCND[1:0] bits select the generating condition for the composite compare match interrupt n.

CCMPTBLm bits (Composite Compare Match Condition Table Selection) (m = 0 to 7)

The CCMPTBLm bits select whether to use or not to use the compare match table m as the generating conditions of the composite compare match interrupt.

53.2.9.4 ADCMPMDR0 : Compare Match Mode Selection Register 0

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x448

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	CMPMD3[1:0]		—	—	—	—	—	—	CMPMD2[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CMPMD1[1:0]		—	—	—	—	—	—	CMPMD0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CMPMD0[1:0]	Compare Match 0 : Match Mode Selection 0 0: Generate the match event when high-side level or more 0 1: Generate the match event when low-side level or less 1 0: Generate the match event when high-side level or more, or low-side level or less 1 1: Generate the match event when low-side level or more and high-side level or less	R/W

Bit	Symbol	Function	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	CMPMD1[1:0]	Compare Match 1 : Match Mode Selection 0 0: Generate the match event when high-side level or more 0 1: Generate the match event when low-side level or less 1 0: Generate the match event when high-side level or more, or low-side level or less 1 1: Generate the match event when low-side level or more and high-side level or less	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
17:16	CMPMD2[1:0]	Compare Match 2 : Match Mode Selection 0 0: Generate the match event when high-side level or more 0 1: Generate the match event when low-side level or less 1 0: Generate the match event when high-side level or more, or low-side level or less 1 1: Generate the match event when low-side level or more and high-side level or less	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
25:24	CMPMD3[1:0]	Compare Match 3 : Match Mode Selection 0 0: Generate the match event when high-side level or more 0 1: Generate the match event when low-side level or less 1 0: Generate the match event when high-side level or more, or low-side level or less 1 1: Generate the match event when low-side level or more and high-side level or less	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADCMPMDR0 register selects the generating condition of each compare match event.

53.2.9.5 ADCMPMDR1 : Compare Match Mode Selection Register 1

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x44C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	CMPMD7[1:0]		—	—	—	—	—	—	CMPMD6[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CMPMD5[1:0]		—	—	—	—	—	—	CMPMD4[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CMPMD4[1:0]	Compare Match 4 : Match Mode Selection 0 0: Generate the match event when high-side level or more 0 1: Generate the match event when low-side level or less 1 0: Generate the match event when high-side level or more, or low-side level or less 1 1: Generate the match event when low-side level or more and high-side level or less	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	CMPMD5[1:0]	Compare Match 5 : Match Mode Selection 0 0: Generate the match event when high-side level or more 0 1: Generate the match event when low-side level or less 1 0: Generate the match event when high-side level or more, or low-side level or less 1 1: Generate the match event when low-side level or more and high-side level or less	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
17:16	CMPMD6[1:0]	Compare Match 6 : Match Mode Selection 0 0: Generate the match event when high-side level or more 0 1: Generate the match event when low-side level or less 1 0: Generate the match event when high-side level or more, or low-side level or less 1 1: Generate the match event when low-side level or more and high-side level or less	R/W

Bit	Symbol	Function	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
25:24	CMPMD7[1:0]	Compare Match 7 : Match Mode Selection 0 0: Generate the match event when high-side level or more 0 1: Generate the match event when low-side level or less 1 0: Generate the match event when high-side level or more, or low-side level or less 1 1: Generate the match event when low-side level or more and high-side level or less	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The ADCMPMDR1 register selects the generating condition of each compare match event.

53.2.9.6 ADCMPTBRn : Compare Match Table Register n (n = 0 to 7)

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x458 + 0x04 × n

Bit position:	31–16	15–0
Bit field:	CMPTBH[15:0]	CMPTBL[15:0]
Value after reset:	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	CMPTBL[15:0]	Compare Match Table n : Low-side level Set CMPTBH > CMPTBL	R/W
31:16	CMPTBH[15:0]	Compare Match Table n : High-side level Set CMPTBH > CMPTBL	R/W

The ADCMPTBRn register specifies the low-side level and the high-side level for the compare match table n.

53.2.9.7 ADCMPTBSR : Compare Match Table Status Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xD00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	CMPTBF7	CMPTBF6	CMPTBF5	CMPTBF4	CMPTBF3	CMPTBF2	CMPTBF1	CMPTBF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CMPTBF0 to CMPTBF7	Compare Match Table n Match Flag The suffix number of each bit symbol corresponds to the compare match table number n. 0: Match event with compare match table n is not detected 1: Match event with compare match table n is detected	R
31:8	—	These bits are read as 0.	R

The ADCMPTBSR register indicates whether the match event with the Compare Match Table n occurred during the A/D conversion. Each flag can be cleared in the ADCMPTBSCR.

53.2.9.8 ADCMPTBSCR : Compare Match Table Status Clear Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xD04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	CMPTBC7	CMPTBC6	CMPTBC5	CMPTBC4	CMPTBC3	CMPTBC2	CMPTBC1	CMPTBC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CMPTBC0 to CMPTBC7	Compare Match Table n: Match Flag Clear The suffix number of each bit symbol corresponds to the compare match table number n. 0: No effect 1: ADCMPTBSR.CMPTBFn is cleared	W
31:8	—	The write value should be 0.	W

The ADCMPTBSCR register clears the match flag for the compare match n (ADCMPTBSR.CMPTBFn).

53.2.9.9 ADCMPCHSR0 : Compare Match Channel Status Register 0

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xD08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	CMPCHF22	CMPCHF21	CMPCHF20	CMPCHF19	CMPCHF18	CMPCHF17	CMPCHF16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPCHF15	CMPCHF14	CMPCHF13	CMPCHF12	CMPCHF11	CMPCHF10	CMPCHF9	CMPCHF8	CMPCHF7	CMPCHF6	CMPCHF5	CMPCHF4	CMPCHF3	CMPCHF2	CMPCHF1	CMPCHF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
22:0	CMPCHF0 to CMPCHF22	Analog Channel n: Compare Match Flag The suffix number of each bit symbol corresponds to the analog channel number n. 0: Compare match is not detected 1: Compare match is detected	R
31:23	—	These bits are read as 0.	R

The ADCMPCHSR0 register indicates whether the compare match event for the Analog Channel n is detected. Each flag can be cleared in ADCMPCHSCR0.

53.2.9.10 ADCMPEXSR : Extended Analog Compare Match Status Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xD14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	CMPEXF22	CMPEXF21	CMPEXF20	—	CMPEXF18	CMPEXF17	CMPEXF16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CMPEXF9	CMPEXF8	—	CMPEXF6	CMPEXF5	CMPEXF4	—	—	CMPEXF1	CMPEXF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMPEXF0	Self-diagnosis channel for A/D unit 0: Compare Match Flag 0: Compare match is not detected 1: Compare match is detected	R
1	CMPEXF1	Self-diagnosis channel for A/D unit 1: Compare Match Flag 0: Compare match is not detected 1: Compare match is detected	R
3:2	—	These bits are read as 0.	R
4	CMPEXF4	Temperature sensor channel: Compare Match Flag 0: Compare match is not detected 1: Compare match is detected	R
5	CMPEXF5	Internal reference voltage channel: Compare Match Flag 0: Compare match is not detected 1: Compare match is detected	R
6	CMPEXF6	VBATT 1/6 voltage monitor output Channel: Compare Match Flag 0: Compare match is not detected 1: Compare match is detected	R
7	—	This bit is read as 0.	R
8	CMPEXF8	D/A converter 0 channel: Compare Match Flag 0: Compare match is not detected 1: Compare match is detected	R
9	CMPEXF9	D/A converter 1 channel: Compare Match Flag 0: Compare match is not detected 1: Compare match is detected	R
15:10	—	These bits are read as 0.	R
16	CMPEXF16	Self-diagnosis channel for sample-and-hold circuit unit 0: Compare Match Flag 0: Compare match is not detected 1: Compare match is detected	R
17	CMPEXF17	Self-diagnosis channel for sample-and-hold circuit unit 1: Compare Match Flag 0: Compare match is not detected 1: Compare match is detected	R
18	CMPEXF18	Self-diagnosis channel for sample-and-hold circuit unit 2: Compare Match Flag 0: Compare match is not detected 1: Compare match is detected	R
19	—	This bit is read as 0.	R
20	CMPEXF20	Self-diagnosis channel for sample-and-hold circuit unit 4: Compare Match Flag 0: Compare match is not detected 1: Compare match is detected	R

Bit	Symbol	Function	R/W
21	CMPEXF21	Self-diagnosis channel for sample-and-hold circuit unit 5: Compare Match Flag 0: Compare match is not detected 1: Compare match is detected	R
22	CMPEXF22	Self-diagnosis channel for sample-and-hold circuit unit 6: Compare Match Flag 0: Compare match is not detected 1: Compare match is detected	R
31:23	—	These bits are read as 0.	R

The ADCMPEXSR register indicates whether the compare match event for the extended analog function channel is detected. Each flag can be cleared in ADCMPEXSCR.

53.2.9.11 ADCMPCHSCR0 : Compare Match Channel Status Clear Register 0

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xD18

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	CMPCHC22	CMPCHC21	CMPCHC20	CMPCHC19	CMPCHC18	CMPCHC17	CMPCHC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPCHC15	CMPCHC14	CMPCHC13	CMPCHC12	CMPCHC11	CMPCHC10	CMPCHC9	CMPCHC8	CMPCHC7	CMPCHC6	CMPCHC5	CMPCHC4	CMPCHC3	CMPCHC2	CMPCHC1	CMPCHC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
22:0	CMPCHC0 to CMPCHC22	Analog channel n: Compare Match Flag Clear bit The suffix number of each bit symbol corresponds to the analog channel number n. 0: No effect 1: ADCMPCHSR0.CMPCHFn is cleared	W
31:23	—	The write value should be 0.	W

The ADCMPCHSCR0 register clears the compare match flag for the Analog Channel n (ADCMPCHSR0.CMPCHFn).

53.2.9.12 ADCMPEXSCR : Extended Analog Compare Match Status Clear Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xD24

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	CMPEXC22	CMPEXC21	CMPEXC20	—	CMPEXC18	CMPEXC17	CMPEXC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CMPEXC9	CMPEXC8	—	CMPEXC6	CMPEXC5	CMPEXC4	—	—	CMPEXC1	CMPEXC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMPEXC0	Self-diagnosis channel for A/D unit 0: Compare Match Flag Clear 0: No effect 1: ADCMPEXSR.CMPEXF0 is cleared	W

Bit	Symbol	Function	R/W
1	CMPEXC1	Self-diagnosis channel for A/D unit 1: Compare Match Flag Clear 0: No effect 1: ADCMPEXSR.CMPEXF1 is cleared	W
3:2	—	The write value should be 0.	W
4	CMPEXC4	Temperature sensor channel: Compare Match Flag Clear 0: No effect 1: ADCMPEXSR.CMPEXF4 is cleared	W
5	CMPEXC5	Internal reference voltage channel: Compare Match Flag Clear 0: No effect 1: ADCMPEXSR.CMPEXF5 is cleared	W
6	CMPEXC6	VBATT 1/6 voltage monitor output channel: Compare Match Flag Clear 0: No effect 1: ADCMPEXSR.CMPEXF6 is cleared	W
7	—	The write value should be 0.	W
8	CMPEXC8	D/A converter 0 channel: Compare Match Flag Clear 0: No effect 1: ADCMPEXSR.CMPEXF8 is cleared	W
9	CMPEXC9	D/A converter 1 channel: Compare Match Flag Clear 0: No effect 1: ADCMPEXSR.CMPEXF9 is cleared	W
15:10	—	The write value should be 0.	W
16	CMPEXC16	Self-diagnosis channel for sample-and-hold circuit unit 0: Compare Match Flag Clear 0: No effect 1: ADCMPEXSR.CMPEXF16 is cleared	W
17	CMPEXC17	Self-diagnosis channel for sample-and-hold circuit unit 1: Compare Match Flag Clear 0: No effect 1: ADCMPEXSR.CMPEXF17 is cleared	W
18	CMPEXC18	Self-diagnosis channel for sample-and-hold circuit unit 2: Compare Match Flag Clear 0: No effect 1: ADCMPEXSR.CMPEXF18 is cleared	W
19	—	The write value should be 0.	W
20	CMPEXC20	Self-diagnosis channel for sample-and-hold circuit unit 4: Compare Match Flag Clear 0: No effect 1: ADCMPEXSR.CMPEXF20 is cleared	W
21	CMPEXC21	Self-diagnosis channel for sample-and-hold circuit unit 5: Compare Match Flag Clear 0: No effect 1: ADCMPEXSR.CMPEXF21 is cleared	W
22	CMPEXC22	Self-diagnosis channel for sample-and-hold circuit unit 6: Compare Match Flag Clear 0: No effect 1: ADCMPEXSR.CMPEXF22 is cleared	W
31:23	—	The write value should be 0.	W

The ADCMPEXSCR register clears the compare match flag for the Extended Analog Function Channel (ADCMPEXSR.CMPEXFn).

53.2.10 Start and Stop Control of A/D Conversion

53.2.10.1 ADCALSTR : A/D Converter Self-calibration Start Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xC00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	ADCALST1[2:0]			—	—	—	—	—	ADCALST0[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	ADCALST0[2:0]*1	A/D Converter Unit 0 (ADC0) Self-calibration Start Control [Function of each bit] b0: Start bit of the internal circuit calibration b1: Start bit of the gain and offset calibration b2: Start bit of the channel-dedicated sample-and-hold circuit calibration [Write value] 0: No effect (not start the calibration) 1: Start the calibration	W
7:3	—	The write value should be 0.	W
10:8	ADCALST1[2:0]*1	A/D Converter Unit 1 (ADC1) Self-calibration Start Control [Function of each bit] b0: Start bit of the internal circuit calibration b1: Start bit of the gain and offset calibration b2: Start bit of the channel-dedicated sample-and-hold circuit calibration [Write value] 0: No effect (not start the calibration) 1: Start the calibration	W
31:11	—	The write value should be 0.	W

Note 1. To perform a calibration operation, write 1 to each of the calibration start bits simultaneously.

The ADCALSTR register controls the start of the self-calibration for each A/D converter.

ADCALSTm[2:0] bits (A/D converter Unit m (ADCm) Self-calibration Start Control bits) (m = 0, 1)

The ADCALSTm[2:0] bits control the start of the self-calibration operation of the A/D converter Unit m. When any bit of ADCALSTm[2:0] is set to 1, the self-calibration operation corresponding to each bit is started. When multiple bits of ADCALSTm[2:0] are set to 1 simultaneously, each self-calibration operation is performed in the order as following:

1. Internal circuit calibration
2. A/D converter gain/offset calibration
3. Channel-specific sample and hold circuit gain/offset calibration

Note: The self-calibration operation corresponding to the bits set to 0 in ADCALSTm[2:0] is skipped.

The ADCALSTm[2:0] bits must be written when all A/D converters are stopped (ADSR.ADACTm = 0 and ADSR.CALACTm = 0);

Writing to the ADCALSTm[2:0] bits are prohibited when A/D converters are operating.

53.2.10.2 ADSYSTR : A/D Conversion Synchronous Software Start Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xC10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ADSYST8	ADSYST7	ADSYST6	ADSYST5	ADSYST4	ADSYST3	ADSYST2	ADSYST1	ADSYST0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	ADSYST0 to ADSYST8	Scan Group n: A/D Conversion start The suffix number of each bit symbol corresponds to the scan group number n. 0: No effect 1: Start the A/D conversion of scan group n	W
31:9	—	The write value should be 0.	W

The ADSYSTR register controls the start of the A/D conversion of scan group n. This register is used to start the A/D conversions of several scan groups simultaneously by software.

ADSYSTn bit (Scan Group n: A/D Conversion start) (n = 0 to 8)

The ADSYSTn bit controls the start of the A/D conversion of scan group n. When the ADSYSTn bit is set to 1, the A/D conversion of scan group n is started. Setting 0 to the ADSYSTn bit has no effect on the operation. Set 1 simultaneously to the ADSYSTn bit of the scan group to start A/D conversion simultaneously.

53.2.10.3 ADSTRn : A/D Conversion Software Start Register n (n = 0 to 8)

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xC20 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ADST	Scan Group n A/D Conversion Start 0: No effect 1: Start the A/D conversion of scan group n	W
31:1	—	The write value should be 0.	W

The ADSTRn register controls the start of the A/D conversion of scan group n. This register is used to start the A/D conversion of a scan group by software.

If A/D conversions of several scan groups are to be started simultaneously by software, the ADSYSTR register should be used.

ADST bit (Scan Group n A/D Conversion Start)

The ADST bit controls the start of the A/D conversion of scan group n. When the ADST bit is set to 1, the A/D conversion of the scan group n is started. Setting 0 to the ADST bit has no effect on the operation.

53.2.10.4 ADSTOPR : A/D Conversion Stop Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xC60

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ADSTOP1	—	—	—	—	—	—	—	ADSTOP0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ADSTOP0	A/D Converter Unit 0 Force Stop 0: No effect 1: Force stop the operation of A/D converter unit 0	W
7:1	—	The write value should be 0.	W
8	ADSTOP1	A/D Converter Unit 1 Force Stop 0: No effect 1: Force stop the operation of A/D converter unit 1	W
31:9	—	The write value should be 0.	W

The ADSTOPR register forces each A/D converter to stop the operation. When the A/D converter is stopped with this register, the A/D conversion results with that A/D converter are not guaranteed.

ADSTOP0 bit (A/D Converter Unit 0 Force Stop)

The ADSTOP0 bit forces the A/D converter unit 0 to stop the A/D conversion operation. Setting 0 to the ADSTOP0 bit has no effect on the operation. When the ADSTOP0 bit is set to 1, the A/D converter unit 0 is forced to stop the operation. If the A/D converter unit 0 is forced to stop, the A/D conversion results with the A/D converter unit 0 are not guaranteed.

ADSTOP1 bit (A/D Converter Unit 1 Force Stop)

The ADSTOP1 bit forces the A/D converter unit 1 to stop the A/D conversion operation. Setting 0 to the ADSTOP1 bit has no effect on the operation. When the ADSTOP1 bit is set to 1, the A/D converter unit 1 is forced to stop the operation. If the A/D converter unit 1 is forced to stop, the A/D conversion results with the A/D converter unit 1 are not guaranteed.

53.2.11 Status Registers

53.2.11.1 ADSR : A/D Conversion Status Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xC80

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CALACT1	CALACT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADACT1	ADACT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ADACT0	A/D Converter Unit 0 (ADC0) A/D Conversion Status 0: ADC0 is not in A/D conversion 1: ADC0 is in A/D conversion	R
1	ADACT1	A/D Converter Unit 1 (ADC1) A/D Conversion Status 0: ADC1 is not in A/D conversion 1: ADC1 is in A/D conversion	R
15:2	—	These bits are read as 0.	R
16	CALACT0	A/D Converter Unit 0 (ADC0) : Calibration Status 0: ADC0 is not in the calibration operation 1: ADC0 is in the calibration operation	R
17	CALACT1	A/D Converter Unit 1 (ADC1) : Calibration Status 0: ADC1 is not in the calibration operation 1: ADC1 is in the calibration operation	R
31:18	—	These bits are read as 0.	R

The ADSR register indicates the status for each A/D converter operation.

53.2.11.2 ADGRSR : Scan Group Status Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xC84

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ACTGR8	ACTGR7	ACTGR6	ACTGR5	ACTGR4	ACTGR3	ACTGR2	ACTGR1	ACTGR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	ACTGR0 to ACTGR8*1	Scan Group n Status The suffix number of each bit symbol corresponds to the scan group number n. 0: Scan group n is idle 1: Scan group n is in the scanning operation	R

Bit	Symbol	Function	R/W
31:9	—	These bits are read as 0.	R

Note 1. In the group priority operation, if the scanning operation of a low priority group is interrupted, the ACTGRn bit of the corresponding scan group is set to 1.

The ADGRSR register indicates the operating status for each scan group.

53.2.11.3 ADSCANENDSR : Scan End Status Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xD50

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SCENDF8	SCENDF7	SCENDF6	SCENDF5	SCENDF4	SCENDF3	SCENDF2	SCENDF1	SCENDF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	SCENDF0 to SCENDF8*1	Scan Group n Scan End Flag The suffix number of each bit symbol corresponds to the scan group number n. 0: Scan group n has not been scanned 1: End of scan for scan group n is detected	R
31:9	—	These bits are read as 0.	R

Note 1. If the A/D conversion operation is stopped with the ADSTOPR register, the SCENDFn bit of the scan group in which the scan operation was stopped is not changed. (It is not set to 1)

The ADSCANENDSR register indicates whether the scanning operation for each scan group has been done. Each flag can be cleared in ADSCANENDSCR register.

53.2.11.4 ADSCANENDSCR : Scan End Status Clear Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xD54

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SCENDC8	SCENDC7	SCENDC6	SCENDC5	SCENDC4	SCENDC3	SCENDC2	SCENDC1	SCENDC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	SCENDC0 to SCENDC8	Scan Group n Scan End Flag Clear The suffix number of each bit symbol corresponds to the scan group number n. 0: No effect 1: ADSCANENDSR.SCENDFn is cleared	W
31:9	—	The write value should be 0.	W

The ADSCANENDSCR register clears the scan end flag for the scan group n.

53.2.11.5 ADERSR : A/D Conversion Error Status Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xC88

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADERF1	ADERF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ADERF0	A/D Converter Unit 0 (ADC0) Error Flag 0: Error is not detected 1: Error is detected	R
1	ADERF1	A/D Converter Unit 1 (ADC1) Error Flag 0: Error is not detected 1: Error is detected	R
31:2	—	These bits are read as 0.	R

The ADERSR register indicates whether the operating error has occurred in each A/D converter. Each flag can be cleared in ADERSCR register.

The A/D conversion data is not guaranteed in the A/D converter operation with an error detected.

53.2.11.6 ADERSCR : A/D Conversion Error Status Clear Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xC8C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADERCLR1	ADERCLR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ADERCLR0	A/D Converter Unit 0 Error Flag Clear 0: No effect 1: ADERSR.ADERF0 is cleared	W
1	ADERCLR1	A/D Converter Unit 1 Error Flag Clear 0: No effect 1: ADERSR.ADERF1 is cleared	W
31:2	—	The write value should be 0.	W

The ADERSCR register clears the error flag for A/D converter unit 0 or unit 1.

53.2.11.7 ADCALENDSR : A/D Converter Self-calibration End Status Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xC98

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CALENDF1	CALENDF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CALENDF0	A/D Converter Unit 0 Calibration End flag 0: End of the calibration is not detected 1: End of the calibration is detected	R
1	CALENDF1	A/D Converter Unit 1 Calibration End flag 0: End of the calibration is not detected 1: End of the calibration is detected	R
31:2	—	These bits are read as 0.	R

The ADCALENDSR register indicates the end of the calibration operation for the A/D converter unit 0 or unit 1. Each flag can be cleared in ADCALENDSCR register.

53.2.11.8 ADCALENDSCR : A/D Converter Self-calibration End Status Clear Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xC9C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CALENDC1	CALENDC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CALENDC0	A/D Converter Unit 0 Calibration End Flag Clear 0: No effect 1: ADCALENDSR.CALENDF0 is cleared	W
1	CALENDC1	A/D Converter Unit 1 Calibration End Flag Clear 0: No effect 1: ADCALENDSR.CALENDF1 is cleared	W
31:2	—	The write value should be 0.	W

The ADCALENDSCR register clears the Calibration End flag for the A/D converter unit 0 or unit 1.

53.2.11.9 ADOVFERSR : A/D Conversion Overflow Error Status Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xCA0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADOVFEF1	ADOVFEF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ADOVFEF0	A/D Converter Unit 0 (ADC0) Overflow Error Flag 0: ADC0 overflow error is not detected 1: ADC0 overflow error is detected	R
1	ADOVFEF1	A/D Converter Unit 1 (ADC1) Overflow Error Flag 0: ADC1 overflow error is not detected 1: ADC1 overflow error is detected	R
31:2	—	These bits are read as 0.	R

The ADOVFERSR register indicates whether the overflow error has occurred in an A/D conversion using A/D converter unit 0 or unit 1. Each flag can be cleared in ADOVFERSCR register.

53.2.11.10 ADOVFCHSR0 : A/D Conversion Overflow Channel Status Register 0

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xCA4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	OVFCHF22	OVFCHF21	OVFCHF20	OVFCHF19	OVFCHF18	OVFCHF17	OVFCHF16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OVFCHF15	OVFCHF14	OVFCHF13	OVFCHF12	OVFCHF11	OVFCHF10	OVFCHF9	OVFCHF8	OVFCHF7	OVFCHF6	OVFCHF5	OVFCHF4	OVFCHF3	OVFCHF2	OVFCHF1	OVFCHF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
22:0	OVFCHF0 to OVFCHF22	Analog Channel n: Overflow Flag The suffix number of each bit symbol corresponds to the analog channel number n. 0: Overflow is not detected 1: Overflow is detected	R
31:23	—	These bits are read as 0.	R

The ADOVFCHSR0 register indicates whether the overflow has occurred in the A/D conversion of the analog channel n. Each flag can be cleared in ADOVFCHSCR0 register.

53.2.11.11 ADOVFEXSR : Extended Analog A/D Conversion Overflow Status Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xCB0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	OVFEXF22	OVFEXF21	OVFEXF20	—	OVFEXF18	OVFEXF17	OVFEXF16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	OVFEXF9	OVFEXF8	—	OVFEXF6	OVFEXF5	OVFEXF4	—	—	OVFEXF1	OVFEXF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OVFEXF0	Self-diagnosis channel for A/D unit 0: Overflow Flag 0: Overflow is not detected 1: Overflow is detected	R
1	OVFEXF1	Self-diagnosis channel for A/D unit 1: Overflow Flag 0: Overflow is not detected 1: Overflow is detected	R
3:2	—	These bits are read as 0.	R
4	OVFEXF4	Temperature sensor channel: Overflow Flag 0: Overflow is not detected 1: Overflow is detected	R
5	OVFEXF5	Internal reference voltage channel: Overflow Flag 0: Overflow is not detected 1: Overflow is detected	R
6	OVFEXF6	VBATT 1/6 voltage monitor output channel: Overflow Flag 0: Overflow is not detected 1: Overflow is detected	R
7	—	This bit is read as 0.	R
8	OVFEXF8	D/A converter 0 channel: Overflow Flag 0: Overflow is not detected 1: Overflow is detected	R
9	OVFEXF9	D/A converter 1 channel: Overflow Flag 0: Overflow is not detected 1: Overflow is detected	R
15:10	—	These bits are read as 0.	R
16	OVFEXF16	Self-diagnosis channel for sample-and-hold circuit unit 0: Overflow Flag 0: Overflow is not detected 1: Overflow is detected	R
17	OVFEXF17	Self-diagnosis channel for sample-and-hold circuit unit 1: Overflow Flag 0: Overflow is not detected 1: Overflow is detected	R
18	OVFEXF18	Self-diagnosis channel for sample-and-hold circuit unit 2: Overflow Flag 0: Overflow is not detected 1: Overflow is detected	R
19	—	This bit is read as 0.	R
20	OVFEXF20	Self-diagnosis channel for sample-and-hold circuit unit 4: Overflow Flag 0: Overflow is not detected 1: Overflow is detected	R

Bit	Symbol	Function	R/W
21	OVFEXF21	Self-diagnosis channel for sample-and-hold circuit unit 5: Overflow Flag 0: Overflow is not detected 1: Overflow is detected	R
22	OVFEXF22	Self-diagnosis channel for sample-and-hold circuit unit 6: Overflow Flag 0: Overflow is not detected 1: Overflow is detected	R
31:23	—	These bits are read as 0.	R

The ADOVFEXSR register indicates whether the overflow occurred when the A/D conversion for the extended analog function is performed. Each flag can be cleared in ADOVFEXSCR.

53.2.11.12 ADOVFERSCR : A/D Conversion Overflow Error Status Clear Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xCB4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADOVFEC1	ADOVFEC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ADOVFEC0	A/D Converter Unit 0 (ADC0) Overflow Error Flag Clear 0: No effect 1: ADOVFERSR.ADOVFEF0 is cleared	W
1	ADOVFEC1	A/D Converter Unit 1 (ADC1) Overflow Error Flag Clear 0: No effect 1: ADOVFERSR.ADOVFEF1 is cleared	W
31:2	—	The write value should be 0.	W

The ADOVFERSCR register clears the Overflow Error flag for the A/D converter unit 0 or unit 1.

53.2.11.13 ADOVFCHSCR0 : A/D Conversion Overflow Channel Status Clear Register 0

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xCB8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	OVFCHC22	OVFCHC21	OVFCHC20	OVFCHC19	OVFCHC18	OVFCHC17	OVFCHC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OVFCHC15	OVFCHC14	OVFCHC13	OVFCHC12	OVFCHC11	OVFCHC10	OVFCHC9	OVFCHC8	OVFCHC7	OVFCHC6	OVFCHC5	OVFCHC4	OVFCHC3	OVFCHC2	OVFCHC1	OVFCHC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
22:0	OVFCHC0 to OVFCHC22	Analog Channel n: Overflow Flag Clear The suffix number of each bit symbol corresponds to the analog channel number n. 0: No effect 1: ADOVFCHSR0.OVFCHFn is cleared	W
31:23	—	The write value should be 0.	W

The ADOVFCHSCR0 register clears the Overflow flag for the analog channel n.

53.2.11.14 ADOVFEXSCR : Extended Analog A/D Conversion Overflow Status Clear Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xCC4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	OVFEXC22	OVFEXC21	OVFEXC20	—	OVFEXC18	OVFEXF17	OVFEXF16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	OVFEXF9	OVFEXF8	—	OVFEXC6	OVFEXC5	OVFEXC4	—	—	OVFEXC1	OVFEXC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OVFEXC0	Self-diagnosis Channel for A/D unit 0: Compare Match Flag Clear 0: No effect 1: ADOVFEXSR.OVFEXF0 is cleared	W
1	OVFEXC1	Self-diagnosis Channel for A/D unit 1: Compare Match Flag Clear 0: No effect 1: ADOVFEXSR.OVFEXF1 is cleared	W
3:2	—	The write value should be 0.	W
4	OVFEXC4	Temperature Sensor Channel: Compare Match Flag Clear 0: No effect 1: ADOVFEXSR.OVFEXF4 is cleared	W
5	OVFEXC5	Internal Reference Voltage Channel: Compare Match Flag Clear 0: No effect 1: ADOVFEXSR.OVFEXF5 is cleared	W
6	OVFEXC6	VBATT 1/6 voltage monitor output Channel: Compare Match Flag Clear 0: No effect 1: ADOVFEXSR.OVFEXF6 is cleared	W
7	—	The write value should be 0.	W
8	OVFEXF8	D/A Converter 0 Channel: Compare Match Flag Clear 0: No effect 1: ADOVFEXSR.OVFEXF8 is cleared	W
9	OVFEXF9	D/A Converter 1 Channel: Compare Match Flag Clear 0: No effect 1: ADOVFEXSR.OVFEXF9 is cleared	W
15:10	—	The write value should be 0.	W
16	OVFEXF16	Self-diagnosis Channel for Sample-and-hold circuit unit0: Compare Match Flag Clear 0: No effect 1: ADOVFEXSR.OVFEXF16 is cleared	W

Bit	Symbol	Function	R/W
17	OVFEXF17	Self-diagnosis Channel for Sample-and-hold circuit unit1: Compare Match Flag Clear 0: No effect 1: ADOVFEXSR.OVFEXF17 is cleared	W
18	OVFEXC18	Self-diagnosis Channel for Sample-and-hold circuit unit2: Compare Match Flag Clear 0: No effect 1: ADOVFEXSR.OVFEXF18 is cleared	W
19	—	The write value should be 0.	W
20	OVFEXC20	Self-diagnosis Channel for Sample-and-hold circuit unit4: Compare Match Flag Clear 0: No effect 1: ADOVFEXSR.OVFEXF20 is cleared	W
21	OVFEXC21	Self-diagnosis Channel for Sample-and-hold circuit unit5: Compare Match Flag Clear 0: No effect 1: ADOVFEXSR.OVFEXF21 is cleared	W
22	OVFEXC22	Self-diagnosis Channel for Sample-and-hold circuit unit6: Compare Match Flag Clear 0: No effect 1: ADOVFEXSR.OVFEXF22 is cleared	W
31:23	—	The write value should be 0.	W

The ADOVFEXSCR register clears the overflow flag for the extended analog function channels.

53.2.12 FIFO

53.2.12.1 ADFIFOCR : FIFO Control Register

Base address: ADC_B = 0x4033_8000

Offset address: 0x4C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	FIFOCE8	FIFOCE7	FIFOCE6	FIFOCE5	FIFOCE4	FIFOCE3	FIFOCE2	FIFOCE1	FIFOCE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	FIFOEN8	FIFOEN7	FIFOEN6	FIFOEN5	FIFOEN4	FIFOEN3	FIFOEN2	FIFOEN1	FIFOEN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	FIFOEN0 to FIFOEN8	Scan Group n FIFO Enable The suffix number of each bit symbol corresponds to the scan group number n. 0: Disable scan group n FIFO function 1: Enable scan group n FIFO function	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
24:16	FIFOCE0 to FIFOCE8	Scan Group n FIFO clear enable at scan start/resume 0: Scan group n FIFO clear disable 1: Scan group n FIFO clear enable	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

The ADFIFOCR register enables/disables the FIFO function of scan group n.

FIFOENn bit (Scan Group n FIFO Enable) (n = 0 to 8)

The FIFOENn bit enables or disables the FIFO function of scan group n. When the FIFOENn bit is set to 1, the FIFO function of scan group n is enabled, and the result of A/D conversion is stored to the FIFO. When the FIFOENn bit is set to 0, the FIFO function of scan group n is disabled.

The A/D conversion result can also be read from the A/D data or extended data registers.

FIFOCEn bit (Scan Group n FIFO clear enable at scan start/resume) (n = 0 to 8)

The FIFOCEn bit controls whether the scan group n FIFO clear function is enabled or disabled. When the FIFOCEn bit is set to 1, FIFO clear for scan group n is enabled and the FIFO is cleared when the scan group n scan starts/resumes. When the FIFOCEn bit is set to 0, FIFO clear for scan group n is disabled.

53.2.12.2 ADFIFOINTCR : FIFO Interrupt Control Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x4C4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	FIFOIE8	FIFOIE7	FIFOIE6	FIFOIE5	FIFOIE4	FIFOIE3	FIFOIE2	FIFOIE1	FIFOIE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	FIFOIE0 to FIFOIE8	Scan Group n FIFO Interrupt Enable The suffix number of each bit symbol corresponds to the scan group number n. 0: Disable scan group n FIFO interrupt 1: Enable scan group n FIFO interrupt	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

The ADFIFOINTCR register enables/disables the FIFO data read request interrupt and FIFO overflow interrupt for scan group n.

53.2.12.3 ADFIFOINTLR0 : FIFO Interrupt Generation Level Register 0

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x4C8

Bit position:	31												19			16													3			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	FIFOILV1[3:0]				—	—	—	—	—	—	—	—	—	—	—	—	FIFOILV0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	FIFOILV0[3:0]	Scan Group 0 FIFO Interrupt Output Timing Setting These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 7. Setting a value in the range from 8 to 15 is prohibited.	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W
19:16	FIFOILV1[3:0]	Scan Group 1 FIFO Interrupt Output Timing Setting These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 7. Setting a value in the range from 8 to 15 is prohibited.	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

The ADFIFOINTLR0 register specifies the generation timing of the FIFO data read request interrupt for the scan group 0 and 1.

53.2.12.4 ADFIFOINTLR1 : FIFO Interrupt Generation Level Register 1

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x4CC

Bit position:	31												19			16													3			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	FIFOILV3[3:0]				—	—	—	—	—	—	—	—	—	—	—	—	FIFOILV2[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	FIFOILV2[3:0]	Scan Group 2 FIFO Interrupt Output Timing Setting These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 7. Setting a value in the range from 8 to 15 is prohibited.	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W
19:16	FIFOILV3[3:0]	Scan Group 3 FIFO Interrupt Output Timing Setting These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 7. Setting a value in the range from 8 to 15 is prohibited.	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

The ADFIFOINTLR1 register specifies the generation timing of the FIFO data read request interrupt for the scan group 2 and 3.

53.2.12.5 ADFIFOINTLR2 : FIFO Interrupt Generation Level Register 2

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x4D0

Bit position:	31												19			16													3			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	FIFOILV5[3:0]				—	—	—	—	—	—	—	—	—	—	—	—	FIFOILV4[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	FIFOILV4[3:0]	Scan Group 4 FIFO Interrupt Output Timing Setting These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 7. Setting a value in the range from 8 to 15 is prohibited.	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W
19:16	FIFOILV5[3:0]	Scan Group 5 FIFO Interrupt Output Timing Setting These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 7. Setting a value in the range from 8 to 15 is prohibited.	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

The ADFIFOINTLR2 register specifies the generation timing of the FIFO data read request interrupt for the scan group 4 and 5.

53.2.12.6 ADFIFOINTLR3 : FIFO Interrupt Generation Level Register 3

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x4D4

Bit position:	31												19			16													3			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	FIFOILV7[3:0]				—	—	—	—	—	—	—	—	—	—	—	—	FIFOILV6[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	FIFOILV6[3:0]	Scan Group 6 FIFO Interrupt Output Timing Setting These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 7. Setting a value in the range from 8 to 15 is prohibited.	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W
19:16	FIFOILV7[3:0]	Scan Group 7 FIFO Interrupt Output Timing Setting These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 7. Setting a value in the range from 8 to 15 is prohibited.	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

The ADFIFOINTLR3 register specifies the generation timing of the FIFO data read request interrupt for the scan group 6 and 7.

53.2.12.7 ADFIFOINTLR4 : FIFO Interrupt Generation Level Register 4

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x4D8

Bit position:	31																												3			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FIFOILV8[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	FIFOILV8[3:0]	Scan Group 8 FIFO Interrupt Output Timing Setting These bits set the generation timing of FIFO interrupt. The FIFO data read request interrupt is generated when the number of vacant stages in FIFO becomes less than or equal to the value set by these bits. Set a value in the range from 0 to 7. Setting a value in the range from 8 to 15 is prohibited.	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

The ADFIFOINTLR4 register specifies the generation timing of the FIFO data read request interrupt for the scan group 8.

53.2.12.8 ADFIFOSR0 : FIFO Status Register 0

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xCD0

Bit position:	31												19			16													3			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	FIFOST1[3:0]				—	—	—	—	—	—	—	—	—	—	—	—	FIFOST0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
3:0	FIFOST0[3:0]	Number of vacant stages in FIFO for Scan Group 0	R
15:4	—	These bits are read as 0.	R
19:16	FIFOST1[3:0]	Number of vacant stages in FIFO for Scan Group 1	R
31:20	—	These bits are read as 0.	R

The ADFIFOSR0 register indicates the number of vacant stages in FIFO for scan group 0 and 1.

53.2.12.9 ADFIFOSR1 : FIFO Status Register 1

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xCD4

Bit position:	31												19			16													3			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	FIFOST3[3:0]				—	—	—	—	—	—	—	—	—	—	—	—	FIFOST2[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
3:0	FIFOST2[3:0]	Number of vacant stages in FIFO for Scan Group 2	R
15:4	—	These bits are read as 0.	R
19:16	FIFOST3[3:0]	Number of vacant stages in FIFO for Scan Group 3	R
31:20	—	These bits are read as 0.	R

The ADFIFOSR1 register indicates the number of vacant stages in FIFO for scan group 2 and 3.

53.2.12.10 ADFIFOSR2 : FIFO Status Register 2

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xCD8

Bit position:	31												19			16													3			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	FIFOST5[3:0]				—	—	—	—	—	—	—	—	—	—	—	—	FIFOST4[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
3:0	FIFOST4[3:0]	Number of vacant stages in FIFO for Scan Group 4	R
15:4	—	These bits are read as 0.	R
19:16	FIFOST5[3:0]	Number of vacant stages in FIFO for Scan Group 5	R

Bit	Symbol	Function	R/W
31:20	—	These bits are read as 0.	R

The ADFIFOSR2 register indicates the number of vacant stages in FIFO for scan group 4 and 5.

53.2.12.11 ADFIFOSR3 : FIFO Status Register 3

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xCDC

Bit position:	31–20	19–16	15–4	3–0
Bit field:	— (each bit)	FIFOST7[3:0]	— (each bit)	FIFOST6[3:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0	1 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0	1 0 0 0

Bit	Symbol	Function	R/W
3:0	FIFOST6[3:0]	Number of vacant stages in FIFO for Scan Group 6	R
15:4	—	These bits are read as 0.	R
19:16	FIFOST7[3:0]	Number of vacant stages in FIFO for Scan Group 7	R
31:20	—	These bits are read as 0.	R

The ADFIFOSR3 register indicates the number of vacant stages in FIFO for scan group 6 and 7.

53.2.12.12 ADFIFOSR4 : FIFO Status Register 4

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xCE0

Bit position:	31–4	3–0
Bit field:	— (each bit)	FIFOST8[3:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 0 0 0

Bit	Symbol	Function	R/W
3:0	FIFOST8[3:0]	Number of vacant stages in FIFO for Scan Group 8	R
31:4	—	These bits are read as 0.	R

The ADFIFOSR4 register indicates the number of vacant stages in FIFO for scan group 8.

53.2.12.13 ADFIFODCR : FIFO Data Clear Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xCF0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	FIFODC8	FIFODC7	FIFODC6	FIFODC5	FIFODC4	FIFODC3	FIFODC2	FIFODC1	FIFODC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	FIFODC0 to FIFODC8	Scan Group n FIFO Data Clear The suffix number of each bit symbol corresponds to the scan group number n. 0: No effect 1: Clear the data of scan group n FIFO	W
31:9	—	The write value should be 0.	W

The ADFIFODCR register clears the data in FIFO for each scan group.

FIFODCn bit (Scan Group n FIFO Data Clear) (n = 0 to 8)

The FIFODCn bit clears the data in FIFO for scan group n. Writing 1 to the FIFODCn clears the data in the FIFO of scan group n. Writing 0 has no effect on the operation, and the data in FIFO is not cleared.

53.2.12.14 ADFIFOERSR : FIFO Error Status Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xCF4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	FIFOFLF8	FIFOFLF7	FIFOFLF6	FIFOFLF5	FIFOFLF4	FIFOFLF3	FIFOFLF2	FIFOFLF1	FIFOFLF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	FIFOOVF8	FIFOOVF7	FIFOOVF6	FIFOOVF5	FIFOOVF4	FIFOOVF3	FIFOOVF2	FIFOOVF1	FIFOOVF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	FIFOOVF0 to FIFOOVF8	Scan Group n FIFO Overflow Flag The suffix number of each bit symbol corresponds to the scan group number n. 0: No overflow 1: FIFO overflow is detected	R
15:9	—	These bits are read as 0.	R
24:16	FIFOFLF0 to FIFOFLF8	Scan Group n FIFO Data Read Request Flag The suffix number of each bit symbol corresponds to the scan group number n. 0: FIFO Data Read Request is not detected. 1: FIFO Data Read Request is detected.	R
31:25	—	These bits are read as 0.	R

The ADFIFOERSR register indicates the status of the FIFO. Each flag can be cleared in ADFIFOERSCR.

FIFOOVFn bit (Scan Group n FIFO Overflow Flag) (n = 0 to 8)

The FIFOOVFn bit indicates whether the overflow occurred in the FIFO of the scan group n.

When FIFOOVFn = 0, an overflow is not detected in the FIFO. When FIFOOVFn = 1, an overflow occurred in the FIFO and the A/D conversion result cannot be stored to the FIFO. FIFOOVFn can be cleared in ADFIFOERSCR.

FIFOFLFn bit (Scan Group n FIFO Data Read Request Flag) (n = 0 to 8)

The FIFOFLFn bit indicates whether a FIFO data read request in scan group n has been detected. The setting and clearing conditions of FIFOFLFn are shown as follows:

[Setting condition]

When the condition of ADFIFOSRm.FIFOSTn[3:0] ≤ ADFIFOINTLRm.FIFOILVn[3:0] is detected.

[Clearing condition]

When ADFIFOERSCR.FIFOFLCn is written to 1 under the condition of ADFIFOSRm.FIFOSTn[3:0] > ADFIFOINTLRm.FIFOILVn[3:0].

When the condition of ADFIFOSRm.FIFOSTn[3:0] > ADFIFOINTLRm.FIFOILVn[3:0] is occurred by the read access to ADFIFODRn register with DMAC or DTC.

Note: m = 0 to 4, n = 0 to 8

53.2.12.15 ADFIFOERSCR : FIFO Error Status Clear Register

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0xCF8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	FIFOFLC8	FIFOFLC7	FIFOFLC6	FIFOFLC5	FIFOFLC4	FIFOFLC3	FIFOFLC2	FIFOFLC1	FIFOFLC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	FIFOOVFC8	FIFOOVFC7	FIFOOVFC6	FIFOOVFC5	FIFOOVFC4	FIFOOVFC3	FIFOOVFC2	FIFOOVFC1	FIFOOVFC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	FIFOOVFC0 to FIFOOVFC8	Scan Group n FIFO Overflow Flag Clear The suffix number of each bit symbol corresponds to the scan group number n. 0: No effect 1: ADFIFOERSR.FIFOOVFn is cleared	W
15:9	—	The write value should be 0.	W
24:16	FIFOFLC0 to FIFOFLC8	Scan Group n FIFO Data Read Request Flag Clear The suffix number of each bit symbol corresponds to the scan group number n. 0: No effect 1: ADFIFOERSR.FIFOFLFn is cleared	W
31:25	—	The write value should be 0.	W

ADFIFOERSCR clears the FIFO overflow flags and the FIFO data read request flags for scan group n.

53.2.13 Data Register

53.2.13.1 ADDRn : A/D Data Register n (n = 0 to 22)

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x2000 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15–0
Bit field:	ERR	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DATA[15:0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	DATA[15:0]	A/D conversion data	R
30:16	—	These bits are read as 0.	R
31	ERR	A/D conversion data error status 0: No error (the A/D conversion data is valid) 1: Error is detected (the A/D conversion data is not guaranteed)	R

The ADDRn registers are read-only registers to read A/D conversion results.

DATA[15:0] bits (A/D conversion data)

The DATA[15:0] bits indicate the data of the A/D conversion result for the analog input channel n.

The data format of the A/D conversion results is determined by the setting of the virtual channel m (m = 0 to 32) to which the analog input channel n is assigned.

ERR bit (A/D conversion data error status)

The ERR bit indicates the error status in the A/D conversion for the analog input channel n.

When ERR = 0, the A/D conversion data is valid. When ERR = 1, the A/D conversion data is invalid, and the accuracy of the A/D conversion data is not guaranteed.

53.2.13.2 ADEXDRn : A/D Extended Analog Data Register n (n = 0, 1, 4, 5, 6, 8, 9, 16, 17, 18, 20, 21, 22)

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x2180 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ERR	—	—	—	—	DIAGSR[2:0]			—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DATA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DATA[15:0]	A/D conversion data	R
23:16	—	These bits are read as 0.	R
26:24	DIAGSR[2:0]*1	Self-diagnosis Status*1	R
30:27	—	These bits are read as 0.	R
31	ERR	A/D Conversion Error Status 0: No error (the A/D conversion data is valid) 1: Error is detected (the A/D conversion data is not guaranteed)	R

Note 1. The ADEXDRn (n = 0, 1, 16 to 18, 20 to 22) registers have the DIAGSR[2:0] bits. The DIAGSR[2:0] bits in the ADEXDRn (n = 4 to 6, 8, 9) registers are reserved bits.

The ADEXDRn registers are read-only registers to read A/D conversion results of the extended analog function. The ADEXDRn registers and the extended analog functions are as follows:

- ADEXDR0 : LIMEXF0 Self-diagnosis Channel for A/D unit0
- ADEXDR1 : LIMEXF1 Self-diagnosis Channel for A/D unit1
- ADEXDR4 : LIMEXF4 Temperature Sensor Channel
- ADEXDR5 : LIMEXF5 Internal Reference Voltage Channel
- ADEXDR6 : LIMEXF6 VBATT 1/6 voltage monitor output
- ADEXDR8 : LIMEXF8 D/A Converter 0 Channel
- ADEXDR9 : LIMEXF9 D/A Converter 1 Channel
- ADEXDR16 : LIMEXF16 Self-diagnosis Channel for Sample-and-hold circuit unit0
- ADEXDR17 : LIMEXF17 Self-diagnosis Channel for Sample-and-hold circuit unit1
- ADEXDR18 : LIMEXF18 Self-diagnosis Channel for Sample-and-hold circuit unit2

- ADEXDR20 : LIMEXF20 Self-diagnosis Channel for Sample-and-hold circuit unit4
- ADEXDR21 : LIMEXF21 Self-diagnosis Channel for Sample-and-hold circuit unit5
- ADEXDR22 : LIMEXF22 Self-diagnosis Channel for Sample-and-hold circuit unit6

DATA[15:0] bits (A/D conversion data)

The DATA[15:0] bits indicate the data of the A/D conversion result for the extended analog function channel. The data format of the A/D conversion results is determined by the setting of the virtual channel m (m = 0 to 36) to which the extended analog function channel is assigned.

DIAGSR[2:0] bits (Self-diagnosis Status)

The DIAGSR[2:0] bits indicate the voltage setting of the self-diagnosis channel.

The voltage setting value is selected among the setting values of the ADSGDCRm.DIAGVAL[2:0] bits (m = 0 to 8), ADSHSDCR0.SHDIAGm bits (m = 0 to 7) and ADSHSDCR1.SHDIAG8 bits according to the setting value of ADCHCRn.CNVCS[6:0].

The DIAGSR[2:0] bits are available in the ADEXDR0, ADEXDR1, ADEXDR16, ADEXDR17, ADEXDR18, ADEXDR20, ADEXDR21 and ADEXDR22 registers.

The DIAGSR[2:0] bits in ADEXDRn except above are reserved.

ERR bit (A/D Conversion Error Status)

The ERR bit indicates the error status in the A/D conversion for the extended analog functions.

When ERR = 0, the A/D conversion data is valid. When ERR = 1, the A/D conversion data is invalid, and the accuracy of the A/D conversion data is not guaranteed.

53.2.13.3 ADFIFODRn : FIFO Data Register n (n = 0 to 8)

Base address: ADC_B = 0x4033_8000
ADC_B_NS = 0x5033_8000

Offset address: 0x2200 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ERR	CH[6:0]							—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DATA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DATA[15:0]	A/D Conversion Data	R
23:16	—	These bits are read as 0.	R
30:24	CH[6:0]	A/D Conversion Channel Number	R
31	ERR	A/D Conversion Data Error Status 0: No error (the A/D conversion data is valid) 1: Error is detected (the A/D conversion data is not guaranteed)	R

The ADFIFODRn register is used to read data stored in the FIFO of scan group n.

DATA[15:0] bits (A/D Conversion Data)

The DATA[15:0] bits are read-only bits that read the A/D conversion data stored in the scan group n FIFO.

CH[6:0] bits (A/D Conversion Channel Number)

The CH[6:0] bits indicate the channel number of the A/D conversion data that is read from the DATA[15:0] bits.

ERR bit (A/D Conversion Data Error Status)

The ERR bit indicates the error status in the A/D conversion data of the analog channel indicated by the CH[6:0] bits.

When ERR = 0, the A/D conversion data is valid. When ERR = 1, the A/D conversion data is invalid, and the accuracy of the A/D conversion data is not guaranteed.

53.3 Operation

53.3.1 A/D Conversion Clock

A/D conversion clock (ADCLK) is the operation clock of ADC. The A/D converters (ADC0 and ADC1) are operated and controlled by ADCLK as the basic clock. Figure 53.4 shows clock structure of ADC16H.

ADCLK is generated from the clock source and the division ratio selected in ADCLKCR register. The frequency of ADCLK should be set so that PCLKA ≥ ADCLK. Also, the frequency of ADCLK should be set so that it is within the guaranteed operating range specified in the electrical characteristics. For details, see section 62, Electrical Characteristics.

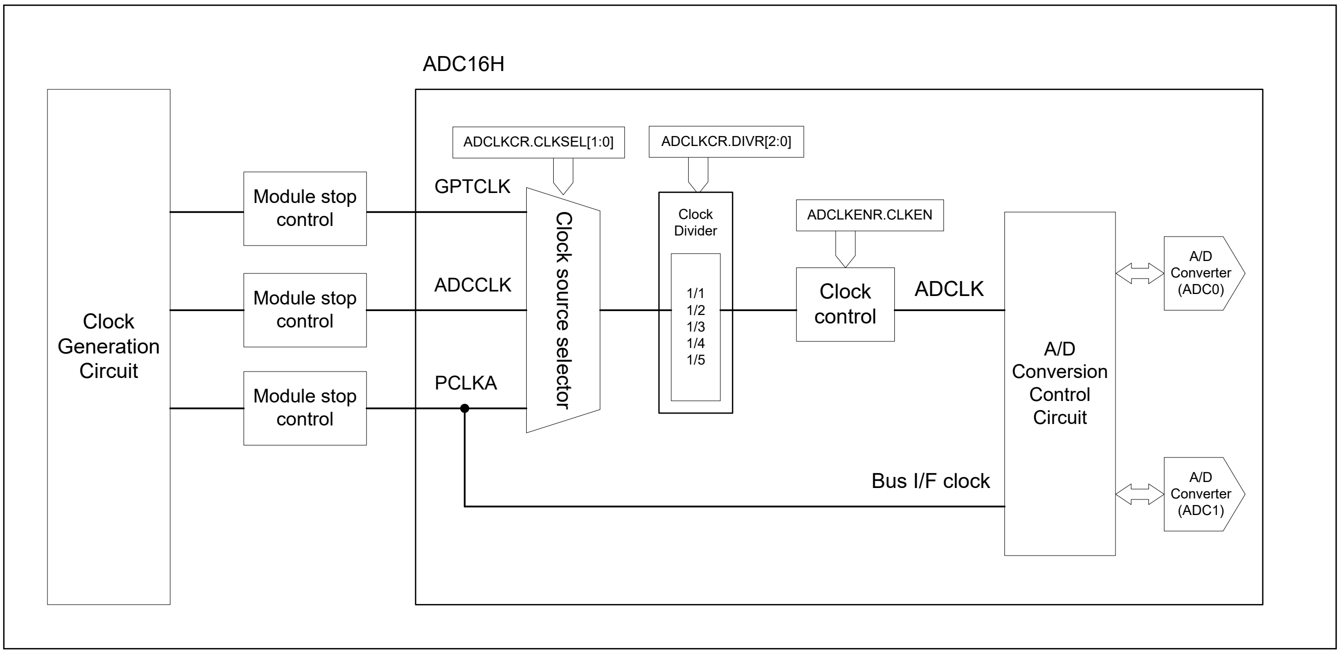


Figure 53.4 Clock structure

53.3.2 A/D Converter Operation Mode

53.3.2.1 SAR Mode

In SAR mode, A/D converter operates as a Nyquist type A/D converter. Analog channels are sampled one time and converted analog to digital by Successive Approximation Register (SAR) method. The features of SAR mode are shown in the following:

[Features of SAR mode]

- Fast conversion: A/D conversion data can be acquired with the shortest group delay.
- A/D conversion of up to 8 analog channels (virtual channels) is possible per scan group.
- Both single-ended input and differential input are supported (digital filter function is not supported)

For details about operation of A/D conversion in SAR mode, see section 53.3.7. Scanning Operation. For restrictions on SAR mode, see section 53.11.14. Restrictions on SAR Mode.

53.3.2.2 Oversampling Mode

In Oversampling mode, A/D converter operates as Oversampling-type A/D converter. Analog channels are oversampled and converted analog to digital by Noise Shaping Successive Approximation Register method (NS-SAR). The features of Oversampling mode are shown in the following:

[Features of Oversampling mode]

- High accuracy conversion: A/D conversion data can be acquired with Noise Shaping technology and Digital Filter function.*1
- A/D conversion of up to 8 analog channels (virtual channels) is possible per scan group.
- Both of single-ended input and differential input are supported.

Note 1. Digital Filter function must be used in Oversampling mode. It is prohibited to operate without digital filter function in Oversampling mode.

For details about operation of A/D conversion in Oversampling mode, see section 53.3.7. Scanning Operation. For restrictions on Oversampling mode, see section 53.11.15. Restrictions on Oversampling Mode.

53.3.2.3 Hybrid Mode

Hybrid mode has both features of SAR mode and Oversampling mode. In Hybrid mode, A/D converter operates as Oversampling-type A/D converter. Analog channels are oversampled and converted analog to digital by Noise Shaping Successive Approximation Register (NS-SAR) method. In the hybrid mode, while switching channels for each sampling, multiple analog channels are oversampled. The features of Hybrid mode are shown in the following:

[Features of Hybrid mode]

- High accuracy conversion: A/D conversion data can be acquired with Noise Shaping technology and Digital Filter function.*1
- Fast conversion: A/D conversion data can be acquired with the shortest group delay when in the Continuous scan mode and after initial delay time.
- A/D conversion of up to 4 analog channels (virtual channels) is possible per scan group.
- Both of single-ended input and differential input are supported.

Note 1. Digital Filter function must be used in Hybrid mode. It is prohibited to operate without digital filter function in Hybrid mode.

For details about operation of A/D conversion in Hybrid mode, see section 53.3.7. Scanning Operation. For restrictions on Hybrid mode, see section 53.11.16. Restrictions on Hybrid Mode.

53.3.3 Single-ended Input and Differential Input

ADC16H supports Single-ended input and Differential input. The selection of Single-ended input or Differential input is specified in ADCHCRn.AINMD (n = 0 to 32) bit. For the analog channels that support differential input, see Table 53.4 and Table 53.5.

Single-ended Input

In Single-ended input mode, the difference between the voltage of the analog channel (signal source) and the analog reference ground voltage (ADC0: VREFL0, ADC1: VREFL) is A/D-converted.

Differential Input

In Differential input mode, the even-numbered analog channel is used as the non-inverting input (+) (A_{INP}) and the odd-numbered analog channel is used as the inverting input (−) (A_{INN}), then the difference voltage between the non-inverting input (+) and the inverting input (−) ($A_{INP} - A_{INN}$) is A/D-converted.*1

For A/D conversion with Differential input mode, select the non-inverting input (+) channel*2 of the target of differential pair in ADCHCRn.CNVCS[6:0] (n = 0 to 32) bits, and select the Differential input mode in ADCHCRn.AINMD (n = 0 to 32) bit.

It is prohibited to perform A/D conversion for channels that do not support differential input with Differential input mode. In that case, A/D conversion result is not guaranteed.

Note 1. The differential input pair is a combination of channels with analog channel number 2i and 2i + 1 (i = 0, 1, 2, 3, ...), and they are named ANxxxP and ANxxxN (xxx = 000, 002, 004, ...). The combination of discontinuous analog channel numbers or the combination of analog channel numbers 2i − 1 and 2i cannot be used as a differential input pair.

Note 2. It is the even-numbered analog channel which is named as ANxxxP (xxx = 000, 002, 004, …).

53.3.4 Analog Channel

The analog channel is the source of the analog signal that is targeted for A/D conversion. The analog channels consist of the following elements:

- Analog input channels: A/D conversion channels for the analog input from MCU’s I/O pin as the signal source.
- Extended analog channels: A/D conversion channels for the source of the analog signal inside the MCU.

To perform A/D conversion for the analog channels, assignments to the virtual channel and scan group are required. For each element, refer to each item.

53.3.5 Virtual Channel

The virtual channel is a group of registers that stores the A/D conversion configurations for an analog channel. The virtual channels can be specified the configurations for A/D conversion such as the selection of the analog channels, the optional settings for A/D conversion, the data processing method of the A/D conversion data, the assignment to scan groups, and so on.

To perform the A/D conversion of the analog channel, the analog channel should be assigned to any of the virtual channels. Then that virtual channel should be assigned to one of the scan groups. See Figure 53.5 for the relationship between the analog channels and the virtual channels and scan groups. Figure 53.6 shows the A/D conversion order in Figure 53.5 settings.

The virtual channel can be assigned to only one scan group. If the same analog channel is to be converted within the different scan group, or the same analog channel is to be converted several times within the same scan group, assign the analog channel to several virtual channels. When performing the A/D conversion more than once using more than one virtual channel within the same scan group for the same analog channel, the FIFO function should be used in combination.

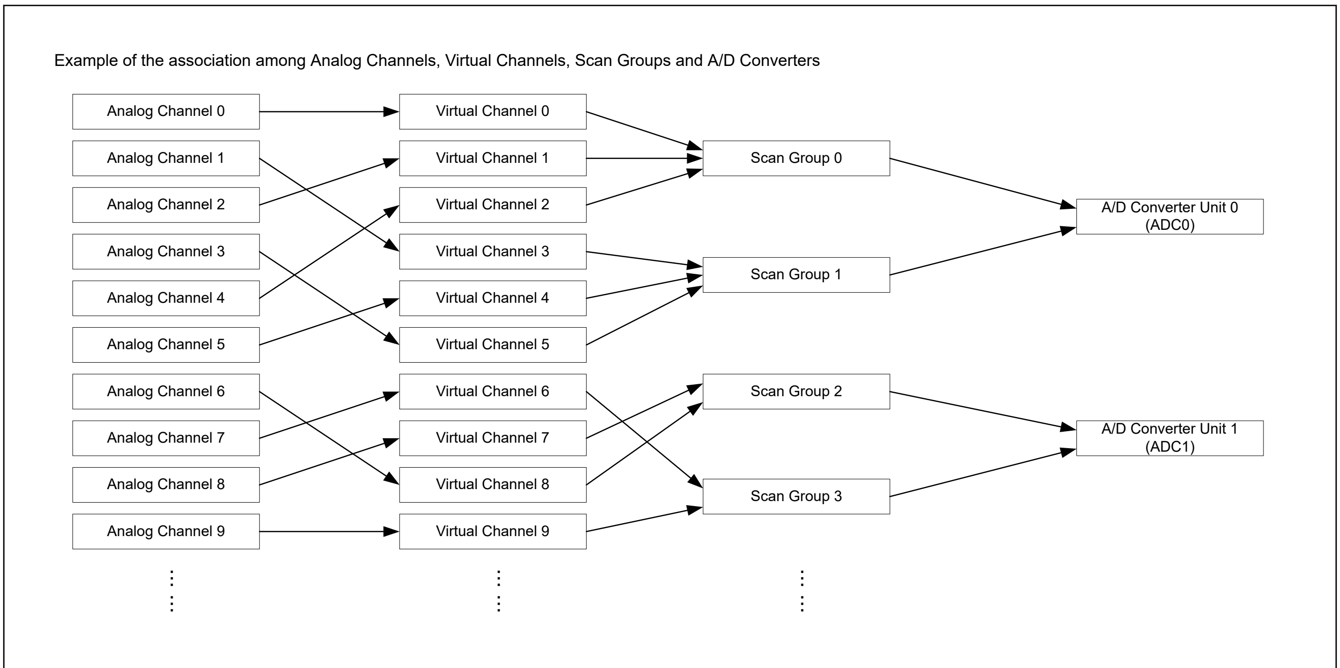


Figure 53.5 The concept of virtual channel

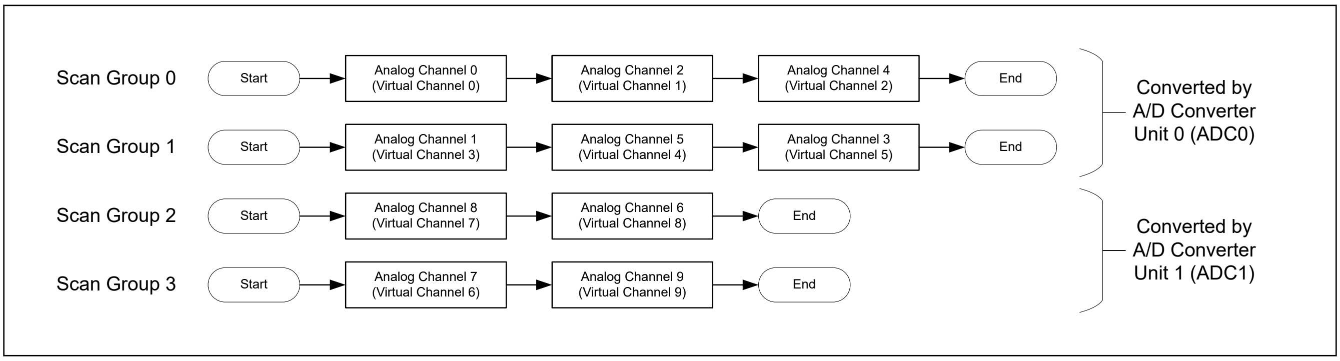


Figure 53.6 Example of A/D conversion order at the scanning operation

53.3.6 Scan Group

The scan group is a group of the analog channels that perform A/D conversion in a scanning operation. To perform the A/D conversion using the scan group, configure the following steps:

- Assign the analog channel for the A/D conversion to the virtual channel.
- Assign the virtual channel to the scan group.
- Assign the scan group to A/D converter.

See Figure 53.5 and Figure 53.6 for the relationship among the analog channels, the virtual channels, and the scan groups.

A single scan group can be assigned virtual channels in the following ranges, according to its operation mode.

- SAR mode: up to 8 virtual channels per scan group
- Oversampling mode: up to 8 virtual channels per scan group
- Hybrid mode: from 2 to 4 virtual channels per scan group

If more than the maximum number of virtual channels are allocated to one scan group, A/D conversion is performed on the 8th (SAR mode or Oversampling mode) or 4th (Hybrid mode) channels in ascending order of virtual channel numbers. Subsequent channels are not subject to A/D conversion.

Notes on the scan groups

The virtual channels assigned to one scan group should be assigned the analog channels that can be converted by the same A/D converter. If any analog channels cannot be converted by the specified A/D converter are included, the A/D conversion result of those channels are undefined.

In Hybrid mode, you must assign at least 2 virtual channels to a scan group. Operation is not guaranteed when any virtual channel is not assigned or only 1 virtual channel is assigned.

53.3.7 Scanning Operation

The scanning operation is the operation of sequential A/D conversion for the analog channels. The modes of the scanning operation are the followings:

- Single scan mode
  - In single scan mode, one scan group is scanned once for each A/D conversion start trigger input.
  - Each time an A/D conversion start trigger is input, A/D conversion is performed once for each analog channel assigned to that scan group.
- Continuous scan mode
  - The continuous scan mode repeats the scanning operation of one scan group.
  - When an A/D conversion start trigger is input, A/D conversion of each analog channel assigned to that scan group is repeated until A/D conversion stop processing is performed.
- Background continuous scan mode
  - The background continuous scan mode repeats the scanning operation of a single scanning group.

  - Scanning operation starts when the first A/D conversion start trigger is input.
  - A/D conversions of each analog channel assigned to that scan group are repeated until A/D conversion stop processing is performed. (Continuous scanning operation)
  - When an A/D conversion start trigger is input during continuous scanning operation, the A/D conversion data at that point is output. (Unless an A/D conversion start trigger is input, the A/D conversion data are not output and the values of the A/D data registers and FIFO data registers are not updated.)
- One-channel continuous scan mode
  - One-channel continuous scan mode repeats the scanning operation of one analog channel and can change conversion channel by a start trigger.
  - The first start trigger activates A/D conversion, and the same channel is continuously converted until the next start trigger is input.
  - Whenever an A/D conversion channel is switched by a start trigger, A/D conversion is suspended and resumed.
- Fixed-channel continuous scan mode
  - Fixed-channel continuous scan mode operates with a combination of fixed channels operating as Hybrid and non-fixed channels operating as SAR.
  - Scan a fixed number of fixed channels first, then scan one non-fixed channel assigned to the scan group. One non-fixed channel is switched in sequence with each scan. And repeat these operations.
  - A/D conversion data will not be output until conversion is completed according to the number of TAPs in the digital filter.

The scan modes supported by each operation mode are shown in Table 53.6.

Table 53.6 Scan mode and operation mode

Scan mode	Operation mode of A/D converter		
	SAR mode	Oversampling mode	Hybrid mode
Single scan mode	✓	✓	✓
Continuous scan mode	✓	✓	✓
Background continuous scan mode	—	—	✓
One-channel continuous scan mode	—	✓	—
Fixed-channel continuous scan mode	—	—	✓

Note: ✓: available, —: not available

The scanning operations are performed on a scan group basis. When the scanning operation is started, A/D conversion is performed for each analog channel based on the virtual channel setting. If the scanning operation starts for 2 or more scan group assigned to the same A/D converter at the same time, the scanning operation of the group with the lowest scan group number is performed.

The A/D conversion order of the analog channel in the scanning operation is the ascending order of the virtual channel numbers assigned to the scan group. The channels with the lowest virtual channel numbers are performed the A/D conversion at the forward end of the scan group. The channels with larger virtual channel numbers are performed the A/D conversion at the backward of the scan group.

See Figure 53.5 and Figure 53.6 for the relationship of the conversion order of the respective analog channel in the scanning operation.

The scanning operations in each operation mode are shown in the following.

53.3.7.1 SAR Mode – Single Scan Mode

Table 53.7 and Figure 53.7 show the operation example in SAR mode – Single scan mode.

Table 53.7 Example of the scanning operation in SAR mode – Single scan mode

Step	Operation
1	When a software trigger or a trigger from a peripheral module is input, the scanning operation of the scan group corresponding to the trigger is started.
2	Each time an A/D conversion of one analog channel is completed, A/D conversion result is stored to the data register (ADDRi (i = 0 to 22), ADEXDRj (j = 0, 1, 4, 5, 6, 8, 9, 16, 17, 18, 20, 21, 22)). If FIFO is used, the A/D conversion result is also stored to the FIFO data register (ADFIFODRk (k = 0 to 8)).
3	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
4	During the scanning operation, ADGRSR.ACTGRn (n = 0 to 8) bit corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0, 1) bit corresponding to the A/D converter performing the A/D conversion is also set to 1. When scanning operation is complete, each bit is cleared to 0 and A/D converter enters the idle state.

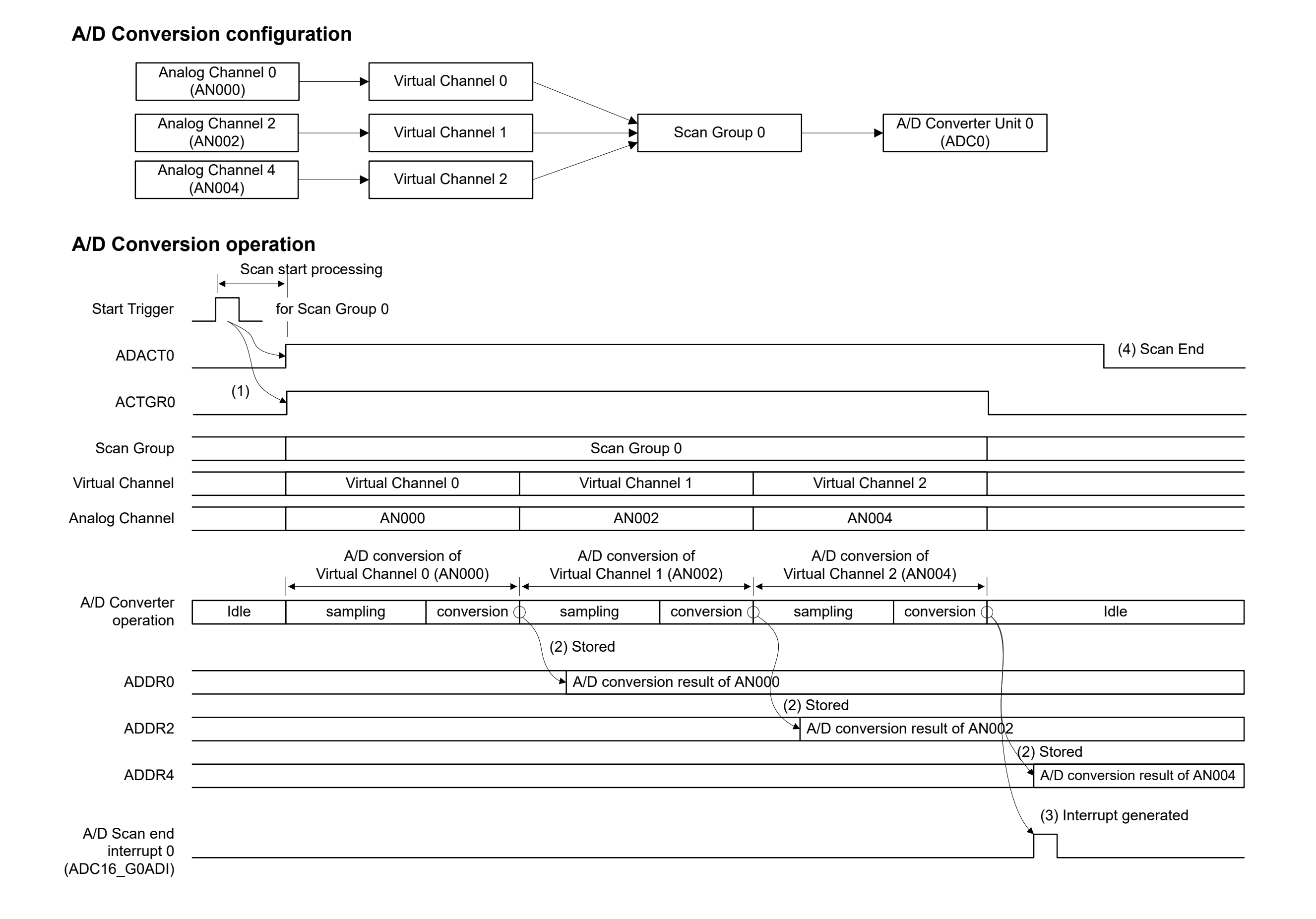


Figure 53.7 Example of the scanning operation in SAR mode – Single scan mode

53.3.7.2 SAR Mode – Continuous Scan Mode

Table 53.8 and Figure 53.8 show the operation example in SAR mode – Continuous scan mode.

Table 53.8 Example of the scanning operation in SAR mode – Continuous scan mode (1 of 2)

Step	Operation
1	When a software trigger or a trigger from a peripheral module is input, the scanning operation of the scan group corresponding to the trigger is started. When scanning operation starts, ADGRSR.ACTGRn (n = 0 to 8) corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0, 1) bit corresponding to the A/D converter performing the A/D conversion is also set to 1.

Table 53.8 Example of the scanning operation in SAR mode – Continuous scan mode (2 of 2)

Step	Operation
2	Each time an A/D conversion of one analog channel is completed, A/D conversion result is stored to the data register (ADDRi (i = 0 to 22), ADEXDRj (j = 0, 1, 4, 5, 6, 8, 9, 16, 17, 18, 20, 21, 22)). If FIFO is used, the A/D conversion result is also stored to the FIFO data register (ADFIFODRk (k = 0 to 8)).
3	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
4	Until the A/D conversion stop process is performed, Step 2 to 3 are repeated, and the scanning operation continues. To stop the A/D conversion, you must follow to section 53.6.4. Force Stops the A/D Conversion Operation.

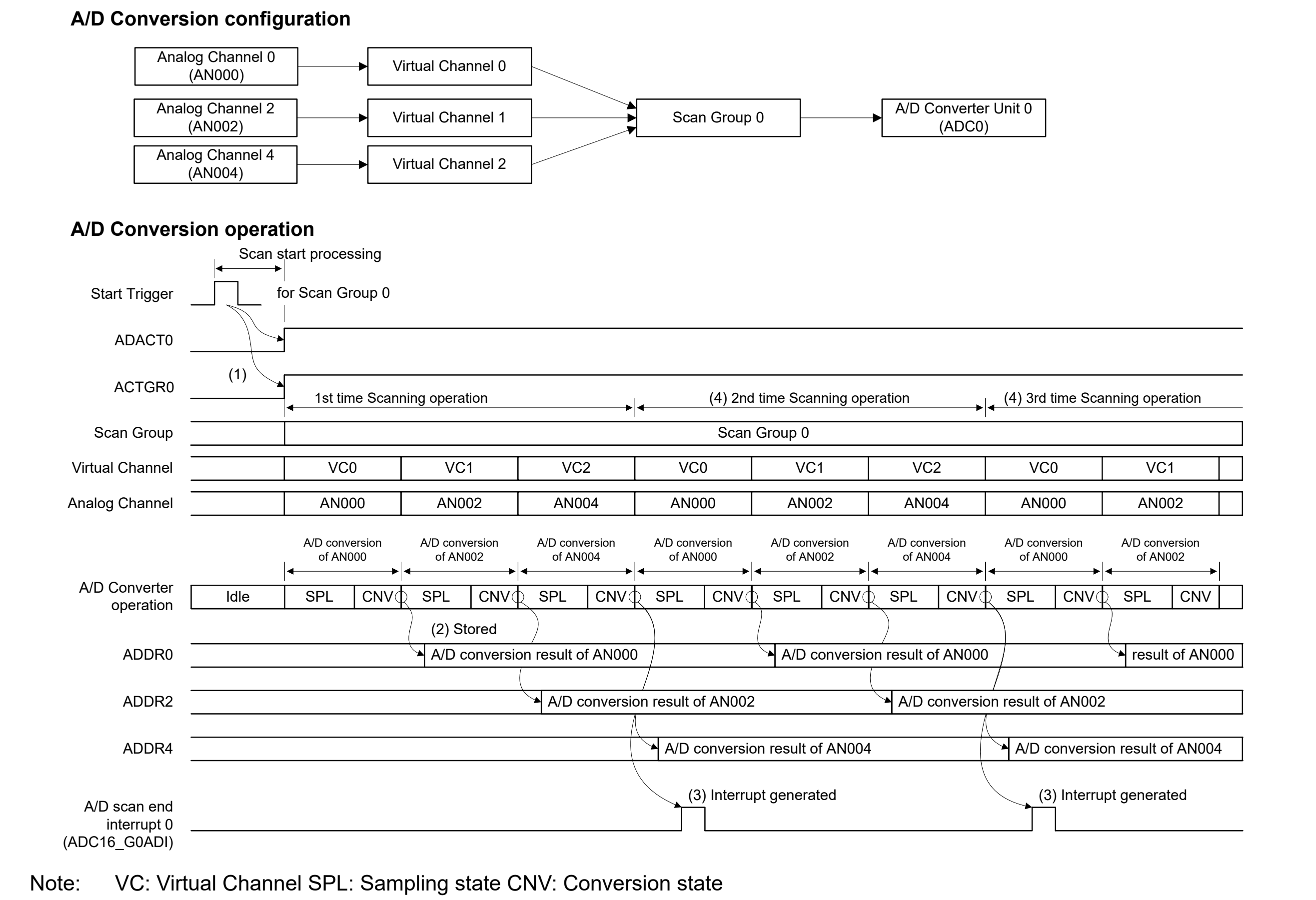


Figure 53.8 Example of scanning operation in SAR mode – Continuous scan mode

53.3.7.3 Oversampling Mode – Single Scan Mode

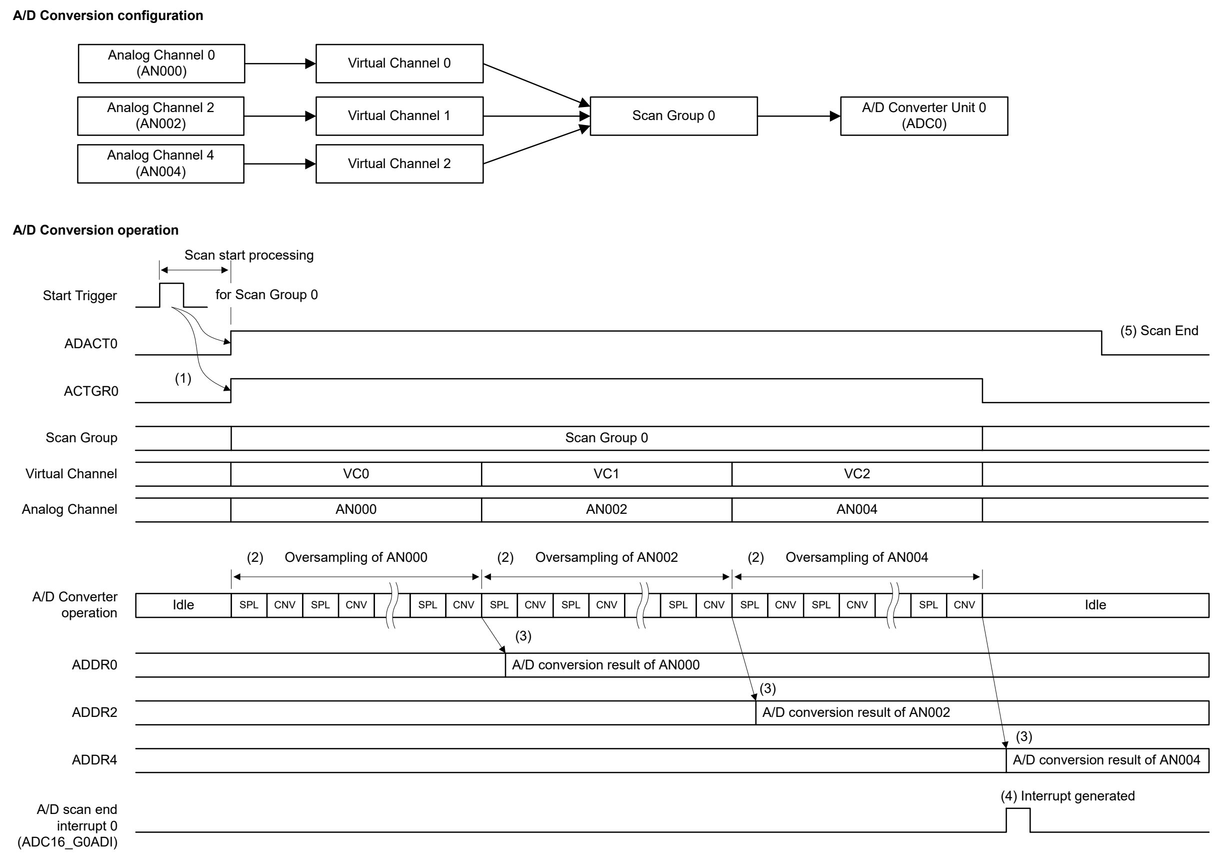
Table 53.9 and Figure 53.9 show the operation example in Oversampling mode – Single scan mode.

Table 53.9 Example of the scanning operation in Oversampling mode – Single scan mode (1 of 2)

Step	Operation
1	When a software trigger or a trigger from a peripheral module is input, the scanning operation of the scan group corresponding to the trigger is started.
2	Each analog channel is oversampled according to the number of TAPs in the digital filter and the number of A/D-converted value addition/averaging times. The oversampled data stored in the digital filter are discarded after the oversampling of each analog channel is completed.
3	A/D conversion data of each analog channel is output after oversampling. The A/D conversion data is stored in the data register (ADDRi (i = 0 to 22), ADEXDRj (j = 0, 1, 4, 5, 6, 8, 9, 16, 17, 18, 20, 21, 22)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 8)).

Table 53.9 Example of the scanning operation in Oversampling mode – Single scan mode (2 of 2)

Step	Operation
4	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
5	During the scanning operation, ADGRSR.ACTGRn (n = 0 to 8) bit corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0, 1) bit corresponding to the A/D converter performing the A/D conversion is also set to 1. When scanning operation is complete, each bit is cleared to 0 and A/D converter enters the idle state.



Note: VC: Virtual Channel
SPL: Sampling state
CNV: Conversion state

Figure 53.9 Example of scanning operation in Oversampling mode – Single scan mode

53.3.7.4 Oversampling Mode – Continuous Scan Mode

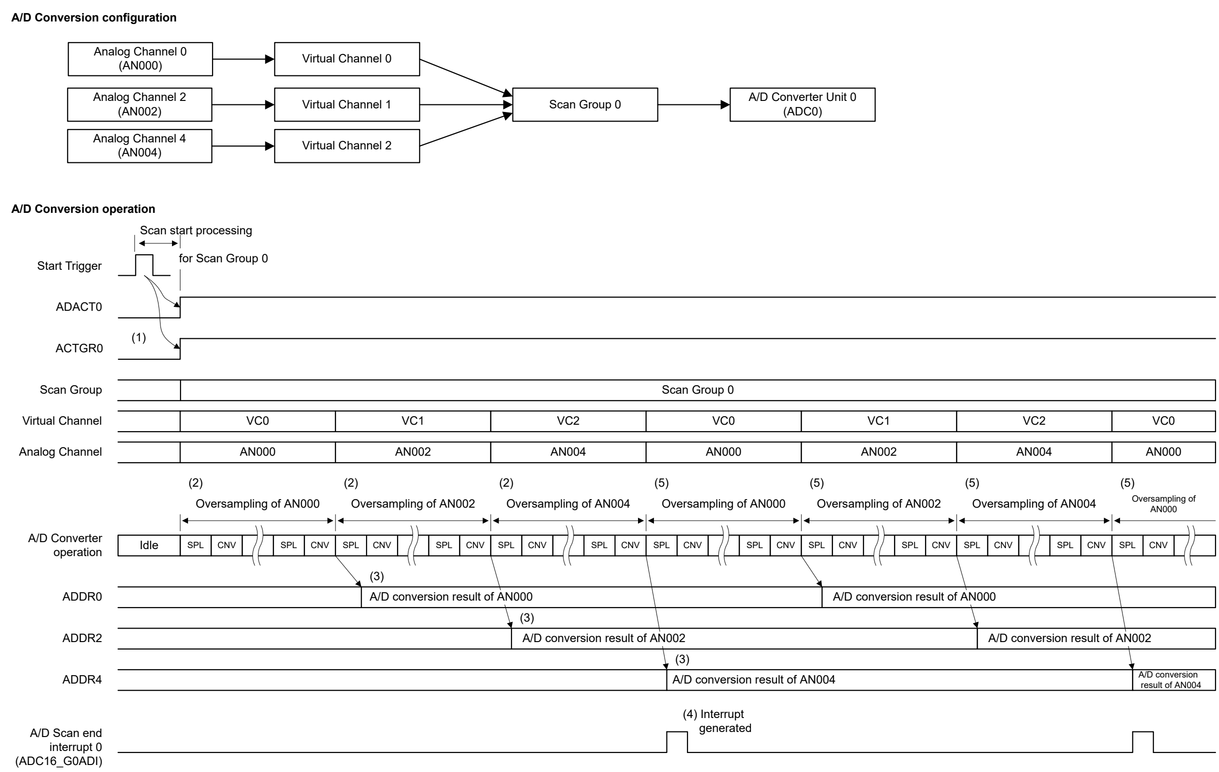
Table 53.10 and Figure 53.10 show the operation example in Oversampling mode – Continuous scan mode.

Table 53.10 Example of the scanning operation in Oversampling mode – Continuous scan mode (1 of 2)

Step	Operation
1	When a software trigger or a trigger from a peripheral module is input, the scanning operation of the scan group corresponding to the trigger is started. When scanning operation starts, ADGRSR.ACTGRn (n = 0 to 8) corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0, 1) bit corresponding to the A/D converter performing the A/D conversion is also set to 1.
2	Each analog channel is oversampled according to the number of TAPs in the digital filter and the number of A/D-converted value addition/averaging times. The oversampled data stored in the digital filter are discarded after the oversampling of each analog channel is completed.

Table 53.10 Example of the scanning operation in Oversampling mode – Continuous scan mode (2 of 2)

Step	Operation
3	A/D conversion data of each analog channel is output after oversampling. The A/D conversion data is stored in the data register (ADDRi (i = 0 to 22), ADEXDRj (j = 0, 1, 4, 5, 6, 8, 9, 16, 17, 18, 20, 21, 22)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 8)).
4	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
5	Until the A/D conversion stop process is performed, Step 2 to 4 are repeated, and the scanning operation continues. To stop A/D conversion, you must follow to section 53.6.4. Force Stops the A/D Conversion Operation.



Note: VC: Virtual Channel
SPL: Sampling state
CNV: Conversion state

Figure 53.10 Example of scanning operation in Oversampling mode – Continuous scan mode

53.3.7.5 Oversampling Mode – One-channel Continuous Scan Mode

Table 53.11 and Figure 53.11 show the operation example in Oversampling mode – One-channel continuous scan mode.

Table 53.11 Example of scanning operation in Oversampling mode – One-channel continuous scan mode (1 of 2)

Step	Operation
1	When a software trigger or a trigger from a peripheral module is input, the scanning operation of the scan group corresponding to the trigger is started. When scanning operation starts, ADGRSR.ACTGRn (n = 0 to 8) corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0, 1) bit corresponding to the A/D converter performing the A/D conversion is also set to 1.
2	Each time an A/D conversion of one analog channel is completed, A/D conversion result is stored to the data register (ADDRi (i = 0 to 22), ADEXDRj (j = 0, 1, 4, 5, 6, 8, 9, 16, 17, 18, 20, 21, 22)). If FIFO is used, the A/D conversion result is also stored to the FIFO data register (ADFIFODRk (k = 0 to 8)).

Table 53.11 Example of scanning operation in Oversampling mode – One-channel continuous scan mode (2 of 2)

Step	Operation
3	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of one analog channel is completed.
4	A/D conversion of the same virtual channel continues until trigger input is made. When the trigger input that is set as the start trigger of the startup scan group is performed, the A/D conversion of the next virtual channel starts.

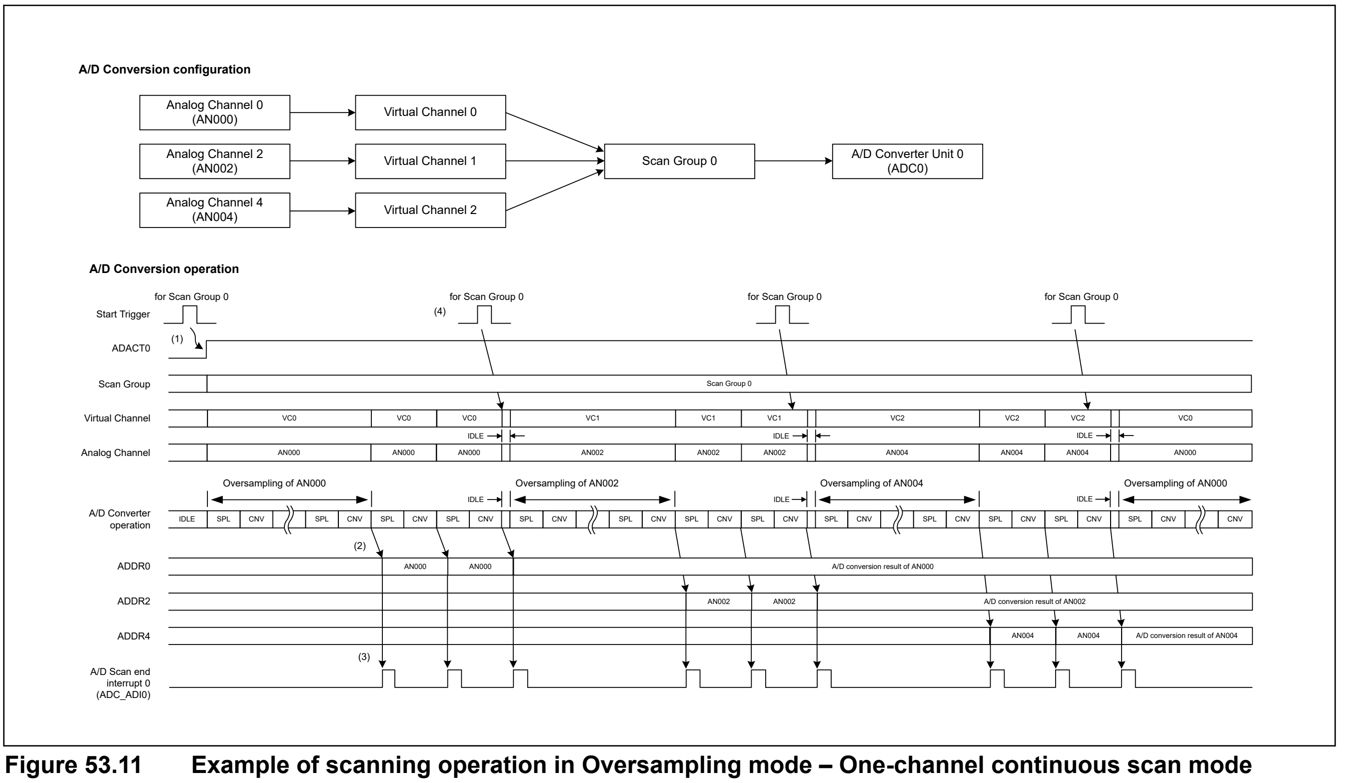


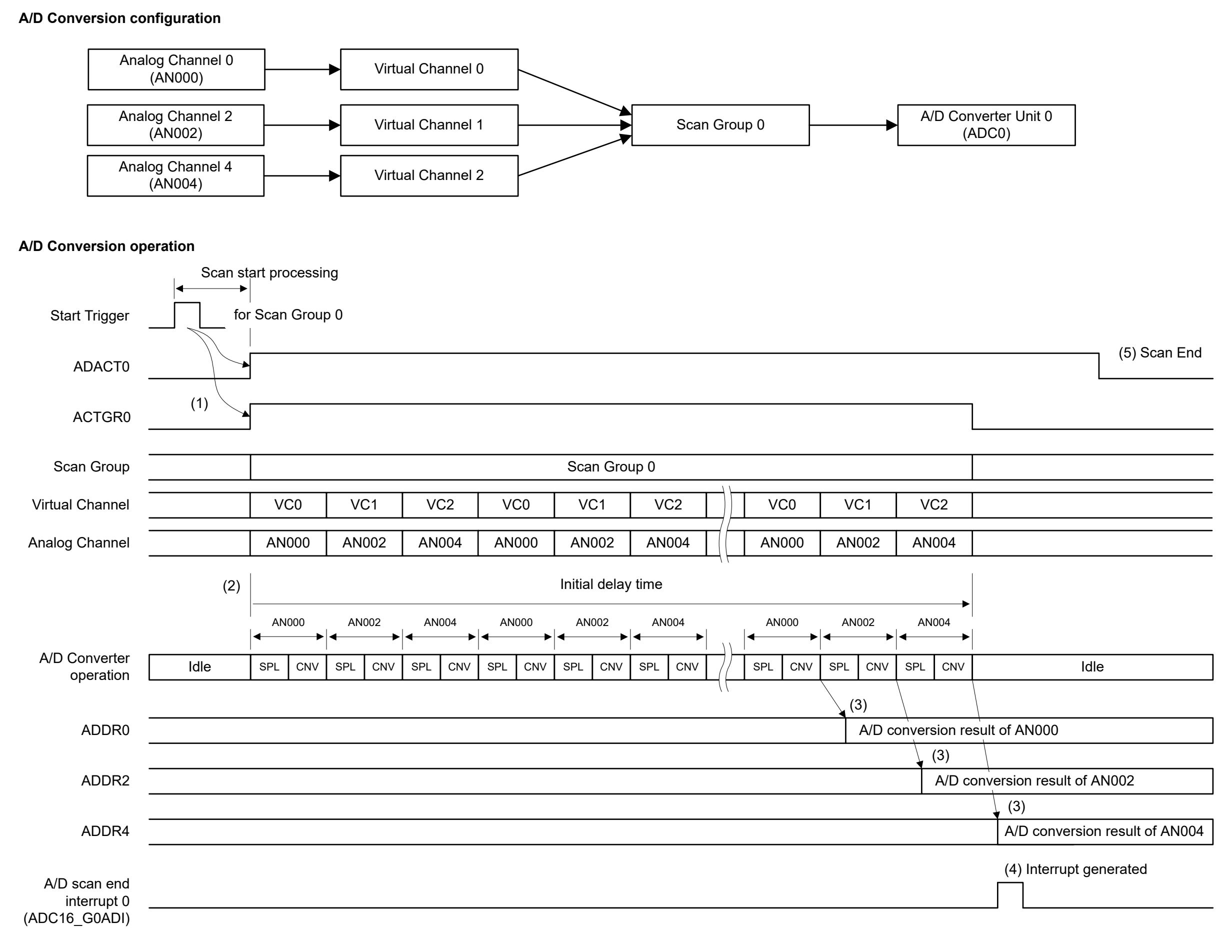
Figure 53.11 Example of scanning operation in Oversampling mode – One-channel continuous scan mode

53.3.7.6 Hybrid Mode – Single Scan Mode

Table 53.12 and Figure 53.12 show the operation example in Hybrid mode – Single scan mode.

Table 53.12 Example of the scanning operation in Hybrid mode – Single scan mode

Step	Operation
1	When a software trigger or a trigger from a peripheral module is input, the scanning operation of the scan group corresponding to the trigger is started.
2	In Hybrid mode, the scanning operation is performed while switching analog channels every time oversampling is performed.
3	After the oversampling corresponding to the number of TAPs of the digital filter and the number of A/D-converted value addition/averaging times is performed to each analog channel, A/D conversion data is output. (The time taken for the oversampling required to obtain the first A/D conversion data is called the initial delay time.) The A/D conversion data is stored in the data register (ADDRi (i = 0 to 22), ADEXDRj (j = 0, 1, 4, 5, 6, 8, 9, 16, 17, 18, 20, 21, 22)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 8)).
4	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
5	During the scanning operation, ADGRSR.ACTGRn (n = 0 to 8) bit corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0, 1) bit corresponding to the A/D converter performing the A/D conversion is also set to 1. When scanning operation is complete, each bit is cleared to 0 and A/D converter enters the idle state. And the oversampled data in the digital filter are also discarded at the end of the scanning operation.



Note: VC: Virtual Channel
SPL: Sampling state
CNV: Conversion state

Figure 53.12 Example of scanning operation in Hybrid mode – Single scan mode

53.3.7.7 Hybrid Mode – Continuous Scan Mode

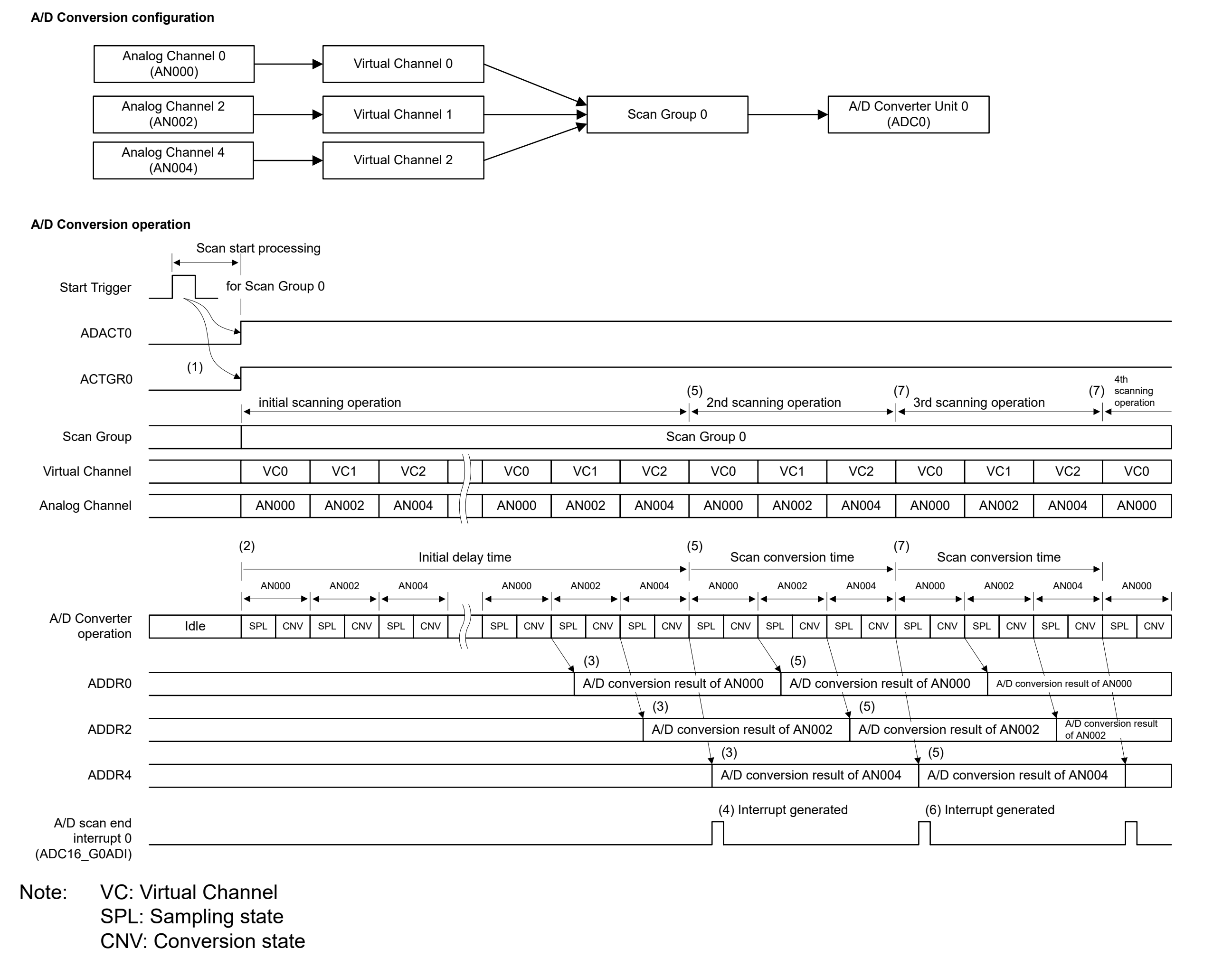
Table 53.13 and Figure 53.13 show the operation example in Hybrid mode – Continuous scan mode.

Table 53.13 Example of the scanning operation in Hybrid mode – Continuous scan mode (1 of 2)

Step	Operation
1	When a software trigger or a trigger from a peripheral module is input, the scanning operation of the scan group corresponding to the trigger is started. When scanning operation starts, ADGRSR.ACTGRn (n = 0 to 8) corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0, 1) bit corresponding to the A/D converter performing the A/D conversion is also set to 1.
2	In Hybrid mode, the scanning operation is performed while switching analog channels every time oversampling is performed.
3	After the oversampling corresponding to the number of TAPs of the digital filter and the number of A/D-converted value addition/averaging times is performed to each analog channel, A/D conversion data is output. (The time taken for the oversampling required to obtain the first A/D conversion data is called the initial delay time.) The A/D conversion data is stored in the data register (ADDRi (i = 0 to 22), ADEXDRj (j = 0, 1, 4, 5, 6, 8, 9, 16, 17, 18, 20, 21, 22)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 8)).
4	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.

Table 53.13 Example of the scanning operation in Hybrid mode – Continuous scan mode (2 of 2)

Step	Operation
5	In scanning operation after the initial delay time has elapsed, each time oversampling is performed for an analog channel, the data in the digital filter is updated. The updated A/D conversion data for each analog channel can be output each one time oversampling or each multiple times oversampling corresponding to the number of A/D-converted value addition/averaging times is performed. The A/D conversion data is stored in the data register (ADDRi (i = 0 to 22), ADEXDRj (j = 0, 1, 4, 5, 6, 8, 9, 16, 17, 18, 20, 21, 22)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 8)).
6	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
7	Thereafter, until the A/D conversion stop process is performed, Step 5 to 6 are repeated, and the scanning operation continues. To stop the A/D conversion, you must follow to section 53.6.4. Force Stops the A/D Conversion Operation.



Note: VC: Virtual Channel
SPL: Sampling state
CNV: Conversion state

Figure 53.13 Example of scanning operation in Hybrid mode – Continuous scan mode

53.3.7.8 Hybrid Mode – Background Continuous Scan Mode

Table 53.14 and Figure 53.14 show the operation example in Hybrid mode – Background continuous scan mode.

Table 53.14 Example of the scanning operation in Hybrid mode – Background continuous scan mode

Step	Operation
1	When a software trigger or a trigger from a peripheral module is input, the scanning operation of the scan group corresponding to the trigger is started. When scanning operation starts, ADGRSR.ACTGRn (n = 0 to 8) corresponding to that scan group is set to 1. ADSR.ADACTm (m = 0, 1) bit corresponding to the A/D converter performing the A/D conversion is also set to 1.
2	In Hybrid mode, the scanning operation is performed while switching analog channels every time oversampling is performed.
3	After the oversampling corresponding to the number of TAPs of the digital filter and the number of A/D-converted value addition/averaging times is performed to each analog channel, A/D conversion data becomes in ready to be output. (The time taken for the oversampling required to obtain the first A/D conversion data is called the initial delay time. In the background continuous scan operation, A/D conversion data can be obtained after the initial delay time has elapsed from the start of the scanning operation.)
4	In scanning operation after the initial delay time has elapsed, each time oversampling is performed for an analog channel, the data in the digital filter is updated. The updated A/D conversion data for each analog channel can be output each one time oversampling or each multiple times oversampling corresponding to the number of A/D-converted value addition/averaging times is performed.
5	When an A/D conversion start trigger is input during background continuous scanning operation, the most recent A/D conversion data at that time is stored in the data register (ADDRi (i = 0 to 22), ADEXDRj (j = 0, 1, 4, 5, 6, 8, 9, 16, 17, 18, 20, 21, 22)). If FIFO function is set to enabled, A/D conversion data is also stored in the FIFO data register (ADFIFODRk (k = 0 to 8)).
6	If scan end interrupt is set to enable, scan end interrupt is generated.
7	Thereafter, until the A/D conversion stop process is performed, Background continuous scanning operation (Step 4) is repeated. And whenever the A/D conversion start trigger is input during Background continuous scanning operation, the A/D conversion data is output (Step 5 and Step 6). To stop A/D conversion, you must follow to section 53.6.4. Force Stops the A/D Conversion Operation.

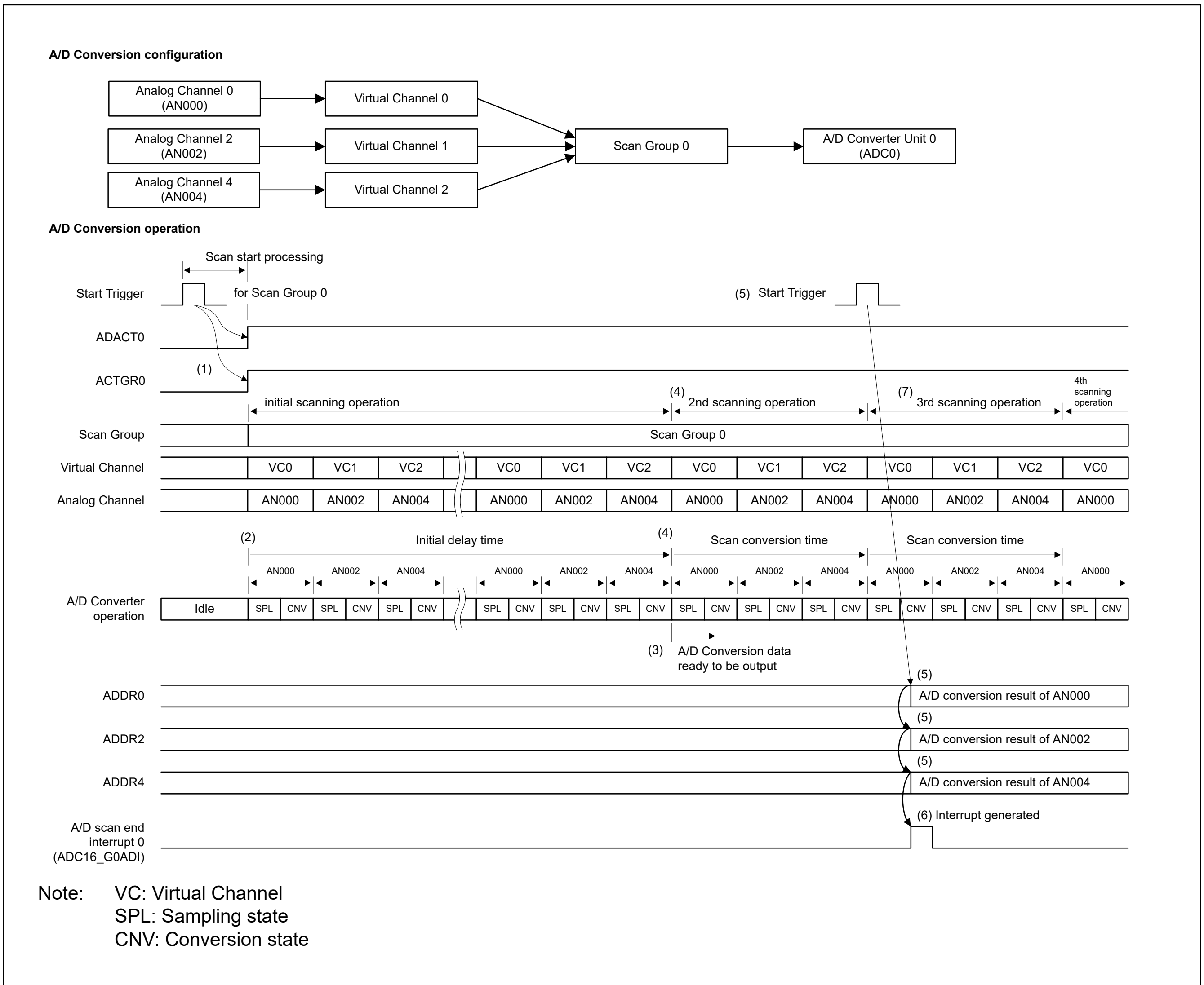


Figure 53.14 Example of scanning operation in Hybrid mode – Background continuous scan mode

53.3.7.9 Hybrid Mode – Fixed-channel Continuous Scan Mode

Table 53.15 and Figure 53.15 show the operation example in Hybrid mode – Fixed-channel continuous scan mode.

Table 53.15 Example of the scanning operation in Hybrid mode – Fixed-channel continuous scan mode (1 of 2)

Step	Operation
1	When a start trigger is input and the ADSR.ADACTm (m = 0, 1) bit becomes 1 (A/D conversion in progress), A/D conversion of the scan group selected by the ADSGCRm (m = 0, 1) register is started.
2	First, the fixed channels are converted for the number of channels set in the ADSWNRm.SWFIXm (m = 0, 1) bit.*1
3	After that, non-fixed channels are converted in ascending order of virtual channel numbers for the number of channels set in the ADSWNRm.SWNOFIXm (m = 0, 1) bits. The waveform example in fig uses 3 virtual channels and the conversion order is virtual channel 20 → virtual channel 21 → virtual channel 22.
4	When A/D conversion for one channel is completed, the A/D conversion result is stored in the A/D data register (ADDRi.DATA (i = 0 to 22), ADEXDRj (j = 0, 1, 4, 5, 6, 8, 9, 16, 17, 18, 20, 21, 22)) corresponding to the conversion target channel. If FIFO is used, the A/D conversion result is also stored to the FIFO data register (ADFIFODRk (k = 0 to 8)).

Table 53.15 Example of the scanning operation in Hybrid mode – Fixed-channel continuous scan mode (2 of 2)

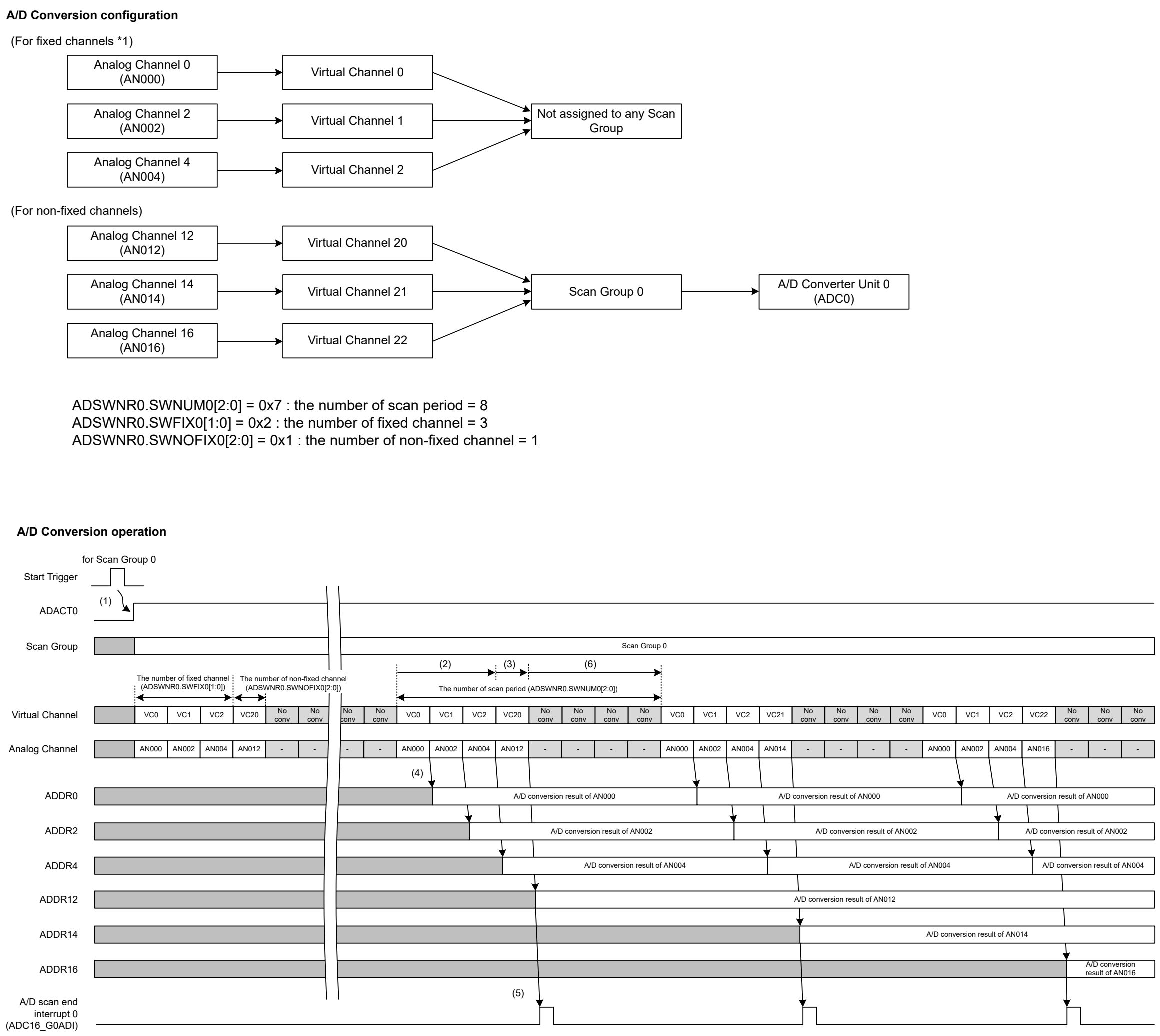
Step	Operation
5	When the A/D data register is updated during the scan cycle, a scan group n interrupt request is generated if the ADINTCR.ADIEn (n = 0 to 8) bit of each scan group in Scan End Interrupt Enable Register is set to 1 (enable scan group n interrupt by end of scan) after conversion of the asynchronous channel ends.
6	After the conversion of fixed channels and non-fixed channels is completed, the conversion time of idle cycles set in the ADSWNRm.BCSSTSLm (m = 0, 1) registers is counted until the end of the scan cycle set in ADSWNRm.SWNUMm (m = 0, 1).
7	After that, the A/D conversion operation, the result stored in the A/D conversion result register, the scan group n interrupt caused by the end of conversion of the non-fixed channel, and the count until the end of the scan cycle are repeated. Fixed channels repeat A/D conversion of the same channel, and non-fixed channels repeat A/D conversion up to the last channel assigned to the scan group, then return to the first channel.*2

Note: It is necessary to set so that the same filter is not selected between channels.

Note: When A/D conversion stop is set (ADSTOPR.ADSTOPm (m = 0, 1)), scan ends.

Note 1. Fixed channels in Fixed-channel continuous scan mode use the scan group and virtual channels determined for each unit. For details, see .

Note 2. If the scan period is set to less than the total number of conversions for fixed and non-fixed channels, conversion cycles beyond the scan period are not subject to A/D conversion.



Note 1. Fixed channels in Fixed channel continuous scan mode use the scan group and virtual channels determined for each unit. For details, see .

Figure 53.15 Example of scanning operation in Hybrid mode – Fixed-channel continuous scan mode

53.3.8 Self-calibration

ADC16H has a built-in self-calibration function. Self-calibration function is used to calibrate the variations of characteristics caused by the chip-by-chip variations. Self-calibration function has the following operations.

1. Internal Circuit Calibration
   - Adjusts the operation of A/D converter's internal circuitry.
2. Gain and Offset Calibration
   - Measure the A/D converter's Gain Error and Offset Error.
   - The calibration processing of the A/D conversion result based on the measured error data is performed after the A/D conversion.
     For details, see section 53.5.3.1. Gain Error and Offset Error Calibration.
   - This self-calibration should be performed after Internal-circuit self-calibration is completed.
3. Channel-dedicated sample-and-hold circuit Gain and Offset Calibration
   - Measure the Gain Error and the Offset Error when using the channel-dedicated sample-and-hold circuit.

- The calibration processing of the A/D conversion result based on the measured error data is performed after the A/D conversion.
  For details, see section 53.5.3.1. Gain Error and Offset Error Calibration.
- This self-calibration should be performed after A/D converter Gain/Offset self-calibration is completed.

53.3.8.1 Conditions Under Which Self-calibration is Required

The conditions under which self-calibration is required are shown in Table 53.16. Under the conditions shown in Table 53.16, self-calibration should be performed before using the A/D converter. If A/D converter is operating, stop all A/D converters and perform the self-calibration again. If the required self-calibration is not performed, the A/D conversion result is not guaranteed.

Table 53.16 List of conditions under which Self-calibration should be performed

Conditions under which self-calibration is required	Internal-circuit Calibration	Gain/Offset Calibration	Channel-dedicated sample-and-hold circuit Gain and Offset Calibration*1
After reset release	✓	✓	✓
After releasing the module stop	✓	✓	✓
When returning from Software Standby mode or Deep Software Standby mode	✓	✓	✓
When changed the ADCLK setting (When clock source or frequency is changed)	✓	✓	✓
When changed the operation mode or the scan mode of the A/D converter (When changed ADMDR.ADMDm bit (m = 0, 1))	✓	✓*2	✓
When changed A/D successive approximation time (When changed ADCNVSTR.CSTm bit (m = 0, 1))	✓	✓	✓
When changed the operation setting of the channel-dedicated sample-and-hold circuit *1	—	—	✓

Note: ✓: Self-calibration should be performed.
—: Self-calibration is not required.

Note 1. When any of ADSHCRm.SHENn bits are set to 1, or any of ADSHCRm.SHENn bits or any of ADSHCRm.SHMDn bits are changed, perform the channel-dedicated sample-and-hold circuits self-calibration (m = 0, 1, n = 0 to 2, 4 to 6). This is not required if the channel-dedicated sample-and-hold circuits are not used.

Note 2. When using fixed-channel continuous scan mode, also perform gain/offset calibration for SAR mode (single scan mode). The interval between performing calibration in SAR mode (single scan mode) and Hybrid mode should be as short as possible.

53.3.8.2 Self-calibration Procedure

Table 53.17 shows the self-calibration procedures.

Table 53.17 Procedure for self-calibration (1 of 2)

No.	Step	Description
1	Disable trigger input	Disables the trigger input from the peripheral module. (Write ADTRGENR.STTRGENn = 0)
2	Waiting for all A/D converters to stop	Check that all A/D converters are stopped. If A/D conversion is operating, wait until the A/D converter stops by either of the following: ● Wait until all A/D conversions are complete ● Force the A/D converter to stop the operation using the ADSTOPR register. For details on forced stopping of A/D conversion operation, see section 53.6.4. Force Stops the A/D Conversion Operation.
3	Setting the number of states for self-calibration	Set the number of states for the self-calibration of A/D converter to ADCALSTCR register. The values in ADCALSTCR register are commonly used for ADC0 and ADC1. Set the value to be set in ADCALSTCR register to satisfy the range defined in section 62, Electrical Characteristics.

Table 53.17 Procedure for self-calibration (2 of 2)

No.	Step	Description
4	Clearing the Error Status Flag	If error flags are detected, clear them before performing the self-calibration. (A/D converter error flag, A/D converter overflow flag, etc.)
5	Performing the self-calibration for ADC0	Performs the self-calibration for ADC0. ● Internal-circuit Calibration ● Gain/Offset Calibration After starting the self-calibration, wait until it is completed.
6	Performing the self-calibration for ADC1	Performs the self-calibration for ADC1 in the same as Step 5.
7	Setting the self-calibration of channel-dedicated sample-and-hold circuit *1	Make settings for self-calibration of channel-dedicated sample-and-hold circuit. Enable channel-dedicated sample-and-hold circuit. Sets the number of states for channel-dedicated sample-and-hold self-calibration (ADCALSTCR register *2 and ADCALSHCR register).
8	Performing channel-dedicated sample-and-hold circuit self-calibration for SH0 to SH2. *1	Performs the self-calibration for channel-dedicated sample-and-hold circuit (SH0 to SH2) connected to ADC0. ● Channel-dedicated sample-and-hold circuit Gain/Offset Calibration After starting self-calibration, wait until it is completed.
9	Performing channel-dedicated sample-and-hold circuit self-calibration for SH4 to SH6 *1	Performs the self-calibration for channel-dedicated sample-and-hold circuit (SH4 to SH6) connected to ADC1 in the same as Step 8.
10	Error Status Check *3	Check for errors in self-calibration. If no operating error occurs, self-calibration is complete.

Note: In order to increase the effectiveness of self-calibration, it is recommended to perform self-calibration for the A/D converters one by one.
Note 1. Not required if the channel-dedicated sample-and-hold circuits are not used.
Note 2. When the self-calibration is performed for the channel-dedicated sample-and-hold circuitry, the number of sampling states at the self-calibration of the A/D converter (ADCALSTCR.CALADSST[9:0] bits) may also need to be changed. Set the value that satisfies the electrical characteristics.
Note 3. If the A/D converter error is detected by the self-calibration, it may be set outside the operation guaranteed range specified in the electric characteristics. Check the operation setting.

53.3.8.3 Restrictions on Self-calibration

(1) Prohibition of the scanning operation during self-calibration

Do not start the scanning operation during self-calibration. When performing the scanning operation, start the scanning operation after self-calibration is completed. Operation is not guaranteed when the scanning operation is started during self-calibration.

(2) Prohibition of additional writes to the ADCALSTR register during self-calibration

After the self-calibration is started, writing to ADCALSTR register is prohibited until self-calibration is completed. Write to ADCALSTR register after self-calibration is completed. Operation is not guaranteed if this restriction is violated.

(3) Restrictions on self-calibration for channel-dedicated sample-and-hold circuit

To perform self-calibration for the channel-dedicated sample-and-hold circuit, set ADSHCRm.SHENn = 1 (m = 0, 1, n = 0 to 2, 4 to 6) for the sample-and-hold circuit channel to be used before performing self-calibration for the channel-dedicated sample-and-hold circuit.

(4) Restrictions on self-calibration

Self-calibration should be performed for the A/D converters one by one. During the self-calibration operation, the other A/D converters should be in idle (not in scan operation or self-calibration operation).

If this restriction is violated, the A/D conversion accuracy will be degraded due to noise during the self-calibration operation. In this case, the A/D converter characteristics are not guaranteed.

In addition, in order to enhance the effect of self-calibration, self-calibration operation should be performed under conditions with as little noise as possible.

(5) Restrictions on setting of the number of status for self-calibration

The number of states to be set in the ADCALSTCR and ADCALSHCR registers should be set to satisfy the values specified in the Electrical Characteristics. In addition, the number of states involved in self-calibration should be set to satisfy the following restrictions.

If these restrictions are violated, A/D conversion results are not guaranteed.

[Restrictions on setting of ADCALSTCR register]

- ADCALSTCR.CALADSST[9:0] bits
  - According to the type of self-calibration, set the values to satisfy those specified in the electrical characteristics.
  - If the setting value differs according to the type of self-calibration, change the register setting value for each self-calibration.
- ADCALSTCR.CALADCST[5:0] bits
  - Set the same value as ADCNVSTR.CSTm[5:0] (m = 0, 1) bits.
    (ADCALSTCR.CALADCST[5:0] = ADCNVSTR.CSTm[5:0])

[Restrictions on setting of ADCALSHCR register] (only when channel-dedicated sample-and-hold circuit is used)

- ADCALSHCR.CALSHSST[7:0] bits
  - Set the value equal to the value of the ADSHSTRm.SHSST[7:0] bits plus 1 (m = 0, 1).
    (ADCALSHCR.CALSHSST[7:0] = ADSHSTRm.SHSST[7:0] + 1)
  - If the setting values of ADSHSTR0.SHSST[7:0] and ADSHSTR1.SHSST[7:0] are different, change the register setting values of CALSHSST[7:0] for each self-calibration of ADC0 (SH0 to SH2) and ADC1 (SH4 to SH6).
- ADCALSHCR.CALSHHST[2:0] bits
  - Set the same value as ADSHSTRm.SHHST[2:0] (m = 0, 1) bits.

53.3.9 Analog Input Channel

For the correspondence between the analog input pins and the analog channel numbers, the A/D converters, and channel-dedicated sample-and-hold circuit, see Table 53.4.

53.3.10 Extended Analog Function

Extended analog function performs A/D conversion on the internal signal source of MCU. The available sources for extended analog function are as follows:

- Self-diagnosis voltage for A/D converters and channel-dedicated sample-and-hold circuits
- Internal reference voltage
- VBATT 1/6 voltage monitor output
- Temperature sensor
- D/A converter output (internal output path to ACMPHS)

53.3.11 Self-diagnosis Function for A/D Converters

The A/D converter has a built-in self-diagnosis function. The self-diagnosis function inputs the self-diagnosis voltage to the A/D converter and performs A/D conversion. The A/D conversion result of the self-diagnosis voltage can be used to confirm that the A/D converter is operating normally.

53.3.11.1 Self-diagnosis

The A/D converter has a built-in self-diagnosis function. The self-diagnosis function inputs the self-diagnosis voltage to the A/D converter and performs A/D conversion. The A/D conversion result of the self-diagnosis voltage can be used to confirm that the A/D converter is operating normally. Table 53.18 shows the operation mode in which the self-diagnostic function is available.

Table 53.18 List of operation mode for which the self-diagnostic function is available

Operation mode – Scan mode	Self-diagnosis function
SAR mode – Single scan mode	✓
SAR mode – Continuous scan mode	✓
Oversampling mode – Single scan mode	✓
Oversampling mode – Continuous scan mode	—
Oversampling mode – One-channel continuous scan mode	—
Hybrid mode – Single scan mode	✓
Hybrid mode – Continuous scan mode	—
Hybrid mode – Background continuous scan mode	—
Hybrid mode – Fixed-channel continuous scan mode	—

Note: ✓: Available, —: Not Available

The procedure for performing self-diagnosis of A/D converter is shown in the following.

1. Assign the self-diagnosis channel to one of the virtual channels and configure the virtual channel settings.
2. Assign the virtual channel that the self-diagnosis channel is selected to one of the scan groups.
3. Set the self-diagnosis voltage in DIAGVAL[2:0] bits of ADSGDCRn (n = 0 to 8) register corresponding to the scan group assigned in 2 above.
4. Start the scanning operation by inputting the A/D conversion start trigger. The self-diagnosis operation is executed at the A/D conversion of the self-diagnosis channel.

Table 53.19 lists the operation settings when the self-diagnosis function is used. When using the self-diagnosis function, follow the setting in Table 53.19. If self-diagnosis operation is performed with settings other than those shown in Table 53.19, the self-diagnosis result (A/D conversion result) is not guaranteed.

Table 53.19 List of operation setting and expected value for ADC0 self-diagnosis function

Self-diagnosis mode	Register setting value				Self-diagnosis operation
	ADSGDCRx	ADCHCRy		ADDOPCRCy	Expected value of A/D conversion data *1 *2 *3
	DIAGVAL[2:0]	CNVCS[6:0]	AINMD	SIGNSEL	
Self-diagnosis mode 1	100b	0x60	1	0	0x0000
Self-diagnosis mode 2	101b	0x60	1	0	0x8000
Self-diagnosis mode 3	110b	0x60	1	0	0x7FFF

Note: x = 0 to 8, y = 0 to 32
Note 1. The expected value is the ideal A/D converter result stored in ADEXDR0.DATA[15:0] or ADFIFODRx.DATA[15:0].
Note 2. The A/D conversion data is the value when 16-bit data format (ADDOPCRCy.ADPRC[1:0] = 00b) is selected.
Note 3. The A/D conversion result becomes (expected value + 1) or more when a positive accuracy error occurs in A/D conversion, and becomes (expected value - 1) or less when a negative accuracy error occurs.

Table 53.20 List of operation setting and expected value for ADC1 self-diagnosis function

Self-diagnosis mode	Register setting value				Self-diagnosis operation
	ADSGDCRx	ADCHCRy		ADDOPCRCy	Expected value of A/D conversion data *1 *3
	DIAGVAL[2:0]	CNVCS[6:0]	AINMD	SIGNSEL	
Self-diagnosis mode 1	100b	0x61	1	0	0x0000
Self-diagnosis mode 2	101b	0x61	1	0	0x8000
Self-diagnosis mode 3	110b	0x61	1	0	0x7FFF

Note: x = 0 to 8, y = 0 to 32
Note 1. The expected value is the ideal A/D converter result stored in ADEXDR1.DATA[15:0] or ADFIFODRx.DATA[15:0].
Note 2. The A/D conversion data is the value when 16-bit data format (ADDOPCRCy.ADPRC[1:0] = 00b) is selected.

Note 3. The A/D conversion result becomes (expected value + 1) or more when a positive accuracy error occurs in A/D conversion, and becomes (expected value - 1) or less when a negative accuracy error occurs.

Table 53.21 List of operation setting and expected value for ADC0 channel-dedicated sample-and-hold circuit self-diagnosis function

Self-diagnosis mode	Register setting value					Self-diagnosis operation
	ADSHSDCRw	ADCHCRy		ADSHCRw	ADDOPCRCy	Expected value of A/D conversion data *1 *2 *3
	SHDIAGx[2:0]	CNVCS[6:0]	AINMD	SHMDz	SIGNSEL	
S&H self-diagnosis mode 1	001b	SH0: 0x70 SH1: 0x71 SH2: 0x72	1	1	0	0x0000
S&H self-diagnosis mode 2	010b	SH0: 0x70 SH1: 0x71 SH2: 0x72	1	1	0	0x8000
S&H self-diagnosis mode 3	011b	SH0: 0x70 SH1: 0x71 SH2: 0x72	1	1	0	0x7FFF
S&H self-diagnosis mode 4	100b	SH0: 0x70 SH1: 0x71 SH2: 0x72	1	1	0	0x0000
S&H self-diagnosis mode 5	101b	SH0: 0x70 SH1: 0x71 SH2: 0x72	1	1	0	0x8000
S&H self-diagnosis mode 6	110b	SH0: 0x70 SH1: 0x71 SH2: 0x72	1	1	0	0x7FFF

Note: x = 0 to 8, y = 0 to 32, z = 0 to 2
Note 1. The expected value is the ideal A/D converter result stored in ADEXDRn.DATA[15:0] (n = 16,17,18) or ADFIFODRx.DATA[15:0].
Note 2. The A/D conversion data is the value when 16-bit data format (ADDOPCRCy.ADPRC[1:0] = 00b) is selected.
Note 3. The A/D conversion result becomes (expected value + 1) or more when a positive accuracy error occurs in A/D conversion, and becomes (expected value - 1) or less when a negative accuracy error occurs.

Table 53.22 List of operation setting and expected value for ADC1 channel-dedicated sample-and-hold circuit self-diagnosis function (1 of 2)

Self-diagnosis mode	Register setting value					Self-diagnosis operation
	ADSHSDCRw	ADCHCRy		ADSHCRw	ADDOPCRCy	Expected value of A/D conversion data *1 *2 *3
	SHDIAGx[2:0]	CNVCS[6:0]	AINMD	SHMDz	SIGNSEL	
S&H self-diagnosis mode 1	001b	SH4: 0x74 SH5: 0x75 SH6: 0x76	1	1	0	0x0000
S&H self-diagnosis mode 2	010b	SH4: 0x74 SH5: 0x75 SH6: 0x76	1	1	0	0x8000
S&H self-diagnosis mode 3	011b	SH4: 0x74 SH5: 0x75 SH6: 0x76	1	1	0	0x7FFF
S&H self-diagnosis mode 4	100b	SH4: 0x74 SH5: 0x75 SH6: 0x76	1	1	0	0x0000
S&H self-diagnosis mode 5	101b	SH4: 0x74 SH5: 0x75 SH6: 0x76	1	1	0	0x8000

Table 53.22 List of operation setting and expected value for ADC1 channel-dedicated sample-and-hold circuit self-diagnosis function (2 of 2)

Self-diagnosis mode	Register setting value					Self-diagnosis operation
	ADSHSDCRw	ADCHCRy		ADSHCRw	ADDOPCRCy	Expected value of A/D conversion data *1 *2 *3
	SHDIAGx[2:0]	CNVCS[6:0]	AINMD	SHMDz	SIGNSEL	
S&H self-diagnosis mode 6	110b	SH4: 0x74 SH5: 0x75 SH6: 0x76	1	1	0	0x7FFF

Note: x = 0 to 8, y = 0 to 32, z = 4 to 6
Note 1. The expected value is the ideal A/D converter result stored in ADEXDRn.DATA[15:0] (n = 20,21, 22) or ADFIFODRx.DATA[15:0].
Note 2. The A/D conversion data is the value when 16-bit data format (ADDOPCRCy.ADPRC[1:0] = 00b) is selected.
Note 3. The A/D conversion result becomes (expected value + 1) or more when a positive accuracy error occurs in A/D conversion, and becomes (expected value - 1) or less when a negative accuracy error occurs.

53.3.11.2 Self-diagnosis Mode

(1) Self-diagnosis mode 1

In self-diagnosis mode 1, the non-inverting input (+) (A_{INP}) and inverting input (−) (A_{INN}) of A/D converter are input VREFL0 level (ADC0) or VREFL level (ADC1). Since the differential voltage is 0, the ideal value of the self-diagnosis result is 0x0000 (±0). The A/D conversion result becomes 0x0001 (+1) or more when a positive accuracy error occurs in A/D conversion, and becomes 0xFFFF (−1) or less when a negative accuracy error occurs.

(2) Self-diagnosis mode 2

In self-diagnosis mode 2, the non-inverting input (+) (A_{INP}) of A/D converter is input VREFL0 level (ADC0) or VREFL level (ADC1), and the inverting input (−) (A_{INN}) of A/D converter is input VREFH0 level (ADC0) or VREFH level (ADC1). Since the differential voltage is −VREFH0 (if VREFL0 = 0V) or −VREFH (if VREFL = 0V), the ideal value of the self-diagnosis result is 0x8000 (−32768). When an A/D conversion error occurs, the A/D conversion result of self-diagnosis is greater than 0x8000.

(3) Self-diagnosis mode 3

In self-diagnosis mode 3, the non-inverting input (+) (A_{INP}) of A/D converter is input VREFH0 level (ADC0) or VREFH level (ADC1), and the inverting input (−) (A_{INN}) of A/D converter is input VREFL0 level (ADC0) or VREFL level (ADC1). Since the differential voltage is +VREFH0 (if VREFL0 = 0V) or +VREFH (if VREFL = 0V), the ideal value of the self-diagnosis result is 0x7FFF (+32767). When an A/D conversion error occurs, the A/D conversion result of self-diagnosis is less than 0x7FFF.

53.3.11.3 Notes on Self-diagnosis Function

(1) Notes on the results of the self-diagnosis

The self-diagnostic result of A/D converter unit 0 can be read from the ADEXDR0 register. The self-diagnostic result of A/D converter unit 1 can be read from the ADEXDR1 register.

(2) Notes on data format during self-diagnosis

When self-diagnosis is performed, if a data format other than 16-bit length is set, an overflow occurs due to A/D conversion data rounding, and the A/D conversion data error is detected. To perform self-diagnosis, select the 16-bit data format.

53.3.12 Internal Reference Voltage

ADC16H can perform A/D conversion of Internal Reference Voltage. Internal Reference Voltage outputs a constant voltage that is independent of the power supply voltage (VCC, VSS, AVCC, AVSS, VREFH0, VREFL0, VREFH and VREFL). The monitoring of Internal Reference Voltage can be used to detect variations in the analog reference supply voltage (VREFH0, VREFL0, VREFH and VREFL), or abnormalities or faults in the system.

When performing A/D conversion of Internal Reference Voltage, it is required to set the Internal Reference Voltage Monitor Enable Register (ADREFCR) and assign the analog channel of Internal Reference Voltage to any of the virtual channels. See Table 53.5 for the correspondence between Internal Reference Voltage function and the analog channels. For the sampling time when A/D conversion of Internal Reference Voltage is performed, see section 62, Electrical Characteristics.

53.3.13 Temperature Sensor

ADC16H can perform A/D conversion of Temperature Sensor. The chip temperature can be estimated from the A/D conversion result of Temperature Sensor. For more details on Temperature Sensor, see section 55, Temperature Sensor (TSN).

When A/D conversion of Temperature Sensor is performed, Temperature Sensor should be configured and the analog channel of Temperature Sensor should be assigned to any of the virtual channels. See Table 53.5 for the correspondence between Temperature Sensor and analog channels. For the sampling time when A/D conversion of Temperature Sensor is performed, see section 62, Electrical Characteristics.

53.3.14 D/A Converter

ADC16H can perform A/D conversion of the voltage that the D/A converter is outputting to the internal module (ACMPHS). When the output of the D/A converter is used for the reference input voltage of High-Speed Analog Comparator (ACMPHS), this function can be used to monitor the output voltage of the D/A converter. For details of the D/A converter function, see section 54, 12-Bit D/A Converter (DAC12).

When A/D conversion is performed on the output of D/A converter (DA0 and DA1) (output to ACMPHS), the D/A converter should be configured and the analog channel of the D/A converter should be assigned to one of the virtual channels. For the sampling time when A/D conversion is performed on the D/A converter output voltage, see section 62, Electrical Characteristics.

53.3.15 VBATT 1/6 Voltage Monitor Output

ADC16H can perform A/D conversion of VBATT 1/6 voltage monitor output. VBATT 1/6 voltage monitor output is a function that can monitor the input voltage level to the VBATT pin at 1/6 level. For more details on VBATT voltage monitor function, see section 12, Battery Backup Function.

When A/D conversion of VBATT 1/6 voltage monitor output is performed, VBATT should be configured and the analog channel of VBATT 1/6 voltage monitor output should be assigned to any of the virtual channels. See Table 53.5 for the correspondence between VBATT 1/6 voltage monitor output and analog channels. For the sampling time when A/D conversion of VBATT 1/6 voltage monitor output is performed, see section 62, Electrical Characteristics.

53.3.16 Channel-dedicated Sample-and-hold Circuit

ADC16H has a built-in channel-dedicated sample-and-hold circuit. By using multiple channel-dedicated sample-and-hold circuits, it is possible to sample multiple analog inputs simultaneously.

The conditions under which channel-dedicated sample-and-hold circuit can be used are shown in Table 53.23.

Table 53.23 Operation mode in which channel-dedicated sample-and-hold circuits can be used

Operation mode	Channel-dedicated sample-and hold circuits
SAR mode – Single scan mode	✓*1
SAR mode – Continuous scan mode	—
Oversampling mode – Single scan mode	—
Oversampling mode – Continuous scan mode	—
Oversampling mode – One-channel continuous scan mode	—
Hybrid mode – Single scan mode	✓
Hybrid mode – Continuous scan mode	✓
Hybrid mode – Background continuous scan mode	✓
Hybrid mode – Fixed-channel continuous scan mode	✓ *2

Note: ✓: available, —: not available

Note 1. Constant Sampling Function is available.
Note 2. Only fixed channels are supported when fixed channels are set to 3ch and all fixed channels enable channel-dedicated sample-and-hold circuit. Non-fixed channels cannot use channel-dedicated sample-and-hold circuit.

53.3.16.1 Configuration of Channel-dedicated Sample-and-hold Circuit

(1) Configuration in Single-ended input mode

This subsection describes the internal configuration when using channel-dedicated sample-and-hold circuit with single-ended inputs.

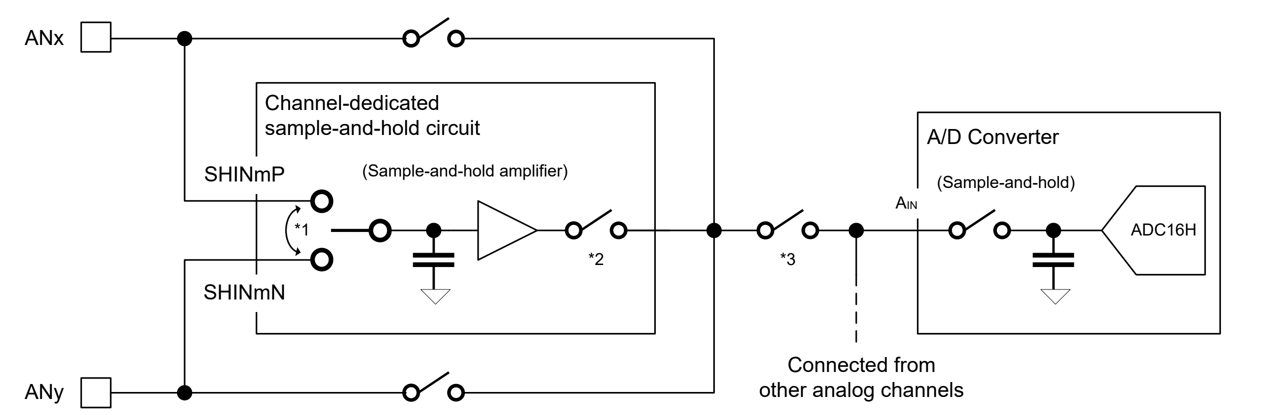
Figure 53.16 shows the configuration of channel-dedicated sample-and-hold circuit in Single-ended input mode.

Channel-dedicated sample-and-hold circuits are connected to two analog input channels.

In the case that one of the channels is used as an input for channel-dedicated sample-and-hold circuit and the other is assigned as an input for normal A/D conversion when the channel-dedicated sample-and-hold circuit is used in Single-ended input mode, one of the following methods can be used.

- Assign each input channel to a different scan group
- Use the function of ADSHSBPCR register for AD conversion in one scan group

For example, in the case of using the function of ADSHSBPCR register for AD conversion in one scan group, set 0 to ADSHSBPCR.SHSBP0 bit and set 1 to ADSHSBPCR.SHSBN0 bit when AN000 is used as an input channel of SH0 in Single-ended input mode and AN001 is used as an input channel of normal AD conversion.



Note m : The unit number of channel-dedicated sample-and-hold circuit
x : Even-numbered analog channel connected to the channel-dedicated sample-and-hold circuit
y : Odd-numbered analog channel connected to the channel-dedicated sample-and-hold circuit

Note 1. Turn on when the channel-dedicated sample-and-hold circuit is enabled and samples ANx or ANy channel.
Note 2. Turn on when the channel-dedicated sample-and-hold circuit is the holding state.
Note 3. Turn on when the A/D conversion (sampling state) of ANx, ANy or output of channel-dedicated sample-and-hold circuit.

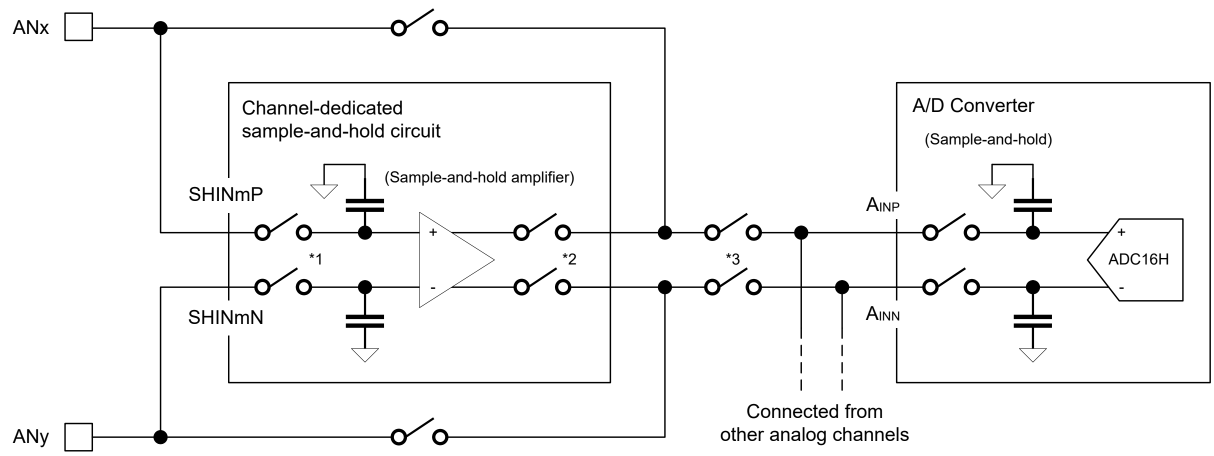
Figure 53.16 Internal configuration of channel-dedicated sample-and-hold circuit in Single-ended input mode

(2) Configuration in Differential input mode

This subsection describes the internal configuration when channel-dedicated sample-and-hold circuit is used with differential inputs.

Figure 53.17 shows the configuration of channel-dedicated sample-and-hold circuit in Differential input mode.

When channel-dedicated sample-and-hold circuit is operating in Differential input mode, the even analog channel is connected to the non-inverting input (+) (SHINmP) and the odd-numbered analog channel is connected to the inverting input (−) (SHINmN). Two analog inputs (differential input) are sampled and held in one scanning operation, and output to the A/D converter. The A/D conversion setting should be set to convert a differential input signal source. (ADCHCRn.AINMD = 1 (n = 0 to 32))



Note m : The unit number of channel-dedicated sample-and-hold circuit
x : Even-numbered analog channel connected to the channel-dedicated sample-and-hold circuit
y : Odd-numbered analog channel connected to the channel-dedicated sample-and-hold circuit

Note 1. Turn on when the channel-dedicated sample-and-hold circuit is enabled and samples ANx or ANy channel.
Note 2. Turn on when the channel-dedicated sample-and-hold circuit is the holding state.
Note 3. Turn on when the A/D conversion (sampling state) of ANx, ANy or output of channel-dedicated sample-and-hold circuit.

Figure 53.17 Internal configuration of channel-dedicated sample-and-hold circuit in Differential input mode

53.3.16.2 Operation in SAR Mode with Channel-dedicated Sample-and-hold Circuit

(1) Basic operation in SAR mode – Single scan mode (Single-ended input mode)

This section describes the basic operation of channel-dedicated sample-and-hold circuit in SAR mode. The configuration and behavior of example operations are shown in Table 53.24, Table 53.25 and Figure 53.18.

Table 53.24 Configuration of example operation for channel-dedicated sample-and-hold circuit in SAR mode – Single scan mode

Item	Configuration
A/D converter	Set ADC0 to SAR mode – Single scan mode
Channel-dedicated sample-and-hold circuit	Set SH0 to SH2 to enabled and Single-ended input mode. (AN000 to AN005 are subject to A/D conversion using channel-dedicated sample-and-hold circuit.)
Virtual channel*1	Assign AN000, AN002, AN004 and AN020 to virtual channel 0 to 3, and set to Single-ended input mode.
Scan group	Assign virtual channel 0 to 3 to scan group 0. Set scan group 0 to convert with ADC0.

Note 1. When channel-dedicated sample-and-hold circuit is used, there are restrictions on the setting of virtual channels. For more details, see section 53.3.16.6. Restrictions on Channel-dedicated Sample-and-hold Circuit.

Table 53.25 Example operation of channel-dedicated sample-and-hold circuit in SAR mode – Single scan mode (1 of 2)

Step	Detail behavior of example operation
1	The scanning operation of scan group 0 starts when a trigger for scan group 0 is input.
2	Channel-dedicated sample-and-hold circuit (SH0 to SH2) starts sampling the analog inputs (AN000, AN002, AN004) and holds them.

Table 53.25 Example operation of channel-dedicated sample-and-hold circuit in SAR mode – Single scan mode (2 of 2)

Step	Detail behavior of example operation
3	A/D conversion is performed on the voltages held by each channel-dedicated sample-and-hold circuit and on other analog input channels (AN020).
4	When A/D conversion of each channel completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
5	If scan end interrupt in scan group 0 is set to enabled, a scan end interrupt occurs.

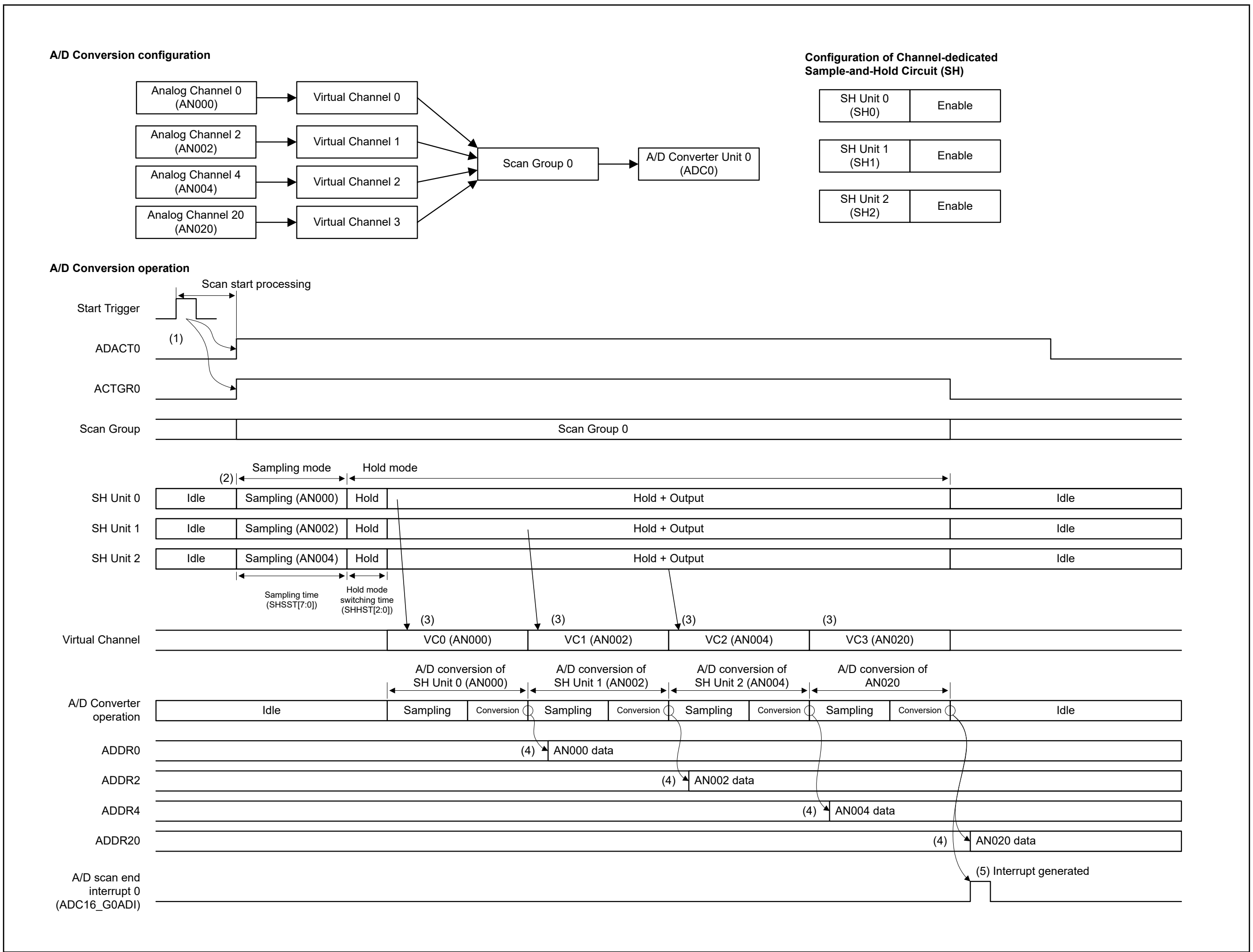


Figure 53.18 Basic operation of channel-dedicated sample-and-hold circuit in SAR mode

(2) Advanced operation in SAR mode - Single scan mode

This section describes the operation of A/D conversion that shares one channel-dedicated sample-and-hold circuit between two analog channels. The operation shown in this section is possible only when all of the following conditions are met.

[Operating Conditions]

- Operation mode: SAR mode – Single scan mode
- Input method: Single-ended input mode

The following shows an example of A/D conversions of even-numbered analog channels and odd-numbered analog channels using channel-dedicated sample-and-hold circuit in SAR mode – Single scan mode. The configuration example is shown in Table 53.26, and the operation detail is shown in Table 53.27 and Figure 53.19.

Table 53.26 Configuration example of advanced operation for channel-dedicated sample-and-hold circuit

Item	Configuration
A/D converter	Set ADC0 to SAR mode – Single scan mode
Channel-dedicated sample-and-hold circuit	Set SH0 to SH2 to enabled and Single-ended input mode. (AN000 to AN005 are subject to A/D conversion using channel-dedicated sample-and-hold circuit.)
Virtual channel*1	• Assign AN000, AN002 and AN004 to virtual channel 0 to 2, and set to single-ended input mode. • Assign AN001, AN003 and AN005 to virtual channel 3 to 5, and set to single-ended input mode.
Scan group	• Scan group 0: Assign AN000, AN002, AN004 (virtual channels 0 to 2). • Scan group 1: Assign AN001, AN003, AN005 (virtual channels 3 to 5). • Set scan group 0 and scan group 1 to convert with ADC0.

Note 1. When channel-dedicated sample-and-hold circuit is used, there are restrictions on the setting of virtual channels. For more details, see section 53.3.16.6. Restrictions on Channel-dedicated Sample-and-hold Circuit.

Table 53.27 Example advanced operation of channel-dedicated sample-and-hold circuit

Step	Detail behavior of example operation
1	The scanning operation of scan group 0 starts when a trigger for scan group 0 is input.
2	Channel-dedicated sample-and-hold circuit (SH0 to SH2) starts sampling the analog inputs (AN000, AN002, AN004) and holds them.
3	A/D conversion is performed on the voltages held by channel-dedicated sample-and-hold circuit.
4	When A/D conversion of each channel in scan group 0 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
5	If scan end interrupt in scan group 0 is set to enabled, a scan end interrupt occurs.
6	After scanning of scan group 0 is completed, the scanning operation of scan group 1 starts when a trigger for scan group 1 is input.
7	Channel-dedicated sample-and-hold circuit (SH0 to SH2) starts sampling the analog inputs (AN001, AN003, AN05) and holds them.
8	A/D conversion is performed on the voltages held by channel-dedicated sample-and-hold circuit.
9	When A/D conversion of each channel in scan group 1 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
10	If scan end interrupt for scan group 1 is set to enabled, a scan end interrupt occurs.

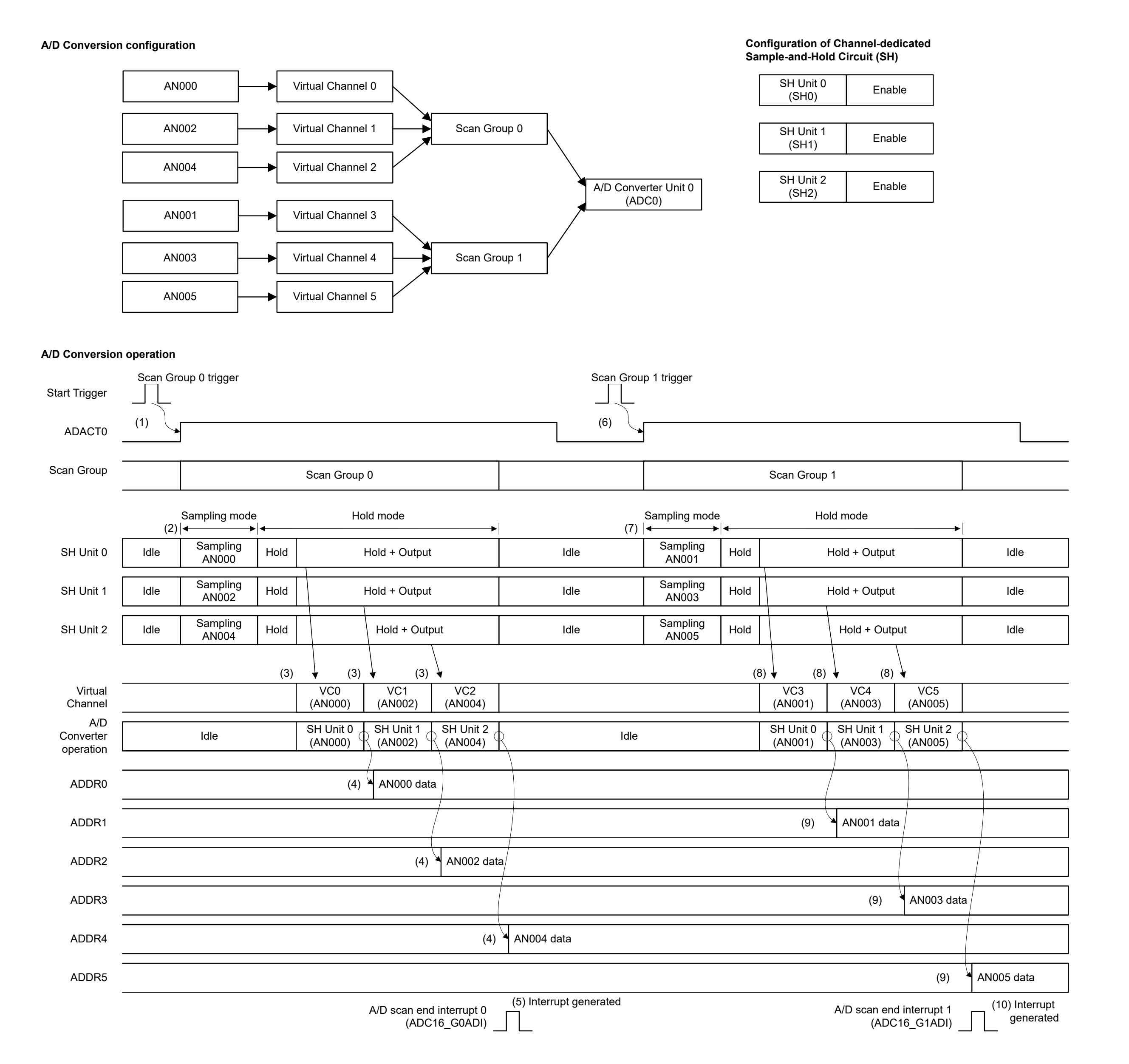


Figure 53.19 Example operation when channel-dedicated sample-and-hold circuit is shared with two analog channels in SAR mode

53.3.16.3 Operation in Hybrid Mode with Channel-dedicated Sample-and-hold Circuit

(1) Basic Operation in Hybrid Mode - Single Scan Mode

This section describes the basic operation of channel-dedicated sample-and-hold circuit in Hybrid mode – Single scan mode. Since Differential input mode is supported in Hybrid mode, this example describes a case that channel-dedicated sample-and-hold circuits are set to Differential input mode. The configuration and behavior of example operation are shown in Table 53.28, Table 53.29 and Figure 53.20.

Table 53.28 Configuration of example operation for channel-dedicated sample-and-hold circuit in Hybrid mode – Single scan mode (1 of 2)

Item	Configuration
A/D converter	Set ADC0 to Hybrid mode – Single scan mode
Channel-dedicated sample-and-hold circuit	Set SH0 to SH2 to enabled and Differential input mode. (AN000 to AN005 are subject to A/D conversion using channel-dedicated sample-and-hold circuit.)

Table 53.28 Configuration of example operation for channel-dedicated sample-and-hold circuit in Hybrid mode – Single scan mode (2 of 2)

Item	Configuration
Virtual channel*1	• Assign AN000, AN002 and AN004 to virtual channel 0 to 2, and set to Differential input mode. (This will function AN000, AN002 and AN004 as the non-inverting inputs (+) of the differential input, and AN001, AN003, and AN005 as the inverting inputs (−) of the differential input.)
Scan group*1	• Assign virtual channel 0 to 2 to scan group 0. • Set scan group 0 to convert with ADC0.

Note 1. When channel-dedicated sample-and-hold circuit is used, there are restrictions on the setting of virtual channels. For more details, see section 53.3.16.6. Restrictions on Channel-dedicated Sample-and-hold Circuit.

Table 53.29 Example operation of channel-dedicated sample-and-hold circuit in Hybrid mode – Single scan mode

Step	Behavior
1	The scanning operation of scan group 0 starts when a trigger for scan group 0 is input.
2	Channel-dedicated sample-and-hold circuit (SH0 to SH2) starts sampling the analog inputs (AN000, AN002, AN004) and holds them.
3	The A/D converter performs oversampling once for each output of channel-dedicated sample-and-hold circuit (SH0 to SH2).
4	After one oversampling for each analog channel, channel-dedicated sample-and-hold circuit releases the held voltage and enters the idle state.
5	Step 2 to 4 is repeated until oversampling corresponding to the number of TAPs of the digital filter and the number of A/D-converted value addition/average times is performed for each analog channel.
6	A/D conversion data corresponding to each analog channel is output. The A/D conversion data is stored in the data register (ADDRi (i = 0, 2, 4)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 8)).
7	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
8	A/D converter and channel-dedicated sample-and-hold circuit enter the idle state when scanning operation is completed. The oversampling data in digital filter are discarded at the end of the scanning operation.

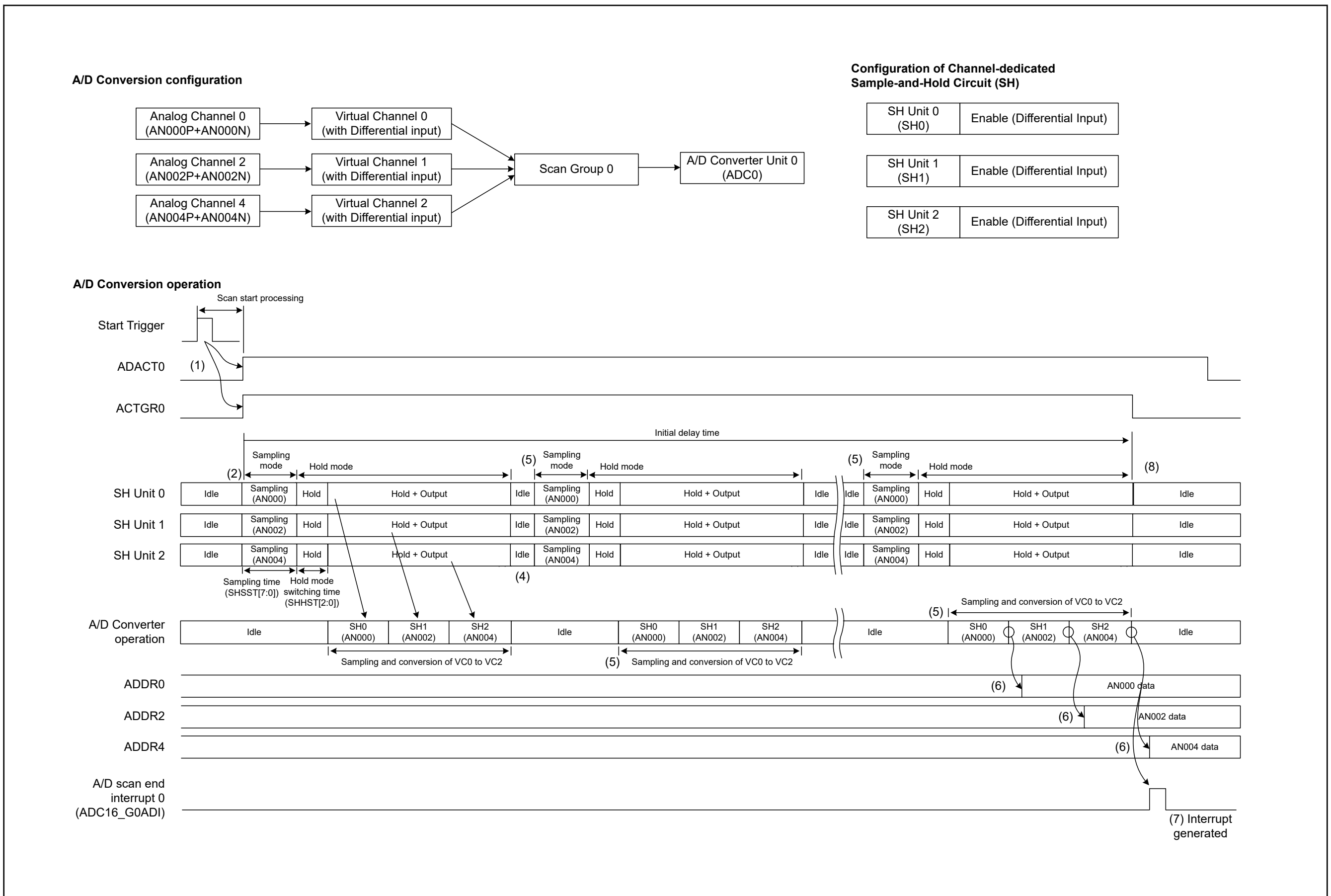


Figure 53.20 Basic operation of channel-dedicated sample-and-hold circuit in Hybrid mode – Single scan mode

(2) Basic Operation in Hybrid Mode - Continuous Scan Mode

This section shows the basic operation of channel-dedicated sample-and-hold circuit in Hybrid mode – Continuous scan mode. Since Differential input mode is supported in Hybrid mode, this example describes a case that channel-dedicated sample-and-hold circuits are set to Differential input mode. The configuration and behavior of example operation are shown in Table 53.30, Table 53.31 and Figure 53.21.

Table 53.30 Configuration of example operation for channel-dedicated sample-and-hold circuit in Hybrid mode – Continuous scan mode

Item	Configuration
A/D converter	Set ADC0 to Hybrid mode – Continuous scan mode
Channel-dedicated sample-and-hold circuit	Set SH0 to SH2 to enabled and Differential input mode. (AN000 to AN005 are subject to A/D conversion using channel-dedicated sample-and-hold circuit.)
Virtual channel[*1]	● Assign AN000, AN002 and AN004 to virtual channel 0 to 2, and set to Differential input mode. This treats AN000, AN002 and AN004 as the non-inverting inputs (+) of the differential input, and AN001, AN003, and AN005 as the inverting inputs (−) of the differential input.
Scan group[*1]	● Assign virtual channel 0 to 2 to scan group 0. ● Set scan group 0 to convert with ADC0.

Note 1. When channel-dedicated sample-and-hold circuit is used, there are restrictions on the setting of virtual channels. For more details, see section 53.3.16.6. Restrictions on Channel-dedicated Sample-and-hold Circuit.

Table 53.31 Example operation of channel-dedicated sample-and-hold circuit in Hybrid mode – Continuous scan mode (1 of 2)

Step	Behavior
1	The scanning operation of scan group 0 starts when a trigger for scan group 0 is input.

Table 53.31 Example operation of channel-dedicated sample-and-hold circuit in Hybrid mode – Continuous scan mode (2 of 2)

Step	Behavior
2	Channel-dedicated sample-and-hold circuit (SH0 to SH2) starts sampling the analog inputs (AN000, AN002, AN004) and holds them.
3	The A/D converter performs oversampling once for each output of channel-dedicated sample-and-hold circuit (SH0 to SH2).
4	After one oversampling for each analog channel, channel-dedicated sample-and-hold circuit releases the held voltage and enters the idle state.
5	Step 2 to 4 is repeated until oversampling corresponding to the number of TAPs of the digital filter and the number of A/D-converted value addition/average times is performed for each analog channel.
6	A/D conversion data corresponding to each analog channel is output. The A/D conversion data is stored in the data register (ADDRi (i = 0, 2, 4)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 8)).
7	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
8	The second and subsequent rounds of scanning operations repeat the operation of Step 2 to 4, retaining the oversampling data stored in the digital filter. Each one time oversampling or each multiple times oversampling corresponding to the number of A/D-converted value addition/averaging times is performed for each analog channel, the next A/D conversion data is output. The A/D conversion data is stored in the data register (ADDRi (i = 0, 2, 4)). If FIFO is used, A/D conversion data is also stored in FIFO data register (ADFIFODRk (k = 0 to 8)).
9	If the scan end interrupt is enabled, the scan end interrupt corresponding to that scan group is generated when the A/D conversion of all virtual channels assigned to that scan group is completed.
10	Thereafter, until the forcibly stop process is performed, Step 8 to 9 are repeated, and the scanning operation continues. To perform the forcibly stop, follow to section 53.6.4. Force Stops the A/D Conversion Operation. If scanning operation is aborted by the forcibly stop process, A/D converter and channel-dedicated sample-and-hold circuit enter the idle state. The oversampling data in digital filter are discarded at the end of the scanning operation.

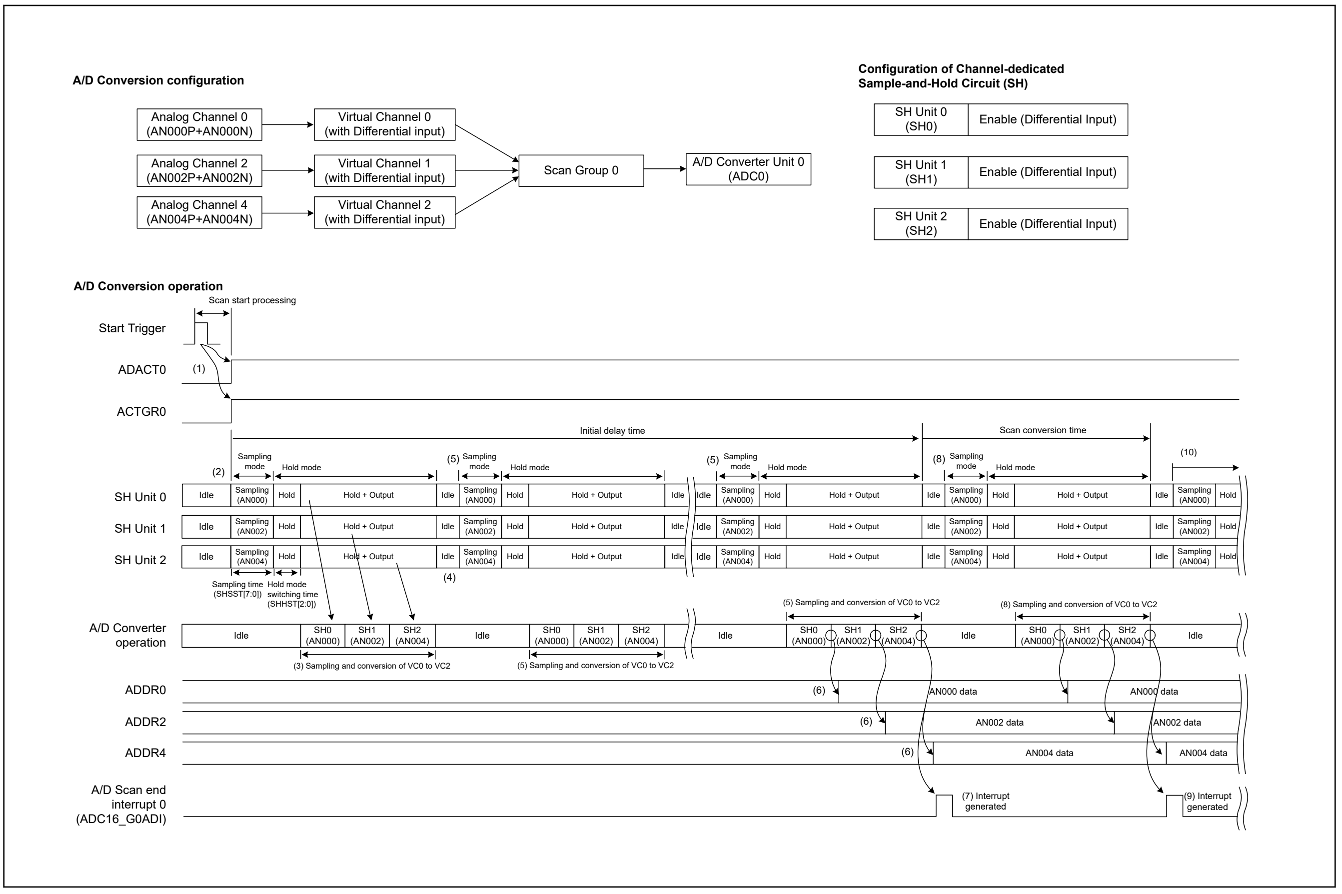


Figure 53.21 Basic operation of channel-dedicated sample-and-hold circuit in Hybrid mode – Continuous scan mode

(3) Basic Operation in Hybrid Mode - Background Continuous Scan Mode

This section shows the basic operation of channel-dedicated sample-and-hold circuit in Hybrid mode – Background continuous scan mode. Since Differential input mode is supported in Hybrid mode, this example describes a case that channel-dedicated sample-and-hold circuits are set to Differential input mode. The configuration and behavior of example operation are shown in Table 53.32, Table 53.33 and Figure 53.22.

Table 53.32 Configuration of example operation for channel-dedicated sample-and-hold circuit in Hybrid mode – Background continuous scan mode

Item	Configuration
A/D converter	Set ADC0 to Hybrid mode – Continuous scan mode
Channel-dedicated sample-and-hold circuit	Set SH0 to SH2 to enabled and Differential input mode. (AN000 to AN005 are subject to A/D conversion using channel-dedicated sample-and-hold circuit.)
Virtual channel*1	● Assign AN000, AN002 and AN004 to virtual channel 0 to 2, and set to Differential input mode. This treats AN000, AN002 and AN004 as the non-inverting inputs (+) of the differential input, and AN001, AN003, and AN005 as the inverting inputs (−) of the differential input.
Scan group*1	● Assign virtual channel 0 to 2 to scan group 0. ● Set scan group 0 to convert with ADC0.

Note 1. When channel-dedicated sample-and-hold circuit is used, there are restrictions on the setting of virtual channels. For more details, see section 53.3.16.6. Restrictions on Channel-dedicated Sample-and-hold Circuit.

Table 53.33 Example operation of channel-dedicated sample-and-hold circuit in Hybrid mode – Background continuous scan mode

Step	Behavior
1	The scanning operation of scan group 0 starts when a trigger for scan group 0 is input.
2	Channel-dedicated sample-and-hold circuit (SH0 to SH2) starts sampling the analog inputs (AN000, AN002, AN004) and holds them.
3	The A/D converter performs oversampling once for each output of channel-dedicated sample-and-hold circuit (SH0 to SH2).
4	After one oversampling for each analog channel, channel-dedicated sample-and-hold circuit releases the held voltage and enters the idle state.
5	Step 2 to 4 is repeated. After the oversampling corresponding to the number of TAPs of the digital filter and the number of A/D-converted value addition/averaging times is performed to each analog channel, first A/D conversion data becomes in ready to be output. Step 2 to 4 is repeated until oversampling corresponding to the number of TAPs of the digital filter and the number of A/D-converted value addition/average times is performed for each analog channel.
6	The second and subsequent rounds of scanning operations repeat the operation of Step 2 to 4, retaining the oversampling data stored in the digital filter. Each time oversampling is performed for each analog channel, the data in the digital filter is updated. Each one time oversampling or each multiple times oversampling corresponding to the number of A/D-converted value addition/averaging times is performed for each analog channel, the next A/D conversion data becomes in ready to be output.
7	When an A/D conversion start trigger is input during background continuous scanning operation, the most recent A/D conversion data at that time is stored in the data register (ADDRi (i = 0, 2, 4)). When FIFO function is set to enabled, A/D conversion data is also stored in the FIFO data register (ADFIFODRk (k = 0 to 8)). Also, if scan end interrupt is set to enable, scan end interrupt is generated. If FIFO interrupt is set to enable, FIFO data read request is generated when the condition for generating it is met.
8	Thereafter, until the forcibly stop process is performed, background continuous scanning operation (Step 6) is repeated. And whenever the A/D conversion start trigger is input during background continuous scanning operation, the A/D conversion data is output (Step 7). To perform the forcibly stop, follow to section 53.6.4. Force Stops the A/D Conversion Operation. If scanning operation is aborted by the forcibly stop process, A/D converter and channel-dedicated sample-and-hold circuit enter the idle state. The oversampling data in digital filter are discarded at the end of the scanning operation.

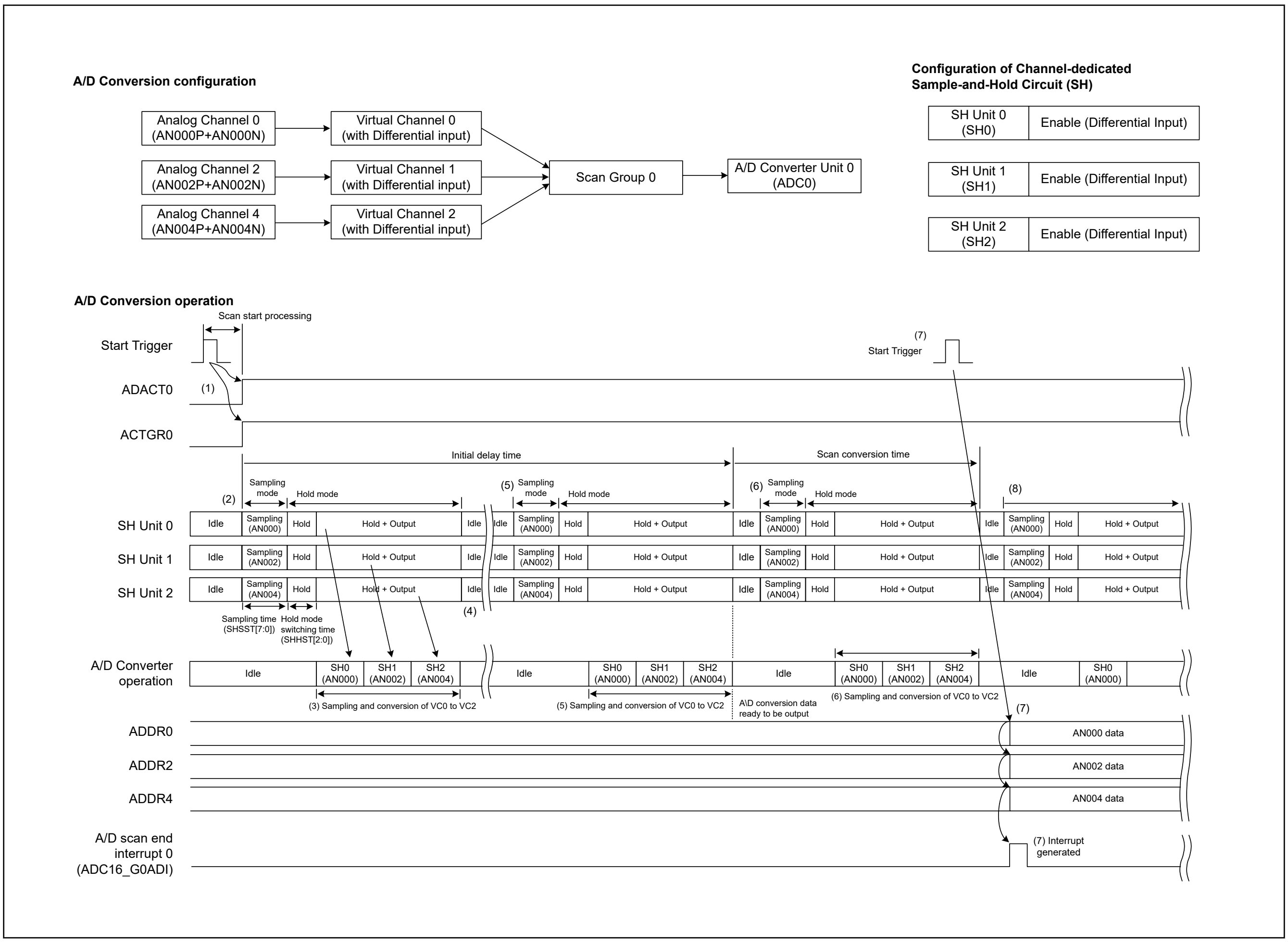


Figure 53.22 Basic operation of channel-dedicated sample-and-hold circuit in Hybrid mode – Background continuous scan mode

(4) Basic Operation in Hybrid Mode - Fixed-channel Continuous Scan Mode

This section shows the basic operation of channel-dedicated sample-and-hold circuit in Hybrid mode – Fixed-channel continuous scan mode. The configuration and behavior of example operation are shown in Table 53.34, Table 53.35. This mode is available only when the number of fixed channels is set to 3 and the fixed channels are used only for channel-dedicated sample-and-hold circuit.

Table 53.34 Configuration of example operation for channel-dedicated sample-and-hold circuit in Hybrid mode – Fixed-channel continuous scan mode

Item	Configuration
A/D converter	Set ADC0 to Hybrid mode – Fixed-channel continuous scan mode.
Channel-dedicated sample-and-hold circuit	Set SH0 to SH2 to enabled and Differential input mode. (AN000 to AN005 are subject to A/D conversion using channel-dedicated sample-and-hold circuit.)
Virtual channel*1	• Assign AN000, AN002 and AN004 to virtual channel 0 to 2, and set to Differential input mode. This treats AN000, AN002 and AN004 as the non-inverting inputs (+) of the differential input, and AN001, AN003, and AN005 as the inverting inputs (−) of the differential input.
Scan group*1	• Assign virtual channel 0 to 2 to scan group 0. • Set scan group 0 to convert with ADC0.

Note 1. When channel-dedicated sample-and-hold circuit is used, there are restrictions on the setting of virtual channels. For more details, see section 53.3.16.6. Restrictions on Channel-dedicated Sample-and-hold Circuit.

Table 53.35 Example operation of channel-dedicated sample-and-hold circuit in Hybrid mode – Fixed-channel continuous scan mode

Step	Behavior
1	When a start trigger is input and the ADSR.ADACTm (m = 0, 1) bit becomes 1 (A/D conversion in progress), A/D conversion of the scan group selected by the ADSGCRm (m = 0, 1) register is started.
2	Three fixed-channel channel-dedicated sample-and-hold circuit (SH0 to SH2) starts sampling the analog inputs (AN000, AN002, AN004) and holds them.
3	The A/D converter performs oversampling once for each output of channel-dedicated sample-and-hold circuit (SH0 to SH2).
4	One non-fixed channel is A/D converted in SAR mode. Non-fixed channels are converted from the virtual channels assigned to the relevant scan group in ascending order of channel number.
5	After performing A/D conversions for the three fixed channels and one non-fixed channel, channel-dedicated sample-and-hold circuit releases the held voltage and enters the idle state.
6	When A/D conversion for one channel is completed, the A/D conversion result is stored in the A/D data register (ADDRi.DATA (i = 0 to 22), ADEXDRj (j = 0, 1, 4, 5, 6, 8, 9, 16, 17, 18, 20, 21, 22)) corresponding to the conversion target channel. If FIFO is used, the A/D conversion result is also stored to the FIFO data register (ADFIFODRk (k = 0 to 8)).
7	If scan end interrupt is set to enable, scan end interrupt is generated.
8	The conversion time of idle cycles is counted until the end of the scan cycle.
9	Thereafter, until the A/D conversion stop process is performed, Background continuous scanning operation (Step 2 to 6) is repeated. For fixed channel scan, conversion of the same channel is repeated, and for non-fixed channel scan, conversion is repeated one channel at a time up to the last channel assigned to the scan group, and then back to the first channel. To stop A/D conversion, you must follow to section 53.6.4. Force Stops the A/D Conversion Operation.

53.3.16.4 Constant Sampling Function

Constant sampling is a function in which the channel-dedicated sample-and-hold circuit continues sampling before and after A/D conversion starts. This makes it possible to simultaneously sample multiple channels and acquire A/D conversion data at the same time.

Figure 53.23 shows the difference in A/D conversion operation when constant sampling is enabled (ADSHCSCR.SHCSSTm = 1 (x = 0, 1)) and when it is disabled (ADSHCSCR.SHCSSTm = 0 (x = 0, 1)).

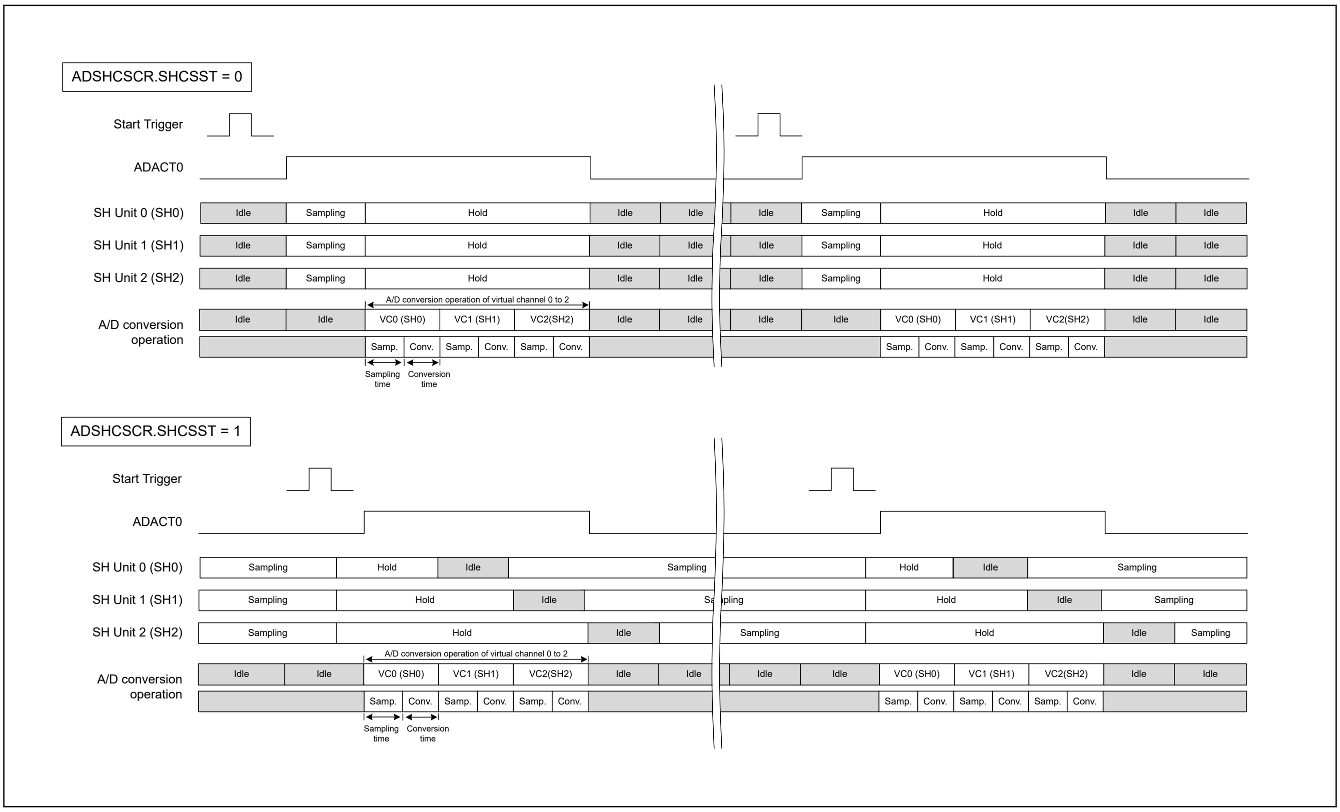


Figure 53.23 Difference between the function of channel-dedicated sample-and-hold circuit when using or not using constant sampling

53.3.16.5 Channel-dedicated Sample-and-hold Circuit Bypass Function

When channel-dedicated sample-and-hold (S&H) circuit is in differential input mode, the differential input bypass function allows disconnection detection with the disconnection detection assist function of A/D converter (single-ended input mode).

This function is intended to avoid the complex calibration procedure that occurs when the S&H switches from differential input mode to single-ended input mode, since the differential input bypass function is supported only in single-ended input mode.

When the ADSHDBPCR.SHDBPn (n = 0 to 8) bit is set to 1, the hold output of the S&H circuit is not selected for the AD conversion during the scan group. Instead, the input channel of the S&H circuit is selected for the AD conversion. The disconnection detection assist function can be executed in the scan group. For example, the following settings are possible in the order:

1. SH0 is enabled.
2. AN000 is assigned to scan group 0. The input mode is set to differential mode.
3. AN000 and AN001 are assigned to scan group 1. The input mode is set to single-ended mode and the disconnection detection assist function is enabled.
4. The ADSHDBPCR.SHDBP0 bit is set to 0 and the ADSHDBPCR.SHDBP1 bit is set to 1.

In the AD conversion of scan group 0, the sample-and-hold operation is performed with SH0, and the hold output is AD converted.

In the AD conversion of scan group 1, the disconnection detection assist function for the input channels of SH0, AN000 and AN001, is performed.

In this case, the SH0 setting is not changed and recalibration for SH0 is not required.

53.3.16.6 Restrictions on Channel-dedicated Sample-and-hold Circuit

When using channel-dedicated sample-and-hold circuit, observe the restrictions described in the sections that follow.

(1) Limitations on usage of channel-dedicated sample-and-hold circuit

Channel-dedicated sample-and-hold circuit is only available in the following:

- SAR mode – Single scan mode
- Hybrid mode – Single scan mode
- Hybrid mode – Continuous scan mode
- Hybrid mode – Background continuous scan mode
- Hybrid mode – Fixed channel continuous scan mode *1

In the operation mode other than above, channel-dedicated sample-and-hold circuit is prohibited.

In addition, the constant sampling function can only be used in SAR mode – Single scan mode.

The sampling time, hold mode switching time, and A/D sampling time of the channel-dedicated sample-and-hold circuit should be set as follows.

- 3.5 × (ADSHSTR0.SHHST[2:0] + ADSSTRx.SSTy[9:0]) ≥ ADSHSTR0.SHSST[7:0]

Note 1. Only fixed channels are supported when fixed channels are set to 3ch and all fixed channels enable channel-dedicated sample-and-hold circuit. Non-fixed channels cannot use channel-dedicated sample-and-hold circuit.

(2) Restrictions on self-calibration operation of channel-dedicated sample-and-hold circuit

Self-calibration operation is required when channel-dedicated sample-and-hold circuit is used. For more details on self-calibration procedures for channel-dedicated sample-and-hold circuit and the restrictions on it, see section 53.3.8. Self-calibration.

(3) Restrictions on assigning virtual channels and scan groups

For analog channels using channel-dedicated sample-and-hold circuit, assign them to virtual channels so that A/D conversion is performed at the front of the scan group. Specifically, assign an analog channel that uses channel-dedicated sample-and-hold circuit from the virtual channel that has the smallest virtual channel number among the virtual channels assigned to that scan group. (Assign the analog channels of channel-dedicated sample-and-hold circuit to the front virtual channels in ascending order of the virtual channel number.)

If this restriction is violated, the operation of channel-dedicated sample-and-hold circuit and A/D converter is not guaranteed.

(4) Prohibition of multiple assignments of channels using the same channel-dedicated sample-and-hold circuit in the same scan group

It is prohibited to assign more than one virtual channel that uses the same channel-dedicated sample-and-hold circuit to a single scan group. Specific restrictions are the followings:

[Restrictions]

- The virtual channel whose signal source is selected as the even-numbered analog channel connected to channel-dedicated sample-and-hold circuit and the virtual channel whose signal source is selected as the odd-numbered analog channel must not be assigned to the same scan group without using the bypass function.
- It is prohibited to assign the even-numbered analog channels or odd-numbered analog channels connected to channel-dedicated sample-and-hold circuit to multiple virtual channels, and assign them to the same scan group.

Operation is not guaranteed if this restriction is violated.

To avoid this restriction, assign virtual channels that use the same channel-dedicated sample-and-hold circuit to separate scan groups. This restriction can be avoided by preventing A/D conversion from occurring more than once for the same channel-dedicated sample-and-hold circuit during one scanning operation. For the example workaround for this restriction, see (2) Advanced operation in SAR mode - Single scan mode.

(5) Restriction on differential input bypass function

When a scan group is set as differential input bypass, all channel-dedicated sample-and-hold circuits including another scan group cannot be used for SH disconnection detection function and constant sampling function.

(6) Restrictions on group priority operation

Channel-dedicated sample-and-hold circuit is available only for the highest priority scan group in group priority operation.

(7) Restrictions on channel-dedicated sample-and-hold circuit (S&H) constant sampling function

- S&H constant sampling function is only available in SAR mode – Single scan mode. In the operation mode other than specified, S&H constant sampling function is prohibited.
- When using the S&H constant sampling function, the virtual channel settings of the channel that converts S&H, use the virtual channel number register corresponding to the S&H unit as shown in Table 53.36.

Table 53.36 Virtual channel number associated with sample-and-hold circuit unit

A/D converter unit	Channel-dedicated sample-and-hold circuit unit	Assigned virtual channel number
ADC0	S&H unit 0	Virtual Channel 0
	S&H unit 1	Virtual Channel 1
	S&H unit 2	Virtual Channel 2
ADC1	S&H unit 4	Virtual Channel 4
	S&H unit 5	Virtual Channel 5
	S&H unit 6	Virtual Channel 6

- When using the S&H constant sampling function, set the S&H sampling time (ADSHSTRm.SHSST[7:0](m = 0, 1)) according to the following rules:
  - When synchronization between units is disabled — ADSHSTRx.SHSSTx (x = 0 to 3) = 8’h4 (reset value)
  - When synchronization between units is enabled — The value for which the total time of S&H sampling time and S&H hold time (ADSHSTRm.SHSST[7:0] + ADSHSTRm.SHHST[2:0] (m = 0, 1)) is equal to the synchronization period.
- When S&H constant sampling function is enabled, it is prohibited to use any of the scan groups with S&H bypass input by setting the ADSHDBPCR register.
- When unit synchronization is enabled and S&H is used with constant sampling, set the synchronous period to 6 ADCLK or longer.
- Set S&H constant sampling (ADSHCSCR.SHCSSTm (m = 0, 1)) before starting A/D conversion after calibration.
- When S&H is used with constant sampling, the S&H sampling period is the period from the end of the S&H conversion to the start of the next S&H conversion. When you enter a conversion start trigger continuously, ensure that the trigger interval is sufficient to ensure the sampling period of S&H.

(8) Restrictions on channel-dedicated sample-and-hold circuit (S&H) disconnection detection assist function

- S&H disconnection detection assist function is only available in SAR mode – Single scan mode. In the operation mode other than specified, S&H disconnection detection assist function is prohibited.
- When S&H disconnection detection assist function is enabled, it is prohibited to use any of the scan groups with S&H bypass input by setting the ADSHDBPCR register.

(9) Restrictions on channel-dedicated sample-and-hold circuit (S&H) self-diagnosis function

- S&H self-diagnosis function is only available in SAR mode – Single scan mode. In the operation mode other than specified, S&H self-diagnosis function is prohibited.
- It is prohibited to set up normal conversion and S&H self-diagnostic channels of the same S&H unit in one scan group.

53.3.17 Disconnection Detection Assist Function

Disconnection detection assist function is the function that fixes the charges of the sampling capacitance of the A/D converter to a specified condition prior to starting A/D conversion. This function can be used to detect the wiring disconnection connected to analog input.

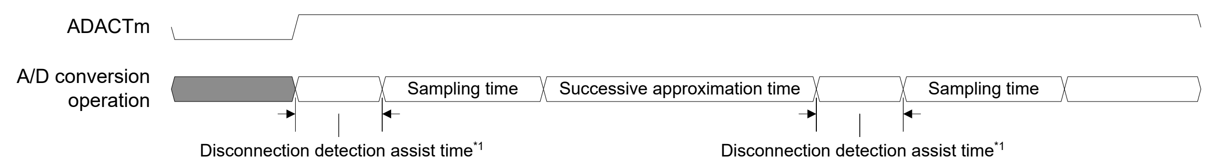
Table 53.37 shows the operation mode in which disconnection detection assist function is available. And when using disconnection detection assist function, it is recommended to enable the high accuracy mode by setting the ADACMDR.ADHACMDn (n = 0,1) bits to 1.

Table 53.37 List of operation mode for which disconnection detection assist function is available

Operation mode – Scan mode	When channel-dedicated sample-and-hold circuit is not used	When channel-dedicated sample-and-hold circuit is used
SAR mode – Single scan mode	✓	✓
SAR mode – Continuous scan mode	✓	—
Oversampling mode – Single scan mode	—	—
Oversampling mode – Continuous scan mode	—	—
Oversampling mode – One-channel continuous scan mode	—	—
Hybrid mode – Single scan mode	✓	—
Hybrid mode – Continuous scan mode	✓	—
Hybrid mode – Background continuous scan mode	—	—
Hybrid mode – Fixed-channel continuous scan mode	—	—

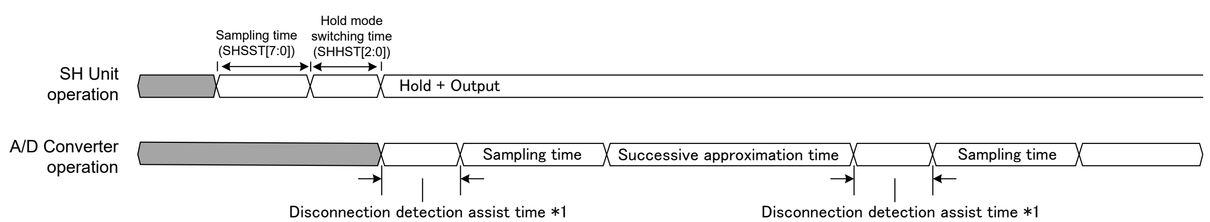
Note: ✓: available, —: not available

Figure 53.24 shows an example operation of A/D conversion when disconnection detection assist function is used, and Figure 53.25 shows an example operation of A/D conversion when disconnection detection assist function and channel-dedicated sample-and-hold circuit are used. Figure 53.26 shows an example of the disconnection detection when precharge is selected, and Figure 53.27 shows an example of the disconnection detection when discharge is selected.



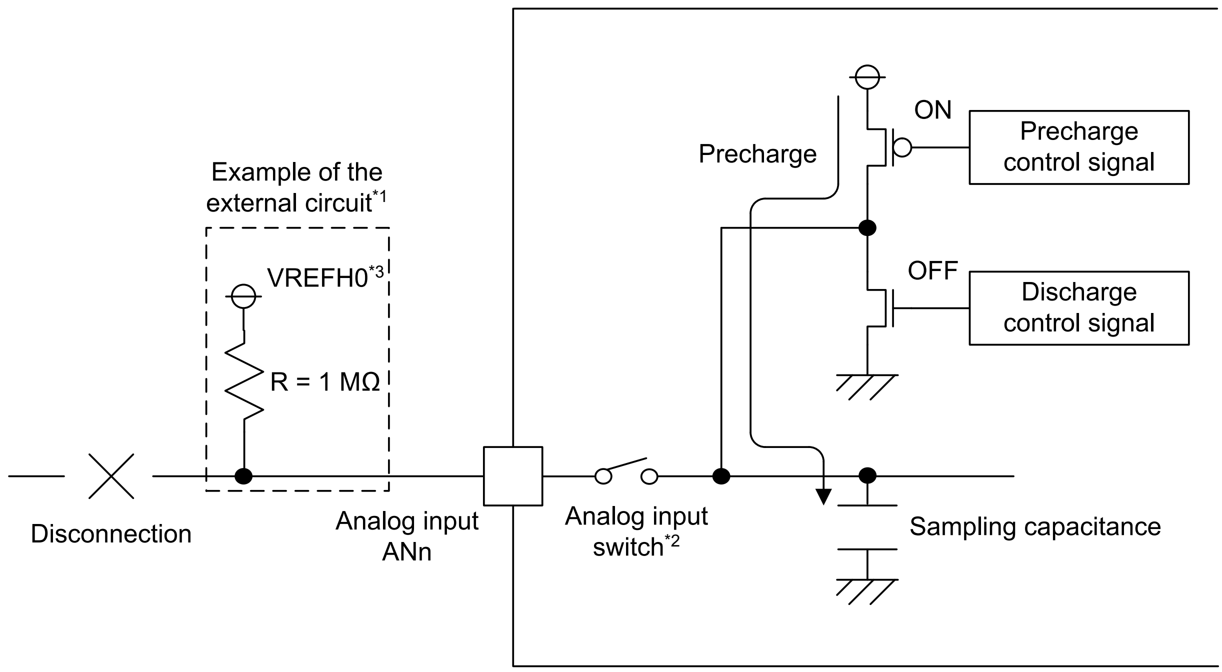
Note: m = 0, 1
Note 1. 3 to 254 cycles of ADCLK

Figure 53.24 Operation of A/D conversion when disconnection detection assist function is used



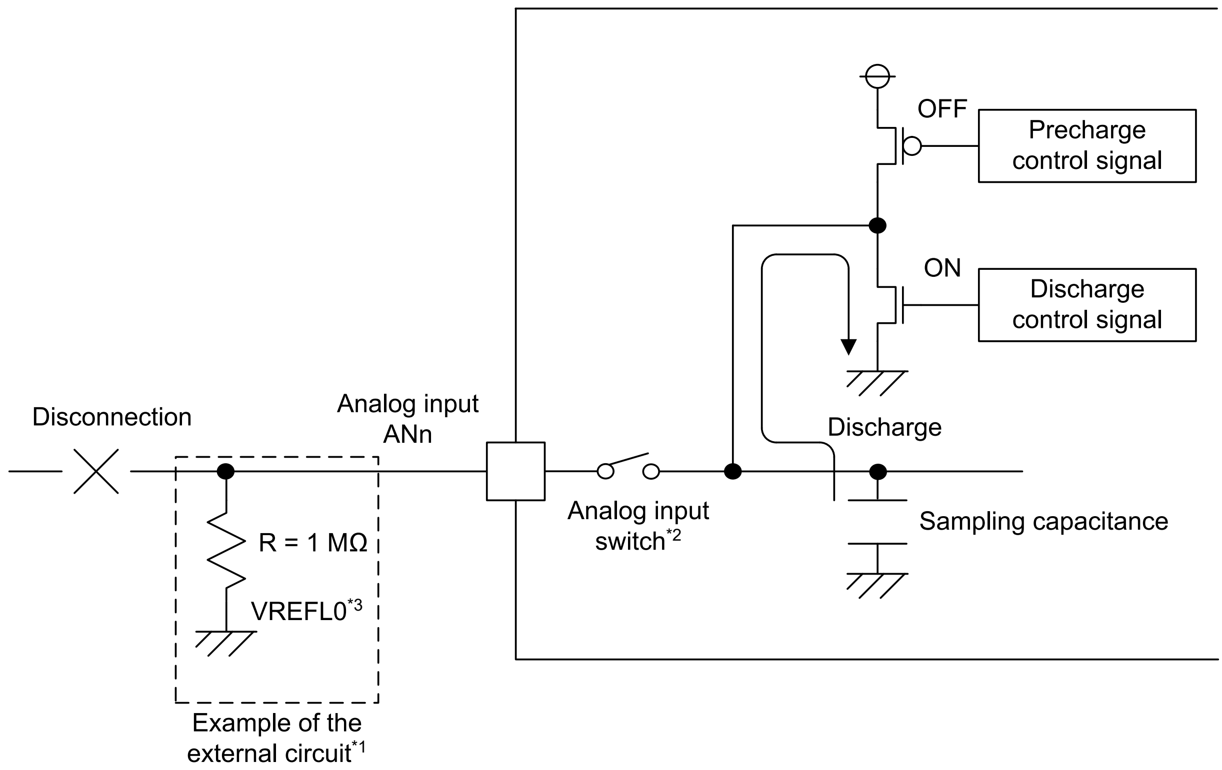
Note 1. 3 to 254 ADCLK cycles

Figure 53.25 Operation of A/D conversion when the disconnection detection assist function and channel-dedicated sample-and-hold circuit are used



Note 1. The converted result should be used after it is fully evaluated because the result data when disconnection occurs varies depending on the external circuit.
Note 2. Turn off when precharge, and turn on when sampling the Analog input ANn.
Note 3. Example is for ADC0; use VREFH for ADC1.

Figure 53.26 Example of disconnection detection when precharge is selected



Note: Connect the same reference ground as the ADC16H which is used.
Note 1. The converted result should be used after it is fully evaluated because the result data when disconnection occurs varies depending on the external circuit.
Note 2. Turn off when discharge and turn on when sampling the Analog input ANn.
Note 3. Example is for ADC0; use VREFL for ADC1.

Figure 53.27 Example of disconnection detection when discharge is selected

53.3.17.1 Restrictions on Disconnection Detection Assist Function

When using disconnection detection assist function, observe the following restrictions:

- Disconnection detection assist function is only available in the following modes:
  - SAR mode – Single scan mode
  - SAR mode – Continuous scan mode
  - Hybrid mode – Single scan mode
  - Hybrid mode – Continuous scan mode

  In the operation mode other than above, disconnection detection assist function is prohibited.
- Disconnection detection assist function with channel-dedicated sample-and-hold circuit is only available in the following modes:
  - SAR mode – Single scan mode

  In the operation mode other than above, disconnection detection assist function with channel-dedicated sample-and-hold circuit is prohibited.

53.3.18 Group Priority Operation

Group priority operation function is the function that performs the scanning operation (A/D conversion) for a scan group with a higher priority based on the priority of scan group.

The priority of scan groups is group 0 > group 1 > … > group n (n: the number of maximum scan group - 1).

Table 53.38 shows the conditions when group priority operation can be used and the corresponding register settings. Set the operation according to Table 53.38.

Table 53.38 Group priority operation usable conditions and corresponding register settings

Operating conditions		Group priority operation				
			Setting for ADGSPCR register [*1]			
Operation mode of A/D converter		Operation	PGSm	RSCNm	LGRRSm	GRPm
SAR mode	Single scan mode	✓	1	1	1	0
	Continuous scan mode	✓	1	1	1	1
Oversampling mode		—	0	0	0	0
Hybrid mode		—	0	0	0	0

Note: ✓: Available, —: Not Available (Prohibited)
m = 0, 1

Note 1. Settings other than the specified value are prohibited.

53.3.18.1 Group Priority Operation in SAR Mode – Single Scan Mode

In group priority operation in SAR mode – Single scan mode, the single scanning operation is performed according to the priority of scan group.

If an A/D conversion start trigger for the high priority group is input during a scanning operation of the low priority group, the scanning operation of the low priority group is interrupted, and the scanning operation of the high priority group is started. After the scanning operation of the high priority group is completed, the scanning operation of the low priority group is resumed.

When an A/D conversion start trigger of the low priority group during a scanning operation of the high priority group is input, the scanning operation of the low priority group is performed after the scanning operation of the high priority group is completed.

Example of the Group Priority Operation in SAR Mode – Single Scan Mode

Table 53.39 and Figure 53.28 show the example of group priority operation in SAR mode – Single scan mode when analog channel 0 (AN000) is assigned to scan group 0 (high priority) and analog channel 1 to 3 (AN001 to AN003) are assigned to scan group 1 (low priority).

Table 53.39 Example of the group priority operation in SAR mode – Single scan mode (2 groups)

Step	Operation
1	The scanning operation of scan group 1 starts when a trigger for scan group 1 is input.
2	When A/D conversion of each channel in scan group 1 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
3	If a trigger for scan group 0 is input during A/D conversion for scan group 1, the scanning operation of scan group 1 is suspended and the scanning operation of scan group 0 is started. [*1]
4	When A/D conversion of each channel in scan group 0 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
5	If a scan end interrupt for scan group 0 is enabled, a scan end interrupt occurs.
6	The scanning operation of scan group 1 is resumed after scanning of scan group 0 is completed. The scanning operation is resumed from the A/D conversion incomplete channel. [*1]
7	When A/D conversion of each channel in scan group 1 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
8	If a scan end interrupt for scan group 1 is enabled, a scan end interrupt occurs.
9	When all scanning operations are completed, ADSR.ADACTm (m = 0, 1) bits are cleared and the A/D converter enters the idle status.

Note 1. The timing of interruption and restart of scanning operation depends on the operating conditions. For details, see section 53.3.18.3. Restrictions on Group Priority Operation.

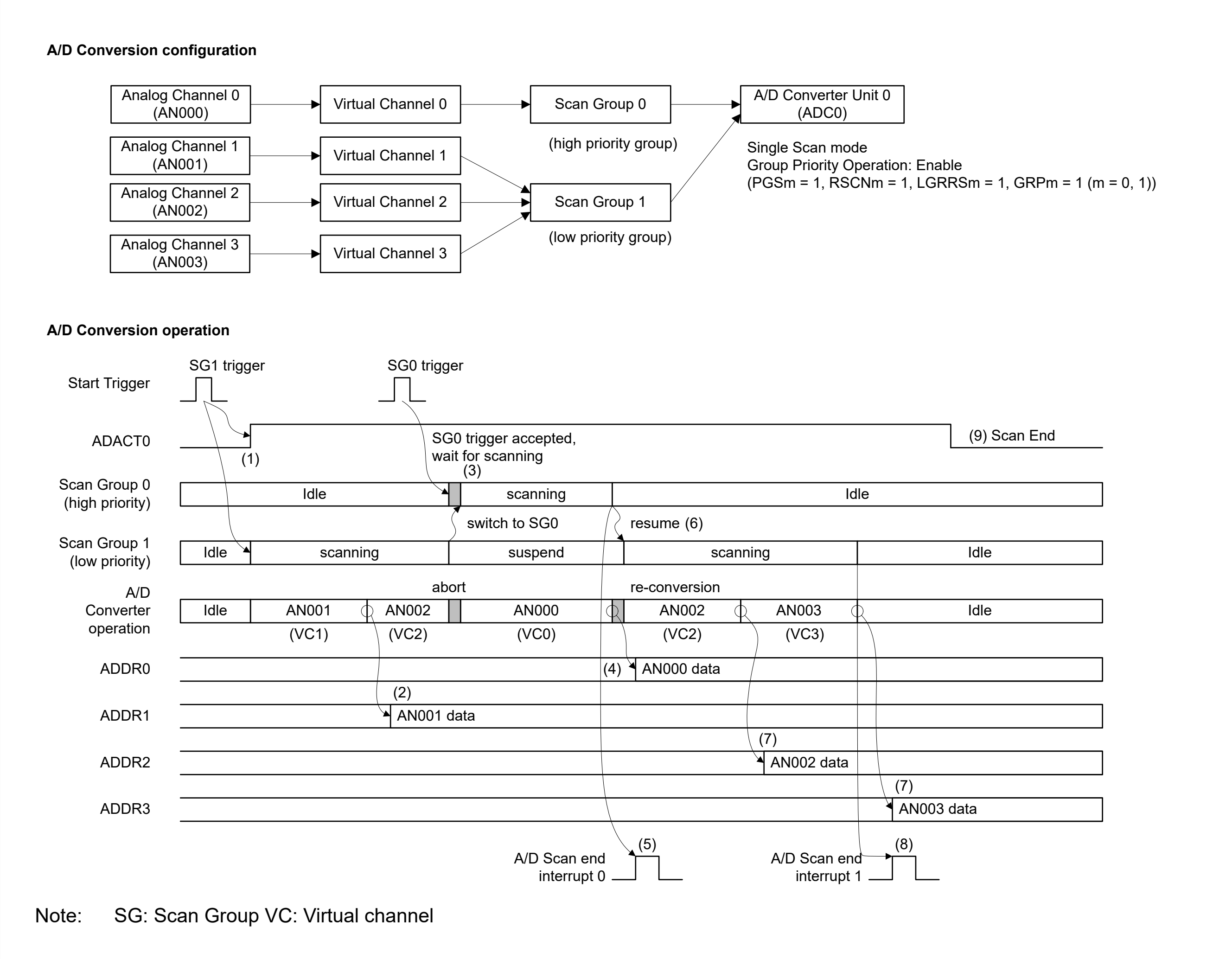


Figure 53.28 Example of group priority operation in SAR mode – Single scan mode (2 groups)

53.3.18.2 Group Priority Operation in SAR Mode – Continuous Scan Mode

In group priority operation in SAR mode – Continuous scan mode, the continuous scanning operation is performed for the scan group (continuous-scan group) in which scanning operation was first started. If an A/D conversion start trigger of another scan group (interrupt-scan group) is input during the continuous scanning operation of the continuous-scan group, the following operations are performed according to the priority of the interrupt-scan group.

1. Priority: When Interrupt-scan group (high) > Continuous-scan group (low)
   - The scanning operation of the continuous-scan group is interrupted, and the single scanning operation of the interrupt-scan group is started.
   - The continuous scanning operation of the continuous-scan group is restarted after the single scanning operation of the interrupt-scan group is completed.
2. Priority: When Continuous-scan group (high) > Interrupt-scan group (low)
   - This operation is prohibited. Operation is not guaranteed.

   In case 1. above, if an A/D conversion start trigger for another scan group (interrupt-scan group B) is input during the single scanning operation of the interrupt-scan group (interrupt-scan group A), the following operation is performed according to the priority of interrupt-scan group B.
3. Priority: When Interrupt-scan group B (high) > Interrupt-scan group A (middle) > Continuous-scan group (low)
   - The single scanning operation of interrupt-scan group A is suspended, and the single scanning operation of interrupt-scan group B is started.

- After the single scanning operation of interrupt-scan group B is completed, the single scanning operation of interrupt-scan group A is resumed.
- After the scanning operation of interrupt-scan group A is completed, the continuous scanning operation of the continuous-scan group is resumed.

4. Priority: When interrupt-scan group A (high) > interrupt-scan group B (medium) > continuous-scan group (low)
   - After the single scanning operation of interrupt-scan group A is completed, the single scanning operation of interrupt-scan group B is started.
   - After the single scanning operation of interrupt-scan group B is completed, the continuous scanning operation of the continuous-scan group is resumed.
5. Priority: When interrupt-scan group A (high) > continuous-scan group (medium) > interrupt-scan group B (low)
   - This operation is prohibited. Operation is not guaranteed.

Example operations are shown in the followings:

(1) Example of two-group priority operation in SAR mode – Continuous scan mode

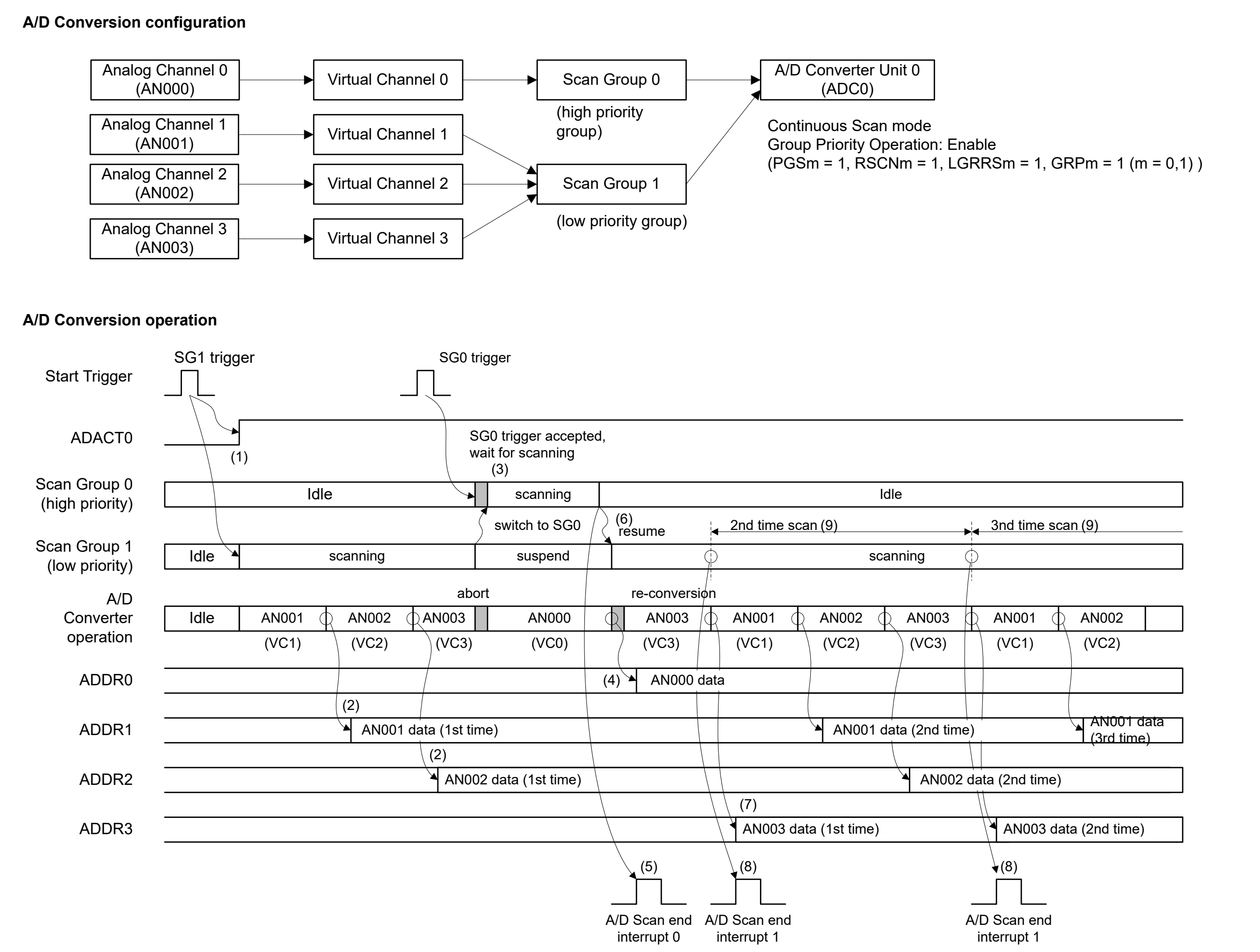
Table 53.40 and Figure 53.29 show the example of group priority operation in SAR mode – Continuous scan mode when analog channel 0 (AN000) is assigned to scan group 0 (high-priority group) and analog channel 1 to 3 (AN001 to AN003) is assigned to scan group 1 (low-priority group).

Table 53.40 Example of the group priority operation in SAR mode – Continuous scan mode (2 groups)

Step	Operation
1	The continuous scanning operation of scan group 1 (low priority group) is started by the trigger input of scan group 1.
2	When A/D conversion of each channel in scan group 1 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
3	If a trigger for scan group 0 is input during A/D conversion for scan group 1, the scanning operation of scan group 1 is suspended and scanning operation of scan group 0 is started. *1
4	When A/D conversion of each channel in scan group 0 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
5	If a scan end interrupt for scan group 0 is enabled, a scan end interrupt occurs.
6	The scanning operation of scan group 1 is resumed after scanning of scan group 0 is completed. The scanning operation is resumed from the A/D conversion incomplete channel. *1
7	When A/D conversion of each channel in scan group 1 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
8	If a scan end interrupt for scan group 1 is enabled, a scan end interrupt occurs.
9	Scan group 1 continues the continuous scanning operation. *2

Note 1. The timing of interruption and restart of scanning operation depends on the operating conditions. For details, see section 53.3.18.3. Restrictions on Group Priority Operation.

Note 2. To stop the continuous scanning operation, follow the procedures described in section 53.6.4. Force Stops the A/D Conversion Operation.



Note: SG: Scan Group VC: Virtual channel

Figure 53.29 Example of group priority operation in SAR mode – Continuous scan mode (2 groups)

(2) Example of three-groups priority operation in SAR mode – Continuous scan mode

Table 53.41 and Figure 53.30 show the example of group priority operation when analog channel 0 (AN000) is assigned to scan group 0 (high priority), analog channels 1 and 2 (AN001 and AN002) are assigned to scan group 1 (middle priority), and analog channel 3 to 5 (AN003 to AN005) are assigned to scan group 2 (low priority).

Table 53.41 Example of the group priority operation in SAR mode – Continuous scan mode (3 groups) (1 of 2)

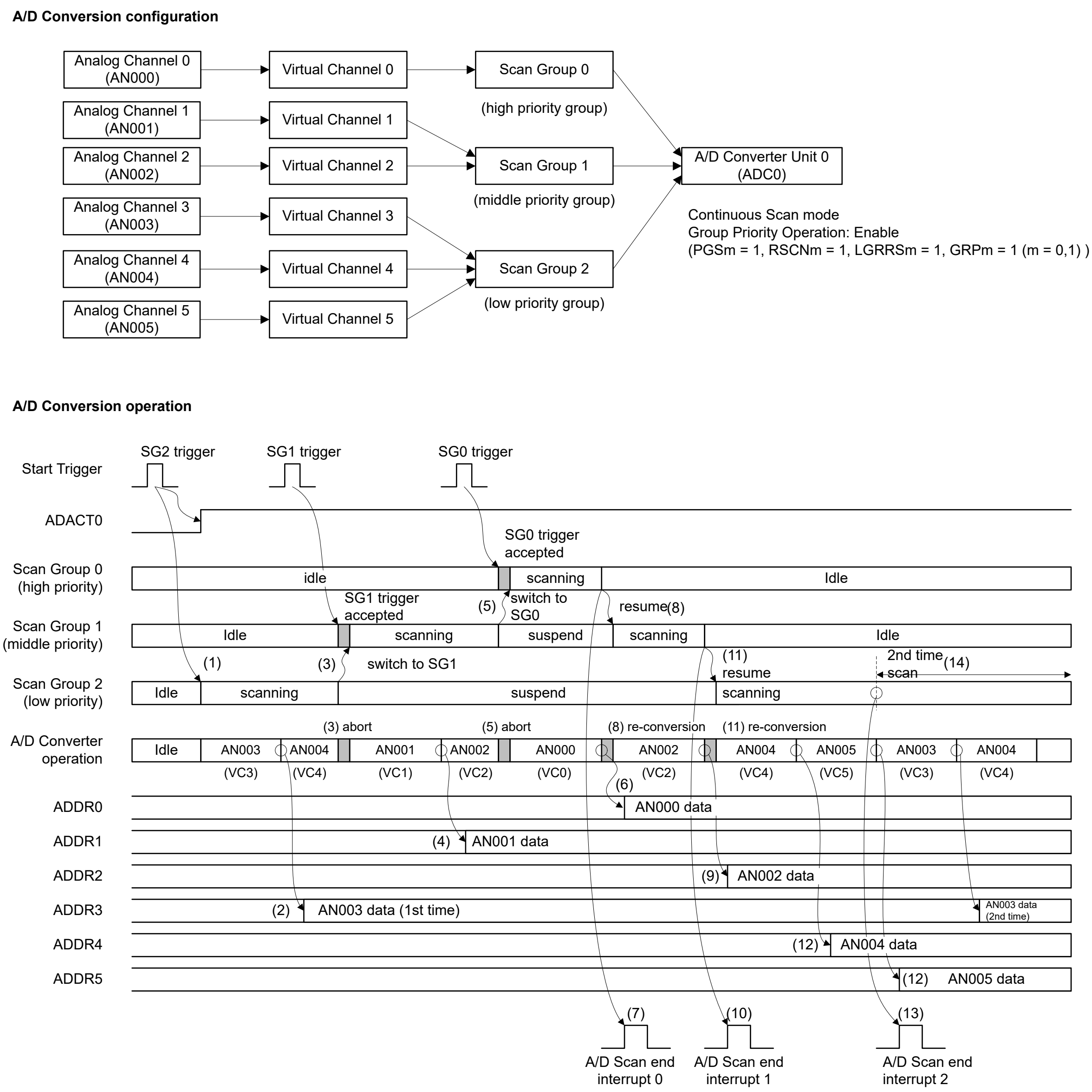
Step	Operation
1	The continuous scanning operation of scan group 2 is started by the trigger input of scan group 2.
2	When A/D conversion of each channel in scan group 2 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
3	If a trigger for scan group 1 is input during A/D conversion on scan group 2, the scanning operation of scan group 2 is suspended and the scanning operation of scan group 1 is started. *1
4	When A/D conversion of each channel in scan group 1 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
5	If a trigger for scan group 0 is input during A/D conversion for scan group 1, the scanning operation of scan group 1 is suspended and the scanning operation of scan group 0 is started. *1
6	When A/D conversion of each channel in scan group 0 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
7	If a scan end interrupt for scan group 0 is enabled, a scan end interrupt occurs.

Table 53.41 Example of the group priority operation in SAR mode – Continuous scan mode (3 groups) (2 of 2)

Step	Operation
8	The scanning operation of scan group 1 is resumed after scanning of scan group 0 is completed. The scanning operation is restarted from the A/D conversion incomplete channel. *1
9	When A/D conversion of each channel in scan group 1 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
10	If a scan end interrupt for scan group 1 is enabled, a scan end interrupt occurs.
11	Scan group 2 is resumed after scanning of scan group 1 is completed. The scanning operation is resumed from the A/D conversion incomplete channel. *1
12	When A/D conversion of each channel in scan group 2 completes, the A/D conversion result is stored in the corresponding A/D data register n (ADDRn).
13	If a scan end interrupt for scan group 2 is enabled, a scan end interrupt occurs.
14	Scan group 1 continues the continuous scanning operation. *2

Note 1. The timing of interruption and restart of scanning operation depends on the operating conditions. For details, see section 53.3.18.3. Restrictions on Group Priority Operation.

Note 2. To stop the continuous scanning operation, follow the procedures described in section 53.6.4. Force Stops the A/D Conversion Operation.



Note: SG: Scan Group VC: Virtual channel

Figure 53.30 Example of group priority operation in SAR mode – Continuous scan mode (3 groups)

53.3.18.3 Restrictions on Group Priority Operation

(1) Channel-dedicated sample-and-hold circuit restrictions

Channel-dedicated sample-and-hold circuit is available only for the highest priority scan group in group priority operation.

(2) Restrictions on the operation mode and operation setting of the A/D converter

Group priority operation is only available in SAR mode; Group priority operation is prohibited in Oversampling mode or Hybrid mode.

If these restrictions are violated, operation is not guaranteed.

(3) Restrictions on group priority operation in SAR mode – Continuous scan mode

In group priority operation in SAR mode – Continuous scan mode, do not input A/D conversion start triggers for scan groups that have a lower priority than the scan group that started continuous scanning operation. If this restriction is violated, the operation is not guaranteed.

(4) Restrictions on trigger input for the same scan group

Input the A/D conversion start triggers for the same scan group in group priority operation for the following periods.

[When synchronous operation is disabled (ADSYCR.ADSYDISm = 1)]

- 2 ADCLK + 2 PCLK cycles after the scan end flag is set *1

[When synchronous operation is enabled (ADSYCR.ADSYDISm = 0)]

- Synchronous operation period cycle × 2 after the scan end flag is set *1

Note: m = 0, 1

Note 1. ADSCANENDSR.SCENDFn = 1 (n = 0 to 8)

If A/D conversion start triggers are input without waiting for the above period, the A/D conversion start triggers may be lost. To accept A/D conversion start triggers reliably, input the trigger after the above period has elapsed.

(5) Restrictions on suspend and resume for scanning operation of low-priority groups

When an A/D conversion start trigger for the high priority group is input during scanning of the low priority group in group priority operation, the scanning operation of the low priority group is interrupted at the timing shown in the following:

[Scan interruption timing of low priority group]

- When synchronous operation is enabled
  - After the A/D conversion start trigger of the high-priority group is accepted, the scanning operation of the low-priority group is suspended after waiting until the next synchronous operation period timing.
- When synchronous operation is disabled
  - The scanning operation of the low-priority group is immediately suspended after the A/D conversion start trigger of the high-priority group is accepted.

If A/D conversion is being performed on channels for which A/D-converted value addition/averaging function is to be used at the timing of scan interruption of a low-priority group, the midway result of A/D-converted value addition/averaging value at that time is discarded. When the scanning operation of the low-priority group is resumed, A/D conversion starts over from the beginning of the number of times specified for A/D-converted value addition/averaging function.

If the scanning operation is suspended by group priority operation, ADGRSR.ACTGRn (n = 0 to 8) bit of the suspended scan group remains set to 1.

53.3.19 Synchronous Operation

Synchronous operation is the function that controls the operations of multiple A/D converters so that they are synchronized.

53.3.19.1 Synchronous Operation Examples

(1) Basic synchronous operation

The basic operation of synchronous operation is shown in the following. The configuration example is shown in Table 53.42, and the detail of operation is shown in Table 53.43 and Figure 53.31.

Table 53.42 Configuration example of basic synchronous operation (1 of 2)

Item	Configuration
A/D converter	• ADC0: Set to SAR mode – Single scan mode • ADC1: Set to SAR mode – Single scan mode
Virtual channel	• Virtual channel 0 to 2: Assign AN000 to AN002 and set to Single-ended input mode. • Virtual channel 6 to 8: Assign AN006 to AN008 and set to Single-ended input mode.

Table 53.42 Configuration example of basic synchronous operation (2 of 2)

Item	Configuration
Scan group	[Scan Group 0] ● Assign AN000 to AN002 (Virtual channel 0 to 2) ● Set to convert with ADC0 [Scan Group 1] ● Assign AN006 to AN008 (Virtual channel 6 to 8) ● Set to convert with ADC1

Table 53.43 Example of basic synchronous operation

Step	Detail behavior of example operation
1	When a trigger of scan group 0 is input, a wait is inserted until the A/D conversion synchronization timing.
2	At the timing of synchronous operation period, the trigger input of scan group 0 is accepted, and the scanning operation of scan group 0 (for ADC0) starts.
3	When a trigger of scan group 1 is input, a wait is inserted until the A/D conversion synchronization timing.
4	At the timing of synchronous operation period, the trigger input of scan group 1 is accepted, and the scanning operation of scan group 1 (for ADC1) starts.
5	During synchronous operation, the A/D converter starts the sampling operation in accordance with synchronous operation period.

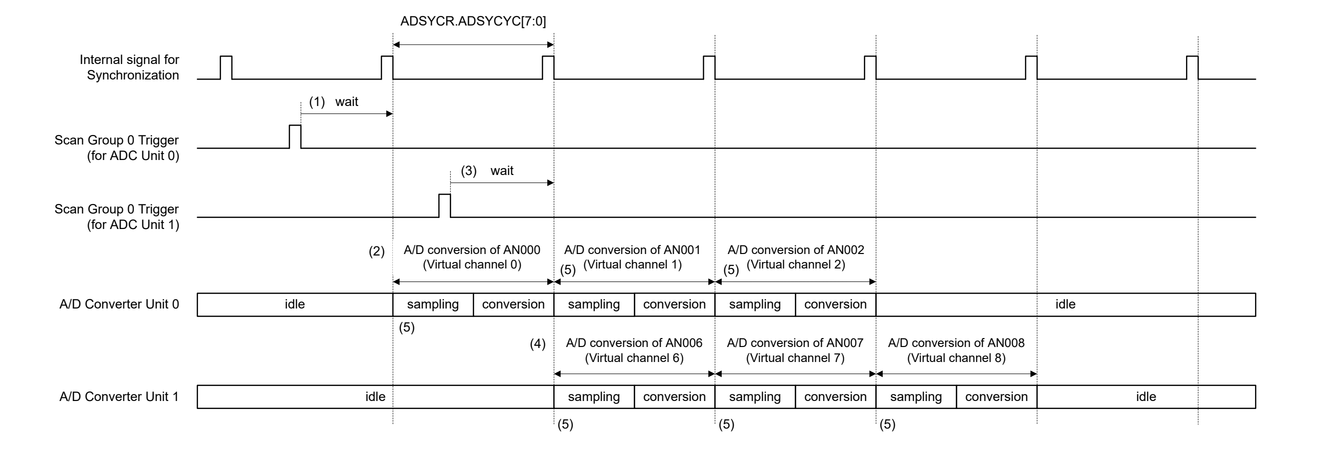


Figure 53.31 Example of synchronous operation in SAR mode – Single scan mode

(2) Synchronous operation using with channel-dedicated sample-and-hold circuit

An example of synchronous operation using with channel-dedicated sample-and-hold circuit is shown in the following. The configuration example is shown in Table 53.44, and the detail of operation is shown in Table 53.45 and Figure 53.32.

Table 53.44 Configuration example of basic synchronous operation (1 of 2)

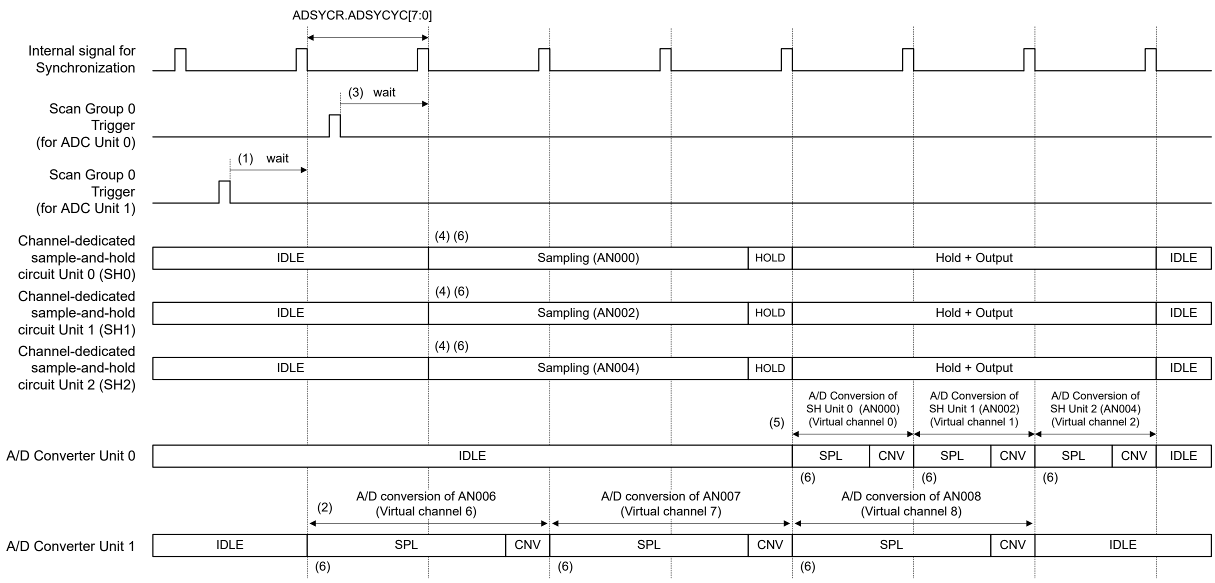
Item	Configuration
A/D converter	● ADC0: Set to SAR mode – Single scan mode ● ADC1: Set to SAR mode – Single scan mode
Channel-dedicated sample-and-hold circuit	● Set SH0 to SH2 to enabled and Single-ended input mode. ● Set SH4 to SH6 to enabled and Single-ended input mode.
Virtual channel	● Virtual channel 0 to 2: Assign AN000, AN002 and AN004, and set to Single-ended input mode. ● Virtual channel 6 to 8: Assign AN006, AN008 and AN010, and set to Single-ended input mode.

Table 53.44 Configuration example of basic synchronous operation (2 of 2)

Item	Configuration
Scan group	[Scan Group 0] ● Assign AN000, AN002 and AN004 (Virtual channel 0 to 2) ● Set to convert with ADC0 [Scan Group 1] ● Assign AN006, AN008 and AN010 (Virtual channel 6 to 8) ● Set to convert with ADC1

Table 53.45 Example of synchronous operation with channel-dedicated sample-and-hold circuit

Step	Detail behavior of example operation
1	When a trigger of scan group 1 is input, a wait is inserted until the A/D conversion synchronization timing.
2	At the timing of synchronous operation period, the trigger input of scan group 1 is accepted, and the scanning operation of scan group 1 (for ADC1) starts.
3	When a trigger of scan group 0 is input, a wait is inserted until the A/D conversion synchronization timing.
4	At the timing of synchronous operation period, the trigger input of scan group 0 is accepted, and the scanning operation of scan group 0 (for ADC0) starts. Channel-dedicated sample-and-hold circuit (SH0 to SH2) samples and holds the analog-channels of AN000/AN002/AN004 at the beginning of the scanning operation in scan group 0.
5	After the holding by channel-dedicated sample-and-hold circuit, A/D converter starts the samplings and the conversions.
6	During synchronous operation, channel-dedicated sample-and-hold circuit and A/D converter starts the sampling operation in accordance with synchronous operation period.



Note: SPL: Sampling state CNV: Conversion state

Figure 53.32 Example of synchronous operation using with channel-dedicated sample-and-hold circuit

53.3.19.2 Restrictions on Synchronous Operation

When synchronous operation is used, observe the following restrictions. If the following restrictions are violated, operations are not guaranteed.

1. Basic restrictions
    - Set the synchronous operation period to a value larger than successive approximation time of ADCm (m = 0, 1).
        - ADSYCR.ADSYCYC[7:0] ≥ ADCNVSTR.CSTm[5:0] + 1
          (m = 0, 1)

- Set so that the sum of the sampling times of the analog channels and successive approximation time of ADCm (m = 0, 1) is an integral multiple of the synchronous operation period.
  - ADSSTRx.SSTy[9:0] + ADCNVSTR.CSTm[5:0] = ADSYCR.ADSYCYC[7:0] × i
    (x = 0 to 7, y = 0 to 15, m = 0, 1, i: Any integer greater than or equal to 1 (i = 1, 2, 3…))
- Set the synchronous operation period to a value larger than successive approximation time at self-calibration.*1
  - ADSYCR.ADSYCYC[7:0] ≥ ADCALSTCR.CALADCST[5:0] + 1
- Set the sampling time and successive approximation time at self-calibration to be integral multiples of the synchronous operation period.*1
  - ADCALSTCR.CALADSST[9:0] + ADCALSTCR.CALADCST[5:0] = ADSYCR.ADSYCYC[7:0] × i
    (i: Any integer greater than or equal to 1 (i = 1, 2, 3…))

2. Restrictions when using channel-dedicated sample-and-hold circuit
   - Set the synchronous operation period to a value greater than the hold mode switching time of channel-dedicated sample-and-hold circuit.
     - ADSYCR.ADSYCYC[7:0] ≥ ADSHSTRm.SHHST[2:0] + 1
       (m = 0, 1)
   - Set the sum of channel-dedicated sample-and-hold circuit sampling time and the hold mode switching time to be an integral multiple of the synchronous operation period.
     - ADSHSTRm.SHSST[7:0] + ADSHSTRm.SHHST[2:0] = ADSYCR.ADSYCYC[7:0] × i
       (m = 0, 1, i: Any integer greater than or equal to 1 (i = 1, 2, 3…))
   - Set the hold mode switching time of channel-dedicated sample-and-hold circuit so that it is the same as successive approximation time of ADCm (m = 0, 1).
     - ADSHSTRm.SHHST[2:0] = ADCNVSTR.CSTm[5:0]
       (m = 0, 1)
   - Set the synchronous operation period to a value greater than the hold mode switching time in channel-dedicated sample-and-hold circuit's self-calibration operation.*1
     - ADSYCR.ADSYCYC[7:0] ≥ ADCALSHCR.CALSHHST[2:0] + 1
   - Set so that the sum of channel-dedicated sample-and-hold circuit sampling time and the hold mode switching time in self-calibration operation is an integral multiple of the synchronous operation period.*1
     - ADCALSHCR.CALSHSST[7:0] + ADCALSHCR.CALSHHST[2:0] = ADSYCR.ADSYCYC[7:0] × i
       (i: Any integer greater than or equal to 1 (i = 1, 2, 3…))
3. Restrictions when using disconnection detection assist function
   - Set Disconnection Detection Assist period so that it is the same as the synchronous operation period.
     - ADSGDCRn.ADNDIS[7:0] = ADSYCR.ADSYCYC[7:0]
       (n = 0 to 8)
4. Restriction when using high accuracy mode
   - Set so that the sum of the sampling times of the analog channels and successive approximation time of ADCm (m = 0, 1) is an even or equal multiple of the synchronous operation period.
     - ADSSTRx.SSTy[9:0] + ADCNVSTR.CSTm[5:0] = ADSYCR.ADSYCYC[7:0] × i
       (x = 0 to 7, y = 0 to 15, m = 0, 1, i: Any even number or equal to 1 (i = 1, 2, 4, 6…))

Note 1. In relation to this restriction, the restrictions on the settings of self-calibration should be satisfied at the same time. For restrictions on the setting of self-calibration, see section 53.3.8.3. Restrictions on Self-calibration.

53.4 A/D Conversion Monitor Signals

This section describes the internal operation of A/D conversion.

The A/D conversion monitor signals are output from pins that are configurable by the function of I/O ports.

The operating timing of A/D conversions can be checked by observing these signals with an oscilloscope.

The following A/D conversion monitor signals are available.

ADSYNC

ADSYNC is a reference signal of the synchronous operation of A/D conversion.

This signal indicates the synchronous timing of A/D conversion. When the synchronous operation function is enabled, the A/D converter operation is executed in synchronization with this signal. When the synchronous operation function is disabled, this signal is ignored. See section 53.3.19. Synchronous Operation for more details about synchronous operation of A/D conversion.

ADnFLAG1 (n = 0,1)

ADnFLAG1 indicates the period of the conversion of A/D converter unit n. While this signal is asserted, the A/D converter is performing the conversion operation.

ADSTn (n = 0,1)

ADSTn indicates the period of the scanning operation of A/D converter unit n. This signal is asserted during the execution of the A/D scanning operation.

Figure 53.33 shows the operation timing of each signal. See section 20, I/O Ports for information on how to select output signals.

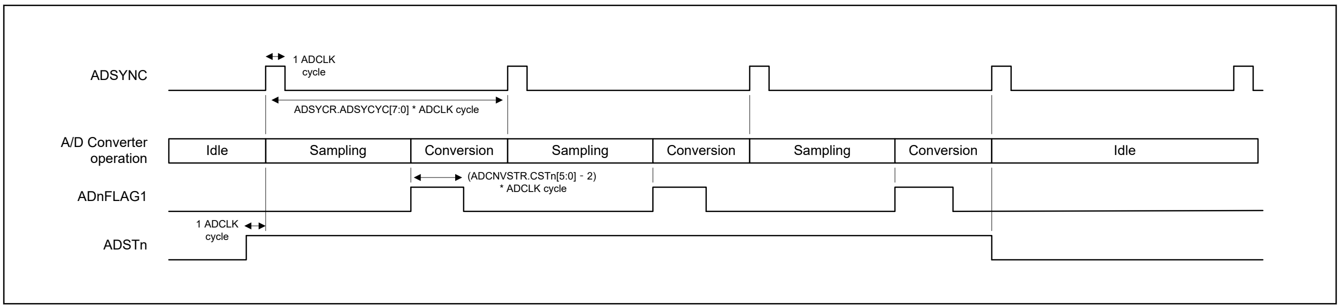


Figure 53.33 A/D conversion monitor signals

53.5 A/D Conversion Data

This section describes the internal processing for A/D conversion data. For details about the operation of A/D conversion, see section 53.3. Operation.

53.5.1 Internal Data Processing Flow

Figure 53.34 shows the internal processing-flow of A/D conversion data.

The A/D conversion data output from the A/D converter is temporarily stored in the data buffer. And then, various data processing is performed based on the register setting. If the function that corresponds to each data processing is not used, its data processing is skipped.

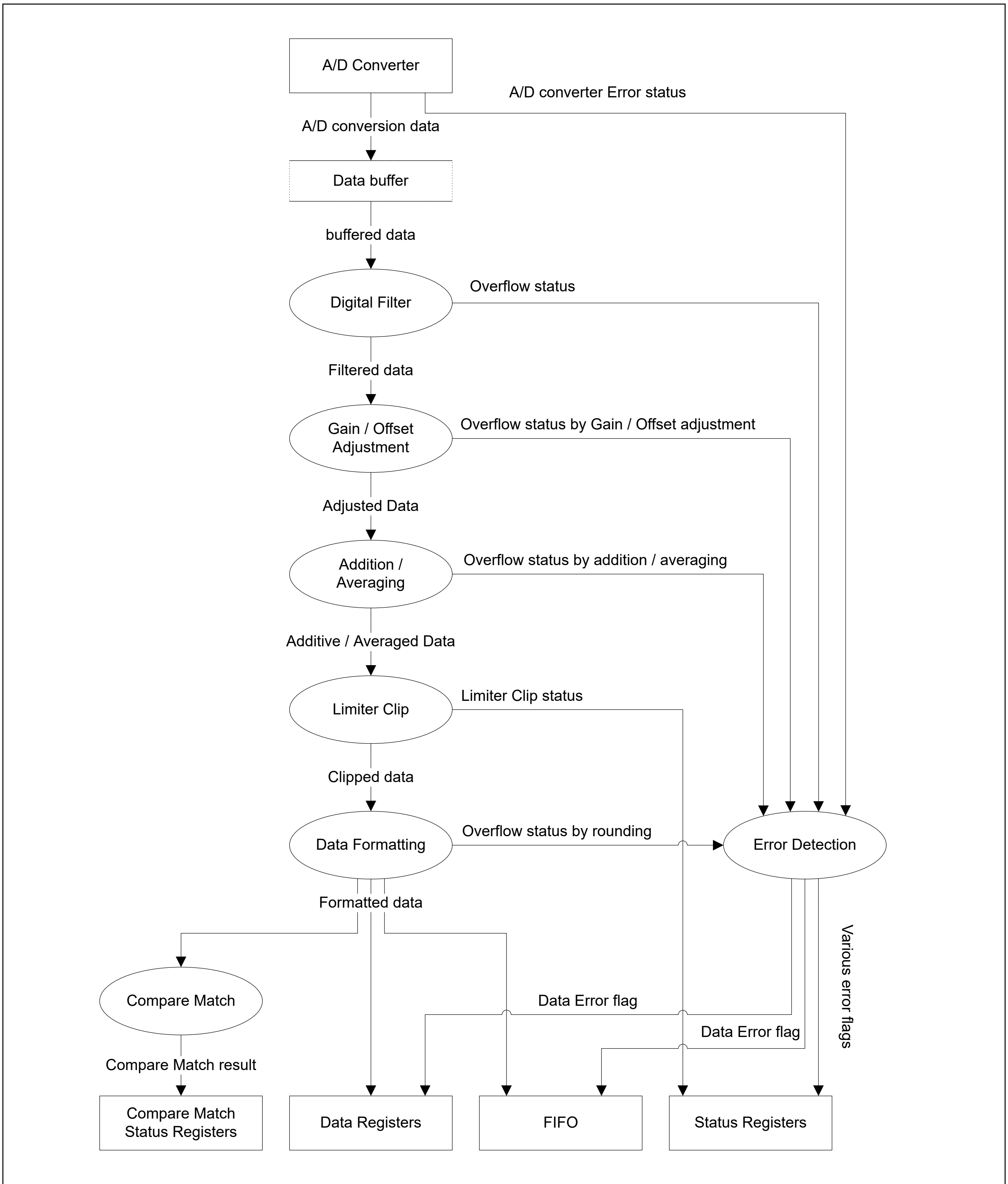


Figure 53.34 Internal data flow diagram of A/D conversion data

53.5.2 Digital Filter Function

ADC16H has 4 digital filters per 1 unit of A/D converter. High-precision A/D conversion results can be obtained by using a digital filter. The digital filter function must be used in Oversampling mode or Hybrid mode. The digital filter function in SAR mode is prohibited.

53.5.2.1 Configuration and Characteristics

The digital filter consists of a FIR type filter with 22 taps. The digital filter is set by ADDOPCRAn.DFSEL[2:0] (n = 0 to 32) and ADDFSRm (m = 0, 1) registers. The configuration of the digital filter is shown in Figure 53.35.

The type of the digital filter (FIR filter) can be selected from the following. For the characteristics of the digital filters, see section 62, Electrical Characteristics.

[List of digital filter characteristics]

- Sinc filter

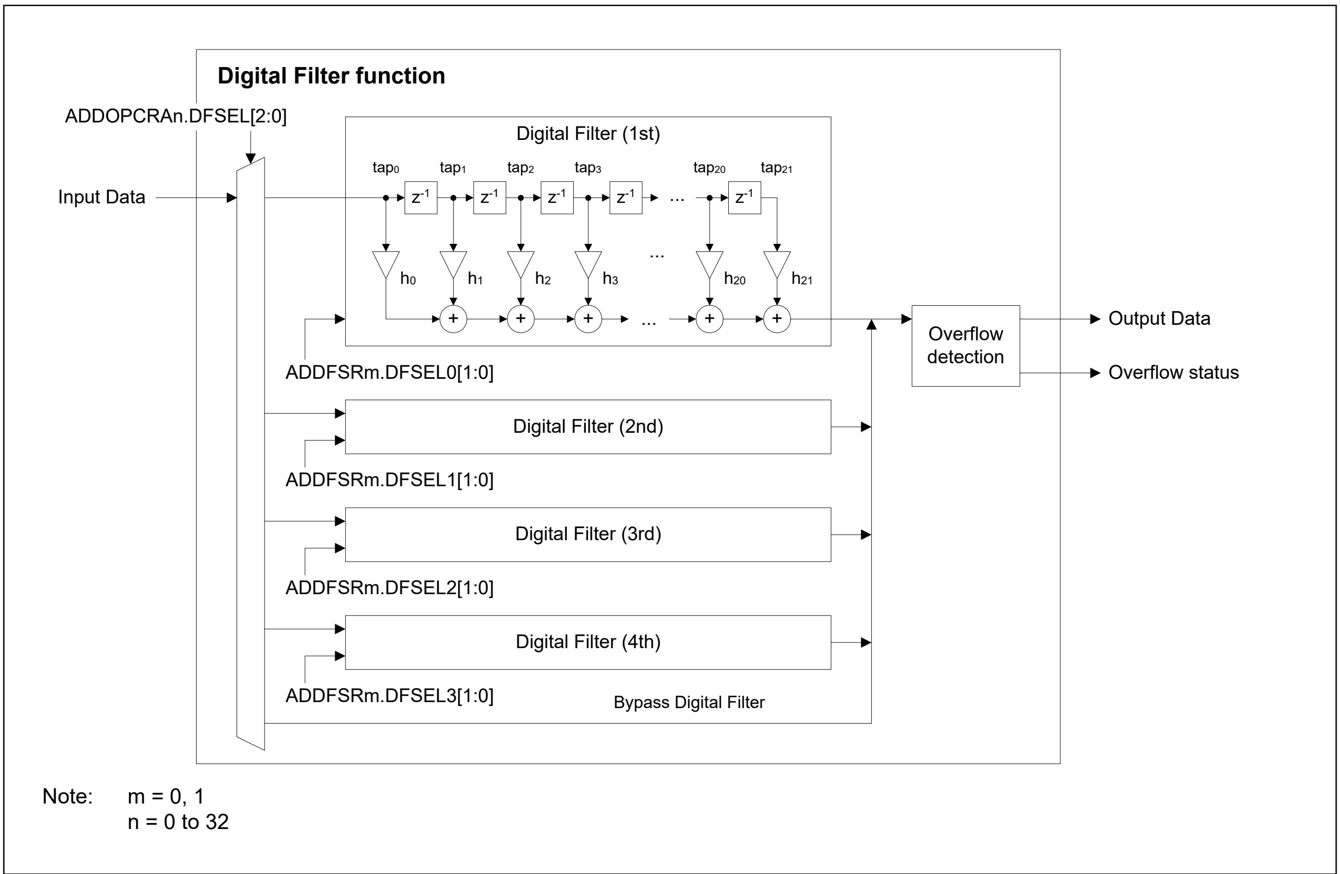


Figure 53.35 Block diagram of digital filter

53.5.2.2 Operation of Digital Filter Function

When the digital filter function is enabled, A/D conversion data is sequentially input to the digital filter. When all the taps in the digital filter are filled with data, the calculation result is output and sent to the next data processing.

The operation of the digital filters in Oversampling mode and Hybrid mode is shown in the followings.

(1) Digital filter operation in Oversampling mode

In Oversampling mode, oversampling (A/D conversion) is performed continuously for one analog channel. Each time oversampling is performed, A/D conversion data is sequentially input to the digital filter. When all the taps in the digital filter are filled with data, the calculation result is output and sent to the next data processing.

In Oversampling mode, the data of the taps in the digital filter is discarded when the A/D conversion data is output to the next data processing. However, when A/D-converted value addition/averaging function is used, the data of taps in the digital filter is kept until the data required to calculate the addition value or average value of the A/D converted value is collected. When the A/D-converted value addition/averaging value is calculated, the tap data in the digital filter is discarded.

(2) Digital filter operation in Hybrid mode

Hybrid mode can process oversampled data (A/D converted data) of up to four analog channels in parallel by using several digital filter circuits at the same time. Each time oversampling is performed, A/D conversion data is sequentially input to the digital filter. When all the taps in the digital filter are filled with data, the calculation result is output and sent to the next data processing.

In Hybrid mode – Single scan mode, taps in the digital filter are discarded at the end of scanning operation.

In Hybrid mode – Continuous scan mode, the data of the taps in the digital filter continues to be updated as long as the continuous scanning operation is continued. Therefore, after the data is filled in all taps in the digital filter, a new calculation result is output for each oversampling. When scanning operation is aborted due to a forced stop of A/D conversion, the data of taps in the filter is discarded.

The operation of the digital filter in Hybrid mode – Background continuous scan mode is the same as in Hybrid mode – Continuous scan mode. During the background continuous scanning operation, the digital filters and other data processing continue to take place in the background. When scanning operation is aborted due to a forced stop of A/D conversion, the data of taps in the filter is discarded.

53.5.3 Calibration and Adjustment

In this process, the following processing is performed for the results output from the A/D converter or for the data output from Digital Filter.

1. Calibrate Gain Error and Offset Error
2. Adjust User's Gain
3. Adjust User's Offset

For details on each process, refer to each item.

53.5.3.1 Gain Error and Offset Error Calibration

This process calibrates the A/D converter's Gain Error and Offset Error due to chip-by-chip characteristic variations.

Calibration for the A/D converter's Gain Error and Offset Error is performed by calculation based on error data measured inside the ADC16H by self-calibration operation.

Using gain error and offset error calibration when the input to the A/D converter is near the VREFH0 (unit 0) and VREFH (unit 1) upper limit or VREFL0 (unit 0) and VREFL (unit 1) lower limit, an overflow of A/D conversion data may occur.

53.5.3.2 User's Gain Adjustment

User's Gain adjusting function multiplies the A/D conversion data by an arbitrary coefficient value.

The coefficient value of User's Gain is set in ADUGTRn.UGAIN[15:0] (n = 0 to 7). User's Gain can be set for each virtual channel. The User's Gain Table to be used is selected in ADDOPCRAm.GAINSEL[3:0] (m = 0 to 32). The coefficient value of User's Gain is the sum of the gain corresponding to the bits that are set to 1 in ADUGTRn.UGAIN[15:0]. Table 53.46 shows the gain factors corresponding to the respective bit in ADUGTRn.UGAIN[15:0]. An example of setting User's Gain is shown in Table 53.47. Figure 53.36 and Figure 53.37 show the relationship between before adjustment (input) and after adjustment (output) to the A/D conversion result when User's Gain adjustment function is used.

The overflow of A/D conversion data may occur when User's Gain adjusting function is used.

Table 53.46 List of gains corresponding to each gain setting bit in User's Gain setting table register

UGAIN[15:8]	Gain value	UGAIN[7:0]	Gain value
b15	2^{1} = 2.0	b7	2^{-7} = 7.813E-03
b14	2^{0} = 1.0	b6	2^{-8} = 3.906E-03
b13	2^{-1} = 0.5	b5	2^{-9} = 1.953E-03
b12	2^{-2} = 0.25	b4	2^{-10} = 9.766E-04
b11	2^{-3} = 0.125	b3	2^{-11} = 4.883E-04
b10	2^{-4} = 0.0625	b2	2^{-12} = 2.441E-04
b9	2^{-5} = 0.03125	b1	2^{-13} = 1.221E-04
b8	2^{-6} = 1.563E-02	b0	2^{-14} = 6.104E-04

Table 53.47 Example of User's Gain Settings

ADUGTRn.UGAIN[15:0] (n = 0 to 7)	Gain value
0x0000	x0.0000
⋮	⋮
0x0400	x0.0625
⋮	⋮
0x0800	x0.1250
⋮	⋮
0x1000	x0.2500
⋮	⋮
0x2000	x0.5000
⋮	⋮
0x4000 (initial value)	x1.0000
⋮	⋮
0x8000	x2.0000
⋮	⋮
0xFFFF	x3.9999

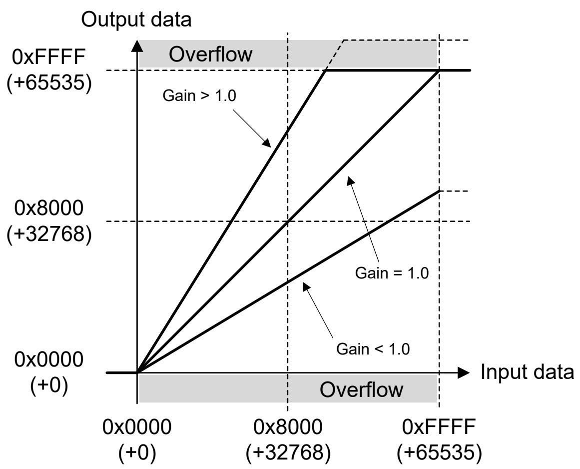


Figure 53.36 User's Gain Adjustment (16-bit data length format and Unsigned data format)

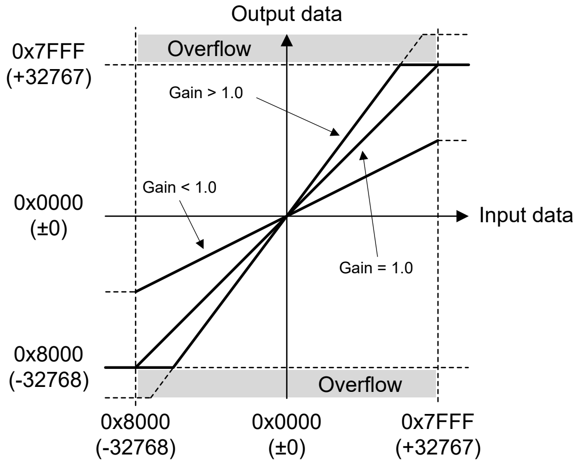


Figure 53.37 User's Gain Adjustment (16-bit data length format and Signed data format)

53.5.3.3 User's Offset Adjustment

User's Offset adjustment function adds or subtracts any constant value to or from the A/D conversion data.

The constant value of User's Offset is set in ADUOFTRn.UOFSET[15:0] (n = 0 to 7). User's Offset can be set for each virtual channel. User's Offset Table to be used is selected in ADDOPCRAm.OFSETSEL[3:0] (m = 0 to 32). Table 53.48 shows the relationship between the register setting value of User's Offset and the offset value. Figure 53.38 and Figure 53.39 show the relationship between before adjustment (input) and after adjustment (output) to the A/D conversion result when User's Offset adjustment function is used.

When User's Offset adjusting function is used, overflow of A/D conversion data may occur.

Table 53.48 Relationship between User's Offset register value and offset value

ADUOFTRn.UOFSET[15:0] (n = 0 to 7)	Offset value (16bit data length format)
0x7FFF	+32767
0x7FFE	+32766
0x7FFD	+32765
⋮	⋮
0x0003	+3
0x0002	+2
0x0001	+1
0x0000 (initial value)	0
0xFFFF	-1
0xFFFE	-2
0xFFFD	-3
⋮	⋮
0x8002	-32766
0x8001	-32767
0x8000	-32768

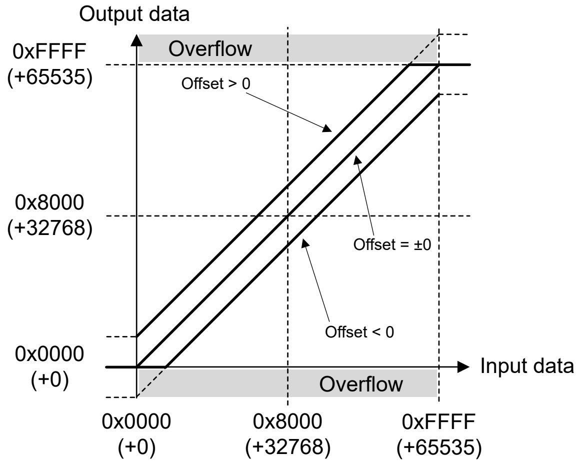


Figure 53.38 User's Offset Adjustment (16-bit data length format and Unsigned data format)

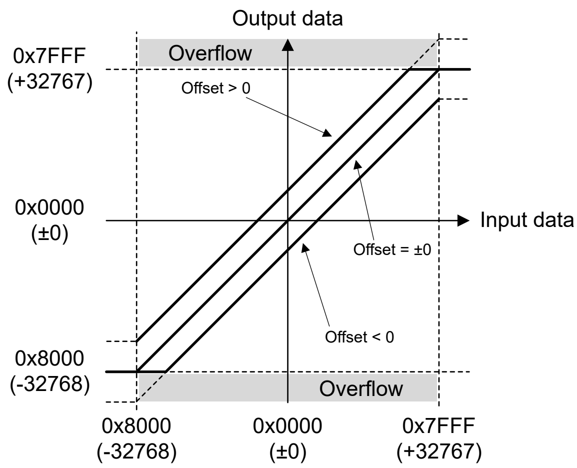


Figure 53.39 User's Offset Adjustment (16-bit data length format and Signed data format)

53.5.4 A/D-converted Value Addition/Averaging Function

A/D-converted value addition/averaging function is a function to calculate the sum value or average value of the result of A/D conversion of analog channels continuously for the specified number of times. A/D-converted value addition/averaging function can be configured for each virtual channel in ADDOPCRBn.AVEMD[1:0] bits and ADDOPCRBn.ADC[3:0] bits (n = 0 to 32).

A/D-converted value addition/averaging function inputs the output of the previous data processing. When the specified number of times of data is input, the added value or average value is calculated, and the result is output to the next data processing. When A/D-converted value addition/averaging function is used, the A/D converter repeats the A/D conversion until the data is collected for the number of times specified by the ADDOPCRBn.ADC[3:0] (n = 0 to 32) bits. One calculation result (total value or average value) is output every time the specified number of times of data is input to A/D-converted value addition/averaging. Figure 53.40 shows the operation of A/D-converted value addition/averaging function in SAR mode or Oversampling mode. Figure 53.41 shows the operation of A/D-converted value addition/averaging function in Hybrid mode.

When the A/D-converted value addition/averaging function is used, overflow of the A/D conversion data may occur. However, A/D conversion overflow may not be detected under certain conditions. For more information on A/D conversion overflow, see section 53.7.2. A/D Conversion Overflow.

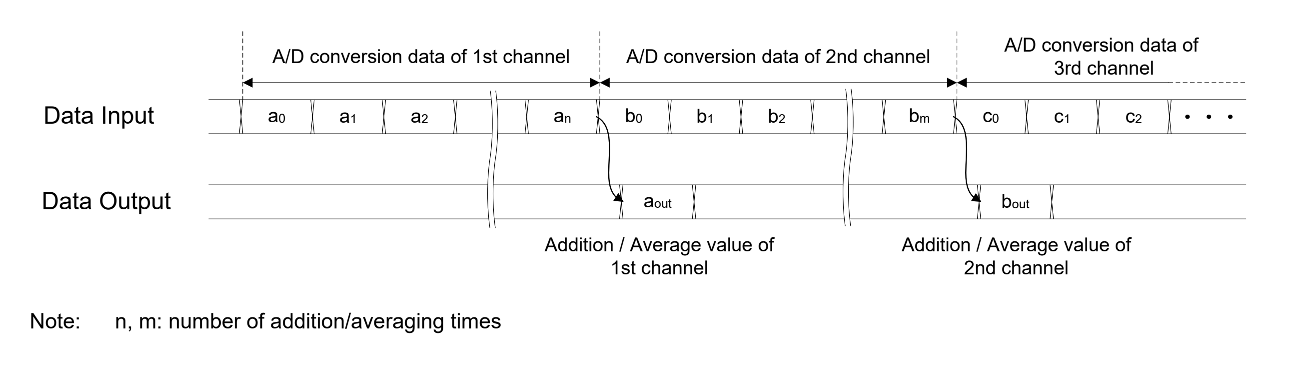


Figure 53.40 Example operation of A/D-converted value addition/averaging function (SAR mode or Oversampling mode)

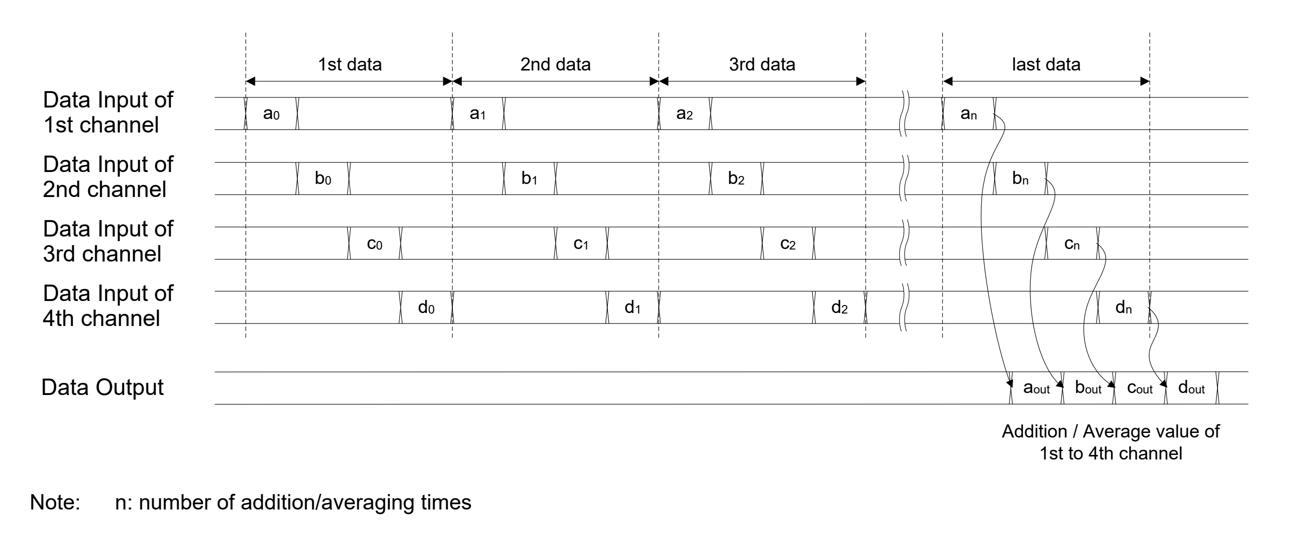


Figure 53.41 Example operation of A/D-converted value addition/averaging function (Hybrid mode)

53.5.5 Limiter Clip Function

Limiter Clip Function is the function to set the upper and lower limits of A/D conversion data. When the A/D conversion data exceeds the specified upper limit value, it is clipped to the upper limit value. If the A/D conversion data falls below the specified lower limit, it is clipped to the lower limit value. Figure 53.42 shows an example of Limiter Clip Function operation.

The upper and lower limits of Limiter Clip Function can be specified in ADLIMTRn.LIMU[15:0] (n = 0 to 7) and ADLIMTRn.LIML[15:0] (n = 0 to 7). The Limiter Clip Function can be selected use or not (enable or disable) for each virtual channel in ADDOPCRCm.LIMTBLS[3:0] (m = 0 to 32).

Limiter Clip Function is processed by 16-bit length. When 14-/12-/10-bit is selected as the data length of the A/D conversion data (when ADDOPCRCm.ADPRC[1:0] = 01b, 10b, or 11b (m = 0 to 32)), the rounding to the specified data length is performed after the clipping by Limiter Clip Function. For details, see section 53.5.6. Data Formatting Process.

The upper and lower limits of the limiter clip are treated as signed or unsigned depending on the setting of the sign selection bit (ADDOPCRCm.SIGNSEL (m = 0 to 32)) in the A/D conversion data.

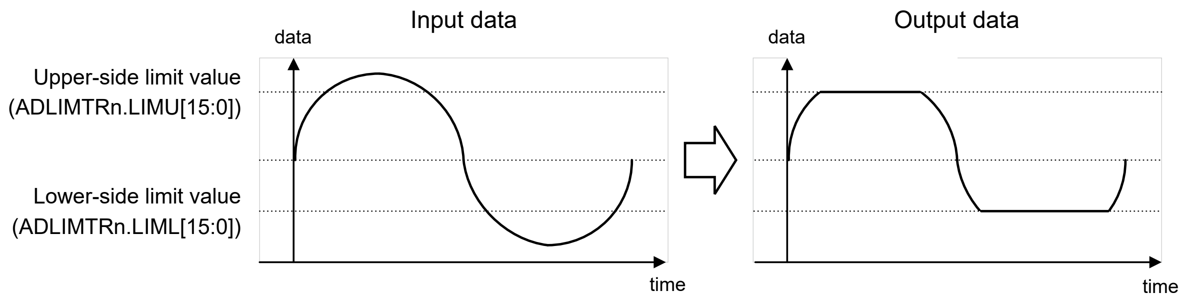
For Limiter Clip Function, the upper limit value should be greater than the lower limit value (ADLIMTRn.LIMU[15:0] > ADLIMTRn.LIML[15:0] (n = 0 to 7)). If the upper limit value is less than or equal to the lower limit value, the A/D conversion data always becomes 0x0000.

If a limiter clip occurs, the flags are set in the following status registers:

- ADLIMGRSR: The flag is set to the bit corresponding to the scan group where the limiter clip occurred.

- ADLIMCHSR0: If a limiter clip occurs during A/D conversion of an analog input channel, the flag is set to the corresponding bit.
- ADLIMEXSR: If a limiter clip occurs during A/D conversion of Extended Analog Function, the flag is set to the corresponding bit.

To clear the flags in the status registers, write 1 to the corresponding bit in ADLIMGRSCR, ADLIMCHSCR0, ADLIMEXSCR.



Note: n = 0 to 7

Figure 53.42 Example of limiter clip function operation

53.5.6 Data Formatting Process

In this process, the following formatting is performed on the A/D conversion data.

- Signed/Unsigned data processing
- Data Rounding process

The data formatted in this process is stored to DATA[15:0] bits of the A/D data register (ADDRi (i = 0 to 22)), the extended A/D data register (ADEXDRj (j = 0, 1, 4, 5, 6, 8, 9, 16, 17, 18, 20, 21, 22)), and the FIFO data register (ADFIFODRk (k = 0 to 8)) as the result of A/D conversion.

For details on the data format, see section 53.5.7. Data Format.

53.5.6.1 Signed/Unsigned Data Process

The A/D conversion data is formatted into Signed or Unsigned data format based on the setting of the ADDOPCRCn.SIGNSEL (n = 0 to 32) bits. Select the signed data format (SIGNSEL = 0) for Differential input mode and the unsigned format (SIGNSEL = 1) for Single-ended input mode.

53.5.6.2 Data Rounding Process

The A/D conversion data is processed as a fraction of the data length based on the setting of the ADDOPCRCn.ADPRC[1:0] (n = 0 to 32) bits. If 14-/12-/10-bit data format is selected (ADPRC[1:0] = 01b, 10b, 11b), the lower bits of the A/D conversion data will be rounding. (If the most significant bit of the digit to be cut off is 0, the digit is rounded down; if it is 1, the digit is rounded up.) If the 16-bit data format is selected (ADPRC[1:0] = 00b), the A/D conversion data is not rounded.

53.5.7 Data Format

This subsection describes the data format of the A/D conversion result stored to DATA[15:0] bits in A/D data register (ADDRi (i = 0 to 22)), Extended A/D data register (ADEXDRj (j = 0, 1, 4, 5, 6, 8, 9, 16, 17, 18, 20, 21, 22)), and FIFO data register (ADFIFODRk (k = 0 to 8)).

Note: The data format does not guarantee the resolution and accuracy of the A/D converter itself. See section 62, Electrical Characteristics for the characteristics of the A/D converter.

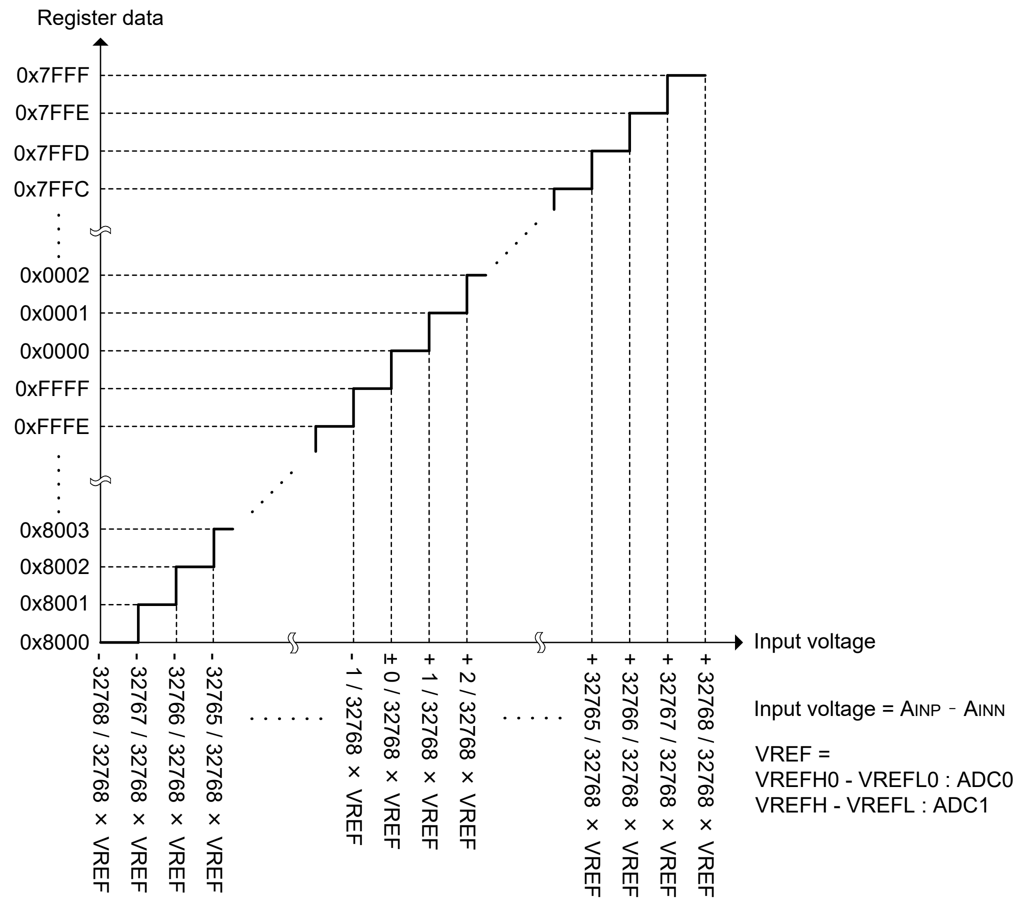
(1) 16-bits data length format

Table 53.49 shows the alignment of A/D conversion data on 16-bit data format. Figure 53.43 and Figure 53.44 show the data range for 16-bit data format.

For 16-bit data length signed data format, the differential input voltage is A/D-converted in the range of 0x8000 to 0x7FFF. This range corresponds to the voltage range of -VREFH0 to +VREFH0 for ADC0 or the voltage range of -VREFH to +VREFH for ADC1. In 16-bit data length unsigned data format, the single-ended input voltage is A/D-converted in the range of 0x0000 to 0xFFFF. This range corresponds to the voltage range of VREFL0 to VREFH0 for ADC0 or the voltage range of VREFL to VREFH for ADC1.

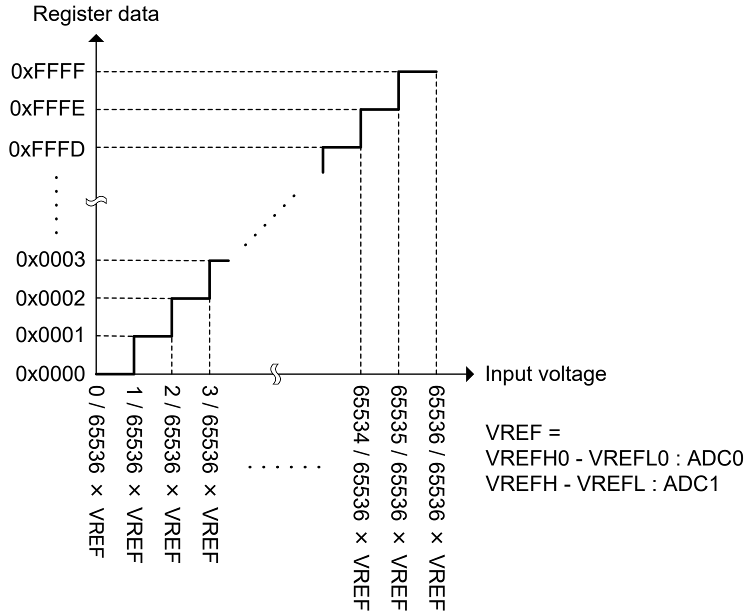
Table 53.49 Alignment of A/D conversion result data (16-bit data length)

Bit position:	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Bit field:	DATA [15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R



Note: This figure illustrates the case of the ideal A/D conversion characteristics.

Figure 53.43 Data range of A/D conversion result (16-bit data length, Signed data, Differential input)



Note: This figure illustrates the case of the ideal A/D conversion characteristics.

Figure 53.44 Data range of A/D conversion result (16-bit data length, Unsigned data, Single-ended input)

(2) 14-bits data length format

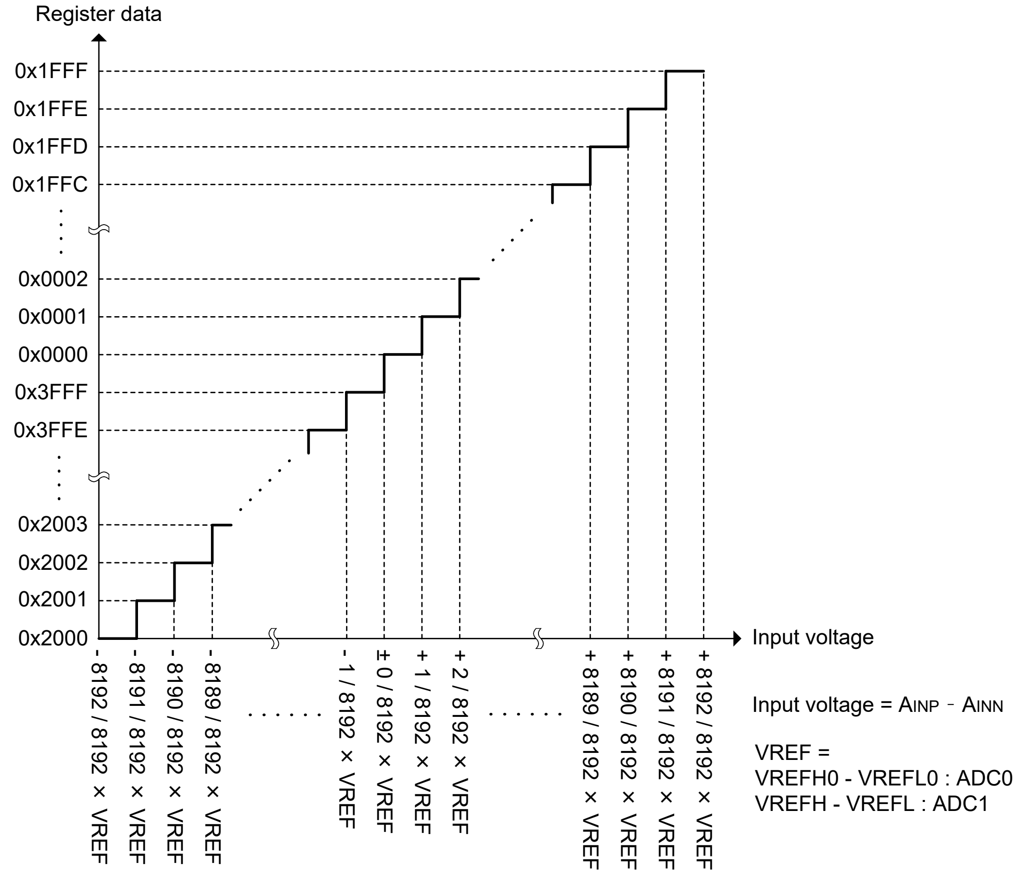
Table 53.50 shows the alignment of A/D conversion data on 14-bit data format. Figure 53.45 and Figure 53.46 show the data range for 14-bit data format.

For 14-bit data length signed data format, the differential input voltage is A/D-converted in the range of 0x2000 to 0x1FFF as the range of -VREFH0 to +VREFH0 for ADC0 or the range of -VREFH to +VREFH for ADC1. In 14-bit data length unsigned data format, the single-ended input voltage is A/D-converted in the range of 0x0000 to 0x3FFF as the range of VREFL0 to VREFH0 for ADC0 or the range of VREFL to VREFH for ADC1.

For 14-bit data length, the upper 2 bits (bit15 to bit14) are always 0.

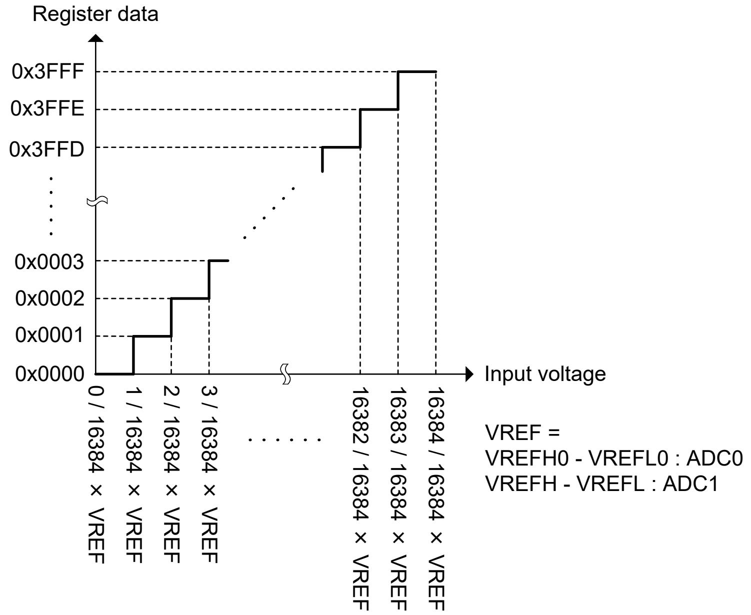
Table 53.50 Alignment of A/D conversion result data (14-bit data length)

Bit position:	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Bit field:	0	0	DATA[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R



Note: This figure illustrates the case of the ideal A/D conversion characteristics.

Figure 53.45 Data range of A/D conversion result (14-bit data length, Signed data, Differential input)



Note: This figure illustrates the case of the ideal A/D conversion characteristics.

Figure 53.46 Data range of A/D conversion result (14-bit data length, Unsigned data, Single-ended input)

(3) 12-bits data length format

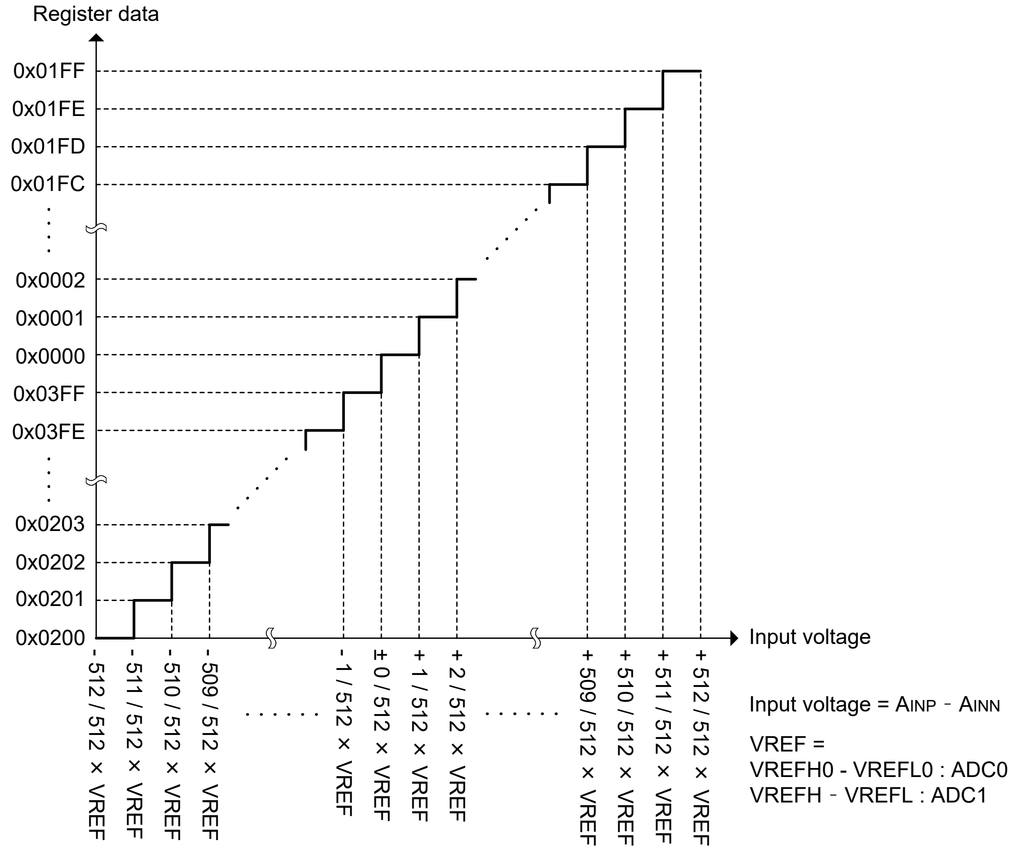
Table 53.51 shows the alignment of A/D conversion data on 12-bit data format. Figure 53.47 and Figure 53.48 show the data range for 12-bit data format.

For 12-bit data length signed data format, the differential input voltage is A/D-converted in the range of 0x0800 to 0x07FF as the range of -VREFH0 to +VREFH0 for ADC0 or the range of -VREFH to +VREFH for ADC1. In 12-bit data length unsigned data format, the single-ended input voltage is A/D-converted in the range of 0x0000 to 0x0FFF as the range of VREFL0 to VREFH0 for ADC0 or the range of VREFL to VREFH for ADC1.

For 12-bit data length, the upper 4 bits (bit15 to bit12) are always 0.

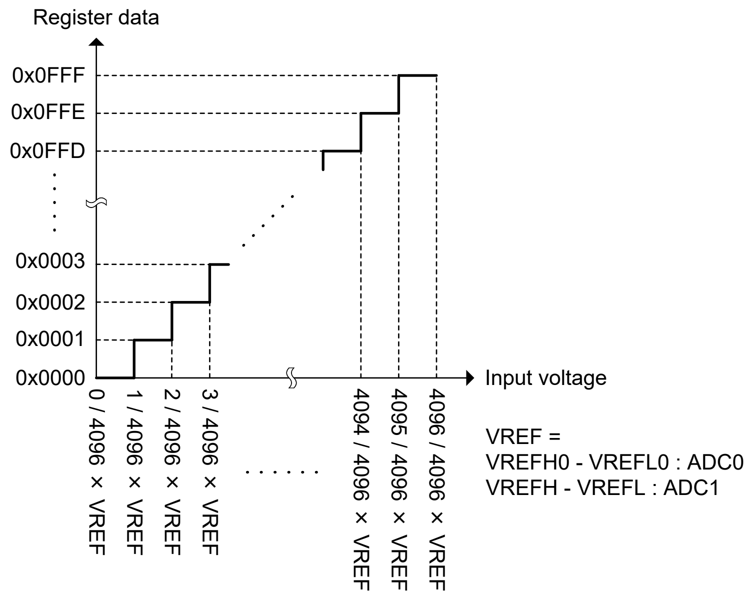
Table 53.51 Alignment of A/D conversion result data (12-bit data length)

Bit position:	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Bit field:	0	0	0	0	DATA[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R



Note: This figure illustrates the case of the ideal A/D conversion characteristics.

Figure 53.47 Data range of A/D conversion result (12-bit data length, Signed data, Differential input)



Note: This figure illustrates the case of the ideal A/D conversion characteristics.

Figure 53.48 Data range of A/D conversion result (12-bit data length, Unsigned data, Single-ended input)

(4) 10-bits data length format

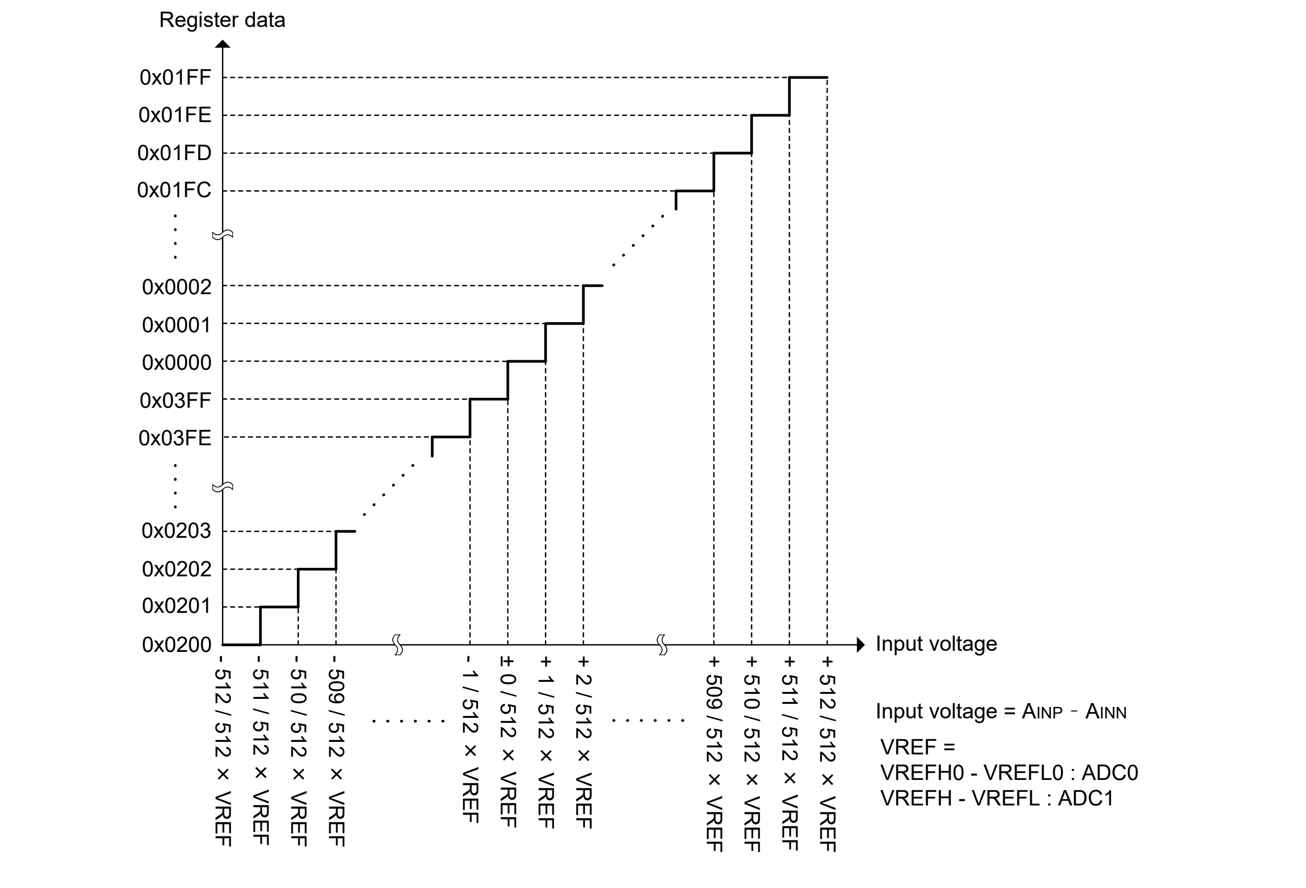
Table 53.52 shows the alignment of A/D conversion data on 10-bit data format. Figure 53.49 and Figure 53.50 show the data range for 10-bit data format.

For 10-bit data length signed data format, the differential input voltage is A/D-converted in the range of 0x0200 to 0x01FF as the range of -VREFH0 to +VREFH0 for ADC0 or the range of -VREFH to +VREFH for ADC1. In 10-bit data length unsigned data format, the single-ended input voltage is A/D-converted in the range of 0x0000 to 0x03FF as the range of VREFL0 to VREFH0 for ADC0 or the range of VREFL to VREFH for ADC1.

For 10-bit data length, the upper 6 bits (bit15 to bit10) are always 0.

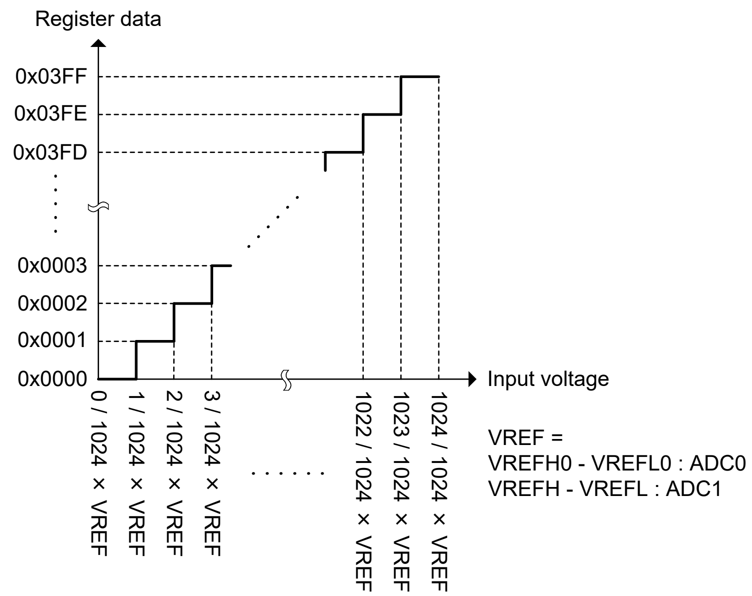
Table 53.52 Alignment of A/D conversion result data (10-bit data length)

Bit position:	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Bit field:	0	0	0	0	0	0	DATA [9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R



Note: This figure illustrates the case of the ideal A/D conversion characteristics.

Figure 53.49 Data range of A/D conversion result (10-bit data length, Signed data, Differential input)



Note: This figure illustrates the case of the ideal A/D conversion characteristics.

Figure 53.50 Data range of A/D conversion result (10-bit data length, Unsigned data, Single-ended input)

53.5.8 Compare Match Function

53.5.8.1 Compare Match

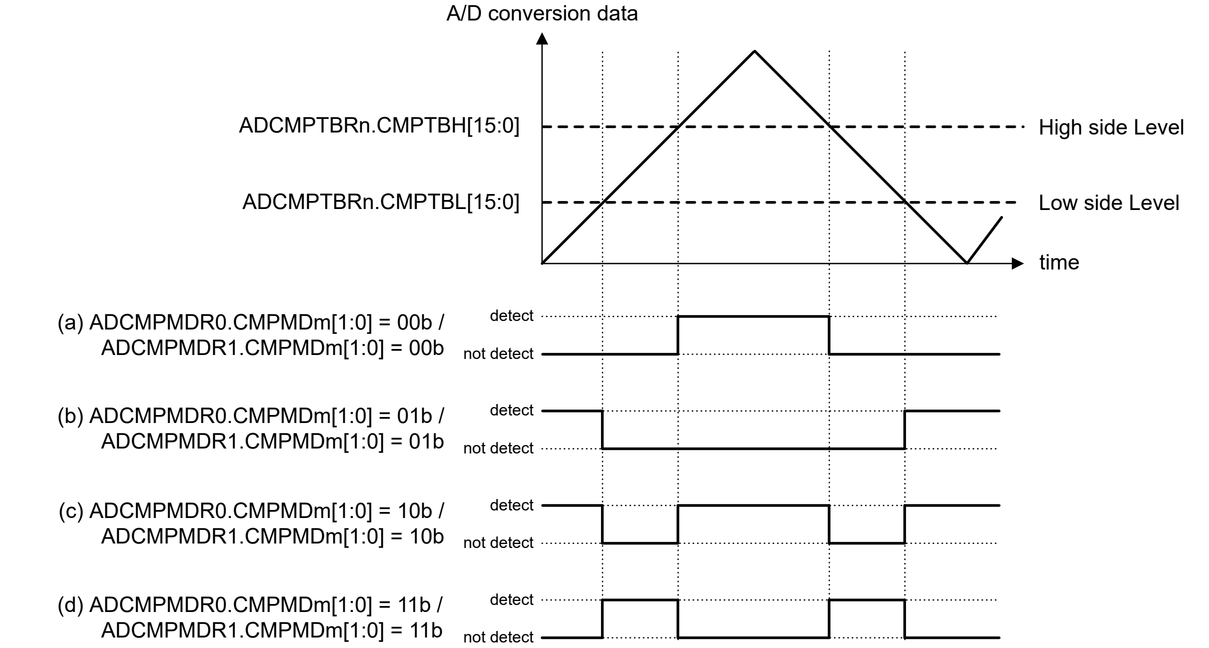
The compare match function compares the A/D conversion result with the reference value set in the compare match table register (ADCMPTBRn (n = 0 to 7)). The compare match function compares the A/D conversion data after the data formatting process.

(1) Compare match mode

The comparison mode for detecting the compare match is selected in ADCMPMDRm.CMPMDn[1:0] (m = 0, 1. n = 0 to 7). The comparison mode can be selected from the following four modes.

1. Compare match is detected when the A/D conversion value is greater than or equal to the specified upper limit value.
2. Compare match is detected when the A/D conversion value is less than or equal to the specified lower limit.
3. Compare match is detected when the A/D conversion value is greater than or equal to the specified upper limit value, or less than or equal to the lower limit value.
4. Compare match is detected when the A/D conversion value is within the specified upper and lower limit values (when the value is lower than or equal to the upper limit value and greater than or equal to the lower limit value).

Figure 53.51 shows an example of compare match detection.



Note: n = 0 to 7
m = 0 to 3 (ADCMPMDR0), m = 4 to 7 (ADCMPMDR1)

Figure 53.51 Compare match detection

(2) Compare match detection

To use the compare match function, enable the compare match table used in ADCMPENR register, and select the compare match table in ADDOPCRBn.CMPTBLEm (n = 0 to 32, m = 0 to 7) for each virtual channel.

When A/D conversion is performed on a virtual channel for which the compare match function is enabled, a compare match is judged based on the A/D conversion result of its channel.

When a compare match is detected, the flags are set in the following status registers:

- ADCMPTBSR: The flag is set to the bit corresponding to the compare match table used when a compare match is detected.
- ADCMPCHSR0: The flag is set to the bit corresponding to the analog input channel that detected the compare match.
- ADCMPEXSR: The flag is set to the bit corresponding to Extended Analog Function where a compare match is detected.

To clear the flag of each status register, write 1 to the corresponding bit of ADCMPTBSCR, ADCMPCHSCR0, ADCMPEXSCR.

If compare match interrupt is enabled in ADCMPINTCR register and a compare match using compare match table 0 to 3 is detected, the corresponding interrupt is generated.

53.5.8.2 Composite Compare Match

The composite compare match function generates interrupts and ELC events by combining the comparison results of multiple compare match tables. The combination and condition of the compare match table for the composite compare match function is set in ADCCMPCR0 or ADCCMPCR1.

Composite compare match interrupt is generated when a compare match is detected that matches the specified condition.

Figure 53.52 shows the relationship between composite compare match and compare match function. Table 53.53 shows the relationship between the composite compare match function and the control register.

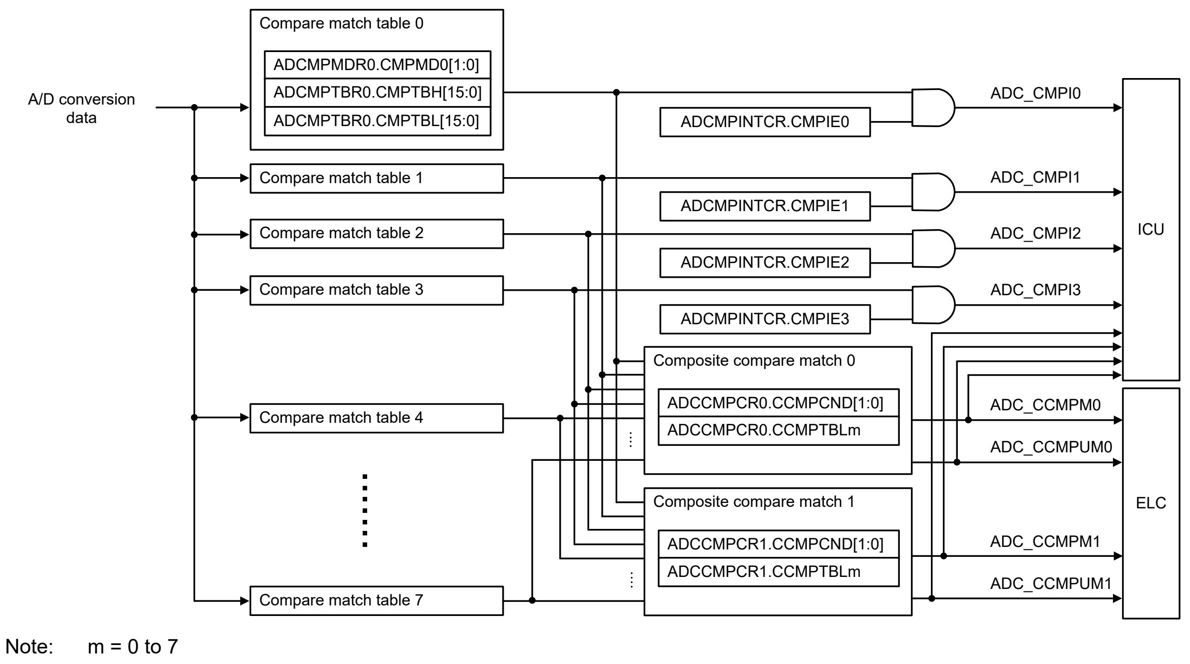


Figure 53.52 Relationship between composite compare match function and compare match function

Table 53.53 Composite compare match and control register

Function name	Enable composite compare function	Composite compare mode selection	Composite compare match interrupt
Composite compare match 0	ADCCMPCR0.CCMPTBLm (m = 0 to 7)	ADCCMPCR0.CCMPCND[1:0]	ADC_CCMPM0 ADC_CCMPUM0
Composite compare match 1	ADCCMPCR1.CCMPTBLm (m = 0 to 7)	ADCCMPCR1.CCMPCND[1:0]	ADC_CCMPM1 ADC_CCMPUM1

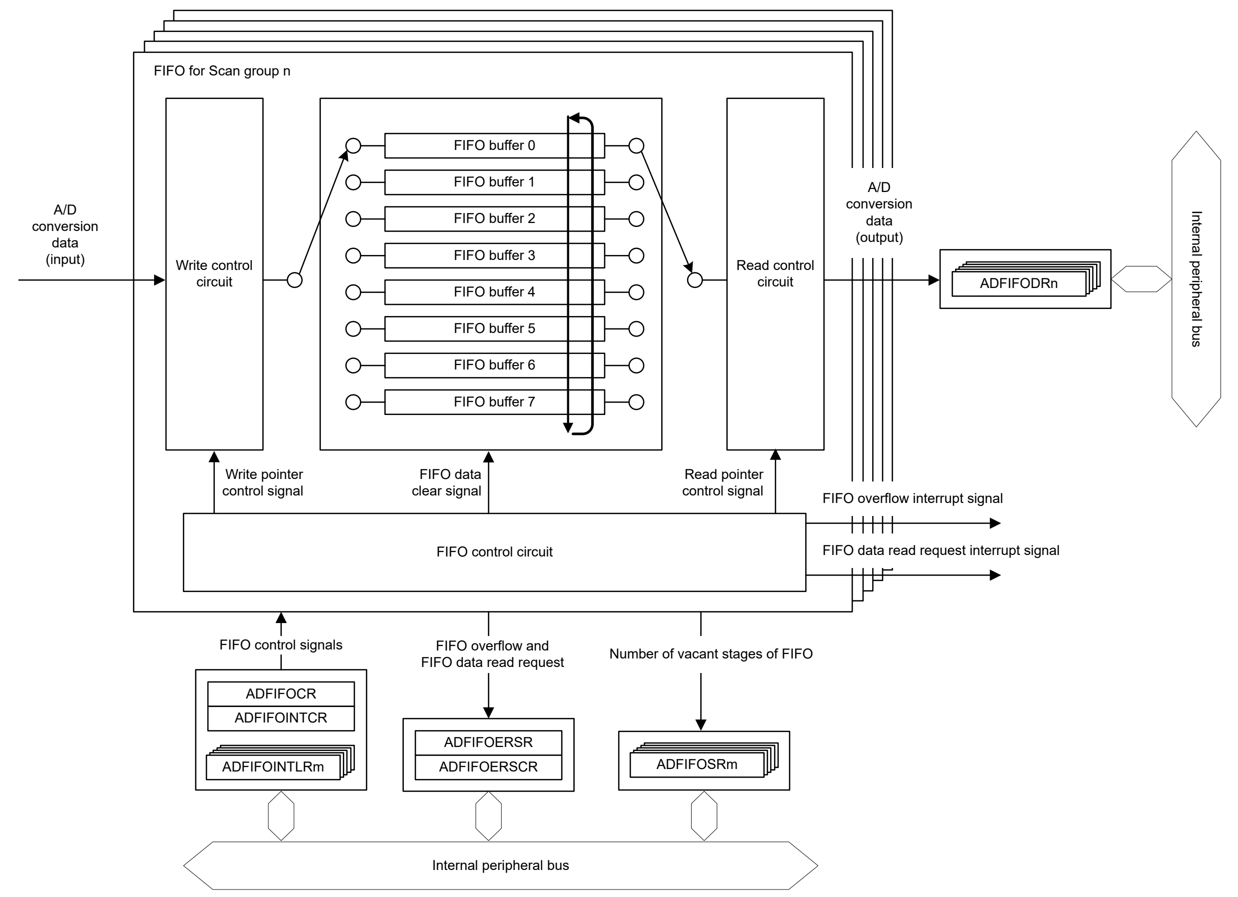
53.5.9 Data Registers

A/D data register (ADDRi (i = 0 to 22)) and Extended A/D data register (ADEXDRj (j = 0, 1, 4, 5, 6, 8, 9, 16, 17, 18, 20, 21, 22)) store the A/D conversion data of the analog channels corresponding to each register. These registers are updated (overwritten) when A/D conversion for the analog channel corresponding to each register is completed.

53.5.10 FIFO Function

FIFO consists of 8-stage registers and can hold up to 8 A/D conversion data. One FIFO is implemented for each scan group. The A/D conversion data stored in FIFO can be read from ADFIFODRn register (n = 0 to 8).

FIFO acts as a ring buffer. Writing and reading A/D conversion data to and from FIFO is controlled by the write pointer and read pointer inside FIFO. The block diagram of FIFO is shown in Figure 53.53.



Note: m = 0 to 4, n = 0 to 8

Figure 53.53 The block diagram of FIFO

53.5.10.1 Operation When Writing A/D Conversion Data to FIFO

The A/D conversion data is stored sequentially to the register pointed by the write pointer. When A/D conversion data is written to a register in FIFO, the write pointer is switched to the next register. At this time, the number of vacant stages (ADFIFOSRm.FIFOSTn[3:0] (m = 0 to 4, n = 0 to 8) bits) is decreased 1.

A FIFO overflow occurs when FIFO is full (ADFIFOSRm.FIFOSTn[3:0] bits are 0000b) and an additional A/D conversion data is written to FIFO. When a FIFO overflow occurs, the A/D conversion data is not written and the write pointer does not change.

53.5.10.2 Operation When Reading A/D Conversion Data from FIFO

When A/D conversion data is read from FIFO data register (ADFIFODRn (n = 0 to 8)), the data in the register pointed to by FIFO read pointer is read. When A/D conversion data is read from FIFO, the read pointer is switched to the next register. At this time, the number of vacant stages (ADFIFOSRm.FIFOSTn[3:0] (m = 0 to 4, n = 0 to 8) bits) is increased 1.

If FIFO is empty (ADFIFOSRm.FIFOSTn[3:0] bits are 1111b) and the reading from the FIFO is occurred, an invalid data is read. (If FIFO is read after a reset or after FIFO data is cleared, 0x00000000 is read. If data had been written to register in FIFO, the previous data is read.) The read pointer does not change in the read operation when FIFO is empty.

Read A/D conversion data from FIFO data register (ADFIFODRn) by 32-bit access. For details, see section 53.5.10.5. Usage Notes on FIFO.

53.5.10.3 FIFO Data Read Request and Overflow

When A/D conversion data is written to or read from FIFO, the write pointer or read pointer changes. Depending on the state of the write pointer and the read pointer, the number of vacant stages in FIFO is reflected in ADFIFOSRm.FIFOSTn[3:0] (m = 0 to 4, n = 0 to 8).

FIFO data read request flag is set (ADFIFOERSR.FIFOFLFn = 1 (n = 0 to 8)) when the number of vacant stages become less than or equal to the value specified in ADFIFOINTLRm.FIFOILVn[3:0] (m = 0 to 4, n = 0 to 8) bit. If a FIFO data read request occurs, FIFO data read request interrupt can be generated.

When FIFO is full (ADFIFOSRm.FIFOSTn[3:0] bits are 0000b) and additional A/D conversion data are written, FIFO overflow flag is set (ADFIFOERSR.FIFOOVFn = 1 (n = 0 to 8)). If a FIFO overflow occurs, FIFO data overflow interrupt can be generated.

53.5.10.4 Clearing FIFO Data

The data stored in the registers in FIFO can be cleared by writing to ADFIFODCR.FIFODCn (n = 0 to 8) bit. When writing 1 to ADFIFODCR.FIFODCn bit, the registers, read pointer, and write pointer in FIFO of the scan group corresponding to FIFODCn bit are initialized. Clearing FIFO data should be performed while A/D conversion is not being performed.

53.5.10.5 Usage Notes on FIFO

- Reading
  Reading the A/D conversion data from the FIFO data register (ADFIFODRn (n = 0 to 8)) should be performed with a single 32-bit access. Access to each bit of ADFIFODRn, 16-bit read access, and 8-bit read access are prohibited. If these restrictions are violated, the operation of FIFO is not guaranteed as the read pointer may not change normally, and correct data may not be able to read. Alternatively, the read pointer may advance unintentionally, and A/D conversion data may be lost.
- Rescanning
  When rescanning from the first channel by resuming the scan in group priority operation (ADGSPCR.PGSm = 1 (m = 0, 1), ADGSPCR.LGRPSm = 0 (m = 0, 1)), enable FIFO clear (ADFIFOCR.FIFOCEn = 1 (n = 0 to 8)) for scan groups other than the highest priority.
- Clearing
  When FIFO clear is enabled (ADFIFOCR.FIFOCEn = 1 (n = 0 to 8)), all data remaining in the FIFO at the start of scan or restart of a scan is cleared. Control the scan start timing so that the FIFO is not cleared before the necessary FIFO data is read.
  Note that the clearing FIFO data function by starting/resuming scanning is prohibited for other than SAR mode and group priority operation.

53.5.11 Data Output Decimation Function

In Oversampling mode — One-channel continuous scan mode, the output of conversion data can be decimated. The conversion data output interval can be set for each unit using ADDECCR.DCIMm (m = 0, 1). When conversion data output is decimated using this function, the interrupt output is decimated as well.

When this function is used in addition/average mode, the conversion data output interval is the period when the number of times the conversion data has been added or averaged and then subtracted by this function. In addition/average mode, the data to be subtracted by this function is the data before addition/averaging.

Figure 53.54 and Figure 53.55 show examples of operation when data is output every 4 times.

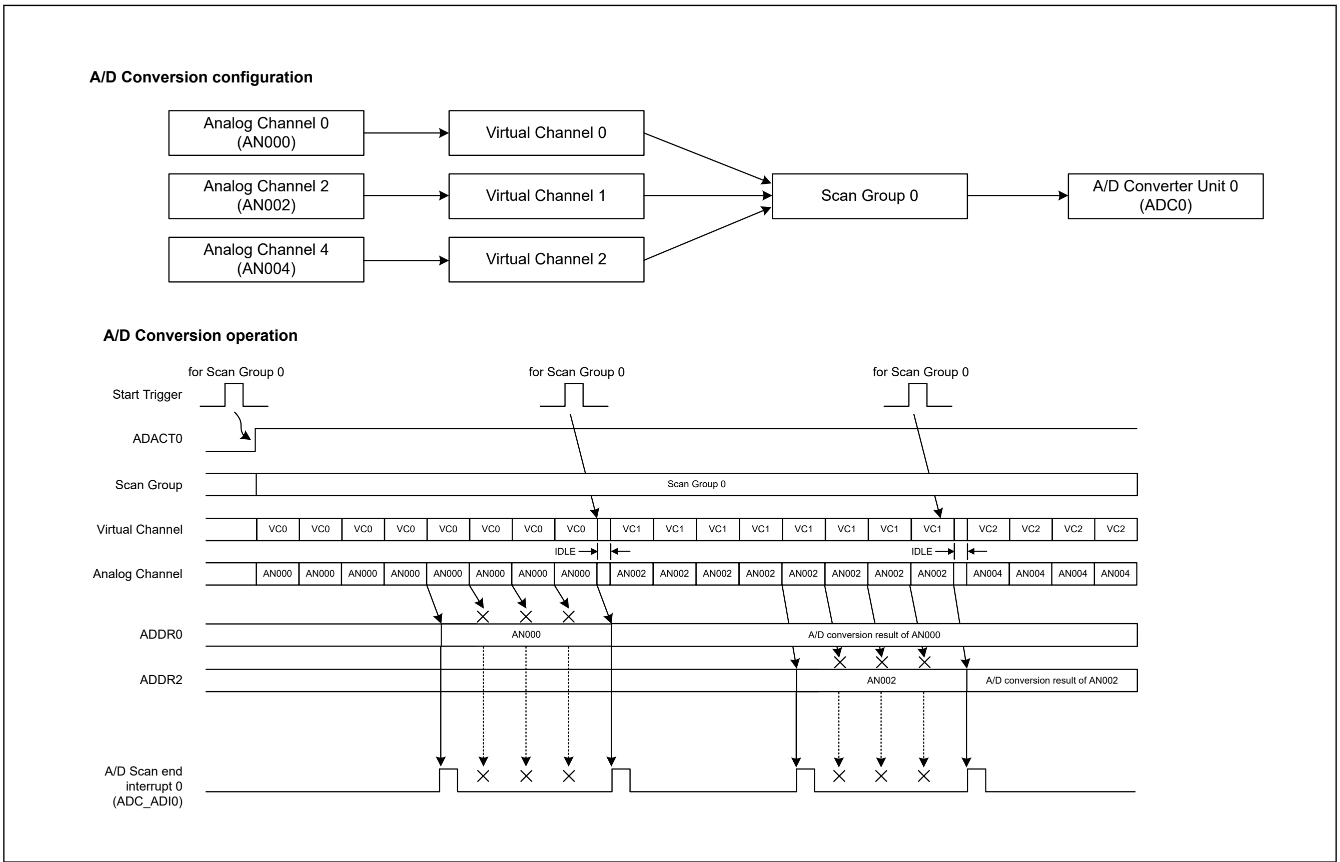


Figure 53.54 Example of data output decimation function (output every 4 times)

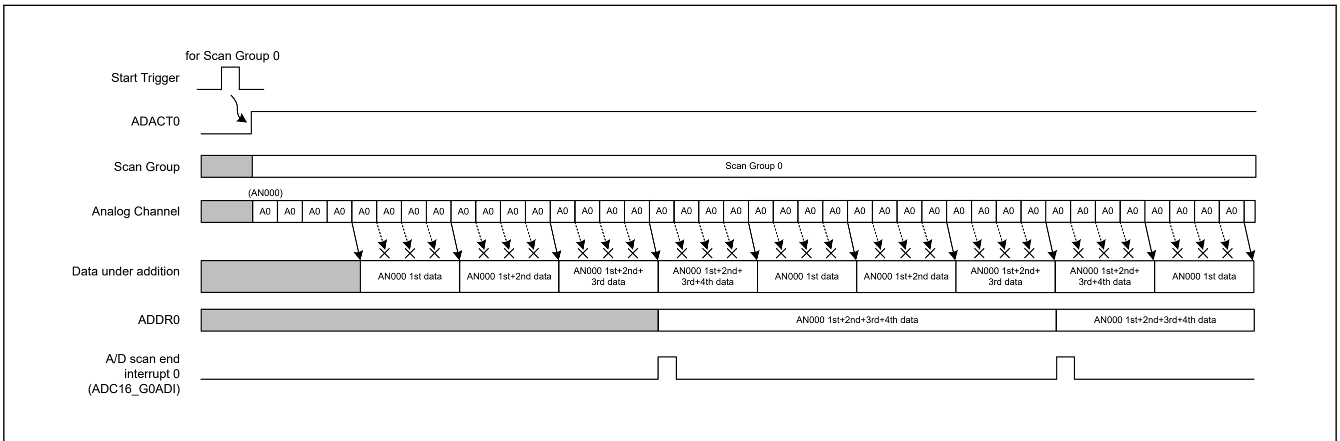


Figure 53.55 Example of data output decimation function (output every 4 times) when addition function is enabled (addition count is 4 times)

53.5.12 A/D Conversion Data Error Detection

The ERR bit in A/D data register (ADDRi (i = 0 to 22)), Extended A/D data register (ADEXDRj (j = 0, 1, 4, 5, 6, 8, 9, 16, 17, 18, 20, 21, 22)), and FIFO data register (ADFIFODRk (k = 0 to 8)) is a flag that indicates an A/D conversion data error. An A/D conversion data error indicates that an invalid A/D conversion data has been detected. If an A/D conversion data error occurs (ERR = 1), the A/D conversion data cannot be guaranteed.

A/D conversion data error is generated by the following factors.

- A/D converter error
- A/D conversion overflow

For more details about these errors, see section 53.7. Error Detection.

53.6 Start and Stop Control of A/D Conversion

53.6.1 Software Trigger

A/D conversion of scan group n can be started by writing 1 to ADSTRn.ADST (n = 0 to 8) bit or ADSYSTR.ADSYSTn (n = 0 to 8) bit.

ADSTRn.ADST bit is used to start A/D conversion individually for each scan group. ADSYSTR.ADSYSTn bit is used to start A/D conversion on the scan group assigned to ADC0 or ADC1 at the same time.

Except when group priority operation, writing to ADSTRn.ADST bit or ADSYSTR.ADSYSTn bit for the scan groups, which use A/D converter that is in A/D conversion, is ignored.

53.6.2 Peripheral Module Triggers

A/D conversion can be started by triggers from the peripheral modules listed in the followings.

- ELC Trigger
- GPT Trigger
- External Trigger (I/O port)

To perform A/D conversion by a trigger from a peripheral module, configure the trigger for each scan group and enable the trigger input from the peripheral module in ADTRGENR register.

53.6.2.1 ELC Trigger

A/D conversion can be started by an event (ELC event) from Event Link Controller. To start A/D conversion using an ELC event, configure the scan group using the ELC event in ADTRGELCn (n = 0 to 8) register and enable the A/D conversion start trigger in ADTRGENR register. Table 53.54 shows the correspondence between ADTRGELCn register and ELC event.

Table 53.54 Correspondence between ELC Trigger Enable Register and ELC Event

Resister bit	Event Name
ADTRGELCn.TRGELC0	ELC_AD00
ADTRGELCn.TRGELC1	ELC_AD01
ADTRGELCn.TRGELC2	ELC_AD02
ADTRGELCn.TRGELC3	ELC_AD10
ADTRGELCn.TRGELC4	ELC_AD11
ADTRGELCn.TRGELC5	ELC_AD12

Note: n = 0 to 8

53.6.2.2 GPT Trigger

A/D conversion can be started by an interrupt source from the general-purpose PWM Timer (GPT). To start A/D conversion using an interrupt source from the GPT, configure the scan group using the GPT interrupt source in ADTRGGPTn (n = 0 to 8) register and enable the A/D conversion start trigger in ADTRGENR register. Table 53.55 shows the correspondence between ADTRGGPTn register and GPT interrupt sources.

Table 53.55 Correspondence between GPT Trigger Enable Registers and GPT Interrupt Sources (1 of 2)

Resister bit	Event Name
ADTRGGPTn.TRGGPTA0	GPT0_ADTRGA
ADTRGGPTn.TRGGPTA1	GPT1_ADTRGA
ADTRGGPTn.TRGGPTA2	GPT2_ADTRGA
ADTRGGPTn.TRGGPTA3	GPT3_ADTRGA
ADTRGGPTn.TRGGPTA4	GPT4_ADTRGA
ADTRGGPTn.TRGGPTA5	GPT5_ADTRGA

Table 53.55 Correspondence between GPT Trigger Enable Registers and GPT Interrupt Sources (2 of 2)

Resister bit	Event Name
ADTRGGPTn.TRGGPTA6	GPT6_ADTRGA
ADTRGGPTn.TRGGPTA7	GPT7_ADTRGA
ADTRGGPTn.TRGGPTA8	GPT8_ADTRGA
ADTRGGPTn.TRGGPTA9	GPT9_ADTRGA
ADTRGGPTn.TRGGPTA10	GPT10_ADTRGA
ADTRGGPTn.TRGGPTA11	GPT11_ADTRGA
ADTRGGPTn.TRGGPTA12	GPT12_ADTRGA
ADTRGGPTn.TRGGPTA13	GPT13_ADTRGA
ADTRGGPTn.TRGGPTB0	GPT0_ADTRGB
ADTRGGPTn.TRGGPTB1	GPT1_ADTRGB
ADTRGGPTn.TRGGPTB2	GPT2_ADTRGB
ADTRGGPTn.TRGGPTB3	GPT3_ADTRGB
ADTRGGPTn.TRGGPTB4	GPT4_ADTRGB
ADTRGGPTn.TRGGPTB5	GPT5_ADTRGB
ADTRGGPTn.TRGGPTB6	GPT6_ADTRGB
ADTRGGPTn.TRGGPTB7	GPT7_ADTRGB
ADTRGGPTn.TRGGPTB8	GPT8_ADTRGB
ADTRGGPTn.TRGGPTB9	GPT9_ADTRGB
ADTRGGPTn.TRGGPTB10	GPT10_ADTRGB
ADTRGGPTn.TRGGPTB11	GPT11_ADTRGB
ADTRGGPTn.TRGGPTB12	GPT12_ADTRGB
ADTRGGPTn.TRGGPTB13	GPT13_ADTRGB

Note: n = 0 to 8

53.6.2.3 External Trigger

A/D conversion can be started by the input from the external trigger pin (ADTRG0 and ADTRG1). To start A/D conversion using an external trigger, configure the scan group using the external trigger in ADTRGEXTn (n = 0 to 8) register and enable the A/D conversion start trigger in ADTRGENR register.

External trigger is active low. Before setting to enable the external trigger (ADTRG0 and ADTRG1), you should input the High level to the external trigger pin.

53.6.3 Trigger Delay

The trigger delay function adds a delay to the A/D conversion start trigger in order to adjust the A/D conversion start timing for each scan group. This function adds a delay to the internal trigger for each scan group generated by accepting an external trigger, ELC trigger or GPT trigger. This function cannot be used to add a delay to a software trigger.

The delay value to be added to the internal trigger is set for each scan group in ADTRGDLRi (i = 0 to 4) register. The delay value to be added is the register setting value × A/D conversion clock (ADCLK) cycle.

53.6.4 Force Stops the A/D Conversion Operation

The scanning operation of the A/D converter can be forcibly stopped by writing 1 to ADSTOPR.ADSTOPm (m = 0, 1) bit during scanning operation. If the scanning operation is forcibly stopped, the A/D conversion data is not guaranteed.

53.6.4.1 Forced Stop Procedure

Table 53.56 shows the procedure for forced stop of A/D conversion operation. Observe Table 53.56 when performing the forced stop. If this procedure is violated, the A/D converter may not be able to stop and then it may not operate normally. In that case, a reset is required for recovery.

Table 53.56 Procedure for forced stop of A/D conversion operation

No.	Step Name	Description
1	Disable trigger input	Disables the trigger input from the peripheral module. Write ADTRGENR.STTRGENn = 0.
2	Waiting time	After setting the above Step 1, a wait time is required to safely stop the A/D converter. Proceed to the next processing after the specified waiting time has elapsed. For details on the wait time, see section 53.6.4.2. Waiting Time after Disabling Trigger Input for Forced Stop Processing.
3	Checking the A/D converter status	Check if the A/D converter is operating after the waiting time of the above Step 2 has elapsed. If the A/D converter is running (ADSR.ADACTm = 1), proceed to Step 4. If the A/D converter is stopped (ADSR.ADACTm = 0), no further processing is required (proceed to Step 6).
4	Forced stop of A/D converter	Forcibly stop the A/D converter by ADSTOPR register. Write ADSTOPR.ADSTOPm = 1.
5	Waiting for A/D converter to stop	Wait until the A/D converter stops. ADSR.ADACTm = 0.
6	End	Processing of forced stop is completed.

Note: m = 0, 1, n = 0 to 8
Do not write to ADSYSTR register or ADSTRn register while this forced stop procedure is in progress.

53.6.4.2 Waiting Time after Disabling Trigger Input for Forced Stop Processing

In the forced stop procedure shown in Table 53.56, after disabling the trigger input from the peripheral module, a wait time is required to perform the forced stop (Step 2 of Table 53.56). This wait time is the time that must be secured to perform a forced stop safely.

The wait time after disabling the trigger input for forced stop processing is calculated as follows.

[Waiting time after disabling trigger input in forced stop processing]

- When synchronous operation is enabled (ADSYCR.ADSYDISm = 0)
  - (ADTRGDLRi.TRGDLYn[7:0] setting value + ADSYCR.ADCSYCYC[7:0] setting value × 2) × t_{ADCLK}
- When synchronous operation is disabled (ADSYCR.ADSYDISm = 1)
  - (4 + ADTRGDLRi.TRGDLYn[7:0] setting value) × t_{ADCLK}

Note: i = 0 to 4, n = 0 to 8, m = 0, 1 t_{ADCLK}: ADCLK period

After the waiting time calculated by the above expressions has elapsed, proceed to the Step 3 in Table 53.56. There are two ways to generate the waiting time: (a) Use the timer function built into MCU. (b) Read the status register multiple times. If you choose (b), the number of register reads to generate the waiting time required is calculated by the following equation:

[Number of register reads to generate the waiting time]

N_{RD} = (Wait time after disabling trigger input ÷ (t_{PCLK} × N_{RDCYC})) + 1

N_{RD}: Number of register reads required to generate the trigger input disable wait time (truncate after the decimal point)

t_{PCLK}: The period of the Bus I/F clock (PCLKA)

N_{RDCYC}: Minimum number of register read access cycles (see section x, I/O Registers [DEADBEEF].)

53.7 Error Detection

53.7.1 A/D Converter Error

An A/D converter error is detected when an abnormal operation of the A/D converter occurs. If an A/D converter error is detected, the A/D conversion result is not guaranteed.

An A/D converter error occurs in the following cases:

[A/D Converter Error Generation Conditions]

- When A/D conversion is performed without self-calibration.
- When the operating frequency of ADCLK exceeds the guaranteed operating frequency range specified in the electric characteristics.
- When the successive approximation time of the A/D converter exceeds the guaranteed operating range specified in the electric characteristics.
- When an accidental abnormal operation occurs in the A/D converter due to external factors.

If an A/D converter error is detected, the flag is set in ADERSR.ADERFn (n = 0, 1) bit. To clear the A/D converter error flag, write 1 to the corresponding bit in ADERSCR register.

Notes on A/D Converter Error

The A/D converter error is an auxiliary function used to detect abnormal operation of the A/D converter. It does not guarantee the reliable detection of abnormal operation. Even under the A/D converter error occurrence conditions described above, an A/D converter error may not be detected depending on the individual chip differences and chip operating conditions.

53.7.2 A/D Conversion Overflow

A/D conversion overflow is detected when A/D conversion data exceeds the range that can be handled in the specified data format. If A/D conversion overflow occurs, the A/D conversion data becomes the upper or lower limit value of the specified data format. For the range that can be handled as A/D conversion data, see section 53.5.7. Data Format.

A/D conversion overflow is detected in the following cases:

- When the input to the A/D converter exceeds the range between VREFH0 to VREFL0 or VREFH to VREFL
  - Single-ended input: When the input to the A/D converter unit 0 exceeds VREFH0 or falls below VREFL0. When the input to the A/D converter unit 1 exceeds VREFH or falls below VREFL.
  - Differential input: When the differential input to the A/D converter unit 0 exceeds +VREFH0 or falls below −VREFH0. When the differential input to the A/D converter unit 1 exceeds +VREFH or falls below −VREFH.
- When overflow occurs by the internal processing (calculation) for the A/D conversion data shown in the following:
  - Gain error and Offset error calibration
  - User’s Gain/User’s Offset adjustment function
  - When using A/D-converted value addition function
  - Data formatting process

When an A/D conversion overflow is detected, the flag is set in the following status register:

- ADOVFERSR: The flag is set to the bit corresponding to the A/D converter that detected the A/D conversion overflow.
- ADOVFCHSR0: The flag is set to the bit corresponding to the analog input channel on which A/D conversion overflow is detected.
- ADOVFEXSR: The flag is set to the bit corresponding to the extended analog channel on which A/D conversion overflow is detected.

To clear the flags in the status registers, write 1 to the corresponding bit in ADOVFERSCR, ADOVFCHSCR0, ADOVFEXSCR.

53.7.3 FIFO Overflow

If A/D conversion data is written (added) to FIFO while FIFO is full, a FIFO overflow is detected. For more details about FIFO overflow, see section 53.5.10. FIFO Function.

53.8 Procedure for Setting up and Changing A/D Conversion

53.8.1 Initial Setup Procedure for A/D Conversion

Table 53.57 shows the initial setup procedure for A/D conversion.

Table 53.57 Initial setup procedure for A/D conversion

No.	Step	Description
1	Release module-stop	Release the module-stop bit for ADC16H in MSTPCR register.
2	I/O port configuration	Set ASEL bit of the pin used as analog input to 1.
3	Synchronous operation configuration	Set the synchronous operation function. The synchronous operation function is disabled at the initial value of the register after reset release. If the synchronous operation function is used, be sure to enable the synchronous operation function (ADSYCR.ADSYDISm = 0 (m = 0,1)).
4	ADCLK configuration	Set the clock source and division ratio for ADCLK.*1 Then, set ADCLKENR.CKEN bit to 1, and wait for ADCLK to supply (ADCLKSR.CLKSR = 1).
5	A/D conversion configuration	Configure the A/D conversion settings.
6	Wait for operation stabilization	Wait until the operating stabilization times specified in the Electrical Characteristics have elapsed.
7	Self-calibration	Self-calibration must be executed prior to starting A/D conversion. Set up for self-calibration and execute it. For details, see section 53.3.8. Self-calibration.
8	Trigger configuration	To start A/D conversion by a trigger from peripheral modules, configure the triggers for each scan group.
9	Start of A/D conversion	When a software trigger or a trigger from peripheral modules is input, A/D conversion (scanning operation) starts.

Note 1. When GPTCLK is selected, release the GPT module-stop state using the Module Stop Register E (MSTPCRE). For details, see section 11, Low Power Mode.

53.8.2 Procedure for Changing ADCLK Settings

Table 53.58 shows the procedure for changing ADCLK settings.

Table 53.58 Procedure for changing ADCLK settings (1 of 2)

No.	Step	Description
1	Disable trigger input	Disables the trigger input from the peripheral module. (Write ADTRGENR.STTRGENn = 0 (n = 0 to 8))
2	Stop A/D conversion	Check that all A/D converter is stopped. When A/D conversion is in progress, wait until all A/D conversion is completed or forcibly stop A/D conversion operation. For details about forcibly stop A/D conversion operation, see section 53.6.4. Force Stops the A/D Conversion Operation.
3	Stop ADCLK supplies	Set ADCLKENR.CKEN bit to 0. Then, wait for ADCLK to stop (ADCLKSR.CLKSR = 0).
4	Change ADCLK setting	Change the clock-source and the division ratio for ADCLK*1.
5	Started supplying ADCLK	Set ADCLKENR.CKEN bit to 1, and wait for ADCLK to supply (ADCLKSR.CLKSR = 1).

Table 53.58 Procedure for changing ADCLK settings (2 of 2)

No.	Step	Description
6	Change the A/D conversion configuration	Change the following settings according to the changed ADCLK frequency. ● Successive approximation time for A/D converter ● Number of sampling states for A/D conversion ● Number of sampling states and hold mode switching states for channel-dedicated sample-and-hold circuit *2 ● Number of states for self-calibration operation (A/D converter and channel-dedicated sample-and-hold circuit *2) ● Synchronous operation period *3 ● Disconnection detection assist period *4 If other settings related to A/D conversion are also to be changed, change them in this step.
7	Wait for operation stabilization	Wait until the operating stabilization times specified in the section 62, Electrical Characteristics have elapsed.
8	Self-calibration	Execute self-calibration operation prior to starting A/D conversion. For details, see section 53.3.8. Self-calibration.
9	Trigger configuration	To start A/D conversion by a trigger from peripheral modules, configure the triggers for each scan group.
10	Start of A/D conversion	When a software trigger or a trigger from peripheral modules is input, A/D conversion (scanning operation) starts.

Note 1. When GPTCLK is selected, release the GPT module stop state using the module stop register E (MSTPCRE). For details, see section 11, Low Power Mode.
Note 2. Setting is not required when channel-dedicated sample-and-hold circuit is not used.
Note 3. No change is required when synchronous operation setting is disabled (ADSYCR, ADSYDISm = 1 (m = 0, 1)).
Note 4. Setting is not required when disconnection detection assist function is not used.

53.8.3 Procedure for Changing the Settings of the A/D Converter

Table 53.59 shows the procedure for changing the A/D conversion configuration. If there is a change in ADCLK setting, follow the procedures in section 53.8.2. Procedure for Changing ADCLK Settings.

Table 53.59 Procedure for changing the settings of the A/D converter

No.	Step	Description
1	Disable trigger input	Disables the trigger input from the peripheral module. (Write ADTRGENR.STTRGENn = 0 (n = 0 to 8))
2	Stop A/D conversion	Check that all A/D converter is stopped. When A/D conversion is in progress, wait until all A/D conversion is completed or forcibly stop A/D conversion operation. For details about forcibly stop A/D conversion operation, see section 53.6.4. Force Stops the A/D Conversion Operation.
3	Change the A/D conversion configuration	Change the A/D conversion settings. (except the settings for ADCLK)
4	Wait for operation stabilization	Wait until the various operating stabilization times specified in the Electrical Characteristics have elapsed.
5	Self-calibration	Execute self-calibration operation prior to starting A/D conversion. For details, see section 53.3.8. Self-calibration.
6	Trigger configuration	To start A/D conversion by a trigger from peripheral modules, configure the triggers for each scan group.
7	Start of A/D conversion	When a software trigger or a trigger from peripheral modules is input, A/D conversion (scanning operation) starts.

53.9 Interrupt Sources and ELC Events

Table 53.60 lists ADC16H interrupt sources or ELC event requests.

For more details about interrupts, see section 14, Interrupt Controller Unit (ICU).

For more details about ELC events, see section 19, Event Link Controller (ELC).

Table 53.60 Interrupt sources (1 of 2)

Interrupt request or ELC event	Symbol	Description	Status flag
A/D converter error interrupt	ADC_ERR0	Generated when abnormal operation of ADC0 is detected.	ADERSR.ADERF0
	ADC_ERR1	Generated when abnormal operation of ADC1 is detected.	ADERSR.ADERF1
A/D converter self-calibration end interrupt	ADC_CALEND0	Generated when self-calibration operation of ADC0 is finished.	ADCALENDSR.CALENDF0
	ADC_CALEND1	Generated when self-calibration operation of ADC1 is finished.	ADCALENDSR.CALENDF1
A/D scan end interrupt	ADC_ADI0	Generated at the scan end of scan group 0	ADSCANENDSR.SCENDF0
	ADC_ADI1	Generated at the scan end of scan group 1	ADSCANENDSR.SCENDF1
	ADC_ADI2	Generated at the scan end of scan group 2	ADSCANENDSR.SCENDF2
	ADC_ADI3	Generated at the scan end of scan group 3	ADSCANENDSR.SCENDF3
	ADC_ADI4	Generated at the scan end of scan group 4	ADSCANENDSR.SCENDF4
	ADC_ADI5	Generated at the scan end of scan group 5	ADSCANENDSR.SCENDF5
	ADC_ADI6	Generated at the scan end of scan group 6	ADSCANENDSR.SCENDF6
	ADC_ADI7	Generated at the scan end of scan group 7	ADSCANENDSR.SCENDF7
	ADC_ADI8	Generated at the scan end of scan group 8	ADSCANENDSR.SCENDF8
Limiter clip interrupt	ADC_LIMCLPI	Generated when a limiter clip using limiter table 0 to 7 is detected for A/D conversion.	ADLIMGRSR.LIMGRFn ADLIMCHSR0.LIMCHFi ADLIMEXSR.LIMEXFj
A/D conversion overflow interrupt	ADC_RESOVF0	Generated when the overflow is detected in A/D conversion result with ADC0.	ADOVFERSR.ADOVFEF0 ADOVFCHSR0.OVFCHFi ADOVFEXSR.OVFEXFj
	ADC_RESOVF1	Generated when the overflow is detected in A/D conversion result with ADC1.	ADOVFERSR.ADOVFEF1 ADOVFCHSR0.OVFCHFi ADOVFEXSR.OVFEXFj
Compare match interrupt	ADC_CMPI0	Generated when a compare match with compare match table 0 is detected.	ADCMPTBSR.CMPTBF0
	ADC_CMPI1	Generated when a compare match with compare match table 1 is detected.	ADCMPTBSR.CMPTBF1
	ADC_CMPI2	Generated when a compare match with compare match table 2 is detected.	ADCMPTBSR.CMPTBF2
	ADC_CMPI3	Generated when a compare match with compare match table 3 is detected.	ADCMPTBSR.CMPTBF3
Composite compare match interrupt	ADC_CCMPM0 ADC_CCMPM1	Generated when a composite compare match with combined conditions using compare match table 0 to 7 is detected.	ADCMPTBSR.CMPTBF0 to ADCMPTBSR.CMPTBF8
	ADC_CCMPUM0 ADC_CCMPUM1	Generated when a composite compare match with combined conditions using compare match table 0 to 7 is not detected.	ADCMPTBSR.CMPTBF0 to ADCMPTBSR.CMPTBF8

Table 53.60 Interrupt sources (2 of 2)

Interrupt request or ELC event	Symbol	Description	Status flag
FIFO data read request interrupt	ADC_FIFOREQ0	Generated when the number of vacant stages in FIFO for scan group 0 become less than or equal to the specified value.	ADFIFOSR0.FIFOST0[3:0]
	ADC_FIFOREQ1	Generated when the number of vacant stages in FIFO for scan group 1 become less than or equal to the specified value.	ADFIFOSR0.FIFOST1[3:0]
	ADC_FIFOREQ2	Generated when the number of vacant stages in FIFO for scan group 2 become less than or equal to the specified value.	ADFIFOSR1.FIFOST2[3:0]
	ADC_FIFOREQ3	Generated when the number of vacant stages in FIFO for scan group 3 become less than or equal to the specified value.	ADFIFOSR1.FIFOST3[3:0]
	ADC_FIFOREQ4	Generated when the number of vacant stages in FIFO for scan group 4 become less than or equal to the specified value.	ADFIFOSR2.FIFOST4[3:0]
	ADC_FIFOREQ5	Generated when the number of vacant stages in FIFO for scan group 5 become less than or equal to the specified value.	ADFIFOSR2.FIFOST5[3:0]
	ADC_FIFOREQ6	Generated when the number of vacant stages in FIFO for scan group 6 become less than or equal to the specified value.	ADFIFOSR3.FIFOST6[3:0]
	ADC_FIFOREQ7	Generated when the number of vacant stages in FIFO for scan group 7 become less than or equal to the specified value.	ADFIFOSR3.FIFOST7[3:0]
	ADC_FIFOREQ8	Generated when the number of vacant stages in FIFO for scan group 8 become less than or equal to the specified value.	ADFIFOSR4.FIFOST8[3:0]
FIFO data overflow interrupt	ADC_FIFOOVF	Generated when FIFO overflow is detected in any of FIFO for scan group 0 to 8.	ADFIFOERSR.FIFOOVF0 to ADFIFOERSR.FIFOOVF8

Note: n = 0 to 8.
i = 0 to 28.
j = 0 to 2, 5 to 8.

(1) A/D converter error interrupt

A/D converter error interrupt can be generated when an A/D converter error is detected.

A/D converter error interrupt (ADC_ERRm (m = 0, 1)) is generated when ADERINTCR.ADEIEm (m = 0, 1) bit is 1 and ADERSR.ADERFm (m = 0, 1) bit is 1.

(2) A/D converter self-calibration end interrupt

A/D converter self-calibration end interrupt can be generated at the end of A/D converter self-calibration operation.

A/D converter self-calibration end interrupt (ADC_CALENDm (m = 0, 1)) is generated when ADCALINTCR.CALENDIEm (m = 0, 1) bit is 1 and ADCALENDSR.CALENDFm (m = 0, 1) bit is 1.

(3) A/D scan end interrupt

A/D scan end interrupt can be generated at the end of scanning operation of scan group n (n = 0 to 8).

A/D scan end interrupt for scan group 0 to 8 (ADC_ADI0 to ADC_ADI8) are generated when ADINTCR.ADIEn (n = 0 to 8) bit is set to 1 and ADSCANENDSR.SCENDFn (n = 0 to 8) bit is set to 1.

However, A/D scan end interrupt is not generated when A/D conversion operation (scanning operation) is forcibly stopped by ADSTOPR register.

(4) Limiter clip interrupt

Limiter clip interrupt (ADC_LIMCLPI) can be generated when a limiter clip with limiter table i (i = 0 to 7) is detected.

Limiter clip interrupt is generated when either ADLIMINTCR.LIMIEi (i = 0 to 8) bit is 1 and ADLIMGRSR.LIMGRFi (i = 0 to 8) bit is 1 is satisfied.

(5) A/D conversion overflow interrupt

A/D conversion overflow interrupt can be generated when A/D conversion overflow in the A/D conversion result using ADCm (m = 0, 1) is detected. For details about A/D conversion overflow, see section 53.7.2. A/D Conversion Overflow.

A/D conversion overflow interrupt (ADC_RESOVFm (m = 0, 1)) is generated when ADOVFINTCR.ADOVFIEm (m = 0, 1) bit is 1 and ADOVFERSR.ADOVFEFm (m = 0, 1) bit is 1.

(6) Compare match interrupt

Compare match interrupt can be generated when a compare match is detected.

Compare match interrupt (ADC_CMPIj (j = 0 to 3)) is generated when ADCMPINTCR.CMPIEj (j = 0 to 3) bit is 1 and ADCMPTBSR.CMPTBFj (j = 0 to 3) bit is 1.

There are no interrupts corresponding to the compare match tables 4 to 7.

(7) Composite compare match interrupt

Composite compare match interrupt (ADC_CCMPMk (k = 0, 1)) can be generated by combining the comparison results of two or more compare match tables. For details about composite compare match, see section 53.5.8.2. Composite Compare Match.

(8) FIFO data read request interrupt

FIFO data read request interrupt can be generated when the number of vacant stages in FIFO become less than or equal to the specified value.

FIFO data read request interrupt for scan group 0 to 8 (ADC_FIFOREQ0 to ADC_FIFOREQ8) are generated when ADFIFOINTCR.FIFOIEn (n = 0 to 8) is set to 1 and ADFIFOSRm.FIFOSTn[3:0] ≤ ADFIFOINTLRm.FIFOILVn[3:0] (m = 0 to 4, n = 0 to 8).

However, FIFO read request interrupt is not generated while the ADFIFOERSR.FIFOFLFn (n = 0 to 8) bit corresponding to its interrupt source is set to 1.

(9) FIFO data overflow interrupt

FIFO data overflow interrupt (ADC _FIFOOVF) can be generated when a FIFO overflow is detected in any of FIFO in scan groups 0 to 8.

FIFO data overflow interrupt is generated when ADFIFOINTCR.FIFOIEn (n = 0 to 8) bit is 1 and ADFIFOERSR.FIFOOVFn (n = 0 to 8) bit is 1.

53.10 Scan Conversion Time

53.10.1 Scan Start Processing Time

The processing time from the input of the A/D conversion start trigger to the start of the scanning operation is shown in Table 53.61 and Figure 53.56.

Table 53.61 Scan start processing time

Item		Symbol	Processing time
Peripheral module trigger input processing time*1		t_{D_TRG}	[When GPIO trigger is selected] ● 2 to 4 ADSRCCLK + (ADTRGDLRi.TRGDLYn[7:0] + (0 to 1)) × ADCLK [When other than above] 1 ADSRCCLK + (ADTRGDLRi.TRGDLYn[7:0] + (0 to 1)) × ADCLK
Software trigger input processing time		t_{D_SW}	[When ADCLK = PCLKA/1]*2 ● Number of access cycles to I/O registers*3 + (0 to 1 ADCLK) [Other than above] ● (Number of access cycles to I/O registers*3) + 1 PCLKA + (1 to 3 ADCLK)
Internal Trigger processing time		t_{D_ITRG}	[When synchronous operation is disabled] (ADSYCR.ADSYDISm = 1) ● 6 ADCLK [Idle start] ● 7 to 13 ADCLK [Suspend start]*4 [When synchronous operation is enabled] (ADSYCR.ADSYDISm = 0) ● 7 ADCLK [Idle start] ● (3+ADSYCR.ADSYCYC[7:0]) × ADCLK to (2+2×ADSYCR.ADSYCYC[7:0]) × ADCLK [Suspend start]*5
Wait Time for Synchronous Operation		t_{D_SYOP}	[When synchronous operation is disabled] (ADSYCR.ADSYDISm = 1) ● 0 [When synchronous operation is enabled] (ADSYCR.ADSYDISm = 0) ● 0 to ADSYCR.ADSYCYC[7:0] × ADCLK
Total scan start processing time	Started by Peripheral Module Trigger	t_{D_ADST}	$t_{D\_TRG} + t_{D\_ITRG} + t_{D\_SYOP}$
	Started by Software Trigger		$t_{D\_SW} + t_{D\_ITRG} + t_{D\_SYOP}$

Note: ADSRCCLK: Clock source of ADCLK
n = 0 to 8, m = 0, 1, i = 0 to 4

Note 1. It does not include the input delay time from the source of the peripheral module trigger to ADC16H.

Note 2. When ADCLKCR.CLKSEL[1:0] = 10b and ADCLKCR.DIVR[2:0] = 000b are set.

Note 3. It is the number of access cycles to ADSTRn register (n = 0 to 8) or ADSYSTR register. For more details about the number of access cycles for I/O registers, see section x, I/O Registers [DEADBEEF].

Note 4. When constant sampling is enabled, it is 7 to (6 + ADSHSTRm.SHSST[7:0]) × ADCLK. When constant sampling is enabled, by setting the sampling time setting of the channel-dedicated sample-and-hold circuit to ADSHSTRm.SHSST[7:0] = 0x4 (reset value), the effect of the ADSHSTRm.SHSST[7:0] setting on the conversion start time can be ignored.

Note 5. When constant sampling is enabled, (3+ADSYCR.ADSYCYC[7:0]) × ADCLK to (2+ADSHSTRm.SHSST[7:0] + ADSYCR.ADSYCYC[7:0]) × ADCLK. When constant sampling is enabled, by setting the sampling time setting of the channel-dedicated sample-and-hold circuit to ADSHSTRm.SHSST[7:0] = 0x4 (reset value), the effect of the ADSHSTRm.SHSST[7:0] setting on the conversion start time can be ignored.

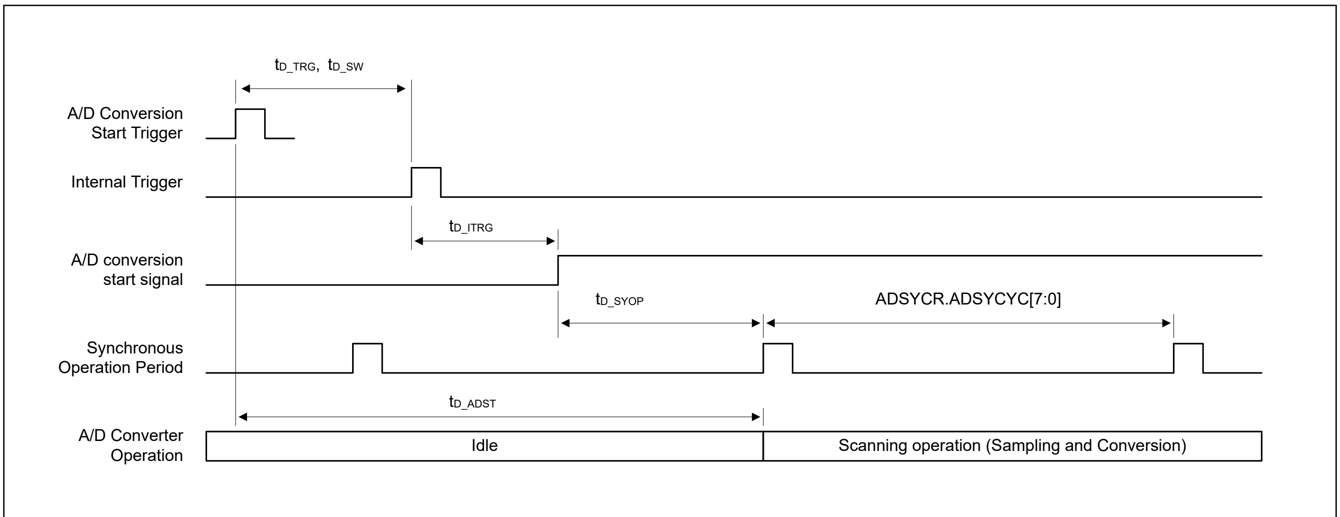


Figure 53.56 Scan start processing time

53.10.2 Conversion Processing Time

The various processing times in the conversion operation are shown in Table 53.62 and Figure 53.57 to Figure 53.61.

Table 53.62 A/D conversion processing time (1 of 2)

Item			Symbol	Processing Time
Channel-dedicated sample-and-hold processing time	Sampling time		t_{SH_SPL}	ADSHSTRm.SHSST[7:0] × ADCLK
	Hold mode switching time		t_{SH_HLD}	ADSHSTRm.SHHST[2:0] × ADCLK
	Sampling mode switching time (only when Hybrid mode)		t_{SH_D}	(ADSYCR.ADSYCYC[7:0] - 1) × ADCLK
Disconnection detection assist processing time			t_{DDA}	[When disconnection detection assist function is disabled] • 0 [When disconnection detection assist function is enabled] • ADSGDCRn.ADNDIS[7:0] × ADCLK
A/D conversion time	Sampling time		t_{AD_SPL}	ADSSTRp.SSTq[9:0] × ADCLK
	Successive approximation time		t_{AD_CNV}	ADCNVSTR.CSTm[5:0] × ADCLK
A/D conversion data processing time	SAR mode (Not using digital filter function)	When ADCLK = PCLKA/1 is set*1	t_{ADDP}	[When A/D-converted value addition/averaging function is not used] • 6 ADCLK + 2 PCLKA [When A/D-converted value addition/averaging function is used] • 7 ADCLK + 2 PCLKA
		Other than above		[When A/D-converted value addition/averaging function is not used] • 7 ADCLK + (5 to 6 PCLKA) [When A/D-converted value addition/averaging function is used] • 8 ADCLK + (5 to 6 PCLKA)
	Oversampling mode or Hybrid mode (Using with digital filter function)	When ADCLK = PCLKA/1 is set*1		[When A/D-converted value addition/averaging function is not used] • 8 ADCLK + 2 PCLKA [When A/D-converted value addition/averaging function is used] • 9 ADCLK + 2 PCLKA
		Other than above		[When A/D-converted value addition/averaging function is not used] • 9 ADCLK + 5 to 6 PCLKA [When A/D-converted value addition/averaging function is used] • 10 ADCLK + 5 to 6 PCLKA
Total A/D conversion time (SAR mode)*3	Channel conversion time*2		t_{ADCH_S}	$(t_{DDA} + t_{AD\_SPL} + t_{AD\_CNV}) \times N_{ADC} \times$ ADCLK
	Scan conversion time*4		t_{SCAN_S}	[When channel-dedicated sample-and-hold circuit is not used] • Σt_{ADCH_S} [When channel-dedicated sample-and-hold circuit is used] • $t_{SH\_SPL} + t_{SH\_HLD} + \Sigma t_{ADCH\_S}$
Total A/D conversion time (Oversampling mode)*3	Oversampling period		t_{OV_OS}	$(t_{DDA} + t_{AD\_SPL} + t_{AD\_CNV}) \times$ ADCLK
	Channel conversion time		t_{ADCH_O}	$(t_{DDA} + t_{AD\_SPL} + t_{AD\_CNV}) \times (N_{TAP} + N_{ADC}) \times$ ADCLK
	Scan conversion time*5		t_{SCAN_O}	Σt_{ADCH_O}

Table 53.62 A/D conversion processing time (2 of 2)

Item			Symbol	Processing Time
Total A/D conversion time (Hybrid mode)*3	Channel-dedicated sample-and-hold processing time in Hybrid mode		t_{HY_SH}	$t_{SH\_SPL} + t_{SH\_HLD} + t_{SH\_D}$
	Oversampling period		t_{HY_OS}	$(t_{DDA} + t_{AD\_SPL} + t_{AD\_CNV}) \times$ ADCLK
	Scan conversion time	Initial delay	t_{HY_ID}	[When channel-dedicated sample-and-hold circuit is not used] • $(N_{TAP} + N_{ADC}) \times \Sigma t_{HY\_OS}$*6 [When channel-dedicated sample-and-hold circuit is used] • $(t_{HY\_SH} \times (N_{ADC} + N_{TAP}) - t_{SH\_D}) + (N_{TAP} + N_{ADC}) \times \Sigma t_{HY\_OS}$*6
		After initial delay time	t_{SCAN_HY}	[When channel-dedicated sample-and-hold circuit is not used] • Σt_{HY_OS}*6 [When channel-dedicated sample-and-hold circuit is used] • $t_{HY\_SH} + \Sigma t_{HY\_OS}$*6

Note: n = 0 to 8, m = 0, 1, p = 0 to 7, q = 0 to 15
N_{ADC}: This value is the number of the times of the addition/averaging according to the setting value of ADDOPCRBx.ADC[3:0] (x = 0 to 32). If A/D-converted value addition/averaging function is not used, this value is 1.
N_{TAP}: This value is the number of TAP of the digital filter that selected in ADDOPCRAx.DFSEL[2:0] (x = 0 to 32) and ADDFSRm.DFSELy[1:0] (m = 0, 1, y = 0 to 4).
N_{SGCH}: This value is the number of the channels in the scan group.

Note 1. When ADCLKCR.CLKSEL[1:0] = 10b and ADCLKCR.DIVR[2:0] = 000b are set.

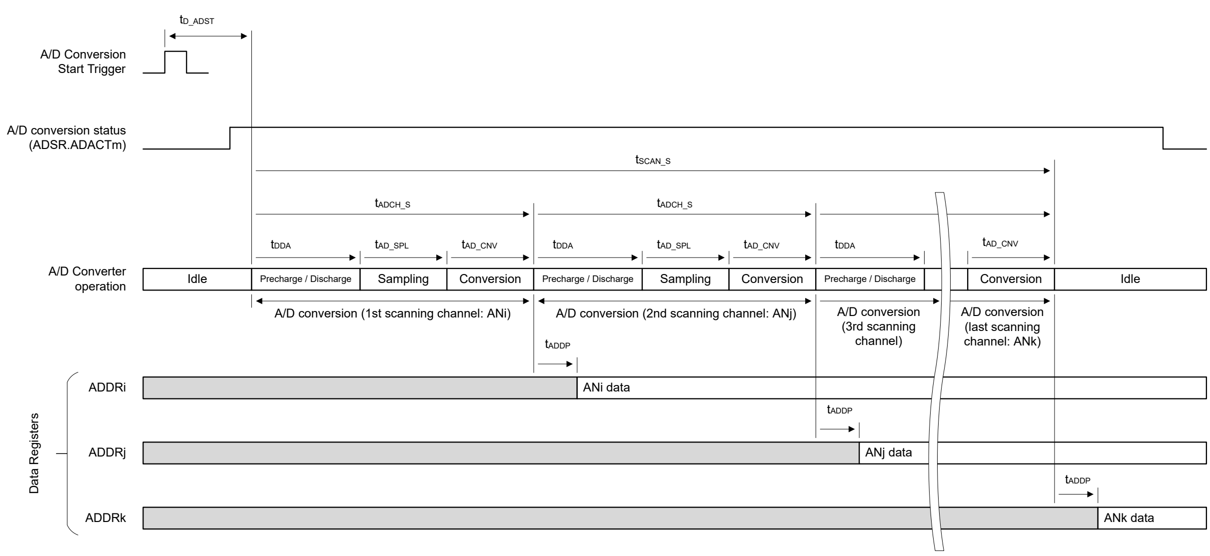
Note 2. It does not include the channel-dedicated sample-and-hold processing time.

Note 3. It does not include the A/D conversion data processing time.

Note 4. This is the sum of the channel conversion times (t_{ADCH_N}) that calculated from the conversion settings for each analog channel assigned to the scan group. If channel-dedicated sample-and-hold circuits are used, channel-dedicated sample-and-hold processing times (t_{SH_SPL} and t_{SH_HLD}) are also added.

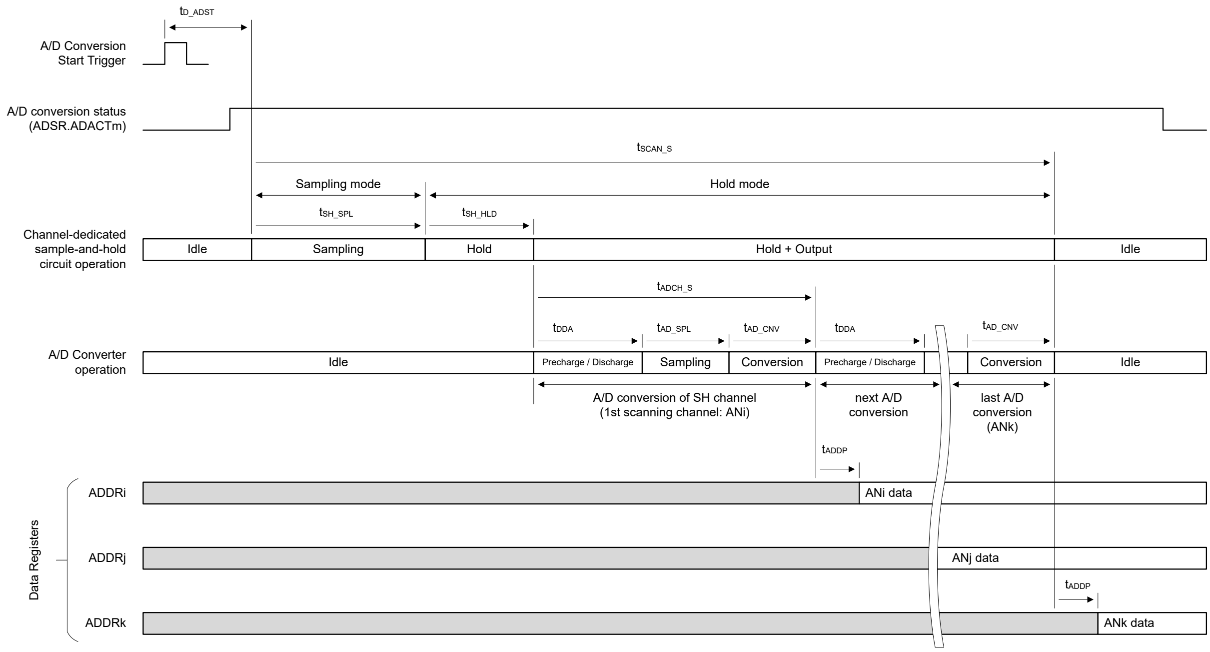
Note 5. This is the sum of the channel conversion times (t_{ADCH_O}) that calculated from the conversion settings for each analog channel assigned to the scan group.

Note 6. This is the sum of the oversampling period (t_{HY_OS}) that calculated from the conversion settings for each analog channel assigned to the scan group.



Note: i, j, k : Any analog channel number.
m = 0, 1

Figure 53.57 A/D conversion processing time (SAR mode)



Note: i, j, k : Any analog channel number.
m = 0, 1

Figure 53.58 A/D conversion processing time (SAR mode with channel-dedicated sample-and-hold circuit)

Note: i, j, k : Any analog channel number.
m = 0, 1

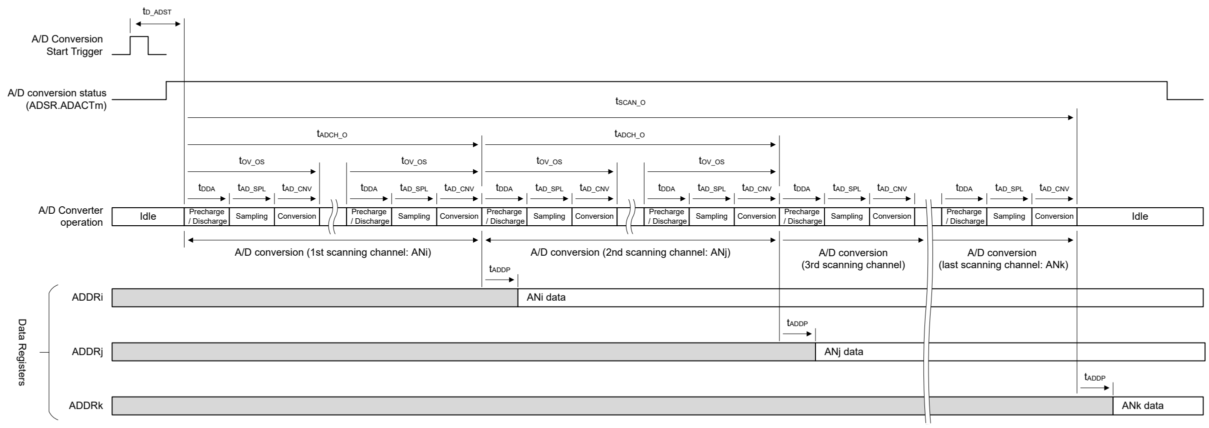
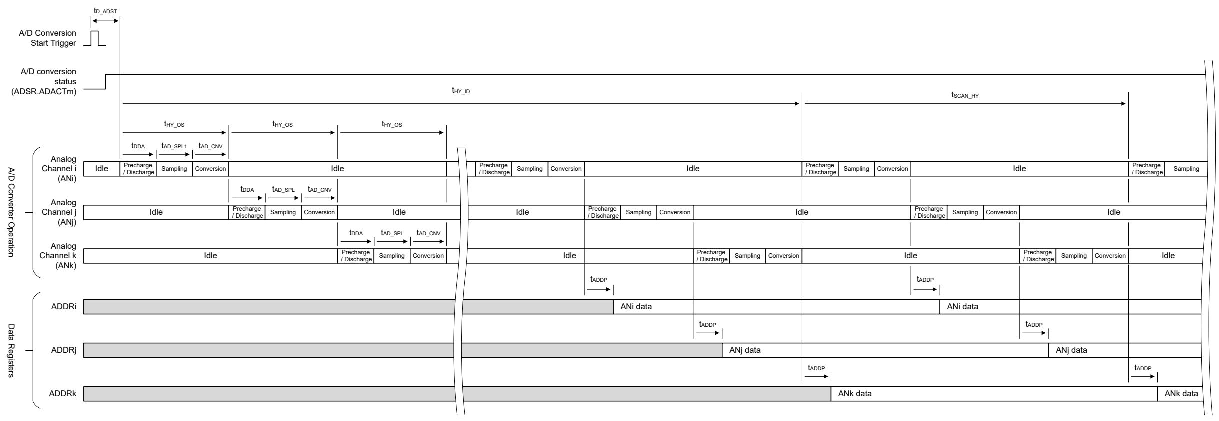
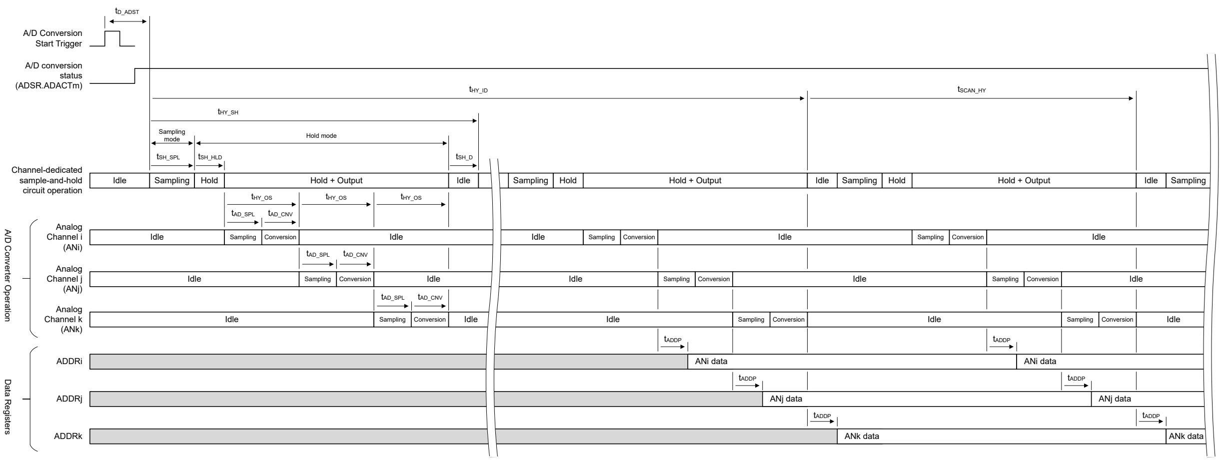


Figure 53.59 A/D conversion processing time (Oversampling mode)



Note: i, j, k : Any analog channel number.
m = 0, 1

Figure 53.60 A/D conversion processing time (Hybrid mode)



Note: i, j, k : Any analog channel number.
m = 0, 1

Figure 53.61 A/D conversion processing time (Hybrid mode with channel-dedicated sample-and-hold circuit)

53.10.3 Scan End Processing Time

The scan end processing time and the forcibly stop processing time are shown in Table 53.63, Figure 53.62, Figure 53.63.

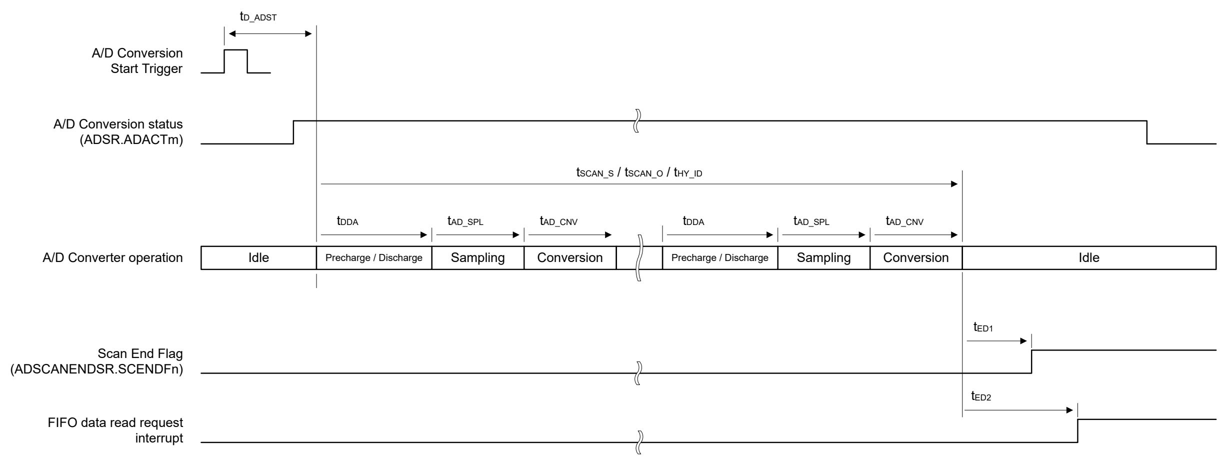
Table 53.63 Scan end processing time and forcibly stop processing time (1 of 2)

Item		Symbol	Processing Time
Scan end processing time	Until reflected in the status register*1	t_{ED1}	t_{ADDP}
	Until output the FIFO data read request interrupt	t_{ED2}	t_{ED1} + 2 PCLKA

Table 53.63 Scan end processing time and forcibly stop processing time (2 of 2)

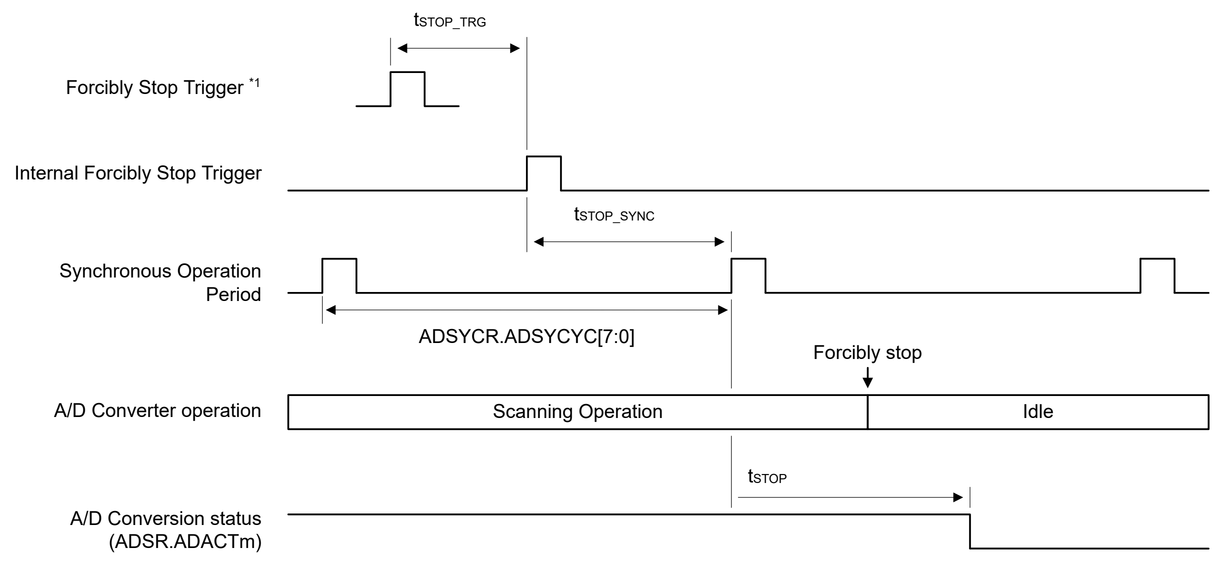
Item		Symbol	Processing Time
Forcibly Stop processing time	Forcibly stop trigger input processing time	t_{STOP_TRG}	[When ADCLK = PCLKA/1 is set]*2 ● Number of access cycles to I/O registers*3 + 1 PCLKA [Other than above] ● (Number of access cycles to I/O registers*3) + 1 PCLKA + (3 to 4 ADCLK)
	Wait time for synchronous operation	t_{STOP_SYNC}	[When synchronous operation is disabled] (ADSYCR.ADSYDISm = 1) ● 0 [When synchronous operation is enabled] (ADSYCR.ADSYDISm = 0) ● 0 to ADSYCR.ADSYCYC[7:0] × ADCLK
	Forcibly stop processing time	t_{STOP}	[When ADCLK = PCLKA/1 is set]*2 (ADSYCR.ADSYDISm = 1) ● 4 PCLKA (ADSYCR.ADSYDISm = 0) ● 3 PCLKA [Other than above] (ADSYCR.ADSYDISm = 1) ● 4 PCLKA + (2 to 3 PCLKA) (ADSYCR.ADSYDISm = 0) ● 3 PCLKA + (2 to 3 PCLKA)

Note: m = 0, 1
Note 1. This is the time it takes for ADSCANENDSR.SCENDFn = 1 (n = 0 to 8) or until an end-of-scan interrupt is generated.
Note 2. When ADCLKCR.CLKSEL[1:0] = 10b and ADCLKCR.DIVR[2:0] = 000b are set.
Note 3. It is the number of access cycles to ADSTOPR register. For more details about the number of access cycles for I/O registers, see section x, I/O Registers [DEADBEEF].



Note: n = 0 to 8, m = 0, 1

Figure 53.62 Scan end processing time (SAR/Oversampling/Hybrid mode – Single scan mode)



Note 1. It is generated by writing to ADSTOPR.ADSTOPm = 1. (m = 0, 1)

Figure 53.63 Forcibly stop processing time (SAR/Oversampling/Hybrid mode – Single scan mode)

53.11 Usage Notes

53.11.1 Prohibition of Changing the Operation Settings During A/D Conversion Operation

The registers related to the operation setting of the A/D converter should be set while all A/D converters are stopped (ADSR.ADACTm = 0 and ADSR.CALACTm = 0 (m = 0, 1)). Changing (writing) of registers except for those listed below is prohibited during A/D conversion. If the operation setting is changed during A/D conversion, operation is not guaranteed.

[Registers that can be written while the A/D converter is operating]

- Status clear registers
  - Status clear registers related to A/D converter operation (ADERSCR, ADCALENDSCR, ADSCANENDSCR)
  - A/D conversion overflow status clear registers (ADOVFERSCR, ADOVFCHSCR0, ADOVFEXSCR)
  - Limiter clip status clear register (ADLIMGRSCR, ADLIMCHSCR0, ADLIMEXSCR)
  - Compare match status clear register (ADCMPTBSCR, ADCMPCHSCR0, ADCMPEXSCR)
  - FIFO error status clear register (ADFIFOERSCR)
- Software trigger register (ADSYSTR, ADSTRn (n = 0 to 8))
- A/D converter stop register (ADSTOPR)
- A/D converter start trigger enable register (ADTRGENR) *1

Note 1. Writing during operation is only permitted for disabling the trigger input (ADTRGENR.STTRGENn = 0 (n = 0 to 8)) in order to stop the A/D conversion. To avoid unintended operation, do not change the trigger input to Enable (ADTRGENR.STTRGENn = 1) during operation.

53.11.2 Usage Notes on Forced Stop of A/D Conversion

If you want to forcibly stop the A/D conversion operation, follow the procedure in section 53.6.4. Force Stops the A/D Conversion Operation.

53.11.3 Usage Notes on A/D Data Registers

When A/D conversion is performed multiple times on the same analog channel, A/D data register ADDRn (n = 0 to 22) or A/D extended analog data register ADEXDRm (m = 0, 1, 4, 5, 6, 8, 9, 16, 17, 18, 20, 21, 22) corresponding to that analog channel is overwritten at the later A/D conversion.

If you intend to keep the data for each A/D conversion for the same analog channel, perform one of the following methods.

- Read out data from registers at each scanning operation
  - At the end of scanning each scan group, read out the A/D conversion data of the analog channel that was subject to A/D conversion from ADDRn or ADEXDRm.
  - This method is effective if A/D conversion of the same analog channel is not performed more than once in the same scan group, and A/D conversion data can be read from the registers before the next scanning operation.
- Use FIFO function to hold data for each A/D conversion
  - By using FIFO, multiple A/D conversion data for the same analog channel can be kept.
  - Read out the A/D conversion data before FIFO overflow.

53.11.4 Settings for the Module-stop Function

The Module Stop Control Register can enable or disable ADC16H operation. The ADC16H is initially stopped after a reset. The registers become accessible on release from the module-stop state.

See section 53.11.6. Restrictions on Entering and Releasing the Low-Power States in connection with this usage note.

53.11.5 Settings for the Software Standby Mode

Before entering Software Standby mode, stop ADC16H operation according to the procedures described in Figure 53.64 and Figure 53.65.

Note: For details on the wait time, see section 53.6.4.2. Waiting Time after Disabling Trigger Input for Forced Stop Processing.

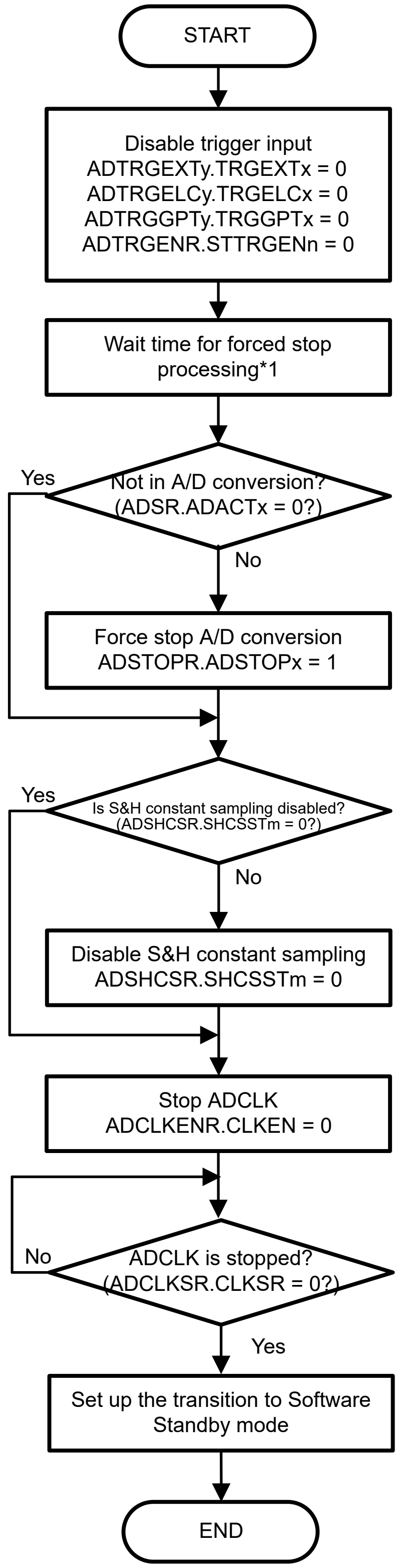


Figure 53.64 Example procedure for entering Software Standby mode

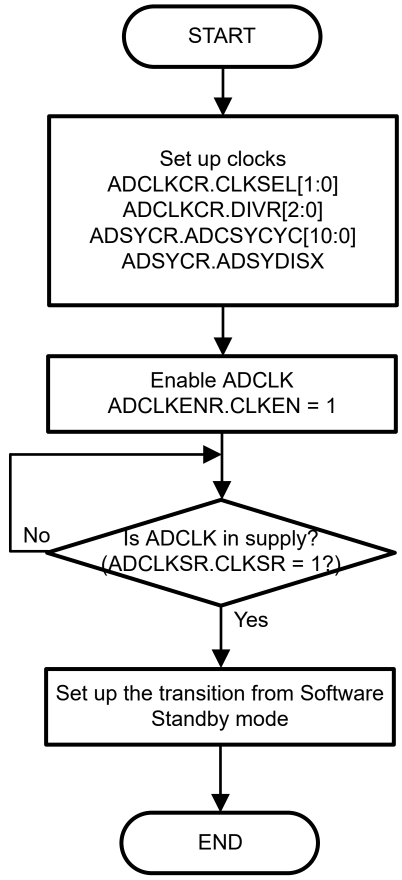


Figure 53.65 Example procedure for resuming Software Standby mode

53.11.6 Restrictions on Entering and Releasing the Low-Power States

Before entering the module-stop state or Software Standby mode, be sure to stop A/D conversion. In Addition, configure the A/D conversion not to start during the entering to the low power state.

To operate the A/D converter after the module-stop state or software standby mode is released, wait for the operation stabilization time specified in the Electrical Characteristics, execute self-calibration operation, and then start A/D conversion.

Also, when releasing the module-stop state again after entering to the module-stop state, make sure the shutdown time specified in the Electrical Characteristics has elapsed before releasing the module stop state.

Operations are not guaranteed if these restriction are violated.

53.11.7 Notes on Board Design

(1) Protection Circuit

To prevent the analog input pins from being destroyed by abnormal voltage such as excessive surge, take the following measures.

- Insert a capacitor between the power pin and the ground pin. The insertion points are below.
  - between AVCC0 and AVSS0
  - between VREFH0 and VREFL0
  - between VREFH and VREFL

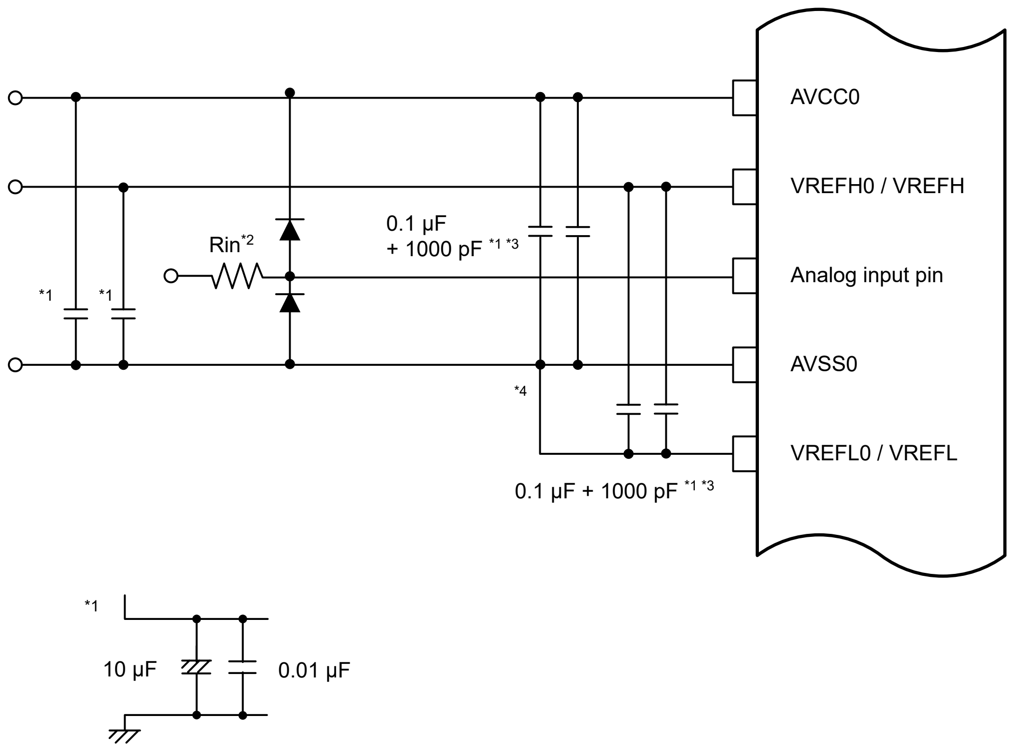
- Connect a protection circuit to protect the analog input pins.

An example of a protection circuit is shown in Figure 53.66.

(2) Board design to ensure A/D conversion accuracy

In order to ensure the accuracy of A/D conversion, design the board considering the following:

- Analog circuits and digital circuits should be separated from each other as far as possible.
- Analog circuit signal lines and digital circuit signal lines should not intersect or be placed near each other.
- The analog input, analog reference power supply (VREFH0, VREFH), analog reference ground (VREFL0, VREFL), and analog power supply (AVCC0) should be separated from digital circuits using the analog ground (AVSS0).
- The analog ground (AVSS0) should be connected to a stable digital ground (VSS) on the board (single-point ground plane connection).
- Place a capacitor for noise filters between analog power supply pin and the ground pin and between analog reference voltage supply pin and the ground pin close to the pin and connect them. Also, connect AVSS0, VREFL0 and VREFL pins to the analog ground on the board as close to the pins as possible. An example of connection is shown in Figure 53.66.



Note 1. The values shown here are for reference only.
The appropriate values depend on your board design, operating conditions, etc.
Note 2. Rin: Signal source impedance
Note 3. To ensure the better A/D conversion accuracy, these capacitors must be place as close as possible to the appropriate pins.
Note 4. Connect to the analog ground of the board at the position as close as possible to the capacitor for noise filtering that ensure the A/D conversion accuracy.

Figure 53.66 Example of protection circuit and noise prevention

53.11.8 Notes on Synchronous Operation

Synchronous operation function is disabled in the initial state after reset released. If using synchronous operation function, follow the restrictions described in section 53.3.19. Synchronous Operation.

53.11.9 Notes on Channel-dedicated Sample-and-hold Circuit

When using channel-dedicated sample-and-hold circuit, follow the restrictions. For details, see section 53.3.16. Channel-dedicated Sample-and-hold Circuit.

53.11.10 Restrictions on Analog Channel Shared Among Multiple A/D Converters (ADC16H) and High-Speed Analog Comparator (ACMPHS)

To avoid degradation of the accuracy of the A/D conversion result, A/D conversion of the same analog channel (same analog signal source) from both ADC0 and ADC1 is prohibited. Also, an analog source during A/D conversion should not be selected as an input for ACMPHS.

If this restriction is violated, the A/D conversion results are not guaranteed because the A/D conversion accuracy of the target analog channel may deteriorate significantly.

For the A/D conversion targets in Table 53.64, unit 0 and unit 1 cannot perform A/D conversion at the same time.

Table 53.64 A/D conversion targets that are mutually exclusive with each other

A/D conversion target	
Unit 0	Unit 1
AN016	
AN017	
AN018	
AN019	
AN020	
AN021	
AN022	
Temperature sensor	
Internal reference voltage	
VBATT 1/6 voltage monitor output	
D/A converter channel 0	
D/A converter channel 1	

The A/D conversion targets in Table 53.65 should not be selected as ACMPHS input during A/D conversion, because these pins are multiplexed with the ADC16H and ACMPHS.

Table 53.65 A/D conversion targets that are mutually exclusive with ACMPHS (1 of 2)

A/D conversion target		
Unit 0	Unit 1	ACMPHS
AN000	—	ACMPHS0.IVCMP2
AN001	—	ACMPHS0.IVCMP3
AN002	—	ACMPHS1.IVCMP2
AN003	—	ACMPHS1.IVCMP3
AN004	—	ACMPHS2.IVCMP2
AN005	—	ACMPHS2.IVCMP3
— —	AN006 AN007	ACMPHS3.IVCMP2 ACMPHS3.IVCMP3
— —	AN008 AN009	ACMPHS0 to ACMPHS3.IVREF0 ACMPHS0 to ACMPHS3.IVREF1
AN014	—	ACMPHS3.IVCMP0

Table 53.65 A/D conversion targets that are mutually exclusive with ACMPHS (2 of 2)

A/D conversion target		
Unit 0	**Unit 1**	**ACMPHS**
—	AN015	ACMPHS2.IVCMP0
AN016	AN016	ACMPHS1.IVCMP0
AN017	AN017	ACMPHS0.IVCMP0
Internal reference voltage	Internal reference voltage	ACMPHS0 to ACMPHS3.IVREF2
D/A converter channel 0	D/A converter channel 0	ACMPHS0 to ACMPHS3.IVREF3
D/A converter channel 1	D/A converter channel 1	ACMPHS0 to ACMPHS3.IVCMP1

53.11.11 Notes on A/D Conversion Start Trigger

(1) Restrictions on A/D conversion start triggers for the same scan group

A/D conversion start triggers for the same scan group are not accepted until scanning operation of the scan group is completed, except Background continuous scan mode and One-channel continuous scan mode (in this case, A/D converter start triggers are ignored).

An A/D conversion trigger for the same scan group should be input after the scanning operation of its scan group is completed (ADSCANENDSR.SCENDFn = 1 (n = 0 to 8)) and after 6 PCLKA clock cycles or more have elapsed.

(2) Prohibition on calibration and A/D conversion

It is prohibited to start calibration and A/D conversion in the ADC unit during sleep operation. In addition, it is prohibited to set the ADC unit to Sleep mode during calibration and A/D conversion operations.

53.11.12 Notes on Self-calibration

Observe the restrictions on self-calibration function. For details, see section 53.3.8. Self-calibration.

53.11.13 Notes on Group Priority Operation

Observe the restrictions when using group priority operation. For details, see section 53.3.18. Group Priority Operation.

53.11.14 Restrictions on SAR Mode

(1) Notes on resolution and accuracy

In SAR mode, the A/D converter functions as a 12-bit A/D converter. Although the data output from the A/D converter is internally extended to 16-bit data length for calculation processing, even if 14-bit or 16-bit length data format is selected, the resolution and accuracy exceeding 12 bits are not guaranteed for that A/D conversion data. Select 14-bit or 16-bit length data format in SAR mode if you want to increase the computational resolution, such as when using A/D-converted value addition/averaging function or User’s Gain/User’s Offset function.

(2) Restrictions on digital filter function

The digital filter is prohibited in SAR mode. Operation is not guaranteed when the digital filter function is enabled.

53.11.15 Restrictions on Oversampling Mode

(1) Restrictions on digital filter configuration

In Oversampling mode, the use of the digital filter function is mandatory. If A/D conversion is performed without using the digital filter function, the A/D conversion characteristics are not guaranteed.

(2) Restrictions on trigger interval in One-channel continuous scan mode

In Oversampling mode – One-channel continuous scan mode, A/D conversion start trigger should be input after the following intervals.

[Required intervals for A/D conversion start trigger input]

- When trigger delay function is not used, or software trigger is used: 8 ADCLK or more
- When trigger delay function is used: (8 + (the setting value of ADTRGDLRm.TRGDLYn[7:0])) × ADCLK, or more

Note: m = 0 to 4, n = 0 to 8

If this restriction is violated, the A/D conversion start trigger is not accepted and is ignored.

53.11.16 Restrictions on Hybrid Mode

(1) Prohibition of scanning operation with only one channel

In Hybrid mode, be sure to assign two or more virtual channels to one scan group (up to 4 virtual channels). Scanning operation with only one virtual channel assigned to one scan group is prohibited. If this restriction is violated, operation is not guaranteed.

(2) Restrictions on virtual channel configuration

In Hybrid mode, do not assign the same analog channel to multiple virtual channels that are assigned to the same scan group. If this restriction is violated, A/D conversion result is not guaranteed.

(3) Restrictions on digital filter configuration

In Hybrid mode, the use of the digital filter function is mandatory. If A/D conversion in Hybrid mode is performed without using the digital filter function, the A/D conversion characteristics are not guaranteed.

In addition, since multiple digital filters are used simultaneously in Hybrid mode, the digital filters used for virtual channels in the same scan group must be set to be exclusive. The digital filters selected by the ADDOPCRAn.DFSEL[2:0] (n = 0 to 32) bits must be set to be exclusive among virtual channels in the same scan group. If the same digital filter is selected from multiple virtual channels in the same scan group, operation is not guaranteed.

(4) Restrictions on A/D-converted value addition/averaging function

When using A/D-converted value addition/averaging function in Hybrid mode, set the number of addition/averaging times for all virtual channels in the same scan group to be the same. If this restriction is violated, operation is not guaranteed.

(5) Restrictions on trigger interval in Background continuous scan mode

In Hybrid mode – Background continuous scan mode, A/D conversion start trigger should be input after the following interval or more.

[Required Interval for A/D Conversion Start Trigger Input]

- When trigger delay function is not used, or software trigger is used: 8 ADCLK, or more
- When trigger delay function is used: (8 + (the setting value of ADTRGDLRm.TRGDLYn[7:0])) × ADCLK, or more

Note: m = 0 to 4, n = 0 to 8

If this restriction is violated, the A/D conversion start trigger is not accepted and is ignored.

(6) Restrictions on Fixed-channel continuous scan mode

Observe the restrictions on Hybrid mode – Fixed-channel continuous scan mode.

- Fixed channels in Fixed-channel continuous scan mode use the scan group and virtual channels determined for each unit. See Table 53.66 for the scan group and virtual channels that can be assigned to fixed channels. The scan group setting for virtual channels that can be assigned to a fixed channel should be set to the no setting (ADCHCRn.SGSEL[4:0] = 0x00 (n = 0 to 32)).

Table 53.66 Virtual channel number of Fixed-channel continuous scan mode (1 of 2)

Unit	Unit 0	Unit 1
Scan group	Scan group 0	Scan group 1
1st Fixed channel	Virtual channel 0	Virtual channel 4
2nd Fixed channel	Virtual channel 1	Virtual channel 5

Table 53.66 Virtual channel number of Fixed-channel continuous scan mode (2 of 2)

Unit	Unit 0	Unit 1
3rd Fixed channel	Virtual channel 2	Virtual channel 6

- The scan period must be set to a number equal to or greater than the total number of conversions for fixed channels and non-fixed channels. If the scan period is set to less than the total number of conversions for fixed and non-fixed channels, conversion cycles beyond the scan period are not subject to A/D conversion.
- To prevent scan cycle variations due to non-fixed channels, set the same time for all virtual channel sampling states (ADSSTRx.SSTy[9:0]) used for non-fixed channels.
- Before starting scan in this mode, it is necessary to set ADMDR.ADMDm[3:0] = 0000 (m = 0, 1) and perform gain/offset calibration for SAR. The gain/offset correction coefficients for SAR are prepared separately from the correction coefficients for Hybrid that are calculated by the gain/offset calibration performed in this mode. These correction factors are used to correct the conversion results for fixed and non-fixed channels respectively.
- The use of fixed channels and non-fixed channels in fixed channel scanning in addition/average mode is prohibited.
- When using fixed-channel continuous scan mode, also perform gain/offset calibration for SAR mode (single scan mode).

For more details on Fixed-channel continuous mode, see section 53.3.7.9. Hybrid Mode – Fixed-channel Continuous Scan Mode.

(7) Restrictions on High Accuracy Mode

In Hybrid mode, enable High accuracy mode.

53.11.17 Restrictions on Composite Compare Match

The use of a common compare table for multi units is prohibited for composite compare match function. If selected, the composite condition MATCH interrupt is asserted when the condition is satisfied at any unit. When the condition is not satisfied for all units, the compound condition UNMATCH interrupt is asserted.

53.11.18 Restrictions on High Accuracy Mode Setting Register

When the High accuracy mode is used (ADACMDR.ADHACMDn (n = 0,1) = 1), observe the following setting restrictions.

- ADSSTRn.SSTm[9:0] ≥ ADCNVSTR.CSTn (n = 0, 1) [5:0] × 1.6
- ADSSTRn.SSTm[9:0] ≥ 10’h4
- ADSSTRn.SSTm[9:0] is set to a multiple of 2
- When synchronous operation is used, the synchronous operation period should be set as follows.
  - ADSYCR.ADSYCYC[7:0] × i = ADSSTRn.SSTm[9:0] + ADCNVSTR.CSTn[5:0]
    (i: Any even number or equal to 1 (i = 1, 2, 4, 6…))
- High accuracy mode can only be used with SAR mode or Hybrid mode. The scan modes supported by high accuracy mode are shown in Table 53.67.

Table 53.67 Scan modes supported by high accuracy mode

Operation mode of A/D converter	High accuracy mode	
	Disabled (ADACMDR.ADHACMDn = 0)	Enabled (ADACMDR.ADHACMDn = 1)
SAR mode	✓	✓
Oversampling mode	✓	—
Hybrid mode	—	✓

53.11.19 Restrictions on Measuring of VBATT Voltage Monitor Output

When measuring VBATT 1/6 voltage monitor output, perform two or more A/D conversions and ignore the first conversion result, or perform multiple A/D conversions and use the average of the conversion results.

53.11.20 Sampling Error and Sampling Time Estimation

This subsection describes how to estimate the sampling error and the sampling time.

The sampling time required in A/D conversion for externally input analog signal is determined by the charging time to the sampling capacitance in A/D converter. The simplified model of the circuit is shown in Figure 53.67.

The sampling time can be roughly estimated by the following equation.

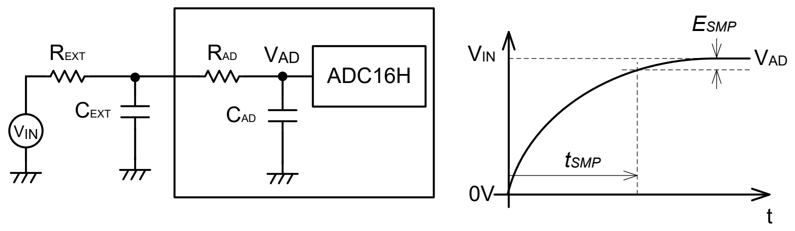


Figure 53.67 Simplified model of sample-and-hold circuit for A/D converter

You can estimate the sampling error (0 to 1, it is 1 when V_{IN}=VREFH0/VREFH). Set sampling time (t_{SMP}) so that sampling error (E_{SMP}) is within the expected range.

[Estimation Equation for Sampling Error]

$$E_{SMP}(t) = \frac{\dfrac{k_1 C_{AD}}{C_{EXT} + k_1 C_{AD}}}{1 - \dfrac{C_{EXT}}{C_{EXT} + k_1 C_{AD}} e^{\frac{-t_{SCAN} + t_{SMP}/k_2\left(1 - \frac{\tau_{EXT}}{\tau}\right)}{\tau_{EXT}}}} e^{\frac{-t_{SMP}/k_2}{\tau}}$$

$$\tau = R_{EXT}(C_{EXT} + k_1 C_{AD}) + R_{AD} k_1 C_{AD}$$

$$\tau_{EXT} = R_{EXT} C_{EXT}$$

E_{SMP}: Estimated sampling error

t_{SMP}: Sampling time

C_{EXT}: External capacitance (pin capacitance + PCB parasitic capacitance)

C_{AD}: Internal sampling capacitance

R_{EXT}: External input signal source Impedance

R_{AD}: Internal resistance

k_1,k_2: Correction coefficients according to operation mode

t_{SCAN}: Channel scan interval

Figure 53.68 shows definition of channel scan interval.

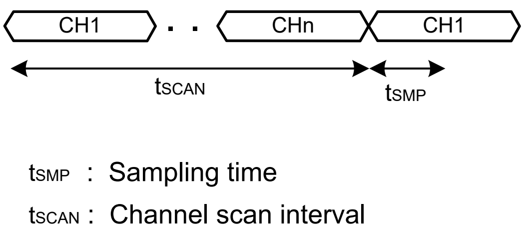


Figure 53.68 Definition of channel scan interval

Note: The typical values for each parameter are as follows.

These are reference value:

- Internal sampling capacitance (C_{AD}): 7 pF

Table 53.68 Reference values of k_1, k_2

Power	Scan mode	k_1		k_2
		High-speed channels (AN000 to AN011) Low-speed channels (AN016 to AN022)	Middle-speed channels (AN012 to AN015)	
AVCC: 2.7 to 3.63 V VCC: 2.7 to 3.63 V VREFH0/VREFH: 2.7 V to AVCC	SAR mode (Normal mode)	1.3	1.3	1.0
	SAR mode (High accuracy mode)	1.3	1.3	1.8
	Oversampling mode	1.0	1.0	1.0
	Hybrid mode	1.3	1.3	1.8
AVCC: 1.62 to 2.7 V VCC: 1.62 to 2.7 V VREFH0/VREFH: 1.62 V to AVCC	SAR mode (Normal mode)	1.2	1.3	1.0
	SAR mode (High accuracy mode)	1.2	1.3	1.8
	Oversampling mode	1.0	1.0	1.0
	Hybrid mode	1.2	1.3	1.8

Table 53.69 Reference values of RAD

Power	RAD (Ω)		
	High-speed channels (AN000 to AN011)	Middle-speed channels (AN012 to AN015)	Low-speed channels (AN016 to AN022)
AVCC: 2.7 to 3.63 V VCC: 2.7 to 3.63 V VREFH0/VREFH: 2.7 V to AVCC	700	2000	3000
	1500	3000	5000

[Usage Notes]

- Sampling time must be greater than or equal to the min value of the sampling time specified on section 62, Electrical Characteristics and restrictions in this chapter.
- The equation described in this subsection is simplified in view of general use case. It does not guarantee the accurate sampling error and sampling time. You should only use it to estimate the roughly sampling error and sampling time.

53.11.21 Port Settings When Using the ADC16H Input

When using the high-speed and middle-speed channels (AN000 to AN015), do not use PORT0 as digital output ports. Renesas recommends that you do not use the digital output that is also used as the AD analog input if low-speed channel (AN016 to AN022) is used. If the digital output that is also used as the AD analog input is used for output signals, perform A/D conversion several times, eliminate the maximum and minimum values, and obtain the average of the other results.

53.11.22 Notes on Simultaneous Operation with Other Modules

The following combinations may degrade the ADC accuracy depending on the specific timing of operation.

- All channels of ADC unit 1 when DAC12 is operated.
- All channels of ADC unit 0 and 1 when there is access to the external bus.
- Specific channels of ADC unit 0 and 1 when ACMPHS is operated. For details, see Table 53.65.

To mitigate deterioration, it is recommended to take following measures.

- Performing A/D conversion with average mode.
- For even better results, performing A/D conversion several times, eliminate the maximum and minimum values, and obtain the average of the other results.

54. 12-Bit D/A Converter (DAC12)

54.1 Overview

This is the DAC_B version of the DAC12 peripheral module. DAC_B is referred to as DAC12 in this chapter.

The MCU provides a 12-bit D/A Converter (DAC12). Table 54.1 lists the DAC12 specifications, Figure 54.1 shows a block diagram, and Table 54.2 lists the I/O pins.

Table 54.1 DAC12 specifications

Parameter	Specifications
Resolution	12 bits
Output channels	2 channels
Module-stop function	Module-stop state can be set to reduce power consumption
Event link function (input)	The DA0 and DA1 conversion can be started on input of an event signal
Destination of D/A output control function	Controls whether the output to the external pin or to the internal modules (ACMPHS and ADC16H) is used
TrustZone Filter	Security and Privilege attribution can be set

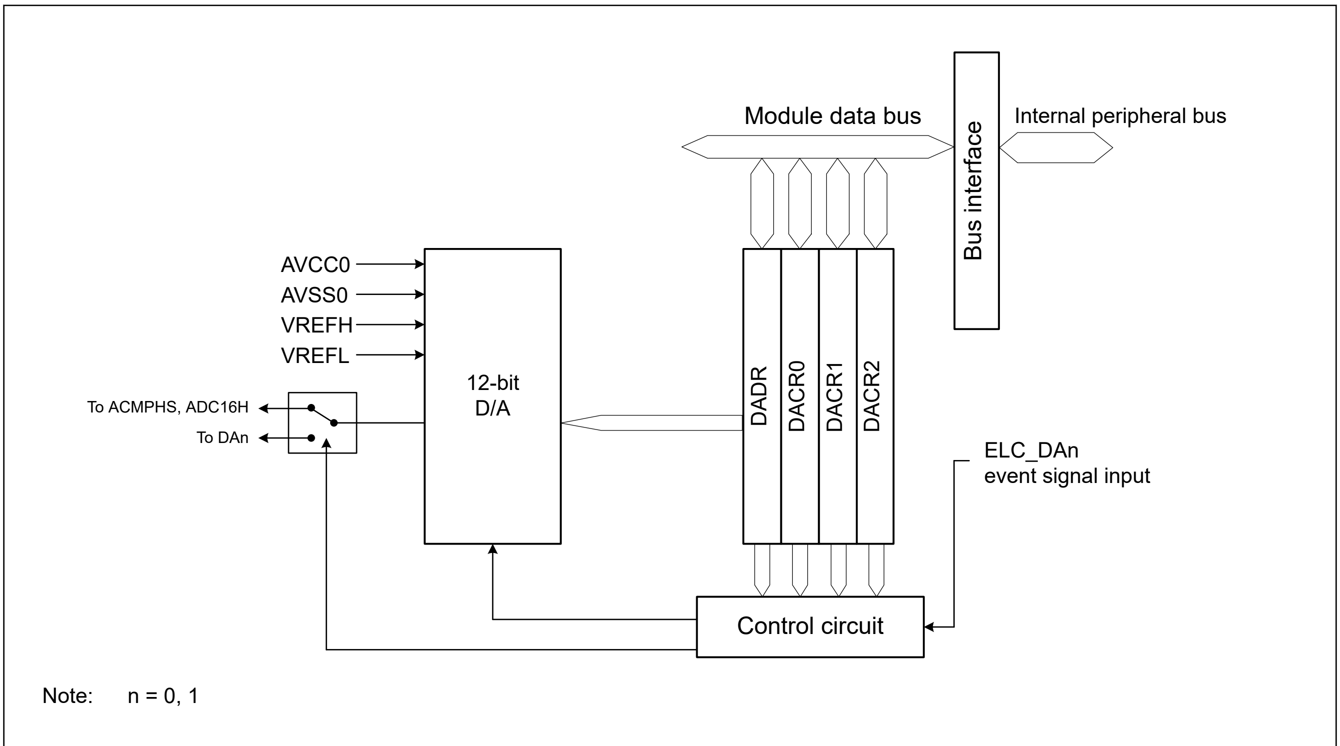


Figure 54.1 DAC12 block diagram

Table 54.2 lists the pin configuration of the DAC12.

Table 54.2 DAC12 I/O pins (1 of 2)

Pin name	I/O	Function
AVCC0	Input	• Analog power and analog reference top voltage supply pin for ADC16H and DAC12. • Connect to VCC when these modules are not used.
AVSS0	Input	• Analog ground and analog reference ground supply pin for ADC16H and DAC12. • Connect to VSS when these modules are not used.
VREFH	Input	Analog reference top voltage supply pin for the ADC16H (unit 1) and the DAC12
VREFL	Input	Analog reference ground pin for the ADC16H (unit 1) and the DAC12

Table 54.2 DAC12 I/O pins (2 of 2)

Pin name	I/O	Function
DA0	Output	Channel 0 output pin for the analog signals processed by the DAC12
DA1	Output	Channel 1 output pin for the analog signals processed by the DAC12

54.2 Register Descriptions

54.2.1 DADR : D/A Data Register

Base address: DACn_B = 0x4023_3000 + 0x0100 × n (n = 0, 1)
DACn_B_NS = 0x5023_3000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0000

Bit position:	15															0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: S-TYPE-3, P-TYPE-3

DADR register is 16-bit read/write registers that store data for D/A conversion. When an analog output is enabled, the values in DADR are converted and output to the analog output pins.

12-bit data can be formatted as left- or right-justified in the DACR1.DPSEL bit setting. In right-justified format (DACR1.DPSEL = 0), the lower 12 bits, [11:0], are valid. In left-justified format (DACR1.DPSEL = 1), the upper 12 bits, [15:4], are valid.

54.2.2 DACR0 : D/A Control 0 Register

Base address: DACn_B = 0x4023_3000 + 0x0100 × n (n = 0, 1)
DACn_B_NS = 0x5023_3000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DAOU TDIS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DAE*1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DACE N
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DACEN	D/A Output Enable 0: Disable analog output of channel n (DAn) 1: Enable D/A conversion of channel n (DAn)	R/W
14:1	—	These bits are read as 0. The write value should be 0.	R/W
15	DAE*1	D/A Enable R/W 0: Control D/A conversion of channels 0 and 1 individually 1: Control D/A conversion of channels 0 and 1 collectively	R/W
30:16	—	These bits are read as 0. The write value should be 0.	R/W
31	DAOUTDIS	Analog Output Disables 0: The output to pin is enabled and the output to comparator is disabled 1: The output to pin is disabled and the output to comparator is enabled	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. This bit controls D/A conversion in combination with the DACR0.DACEN bit (n = 0, 1), which controls the output of the conversion results. For details, see Table 54.3.

Table 54.3 D/A conversion controls

DACR0.DAE of DAC120	DACR0.DAE of DAC121	DACR0.DACEN of DAC120	DACR0.DACEN of DAC121	Description
0	0	0	0	Disable D/A conversion and analog output pins (DA0, DA1)*1
0	0	1	0	● Enable D/A conversion of channel 0 and disable D/A conversion of channel 1 ● Enable analog output of channel 0 (DA0) and disable analog output of channel 1 (DA1)*1
0	0	0	1	● Disable D/A conversion of channel 0 and enable D/A conversion of channel 1 ● Disable analog output of channel 0 (DA0)*1 and enable analog output of channel 1 (DA1)
0	0	1	1	● Enable D/A conversion of channels 0 and 1 ● Enable analog output of channels 0 and 1 (DA0, DA1)
1*2	x	x	x	● Enable D/A conversion of channels 0 and 1 ● Collective enable analog output of channels 0 and 1 (DA0, DA1)
x	1*2	x	x	● Enable D/A conversion of channels 0 and 1 ● Collective enable analog output of channels 0 and 1 (DA0, DA1)

Note: x: Don't care
Note 1. When analog output is disabled, the analog output signal is placed in the Hi-Z state.
Note 2. When security attribution of DAC120 and DAC121 does not match, setting 1 to DAE bit has no effect on D/A conversion and analog output pins.

DACEN bit (D/A Output Enable)

The DACEN bit controls D/A conversion and analog output in combination with the DAE bit. For details, see Table 54.4.

When both the DACEN bit and DAE bit are 0, D/A conversion of channel n (n = 0, 1) is not processed, and no conversion result is output.

DAE bit (D/A Enable R/W)

The DAE bit controls D/A conversion and analog output in combination with the DACEN bit. For details, see Table 54.4.

DAOUTDIS bit (Analog Output Disables)

The DAOUTDIS bit controls the analog output destination of the DAC12. Set 0 to output to the pin or 1 to output to the comparator. It is not possible to output to both the pin and the comparator at the same time.

The event link function can be used to set the DACEN bit to 1. The DACR0.DACEN bit of DAC120 is set to 1 when the event specified in the ELSR12 register of the ELC (ELC_DA0 event) occurs, and output of the D/A conversion results starts. The DACR0.DACEN bit of DAC121 is set to 1 when the event specified in the ELSR13 register of the ELC (ELC_DA1 event) occurs, and output of the D/A conversion results starts.

Table 54.4 D/A conversion and analog output control

DACR0.DAE	DACR0.DACEN	DACR0.DAOUTDIS	Channel n operation	Analog external output of channel n*1	Analog internal output of channel n*2
0	0	x	Stop	Hi-Z	Hi-Z
	1	0	Run	D/A output	Hi-Z
		1	Run	Hi-Z	D/A output
1	x	0	Run	D/A output	Hi-Z
		1	Run	Hi-Z	D/A output

Note: x : Don't care
Note 1. Output to pin
Note 2. Output to ACMPHS and ADC16H

54.2.3 DACR1 : D/A Control 1 Register

Base address: DACn_B = 0x4023_3000 + 0x0100 × n (n = 0, 1)
DACn_B_NS = 0x5023_3000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DPSEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
16	DPSEL	DADR Format Select R/W 0: Right-justified format 1: Left-justified format	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

DPSEL bit (DADR Format Select R/W)

The DPSEL bit controls the data format of the DADR register.

54.2.4 DACR2 : D/A Control 2 Register

Base address: DACn_B = 0x4023_3000 + 0x0100 × n (n = 0, 1)
DACn_B_NS = 0x5023_3000 + 0x0100 × n (n = 0, 1)

Offset address: 0x000C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	OFSSEL	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	OFSSEL	DAC Operating Voltage Mode Selection 0: Normal voltage mode (VREFH ≥ 2.7 V) 1: Low voltage mode (VREFH < 2.7 V)	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

OFSSEL bit (DAC Operating Voltage Mode Selection)

The OFSSEL bit controls the optimization of the DAC output voltage according to the analog reference voltage VREFH. When the VREFH voltage is less than 2.7 V, set the OFSSEL bit to 1.

54.3 Operation

The DAC12 performs D/A conversion when the DACR0.DAE bit is 1 or the DACR0.DACEN bit is 1. The output destination of the D/A conversion result can be switched by the DACR0.DAOUTDIS bit value.

When the DACR0.DAOUTDIS bit is 0, the D/A conversion result is output to the DAn pin in the DAC output mode. When the DACR0.DAOUTDIS bit is 1, the D/A conversion result is output to the comparator in the comparator output mode.

The output value (reference) is expressed by the following formula:

$$\frac{\text{Setting in DADR}}{4096} \times \text{VREFH}$$

54.3.1 Configuration Update Transfer the Coefficient Value

Before using the DAC12, the coefficient value should be transferred from extra MRAM. See section 59.10. Configuration Update Transfer.

54.3.2 DAC Output Mode

This following example shows D/A conversion on channel n (n = 0, 1) in DAC output mode. Figure 54.2 shows the timing of this operation.

1. Set the DACR0.DAOUTDIS bit to 1.
2. Set the DACR1.DPSEL bit and the DACR2.OFSSEL bit as required. Set the D/A conversion initial data in the DADR register.
   If the DACR2.OFSSEL bit is set to 0, then set 0x0E0 to the DADR register, otherwise set 0x1F8 to the DADR register as an initial value of starting up D/A conversion.
3. After t_{SU} elapses from the DAOUTDIS bit setting, set the DACR0.DACEN bit or the DACR0.DAE bit to 1.
4. After t_{DISOUT} elapses, set the DACR0.DAOUTDIS bit to 0. The D/A conversion result of the initial value is output to the DAn pin after the time of $t_{DSLPUP2}$.
5. Set the data for D/A conversion to the DADR register.
6. After t_{DCONV2} elapses, the D/A conversion result is output to the DAn pin.

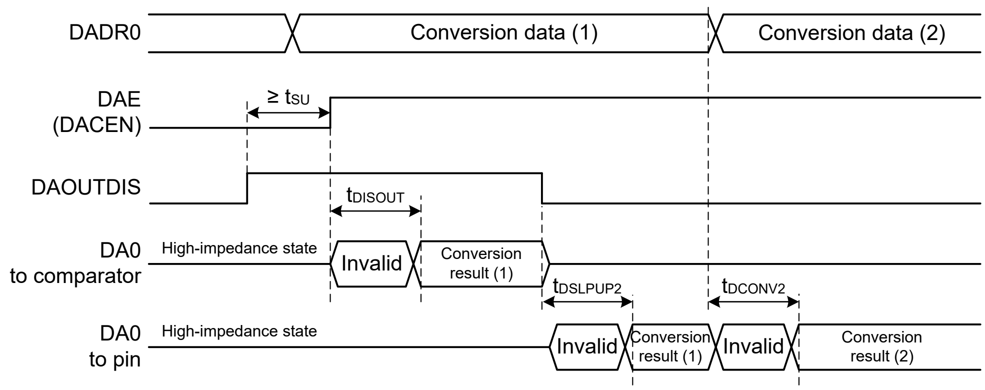


Figure 54.2 Example of DAC12 operation (DAC output mode)

54.3.3 Comparator Output Mode

This following example shows D/A conversion on channel n (n = 0, 1) in comparator output mode. Figure 54.3 shows the timing of this operation.

1. Set the DACR0.DAOUTDIS bit to 1.
2. Set the DACR1.DPSEL bit and the DACR2.OFSSEL bit as required. Set the D/A conversion initial data in the DADR register.
   If the DACR2.OFSSEL bit is set to 0, then set 0x0E0 to the DADR register, otherwise set 0x1F8 to the DADR register as an initial value of starting up D/A conversion.
3. After t_{SU} elapses from the DAOUTDIS bit setting, set the DACR0.DACEN bit or the DACR0.DAE bit to 1.

4. The D/A conversion result of the initial value is output to the comparator after the time of t_{DISOUT}.
5. Set the data for D/A conversion to the DADR register.
6. After t_{DCONV1} elapses, the D/A conversion result is output to the comparator.

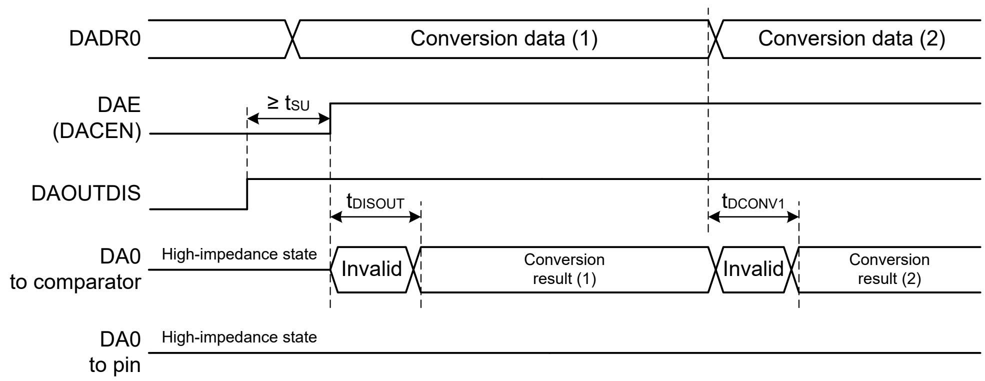


Figure 54.3 Example of DAC12 operation (comparator output mode)

54.4 Event Link Operation Setting Procedure

This section describes the procedures used in event link operation.

54.4.1 DA0 Event Link Operation Setting Procedure

To set up DA0 event link operation:

1. Set the DACR1.DPSEL bit to any value.
2. Set the DACR0.DACEN bit to 0.
3. Set the data for D/A conversion in the DADR.
4. Set the ELC_DA0 event signal to be linked to each peripheral module in the ELSR12 register.
5. Set the ELCR.ELCON bit to 1. This enables event link operation for all modules with the event link function selected.
6. Set the event output source module to activate the event link. After the event is output from the module, the DAC0DACR0.DACEN bit becomes 1, and D/A conversion starts on channel 0.
   The following time is required until the output value stabilizes.
   DAC output mode: 10.5µs
   Comparator output mode: 7µs
7. Set the ELSR12 register to 0x0000 to stop event link operation of DAC12 channel 0. All event link operation is stopped when the ELCR.ELCON bit is set to 0.

54.4.2 DA1 Event Link Operation Setting Procedure

To set up DA1 event link operation:

1. Set the DACR1.DPSEL bit to any value.
2. Set the DACR0.DACEN bit to 0.
3. Set the data for D/A conversion in the DADR.
4. Set the ELC_DA1 event signal to be linked to each peripheral module in the ELSR13 register.
5. Set the ELCR.ELCON bit to 1. This enables the event link operation for all modules with the event link function selected.
6. Set the event output source module to activate the event link. After the event is output from the module, the DACR0.DACEN bit becomes 1, and D/A conversion starts on channel 1.

The following time is required until the output value stabilizes.
DAC output mode: 10.5µs
Comparator output mode: 7µs

7. Set the ELSR13 register to 0x0000 to stop event link operation of DAC12 channel 1. All event link operation is stopped when the ELCR.ELCON bit is set to 0.

54.5 Analog Output Control by Security Attribution

Analog output to DA0/DA1 pin is controlled by the security attribution of Port and DAC12 as shown in Table 54.5.

Analog output to ACMPHS is not controlled by the security attribution.

Table 54.5 Analog output control by security attribution

Security Attribution			Analog Output	
DAC12 (PSARD)	**P014 (PmSAR (m = 0 to 9, A, B))**	**P015 (PmSAR (m = 0 to 9, A, B))**	**P014 pin output**	**P015 pin output**
0 (secure)	1 (non-secure)	1 (non-secure)	enable	enable
0 (secure)	1 (non-secure)	0 (secure)	enable	enable
0 (secure)	0 (secure)	1 (non-secure)	enable	enable
0 (secure)	0 (secure)	0 (secure)	enable	enable
1 (non-secure)	1 (non-secure)	1 (non-secure)	enable	enable
1 (non-secure)	1 (non-secure)	0 (secure)	enable	disable
1 (non-secure)	0 (secure)	1 (non-secure)	disable	enable
1 (non-secure)	0 (secure)	0 (secure)	disable	disable

54.6 Usage Notes on Event Link Operation

- When the event link function is used, set the DACR0.DAE bit to 0.
- When the event specified for the ELC_DA0 event signal is generated while writing to the DACR0.DACEN bit of DAC120 is performed, the write cycle is stopped, and the generated event takes precedence in setting the bit to 1.
- When the event specified for the ELC_DA1 event signal is generated while writing to the DACR0.DACEN bit of DAC121 is performed, the write cycle is stopped, and the generated event takes precedence in setting the bit to 1.

54.7 Usage Notes

54.7.1 Settings for the Module-Stop Function

DAC12 operation can be disabled or enabled using the Module Stop Control Register. The DAC12 is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Mode.

54.7.2 DAC12 Operation in the Module-Stop State

When the MCU enters the module-stop state with D/A conversion enabled, the D/A outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current must be reduced in the module-stop state, disable D/A conversion by setting the DACR0.DACEN and DACR0.DAE bits to 0.

54.7.3 DAC12 Operation in Software Standby Mode

When the MCU enters Software Standby mode with D/A conversion enabled, the D/A outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current must be reduced in Software Standby mode, disable D/A conversion by setting the DACR0.DACEN and DACR0.DAE bits to 0.

54.7.4 Constraint on Entering Deep Software Standby Mode

When the MCU enters Deep Software Standby mode with D/A conversion enabled, the D/A outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current must be reduced in Deep Software Standby mode, disable D/A conversion by setting the DACR0.DACEN and DACR0.DAE bits to 0.

55. Temperature Sensor (TSN)

55.1 Overview

The on-chip Temperature Sensor (TSN) determines and monitors the die temperature for reliable operation of the device. The sensor outputs a voltage directly proportional to the die temperature, and the relationship between the die temperature and the output voltage is fairly linear. The output voltage is provided to the ADC16H for conversion and can be further used by the end application.

Table 55.1 lists the TSN specifications, and Figure 55.1 shows a block diagram.

Table 55.1 TSN specifications

Item	Description
Temperature sensor voltage output	Temperature sensor outputs a voltage to the A/D converter
Module-stop function	Module-stop state can be set to reduce power consumption
Temperature sensor calibration data	Reference data measured for each chip at factory shipment is stored in a register
Temperature monitor reset source	The sensor outputs an abnormal temperature detection signal to the reset control circuit
TrustZone Filter	Security and Privilege attribution can be set

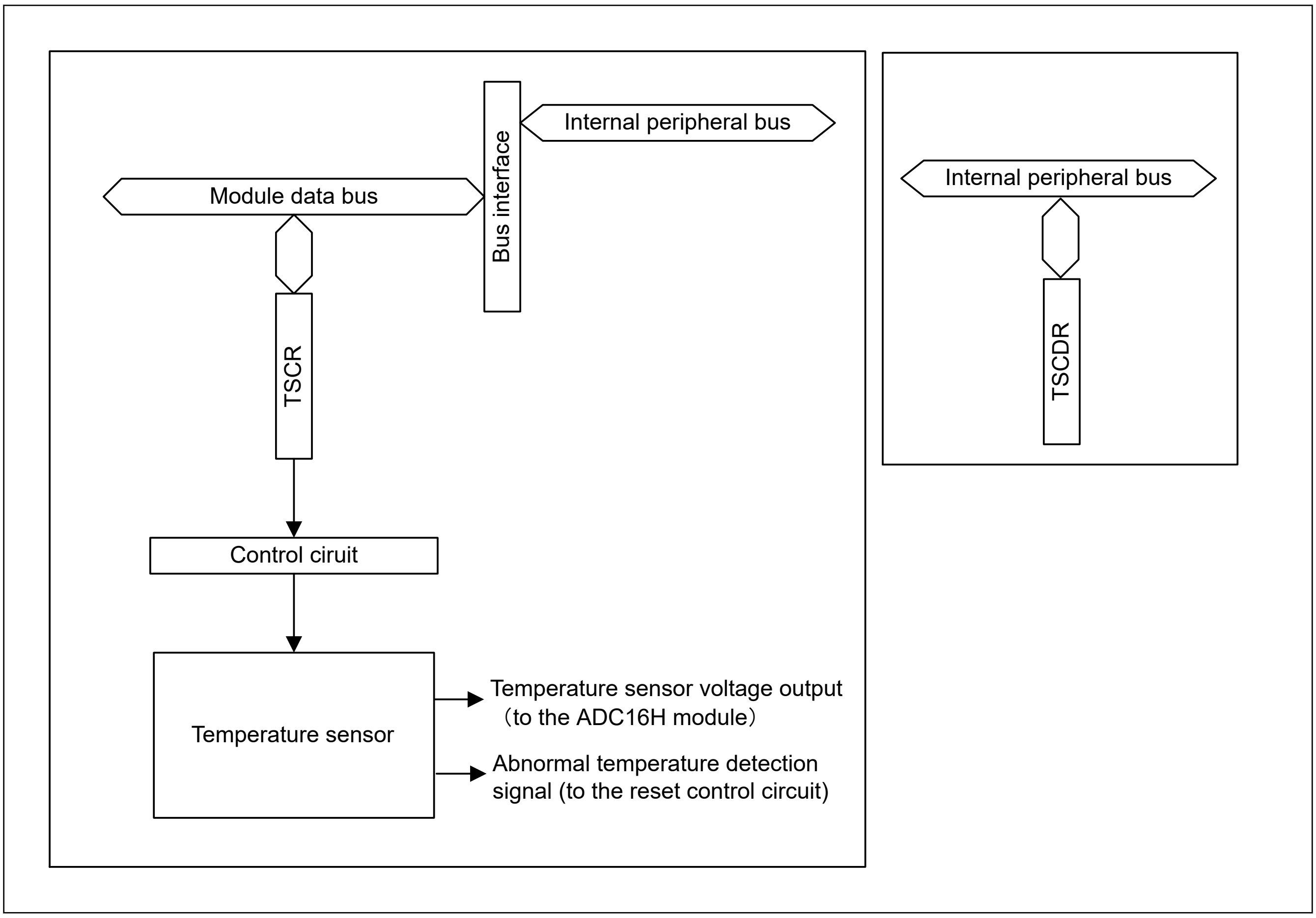


Figure 55.1 TSN block diagram

55.2 Register Descriptions

55.2.1 TSCR : Temperature Sensor Control Register

Base address: TSN = 0x4023_5000
TSN_NS = 0x5023_5000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TSEN	—	—	TSOE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	TSOE	Temperature Sensor Output Enable 0: Disable output from the temperature sensor to the ADC16H 1: Enable output from the temperature sensor to the ADC16H	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	TSEN	Temperature Sensor Enable 0: Stop the temperature sensor 1: Start the temperature sensor	R/W

Note: S-TYPE3, P-TYPE3

The TSCR is a register which controls the temperature sensor. The timing constraints shown in Figure 55.3 apply to the settings of the TSCR register.

TSOE bit (Temperature Sensor Output Enable)

The TSOE bit enables or disables the temperature sensor output to ADC16H.

TSEN bit (Temperature Sensor Enable)

The TSEN bit starts or stops the temperature sensor.

55.2.2 TSCDR : Temperature Sensor Calibration Data Register

Base address: TSD = 0x02C1_EDA0
TSD_NS = 0x12C1_EDA0

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TSCD[15:0]															
Value after reset:	Chip-specific value															

Bit	Symbol	Function	R/W
15:0	TSCD[15:0]	Temperature Sensor Calibration Data Chip-specific value	R
31:16	—	These bits are read as 0.	R

Note: S-TYPE5, P-TYPE5

The TSCDR register stores temperature sensor calibration data measured for each chip at factory shipment.

Temperature sensor calibration data is the output voltage of the temperature sensor under the conditions Tj = 125±2 °C or 105±2 °C and AVCC0 = VREFH0 = 3.3 V converted to a digital value by the A/D converter. For details on the measurement temperature condition, see section 55.3.1. Preparation for Using the Temperature Sensor.

The TSCDR register is a read-only 32-bit register. Read from this register in 32-bit units.

Temperature sensor calibration data is stored in the lower 12 bits of the TSCDR register.

55.2.3 TSCDR2 : Temperature Sensor Calibration Data Register 2

Base address: TSD = 0x02C1_EDA0
TSD_NS = 0x12C1_EDA0

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TSCD[15:0]															
Value after reset:	Chip-specific value															

Bit	Symbol	Function	R/W
15:0	TSCD[15:0]	Temperature Sensor Calibration Data Chip-specific value	R
31:16	—	These bits are read as 0.	R

Note: S-TYPE5, P-TYPE5

The TSCDR2 register stores temperature sensor calibration data measured for each chip at factory shipment.

Temperature sensor calibration data is the output voltage of the temperature sensor under the conditions Tj = -40±2 °C and AVCC0 = VREFH0 = 3.3 V converted to a digital value by the A/D converter. For details on the measurement temperature condition, see section 55.3.1. Preparation for Using the Temperature Sensor.

The TSCDR2 register is a read-only 32-bit register. Read from this register in 32-bit units.

Temperature sensor calibration data is stored in the lower 12 bits of the TSCDR2 register.

55.3 Using the Temperature Sensor

The temperature sensor outputs a voltage that varies with the temperature. This voltage is converted to a digital value by the A/D converter. To obtain the die temperature, convert this value into the temperature.

55.3.1 Preparation for Using the Temperature Sensor

The ambient temperature (T) is proportional to the temperature sensor voltage output (Vs), so ambient temperature is calculated with the following formula:

T = (Vs - V1) / slope + T1

- T: Ambient temperature of MCU as calculation result (°C)
- Vs: Voltage output by the temperature sensor on temperature measurement (V)
- T1: Temperature experimentally measured at one point (°C)
- V1: Voltage output by the temperature sensor on measurement of T1 (V)
- T2: Temperature experimentally measured at a second point (°C)
- V2: Voltage output by the temperature sensor on measurement of T2 (V)
- Slope: Temperature gradient of the temperature sensor (V / °C), slope = (V2 - V1) / (T2 -T1)

Characteristics vary between sensors, so Renesas recommends measuring two different sample temperatures as follows:

1. Use the A/D converter to measure the voltage V1 output by the temperature sensor at temperature T1.
2. Again use the A/D converter to measure the voltage V2 output by the temperature sensor at a different temperature T2.
3. Obtain the temperature gradient (slope = (V2 - V1) / (T2 - T1)) from these results.
4. Subsequently, obtain temperatures by substituting the slope into the formula for the temperature characteristic (T = (Vs -V1) / slope + T1).

In order to make the temperature Ta and Tj as close as possible, perform the measurement with the power consumption of the MCU as low as possible.

If you are using the temperature gradient given in section 62, Electrical Characteristics, use the A/D converter to measure the voltage V1 output by the temperature sensor at temperature T1, then calculate the temperature characteristic using the following formula:

T = (Vs - V1) / slope + T1

Note: This method produces less accurate temperatures than measurement at two points.

In this MCU, the TSCDR register and TSCDR2 register store the temperature values of the temperature sensor measured under the condition Ta = Tj = 125±2 °C or 105±2 °C, -40±2 °C and AVCC0 = VREFH0 = 3.3 V. If you use this value as the sample measurement result at the first and second points, you can omit the preparation before using the temperature sensor.

In case of Tj Max = 125 °C*1, the TSCDR register and TSCDR2 register store the temperature values (CAL125 and CAL-40) of the temperature sensor measured under the condition Ta = Tj = 125±2 °C, -40±2 °C and AVCC0 = VREFH0 = 3.3 V.

V1 is calculated from CAL125 or CAL-40:

V1 = 3.3 × CAL125 or CAL-40 / 4096 [V] (for 12 bits accuracy)

Using this value, the measured temperature can be calculated according to the following formula:

T = (Vs - V1) / slope + 125 or -40 [°C]

- T: Ambient temperature of MCU as calculation result (°C)
- Vs: Voltage output by the temperature sensor when the temperature is measured (V)
- V1: Voltage output by the temperature sensor when Ta = Tj = 125 or -40°C and AVCC0 = VREFH0 = 3.3 V (V)
- Slope: Temperature gradient of the temperature sensor*2 / 1000 (V/°C)
  Slope also can be obtained by using TSCDR register or TSCDR2 register value as the sample measurement result at the second point.

In case of Tj Max = 95 °C or 105 °C*1, the TSCDR register and TSCDR2 register store the temperature values (CAL105 and CAL-40) of the temperature sensor measured under the condition Ta = Tj = 105±2 °C, -40±2 °C and AVCC0 = VREFH0 = 3.3 V. The TSCDR2 register stores the temperature value of the temperature sensor measured under the condition Ta = Tj = -40±2 °C and AVCC0 = VREFH0 = 3.3 V.

V1 is calculated from CAL105 or CAL-40.

V1 = 3.3 × CAL105 or CAL-40 / 4096 [V] (for 12 bits accuracy)

Using this value, the measured temperature can be calculated according to the following formula:

T = (Vs - V1) / slope + 105 or -40 [°C]

- T: Ambient temperature of MCU as calculation result (°C)
- Vs: Voltage output by the temperature sensor when the temperature is measured (V)
- V1: Voltage output by the temperature sensor when Ta = Tj = 105 or -40 °C and AVCC0 = VREFH0 = 3.3 V (V)
- Slope: Temperature gradient of the temperature sensor*2 / 1000 (V/°C)
  Slope also can be obtained by using TSCDR register or TSCDR2 register value as the sample measurement result at the second point.

Note 1. The upper limit of operating junction temperature is 95 °C, 105 °C or 125 °C, depending on the product. For details, see section 1.3. Part Numbering.
Note 2. See section 62, Electrical Characteristics.

55.3.2 Procedures for Measuring Temperature Sensor Using the Temperature Sensor with ADC16H

Figure 55.2 shows the procedure for using the TSN.

For details on the ADC16H setting procedures, see section 53, 16-bit A/D Converter (ADC16H). For details on the TEMPRCR register, see section 6, Resets.

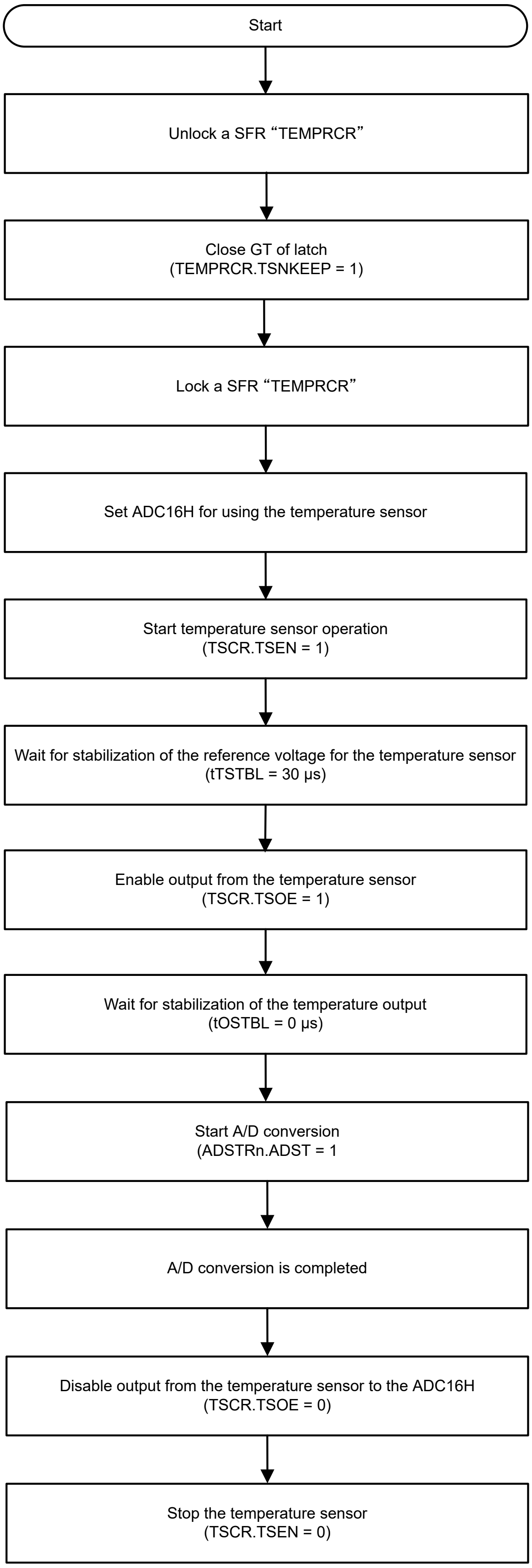


Figure 55.2 Procedure example for measuring temperature using TSN with ADC16H

Figure 55.3 shows the timing from the start of temperature sensor operation until the completion of A/D conversion when the ADC16H is in single scan mode (the conversion target is the temperature sensor output only). The times shown in the figure are described in Table 55.2.

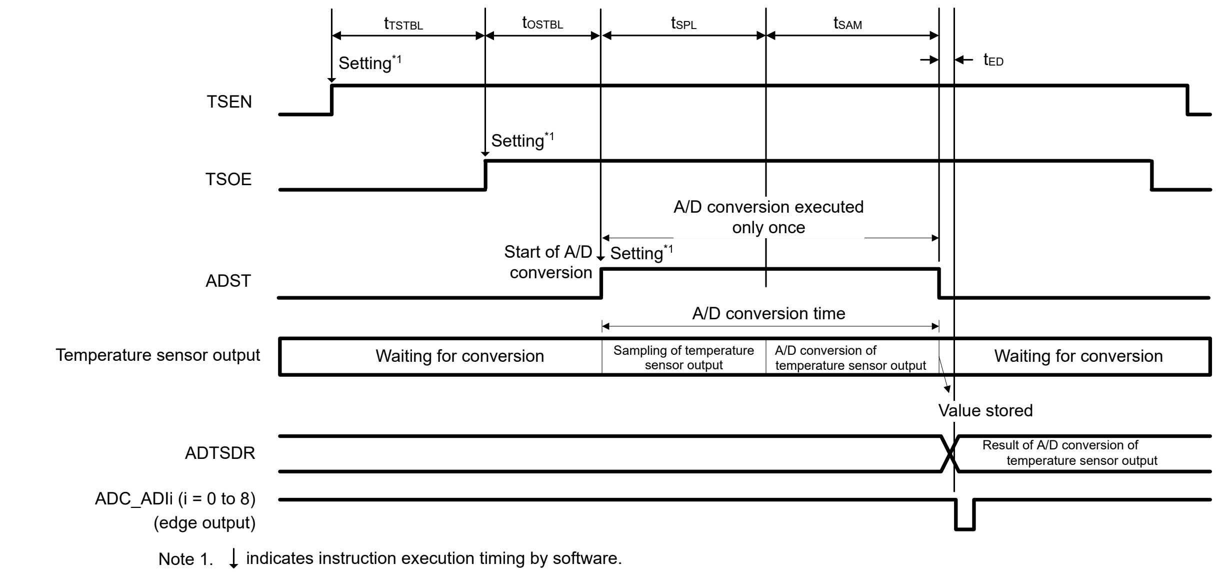


Figure 55.3 Timing from start of temperature sensor operation until completion of A/D conversion

Table 55.2 Time until completion of A/D conversion after start of temperature sensor operation

Parameter	Symbol	Time
Wait time for temperature sensor reference voltage stabilization	t_{TSTBL}	30 μs (min)
Wait time for temperature sensor output stabilization	t_{OSTBL}	0 μs (min)
Wait time for temperature sensor abnormal temperature detection signal stabilization	t_{RSTBL}	30 μs (min)
A/D converter input sampling time	t_{SPL}	ADSSTRn setting × ADCLK period
A/D conversion time	t_{SAM}	See Table XX.XX, Conversion times during scanning (in numbers of cycles of ADCLK and PCLKA) [DEADBEEF].
Scan conversion end delay	t_{ED}	

55.3.3 Procedures for Using the Temperature Monitor Reset

TSN can detect abnormal temperature and generate a signal to the reset circuit. If the temperature is outside the specified detection range*1, reset is issued. The function of temperature monitor reset is enabled using specific procedures that are described in Figure 55.4 and Figure 55.5.

For more information, see section 6.3.12. Temperature Monitor Reset.

Note 1. For details on the detection range of abnormal temperature, see section 62, Electrical Characteristics.

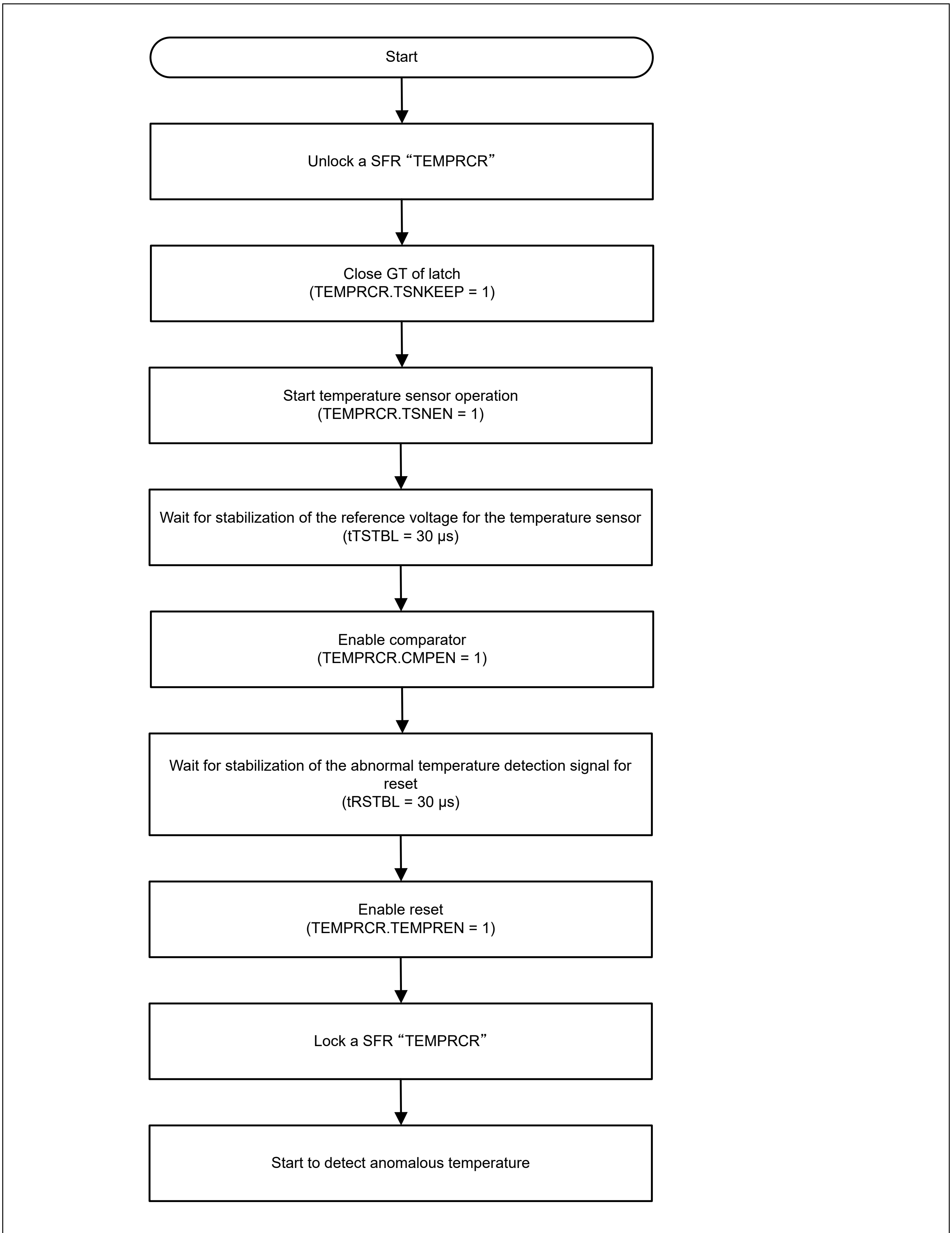


Figure 55.4 Procedure example for enabling temperature monitor reset but not measuring temperature with ADC16H

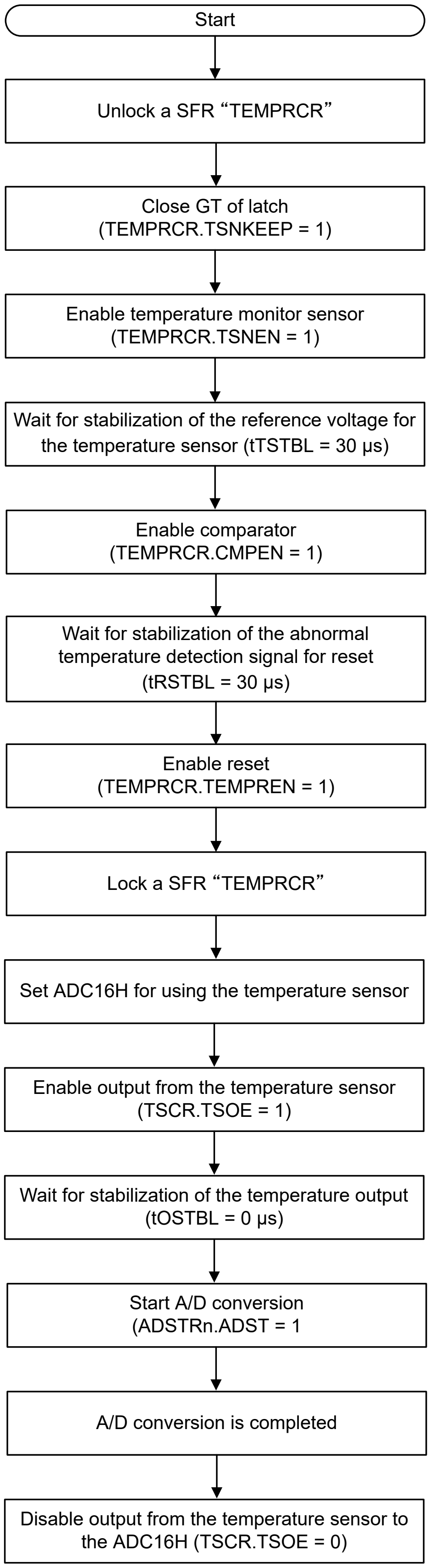


Figure 55.5 Procedure example for enabling temperature monitor reset and measuring temperature with ADC16H

55.4 Usage Notes

55.4.1 Settings for the Module-Stop Function

TSN operation can be disabled or enabled using the associated bit in Module Stop Control Register D (MSTPCRD). The TSN is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Mode.

55.4.2 Settings for the Software Standby Mode

Before entering Software Standby mode, set TSCR.TSOE bit to 0 after stopping A/D conversion, then set TSCR.TSEN bit to 0 to stop the TSN. Figure 55.6 shows the procedure for entering Software Standby mode and Figure 55.7 shows the procedure for returning from Software Standby mode.

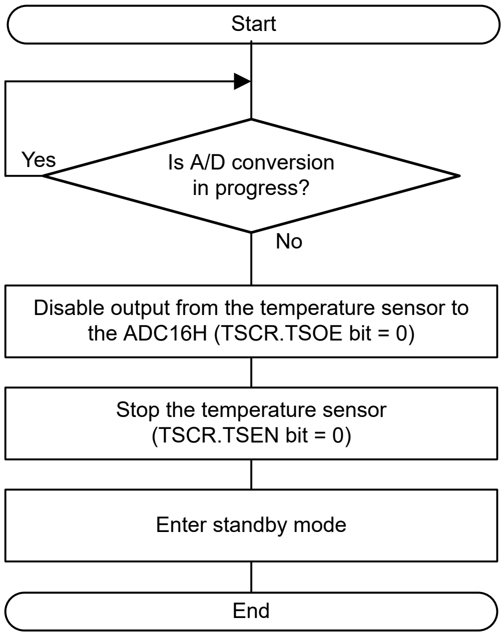


Figure 55.6 Example procedure for entering Software Standby mode

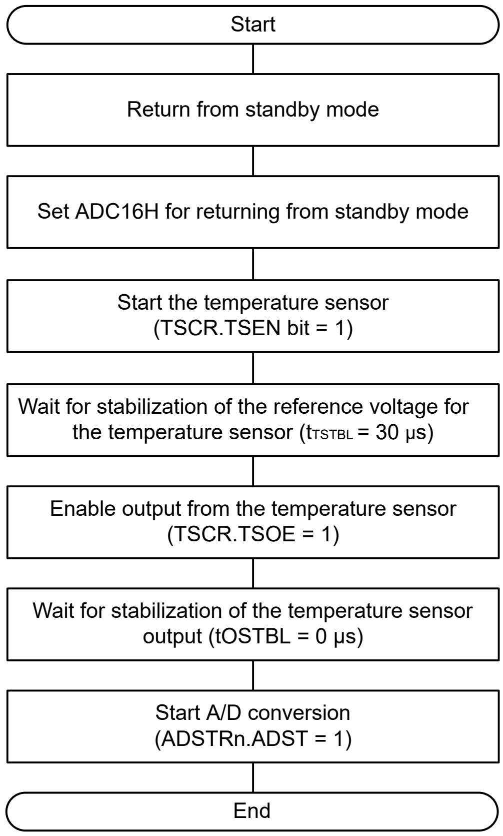


Figure 55.7 Example procedure for returning from Software Standby mode

56. High-Speed Analog Comparator (ACMPHS)

56.1 Overview

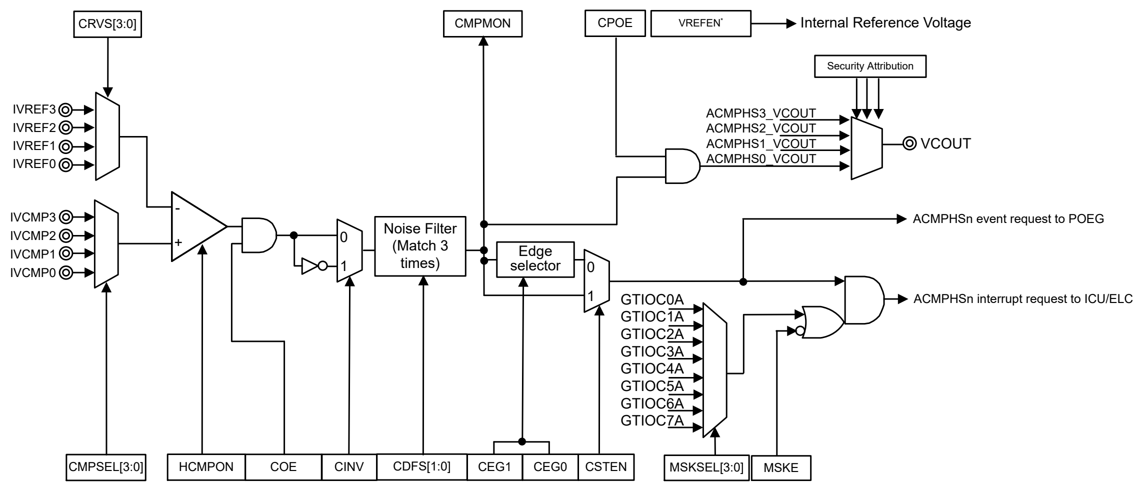
The High-Speed Analog Comparator (ACMPHS) can be used to compare an analog input voltage with a reference voltage and to provide a digital output based on the result of conversion. Both the analog input voltage and the reference voltage can be provided to the ACMPHS from internal sources (D/A converter output or internal reference voltage) and an external source. Such flexibility is useful in applications that require go/no-go comparisons to be performed between analog signals without necessarily requiring A/D conversion.

Table 56.1 lists the ACMPHS specifications, Figure 56.1 shows a block diagram, and Table 56.2 shows the input source configurations.

Table 56.1 ACMPHS specifications

Parameter	Specifications
Number of channels	4 channels: ACMPHSn (n = 0 to 3)
Analog input voltage	● Output from internal D/A converter ● Input from an external source (compatible with internal A/D converter input pin (one selectable))
Reference voltage	● Internal reference voltage (Vref) ● Output from internal D/A converter ● Input from an external source (compatible with internal A/D converter input pin (one selectable))
ACMPHS output	● Comparison result output to terminal PIN ● Generation of ELC event output ● Monitor output from register ● Generation of interrupt request output
Interrupt request signal	● Interrupt request generated on valid edge detection from comparison result (when CMPCTL.CSTEN = 0*1) ● Selectable to rising edge, falling edge, or both edges (when CMPCTL.CSTEN = 0*1) ● Rising edge only (when CMPCTL.CSTEN = 1*1)
Digital filter function	● Selectable to one of three sampling frequencies ● Not using the filter function is selectable
Module-stop function	Module-stop state can be set for each groups to reduce power consumption.
TrustZone Filter	Security and Privilege attribution can be set for each group

Note 1. Interrupt request signal selection restriction detail refers to section 56.5. ACMPHS Interrupts and section 56.6. ACMPHS Output to the Event Link Controller (ELC).



n = 0 to 3

CMPSEL[3:0] : bits in ACMPHSn.CMPSEL0 register
CRVS[3:0] : bits in ACMPHSn.CMPSEL1 register
CPOE, VREFEN : bits in ACMPHSn.CPIOC register
CMPMON : bits in ACMPHSn.CMPMON register
HCMPON, COE, CINV, CDFS[1:0], CEG1, CEG0, CSTEN : bits in ACMPHSn.CMPCTL register
MSKSEL[3:0] : bits in ACMPHSn.CMPSKCTL register
MSEK : bits in ACMPHSn.CMPINTCTL register

Note: VREFEN exists only in ACMPHS0.

Figure 56.1 ACMPHS block diagram

Table 56.2 Input source configuration of the ACMPHS

Comparator	Reference voltage input source				Analog voltage input source				Output pin
	IVREF3	IVREF2	IVREF1	IVREF0	IVCMP3	IVCMP2	IVCMP1	IVCMP0	
ACMPHS0	DA0	Vref	AN009	AN008	AN001	AN000	DA1	AN017	VCOUT[*1]
ACMPHS1	DA0	Vref	AN009	AN008	AN003	AN002	DA1	AN016	VCOUT[*1]
ACMPHS2	DA0	Vref	AN009	AN008	AN005	AN004	DA1	AN015	VCOUT[*1]
ACMPHS3	DA0	Vref	AN009	AN008	AN007	AN006	DA1	AN014	VCOUT[*1]

Note 1. ACMPHS0 to ACMPHS3 compare outputs are bundled with the VCOUT pin.

56.2 Register Descriptions

56.2.1 CMPCTL : Comparator Control Register

Base address: ACMPHSn = 0x4023_6000 + 0x0100 × n (n = 0 to 3)
ACMPHSn_NS = 0x5023_6000 + 0x0100 × n (n = 0 to 3)

Offset address: 0x000

Bit position:	7	6	5	4	3	2	1	0
Bit field:	HCMPON	CDFS[1:0]		CEG[1:0]		CSTEN	COE	CINV
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CINV	Comparator Output Polarity Selection[*1 *2] 0: Do not invert comparator output 1: Invert comparator output	R/W

Bit	Symbol	Function	R/W
1	COE	Comparator Output Enable 0: Disable comparator output (output signal is low level) 1: Enable comparator output	R/W
2	CSTEN	Interrupt Select*3 0: Output through the edge selector 1: Output directly	R/W
4:3	CEG[1:0]	Selection of Valid Edge (Edge Selector) 0 0: Do not detect edge 0 1: Detect rising edge 1 0: Detect falling edge 1 1: Detect both edges	R/W
6:5	CDFS[1:0]	Noise Filter Selection *1 *2 *3 *4 0 0: Do not use noise filter 0 1: Use noise filter sampling frequency of PCLKB/2^3 1 0: Use noise filter sampling frequency of PCLKB/2^4 1 1: Use noise filter sampling frequency of PCLKB/2^5	R/W
7	HCMPON	Comparator Operation Control*5 0: Stop operation (comparator outputs a low-level signal) 1: Enable operation (enables input to the comparator pins)	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Disable the ACMPHS output (COE = 0) before changing the CDFS[1:0] and CINV bits.

Note 2. If the CDFS[1:0] and CINV bits are changed, an ACMPHS interrupt request and an ELC event might be generated. Before changing these bits, set the ELSRn register to 0 (the ACMPHS output is not linked). After changing these bits, clear the IR flag in the IELSRn register to 0 to clear the interrupt status.

Note 3. Set the CSTEN bit to 1 and the CDFS[1:0] bits to 00b if the ACMPHS interrupt causes release of Software Standby mode. CSTEN is supported only by the ACMPHS0. ACMPHSn.CMPCTL.CTESN (n = 1 to 3) must be set to 0.

Note 4. If the CDFS[1:0] bits are changed from 00b (noise filter not used) to a value other than 00b (noise filter used), perform sampling four times and update the filter output, and then use the ACMPHS interrupt request or the ELC event.

Note 5. A stabilization wait time is required to permit ACMPHS operation after enabling it (HCMPON = 1). The operation stabilization wait time for ACMPHS modules 0 to 1 is 300 ns.

Note: Set this register before setting registers in the POEG when using comparator output as a POEG source.

The CMPCTL register controls the ACMPHS operation, enables or disables the ACMPHS output, selects the noise filter, selects the valid edge of the interrupt signal, and selects the interrupt.

56.2.2 CMPSEL0 : Comparator Input Select Register

Base address: ACMPHSn = 0x4023_6000 + 0x0100 × n (n = 0 to 3)
ACMPHSn_NS = 0x5023_6000 + 0x0100 × n (n = 0 to 3)

Offset address: 0x004

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	CMPSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CMPSEL[3:0]	Comparator Input Selection*1 *2 *3 0x00: Do not input 0x01: Select IVCMP0 0x02: Select IVCMP1 0x04: Select IVCMP2 0x08: Select IVCMP3 Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Use the following procedure to change the CMPSEL[3:0] bits. Writing a value other than 0x00 while the value of the CMPSEL0 register is not 0x00 is invalid. Writing 1 to two or more bits is also invalid. In both cases, the previous value is retained.

To change the CMPSEL[3:0] bits:
1. Set the CMPCTL.COE bit to 0.
2. Set the CMPSEL0 register to 0x00.
3. Set a new value in the CMPSEL[3:0] bits, with 1 set in only one of the bits.
4. Wait for the input switching stabilization wait time (200 ns).
5. Set the CMPCTL.COE bit to 1.
6. Clear the IR flag in the IELSRn register to clear the interrupt status.

Note 2. For details, see Table 56.2.
Note 3. If ACMPHSn level detection signal is used as a POEG source, write access to these bits after the setting of any register in the POEG may generate a POEG source.

56.2.3 CMPSEL1 : Comparator Reference Voltage Select Register

Base address: ACMPHSn = 0x4023_6000 + 0x0100 × n (n = 0 to 3)
ACMPHSn_NS = 0x5023_6000 + 0x0100 × n (n = 0 to 3)

Offset address: 0x008

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	CRVS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CRVS[3:0]	Reference Voltage Selection*1*2*3*4 0x00: Do not input 0x01: Select IVREF0 0x02: Select IVREF1 0x04: Select IVREF2 0x08: Select IVREF3 Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3
Note 1. Use the following procedure to change the CRVS[3:0] bits. Writing a value other than 0x00 while the value of the CMPSEL1 register is not 0x00 is invalid. Writing 1 to two or more bits is also invalid. In both cases, the previous value is retained.
To change the CRVS[3:0] bits:
1. Set the CMPCTL.COE bit to 0.
2. Set the CMPSEL1 register to 0x00.
3. Set a new value to the CRVS[3:0] bits, with 1 set in only one of the bits.
4. Wait for the input switching stabilization wait time (200 ns)
5. Set the CMPCTL.COE bit to 1.
6. Clear the IR flag in the IELSRn register to clear the interrupt status.

Note 2. For details, see Table 56.2.
Note 3. When the on-chip D/A converter output voltage is used, set the D/A converter to generate comparator reference voltage before enabling comparator operation (CMPCTL.HCMPON = 1). For details on setting the D/A converter, see section 54, 12-Bit D/A Converter (DAC12).
Note 4. If ACMPHSn level detection signal is used as a POEG source, write access to these bits after the setting of any register in the POEG may generate a POEG source.

56.2.4 CMPMON : Comparator Output Monitor Register

Base address: ACMPHSn = 0x4023_6000 + 0x0100 × n (n = 0 to 3)
ACMPHSn_NS = 0x5023_6000 + 0x0100 × n (n = 0 to 3)

Offset address: 0x00C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	COMPMON
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	COMPMON	Comparator Output Monitor*1 0: Comparator output is low 1: Comparator output is high	R
7:1	—	These bits are read as 0. The write value should be 0.	R

Note: S-TYPE-3, P-TYPE-3

Note 1. When ACMPHS operation is enabled (CMPCTL.HCMPON = 1 and CMPCTL.COE = 1) but the noise filter is not in use (CDFS[1:0] = 00b), design the software so that the COMPMON bit is read twice and the values are only used after the two consecutive values match.

56.2.5 CPIOC : Comparator Output Control Register

Base address: ACMPHSn = 0x4023_6000 + 0x0100 × n (n = 0 to 3)
ACMPHSn_NS = 0x5023_6000 + 0x0100 × n (n = 0 to 3)

Offset address: 0x010

Bit position:	7	6	5	4	3	2	1	0
Bit field:	VREFEN	—	—	—	—	—	—	CPOE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CPOE	External Pin Output Enable Comparison result by the comparator is output to an external pin. 0: Output to the comparator external pin is disabled (the output signal is fixed to low) 1: Output to the comparator external pin is enabled	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	VREFEN	Internal Vref Enable*1 0: Disable internal Vref 1: Enable internal Vref	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. For ACMPHS modules 0 to 3, VREFEN exists only in ACMPHS0.CPIOC. When using the internal Vref in ACMPHSn.CMPCTL.CTESN (n = 0 to 3), set the VREFEN bit in ACMPHS0.CPIOC to 1. Bit [7] in ACMPHSn.CPIOC (n = 1 to 3) should be 0 regardless of whether or not the internal Vref is used.

56.2.6 CPINTCTL : Comparator Interrupt Control Register

Base address: ACMPHSn = 0x4023_6000 + 0x0100 × n (n = 0 to 3)
ACMPHSn_NS = 0x5023_6000 + 0x0100 × n (n = 0 to 3)

Offset address: 0x040

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	MSKE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MSKE	Comparator Interrupt Periodic Mask Enable 0: Disable interrupt masking (Default) 1: Enable interrupt masking by GPT output signal selected by CPMSKCTL.MSKSEL [2:0]	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register controls the interrupt to ICU/ELC, while the event to POEG (Port Output Enable) is not controlled by this register.

56.2.7 CPMSKCTL : Comparator Interrupt Mask Control Register

Base address: ACMPHSn = 0x4023_6000 + 0x0100 × n (n = 0 to 3)
ACMPHSn_NS = 0x5023_6000 + 0x0100 × n (n = 0 to 3)

Offset address: 0x044

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	MSKSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	MSKSEL[2:0]	Comparator Interrupt Periodic Mask Selection 0 0 0: Enable interrupt masking by GTIOC0A output signal 0 0 1: Enable interrupt masking by GTIOC1A output signal 0 1 0: Enable interrupt masking by GTIOC2A output signal 0 1 1: Enable interrupt masking by GTIOC3A output signal 1 0 0: Enable interrupt masking by GTIOC4A output signal 1 0 1: Enable interrupt masking by GTIOC5A output signal 1 1 0: Enable interrupt masking by GTIOC6A output signal 1 1 1: Enable interrupt masking by GTIOC7A output signal	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register controls the interrupt to ICU/ELC, while the event to POEG(Port Output Enable) is not controlled by this register.

56.3 Operation

The ACMPHS compares a reference voltage to an analog input voltage. Operation is not guaranteed when the values of registers are changed during ACMPHS operation. Table 56.3 shows the procedures for setting the registers associated with ACMPHS.

Table 56.3 Procedure for setting registers associated with ACMPHSn (n = 0 to 3) (1 of 2)

Step	Register	Bit	Setting
1	Associated MSTPCRD register	MSTPD28 to MSTPD27	0: Input clock supply.
2	Associated pin function control register (PFS)	ASEL	1: Select the function of pins IVREF and IVCMP.
3	ACMPHSn.CPIOC	VREFEN	1: When using the internal Vref.
4	Associated D/A convertor		When using the D/A convertor, select in the register.
5	CMPSEL0, CMPSEL1	CMPSEL[3:0] CRVS[3:0]	Select the ACMPHSn input, with 1 set in only one of the bits.
6	CMPCTL	CDFS[1:0], CEG[1:0], and CINV	Set up ACMPHSn control.
		HCMPON	1: Enable ACMPHSn operation.
7	CMPSKCTL	MSKCTL[2:0]	Select the interrupt mask source signal (from GPT)
8	CPINT	MSKE	1: Enable the interrupt mask function if needed
9	Waiting for the ACMPHS stabilization time (minimum 300 ns).		
10	CMPCTL	COE	1: Enable ACMPHSn output.
11	CPIOC	CPOE	Set the VCOUT output
	Associated pin function control register (PFS)	PSEL, PMR	Select the VCOUT port function.
12	IELSRn	IR, IELS[8:0]	When using an interrupt, select the interrupt status flag and the ICU event link.*1
13	ELSRn	ELS[8:0]	When using an ELC, select the event link*2.

Table 56.3 Procedure for setting registers associated with ACMPHSn (n = 0 to 3) (2 of 2)

Step	Register	Bit	Setting
14	Operation started		
15	CMPCTL	COE	0: When changing IVREF or IVCMP, to disable ACMPHSn output.
16	CMPSEL1	CRVS[3:0]	Change the CMPSEL1 bits as follows: 1. Set bits CMPSEL1 to 0000 0000b. 2. Set a new value to the CMPSEL1 bits, with 1 set in only one of the bits.
	CMPSEL0	CMPSEL[3:0]	Change the CMPSEL0 bits as follows: 1. Set bits CMPSEL0 to 0000 0000b. 2. Set a new value to the CMPSEL0 bits, with 1 set in only one of the bits.
17	Waiting for the ACMPHS switching stabilization time (minimum 200 ns).		
18	CMPCTL	COE	1: Enable ACMPHSn output.
19	Operation restarted		

Note 1. After ACMPHSn is set, an unnecessary interrupt might occur until operation becomes stable, so initialize the interrupt flag.
Note 2. After ACMPHSn is set, an unnecessary interrupt might occur until operation becomes stable, so initialize the event link select.

Figure 56.2 and Figure 56.3 show an example of ACMPHS operation. The VCOUT output becomes 1 when the analog input voltage is higher than the ACMPHS reference input voltage, and the VCOUT output becomes 0 when the analog input voltage is lower than the reference voltage. When the ACMPHS output changes, an interrupt request and an ELC event are output.

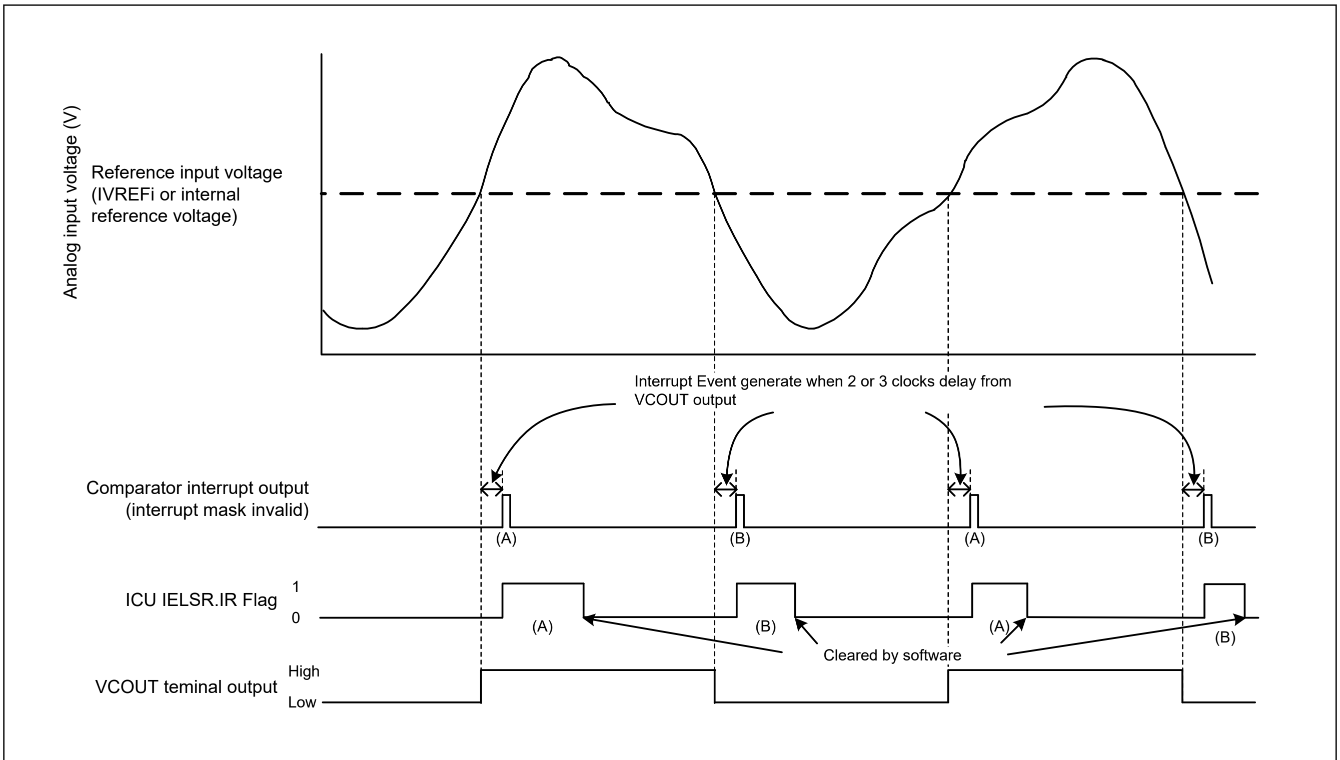


Figure 56.2 ACMPHS operation example (interrupt mask function invalid)

Figure 56.2 applies when CPOE = 1 (pin output enabled), CDFS[1:0] = 00b (filter not used), and CEG[1:0] = 11b (both-edge detection selected). When CINV = 0, CEG[1:0] = 01b (rising-edge detection selected for non-inversion output signal from the ACMPHS), the IELSR.IR flag changes as shown by (A) only. When CINV = 0, CEG[1:0] = 10b (falling-edge detection selected for non-inversion output signal from the ACMPHS), the IR flag changes as shown by (B) only. When CPOE = 1, VCOUT directly outputs.

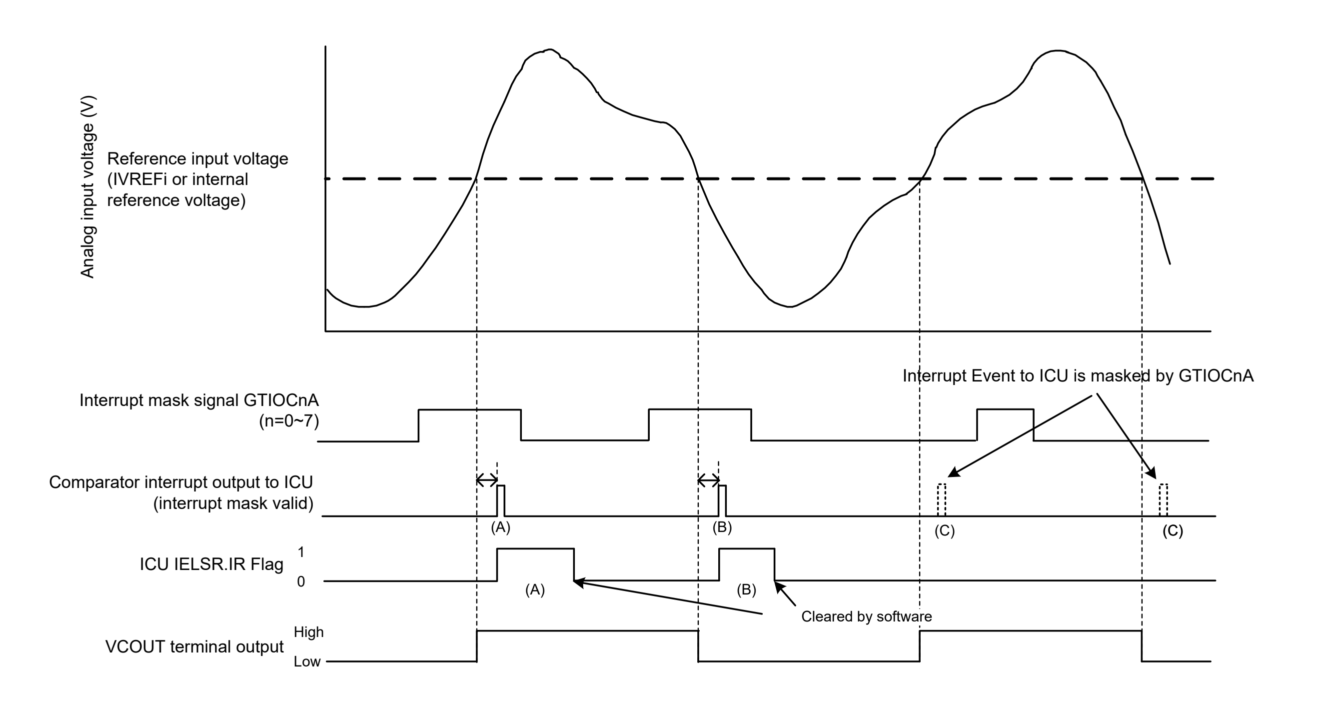


Figure 56.3 ACMPHS operation example (interrupt mask function valid)

Figure 56.3 applies when CPOE = 1 (pin output enabled), CDFS[1:0] = 00b (filter not used), CEG[1:0] = 11b (both-edge detection selected). When CINV = 0, CEG[1:0] = 01b (rising-edge detection selected for non-inversion output signal from the ACMPHS), the IELSR.IR flag changes as shown by (A) only. When CINV = 0, CEG[1:0] = 10b (falling-edge detection selected for non-inversion output signal from the ACMPHS), the IR flag changes as shown by (B) only.

When MSKE = 1 and MSKSEL[2:0] = 000b, the interrupt signal output to ICU is masked by GTIOC0A when GTIOC0A = low level as shown by (C), while it passes through when GTIOC0A = high level.

When CPOE = 1, VCOUT directly outputs, regardless of the MSKE setting.

56.4 Noise Filter

The ACMPHS contains a noise filter. The sampling clock can be selected in the CMPCTL.CDFS[1:0] bits. The comparator output signal is sampled every sampling clock, and if the same value is sampled three times, the noise filter output at the next sampling clock cycle is used as the ACMPHS output.

Figure 56.4 shows the configuration of the noise filter and edge detector, and Figure 56.5 shows an example of noise filter and interrupt operation.

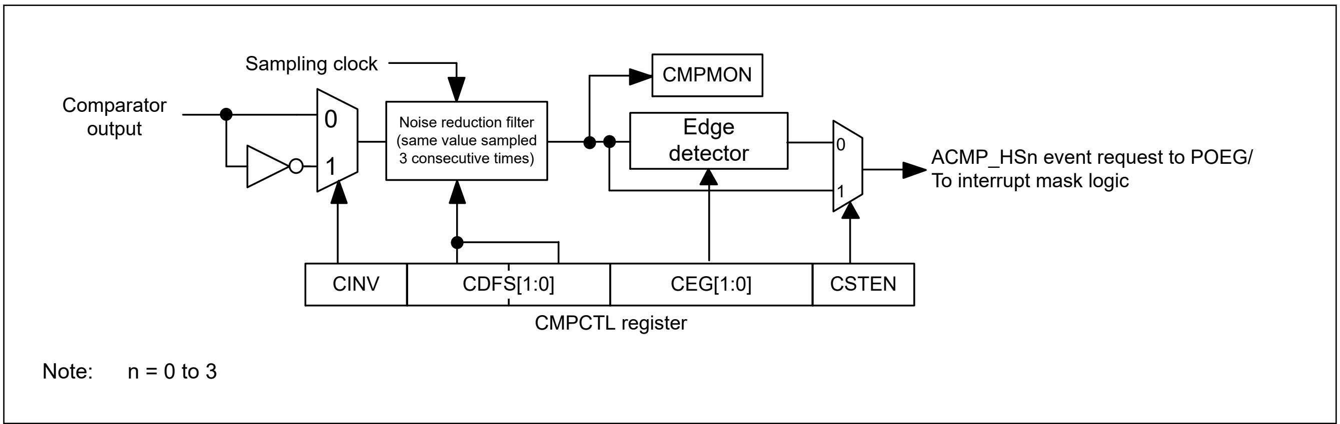


Figure 56.4 Noise filter and edge detection configuration

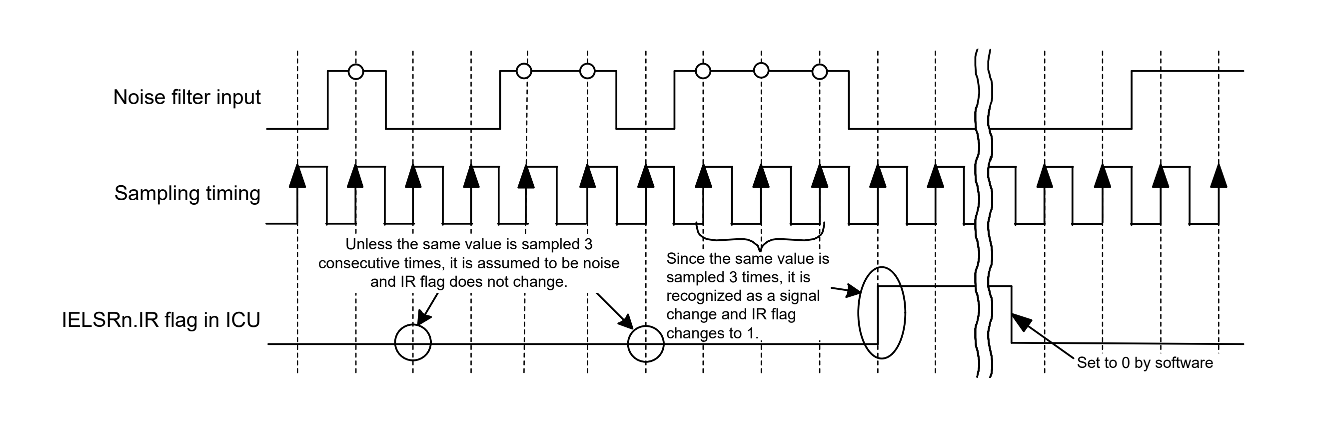


Figure 56.5 Noise filter and interrupt operation example

The operation example in Figure 56.5 applies when the CMPCTL.CDFS[1:0] bits are 01b, 10b, or 11b (noise filter used).

56.5 ACMPHS Interrupts

The ACMPHS generates four interrupt requests from sources ACMPHSn (n = 0 to 3). To use an ACMPHS interrupt, select it in the IELSR register in the Interrupt Controller Unit (ICU). Select the interrupt request in the CMPCTL.CSTEN bit either through the edge selector, or not.

Interrupt event to ICU/ELC can be masked by GPT output (GTIOCnA (n = 0 to 7)), it's controlled by CPINTCTL.MSKE bit and CPMSKCTL.MSKSEL[2:0] bits. Detail refer to Figure 56.3.

Event to POEG can't be masked by GPT output (GTIOCnA (n = 0 to 7)), it's not controlled by CPINTCTL.MSKE bit and CPMSKCTL.MSKSEL[2:0] bits.

When using the ACMPHS interrupt through the edge selector, set at least one of the CMPCTL.CEG[1:0] bits to 1 (to a value other than 00b for no edge selection). Set the CMPCTL.CSTEN bit to 0 (output through the edge selector) in Normal mode, CPU Sleep mode, and CPU Deep Sleep mode.

To use the ACMPHS interrupt in Software Standby mode, set the CMPCTL.CSTEN bit to 1 (direct output), set the CMPCTL.CDFS[1:0] bit to 00b (digital noise filter not used), and set CMPCTL.CINV as follows:

- When detecting compare result 0 to 1, set CMPCTL.CINV to 0 (comparator output not inverted)
- When detecting compare result 1 to 0, set CMPCTL.CINV to 1 (comparator output inverted).

An ACMPHS0 interrupt request can be used to release Software Standby mode. ACMPHS1, ACMPHS2, and ACMPHS3 cannot be used.

ACMPHSn (n = 0 to 3) cannot be used in Deep Software Standby mode.

For details on the register settings related to ACMPHS interrupt requests, see section 56.2.1. CMPCTL : Comparator Control Register, section 56.2.6. CPINTCTL : Comparator Interrupt Control Register and section 56.2.7. CPMSKCTL : Comparator Interrupt Mask Control Register.

56.6 ACMPHS Output to the Event Link Controller (ELC)

The ELC uses the ACMPHS interrupt request signal as an ELC event signal, enabling link operation for the preset module. To use the ACMPHS ELC event, select them in the ELSR register in the ELC. When using the ELC event request, set the CMPCTL.CSTEN bit to 0 (output through the edge selector). Also set at least one of the CMPCTL.CEG[1:0] bits to 1 (to a value other than 00b for no edge selection).

Interrupt event to ELC can be masked by GPT output (GTIOCnA (n = 0 to 7)), it's controlled by CPINTCTL.MSKE bit and CPMSKCTL.MSKSEL[2:0] bits.

56.7 ACMPHS Pin Output

The comparison result from the ACMPHS can be output to external pins. Use the CMPCTL.CINV and CPIOC.CPOE bits to set the output polarity (non-inverted or inverted output) and enable or disable output. To output the ACMPHS comparison result to the VCOUT output pin, set the associated port mn pin function control register (PmnPFS) in the I/O register.

Only when the security attribution of Port and ACMPHS match, the ACMPHS comparison result is output to the VCOUT pin as shown in the Table 56.4.

Table 56.4 The output condition of VCOUT pin by the security setting

PORT Security Attribution (PmSAR (m = 0 to 9, A, B))	ACMPHSn Security Attribution (PSARm (m = B to E))	VCOUT pin output source
0 (secure)	0 (secure)	ACMPHSn output enable
0 (secure)	1 (non-secure)	ACMPHSn output disable
1 (non-secure)	0 (secure)	ACMPHSn output disable
1 (non-secure)	1 (non-secure)	ACMPHSn output enable

Note: n = 0 to 3

56.8 Usage Notes

56.8.1 Settings for the Module-Stop Function

ACMPHS operation can be disabled or enabled using the Module Stop Control Register. The ACMPHS is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Mode.

56.8.2 Settings for the DAC12

ACMPHS is connected to the internal module output of the D/A converter. For details, see section 54, 12-Bit D/A Converter (DAC12).

56.8.3 Relationship with the ADC16H

Constraints apply on the simultaneous use of ACMPHS analog input and ADC16H analog input. For details, see section section 53.11.10. Restrictions on Analog Channel Shared Among Multiple A/D Converters (ADC16H) and High-Speed Analog Comparator (ACMPHS).

56.8.4 ACMPHS Operation in Module-Stop State

When the module-stop state is entered while ACMPHS is operating, analog circuits in the ACMPHS are not stopped and the analog power supply current is the same as that when ACMPHS is being used. If the analog power supply current needs to be reduced in the module-stop state, set the CMPCTL.HCMPON bit to 0 to stop the ACMPHS.

56.8.5 ACMPHS Operation in Software Standby Mode

When Software Standby mode is entered while ACMPHS is operating, analog circuits in the ACMPHS are not stopped and the analog power supply current is the same as that when ACMPHS is being used. If the analog power supply current needs to be reduced in Software Standby mode, set the CMPCTL.HCMPON bit to 0 to stop the ACMPHS.

56.8.6 Setting the D/A Converter for Generating Reference Voltage

Set the D/A converter to generate reference voltage and wait for the D/A converter output settling time before enabling the comparator. Similarly, before making any changes to the settings of the D/A converter, stop the comparator temporarily. After the changes are made, wait for the D/A converter output settling time before enabling the comparator.

57. Data Operation Circuit (DOC)

This is the DOC_B version of the DOC peripheral module.

DOC_B is referred to as DOC in this chapter.

57.1 Overview

The data operation circuit (DOC) is used to compare, add, and subtract 16 or 32-bit data. An interrupt can be generated when the following conditions apply.

- When the 16 or 32-bit compared values match the detection condition
- When the result of 16 or 32-bit data addition overflows
- When the result of 16 or 32-bit data subtraction underflows

Table 57.1 lists the data operation circuit specifications and Figure 57.1 shows a block diagram of the data operation circuit.

Table 57.1 DOC specifications

Item	Description
Data operation function	● 16 or 32-bit data comparison, comparison to detect data above or below thresholds, and window comparison ● 16 or 32-bit data addition, and subtraction
Module-stop function	The module-stop state can be set to reduce power consumption.
Interrupts	● The compared values match the detection condition ● The result of data addition is greater than 0xFFFF (DOCR.DOBW = 0) or 0xFFFF_FFFF (DOCR.DOBW = 1) ● The result of data subtraction is less than 0x0000 (DOCR.DOBW = 0) or 0x0000_0000 (DOCR.DOBW = 1)
Event link function (output)	● The result of data comparison is consistent with detection condition ● The result of data addition is greater than 0xFFFF (DOCR.DOBW = 0) or 0xFFFF_FFFF (DOCR.DOBW = 1) ● The result of data subtraction is less than 0x0000 (DOCR.DOBW = 0) or 0x0000_0000 (DOCR.DOBW = 1)
TrustZone Filter	Security and Privilege attribution can be set

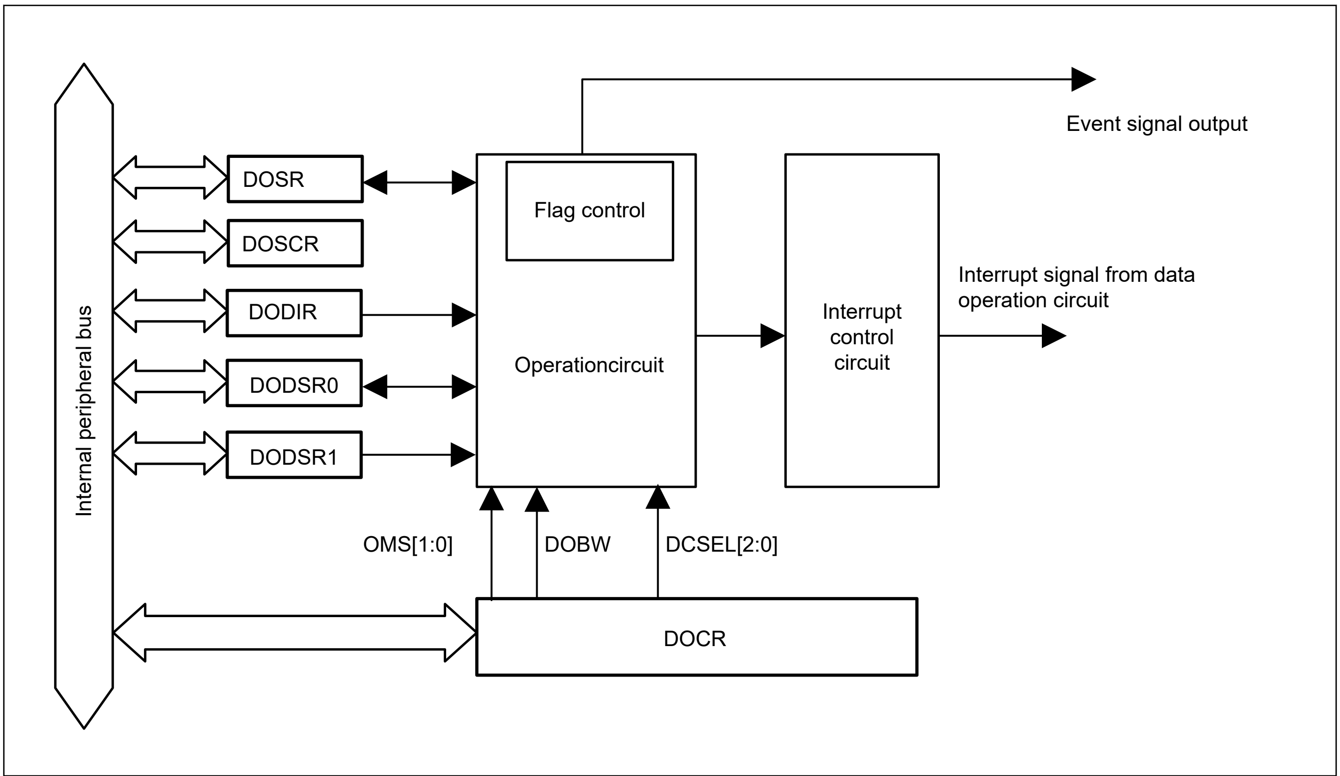


Figure 57.1 DOC block diagram

57.2 DOC Register Descriptions

57.2.1 DOCR : DOC Control Register

Base address: DOC_B = 0x4031_1000
DOC_B_NS = 0x5031_1000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	DCSEL[2:0]			DOBW	—	OMS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	OMS[1:0]	Operating Mode Select 0 0: Data comparison mode 0 1: Data addition mode 1 0: Data subtraction mode 1 1: Setting prohibited	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	DOBW	Data Operation Bit Width Select 0: 16-bit 1: 32-bit	R/W
6:4	DCSEL[2:0]*1	Detection Condition Select 0 0 0: Mismatch (DODSR0 ≠ DODIR) 0 0 1: Match (DODSR0 = DODIR) 0 1 0: Lower (DODSR0 > DODIR) 0 1 1: Upper (DODSR0 < DODIR) 1 0 0: Inside window (DODSR0 < DODIR < DODSR1) 1 0 1: Outside window (DODIR < DODSR0, DODSR1 < DODIR) Others: Setting prohibited	R/W

Bit	Symbol	Function	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3
Note 1. Valid only when data comparison mode is selected.

The DOCR is a register which can set the operation mode of data operation circuit and interrupt enable/disable.

OMS[1:0] bits (Operating Mode Select)

These bits select the operating mode of the data operation circuit.

DOBW bit (Data Operation Bit Width Select)

This bit selects the bit width of data operation.

DCSEL[2:0] bits (Detection Condition Select)

These bits are valid only when data comparison mode is selected.

These bits select the condition for detection in data comparison mode.

57.2.2 DOSR : DOC Flag Status Register

Base address: DOC_B = 0x4031_1000
DOC_B_NS = 0x5031_1000

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	DOPCF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DOPCF	Data Operation Circuit Flag Indicates the result of an operation.	R
7:1	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3

The DOSR register indicates the status of the data operation.

DOPCF flag (Data Operation Circuit Flag)

[Setting conditions]

- DOCR.OMS[1:0] bits = 00b (Data comparison mode): The compared value matches the detection condition selected by DOCR.DCSEL[2:0] bits
- DOCR.OMS[1:0] bits = 01b (Data addition mode): The result of data addition is greater than 0xFFFF (DOCR.DOBW = 0) or 0xFFFF_FFFF (DOCR.DOBW = 1)
- DOCR.OMS[1:0] bits = 10b (Data subtraction mode): The result of data subtraction is less than 0x0000 (DOCR.DOBW = 0) or 0x0000_0000 (DOCR.DOBW = 1)

[Clearing condition]

- Writing 1 to the DOSCR.DOPCFCL bit

57.2.3 DOSCR : DOC Flag Status Clear Register

Base address: DOC_B = 0x4031_1000
DOC_B_NS = 0x5031_1000

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	DOPCFCL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DOPCFCL	DOPCF Clear 0: Maintains the DOPCF flag state. 1: Clears the DOPCF flag.	W
7:1	—	The write value should be 0.	W

Note: S-TYPE3, P-TYPE3

The DOSCR is a register which can clear the status of data operation. This register is read as 0x00.

DOPCFCL bit (DOPCF Clear)

Setting this bit to 1 clears the DOPCF flag.

57.2.4 DODIR : DOC Data Input Register

Base address: DOC_B = 0x4031_1000
DOC_B_NS = 0x5031_1000

Offset address: 0x0C

Bit position: 31 ... 0

Bit field: (31:0)

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	It stores data used in the operations. Access the DODIR with the bit width of data operation selected by the DOCR.DOBW bit.	R/W

Note: S-TYPE3, P-TYPE3

57.2.5 DODSR0 : DOC Data Setting Register 0

Base address: DOC_B = 0x4031_1000
DOC_B_NS = 0x5031_1000

Offset address: 0x10

Bit position: 31 ... 0

Bit field: (31:0)

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	Access the DODSR0 with the bit width of data operation selected by the DOCR.DOBW bit. This register stores data for use as a reference in data comparison mode. When selecting window comparison (DOCR.DCSEL[2:0] = 100b, 101b), set a value less than DODSR1 (DODSR1 > DODSR0). This register also stores the results of operations in data addition and data subtraction modes.	R/W

Note: S-TYPE3, P-TYPE3

57.2.6 DODSR1 : DOC Data Setting Register 1

Base address: DOC_B = 0x4031_1000
DOC_B_NS = 0x5031_1000

Offset address: 0x14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Access the DODSR1 with the bit width of data operation selected by the DOCR.DOBW bit. This register stores data for use as a reference in data comparison mode. When selecting window comparison (DOCR.DCSEL[2:0] = 100b, 101b), set a value greater than DODSR0 (DODSR1 > DODSR0). This register is only used for window comparisons.	R/W

Note: S-TYPE3, P-TYPE3

57.3 Operation

57.3.1 Data Comparison Mode

Figure 57.2 to Figure 57.7 shows an example of the steps involved in data comparison mode operation by the data operation circuit.

The following is an example of operation when the bit width of data operation is 32-bit.

1. Writing 00b to the DOCR.OMS[1:0] bits selects data comparison mode, and setting the DOCR.DCSEL[2:0] to selects detection condition.
2. The 32-bit reference data is set in DODSR0 and DODSR1.*1
3. 32-bit data for comparison is written to DODIR.
4. If a value written to DODIR match the detection condition set by DOCR.DCSEL[2:0], the DOCR.DOPCF flag is set to 1 and an ELC event and a data operation circuit interrupt are generated.

Note: The comparison operation is executed only by writing to the DODIR

Note 1. The DODSR1 register setting is required only when window comparison is selected. Set a value greater than DODSR0 (DODSR1 > DODSR0).

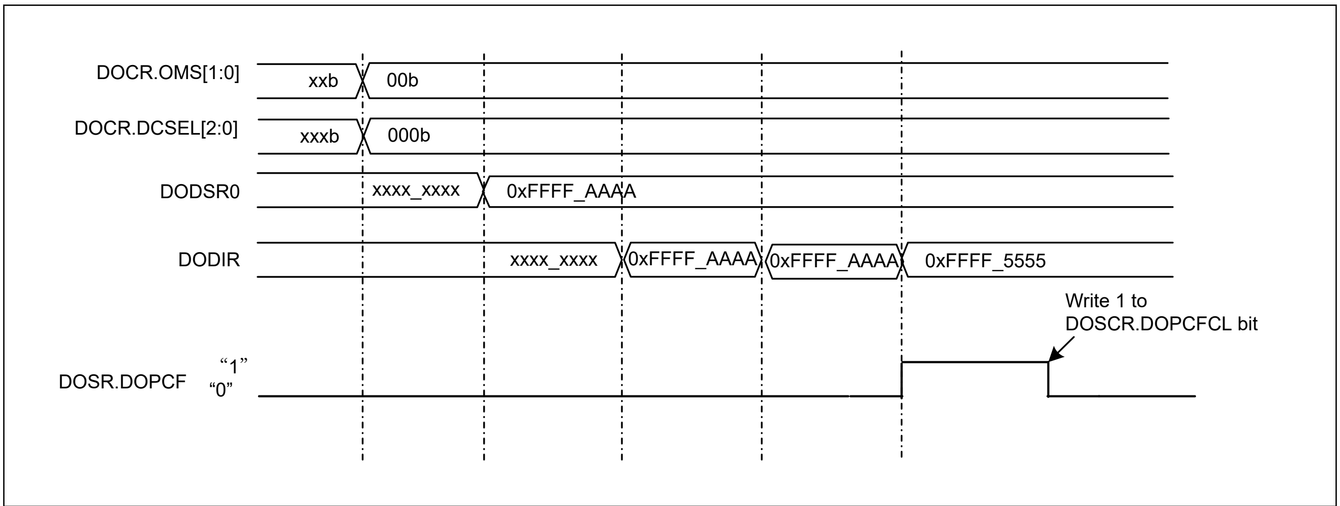


Figure 57.2 Example of Operation in Data Comparison Mode (Detection condition: Mismatch)

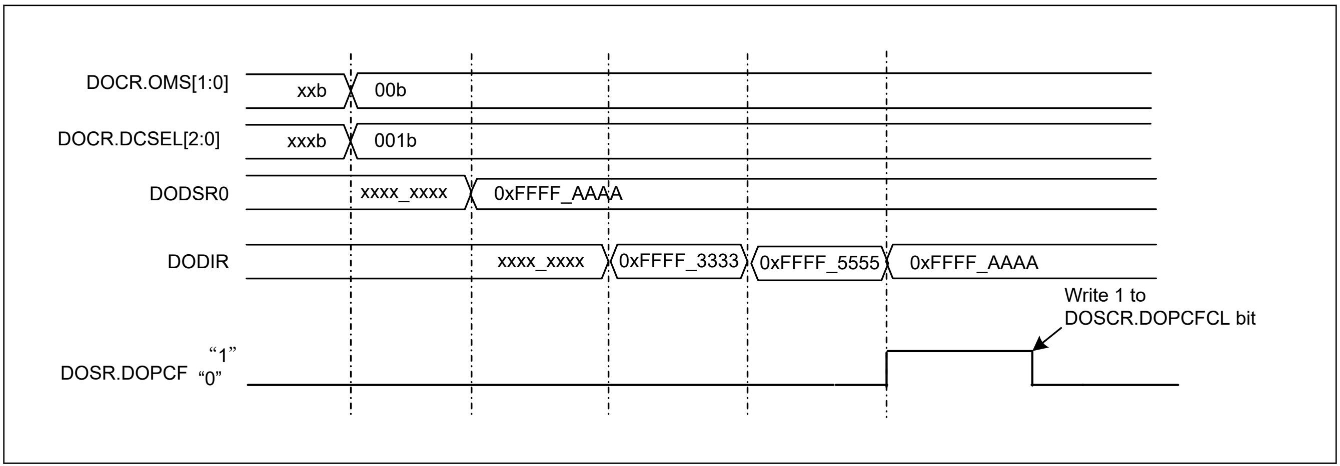


Figure 57.3 Example of Operation in Data Comparison Mode (Detection condition: Match)

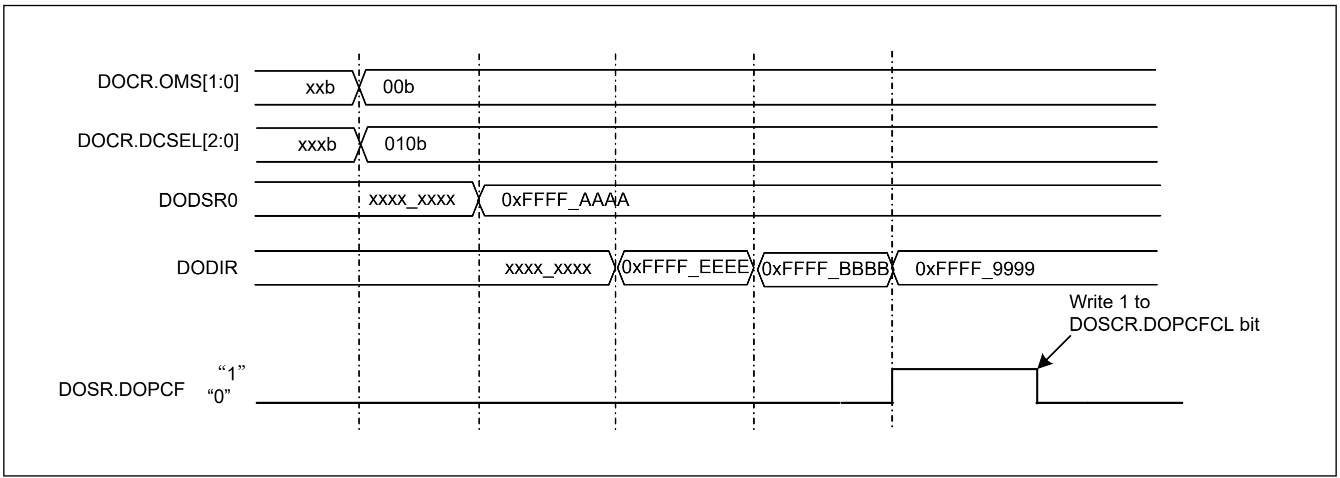


Figure 57.4 Example of Operation in Data Comparison Mode (Detection condition: Lower)

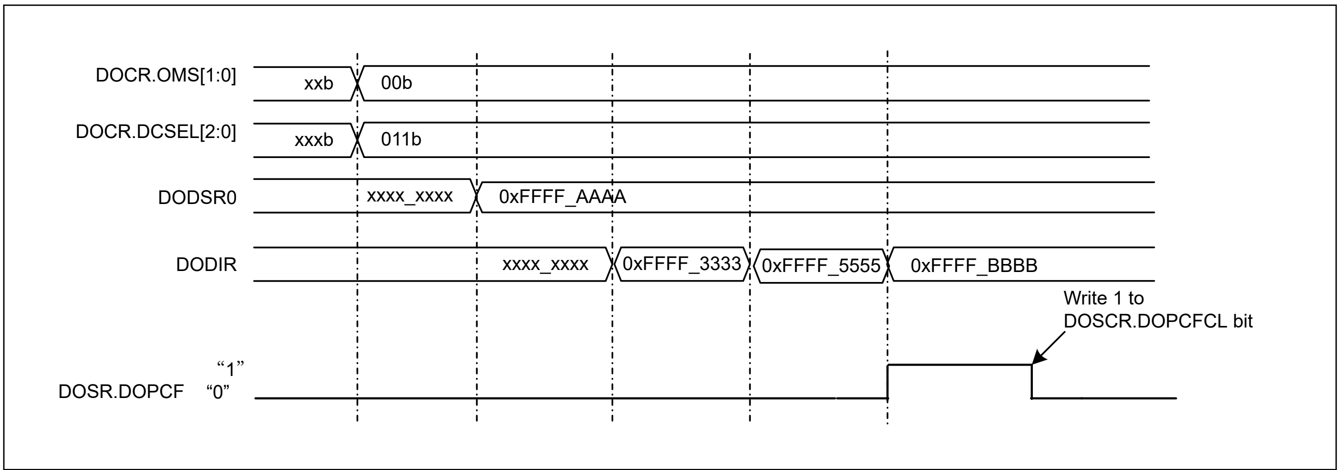


Figure 57.5 Example of Operation in Data Comparison Mode (Detection condition: Upper)

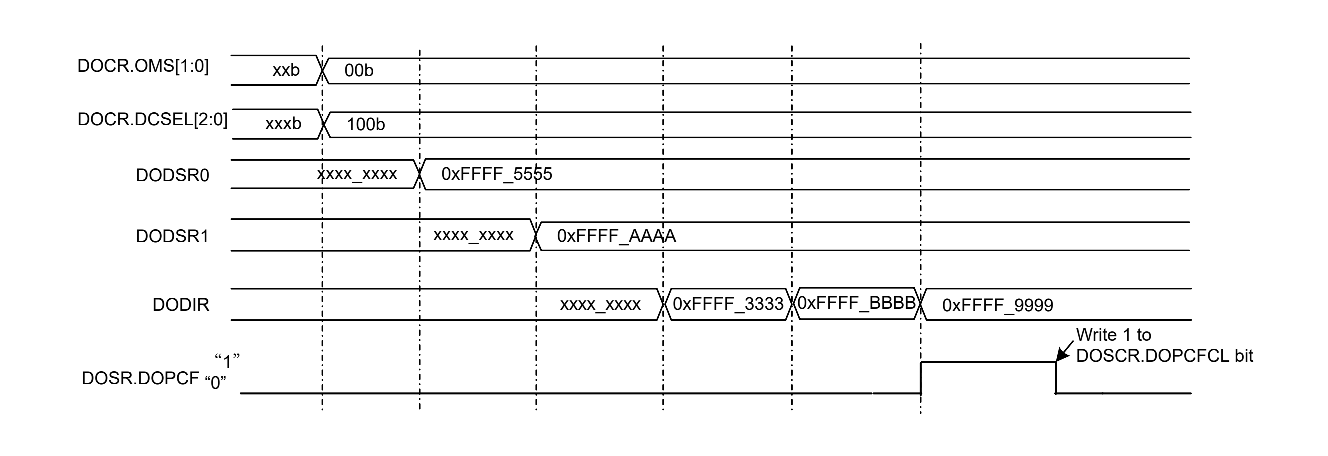


Figure 57.6 Example of Operation in Data Comparison Mode (Detection condition: Inside window)

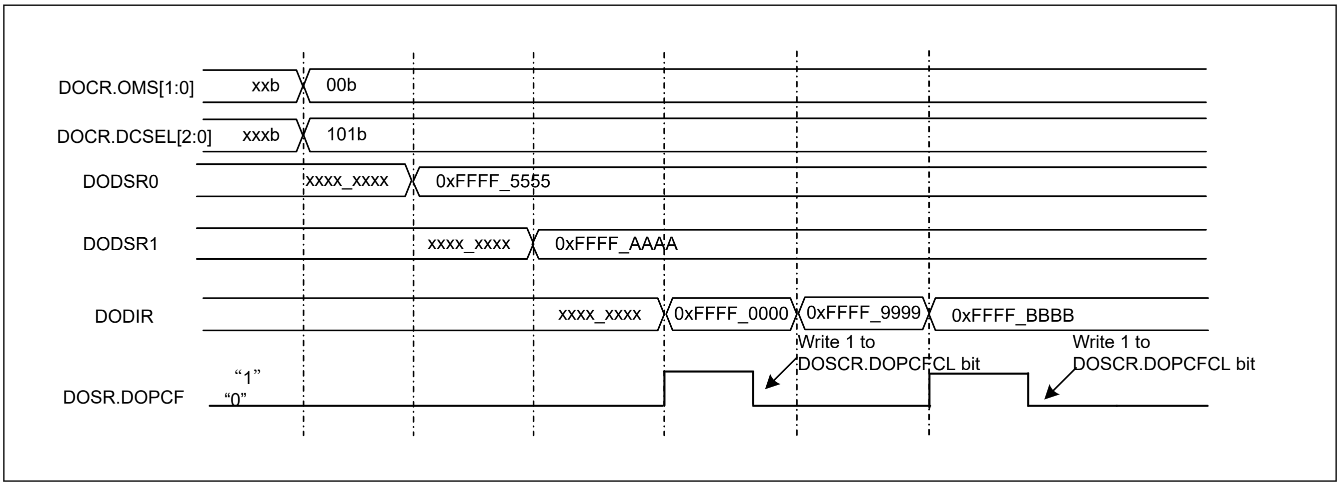


Figure 57.7 Example of Operation in Data Comparison Mode (Detection condition: Outside window)

57.3.2 Data Addition Mode

Figure 57.8 shows an example of the steps involved in data addition mode *1 operation by the data operation circuit.

The following is an example of operation when the bit width of data operation is 32-bit.

1. Writing 01b to the DOCR.OMS[1:0] bits selects data addition mode.
2. 32-bit data is set in the DODSR0 register as the initial value.
3. 32-bit data to be added is written to DODIR. The result of the operation is stored in DODSR0.
4. Writing of 32-bit data continues until all data for addition have been written to DODIR.
5. If the result of an operation is greater than 0xFFFF_FFFF, the DOSR.DOPCF flag is set to 1 and an ELC event and a data operation circuit interrupt are generated.

Note 1. Addition is executed only by writing to the DODIR.

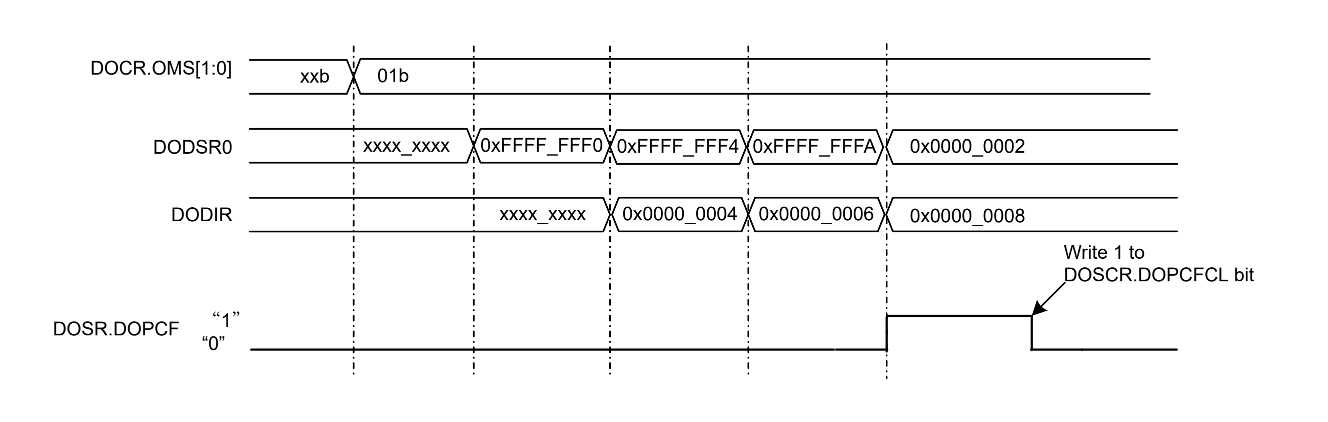


Figure 57.8 Example of Operation in Data Addition Mode

57.3.3 Data Subtraction Mode

Figure 57.9 shows an example of the steps involved in data subtraction mode [*1] operation by the data operation circuit.

The following is an example of operation when the bit width of data operation is 32-bit.

1. Writing 10b to the DOCR.OMS[1:0] bits selects data subtraction mode.
2. 32-bit data is set in the DODSR0 register as the initial value.
3. 32-bit data to be subtracted is written to DODIR. The result of the operation is stored in DODSR0.
4. Writing of 32-bit data continues until all data for subtraction have been written to DODIR.
5. If the result of an operation is less than 0x0000_0000, the DOSR.DOPCF flag is set to 1 and an ELC event and a data operation circuit interrupt are generated.

Note 1. Subtraction is executed only by writing to the DODIR.

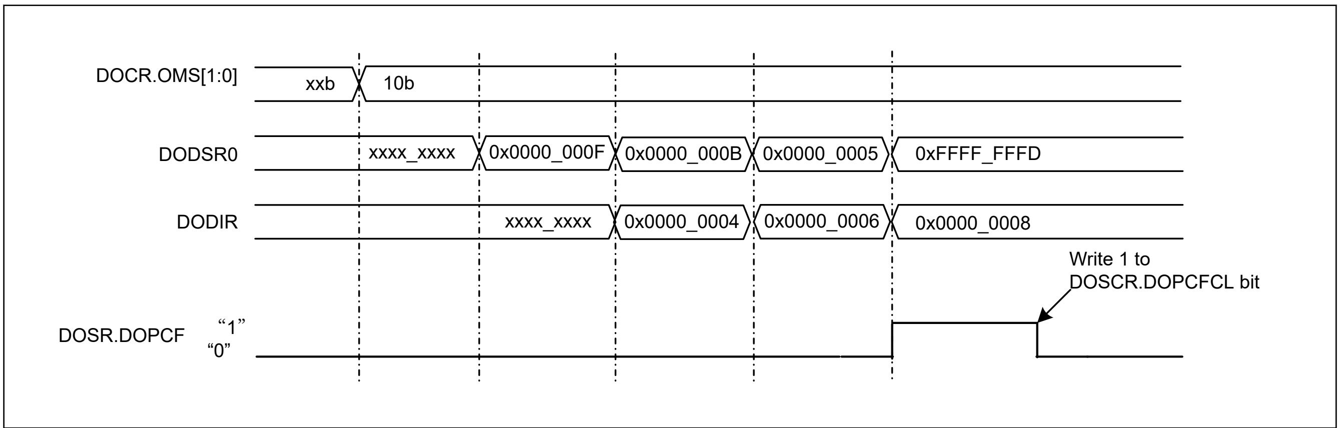


Figure 57.9 Example of Operation in Data Subtraction Mode

57.4 Interrupt Source

The data operation circuit generates the data operation circuit interrupt (DOC_DOPCI) as an interrupt request. When an interrupt source is generated, the data operation circuit flag corresponding to the interrupt is set to 1, and then interrupt request signal is generated. Table 57.2 describes the interrupt request.

Table 57.2 Interrupt request from DOC

Interrupt request	Status flag	Interrupt source
DOC interrupt	DOPCF	● The compared values match the detection condition. ● The result of data addition is greater than 0xFFFF (DOCR.DOBW = 0) or 0xFFFF_FFFF (DOCR.DOBW = 1). ● The result of data subtraction is less than 0x0000 (DOCR.DOBW = 0) or 0x0000_0000 (DOCR.DOBW = 1).

57.5 Event Link Output

The DOC outputs event signals for the event link controller (ELC) under the following conditions, and these can be used to initiate operations by other modules selected in advance.

- The compared values match the detection condition
- The data addition result is greater than 0xFFFF (DOCR.DOBW = 0) or 0xFFFF_FFFF (DOCR.DOBW = 1)
- The data subtraction result is less than 0x0000 (DOCR.DOBW = 0) or 0x0000_0000 (DOCR.DOBW = 1)

57.6 Interrupt Handling and Event Linking

The DOC has a bit to enable or disable interrupts. An interrupt request signal is output for the CPU when an interrupt source is generated while the corresponding enable bit is enabled.

In contrast, an event link output signal is sent to other modules as an event signal via the ELC when an interrupt source is generated, regardless of the setting of the corresponding interrupt enable bit.

57.7 Usage Notes

57.7.1 Settings for the Module-Stop State

The module Stop Control Register C (MSTPCRC) can enable or disable DOC operation. The DOC is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Mode.

58. SRAM

58.1 Overview

The MCU provides an on-chip, high-density SRAM module with Error Correction Code (ECC).

In single-core, C-TCM and S-TCM at addresses 0x2A00_0000 to 0x2A01_FFFF can be used as SRAM.

See the section 2, CPU for details on TCM.

Table 58.1 SRAM specifications

Parameter		SRAM0	SRAM1	SRAM2	SRAM3
SRAM capacity		512 KB	512 KB	512 KB	128 KB
SRAM address	Secure alias	0x2200_0000 to 0x2207_FFFF	0x2208_0000 to 0x220F_FFFF	0x2210_0000 to 0x2217_FFFF	0x2218_0000 to 0x2219_FFFF
	Non-secure alias	0x3200_0000 to 0x3207_FFFF	0x3208_0000 to 0x320F_FFFF	0x3210_0000 to 0x3217_FFFF	0x3218_0000 to 0x3219_FFFF
ECC region*1	Secure alias	0x221A_0000 to 0x221A_FFFF	0x221B_0000 to 0x221B_FFFF	0x221C_0000 to 0x221C_FFFF	0x221D_0000 to 0x221D_3FFF
	Non-secure alias	0x321A_0000 to 0x321A_FFFF	0x321B_0000 to 0x321B_FFFF	0x321C_0000 to 0x321C_FFFF	0x321D_0000 to 0x321D_3FFF
Access		Wait states are not inserted into the access cycle by default. If the ICLK frequency is greater than half the maximum frequency in the electrical characteristics, a wait state is required. If the ICLK frequency is less than or equal half the maximum frequency in the electrical characteristics, a wait state is not required.			
Data retention function		Not available in Deep Software Standby mode			
Module-stop function		Module-stop state can be set to reduce power consumption			
Error checking		SEC-DED (Single-Error Correction and Double-Error Detection Code: 64-bit data with 8-bit ECC code)			
Security		TrustZone Filter is integrated for memory access and SFR access. Access to the memory space is controlled by setting the memory Security Attribution (SA). And access to I/O space (SFR) space is controlled by setting the register Security Attribution (SA). See section 58.3.5. TrustZone Filter Function.			

Note 1. When ECC function is disabled, it is used as a data region and can be accessed directly. When ECC function is enabled, direct access is disabled. When bypass is enabled, direct access to ECC bit is possible.

58.2 Register Descriptions

58.2.1 SRAMSABARn : SRAM Security Attribute Boundary Address Register (n = 0 to 3)

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x400 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Boundary address between secure and non-secure. (Offset of the start address of non-secure region)	R/W

Note: S-TYPE-1, P-TYPE-1
Note: This register is write-protected by the PRCR_S.PRC4 bit.

SRAMSABARn specify the offset of the boundary between the secure and non-secure regions of each SRAM. Write the absolute address to the SRAMSABARn. When writing to SRAMSABARn, writing from b31 to b21 is ignored and the value written from b12 to b0 should be 0.

The region lower than the boundary address is marked as “Secure”, and higher than or equal to the boundary address is marked as “Non-secure”.

The boundary address is as follows:

0x2200_0000 + SRAMSABARn (Secure alias)

0x3200_0000 + SRAMSABARn (Non-secure alias)

For SRAMSABAR0

When the offset of the boundary address is 0x0000_0000, whole SRAM0 is marked as Non-secure.

When the offset of the boundary address is 0x0008_0000 or large, whole SRAM0 is marked as Secure.

For SRAMSABAR1

When the offset of the boundary address is 0x0008_0000 or less, whole SRAM1 is marked as Non-secure.

When the offset of the boundary address is 0x0010_0000 or large, whole SRAM1 is marked as Secure.

For SRAMSABAR2

When the offset of the boundary address is 0x0010_0000 or less, whole SRAM2 is marked as Non-secure.

When the offset of the boundary address is 0x0018_0000 or large, whole SRAM2 is marked as Secure.

For SRAMSABAR3

When the offset of the boundary address is 0x0018_0000 or less, whole SRAM3 is marked as Non-secure.

When the offset of the boundary address is 0x001A_0000 or large, whole SRAM3 is marked as Secure.

58.2.2 SRAMSAR : SRAM Security Attribution Register

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SRAM WTSA	—	—	—	—	SRAM SA3	SRAM SA2	SRAM SA1	SRAM SA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SRAMSA0	SRAM0 Register Security Attribution 0: Secure 1: Non-Secure	R/W
1	SRAMSA1	SRAM1 Register Security Attribution 0: Secure 1: Non-Secure	R/W
2	SRAMSA2	SRAM2 Register Security Attribution 0: Secure 1: Non-Secure	R/W
3	SRAMSA3	SRAM3 Register Security Attribution 0: Secure 1: Non-Secure	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
8	SRAMWTSA	SRAMWTSC Security Attribution 0: Secure 1: Non-Secure	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1
Note: Only Secure access can write to this register. Both Secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.
Note: This register is write-protected by the PRCR_S.PRC4 bit.

SRAMSA0 bit (SRAM0 Register Security Attribution)

Security attributes of SRAMCR0 registers. The target registers are as follows.

- SRAMCR0
- SRAMECCRGN0
- SRAMESCLR.CLR00, SRAMESCLR.CLR01

SRAMSA1 bit (SRAM1 Register Security Attribution)

Security attributes of SRAMCR1 registers. The target registers are as follows.

- SRAMCR1
- SRAMECCRGN1
- SRAMESCLR.CLR10, SRAMESCLR.CLR11

SRAMSA2 bit (SRAM2 Register Security Attribution)

Security attributes of SRAMCR2 registers. The target registers are as follows.

- SRAMCR2
- SRAMECCRGN2
- SRAMESCLR.CLR20, SRAMESCLR.CLR21

SRAMSA3 bit (SRAM3 Register Security Attribution)

Security attributes of SRAMCR3 registers. The target registers are as follows.

- SRAMCR3
- SRAMECCRGN3
- SRAMESCLR.CLR30, SRAMESCLR.CLR31

SRAMWTSA bit (SRAMWTSC Security Attribution)

Security attributes of SRAMWTSC.

58.2.3 SRAMESAR : SRAM ECC region Security Attribute Register

Base address: CPSCU = 0x4000_8000
CPSCU_NS = 0x5000_8000

Offset address: 0x510

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SRAM ESA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SRAMESA	ECC region Security Attribution 0: Secure 1: Non-Secure	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1
Note: This register is write-protected by PRCR_S.PRC4.

SRAMESA bit (ECC region Security Attribution)

This bit indicates the security attribute of the ECC region of all SRAMs.

58.2.4 SRAMPRCR_S : SRAM Protection Control Register for Secure

Base address: SRAM = 0x4000_2000
Offset address: 0x00

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KW[7:0]								—	—	—	—	—	—	—	PR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PR	Register Write Control 0: Writing to registers are disabled 1: Writing to registers are enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KW[7:0]	Write Key Code Key code protection to the PR bit	W

Note: S-TYPE-6, P-TYPE-2
Note: It is necessary to write by half word access.
Byte write access is prohibited. When byte write access is executed, operation is not guaranteed.

PR bit (Register Write Control)

The PR bit controls the write mode of the SRAMWTSC, SRAMCRn, SRAMECCRGNn registers that are marked as “Secure” by SRAMSAR register. When this bit is set to 1, writing to the SRAMWTSC, SRAMCRn, SRAMECCRGNn marked as “Secure” is enabled.

While writing to this register it is necessary to write 0xA5 to the KW[7:0] bits simultaneously.

KW[7:0] bits (Write Key Code)

The KW[7:0] bits enable or disable writing to the PR bit. When you write to the PR bit, write 0xA5 to the KW[7:0] bits simultaneously. When a value other than 0xA5 is written to KW[7:0], the PR bit is not updated. The KW[7:0] bits are always read as 0x00.

58.2.5 SRAMPRCR_NS : SRAM Protection Control Register for Non-Secure

Base address: SRAM_NS = 0x5000_2000
Offset address: 0x04

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KW[7:0]								—	—	—	—	—	—	—	PR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PR	Register Write Control 0: Writing to registers are disabled 1: Writing to registers are enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KW[7:0]	Write Key Code Key code protection to the PR bit	W

Note: S-TYPE-7, P-TYPE-2
Note: It is necessary to write by half word access.
Byte write access is prohibited. When byte write access is executed, operation is not guaranteed.

PR bit (Register Write Control)

The PR bit controls the write mode of the SRAMWTSC, SRAMCRn, SRAMECCRGNn registers that are marked as “Non-secure” by SRAMSAR register. When this bit is set to 1, writing to the SRAMWTSC, SRAMCRn, SRAMECCRGNn marked as “Non-secure” is enabled.

While writing to this register it is necessary to write 0xA5 to the KW[7:0] bits simultaneously.

KW[7:0] bits (Write Key Code)

The KW[7:0] bits enable or disable writing to the PR bit. When you write to the PR bit, write 0xA5 to the KW[7:0] bits simultaneously. When a value other than 0xA5 is written to KW[7:0], the PR bit is not updated. The KW[7:0] bits are always read as 0x00.

58.2.6 SRAMWTSC : SRAM Wait State Control Register

Base address: SRAM = 0x4000_2000
SRAM_NS = 0x5000_2000

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	WTEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	WTEN	SRAM wait enable 0: Do not add wait state in the access cycle to SRAMs. 1: Add wait state in the access cycle to SRAMs.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3 (SRAMSAR.SRAMWTSA), P-TYPE-2
Note: SRAMWTSC is valid only for SRAM0, 1, 2, 3.
Note: The protection register (SRAMPRCR_S or SRAMPRCR_NS) protects this register against writing. This register can be written only when the PR bit in the SRAMPRCR_S or SRAMPRCR_NS register is 1.

WTEN bit (SRAM wait enable)

This bit sets the wait cycle insertion to the access cycle of SRAM. When it is set 1 to the WTEN bit, 1 wait cycle inserted to the read access cycle of SRAM. Also When the WTEN bit is 1, and access to the same region[*1] and the same mat[*2] is continuous, 1 wait cycle is inserted to the second access cycle. When ICLK frequency is greater than half the maximum frequency in the electrical characteristics, it is necessary to set 1 wait cycle in the WTEN bit.

Note 1. The region of SRAM is divided to 128KB units. The ECC region is not divided.
Note 2. The mat is divided to 0x0 to 0x7 and 0x8 to 0xF in the lower 4-bits of the address.

For example, in the case of SRAMn, the same region and the same mat is 8 areas of SRAM region, and 2 areas of ECC region in the figure below. If access 1 area continuously, 1 wait cycle is inserted. If ECC function is enabled or bypass is enabled, ECC region is accessed, so if access to the same mat continues, 1 wait cycle is inserted.

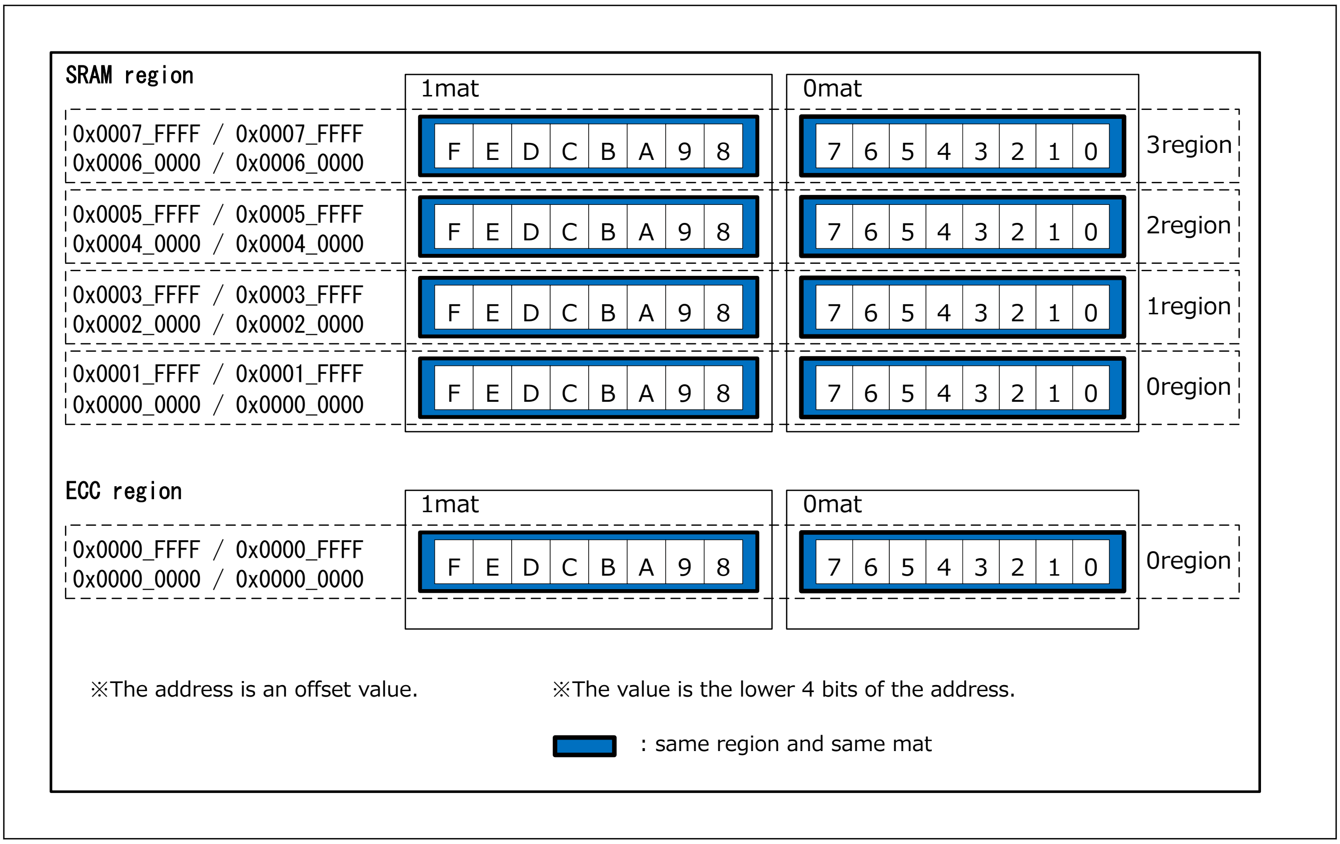


Figure 58.1 Mat configuration of each SRAMs

58.2.7 SRAMCRn : SRAM Control Register n For ECC RAM (n = 0 to 3)

Base address: SRAM = 0x4000_2000
SRAM_NS = 0x5000_2000

Offset address: 0x10 + 0x04 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TSTBYP	—	—	E1STSEN	ECCMOD[1:0]		—	OAD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OAD	Operation after error detection 0: Interrupt 1: Reset	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
3:2	ECCMOD[1:0]	ECC Operating Mode Select 0 0: Disable ECC function 0 1: Setting prohibited 1 0: Enable ECC function without error checking 1 1: Enable ECC function with error checking	R/W
4	E1STSEN	ECC 1-bit Error Update Enable 0: Disable updating of 1-bit ECC error 1: Enable updating of 1-bit ECC error	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	TSTBYP	ECC Test Enable / ECC Bypass Select 0: Disable ECC bypass 1: Enable ECC bypass	R/W

Note: S-TYPE-3 (SRAMSAR.SRAMSA0), P-TYPE-2

Note: The protection register (SRAMPRCR_S or SRAMPRCR_NS) protects this register against writing. This register can be written only when the PR bit in the SRAMPRCR_S or SRAMPRCR_NS register is 1.

If CPU0 and CPU1 issue speculative access to an uninitialized SRAM with ECC enabled, it may trigger an interrupt or reset. After enabling ECC function with error checking, initialize the SRAM before setting the interrupt in the IELSRn register or setting the reset in the OAD bit.

OAD bit (Operation after error detection)

This bit selects a reset or non-maskable interrupt when an ECC error is detected.

ECCMOD[1:0] bits (ECC Operating Mode Select)

These bits set the access mode to the ECC operating.

E1STSEN bit (ECC 1-bit Error Update Enable)

This bit enables or disables updating of the ECC 1-bit error status bit in the SRAMESR, in response to a 1-bit error. This bit also functions as an interrupt or a reset mask.

TSTBYP bit (ECC Test Enable / ECC Bypass Select)

This bit enables direct access to ECC code by bypassing the ECC function. The ECC Bypass function is used along with setting the ECCMOD[1:0] bits in the same register to 00b.

For "64-bit data with 8-bit ECC code"

The ECC must be accessed the same address. The ECC code is assigned to the lower 8 bits. When writing the ECC code, the bits above the ECC bits are ignored. When reading the ECC code, the bits above the ECC bits are invalid.

Note: For details about ECC test, see section 58.3.4. ECC Decoder Testing.

58.2.8 SRAMECCRGN0 : SRAM ECC Region Control Register 0

Base address: SRAM = 0x4000_2000
SRAM_NS = 0x5000_2000

Offset address: 0x30

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	ECCRGN[2:0]		
Value after reset:	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
2:0	ECCRGN[2:0]	ECC target region select The ECC target region selected by this register can be controlled by SRAMCR0. 0 0 0: No ECC target region 0 0 1: 0x2200_0000–0x2201_FFFF 0x3200_0000–0x3201_FFFF (128 KB) 0 1 0: 0x2200_0000–0x2203_FFFF 0x3200_0000–0x3203_FFFF (256 KB) 0 1 1: 0x2200_0000–0x2205_FFFF 0x3200_0000–0x3205_FFFF (384 KB) 1 0 0: 0x2200_0000–0x2207_FFFF 0x3200_0000–0x3207_FFFF (512 KB) Others: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3 (SRAMSAR.SRAMSAn), P-TYPE-2

Note: The protection register (SRAMPRCR_S or SRAMPRCR_NS) protects this register against writing. This register can be written only when the PR bit in the SRAMPRCR_S or SRAMPRCR_NS register is 1.

ECCRGN[2:0] bits (ECC target region select)

These bits select ECC target region in SRAM0.

58.2.9 SRAMECCRGN1 : SRAM ECC Region Control Register 1

Base address: SRAM = 0x4000_2000
SRAM_NS = 0x5000_2000

Offset address: 0x34

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	ECCRGN[2:0]		
Value after reset:	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
2:0	ECCRGN[2:0]	ECC target region select The ECC target region selected by this register can be controlled by SRAMCR1. 0 0 0: No ECC target region 0 0 1: 0x2208_0000–0x2209_FFFF 0x3208_0000–0x3209_FFFF (128 KB) 0 1 0: 0x2208_0000–0x220B_FFFF 0x3208_0000–0x320B_FFFF (256 KB) 0 1 1: 0x2208_0000–0x220D_FFFF 0x3208_0000–0x320D_FFFF (384 KB) 1 0 0: 0x2208_0000–0x220F_FFFF 0x3208_0000–0x320F_FFFF (512 KB) Others: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3 (SRAMSAR.SRAMSAn), P-TYPE-2
Note: The protection register (SRAMPRCR_S or SRAMPRCR_NS) protects this register against writing. This register can be written only when the PR bit in the SRAMPRCR_S or SRAMPRCR_NS register is 1.

ECCRGN[2:0] bits (ECC target region select)

These bits select ECC target region in SRAM1.

58.2.10 SRAMECCRGN2 : SRAM ECC Region Control Register 2

Base address: SRAM = 0x4000_2000
SRAM_NS = 0x5000_2000

Offset address: 0x38

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	ECCRGN[2:0]		
Value after reset:	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
2:0	ECCRGN[2:0]	ECC target region select The ECC target region selected by this register can be controlled by SRAMCR2. 0 0 0: No ECC target region 0 0 1: 0x2210_0000–0x2211_FFFF 0x3210_0000–0x3211_FFFF (128 KB) 0 1 0: 0x2210_0000–0x2213_FFFF 0x3210_0000–0x3213_FFFF (256 KB) 0 1 1: 0x2210_0000–0x2215_FFFF 0x3210_0000–0x3215_FFFF (384 KB) 1 0 0: 0x2210_0000–0x2217_FFFF 0x3210_0000–0x3217_FFFF (512 KB) Others: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3 (SRAMSAR.SRAMSAn), P-TYPE-2
Note: The protection register (SRAMPRCR_S or SRAMPRCR_NS) protects this register against writing. This register can be written only when the PR bit in the SRAMPRCR_S or SRAMPRCR_NS register is 1.

ECCRGN[2:0] bits (ECC target region select)

These bits select ECC target region in SRAM2.

58.2.11 SRAMECCRGN3 : SRAM ECC Region Control Register 3

Base address: SRAM = 0x4000_2000
SRAM_NS = 0x5000_2000

Offset address: 0x3C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	ECCRGN[2:0]		
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
2:0	ECCRGN[2:0]	ECC target region select The ECC target region selected by this register can be controlled by SRAMCR3. 0 0 0: No ECC target region 0 0 1: 0x2218_0000–0x2219_FFFF 0x3218_0000–0x3219_FFFF (128 KB) Others: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3 (SRAMSAR.SRAMSAn), P-TYPE-2

Note: The protection register (SRAMPRCR_S or SRAMPRCR_NS) protects this register against writing. This register can be written only when the PR bit in the SRAMPRCR_S or SRAMPRCR_NS register is 1.

ECCRGN[2:0] bits (ECC target region select)

These bits select ECC target region in SRAM3.

58.2.12 SRAMESR : SRAM Error Status Register For ECC RAM

Base address: SRAM = 0x4000_2000
SRAM_NS = 0x5000_2000

Offset address: 0x40

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	ERR31	ERR30	ERR21	ERR20	ERR11	ERR10	ERR01	ERR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ERR00	SRAM0 1-bit ECC Error Status 0: 1-bit ECC error has not occurred. 1: 1-bit ECC error has occurred.	R
1	ERR01	SRAM0 2-bit ECC Error Status 0: 2-bit ECC error has not occurred. 1: 2-bit ECC error has occurred.	R
2	ERR10	SRAM1 1-bit ECC Error Status 0: 1-bit ECC error has not occurred. 1: 1-bit ECC error has occurred.	R
3	ERR11	SRAM1 2-bit ECC Error Status 0: 2-bit ECC error has not occurred. 1: 2-bit ECC error has occurred.	R
4	ERR20	SRAM2 1-bit ECC Error Status 0: 1-bit ECC error has not occurred. 1: 1-bit ECC error has occurred.	R

Bit	Symbol	Function	R/W
5	ERR21	SRAM2 2-bit ECC Error Status 0: 2-bit ECC error has not occurred. 1: 2-bit ECC error has occurred.	R
6	ERR30	SRAM3 1-bit ECC Error Status 0: 1-bit ECC error has not occurred. 1: 1-bit ECC error has occurred.	R
7	ERR31	SRAM3 2-bit ECC Error Status 0: 2-bit ECC error has not occurred. 1: 2-bit ECC error has occurred.	R
15:8	—	These bits are read as 0.	R

Note: S-TYPE-5, P-TYPE-2
Note: This register is initialized after returning from software standby mode while the debugger is connected.

This register is cleared by the corresponding bit in SRAMESCLR register or resets other than Bus Error Reset and Memory Error Reset. If error and clear occur at the same time, clear has priority. Also during accessing from debugger it is to stop updating register.

ERRn0 bit (SRAMn 1-bit ECC Error Status)

This bit shows whether there is a 1-bit ECC error in SRAMn.

When ECC function with error checking is enabled and, if the SRAMCRn.E1STSEN bit is 1, the ERRn0 bit is set to 1 if a 1-bit error is detected.

When ERRn0 bit is set to 1, outputs a reset or an interrupt request according to the SRAMCRn.OAD bit.

ERRn1 bit (SRAMn 2-bit ECC Error Status)

This bit shows whether there is a 2-bit ECC error in SRAMn.

When ECC function with error checking is enabled, the ERRn1 bit is set to 1 if a 2-bit error is detected.

When ERRn1 bit is set to 1, outputs a reset or an interrupt request according to the SRAMCRn.OAD bit.

58.2.13 SRAMESCLR : SRAM Error Status Clear Register For ECC RAM

Base address: SRAM = 0x4000_2000
SRAM_NS = 0x5000_2000

Offset address: 0x48

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	CLR31	CLR30	CLR21	CLR20	CLR11	CLR10	CLR01	CLR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CLR00	SRAM0 1-bit ECC Error Status Clear Writing to the CLR00 clears SRAMESR.ERR00. 0: No effect 1: Clear 1-bit ECC error	W
1	CLR01	SRAM0 2-bit ECC Error Status Clear Writing to the CLR01 clears SRAMESR.ERR01. 0: No effect 1: Clear 2-bit ECC error.	W
2	CLR10	SRAM1 1-bit ECC Error Status Clear Writing to the CLR10 clears SRAMESR.ERR10. 0: No effect 1: Clear 1-bit ECC error.	W

Bit	Symbol	Function	R/W
3	CLR11	SRAM1 2-bit ECC Error Status Clear Writing to the CLR11 clears SRAMESR.ERR11. 0: No effect 1: Clear 2-bit ECC error.	W
4	CLR20	SRAM2 1-bit ECC Error Status Clear Writing to the CLR20 clears SRAMESR.ERR20. 0: No effect 1: Clear 1-bit ECC error.	W
5	CLR21	SRAM2 2-bit ECC Error Status Clear Writing to the CLR21 clears SRAMESR.ERR21. 0: No effect 1: Clear 2-bit ECC error.	W
6	CLR30	SRAM3 1-bit ECC Error Status Clear Writing to the CLR30 clears SRAMESR.ERR30. 0: No effect 1: Clear 1-bit ECC error.	W
7	CLR31	SRAM3 2-bit ECC Error Status Clear Writing to the CLR31 clears SRAMESR.ERR31. 0: No effect 1: Clear 2-bit ECC error.	W
15:8	—	The write value should be 0.	W

Note: S-TYPE-4 (SRAMSAR.SRAMSAn), P-TYPE-2

CLRn0 bit (SRAMn 1-bit ECC Error Status Clear)

This bit can clear 1-bit ECC error status bit in the SRAMESR. If error and clear occur at the same time, clear has priority.

SRAMESR.ERRn0 can be cleared, when writing 1 to CLRn0.

CLRn1 bit (SRAMn 2-bit ECC Error Status Clear)

This bit can clear 2-bit ECC error status bit in the SRAMESR. If error and clear occur at the same time, clear has priority.

SRAMESR.ERRn1 can be cleared, when writing 1 to CLRn1.

58.2.14 SRAMEARnm : SRAM Error Address Register nm For ECC RAM (n = 0 to 3, m = 0 to 1)

Base address: SRAM = 0x4000_2000
SRAM_NS = 0x5000_2000

Offset address: 0x50 + 0x10 × n + 0x04 × m

Bit position:	31 … 0
Bit field:	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	When an SRAM error occurs, it stores offset of an error address.	R

Note: S-TYPE-5, P-TYPE-2
Note: This register is initialized after returning from software standby mode while the debugger is connected.

This register is cleared by the corresponding bit in SRAMESCLR register, or resets other than Bus Error Reset and Memory Error Reset. If error and clear occur at the same time, clear has priority. Also during accessing from debugger it is to stop updating register.

The error address is as follows.

0x2200_0000 + SRAMEARnm (Secure alias)

0x3200_0000 + SRAMEARnm (Non-secure alias)

For SRAMEARn0

This register stores offset of the error address where 1-bit ECC error is detected. These bits hold offset of the error address that occurred first. These bits are cleared by clearing 1-bit ECC error from SRAMESCLR.

For SRAMEARn1

This register stores offset of the error address where 2-bit ECC error is detected. These bits hold offset of the error address that occurred first. These bits are cleared by clearing 2-bit ECC error from SRAMESCLR.

58.3 Operation

58.3.1 Module Stop Function

Power consumption can be reduced by setting module stop control register A (MSTPCRA) to stop supply of the clock signal to SRAM.

SRAMn (n = 0, 1, 2, 3) is controlled by SRAMn bit in MSTPCRA register and, in the case of 1, SRAMn becomes the clock stop state.

The SRAM is thus placed in the module-stop state by stopping supply of the clock signals. The SRAM operates after a reset.

SRAM is not accessible if it is in the module-stop state. A transition to the module-stop state should not be made while access to SRAM is in progress.

Access to the SRAM in the module-stop state is prohibited. If access is attempted, correct operation is not guaranteed.

For details on the MSTPCRA register, see section 11, Low Power Mode.

58.3.2 Correction of ECC Errors

Enabling and disabling of ECC error correction can be selected through the setting by ECCMOD[1:0] bits of SRAMCRn register. In the initial state, ECC error correction is disabled. The ECC check type is SEC-DED (Single-Error Correction and Double-Error Detection Code).

When ECC function is enabled, 8-bit check bits are appended to 64-bit data for writing. For reading, 72-bit (data: 64 bits, check bits: 8 bits) data is read out from the SRAM (ECC area).

When ECC function with error checking are enabled, error correction is done if a 1-bit error occurs and the ERRn0 bit in the SRAMESR register is set to 1 if the E1STSEN bit in the SRAMCRn register is 1. If a 2-bit error occurs, error detection is done and the ERRn1 bit in the SRAMESR register is set to 1, though error correction is not performed.

When ECC function without error checking is enable, error correction is done if a 1-bit error occurs but ERRn0 bit in the SRAMESR register is not updated although E1STSEN bit in the SRAMCRn register is 1. If a 2-bit error occurs, this error is detected but the ERRn1 bit in the SRAMESR register is not updated, and error correction is not performed.

When ECC function is disable, neither error correction nor error detection is done although 1-bit or 2-bit error occur.

So ERRn0 and ERRn1 bits are not updated.

When updating all the data after the occurrence of an error, the only support of 64 bit data writing.

Since the SRAM data is undefined after power on and release from Deep Software Standby mode, accessing the SRAM when ECC function is enabled and error checking is selected causes an ECC error to occur. Therefore, before using ECC function, initial writing with ECC function enabled and 64-bit data size, or initial writing with ECC function enabled and the error checking disabled, to the area to be used in the SRAM should be done.

58.3.3 ECC Error Interrupt Function

When ECC function with error checking is enabled, error checking is possible in the SRAMn. In the case of a 2-bit error, the ERRn1 bit in the SRAMESR register set to 1. In the case of a 1-bit error, the ERRn0 bit in the SRAMESR register set to 1 when SRAMCRn.E1STSEN bit is 1.

When the ECC 1-bit error is to be masked, it is necessary to set the SRAMCRn.E1STSEN bit to 0 to disable updating to 1 of the SRAMESR.ERRn0 bit. An ECC error will not be generated while ECC function is disabled or ECC function without error checking is enabled.

ECC error may result in interrupt or reset. When ECC error occurs, When the SRAMCRn.OAD bit is 1, outputs a reset request. When it is 0, outputs a non-maskable interrupt request to the ICU.

ECC Error interrupt occurs when one flag in the SRAMESR register is set. ECC Error interrupt continues to occur until the SRAMESR register flag is cleared.

58.3.4 ECC Decoder Testing

Figure 58.2 shows the ECC decoder testing flowchart.

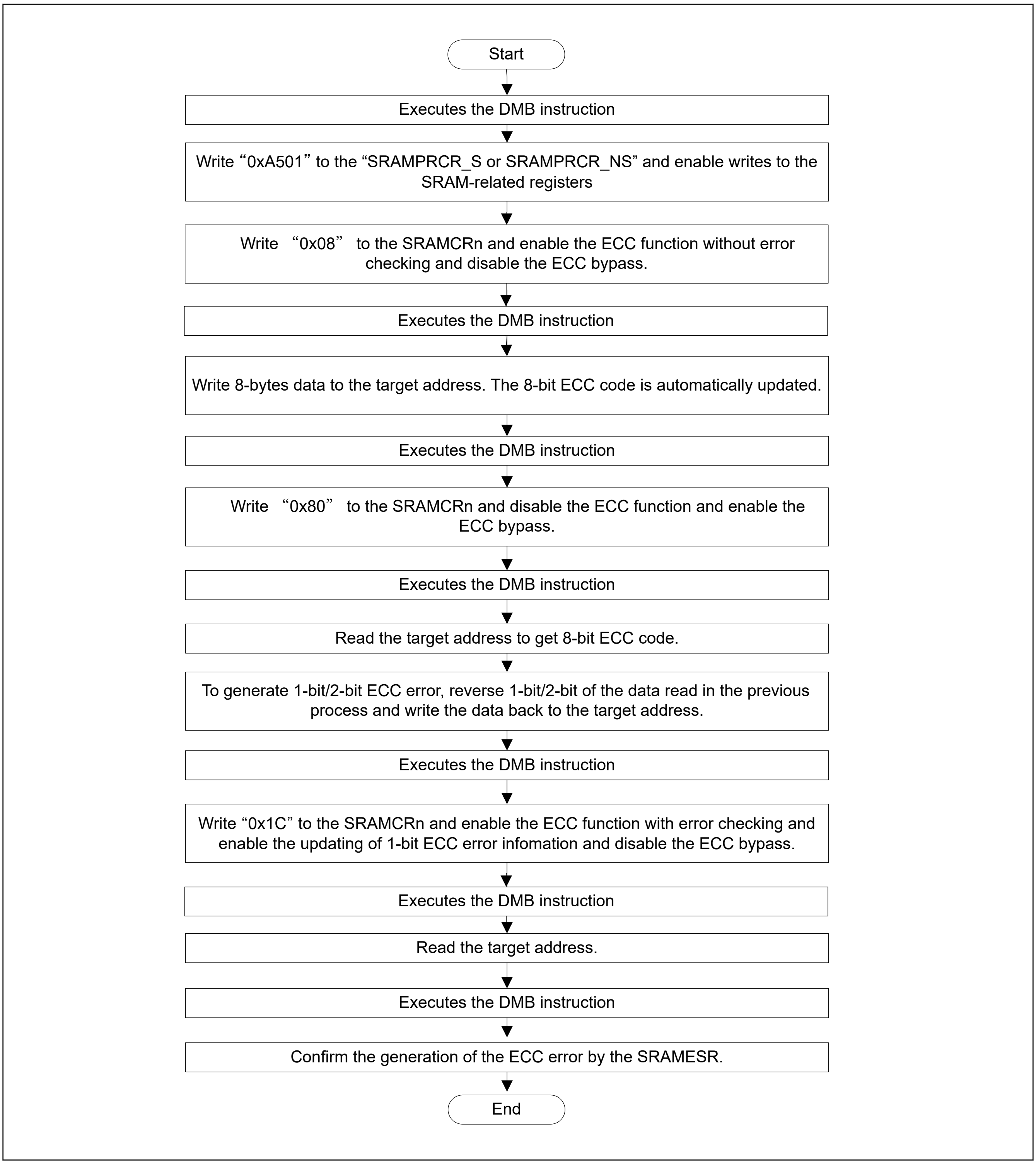


Figure 58.2 ECC decoder testing

58.3.5 TrustZone Filter Function

58.3.5.1 TrustZone Filter for SRAM Registers

SRAM registers can be protected with a Security Attribution (SA) from Non-secure or secure transaction. When SA indicates that SRAM registers are secure status, non-secure transaction can not access them because TrustZone Filter detects an error and protects the access. When SA indicates that SRAM registers are non-secure status, secure transaction can not access them because Trust Zone Filter detects an error and protects the access. SA for SRAM registers is just one to be used commonly among SRAM registers.

In the case of a failed access due to TrustZone error, error response is generated.

Table 58.2 Register protection

SA	Transaction	Write access	Read access
Secure	Secure	Permitted	Permitted
	Non-secure	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)
Non-secure	Secure	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)
	Non-secure	Permitted	Permitted

58.3.5.2 TrustZone Filter for SRAM Memory Regions

SRAM0, SRAM1, SRAM2, SRAM3 regions can divide into secure, non-secure independently. The access permissions for these regions are as below.

Table 58.3 Memory protection

SA	Transaction	Write access	Read access
Secure	Secure	Permitted	Permit
	Non-secure	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)
Non-secure	Secure	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)
	Non-secure	Permitted	Permitted

TrustZone Filter error for SRAM memory, generates an error notification.

58.3.6 Interrupt Source

The SRAM interrupt source includes an ECC error and TrustZone filter error. ECC error can choose interrupt or reset by the SRAMCRn.OAD bit. The SRAM interrupt occurs when one error status in the SRAMESR register is set to 1, the SRAM interrupt continues to occur until the SRAMESR register flag is cleared. When Common memory error occur (NMISR.CMST = 1 or RSTSR1.CMRF = 1), read SRAMESR and check SRAM interrupt source. When access is from the debugger, the error is detected and corrected, but no error flag is set, and no reset request and interrupt request is output.

Table 58.4 SRAM Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
ECCERR	ECC error (SRAMs with ECC)	Not possible	Not possible
TZFLT	TrustZone filter error	Not possible	Not possible

58.3.7 Wait State

When ICLK frequency is greater than half the maximum frequency in the electrical characteristics do not set 0x00 in wait enable bit for the SRAMWTSC register, in order to insert a wait cycle.

When the wait is not inserted, the operation is not guaranteed.

When ICLK frequency is less than or equal half the maximum frequency in the electrical characteristics wait state is not required.

When the maximum ICLK frequency is 250MHz

- 250MHz ≥ ICLK > 125MHz = 1 wait
- 125MHz ≥ ICLK = No-wait

When the maximum ICLK frequency is 200MHz

- 200MHz ≥ ICLK > 100MHz = 1 wait
- 100MHz ≥ ICLK = No-wait

When the maximum ICLK frequency is 150MHz

- 150MHz ≥ ICLK > 75MHz = 1 wait
- 75MHz ≥ ICLK = No-wait

58.3.8 Access Cycle

The following shows the delay information of the access cycle.

- When ECC is enabled, writing less than 64 bits is 2 cycles slower than normal writing.
- Insert the wait cycle according to the setting of SRAMWTSC.WTEN. If SRAMWTSC.WTEN = 1, the read will be delayed by one cycle.

58.3.9 Access to the ECC Region

When ECC function is disabled, ECC region can be used as data region.

When ECC function is disabled, direct access is possible. When ECC function is enabled, direct access is disabled. When bypass is enabled, direct access to ECC bit is possible.

Bypass	ECC	Access region	SRAM region	ECC region	Note
enable	—	to SRAM region	—	✔	—
—	—	to ECC region	—	—	Access invalid
disable	enable	to SRAM region	✔	✔	—
—	—	to ECC region	—	—	Access invalid
—	disable	to SRAM region	✔	—	—
—	—	to ECC region	—	✔	—

Note: When using bypass, set SRAMCRn.ECCMOD[1:0] = 00b. See section 58.2.7. SRAMCRn : SRAM Control Register n For ECC RAM (n = 0 to 3).

58.4 Usage Note

58.4.1 Instruction Fetch from SRAM area

When using SRAM to operate a program, initialize the SRAM area so that the CPU can correctly prefetch data. ECC error might occur if the CPU prefetches from an area that is not initialized. Initialize the additional 12-byte area from the end address of the program with the 8-byte boundary. Renesas recommends using the NOP instruction for data initialization.

58.4.2 Notes on Using Error Checking of SRAM

Data in SRAM are undefined when the power is turned on. Therefore, ECC errors occur if the data are read before initialization. The SRAM are read in 8-byte (64-bit) units. Initialize them on 8-byte boundaries.

59. MRAM

This MCU incorporates code MRAM and option-setting memory. The code MRAM stores instructions and operands. For option-setting memory, see section 7, Option-Setting Memory.

59.1 Overview

Table 59.1 lists the specification of MRAM, and Figure 59.1 shows a block diagram of the MRAM-related modules.

For the I/O pins used in serial programing mode, see Table 59.39.

The CPSEQ (Code MRAM programming sequencer) controls programming of the code MRAM. EPSEQ (Extra MRAM programming sequencer) controls programming of extra MRAM which includes option-setting memory. The MACI (MRAM application command interface) controls EPSEQ according to the specified MACI commands.

For the configuration of the code MRAM, see section 59.2. Structure of Memory.

Table 59.1 Specifications of MRAM (1 of 2)

Item	Code MRAM	Option-setting memory in extra MRAM
Memory capacity	User area: 1 MB max	See section 7, Option-Setting Memory.
Read cycle	See section 59.5.2. MRCFREQ : Code MRAM Frequency Notifications Register	See section 59.5.3. MREFREQ : Extra MRAM Frequency Notifications Register
Programming method	Write page buffer	MACI command
Protection	Protects against erroneous rewriting of the MRAM	
Dual bank function	Not available	Not available
Block swap function	Not available	Not available
Background operations (BGOs)	• The code MRAM can be read while option-setting memory is being programmed • Option-setting memory can be read while the code MRAM is being programmed.	
Units of programming	1 to 32 bytes	16 bytes
MACI command	Not available	Program: 16 bytes Forced stop Status clear Configuration set: 16 bytes Increment counter: 1 bit Read counter: 8 bytes

Security function	• Protect against illicit tampering with or reading out of data in MRAM
	• OFS area protection
	• Startup area select setting protection
	  – BTFLG, BTSIZE, and MSUACR registers are protected by the FSPR bit.
	• Permanent block protect setting protection
	  – User area is permanently protected from programming operation by the permanent block protect function.
	• MRAM protection for TrustZone
	  – Protection for MRAM area (program)
	  – Protection for MRAM area (read)
	  – Protection for register
	• Extra MRAM program mode entry protection
	• Anti-rollback counter
	• HUK protection	

Safety function	Software protection • MACI command protection by MENTRYR register • User area protection by MRCPC0 and MRCPC1 registers • User area protection by the block protection setting. Error protection • Error is detected when unintended commands or prohibited settings occur. The MACI command is not accepted after an error detection. Boot area protection • The startup area select function allows the customer to safely update the startup program. The size of the startup area can be selected from 8 KB, 16 KB, or 32 KB. ECC support for double-bit error correction and triple-bit error detection.	

Table 59.1 Specifications of MRAM (2 of 2)

Item	Code MRAM	Option-setting memory in extra MRAM
Interrupt request	● MRAM_MRCRD (Code MRAM read error) Enabled by INTENBTC and INTENBDC bits. ● MRAM_MRERD (Extra MRAM read error) Enabled by INTENBTE and INTENBDE bits. ● MRAM_MRCPR (Code MRAM sequencer error) Enabled by MRCAEIE bits. ● MRAM_MREPR (Extra MRAM sequencer error) Enabled by CMDLKIEE and MREAEIE bits. ● MRAM_ENDOFPE (Extra MRAM sequencer ready (processing end)) Enabled by MRDYIE bit.	

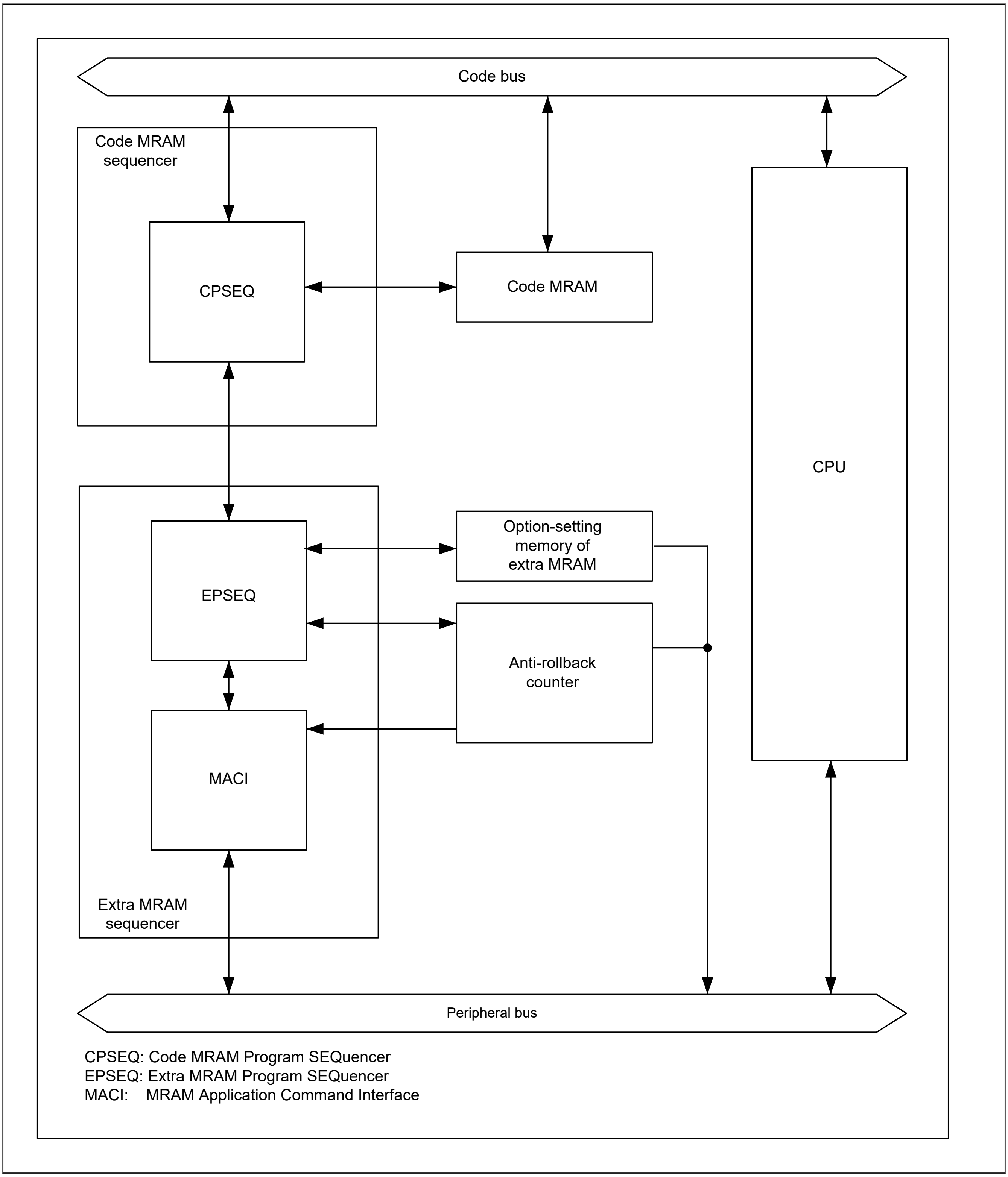
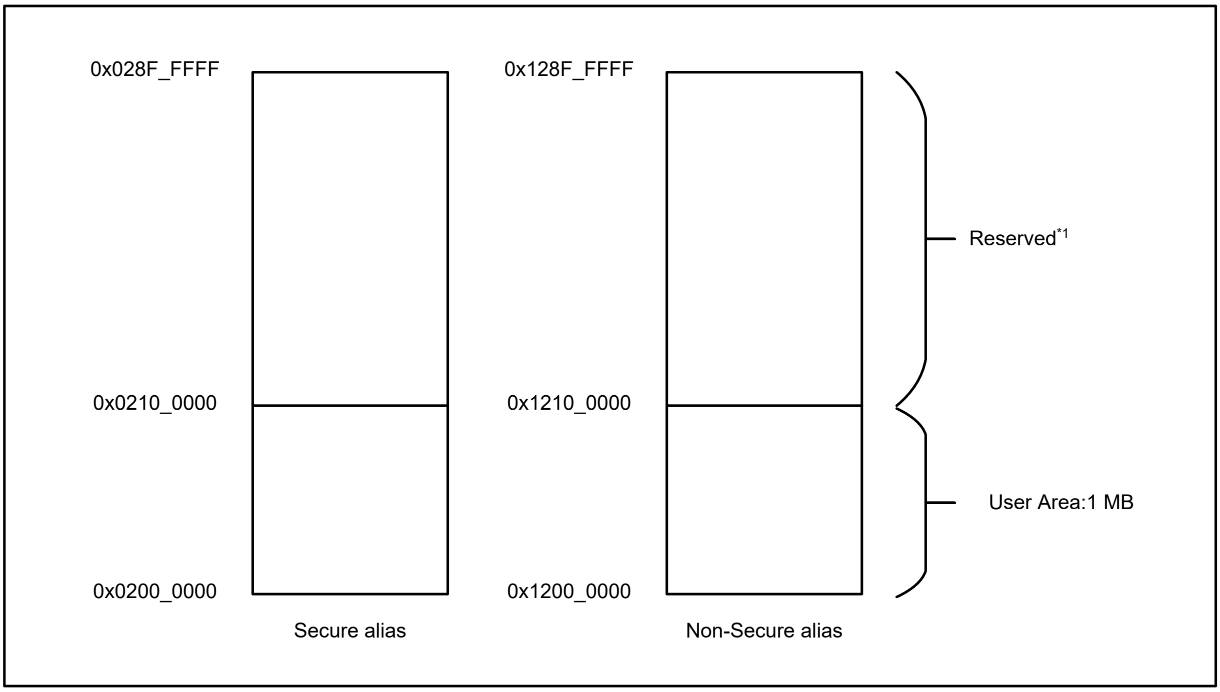


Figure 59.1 Block diagram of MRAM-related module

59.2 Structure of Memory

Figure 59.2 shows the memory map of the code MRAM.



Note 1. Bus error is generated.

Figure 59.2 Map of the code MRAM

Table 59.2 Read and programming address by product for the code MRAM

Product	Address
1 MB product	Secure alias: 0x0200_0000 to 0x020F_FFFF Non-Secure alias: 0x1200_0000 to 0x120F_FFFF

59.3 Address Space

Using the hardware interface with MRAM requires access to all registers of the hardware, which is for issuing MACI commands. The base address is shown in Table 59.3. Table 59.4 provides information about the hardware interface.

Table 59.3 Base address

Base address symbol	Address	Secure alias
BASE_MC	0x0200_0000 or 0x1200_0000	Secure or Non-Secure
BASE_MCS	0x0200_0000	Secure
BASE_MCN	0x1200_0000	Non-Secure
BASE_MR	0x4012_0000	Secure
	0x5012_0000	Non-Secure

Table 59.4 Information on the hardware interface area

Area	Address	Capacity
Area containing the various registers of the hardware	See section 59.5. Register Descriptions.	See section 59.5. Register Descriptions.
MACI command-issuing area	BASE_MR	4 bytes

For the address information of the extra MRAM, see section 5, Address Space.

59.4 MRAM Read and Program Function

59.4.1 Prefetch Buffer

Prefetch buffer speeds up CPU instruction fetches from bus master to code MRAM.

Table 59.5 Prefetch buffer overview

Function	Implementation
Target region	0x000_0000 to 0x00F_FFFF (1 MB)
Target bus master	CPU0 and CPU1 instruction fetch
Capacity	64 bytes/CPU
Associativity	Full associative
	256 bits per entry (256 bits aligned data), 2 entries
Access cycle	Hit: 0 wait Miss: See section 59.5.2. MRCFREQ : Code MRAM Frequency Notifications Register.

59.4.2 Programming Procedure

Program data for code MRAM is entered to the program buffer by bus access and is written to code MRAM in 32-bytes units. Renesas recommends the following to avoid system bus stall while processing code MRAM program.

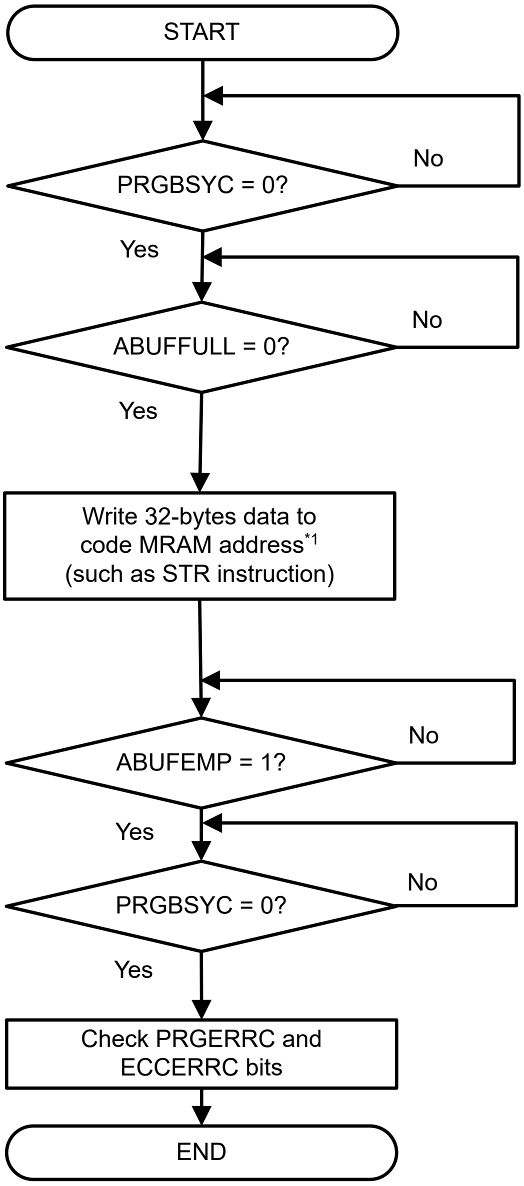
- Only one bus master can perform code MRAM write access
- Program data is 32 bytes or less in the same 32-bytes boundary.

The programming procedures of code MRAM are shown in Figure 59.3 and Figure 59.4. Before writing to code MRAM address, confirm that other program processing is not performed by PRGBSYC and ABUFFULL bits in the MRCPS register. The write data to the code MRAM address is stored in the 32-bytes program data buffer, and is programmed to the code MRAM in either of the following conditions.

- Program data buffer is full (Figure 59.3)
- Written code MRAM address is out of 32-bytes boundary from stored data in the program data buffer
- MRCFL bit in MRCFLR register is set to 1 (Figure 59.4).

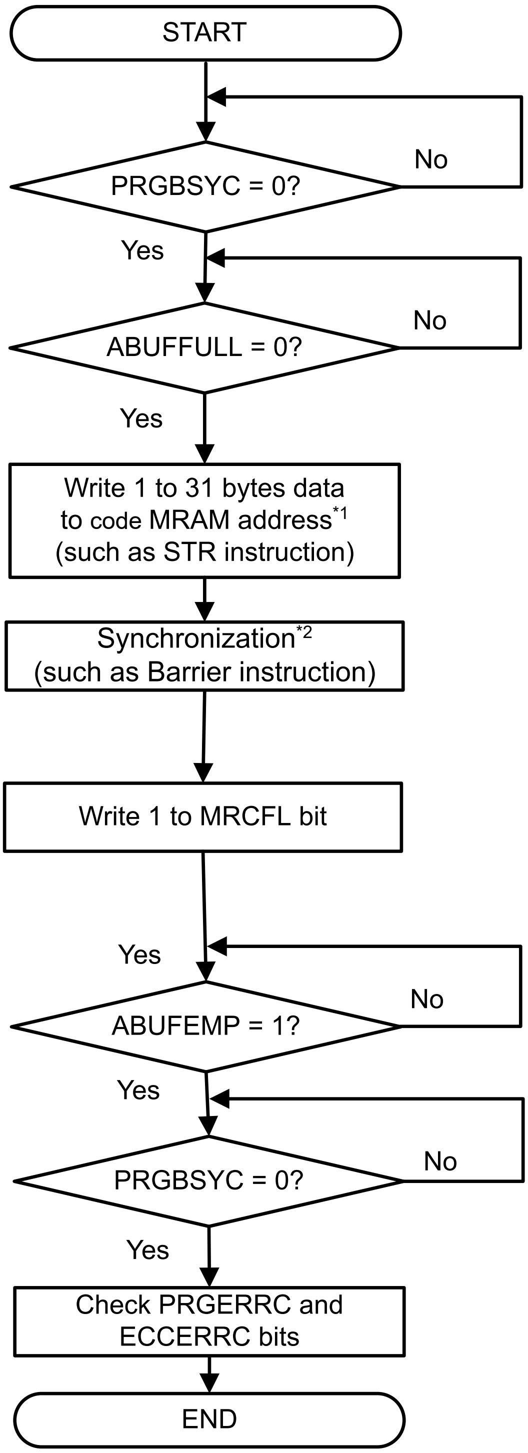
Program processing error status can be confirmed by the PRGERRC and ECCERRC bits in the MRCPS register.

See section 59.7. MACI Commands for programming procedure of extra MRAM.



Note 1. Program address starts from 32-bytes boundary and continuous address.

Figure 59.3 Code MRAM programming procedure (32-bytes programming)

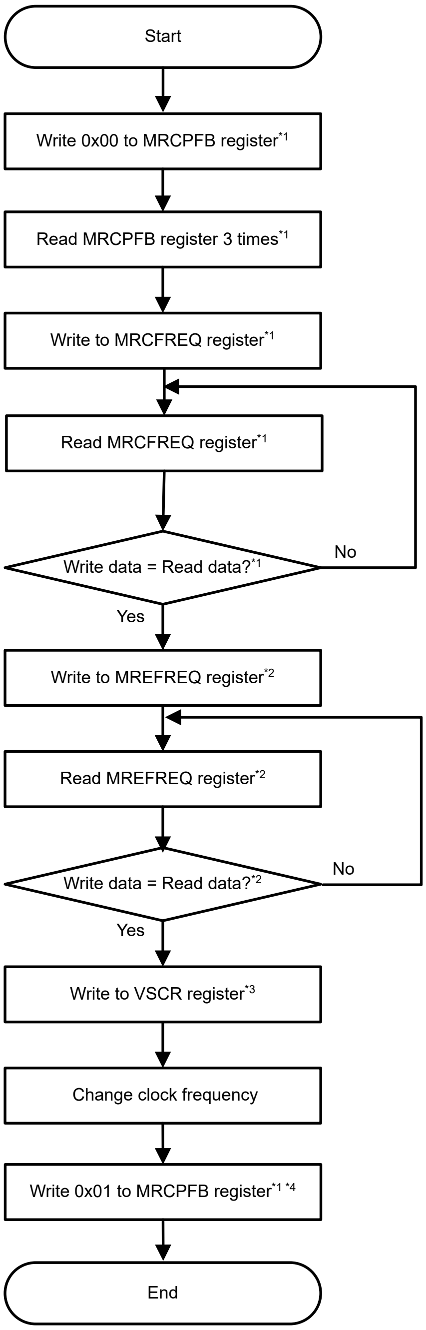


Note 1. Program address is continuous address.
Note 2. See section 59.14. Usage Notes for details.

Figure 59.4 Code MRAM programming procedure (smaller than 32-bytes programming)

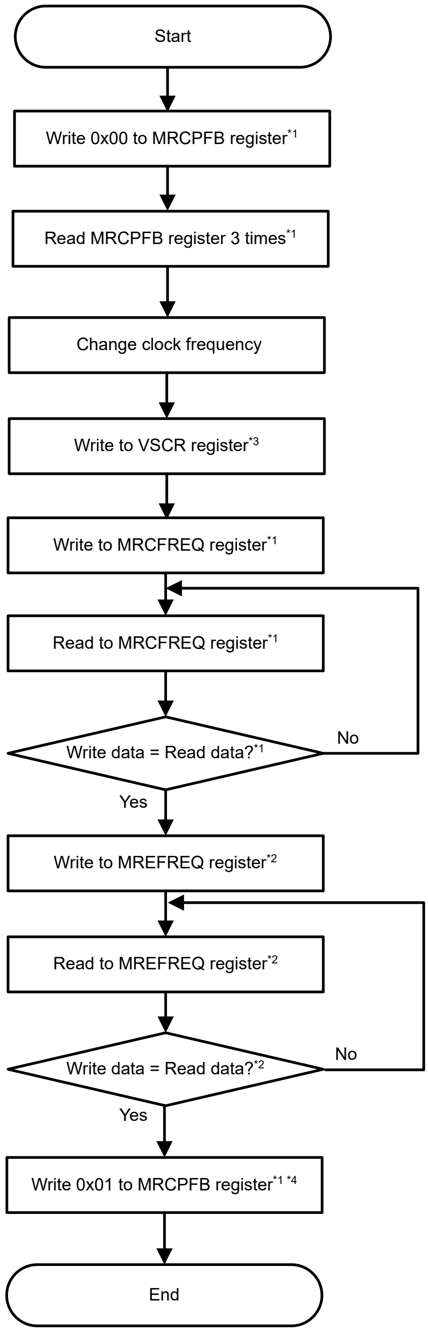
59.4.3 Frequency Change Procedure

The frequency setting is not reflected even if the register is written while programming or reading MRAM. To change the clock frequency, check that the register value is set to the write value before changing it. The frequency change procedures for code MRAM and extra MRAM are shown in Figure 59.5 and Figure 59.6.



Note 1. Only when changing MRCFREQ register.
Note 2. Only when changing MREFREQ register.
Note 3. Only when changing the voltage.
Note 4. When the value of MRCFREQ register is 0x65 or higher and prefetch buffer function is enabled to operate the device.

Figure 59.5 Frequency up procedure



Note 1. Only when changing MRCFREQ register.
Note 2. Only when changing MREFREQ register.
Note 3. Only when changing the voltage.
Note 4. When the value of MRCFREQ register is 0x65 or higher, and prefetch buffer function is enabled to operate the device.

Figure 59.6 Frequency down procedure

59.5 Register Descriptions

59.5.1 MRCPFB : Code MRAM Prefetch Buffer Enable Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x000

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	MPFB EN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MPFBEN	Code MRAM Prefetch Buffer Enable 0: Prefetch buffer disable 1: Prefetch buffer enable	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

MPFBEN bit (Code MRAM Prefetch Buffer Enable)

This bit is used to enable or disable code MRAM prefetch buffer.

It is prohibited to enable prefetch buffer when the MRCFREQ register value is 0x064 or less.

59.5.2 MRCFREQ : Code MRAM Frequency Notifications Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	MRCMHZ[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

Bit	Symbol	Function	R/W
9:0	MRCMHZ[9:0]	Code MRAM Frequency Notifications These bits are used to set the operating frequency of the code MRAM. Set the MRICLK frequency. Number of read access waits to code MRAM. MRCMHZ[9:0] ≤ 0x064 (100 MHz): non wait access 0x064 (100 MHz) < MRCMHZ[9:0] ≤ 0x0C8 (200 MHz): 1 wait access 0x0C8 (200 MHz) < MRCMHZ[9:0] ≤ 0x0FA (250 MHz): 2 wait access Other setting is reserved.	R/W*1
23:10	—	These bits are read as 0. The write value should be 0.	R/W
31:24	KEY[7:0]	Key Code	R/W*2

Note: S-TYPE-3, P-TYPE-2
Note 1. Writing to these bits is only possible when 32 bits are written, and the value written to the KEY bits is 0x1E.
Note 2. Written values are not retained by these bits (always reads 0x00).

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

The MRCFREQ register sets read and program frequency of code MRAM. For a faster clock frequency, set MRCMHZ[9:0] before changing the clock frequency. The ratio of code MRAM to extra MRAM frequency is limited to 5 or less. To slow down the clock frequency, set MRCMHZ[9:0] after changing the clock frequency. For information on the frequency setting, see section 59.4.3. Frequency Change Procedure.

MRCMHZ[9:0] bits (Code MRAM Frequency Notifications)

These bits specify the operating frequency of the code MRAM. Specifically, convert the frequency represented in MHz into a binary number and set it in these bits. If the operating frequency is 32.768 KHz or less, set the MCMHZ bits to 0x000.

Example:

Frequency is 249.9 MHz (MRCMHZ = 0x0FA)

Round up the first decimal place of 250 MHz to a whole number (= 250) and convert it into a binary number.

Frequency is 0.5 MHz (MRCMHZ = 0x001)

Round up the first decimal place of 0.5 MHz to a whole number (= 1) and convert it into a binary number.

Frequency is 32.768 kHz (MRCMHZ = 0x000)

KEY[7:0] bits (Key Code)

These bits control permission and prohibition of writing to the MCMHZ bits.

59.5.3 MREFREQ : Extra MRAM Frequency Notifications Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	MREMHZ[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

Bit	Symbol	Function	R/W
7:0	MREMHZ[7:0]	Extra MRAM Frequency Notifications These bits are used to set the operating frequency of the extra MRAM. Set the MRPCLK frequency. Number of read access waits to extra MRAM. MREMHZ[7:0] ≤ 0x64 (100 MHz): non wait access 0x64 (100 MHz) < MREMHZ[7:0] ≤ 0x7D (125 MHz): 1 wait access Other setting is reserved.	R/W*1
23:8	—	These bits are read as 0. The write value should be 0.	R/W
31:24	KEY[7:0]	Key Code	R/W*2

Note: S-TYPE-3, P-TYPE-2
Note 1. Writing to these bits is only possible when 32 bits are written, and the value written to the KEY bits is 0xE1.
Note 2. Written values are not retained by these bits (always reads 0x00).

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

MREFREQ sets the read and program frequency of the extra MRAM. For a faster clock frequency, set MREMHZ[7:0] before changing the clock frequency. To slow down the clock frequency, set MREMHZ[7:0] after changing the clock frequency. For information on the frequency setting, see section 59.4.3. Frequency Change Procedure.

MREMHZ[7:0] bits (Extra MRAM Frequency Notifications)

These bits specify the operating frequency of the extra MRAM. Specifically, convert the frequency represented in MHz into a binary number and set it in these bits. If the operating frequency is 32.768 kHz or less, set the MREMHZ[7:0] bits to 0x000.

Example:

Frequency is 124.9 MHz (MREMHZ = 0x7D)

Round up the first decimal place of 124.9 MHz to a whole number (= 125) and convert it into a binary number.

Frequency is 0.5 MHz (MREMHZ = 0x001)

Round up the first decimal place of 0.5 MHz to a whole number (= 1) and convert it into a binary number.

Frequency is 32.768 kHz (MREMHZ = 0x000)

KEY[7:0] bits (Key Code)

These bits control permission and prohibition of writing to the MREMHZ[7:0] bits.

59.5.4 MRCDECC : Code MRAM ECC Decoder Control Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x010

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	ECCSELC	DECDISC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DECDISC	Code MRAM ECC Decoder Disable 0: ECC decoder enable 1: ECC decoder disable	R/W*1
1	ECCSELC	Code MRAM ECC Bits Select 0: ECC bits unselect 1: ECC bits select	R/W*1
7:2	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code	R/W*2

Note: S-TYPE-3, P-TYPE-2
Note 1. Writing to these bits is only possible when 16 bits are written, and the value written to the KEY bits is 0x5A.
Note 2. Written values are not retained by these bits (always reads 0x00).

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

The MRCDECC register is used to test the ECC decoder circuits for the code MRAM. For information on the test flow of the ECC decoder circuits, see section 59.9.4. Fault Correction and Detection.

DECDISC bit (Code MRAM ECC Decoder Disable)

This bit enables or disables the ECC decoder circuit in the code MRAM.

[Setting condition]

- 1 is written to while write enabling conditions are satisfied.

[Clearing condition]

- 0 is written to while write enabling conditions are satisfied.

ECCSELC bit (Code MRAM ECC Bits Select)

The ECC bits of code MRAM is output to the [144:128] bits and [16:0] bits of the read data when ECCSELC is set to 1. A value of 0 is output for the [255:145] and [127:17] bits of the read data. When a DEC error occurs, the corrected data is output. Read data is not guaranteed when a TED error is detected.

[Setting condition]

- 1 is written to while write enabling conditions are satisfied.

[Clearing condition]

- 0 is written to while write enabling conditions are satisfied.

KEY[7:0] bits (Key Code)

These bits control permission and prohibition of writing to the DECDISC and ECCSELC bits.

59.5.5 MRCRAEINT : Code MRAM Read Access Error Interrupt Enable Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x014

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	INTEN BTC	INTEN BDC
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
0	INTENBDC	Code MRAM DEC Error Interrupt Enable 0: Generation of an MRAM_MRCRD interrupt request is disabled when MRCRAES.DECERRC is set to 1 1: Generation of an MRAM_MRCRD interrupt request is enabled when MRCRAES.DECERRC is set to 1	R/W
1	INTENBTC	Code MRAM TED Error Interrupt Enable 0: Generation of an MRAM_MRCRD interrupt request is disabled when MRCRAES.TEDERRC is set to 1 1: Generation of an MRAM_MRCRD interrupt request is enabled when MRCRAES.TEDERRC is set to 1	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

The MRCRAEINT register enables or disables generation of a code MRAM read access error (MRAM_MRCRD) interrupt request.

INTENBDC bit (Code MRAM DEC Error Interrupt Enable)

This bit enables or disables generation of an MRAM_MRCRD interrupt request when a code MRAM read access violation occurs. This violation leads to the DECERRC bit in the MRCRAES register being set to 1.

INTENBTC bit (Code MRAM TED Error Interrupt Enable)

This bit enables or disables generation of an MRAM_MRCRD interrupt request when a code MRAM read access violation occurs. This violation leads to the TEDERRC bit in the MRCRAES register being set to 1.

59.5.6 MRCRAES : Code MRAM Read Access Error Status Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x018

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TEDE RRC	DECE RRC
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DECERRC	Code MRAM DEC Error Detected 0: DEC error undetected 1: DEC error detected	R/W*1
1	TEDERRC	Code MRAM TED Error Detected 0: TED error undetected 1: TED error detected	R/W*1
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note 1. Only 0 can be written to clear the flag after 1 is read. These bits should be checked after 0 is written and rewritten to 0 if 1 is read.

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

DECERRC bit (Code MRAM DEC Error Detected)

This bit indicates the result of a DEC error check for read data. When accessing reads from the debugger, DECERRC is not set to 1 even if a DEC error occurs. To resolve the DEC error, read 32 bytes of data from the address indicated by the MRCRDEA register and write it back to the same address.

[Setting condition]

- Error detected
  Error is detected to read data of bus masters (this bit is not set when error is detected to prefetched data).

[Clearing condition]

- 0 is written after reading 1 from this bit.

TEDERRC bit (Code MRAM TED Error Detected)

This bit indicates the result of a TED error check for read data. When accessing reads from the debugger, TEDERRC is not set to 1 even if a TED error occurs. To resolve the TED error, write 32 bytes of data to the address indicated by the MRCRTEA register.

[Setting condition]

- Error detected
  Error is detected to read data of bus masters (this bit is not set when error is detected to prefetched data).

[Clearing condition]

- 0 is written after reading 1 from this bit.

59.5.7 MRCRTEA : Code MRAM TED Error Address Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x01C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MRCRTEA[26:11]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MRCRTEA[10:0]											—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	—	These bits are read as 0.	R
31:5	MRCRTEA[26:0]	Code MRAM Read Access TED Error Address Indicates the address when a TED error occurs.	R

Note: S-TYPE-3, P-TYPE-2

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

The MRCRTEA register stores the address when a TED error occurs in the code MRAM readout. The address of an ECC error is stored when the MRCRAES.TEDERRC bit is set to 1. MRCRTEA bits do not capture the error address when a TED error occurs until the MRCRAES.TEDERRC bit is cleared to 0.

MRCRTEA bits (Code MRAM Read Access TED Error Address)

[Setting condition]

- The TEDERRC bit in the MRCRAES register changes from 0 to 1.

59.5.8 MRCRDEA : Code MRAM DEC Error Address Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x020

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MRCRDEA[26:11]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MRCRDEA[10:0]											—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	—	These bits are read as 0.	R
31:5	MRCRDEA[26:0]	Code MRAM Read Access DEC Error Address Indicates the address when a DEC error occurs.	R

Note: S-TYPE-3, P-TYPE-2

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

The MRCRDEA register stores the address when a DEC error occurs in the code MRAM readout. The address of an ECC error is stored when the MRCRAES.DECERRC bit is set to 1. MRCRDEA bits do not capture the error address when a DEC error occurs until the MRCRAES.DECERRC bit is cleared to 0.

MRCRDEA[26:0] bits (Code MRAM Read Access DEC Error Address)

[Setting condition]

- The DECERRC bit in the MRCRAES register changes from 0 to 1.

59.5.9 MRERAEINT : Extra MRAM Read Access Error Interrupt Enable Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x034

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	INTEN BTE	INTEN BDE
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
0	INTENBDE	Extra MRAM DEC Error Interrupt Enable 0: Generation of an MRAM_MRERD interrupt request is disabled when MERAES.DECERRE is set to 1 1: Generation of an MRAM_MRERD interrupt request is enabled when MERAES.DECERRE is set to 1	R/W
1	INTENBTE	Extra MRAM TED Error Interrupt Enable 0: Generation of an MRAM_MRERD interrupt request is disabled when MERAES.TEDERRE is set to 1 1: Generation of an MRAM_MRERD interrupt request is enabled when MERAES.TEDERRE is set to 1	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

The MRERAEINT register enables or disables generation of an extra MRAM read access error (MRAM_MRERD) interrupt request.

INTENBDE bit (Extra MRAM DEC Error Interrupt Enable)

This bit enables or disables generation of an MRAM_MRERD interrupt request when an extra MRAM read access violation occurs. This violation leads to the DECERRE bit in the MRERAES register being set to 1.

INTENBTE bit (Extra MRAM TED Error Interrupt Enable)

This bit enables or disables generation of an MRAM_MRERD interrupt request when an extra MRAM read access violation occurs. This violation leads to the TEDERRE bit in the MRERAES register being set to 1.

59.5.10 MRERAES : Extra MRAM Read Access Error Status Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x038

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TEDERRE	DECERRE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DECERRE	Extra MRAM DEC Error Detected 0: DEC error undetected 1: DEC error detected	R/W*1
1	TEDERRE	Extra MRAM TED Error Detected 0: TED error undetected 1: TED error detected	R/W*1
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note 1. Only 0 can be written to clear the flag after 1 is read. These bits should be checked after 0 is written and rewritten to 0 if 1 is read.

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

DECERRE bit (Extra MRAM DEC Error Detected)

This bit indicates the result of a DEC error check for read data. When accessing reads from the debugger, DECERRE is not set to 1 even if a DEC error occurs. To resolve the DEC error, read 16 bytes of data from the address indicated by the MRERDEA register and write it back to the same address.

[Setting condition]

- Error detected
  Error is detected to read data of bus masters (this bit is not set when error is detected to prefetched data).

[Clearing condition]

- 0 is written after reading 1 from this bit.

TEDERRE bit (Extra MRAM TED Error Detected)

This bit indicates the result of a TED error check for read data. When accessing reads from the debugger, TEDERRE is not set to 1 even if a TED error occurs. To resolve the TED error, write 16 bytes of data to the address indicated by the MRERTEA register.

[Setting condition]

- Error detected
  Error is detected to read data of bus masters (this bit is not set when error is detected to prefetched data).

[Clearing condition]

- 0 is written after reading 1 from this bit.

59.5.11 MRERTEA : Extra MRAM TED Error Address Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x03C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MRERTEA[27:12]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MRERTEA[11:0]												—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0.	R
31:4	MRERTEA[27:0]	Extra MRAM Read Access TED Error Address Indicates the address when a TED error occurs.	R

Note: S-TYPE-3, P-TYPE-2

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

The MRERTEA register stores the address when a TED error occurs in the extra MRAM readout. The address of an ECC error is stored when the MRERAES.TEDERRE bit is set to 1. MRERTEA bits do not capture the error address when a TED error occurs until the MRERAES.TEDERRE bit is cleared to 0.

MRERTEA bits (Extra MRAM Read Access TED Error Address)

[Setting condition]

- The TEDERRE bit in the MRERAES register changes from 0 to 1.

59.5.12 MRERDEA : Extra MRAM DEC Error Address Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x040

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MRERDEA[27:12]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MRERDEA[11:0]												—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0.	R
31:4	MRERDEA[27:0]	Extra MRAM Read Access DEC Error Address Indicates the address when a DEC error occurs.	R

Note: S-TYPE-3, P-TYPE-2

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

The MRERDEA register stores the address when a DEC error occurs in the extra MRAM readout. The address of the ECC error is stored when the MRERAES.DECERRE bit is set to 1. The MRERDEA bits do not capture the error address when a DEC error occurs until the MRERAES.DECERRE bit is cleared to 0.

MRERDEA bits (Extra MRAM Read Access DEC Error Address)

[Setting condition]

- The DECERRE bit in the MRERAES register changes from 0 to 1.

59.5.13 MSAR : MRAM Security Attribution Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x100

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MRCPSEQSA	MREPSEQSA	MRCPSA	—	MACITRSA	MACICMRSA	MACICMISA	—	—	—	—	MPFBENSA	MRCFREQSA	MRCECCSA	MREFREQSA	MREECCSA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MREECCSA	Extra MRAM ECC Register Security Attribution 0: Secure 1: Non-Secure	R/W
1	MREFREQSA	MREFREQ Register Security Attribution 0: Secure 1: Non-Secure	R/W
2	MRCECCSA	Code MRAM ECC Register Security Attribution 0: Secure 1: Non-Secure	R/W
3	MRCFREQSA	MRCFREQ Register Security Attribution 0: Secure 1: Non-Secure	R/W

Bit	Symbol	Function	R/W
4	MPFBENSA	MRCPFB Register Security Attribution 0: Secure 1: Non-Secure	R/W
8:5	—	These bits are read as 0. The write value should be 0.	R/W
9	MACICMISA	MACI command Issuing Security Attribution 0: Secure 1: Non-Secure	R/W
10	MACICMRSA	MACI Command Registers Security Attribution 0: Secure 1: Non-Secure	R/W
11	MACITRSA	MACI Transfer Security Attribution 0: Secure 1: Non-Secure	R/W
12	—	This bit is read as 0. The write value should be 0.	R/W
13	MRCPSA	Code MRAM Program Register Security Attribution 0: Secure 1: Non-Secure	R/W
14	MREPSEQSA	EPSEQ Area Register Security Attribution 0: Secure 1: Non-Secure	R/W
15	MRCPSEQSA	CPSEQ Area Register Security Attribution 0: Secure 1: Non-Secure	R/W

Note: S-TYPE-1, P-TYPE-1

Write access is invalid when PRCR_S.PRC4 = 0. For details, see section 13, Register Write Protection Function.

MREECCSA bit (Extra MRAM ECC Register Security Attribution)

This bit sets the security attribute of MREDECC, MRERAEINT, MRERAES, MRERTEA, and MRERDEA registers.

MREFREQSA bit (MREFREQ Register Security Attribution)

This bit sets the security attribute of the MREFREQ register.

MRCECCSA bit (Code MRAM ECC Register Security Attribution)

This bit sets the security attribute of MRCDECC, MRCRAEINT, MRCRAES, MRCRTEA, and MRCRDEA registers.

MRCFREQSA bit (MRCFREQ Register Security Attribution)

This bit sets the security attribute of the MRCFREQ register.

MPFBENSA bit (MRCPFB Register Security Attribution)

This bit sets the security attribute of the MCPFB register.

MACICMISA bit (MACI command Issuing Security Attribution)

This bit sets the security attribute of the MRAM access area.

MACICMRSA bit (MACI Command Registers Security Attribution)

This bit sets the security attribute of MASTAT, MREPAEINT, MRDYIE, MSADDR, MSTATR, MENTRYR, MSUINITR, MCMDR, and MSUASMON registers.

MACITRSA bit (MACI Transfer Security Attribution)

This bit sets the security attribute of MCTRCNTR, MCTRLSR, and MCTRSTATR registers.

MRCPSA bit (Code MRAM Program Register Security Attribution)

This bit sets the security attribute of MRCPC0, MRCBPROT0, MRCPS, MRCPEAINT, and MRCPEA registers.

MREPSEQSA bit (EPSEQ Area Register Security Attribution)

This bit sets the security attribute of MRPSC and MREEECC registers.

MRCPSEQSA bit (CPSEQ Area Register Security Attribution)

This bit sets the security attribute of the MRCEECC register.

59.5.14 MREZS : Extra MRAM Zeroization Status Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x400

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	WHUK EXE	WHUK ZF
Value after reset:	1	1	1	1	1	1	0	1/0*1

Bit	Symbol	Function	R/W
0	WHUKZF	W-HUK Zero Flag Status 0: W-HUK is zeroized 1: W-HUK is not zeroized	R
1	WHUKEXE	W-HUK Zeroization Executing Status 0: W-HUK zeroization is not executing 1: W-HUK zeroization is executing	R
7:2	—	These bits are read as 0.	R

Note: S-TYPE-6, P-TYPE-2
Note 1. The initial value is changed by the status of W-HUK.

WHUKZF bit (W-HUK Zero Flag Status)

See section 59.8.6. HUK Protection for the update conditions of WHUKZF.

[Setting condition]

- W-HUK zeroing not executed.

[Clearing conditions]

- After W-HUK zeroing is complete.

WHUKEXE bit (W-HUK Zeroization Executing Status)

[Setting condition]

- W-HUK zeroization is started.

[Clearing condition]

- W-HUK zeroization is finished.

59.5.15 MREZC : Extra MRAM Zeroization Control Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x404

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	WHUKZE[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	WHUKZE[2:0]	W-HUK Zeroization Execute 0XX, X0X, X X 0: W-HUK zeroization not executes 1 1 1: W-HUK zeroization executes	R/W*1*2
7:3	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code	R/W*3

Note: S-TYPE-6, P-TYPE-2
Note 1. Writing to these bits is only possible when 16 bits are written, and the value written to the KEY bits is 0x3C.
Note 2. MREZC register cannot be written in the following conditions.

- When the W-HUK pseudo-zeroing operation in Table 59.31 has already been completed and pseudo-zeroing is to be started again by register writing without any system resets.
- If the operation of W-HUK zeroing in Table 59.31 has already been completed.

Note 3. Written values are not retained by these bits (always reads 0x00).

The MREZC register is used to trigger zeroization of W-HUK by secure software.

WHUKZE[2:0] bits (W-HUK Zeroization Execute)

0 cannot be written to these bits.

[Setting condition]

- 1 is written to while write enabling conditions are satisfied.

[Clearing condition]

- After W-HUK zeroing is complete.

59.5.16 MASTAT : Extra MRAM Access Status Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x2010

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	CMDLK	MREAE	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	MREAE	Extra MRAM Access Violation Flag 0: No extra MRAM access violation has occurred 1: An extra MRAM access violation has occurred	R/W*1
4	CMDLK	Command Lock Flag 0: The MRAM sequencer is not in the command-locked state 1: The MRAM sequencer is in the command-locked state	R
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note 1. Only 0 can be written to clear the flag after 1 is read.

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

The MASTAT register indicates whether an extra MRAM access violation has occurred. If any of the CMDLK and MREAE flags is set to 1, the extra MRAM sequencer enters the command-locked state (see section 59.9.2. Error Protection). To release it from the command-locked state, the Status Clear or Forced Stop command must be issued to the extra MRAM sequencer.

MREAE flag (Extra MRAM Access Violation Flag)

This flag indicates whether an extra MRAM access violation occurred. When this flag is set to 1, the ILGLERR bit in the MSTATR register is set to 1, placing the extra MRAM sequencer in the command-locked state.

[Setting conditions]

In general, commands are issued in extra MRAM program mode as follows.

- The setting of the MSADDR register is a reserved portion of the extra MRAM area.
- The Configuration Set command is issued while the setting of the MSADDR register is out of BASE_MC + 0x00C9 F020 to BASE_MC + 0x00C9 F0A0 and BASE_MC + 0x00C9 F0C0 to BASE_MC + 0x00C9 F7FF.
- MACI command is issued to extra MRAM with the wrong security attribution.

[Clearing conditions]

- When 0 is written to this flag after 1 is set
- When the extra MRAM sequencer starts to process the Status Clear or Forced Stop command.

CMDLK flag (Command Lock Flag)

This flag indicates that the extra MRAM sequencer is in the command-locked state.

[Setting condition]

- The extra MRAM sequencer detects an error and enters the command-locked state.

[Clearing condition]

- When an extra MRAM sequencer starts to process the Status Clear or Forced Stop command.

59.5.17 MPAEINT : Extra MRAM Access Error Interrupt Enable Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x2014

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	CMDLKIE	MREAEIE	—	—	—
Value after reset:	0	0	0	1	1	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	MREAEIE	Extra MRAM Access Violation Interrupt Enable 0: Generation of an MRAM_MREPR interrupt request is disabled when MASTAT.MREAE is set to 1 1: Generation of an MRAM_MREPR interrupt request is enabled when MASTAT.MREAE is set to 1	R/W
4	CMDLKIE	Command Lock Interrupt Enable 0: Generation of an MRAM_MREPR interrupt request is disabled when MASTAT.CMDLK is set to 1 1: Generation of an MRAM_MREPR interrupt request is enabled when MASTAT.CMDLK is set to 1	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

The MPAEINT register enables or disables generation of an extra MRAM access error (MRAM_MREPR) interrupt request.

MREAEIE bit (Extra MRAM Access Violation Interrupt Enable)

This bit enables or disables generation of an MRAM_MREPR interrupt request when an extra MRAM access violation occurs. This violation leads to the MREAE bit in the MASTAT register being set to 1.

CMDLKIE bit (Command Lock Interrupt Enable)

This bit enables or disables generation of an MRAM_MREPR interrupt request when an extra MRAM sequencer enters the command-locked state. This violation leads to the CMDLK bit in the MASTAT register being set to 1.

59.5.18 MRDYIE : Extra MRAM Ready Interrupt Enable Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x2018

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	MRDYIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MRDYIE	Extra MRAM Ready Interrupt Enable 0: Generation of an MRAM_ENDOFPE interrupt request is disabled 1: Generation of an MRAM_ENDOFPE interrupt request is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

The MRDYIE register enables or disables generation of an extra MRAM ready (MRAM_ENDOFPE) interrupt.

MRDYIE bit (Extra MRAM Ready Interrupt Enable)

This bit is used to enable or disable generation of an MRAM_ENDOFPE interrupt request when the MRDY bit in the MSTATR register is changed from 0 to 1. This change occurs on completion of processing by the extra MRAM sequencer of Program and Configuration set command.

59.5.19 MSADDR : MACI Command Start Address Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x2030

Bit position:	31 … 0
Bit field:	MSADDR[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	MSADDR[31:0]	Start Address for MACI Command Processing [Command]: [Address boundary] Program (extra MRAM): 16 bytes Configuration set (extra MRAM): 16 bytes	R/W*1

Note: S-TYPE-3, P-TYPE-2

Note 1. These bits can be written when the MRDY bit in the MSTATR register is 1. Writing to these bits are ignored when the MRDY bit is 0. Bits [3:0] are read-only.

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

The MSADDR register specifies the address where the target area for command processing starts when the MACI command for Program or Configuration set is issued.

The MSADDR value is initialized when the SUINIT bit in the MSUINITR register is set to 1. It is also initialized by a reset.

MSADDR[31:0] bits (Start Address for MACI Command Processing)

These bits specify the start address for MACI command processing. Bits [31:29] and bits [27:25] are ignored in MACI command processing for extra MRAM. Bit [28] is used for the security alias for extra MRAM sequencer command. When bit [28] is 0, the extra MRAM sequencer can update the secure region with secure access. When bit [28] is 1, the extra MRAM sequencer can update the non-security region with secure and non-secure access. Bits corresponding to address bits of lower order than the corresponding boundary listed above are also ignored.

See the User's Manual for information on the addresses of the extra MRAM.

See Table 59.15 and Table 59.16 for information on the addresses of Program and Configuration set commands.

59.5.20 MCNTSELR : MRAM Counter Select Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x2048

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	CNTSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CNTSEL[2:0]	Counter select 0 0 0: Anti-rollback counter is not selected. 0 0 1: ARC_SEC is selected. Counter size is 64 bits. 0 1 0: ARC_OEMBL is selected. Counter size is 64 bits. 0 1 1: Anti-rollback counter is not selected. 1 x x: ARC_NSEC is selected. Counter configuration is 64 or 256 bits. See Table 59.6, Table 59.7, and Table 59.8 for details of the counter select condition.	R/W*1
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-6, P-TYPE-2

Note 1. This bit can be written when MSTATR.MRDY = 1. Writing to this bit is ignored when MSTATR.MRDY = 0.

CNTSEL[2:0] bits (Counter select)

The MCNTSELR register specifies 3 types of anti-rollback counter (ARC_SEC, ARC_NSEC or ARC_OEMBL) in Increment Counter or Read Counter command.

The MCNTSELR value is initialized when the SUINIT bit in MSUINITR is set to 1.

[Setting condition]

- This register is written to in 8-bit access while MSTATR.MRDY = 1.

[Clearing condition]

- This register is written to in 8-bit access while MSTATR.MRDY = 1.

Before using the ARC_NSEC counter, it is necessary to set whether the counter is to be used with 4 lines × 64 bits or 1 line × 256 bits using the program command. The bits to be set are CNF_ARCNS0 and CNF_ARCNS1. If the CNF_ARCNS1 bit is set to 1, the ARC_NSEC counter cannot be used. Once the counter type is decided, it cannot be changed.

Table 59.6 Select condition of ARC_NSEC (64 bits) in increment/read counter command

MCNTSELR[2:0]	Target counter	Counter configuration
100b	ARC_NSEC[63:0]	Counter number is 4. Counter size is 64 bit.
101b	ARC_NSEC[127:64]	
110b	ARC_NSEC[191:128]	
111b	ARC_NSEC[255:192]	

Table 59.7 Select condition of ARC_NSEC (256 bits) in increment counter command

MCNTSELR[2:0]	Target counter	Counter configuration
100b	ARC_NSEC[255:0]	Counter number is 1. Counter size is 256 bit.
101b		
110b		
111b		

Table 59.8 Select condition of ARC_NSEC (256 bits) in read counter command

MCNTSELR[2:0]	Target counter	Counter configuration
100b	ARC_NSEC[63:0]	Counter number is 1. Counter size is 256 bit.
101b	ARC_NSEC[127:64]	
110b	ARC_NSEC[191:128]	
111b	ARC_NSEC[255:192]	

59.5.21 MCNTDTRn : MRAM Counter Data Register (n = 0, 1)

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x204C + 0x04 × n

Bit position:	31 ... 0
Bit field:	CNTRDAT[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	CNTRDAT[31:0]	Counter Read Data Reading data of 32 bits (LSB side of counter values) is output to MCNTDTR0. Reading data of 32 bits (MSB side of counter values) is output to MCNTDTR1.	R

Note: S-TYPE-6, P-TYPE-2

The MCNTDTR0 and MCNTDTR1 registers indicate anti-rollback counter values (64 bits) that read by the Read Counter command.

CNTRDAT[31:0] bits (Counter Read Data)

[Setting condition]

- MACI has accepted Read Counter command.

[Clearing condition]

- MACI has accepted the extra MRAM sequencer command except for the Read Counter command
- The extra MRAM sequencer enters the command-locked state
- W-HUK zeroization started.

59.5.22 MCTRCNTR : MRAM Configuration Update Transfer Control Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x2060

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	Key[7:0]								—	—	—	—	—	—	—	TRTRG
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TRTRG	Transfer Start Trigger 0: Configuration update transfer processing does not start 1: Configuration update transfer processing starts	R/W[*1 *2 *3]
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	Key[7:0]	Key code These bits enable or disable TRTRG bit modification. Data written to these bits are not stored.	R/W[*1]

Note: S-TYPE-3, P-TYPE-2
Note 1. Written data is not stored in these bits.
Note 2. This bit can be written when MCTRSTATR register is 0x00. Writing to this bit is ignored when MCTRSTATR register is other than 0x00 or when W-HUK zeroing is in progress.
Note 3. Writing to this bit is enabled only when this register is accessed in 16-bit size and 0x45 is written to the Key[7:0] bits.

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

The MCTRCNTR register specifies the start trigger and setting information in the configuration update transfer processing.

TRTRG bit (Transfer Start Trigger)

Starts the configuration update transfer processing.

[Setting condition]

- 1 is written to the TRTRG bit while write enabling conditions are satisfied.

59.5.23 MCTRLSR : MRAM Configuration Update Transfer List Select Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x2064

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TRLIST[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TRLIST[2:0]	Configuration Update Transfer List 0 0 1: Transfer list 1 is selected Others: Transfer list 1 is not selected	R/W[*1]
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2
Note 1. This bit can be written when the TRBUSY bit in the MCTRSTATR register is 0. Writing to this bit when TRBUSY = 1 is ignored.

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

The MCTRLSR register specifies the target transfer list in the configuration update transfer.

TRLIST[2:0] bits (Configuration Update Transfer List)

Transfer list 1 includes the coefficient value for DAC12.

[Setting condition]

- 1 is written to the TRLIST[2:0] bits while write enabling conditions are satisfied.

[Clearing condition]

- 0 is written to the TRLIST[2:0] bits while write enabling conditions are satisfied.

59.5.24 MCTRSTATR : MRAM Configuration Update Transfer Status Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x206C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TRMD	—	TRBUSY
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TRBUSY	Transfer Busy Status 0: Configuration update transfer is not processing 1: Configuration update transfer is processing	R
1	—	This bit is read as 0.	R
2	TRMD	Transfer Mode Setting Status 0: MENTRYR register is a data transferable setting 1: MENTRYR register is not a data transferable setting	R
7:3	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-2

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

TRBUSY bit (Transfer Busy Status)

Indicates status of the configuration update transfer processing.

[Setting condition]

- 1 is written to the TRTRG bit while write enabling conditions are satisfied.

[Clearing condition]

- The configuration update transfer processing is finished.

TRMD bit (Transfer Mode Setting Status)

Indicates the MENTRYR register setting that the configuration update transfer can be started.

[Setting condition]

- MENTRYR is 0x0080.

[Clearing condition]

- MENTRYR is 0x0000.

59.5.25 MSTATR : Extra MRAM Status Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x2080

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	ILGCOMERR	—	SECERR	OTERR	TZFERR	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MRDY	ILGLERR	—	PRGERR	—	—	—	—	—	—	CFGSETERR	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	—	These bits are read as 0.	R
5	CFGSETERR	Configuration Set Error Flag 0: Configuration set has been completed successfully 1: An error has occurred during configuration set	R
11:6	—	These bits are read as 0.	R
12	PRGERR	Programming Error Flag 0: Programming has been completed successfully 1: An error has occurred during programming	R
13	—	These bits are read as 0.	R
14	ILGLERR	Illegal Command Error Flag 0: The extra MRAM sequencer has not detected an illegal MACI command or illegal extra MRAM access 1: The extra MRAM sequencer has detected an illegal MACI command or illegal extra MRAM access	R
15	MRDY	Extra MRAM Ready Flag 0: Program, Configuration set, Increment Counter, or Read Counter command processing is in progress 1: None of the above is in progress	R
18:16	—	These bits are read as 0.	R
19	TZFERR	TrustZone Filter Error 0: A TrustZone filter error has not been detected 1: A TrustZone filter error has been detected	R
20	OTERR	Other Error 0: An error has not been detected 1: An error has been detected	R
21	SECERR	Security Error 0: A write protection error against MSUASMON.FSPR bit has not been detected 1: A write protection error against MSUASMON.FSPR bit has been detected	R
22	—	This bit is read as 0.	R
23	ILGCOMERR	Illegal Command Error 0: An illegal MACI command error has not been detected 1: An illegal MACI command error has been detected	R
31:24	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-2

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

The MSTATR register indicates the state of the extra MRAM sequencer.

CFGSETERR flag (Configuration Set Error Flag)

This flag indicates the result of an extra MRAM configuration setting. If this flag is 1, the extra MRAM sequencer is in the command-locked state.

[Setting condition]

- A TED error occurs when an extra MRAM sequencer reads data not to update from the configuration area during configuration setting.

[Clearing condition]

- The extra MRAM sequencer starts processing of the Status Clear or Forced Stop command.

PRGERR flag (Programming Error Flag)

This flag indicates the result of an extra MRAM programming. If this flag is 1, the extra MRAM sequencer is in the command-locked state.

[Setting condition]

- An error occurs during programming.

[Clearing condition]

- The extra MRAM sequencer starts processing of the Status Clear or Forced Stop command.

ILGLERR flag (Illegal Command Error Flag)

This flag indicates that the extra MRAM sequencer has detected an illegal MACI command or extra MRAM access. If this flag is 1, the extra MRAM sequencer is in the command-locked state.

[Setting Conditions] (See section 59.9.2. Error Protection)

- The extra MRAM sequencer has detected an illegal command
- The extra MRAM sequencer has detected illegal extra MRAM access.

[Clearing condition]

- The extra MRAM sequencer starts processing of the Status Clear or Forced Stop command.

MRDY flag (Extra MRAM Ready Flag)

This flag indicates the command processing state of the extra MRAM sequencer.

[Setting conditions]

- The extra MRAM sequencer completes command processing
- The extra MRAM sequencer has received the Forced Stop command and ended command processing
- W-HUK zeroing in processing.

[Clearing conditions]

- The extra MRAM sequencer has received the MACI command
- For Program and Configuration Set commands, the first write access to the MACI command-issuing area
- For other commands, the last write access to the MACI command issuing area.

TZFERR bit (TrustZone Filter Error)

This bit indicates a TrustZone protection error in the MACI command.

See Table 59.36. When this bit is 1, the extra MRAM sequencer enters the command-locked state.

[Setting condition]

- An error has been detected.

[Clearing condition]

- Status Clear or Forced Stop command processing is completed.

OTERR bit (Other Error)

This bit indicates that an MACI command has been issued when the condition of accepting commands is not satisfied.

See Table 59.36. When this bit is 1, the extra MRAM sequencer enters the command-locked state.

[Setting condition]

- An error has been detected.

[Clearing condition]

- Status Clear or Forced Stop command processing is completed.

SECERR bit (Security Error)

This bit indicates that the write protection by the MSUASMON.FSPR bit is violated.

See Table 59.36. When this bit is 1, the extra MRAM sequencer enters the command-locked state.

[Setting condition]

- An error has been detected.

[Clearing condition]

- Status Clear or Forced Stop command processing is completed.

ILGCOMERR bit (Illegal Command Error)

This bit indicates that the extra MRAM sequencer has detected an illegal MACI command.

See Table 59.36. When this bit is 1, the extra MRAM sequencer enters the command-locked state.

[Setting condition]

- An error has been detected.

[Clearing condition]

- Status Clear or Forced Stop command processing is completed.

59.5.26 MENTRYR : Extra MRAM Program Mode Entry Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x2084

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								MENTRY	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	MENTRY	Extra MRAM Program Mode Entry 0: Extra MRAM is in read mode 1: Extra MRAM is in program mode	R/W[*1][*2][*3]
15:8	KEY[7:0]	Key Code	R/W[*4]

Note: S-TYPE-3, P-TYPE-2
Note 1. These bits can be written when the MRDY bit in the MSTATR register is 1. Writing to these bits are ignored when the MRDY bit is 0.
Note 2. Writing to these bits is only possible when 16 bits are written, and the value written to the KEY bits is 0xAA.
Note 3. The MENTRY bit cannot be set to 1 during configuration update, and the MENTRYR register cannot be written to during W-HUK zeroing.
Note 4. Written values are not retained by these bits (always reads 0x00).

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

The MENTRYR register is used to specify the extra MRAM program mode. To specify an extra MRAM program mode so that the extra MRAM sequencer can receive MACI commands, set the MENTRY bit to 1 to place the extra MRAM sequencer in program mode.

The MENTRYR register is initialized when the SUINIT bit in MSUINITR is set to 1. It is also initialized by a reset.

MENTRY bit (Extra MRAM Program Mode Entry)

This bit specifies the extra MRAM program mode.

[Setting condition]

- Write 1 to the MENTRY bit while writing to MENTRYR is enabled and MENTRYR is 0x00.

[Clearing conditions]

- Write 8 bits to MENTRYR while the MRDY bit is 1
- A value other than 0xAA is specified in the KEY[7:0] bits and 16 bits are written to MENTRYR while the MRDY bit is 1
- Write 0 to the MENTRY bit while writing to MENTRYR is enabled

- Write to MENTRYR while writing is enabled and its value is other than 0x00.

KEY[7:0] bits (Key Code)

These bits control permission and prohibition of writing to the MENTRY bit.

59.5.27 MSUINITR : Extra MRAM Sequencer Setup Initialization Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x208C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	SUINIT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SUINIT	Setup Initialization 0: The MSADDR, MENTRYR, and MCNTSELR extra MRAM sequencer setup registers keep their current values 1: The MSADDR, MENTRYR, and MCNTSELR extra MRAM sequencer setup registers are initialized	R/W*1 *2
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code	R/W*3

Note: S-TYPE-3, P-TYPE-2
Note 1. This bit can be written when the MRDY bit in the MSTATR register is 1. Writing to this bit is ignored when the MRDY bit is 0.
Note 2. Writing to these bits is only possible when 16 bits are written, and the value written to the KEY bits is 0x2D.
Note 3. Written values are not retained by these bits (always reads 0x00).

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

The MSUINITR register is used for initialization of the extra MRAM sequencer setup.

SUINIT bit (Setup Initialization)

This bit initializes the following extra MRAM sequencer setup registers.

- MSADDR
- MENTRYR
- MCNTSELR

KEY[7:0] bits (Key Code)

These bits control permission and prohibition of writing to the SUINIT bit.

59.5.28 MCMDR : MACI Command Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x20A0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMDR[7:0]								PCMDR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	PCMDR[7:0]	Pre-command Flag The command just before the latest command is stored.	R

Bit	Symbol	Function	R/W
15:8	CMDR[7:0]	Command Flag The latest command is stored.	R

Note: S-TYPE-3, P-TYPE-2

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

The MCMDR register records the two most recent commands accepted by the extra MRAM sequencer.

PCMDR[7:0] flags (Pre-command Flag)

These flags indicate the command received immediately before the latest command received by the extra MRAM sequencer.

CMDR[7:0] flags (Command Flag)

These flags indicate the latest command received by the extra MRAM sequencer.

Table 59.9 States of MCMDR after receiving commands

Command	CMDR	PCMDR
Program	0xE8	Previous Command
Status Clear	0x50	Previous Command
Forced Stop	0xB3	Previous Command
Configuration set	0x40	Previous Command
Increment Counter	0xD0	0x35
Read Counter	0xD0	0x39

59.5.29 MSUASMON : MRAM Startup Area Select Monitor Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x20DC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	BTFLG	BTSIZE[1:0]		—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1/0	1/0	1/0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FSPR	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1/0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
14:0	—	These bits are read as 0.	R
15	FSPR	Protection flag of programming to set boot flag and startup area control 0: Protected state 1: Non-protected state	R
28:16	—	These bits are read as 0.	R
30:29	BTSIZE[1:0]	Size of startup area select for boot swap 0 0: Reserved 0 1: Startup area size is set to 32 KB 1 0: Startup area size is set to 16 KB 1 1: Startup area size is set to 8 KB	R
31	BTFLG	Flag of startup area select for boot swap 0: The startup area is the alternate block (block 1) 1: The startup area is the default block (block 0)	R

Note: S-TYPE-3, P-TYPE-2

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

FSPR flag (Protection flag of programming to set boot flag and startup area control)

This flag indicates the protection state against the Configuration Set command for the BTFLG bit, BTSIZE[1:0] bits, and the MSUACR register.

In response to a reset or Program command, the MACI transfers data from the extra MRAM to this register. The data corresponding to the FSPR flag is POFSPS[13] bit. See section 59.8.1. Permanent OFS Protect Setting for details.

BTSIZE[1:0] bits (Size of startup area select for boot swap)

These bits indicate the size of the startup area.

In response to a reset or Configuration Set command, the MACI transfers data from the extra MRAM to this register.

BTFLG flag (Flag of startup area select for boot swap)

This flag indicates whether the address of the startup area is exchanged for the boot swap function or not.

In response to a reset or Configuration Set command, the MACI transfers data from the extra MRAM to this register.

59.5.30 MSUACR : MRAM Startup Area Control Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x20E8

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	SAS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	SAS[1:0]	Startup Area Select 0 X: Startup area is selected by BTFLG bit 1 0: Startup area is temporarily switched to the default area 1 1: Startup area is temporarily switched to the alternate area	R/W*1
7:2	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code	R/W*2

Note: S-TYPE-6, P-TYPE-2

Note 1. The following described the write condition of this bit (these conditions are required at the same time).
1. The access size to this register is 16 bits.
2. The value of KEY [7:0] is 0x66.
3. The FSPR bit is 1.

Note 2. Written values are not retained by these bits (always reads 0x00).

The MSUACR register sets the startup area for the boot swap function.

SAS[1:0] bits (Startup Area Select)

These bits are used to select the startup area. Three methods are capable for changing the startup area.

KEY[7:0] bits (Key Code)

These bits control permission and prohibition of writing to the SAS[1:0] bits.

59.5.31 MRPSC : MRAM Program Speed Control Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x2800

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	MHSPEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MHSPEN	MRAM High-Speed Program Mode Enable 0: Normal program mode 1: High-speed program mode	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

MHSPEN bit (MRAM High-Speed Program Mode Enable)

This bit enables or disables high-speed program mode of the code MRAM or extra MRAM. After setting this bit to 1 and programming the code MRAM or extra MRAM, the mode is changed to high-speed program mode. To change to normal program mode, set this bit to 0, read the MRPSC register once, and program or read the MRAM. See section 59.6.2. MRAM High-speed Program Mode for details of mode changing.

59.5.32 MRCPC0 : Code MRAM Program Control Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x3000

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	MRCPNEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MRCPNEN	Code MRAM Program Enable for Non-secure 0: Code MRAM non-secure aliases are not programmable 1: Code MRAM non-secure aliases are programmable	R/W*1
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code	R/W*2

Note: S-TYPE-3, P-TYPE-2
Note 1. Writing to these bits is only possible when 16 bits are written, and the value written to the KEY bits is 0x86.
Note 2. Written values are not retained by these bits (always reads 0x00).

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

MRCPNEN bit (Code MRAM Program Enable for Non-secure)

The MRCPNEN bit enables programming of non-secure alias.

Setting the NVMWE bit in the DBGNVMCR register to 1 allows programming to a non-secure alias that is not protected by the block protection of code MRAM, regardless of the value of MRCPNEN.

59.5.33 MRCPC1 : Code MRAM Program Control Register for Secure

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x3004

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	MRCPSEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MRCPSEN	Code MRAM Program Enable for Secure 0: Code MRAM secure aliases are not programmable 1: Code MRAM secure aliases are programmable	R/W*1
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code	R/W*2

Note: S-TYPE-6, P-TYPE-2
Note 1. Writing to these bits is only possible when 16 bits are written, and the value written to the KEY bits is 0x68.
Note 2. Written values are not retained by these bits (always reads 0x00).

MRCPSEN bit (Code MRAM Program Enable for Secure)

The MRCPSEN bit enables programming of secure alias.

Setting the NVMWE bit in the DBGNVMCR register to 1 allows programming to a secure alias that is not protected by the block protection of the code MRAM, regardless of the value of MRCPSEN.

59.5.34 MRCBPROT0 : Code MRAM Block Protection Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x3008

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	BPCN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BPCN0	Code MRAM Block Protection Cancel for Non-secure 0: Code MRAM block protection is enabled 1: Code MRAM block protection is disabled	R/W*1 *2
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code	R/W*3

Note: S-TYPE-3, P-TYPE-2
Note 1. Writing to these bits is only possible when 16 bits are written, and the value written to the KEY bits is 0x78.
Note 2. Writing to this bit is ignored when the MRCPNEN bit in the MRCPC0 register is 0x0.
Note 3. Written values are not retained by these bits (always reads 0x00).

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

The MRCBPROT0 register is used to disable block protect function for non-secure. When the block protect setting is locked by permanent block setting, it cannot be disabled by this register.

BPCN0 bit (Code MRAM Block Protection Cancel for Non-secure)

This bit disables block protect for non-secure function.

[Setting condition]

- 1 is written to BPCN0 while the write enabling conditions are satisfied.

[Clearing conditions]

One of the following conditions is satisfied.

- 0 is written to BPCN0 while the write enabling conditions are satisfied
- 0 is written to the MRCPC0.MRCPNEN bit.

59.5.35 MRCBPROT1 : Code MRAM Block Protection Register for Secure

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x300C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	BPCN1
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BPCN1	Code MRAM Block Protection Cancel for Secure 0: Code MRAM block protection is enabled 1: Code MRAM block protection is disabled	R/W*1 *2
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code	R/W*3

Note: S-TYPE-6, P-TYPE-2

Note 1. Writing to these bits is only possible when 16 bits are written, and the value written to the KEY bits is 0xB1.

Note 2. Writing to this bit is ignored when the MRCPSEN bit in the MRCPC1 register is 0x0.

Note 3. Written values are not retained by these bits (always reads 0x00).

The MRCBPROT1 register is used to disable block protect function for secure. When the block protect setting is locked by permanent block setting, it cannot be disabled by this register.

BPCN1 bit (Code MRAM Block Protection Cancel for Secure)

This bit disables block protect for secure function.

[Setting condition]

- 1 is written to BPCN1 while the write enabling conditions are satisfied.

[Clearing conditions]

One of the following conditions is satisfied.

- 0 is written to BPCN1 while the write enabling conditions are satisfied
- 0 is written to the MRCPC1.MRCPSEN bit.

59.5.36 MRCPS : Code MRAM Program Status Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x3010

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PRGBSYC	ABUFFULL	ABUFEMP	—	—	—	ECCERRC	PRGERRC
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PRGERRC	Programming Error 0: Programming has been completed successfully 1: An error has occurred during programming	R/W*1

Bit	Symbol	Function	R/W
1	ECCERRC	ECC Error 0: Programming has been completed successfully 1: An error has occurred during programming	R/W*1
4:2	—	These bits are read as 0. The write value should be 0.	R/W
5	ABUFEMP	Address Buffer Empty 0: Address buffer is not empty, and code MRAM write data can flush 1: Address buffer is empty, and code MRAM write data cannot flush	R
6	ABUFFULL	Address Buffer Full 0: Address buffer is not full, and code MRAM write transaction is acceptable 1: Address buffer is full, and code MRAM write transaction is not acceptable	R
7	PRGBSYC	Code MRAM Program Busy 0: Code MRAM is not in program status 1: Code MRAM is in program status	R

Note: S-TYPE-3, P-TYPE-2

Note 1. Only 0 can be written to clear the flag after 1 is read. These bits should be checked after 0 is written and rewritten to 0 if 1 is read.

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

The MRCPS register indicates the code MRAM program status.

PRGERRC bit (Programming Error)

This bit indicates the result of code MRAM programming.

This bit can only be clear. Setting it is ignored.

When the DBGSTOPNVMER bit in the DBGSTOPCR register is set to 1, the PRGERRCbit is not set to 1 even if a program error occurs.

[Setting condition]

- An error has occurred during programming and the ECCERRC bit is 0.

[Clearing condition]

- 0 is written after reading 1 from this bit.

ECCERRC bit (ECC Error)

This bit indicates the occurrence of an uncorrectable error when programming data from the code bus is less than 32 bytes. Programming data to the code MRAM is generated from data in the target address and data from the code bus. If a TED error occurs when reading data from the target address, programming data to the code MRAM cannot be correctly generated if the data from the code bus is less than 32 bytes. To perform programming to an address with ECCERRC set to 1, all 32 bytes data must be input from the code bus.

This bit can only be clear. Setting it is ignored.

When the DBGSTOPNVMER bit in the DBGSTOPCR register is set to 1, the ECCERRC bit is not set to 1 even if an ECC error occurs.

[Setting condition]

- Uncorrectable error occurred when programming data from the code bus is less than 32 bytes and the PRGERRC bit is 0.

[Clearing condition]

- 0 is written after reading 1 from this bit.

ABUFEMP bit (Address Buffer Empty)

If the address buffer for the code MRAM programming is empty, the ABUFEMP bit is set to 1. When the ABUFEMP bit is 1, data in the program data buffer cannot be flushed because the MRCFL bit in the MRCFLR register cannot be written.

ABUFFULL bit (Address Buffer Full)

If the address buffer for the code MRAM programing is full, the ABUFFULL bit is set to 1. When the ABUFFULL bit is 1, a new write transaction cannot be accepted until the code MRAM programing starts or finishes.

PRGBSYC bit (Code MRAM Program Busy)

If CPSEQ executes the code MRAM programming processing, the PRGBSYC bit is set to 1. When the PRGBSYC bit is 1, a new read transaction to the code MRAM cannot be accepted until the code MRAM programming finishes.

59.5.37 MRCPAEINT : Code MRAM Program Access Error Interrupt Enable Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x3014

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MRCAEIE	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	MRCAEIE	Code MRAM Program Access Error Interrupt Enable 0: Generation of an MRAM_MRCPR interrupt request is disabled when the ECCERRC or PRGERRC bits are 1 1: Generation of an MRAM_MRCPR interrupt request is enabled when the ECCERRC or PRGERRC bits are 1	R/W

Note: S-TYPE-3, P-TYPE-2

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

The MRCPAEINT register enables or disables generation of a code MRAM program error (MRAM_MRCPR) interrupt request.

MRCAEIE bit (Code MRAM Program Access Error Interrupt Enable)

The MRCAEIE bit enables or disables MRAM_MRCPR interrupt request when a code MRAM program access error occurs and the ECCERRC or PRGERRC bits in MRCPS becomes 1.

59.5.38 MRCPEA : Code MRAM Program Error Address Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x3018

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MCPEA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MCPEA[10:0]											—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	—	These bits are read as 0.	R
31:5	MCPEA[26:0]	Code MRAM Program Error Address Indicates the address when a program error occurs.	R

Note: S-TYPE-3, P-TYPE-2

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

The MRCPEA register stores the error address when a program or ECC error occurs.

MCPEA[26:0] bits (Code MRAM Program Error Address)

[Setting condition]

- When a program or ECC error occurs while the ECCERRC bit is 0 and the PROGERRC bit is 0, the error address is stored in the MRCPEA register.

59.5.39 MRCFLR : Code MRAM Flush Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x3030

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	MRCFL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MRCFL	Flush Write Data Buffer for Code MRAM 0: Not execute to flush program data buffer 1: Execute to flush program data buffer	R/W*1 *2
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code	R/W*3

Note: S-TYPE-5, P-TYPE-2
Note 1. Writing to these bits is only possible when 16 bits are written, and the value written to the KEY bits is 0xC3.
Note 2. When the ABUFEMP bit in the MRCPS register is 1, it is not possible to write 1 to this bit.
Note 3. Written values are not retained by these bits (always reads 0x00).

The MRCFLR register is used for programming data less than 32 bytes.

MRCFL bit (Flush Write Data Buffer for Code MRAM)

This bit is used for starting code MRAM programming operation.

[Setting condition]

- 1 is written to while write enabling conditions are satisfied.

[Clearing condition]

- When the ABUFEMP bit in the MRCPS register is set to 1.

59.5.40 MRCEECC : Code MRAM ECC Encoder Control Register

Base address: MRAM = 0x4013_C000
MRAM_NS = 0x5013_C000

Offset address: 0x3804

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	ECCBYPC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ECCBYPC	Code MRAM ECC Encoder Outputs Bypass Enable 0: ECC encoder output is programmed to the ECC bits of the code MRAM 1: Bits [144:128] and bits [16:0] of written data are programmed to the ECC bits of the code MRAM	R/W*1

Bit	Symbol	Function	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code	R/W*2

Note: S-TYPE-3, P-TYPE-2
Note 1. Writing to these bits is only possible when 16 bits are written, and the value written to the KEY bits is 0xCD.
Note 2. Written values are not retained by these bits (always reads 0x00).

Associated SA: MSAR register. See section 59.5.13. MSAR : MRAM Security Attribution Register.

ECCBYPC bit (Code MRAM ECC Encoder Outputs Bypass Enable)

The ECCBYPC bit is used to test the ECC decoder circuits for the code MRAM. For information on the test flow of the ECC decoder circuits, see section 59.9.4. Fault Correction and Detection.

59.5.41 UIDRn : Unique ID Registers n (n = 0 to 3)

Address: 0x02F0_7B00 + 4 × n

Bit position:	31	0
Bit field:	UID	
Value after reset:	Unique value for each chip	

Bit	Symbol	Function	R/W
31:0	UID	Unique ID	R

Note: S-TYPE-5, P-TYPE-5

The UIDRn is a read-only register that stores a 16-byte ID code (unique ID) for identifying the individual MCU. The UIDRn register should be read in 32-bit units. When reading by the signature request command of the serial programming interface, data is read in order from the data with the large address. That is, the data in 0x02F0_7B0F is read first, and in 0x02F0_7B00 is read last.

59.5.42 PNRn : Part Numbering Register n (n = 0 to 3)

Address: 0x02C1_EC38 + 4 × n

Bit position:	31	0
Bit field:	PNR	
Value after reset:	Unique value for each chip	

Bit	Symbol	Function	R/W
31:0	PNR	Part Number	R

Note: S-TYPE-5, P-TYPE-5

The PNRn is a read-only register that stores a 16-byte part numbering. The PNRn register should be read in 32-bit units. Each byte corresponds to the ASCII code representation of the product part number as detailed in Table 1.13. The first character (R, 0x52 in ASCII code) of the part number is stored in the byte with the smallest address (0x02C1_EC38). When reading by the signature request command of the serial programming interface, the data is read in order from the data with the small address. That is, the data in 0x02C1_EC38 is read first, and in 0x02C1_EC47 is read last.

59.5.43 MCUVER : MCU Version Register

Address: 0x02C1_EC48

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MCUVE							
Value after reset:	Values depend on the chip							

Bit	Symbol	Function	R/W
7:0	MCUVE	MCU Version	R

Note: S-TYPE-5, P-TYPE-5

The MCUVER is a read-only register that stores the MCU version. The MCUVER register should be read in 8-bit units.

59.5.44 MWMCR : OTP Write Mode Control Register

Base address: SYSC = 0x4001_E000
SYSC_NS = 0x5001_E000

Offset address: 0xAB4

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	MWM[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	MWM[1:0]	OTP Write Mode [1:0] 0 0: OTP normal speed write mode 0 1: OTP high speed write mode 0 1 0: OTP high speed write mode 1 1 1: Prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

The MWMCR register controls the OTP write mode when writing data to OTP.

MWM[1:0] bits (OTP Write Mode [1:0])

1. MWM[1:0] = 00: OTP is set to OTP normal speed write mode.
   You can set MWM[1:0] to 00 when 1.62 V ≦ VCC ≦ 3.63 V.
   Refer to the write period of OTP normal speed write mode to section 62, Electrical Characteristics.
2. MWM[1:0] = 01: OTP is set to OTP high speed write mode 0.
   You can set MWM[1:0] to 01 when 2.5 V ≦ VCC ≦ 3.63 V.
   Refer to the write period of OTP high speed write mode 0 to section 62, Electrical Characteristics.
3. MWM[1:0] = 10: OTP is set to OTP high speed write mode 1.
   You can set MWM[1:0] to 10 when 3.0 V ≦ VCC ≦ 3.63 V.
   Refer to the write period of OTP high speed write mode 1 to section 62, Electrical Characteristics.
   While writing OTP in high speed mode 1:
   - Make CPUCLK0 and ICLK frequencies the same
   - Stop ESWM operation
   - Do not call Open process in the API of security engine.

59.6 Operating Mode

59.6.1 MRAM Read/Program Mode

Code MRAM and extra MRAM have modes for reading and programming as shown in Figure 59.7.

Transition from code MRAM read mode to code MRAM program mode is initiated by a software request generated by either of following conditions.

Programming of code MRAM starts automatically after transition to code MRAM program mode. When programming of code MRAM is completed, mode is transit from code MRAM program mode to code MRAM read mode automatically.

- Program data buffer is full
- Written code MRAM address is out of 32 bytes boundary from stored data in a program data buffer
- The MRCFL bit in the MRCFLR register is set to 1.

Transitions between extra MRAM read mode and extra MRAM program mode are initiated by changing the value of the MENTRYR register or by W-HUK zeroization. See section 59.8.6. HUK Protection. MRAM is in extra MRAM read mode when the MENTRYR register is 0x0000 and W-HUK zeroization is not executing. MRAM is in extra MRAM program mode in other cases. Programming of extra MRAM can be started using the MACI command in extra MRAM program mode or by starting W-HUK zeroization.

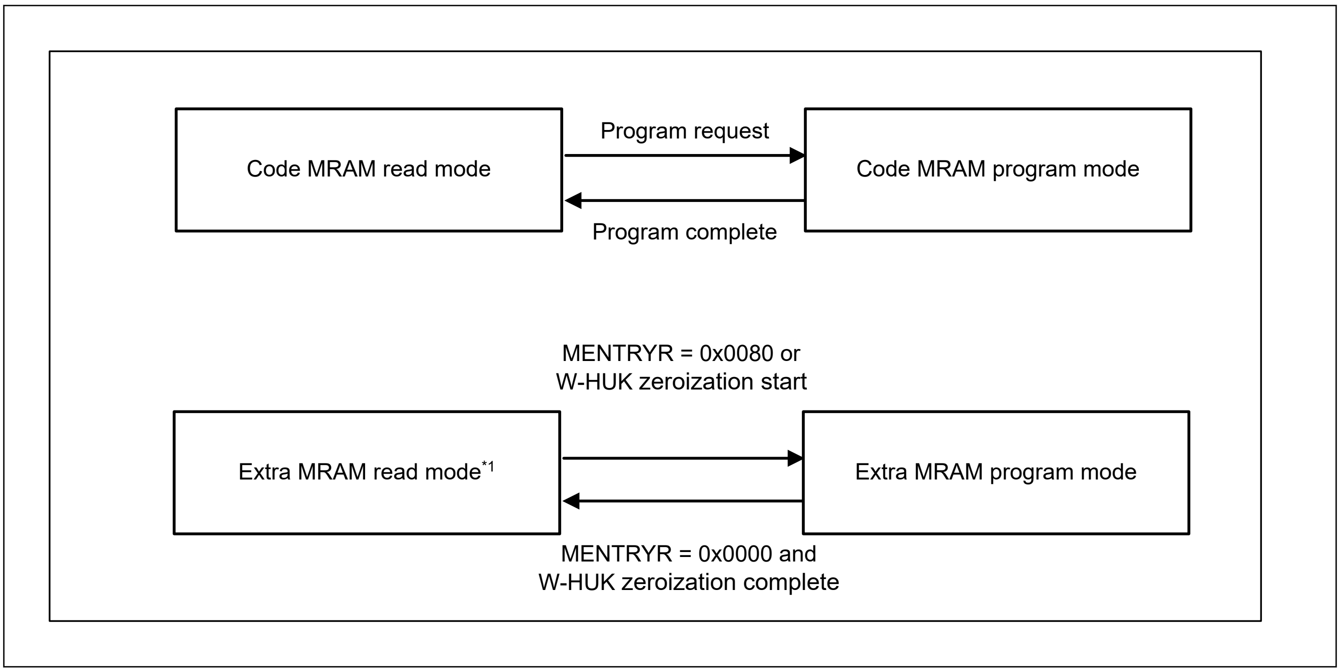


Figure 59.7 MRAM read/program mode transition

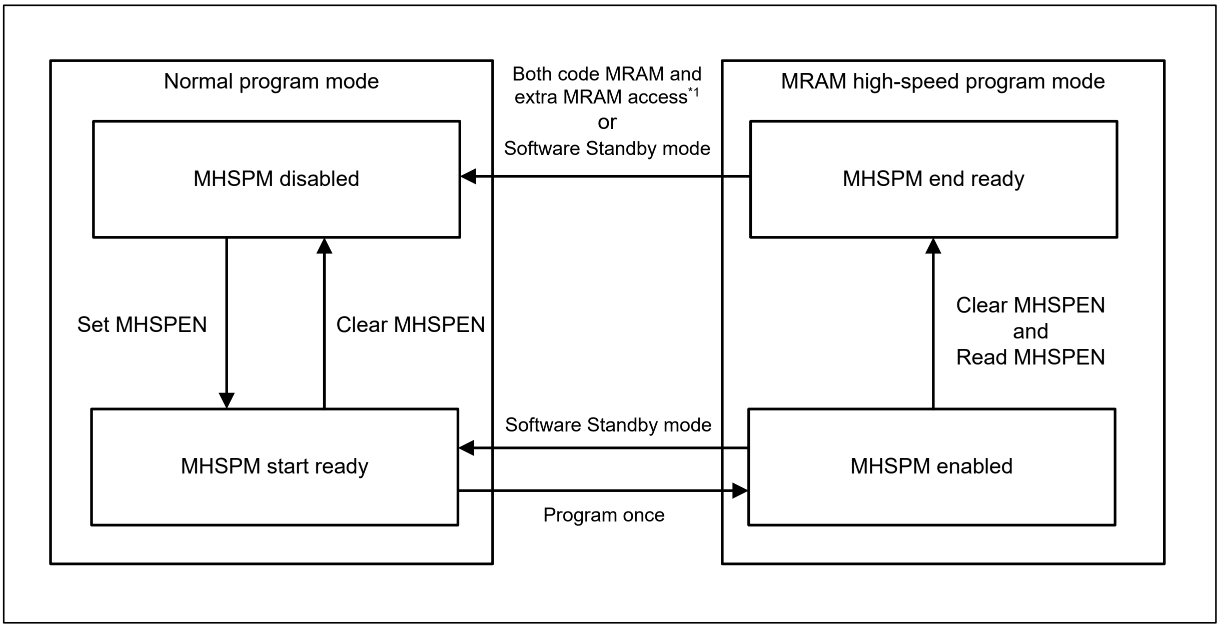
59.6.2 MRAM High-speed Program Mode

MRAM has normal program mode and high-speed program mode as shown in Figure 59.8, and the same mode setting is applied for both code MRAM program mode and extra program mode.

After setting the MHSPEN bit of the MRPSC register to 1 in MHSPM disabled status in normal program mode, the status changes to MHSPM start ready. When code MRAM or extra MRAM is programmed in MHSPM start ready status once, MRAM program mode changes to MHSPM enabled status in high-speed program mode.

To return to normal program mode from MHSPM enabled status, set the MHSPEN bit to 0, read the MHSPEN bit once, and transition into MHSPM end ready status at first. When both code MRAM and extra MRAM are accessed in MHSPM end ready status, MRAM program mode changes to MHSPM disabled status in normal program mode.

When the MCU transitions to Software Standby mode, the MRAM program mode changes to normal program mode while retaining the MHSPEN bit. If the MHSPEN bit is set to 1, programming the code MRAM or extra MRAM once after returning from Software Standby mode changes the MRAM program mode to high-speed program mode.



Note 1. Code MRAM access: Read, Program, W-HUK zeroization
Extra MRAM access: Read, Program, W-HUK zeroization, Forced stop

Figure 59.8 High-speed program mode transition

59.6.3 Debugger Support Functions

Precondition of functions

For non-secure debugging, the debugger can read and program to the Code MRAM area with non-secure attributes. Registers set to the secure attribute cannot be read and written by non-secure access. It is assumed that the MRCFREQ register and MREFREQ register are set to the appropriate values in user software when user software is running. For the debugger download, or break setting is enabled before user software startup, the chip startup frequency must be set to the initial register values of MRCFREQ and MREFREQ registers.

Program function

Code MRAM can be programmed by setting the NVMWE bit in the DBGNVMCR register to 1 without setting the MRCPC0 and MRCPC1 registers. However, it cannot be programmed in areas protected by block protection. If the block protection needs to be cleared, execute the Configuration program command in serial programing mode to do so. To avoid interfering with the operation of the debugger, when the DBGSTOPNVMER bit in the DBGSTOPCR register is set to 1, it masks the setting of the ECCERRC and PRGERRC bits in the MRCPS register, which is an error notification register during code MRAM programming. If the error bits were already set before the DBGSTOPNVMER bit is set, the value is held until the register is cleared.

Debugging by the debugger targets the code and data areas, so errors in the program caused by MACI commands other than in the data area of the extra MRAM are not masked.

Read function

The TEDERRC and DECERRC bits of the MRCRAES register and the TEDERRC and DECERRC bits of the MRERAES register are not set if an ECC error occurs during access from the debugger. DBGSTOPNVMER bit is not used for masking ECC error in reading from the debugger.

59.7 MACI Commands

59.7.1 Programmable Area with MACI Commands

Table 59.10 shows the MRAM and OTP areas in extra MRAM that can be programmed with the MACI command. These areas store setting data for security, safety, and system configuration. Each bit (both data and ECC) in the OTP area can only be programmed once from 1 to 0. There are two types of OTP areas with ECC and without ECC as shown in Table 59.10. A unit of 16-bytes data can be programmed only once to the OTP area with ECC. Reprogramming of different data cannot be retried because some bits of ECC may need to change from 0 to 1. Reliability of OTP area without ECC is secured by redundant data configuration. A unit of 16-bytes data can be programmed multiple times to OTP area without ECC when there is no bit changing from 0 to 1.

Table 59.10 Programmable area with MACI commands

Address (Offset address: BASE_MC)	Area name	Memory type	ECC
0x00C9_F020 to 0x00C9_F0AF	Configuration	MRAM	Use
0x00C9_F0C0 to 0x00C9_F7FF	Configuration	MRAM	Use
0x00E0_7400 to 0x00E0_74BF	Hash of OEM_ROOT_PKn (n = 0 to 3)	OTP	Use
0x00E0_7600 to 0x00E0_760F	FSBL setting	OTP	Use
0x00E0_7610 to 0x00E0_769F	Start address of measurement report Start address of Code Certificate	OTP	Use
0x00E0_76A0 to 0x00E0_76FF	General purpose OTP	OTP	Use
0x00E1_7700 to 0x00E1_777F	Permanent Block Protect Setting for Secure	OTP	Do not use
0x00E1_7780 to 0x00E1_77FF	Permanent Block Protect Setting	OTP	Do not use
0x00E1_7900 to 0x00E1_790F	OFS Permanent Lock Setting	OTP	Do not use
0x00E1_7910 to 0x00E1_791F	REVOKE	OTP	Do not use
0x00E1_7920 to 0x00E1_792F	Zeroization HUK in OTP Enable	OTP	Do not use
0x00E1_7930 to 0x00E1_793F	Anti-Rollback Counter Setting	OTP	Do not use
0x00E1_7950 to 0x00E1_79FF	Reserved	OTP	Do not use
—	Anti-Rollback Counter	OTP	Do not use

59.7.2 List of MACI Commands

Table 59.11 List of MACI commands

MACI command	Function
Program	This command is used to program extra MRAM area except for the configuration area. The program unit is 16 bytes.
Status Clear	This command initializes the ILGLERR, PRGERR, ILGCOMERR, SECERR, OTERR, TZFERR, and CFGPRGERR bits in the MSTATR register and the CMDLK and MREAE bits in the MASTAT register, including the extra MRAM sequencer released from command-locked state.
Forced Stop	This command forcibly stops processing of MACI commands and initializes the MSTATR and MASTAT register.
Configuration set	This command is used to program configuration area for setting security function and safety function Units of setting are 16 bytes.
Increment Counter (Secure access only)	The anti-rollback counter value can be incremented. The target counter is selected by MCNTSELR register. Update data unit is 1 bit.
Read Counter (Secure access only)	The anti-rollback counter value can be read. The target counter is selected by the MCNTSELR register. Reading data is output to MCNTDATAR0 and MCNTDATAR1 registers. Read unit is 64 bits.

The MACI commands are issued by writing to the MACI command-issuing area (see Table 59.4). When write access as shown in Table 59.12 proceeds in the specified state, the extra MRAM sequencer executes the processing corresponding to the given command (see section 59.7.3. Relationship Between the Extra MRAM Sequencer State and MACI Commands).

Table 59.12 MACI command formats

MACI commands	Number of write access	Write data to the MACI command-issuing area			
		1st access	2nd access	3rd to (N+2)th access	(N+3)th access
Program 16-byte programming: N = 8	N + 3	0xE8	0x08 (= N)	WD_1 to WD_N	0xD0
Status Clear	1	0x50	—	—	—
Forced Stop	1	0xB3	—	—	—
Configuration set N = 8 (16 Bytes)	N + 3	0x40	0x08 (= N)	WD_1 to WDn	0xD0
Increment Counter	2	0x35	0xD0	—	—
Read Counter	2	0x39	0xD0	—	—

Note: WDN (N = 1, 2,…): Nth 16-bit data to be programmed.

The extra MRAM sequencer clears the MSTATR.MRDY bit to 0 at the start of processing of a command other than the Status Clear command. The extra MRAM sequencer also sets this bit to 1 on completion of command processing.

If the setting of the MRDYIE.MRDYIE bit is 1, MRAM_ENDOFPE interrupt is generated when the MSTATR.MRDY bit is set to 1.

59.7.3 Relationship Between the Extra MRAM Sequencer State and MACI Commands

The MACI commands are accepted according to the mode/state of the extra MRAM sequencer. MACI commands should be issued after the transition to the extra MRAM program mode and checking the state of the extra MRAM sequencer. Use the MSTATR and MASTAT registers to check the state of the extra MRAM sequencer. In addition, the occurrence of errors in general can be checked by reading the CMDLK bit in the MASTAT register. Its value is the logical OR of the ILGLERR, ILGCOMERR, SECERR, OTERR, TZFERR, PRGERR, and CFGPRGERR bits in the MSTATR register. Table 59.13 lists the available commands in each operating mode.

Table 59.13 Operating mode and available commands

Operation mode	MENTRYR	Available commands
Extra MRAM read mode	0x0000	None
Extra MRAM program mode (W-HUK zeroization is not executed)	0x0080	Program Status Clear Forced Stop Configuration Set Increment Counter Read Counter
Extra MRAM program mode (W-HUK zeroization is executed)	0x0000/0x0080*1	None

Note 1. MRAM is in extra MRAM program mode when W-HUK zeroization is executed regardless of MENTRYR value. See section 59.6.1. MRAM Read/Program Mode.

Table 59.14 shows the state of the extra MRAM sequencer and acceptable MACI commands. An appropriate mode is assumed to have been set before the commands are executed.

Table 59.14 Acceptable MACI commands and the state of the extra MRAM sequencer

	Program, Configuration set, Increment Counter, Read Counter, Forced Stop command processing	Command-locked state (MRDY = 1)	Command-locked state (MRDY = 0)	W-HUK zeroization is executed	Other state
MRDY bit	0	1	0	0/1	1
CMDLK bit	0	1	1	0/1	0
Program	X	X	X	X	O
Status Clear	X	O	X	X	O
Forced Stop	O	O	O	X	O
Configuration set	X	X	X	X	O
Increment Counter	X	X	X	X	O
Read Counter	X	X	X	X	O

Remarks:

O: Acceptable

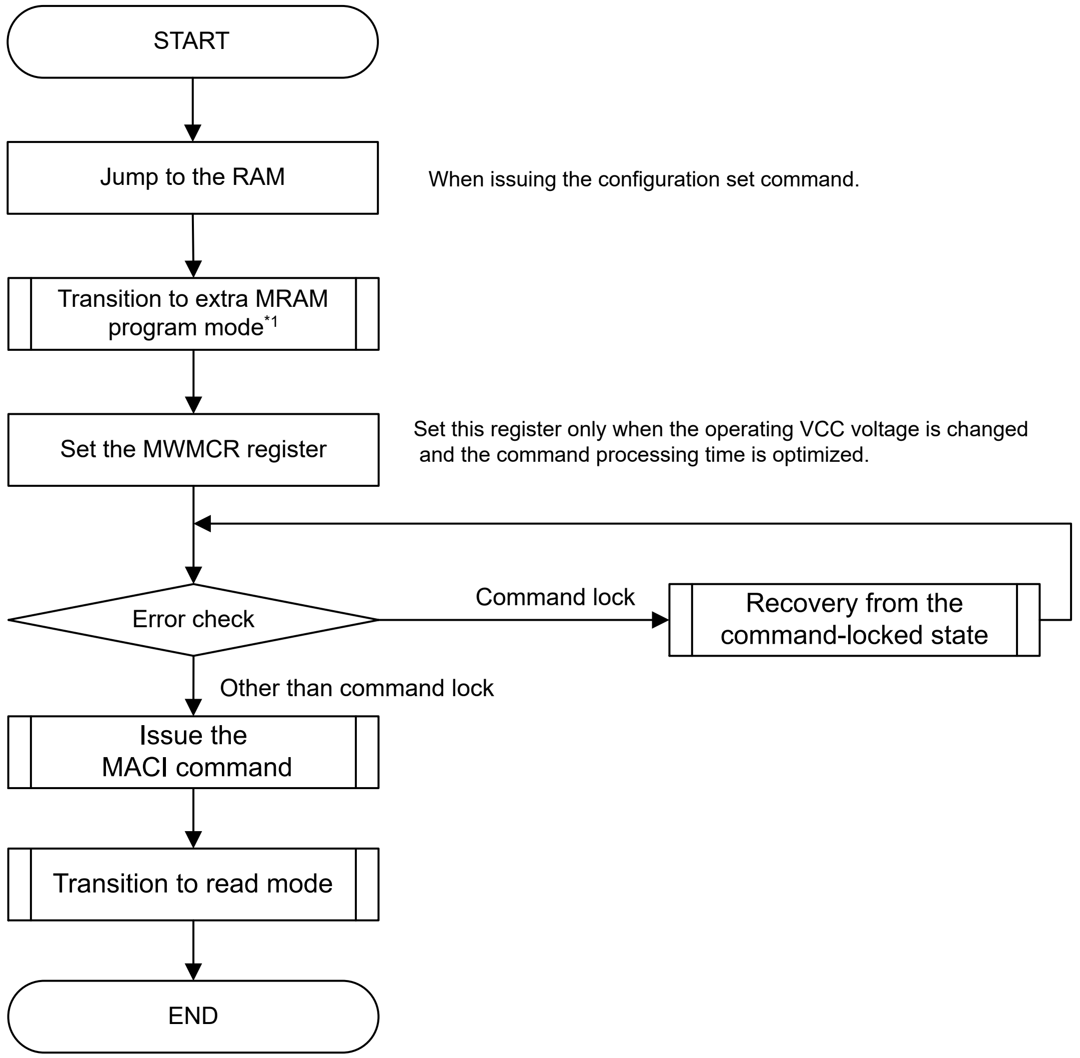
X: Not acceptable (places the sequencer in the command-locked state)

59.7.4 Usage of MACI Commands

This section gives an overview of the usage of MACI commands.

59.7.4.1 Overview of Command Usage in Extra MRAM Program Mode

Figure 59.9 shows an overview of MACI command usage in extra MRAM program mode. See Table 59.13 for the available commands in extra MRAM program mode.



Note 1. See section 59.7.4.2. Transition to Extra MRAM Program Mode.

Figure 59.9 Overview of command usage in extra MRAM program mode

59.7.4.2 Transition to Extra MRAM Program Mode

To issue the MACI commands for the extra MRAM, a transition to extra MRAM program mode is required. To cause the transition to extra MRAM program mode, set the MENTRY bit in the MENTRYR register to 1.

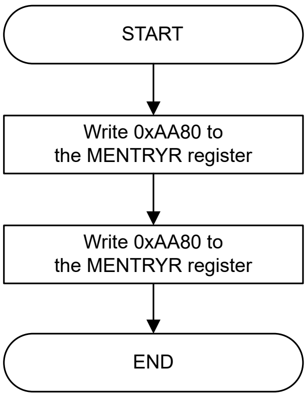
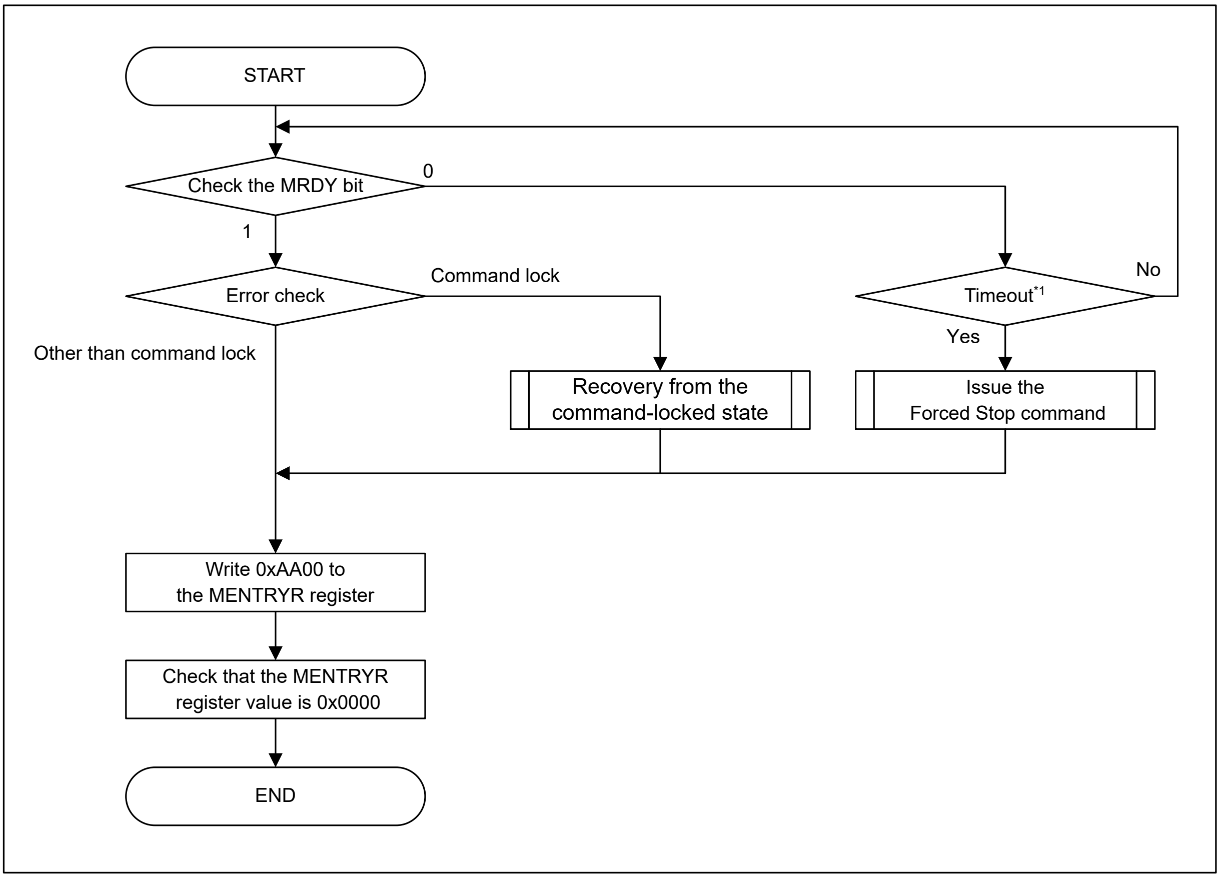


Figure 59.10 Procedure for transition to extra MRAM program mode

59.7.4.3 Transition to Read Mode

To read the extra MRAM, the MRDY bit in the MSTATR register must be set to 1 or a transition to read mode is required. To cause a transition to read mode, set the MENTRYR register to 0x0000. The transition to read mode should be made while processing by the extra MRAM sequencer is completed and the operation is in a state other than command-locked.



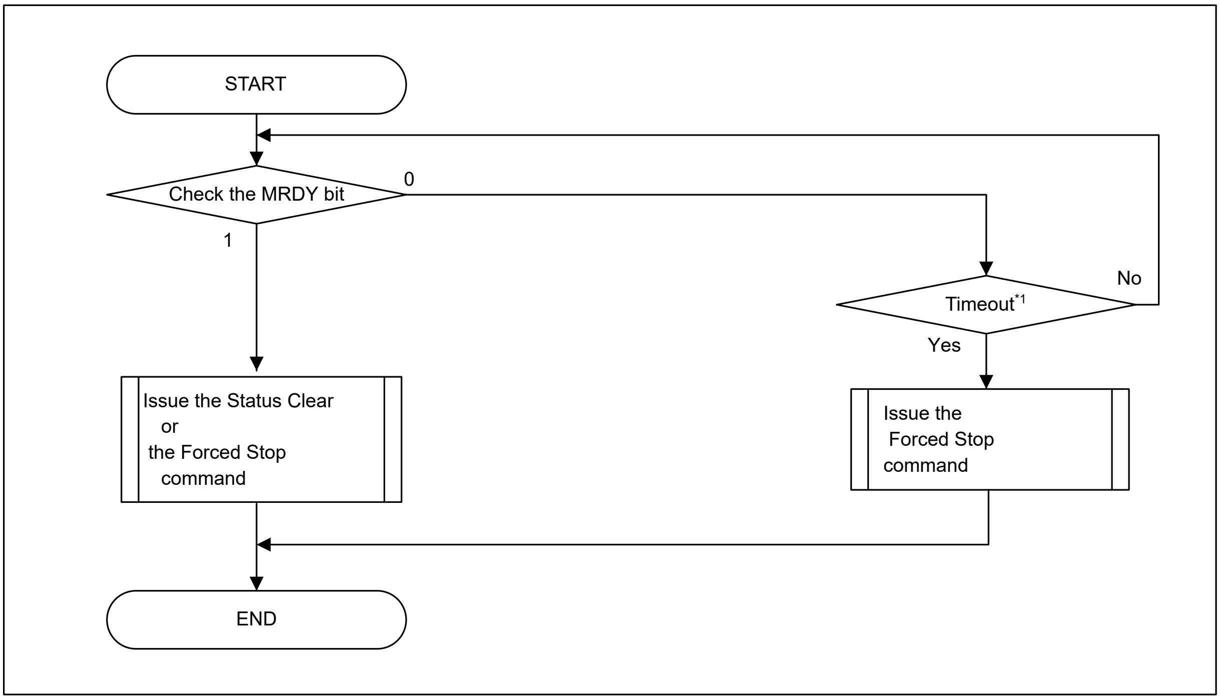
Note 1. Judgment of the timeout is based on 1.1 times the maximum time for processing of the MACI command that is in progress. For details of the updating time, see section 62, Electrical Characteristics.

Figure 59.11 Procedure for transition to read mode

59.7.4.4 Recovery from the Command-Locked State

When the extra MRAM sequencer enters the command-locked state, MACI commands cannot be accepted. To release the sequencer from the command-locked state, use the Status Clear command, Forced Stop command.

If processing is not completed in time beyond the maximum programming/erasure time specified in the electrical characteristics, this can be considered a timeout, and the extra MRAM sequencer should be stopped by the Forced Stop command.



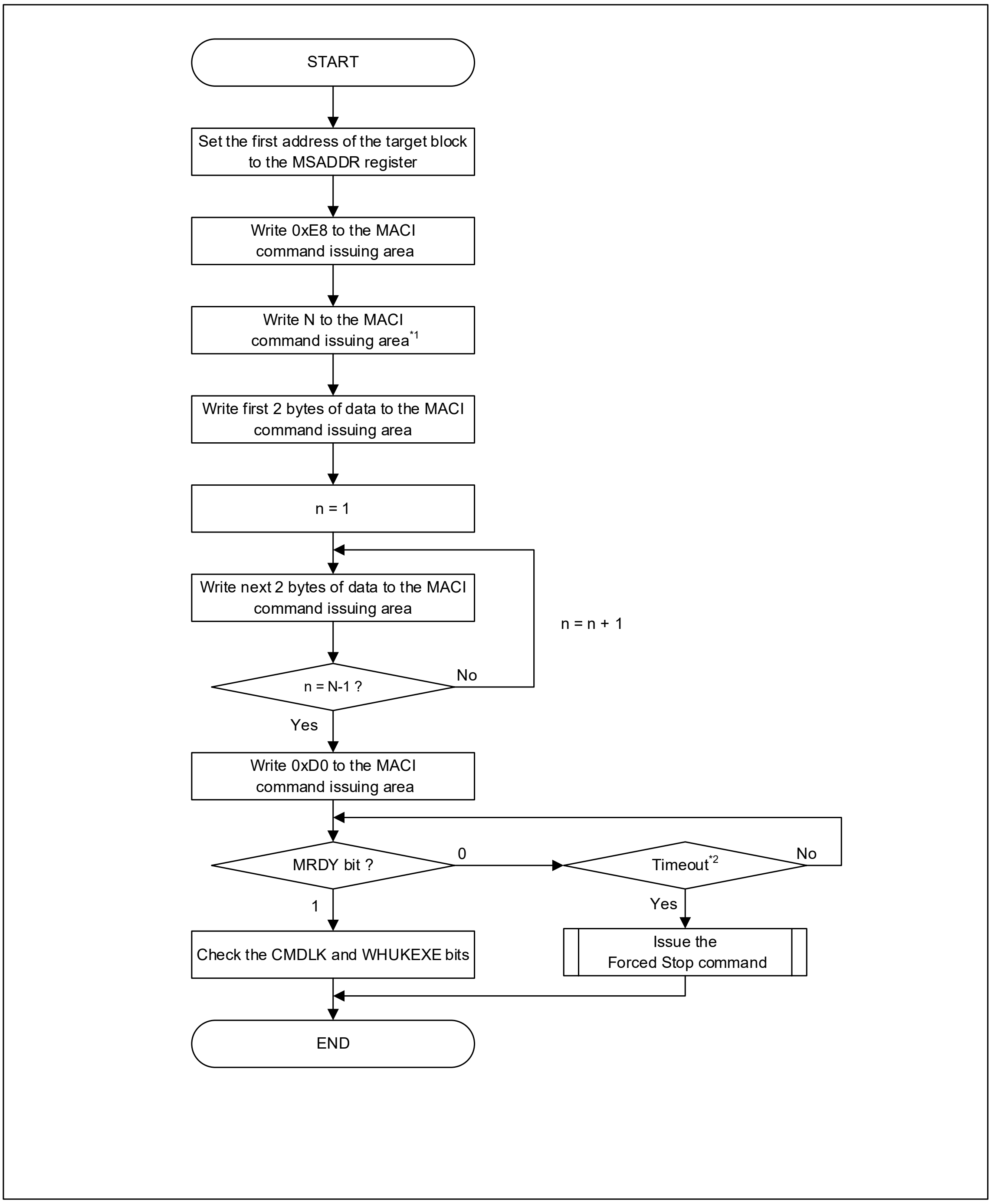
Note 1. Judgment of the timeout is based on 1.1 times the maximum time for processing of the MACI command that is in progress. For details of the updating time, see section 62, Electrical Characteristics.

Figure 59.12 Recovery from the command-locked state

59.7.4.5 Program Command

The Program command is used for programming to the Extra MRAM area, excluding the configuration area. Before issuing the Program command, set the first address of the target block in the MSADDR register.

Writing 0xD0 to the MACI command issuing area at the final access of the MACI command issuing starts the program command processing.



Note 1. N
0x08: 16-bytes programming

Note 2. Judgment of the timeout is based on 1.1 times the maximum time of programming. For details of the updating time, see section 62, Electrical Characteristics.

Figure 59.13 Usage of the Program command

The correspondence between the possible target data and the address value set in the MSADDR register is shown in Table 59.15. For details on the MSADDR register, see section 59.5.19. MSADDR : MACI Command Start Address Register.

Table 59.15 Address used by Program command

Address	MSADDR register value	Setting data	Timing when the setting is enabled
BASE_MC + 0x00E0_7600	BASE_MCS + 0x00E0_7600	FSBL setting	At a reset
BASE_MC + 0x00E0_7610	BASE_MCS + 0x00E0_7610 to BASE_MCS + 0x00E0_7690	Start address of measurement report Start address of code certificate	—
BASE_MC + 0x00E0_76A0	BASE_MCS + 0x00E0_76A0 to BASE_MCS + 0x00E0_76F0	General purpose OTP	—
BASE_MC + 0x00E1_7700	BASE_MCS + 0x00E1_7700 to BASE_MCS + 0x00E1_7770	Permanent Block Protect Setting for Secure (PBPS_SEC)*1	When a reset or command is executed
BASE_MC + 0x00E1_7780	BASE_MCN + 0x00E1_7780 to BASE_MCN + 0x00E1_77F0	Permanent Block Protect Setting (PBPS)*2	When a reset or command is executed
BASE_MC + 0x00E1_7900	BASE_MCS + 0x00E1_7900	Permanent OFS Protect Setting (POFSPS)	When a reset or command is executed
BASE_MC + 0x00E1_7910	BASE_MCS + 0x00E1_7910	REVOKE	When a reset or command is executed
BASE_MC + 0x00E1_7920	BASE_MCS + 0x00E1_7920	Zeroization HUK Enable	When a reset or command is executed
BASE_MC + 0x00E1_7930	BASE_MCS + 0x00E1_7930	Anti-rollback Counter Setting	—
BASE_MC + 0x00E1_7950	BASE_MCS + 0x00E1_7950 to BASE_MCS + 0x00E1_79F0	Reserved	—

Note 1. The PBPS_SEC[n] bit can be programmed to 0 when the BPS_SEC[4n], BPS_SEC[4n + 1], BPS_SEC[4n + 2], and BPS_SEC[4n + 3] bits are all 0.
Note 2. The PBPS[n] bit can be programmed to 0 when the BPS[4n], BPS[4n + 1], BPS[4n + 2], and BPS[4n + 3] bits are all 0.

59.7.4.6 Status Clear Command

The Status Clear command is used to clear the command-locked state (see section 59.7.4.4. Recovery from the Command-Locked State). To clear the ILGLERR, ILGCOMERR, SECERR, OTERR, PRGERR, TZFERR, and CFGPRGERR bits in the MSTATR register in the command-locked state, the Status Clear command is available.

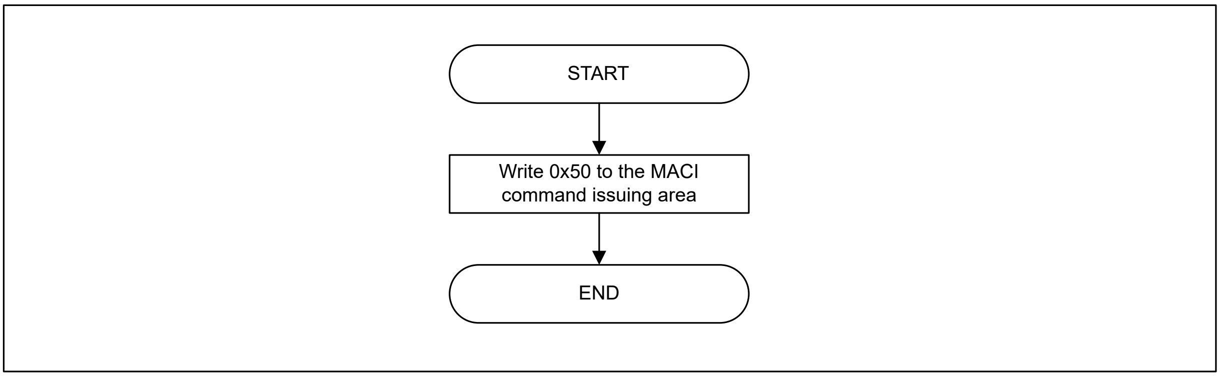
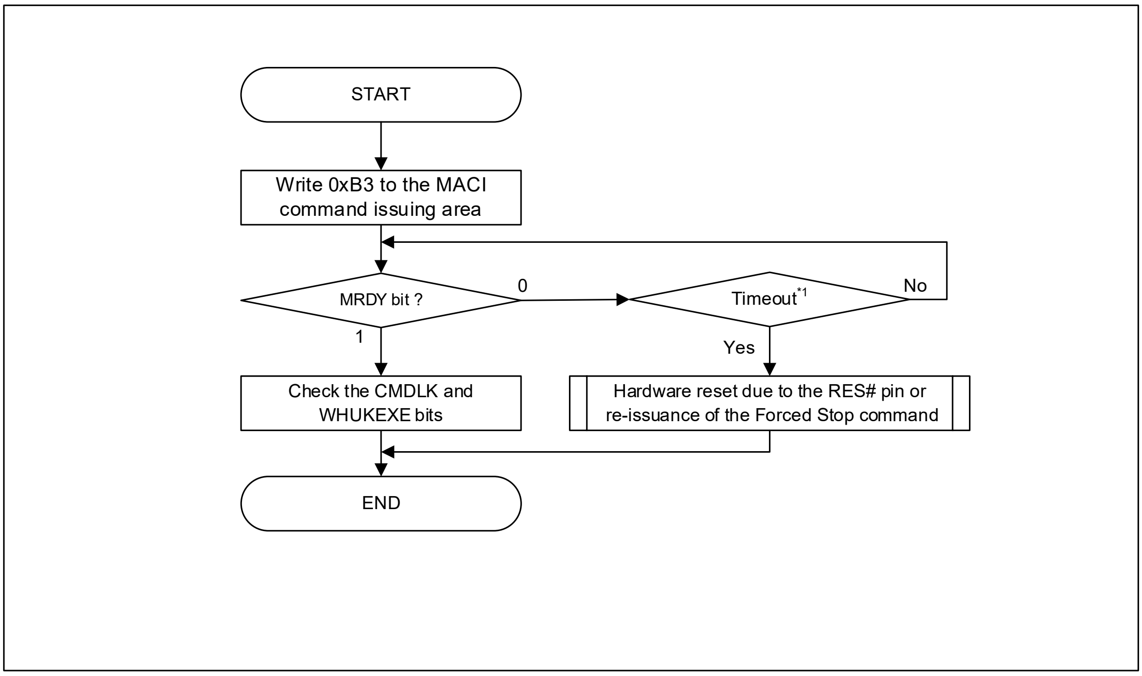


Figure 59.14 Usage of the Status Clear command

59.7.4.7 Forced Stop Command

The Forced Stop command forcibly ends command processing by the extra MRAM sequencer. Although this command halts command processing, values from the programming that was in progress are not guaranteed. Furthermore, resumption of processing is not possible. Processing of programming that was halted by the Forced Stop command is also defined as one from the perspective of programming endurance.

Executing the Forced Stop command also initializes part of the MACI, the whole extra MRAM sequencer, the MSTATR register, and the MASTAT register. Accordingly, this command can be used in the procedure for recovery from the command-locked state and in processing in response to a timeout of the extra MRAM sequencer (see section 59.7.4.4. Recovery from the Command-Locked State).

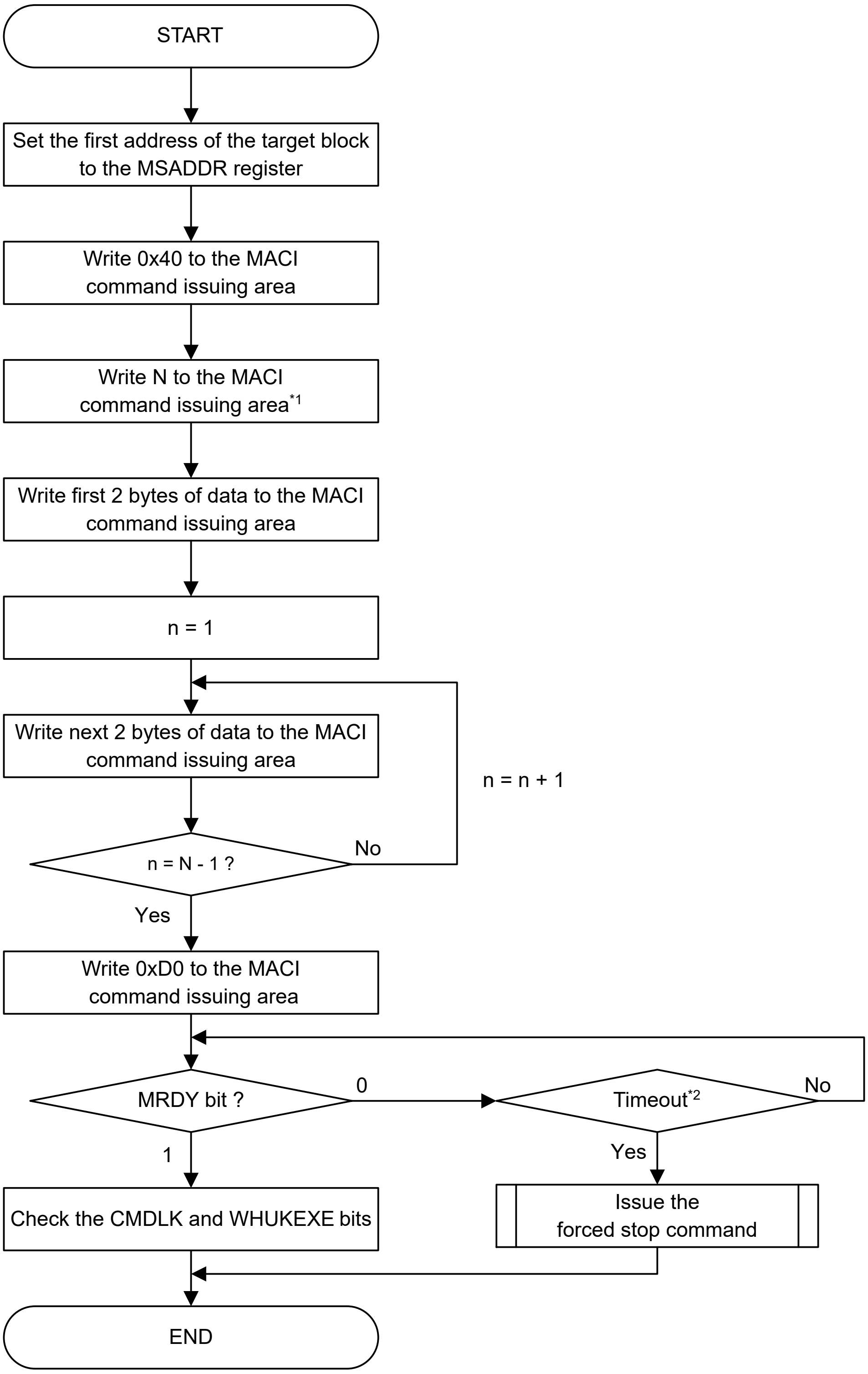


Note 1. Judgment of the timeout is based on 1.1 times the maximum time of the Forced Stop command latency. For details of the updating time, see section 62, Electrical Characteristics.

Figure 59.15 Usage of the Forced Stop command

59.7.4.8 Configuration Set Command

The Configuration Set command is used for security function and safety function. Before issuing the Configuration Set command, set the specified address (shown in Table 59.16) in the MSADDR register. Writing 0xD0 to the MACI command issuing area in the final access (for issuing the MACI command) starts MACI processing of the Configuration Set command.



Note 1. N
0x08: 16-bytes programming
Note 2. Judgment of the timeout is based on 1.1 times the maximum time of the command.
See the Electrical Characteristics of the User's Manual about those time.

Figure 59.16 Usage of the Configuration Set command

The correspondence between the possible target data for configuration setting and the address value set in the MSADDR register is shown in Table 59.16. For details on the MSADDR register, see section 59.5.19. MSADDR : MACI Command Start Address Register.

Table 59.16 Address used by the Configuration Set command

Address	MSADDR register value	Setting data	Operation of additional writing		Timing when the setting is enabled
			POFSPS.POFSPS [13] bit is 1	POFSPS.POFSPS [13] bit is 0	
BASE_MC + 0x00C9_F040	BASE_MCS + 0x00C9_F040	Option Function Select Register 0 (OFS0)	Writable*1	Writable*1	At a reset
BASE_MC + 0x00C9_F044	BASE_MCS + 0x00C9_F040	Option Function Select Register 2 (OFS2)	Writable*1	Writable*1	At a reset
BASE_MC + 0x00C9_F074	BASE_MCS + 0x00C9_F070	Startup Area Setting Register (SAS)	Writable*1	Not writable*1*2	When a reset or command is executed
BASE_MC + 0x00C9_F0C0	BASE_MCS + 0x00C9_F0C0	Option Function Select Register 1 Secure (OFS1_SEC)	Writable*1	Writable*1	At a reset
BASE_MC + 0x00C9_F0C4	BASE_MCS + 0x00C9_F0C0	Option Function Select Register 3 Secure (OFS3_SEC)	Writable*1	Writable*1	At a reset
BASE_MC + 0x00C9_F120	BASE_MCS + 0x00C9_F120	Option Function Select Register 1 Secure (OFS1_SEL)	Writable*1	Writable*1	At a reset
BASE_MC + 0x00C9_F124	BASE_MCS + 0x00C9_F120	Option Function Select Register 3 Secure (OFS3_SEL)	Writable*1	Writable*1	At a reset
BASE_MC + 0x00C9_F200	BASE_MCS + 0x00C9_F200 BASE_MCS + 0x00C9_F210 BASE_MCS + 0x00C9_F220 BASE_MCS + 0x00C9_F230	Block Protect Setting Register Secure (BPS_SEC)	Writable*3	Writable*3	When a reset or command is executed
BASE_MC + 0x00C9_F4C0	BASE_MCN + 0x00C9_F4C0	Option Function Select Register 1 (OFS1)	Writable	Writable	At a reset
BASE_MC + 0x00C9_F4C4	BASE_MCN + 0x00C9_F4C0	Option Function Select Register 3 (OFS3)	Writable	Writable	At a reset
BASE_MC + 0x00C9_F600	BASE_MCN + 0x00C9_F600 BASE_MCN + 0x00C9_F610 BASE_MCN + 0x00C9_F620 BASE_MCN + 0x00C9_F630	Block Protect Setting Register (BPS)	Writable*4	Writable*4	When a reset or command is executed

Note 1. This area can be protected by the POFSPS.POFSPS bit. If the POFSPS.POFSPS bit corresponding to the update area is 0, it cannot be updated by the Configuration Set command.

Note 2. The POFSPS.POFSPS[13] bit cannot be restored to 1 using the Program command once it is set to 0. Therefore, setting the startup area select flags and startup size select flags again becomes impossible. When the Configuration Set command is issued to the address of BASE_MCS + 0x00C9 F074, the command is locked. Exercise extra caution when handling the POFSPS.POFSPS[13] bit.

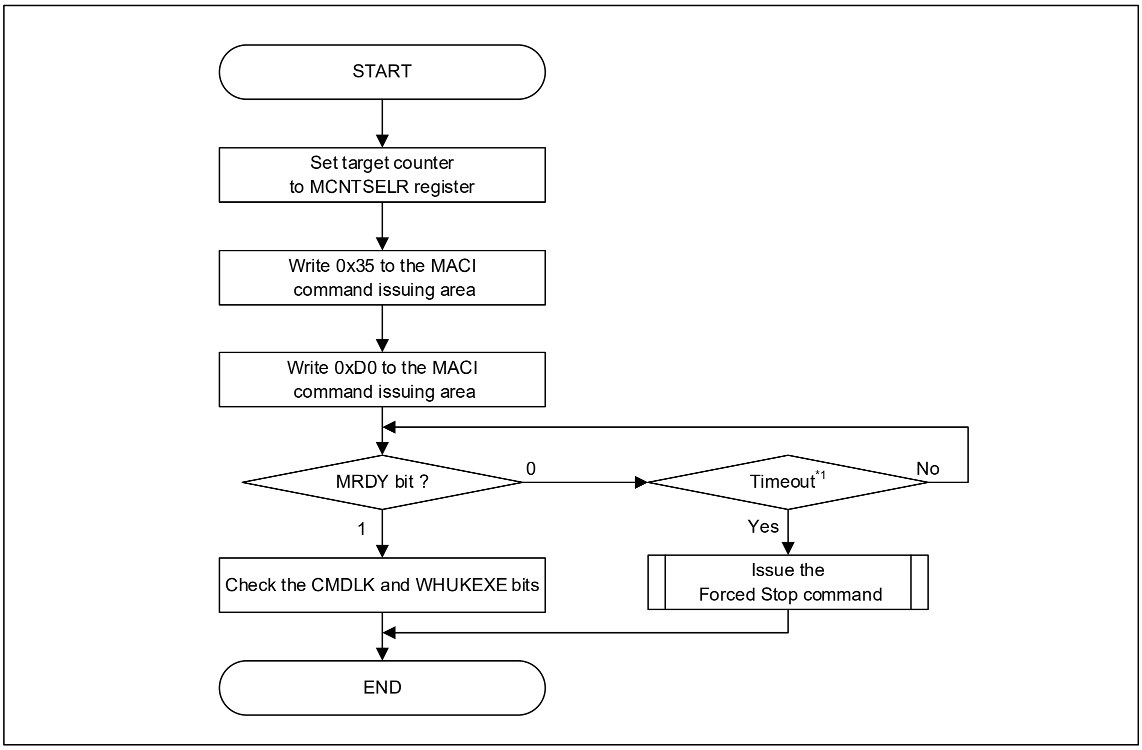
Note 3. Once the PBPS_SEC[n] bit is set to 0, the BPS_SEC[4n], BPS_SEC[4n + 1], BPS_SEC[4n + 2], and BPS_SEC[4n + 3] bits cannot be restored to 1 using the Configuration Set command.

Note 4. Once the PBPS[n] bit is set to 0, the BPS[4n], BPS[4n + 1], BPS[4n + 2], and BPS[4n + 3] bits cannot be restored to 1 using the Configuration Set command.

59.7.4.9 Increment Counter Command

The Increment Counter command is used for security function and safety function, and for updating three types of anti-rollback counter (ARC_SEC, ARC_NSEC, and ARC_OEMBL). The counter update is executed for increment processing and not for decrement processing.

Writing 0xD0 at the 2nd write access of the extra MRAM sequencer command triggers MACI to start the Increment Counter command processing. Completion of command processing can be confirmed by the MRDY bit of the MSTATR register.



Note 1. Judgment of the timeout is based on 1.1 times the maximum time of the command.
See the Electrical Characteristics of the User's Manual about those time.

Figure 59.17 Usage of the Increment Counter command

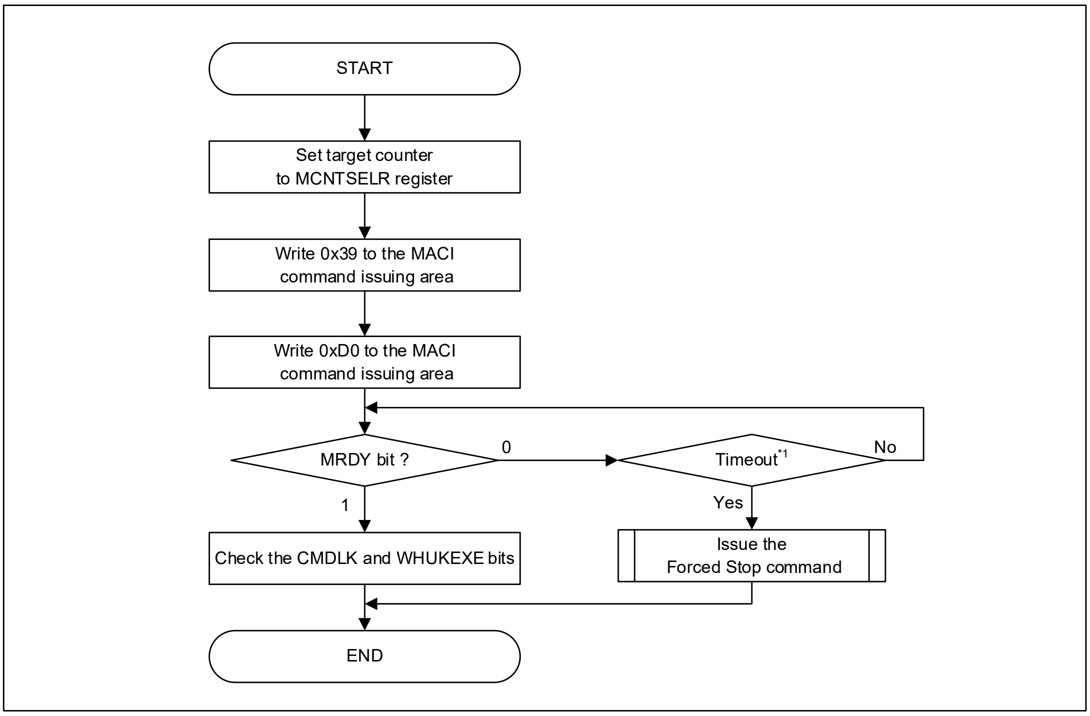
The anti-rollback counter value is not increased by the following conditions using the Increment Counter command.

- The target counter is protected by ARCSEC_LK, ARCNSEC_LK0 to 3, or ARCOEMBL_LK.
- The previous command is not the Read Counter command against ARC_OEMBL when the target counter is ARC_OEMBL.
- The update value of ARC_OEMBL is equal to or higher than the OEM boot loader version (OEM BL Ver.) output from the security engine.
- The target counter value is maximum (error does not occur).
- The target counter is ARC_NSEC when CNF_ARCNSEC0 to 1 is an invalid setting.

59.7.4.10 Read Counter Command

The Read Counter command is used for security function and safety function, and for reading three types of anti-rollback counter (ARC_SEC, ARC_NSEC, and ARC_OEMBL). The extra MRAM sequencer reads 64-bits data, and stores to MCNTDATAR0 and MCNTDATAR1 registers.

Writing 0xD0 at the 2nd write access of the extra MRAM sequencer command triggers MACI to start the Read Counter command processing. Completion of command processing can be confirmed by the MRDY bit of the MSTATR register.



Note 1. Judgment of the timeout is based on 1.1 times the maximum time of the command.
See the Electrical Characteristics of the User's Manual about those time.

Figure 59.18 Usage of the Read Counter command

59.8 Security Function

The extra MRAM sequencer supports the following security functions in a security-enabled product:

- Permanent OFS protect setting
- Security flag for startup area and startup size
- Permanent block protect setting
- MRAM protection for TrustZone
- Extra MRAM configuration area protection
- Anti-rollback counter
- HUK protection
- OEM_ROOT_KEY protection.

59.8.1 Permanent OFS Protect Setting

OFS area in the configuration area can be permanent protected by POFSPS bits in the OTP area as shown in Table 59.17.
See the User's Manual: Section xx, Option-Setting Memory.

Table 59.17 Protection condition of OFS area in configuration area (1 of 2)

Protect address	MSADDR register value	Protection size	Protect bit	Remarks
BASE_MC + 0x00C9_F040	BASE_MCS + 0x00C9_F040	4 Bytes	POFSPS[0]	—
BASE_MC + 0x00C9_F044	BASE_MCS + 0x00C9_F040	4 Bytes	POFSPS[1]	—

Table 59.17 Protection condition of OFS area in configuration area (2 of 2)

Protect address	MSADDR register value	Protection size	Protect bit	Remarks
BASE_MC + 0x00C9_F048	BASE_MCS + 0x00C9_F040	4 Bytes	POFSPS[2]	—
BASE_MC + 0x00C9_F04C	BASE_MCS + 0x00C9_F040	4 Bytes	POFSPS[3]	—
⋮	⋮	⋮	⋮	⋮
BASE_MC + 0x00C9_F06C	BASE_MCS + 0x00C9_F060	4 Bytes	POFSPS[11]	—
BASE_MC + 0x00C9_F070	BASE_MCS + 0x00C9_F070	4 Bytes	POFSPS[12], POFSPS[13]	When POFSPS[13] is 0, programming in the protect area with the program command generates SECERR.
BASE_MC + 0x00C9_F074	BASE_MCS + 0x00C9_F070	4 Bytes	POFSPS[13]	
BASE_MC + 0x00C9_F078	BASE_MCS + 0x00C9_F070	4 Bytes	POFSPS[13], POFSPS[14]	
BASE_MC + 0x00C9_F07C	BASE_MCS + 0x00C9_F070	4 Bytes	POFSPS[13], POFSPS[15]	
BASE_MC + 0x00C9_F080	BASE_MCS + 0x00C9_F080	4 Bytes	POFSPS[16]	—
⋮	⋮	⋮	⋮	⋮
BASE_MC + 0x00C9_F0B0	BASE_MCS + 0x00C9_F0B0	4 Bytes	POFSPS[28]	For the target address, even if 0 is written to the POFSPS bit, the protection is disabled.
BASE_MC + 0x00C9_F0B4	BASE_MCS + 0x00C9_F0B0	4 Bytes	POFSPS[29]	
BASE_MC + 0x00C9_F0B8	BASE_MCS + 0x00C9_F0B0	4 Bytes	POFSPS[30]	
BASE_MC + 0x00C9_F0BC	BASE_MCS + 0x00C9_F0B0	4 Bytes	POFSPS[31]	
BASE_MC + 0x00C9_F0C0	BASE_MCS + 0x00C9_F0C0	4 Bytes	POFSPS[32]	—
⋮	⋮	⋮	⋮	⋮
BASE_MC + 0x00C9_F170	BASE_MCS + 0x00C9_F170	4 Bytes	POFSPS[76]	—
BASE_MC + 0x00C9_F174	BASE_MCS + 0x00C9_F170	4 Bytes	POFSPS[77]	—
BASE_MC + 0x00C9_F178	BASE_MCS + 0x00C9_F170	4 Bytes	POFSPS[78]	—
BASE_MC + 0x00C9_F17C	BASE_MCS + 0x00C9_F170	4 Bytes	POFSPS[79]	—

59.8.2 Security Flag for Startup Area and Startup Size Select

The security flag (POFSPS.POFSPS[13]) for the startup area and startup size are located in the OTP area as the section 59.7.4.5. Program Command. The POFSPS[13] bit can be monitored as the FSPR bit in the MSUASMON register.

When the POFSPS[13] bit is 0, issuing the Configuration set command to change the BTFLG and BTSIZE bits causes the command-locked state. Also, when the POFSPS[13] bit is 0, it is not possible to write the startup area select bits (SAS[1:0]) in the MSUACR register. The POFSPS[13] bit provides the protection. See section 7, Option-Setting Memory.

59.8.3 Permanent Block Protect Setting

The permanent block protect setting is the clear protection for block protection setting. User area cannot be permanently updated by the MACI command when the permanent block protect setting is enabled. See section 59.9.1.2. Protection by Block Protect Setting in details of the block protect setting.

The block protect setting and the permanent block protect setting have the write/clear protection against the Configuration set command and Program command for each other. The extra MRAM sequencer does not detect an error when the Program command is issued to the write/clear protected settings.

Figure 59.19, Table 59.18, Table 59.19, Table 59.20, Table 59.21 and Table 59.22 show the write/clear protection against the block protect setting (BPS[4n]), (BPS[4n+1]), (BPS[4n+2]), (BPS[4n+3]) and the permanent protect setting (PBPS[n]). Figure 59.20, Table 59.23, Table 59.24, Table 59.25, Table 59.26 and Table 59.27 show the write/clear protection against the block protect setting for secure (BPS_SEC[4n]), (BPS_SEC[4n+1]), (BPS_SEC[4n+2]), (BPS_SEC[4n+3]) and permanent protect setting for secure (PBPS_SEC[n]).

Effective permanent block protect setting (PBPS or PBPS_SEC) depends on Code MRAM boundary setting.

For details of permanent block protect setting (PBPS or PBPS_SEC), see the User’s Manual: Section xx, Option-Setting Memory.

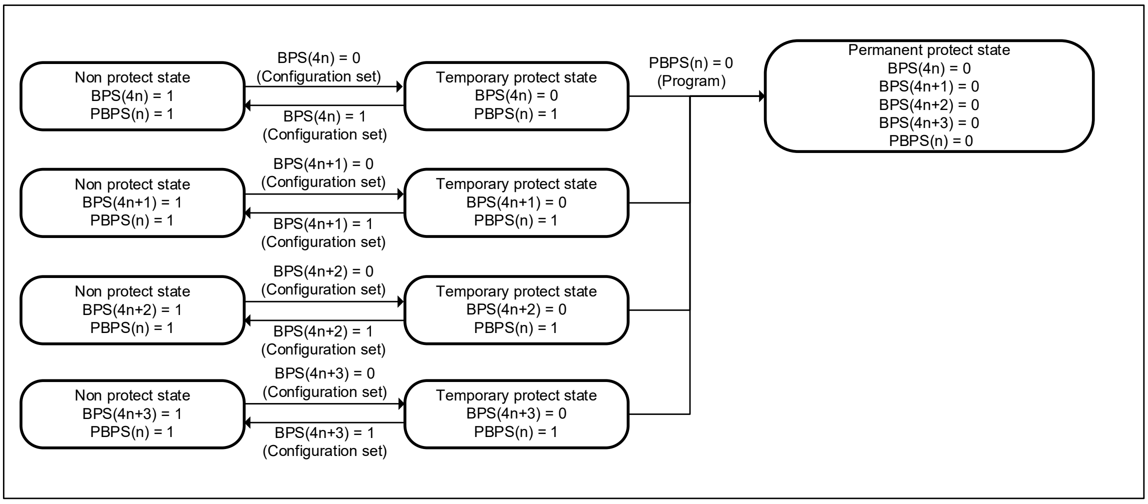


Figure 59.19 Status transition of extra MRAM sequencer by BPS[4n, 4n+1, 4n+2, 4n+3] and PBPS[n]

Table 59.18 Write/clear protection of BPS[4n] and PBPS[n]

Current state		Updatable state by configuration set command	
BPS[4n]	PBPS[n]	BPS[4n] = 1	BPS[4n] = 0
1	1	O	O
1	0	—	—
0	1	O	O
0	0	X	O

Table 59.19 Write/clear protection of BPS[4n+1] and PBPS[n]

Current state		Updatable state by configuration set command	
BPS[4n+1]	PBPS[n]	BPS[4n+1] = 1	BPS[4n+1] = 0
1	1	O	O
1	0	—	—
0	1	O	O
0	0	X	O

Table 59.20 Write/clear protection of BPS[4n+2] and PBPS[n]

Current state		Updatable state by configuration set command	
BPS[4n+2]	PBPS[n]	BPS[4n+2] = 1	BPS[4n+2] = 0
1	1	O	O
1	0	—	—
0	1	O	O
0	0	X	O

Table 59.21 Write/clear protection of BPS[4n+3] and PBPS[n] (1 of 2)

Current state		Updatable state by configuration set command	
BPS[4n+3]	PBPS[n]	BPS[4n+3] = 1	BPS[4n+3] = 0
1	1	O	O

Table 59.21 Write/clear protection of BPS[4n+3] and PBPS[n] (2 of 2)

Current state		Updatable state by configuration set command	
BPS[4n+3]	PBPS[n]	BPS[4n+3] = 1	BPS[4n+3] = 0
1	0	—	—
0	1	O	O
0	0	X	O

Remarks of Table 59.18, Table 59.19, Table 59.20 and Table 59.21:

O: Updatable by Configuration set command.

X: Not updatable by Configuration set command. (error does not occur.)

—: Not reach to this state.

Table 59.22 Write/clear protection of PBPS[n]

Current state					Updatable state by program command	
BPS[4n]	BPS[4n+1]	BPS[4n+2]	BPS[4n+3]	PBPS[n]	PBPS[n] = 1	PBPS[n] = 0
0	0	0	0	1	O	O
1	1 or 0	1 or 0	1 or 0	0	—	—
1 or 0	1	1 or 0	1 or 0	0	—	—
1 or 0	1 or 0	1	1 or 0	0	—	—
1 or 0	1 or 0	1 or 0	1	0	—	—
1	1 or 0	1 or 0	1 or 0	1	O	X
1 or 0	1	1 or 0	1 or 0	1	O	X
1 or 0	1 or 0	1	1 or 0	1	O	X
1 or 0	1 or 0	1 or 0	1	1	O	X
0	0	0	0	0	X	O

Remarks:

O: Updatable by Configuration set command.

X: Not updatable by Configuration set command. (error does not occur.)

-: Not reach to this state.

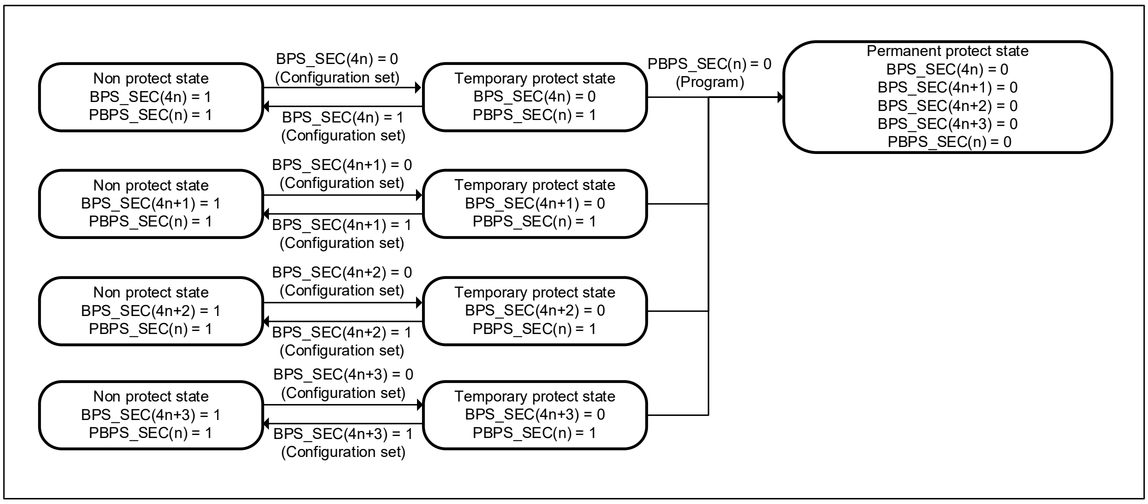


Figure 59.20 Status transition of extra MRAM sequencer by BPS_SEC[4n, 4n+1, 4n+2, 4n+3] and PBPS_SEC[n]

Table 59.23 Write/clear protection of BPS_SEC[4n] and PBPS_SEC[n]

Current state		Updatable state by configuration set command	
BPS_SEC[4n]	**PBPS_SEC[n]**	**BPS_SEC[4n] = 1**	**BPS_SEC[4n] = 0**
1	1	O	O
1	0	—	—
0	1	O	O
0	0	X	O

Table 59.24 Write/clear protection of BPS_SEC[4n+1] and PBPS_SEC[n]

Current state		Updatable state by configuration set command	
BPS_SEC[4n+1]	**PBPS_SEC[n]**	**BPS_SEC[4n+1] = 1**	**BPS_SEC[4n+1] = 0**
1	1	O	O
1	0	—	—
0	1	O	O
0	0	X	O

Table 59.25 Write/clear protection of BPS_SEC[4n+2] and PBPS_SEC[n]

Current state		Updatable state by configuration set command	
BPS_SEC[4n+2]	**PBPS_SEC[n]**	**BPS_SEC[4n+2] = 1**	**BPS_SEC[4n+2] = 0**
1	1	O	O
1	0	—	—
0	1	O	O
0	0	X	O

Table 59.26 Write/clear protection of BPS_SEC[4n+3] and PBPS_SEC[n]

Current state		Updatable state by configuration set command	
BPS_SEC[4n+3]	PBPS_SEC[n]	BPS_SEC[4n+3]=1	BPS_SEC[4n+3]=0
1	1	O	O
1	0	—	—
0	1	O	O
0	0	X	O

Remarks of Table 59.23, Table 59.24, Table 59.25 and Table 59.26:

O: Updatable by Configuration set command.

X: Not updatable by Configuration set command. (error does not occur.)

—: Not reach to this state.

Table 59.27 Write/clear protection of PBPS_SEC[n]

Current state					Updatable state by program command	
BPS_SEC [4n]	BPS_SEC [4n+1]	BPS_SEC [4n+2]	BPS_SEC [4n+3]	PBPS_SEC [n]	PBPS_SEC[n] = 1	PBPS_SEC[n] = 0
0	0	0	0	1	O	O
1	1 or 0	1 or 0	1 or 0	0	—	—
1 or 0	1	1 or 0	1 or 0	0	—	—
1 or 0	1 or 0	1	1 or 0	0	—	—
1 or 0	1 or 0	1 or 0	1	0	—	—
1	1 or 0	1 or 0	1 or 0	1	O	X
1 or 0	1	1 or 0	1 or 0	1	O	X
1 or 0	1 or 0	1	1 or 0	1	O	X
1 or 0	1 or 0	1 or 0	1	1	O	X
0	0	0	0	0	X	O

Remarks:

O: Updatable by Configuration set command.

X: Not updatable by Configuration set command. (error does not occur.)

—: Not reach to this state.

59.8.4 MRAM Protection for TrustZone

The MRAM protection for TrustZone is only available for the security-enabled product.

MRAM has the following four types of protect function against non-secure access to support the TrustZone specification for MRAM. Descriptions of this section are focused on extra MRAM sequencer operation.

- Protection for code MRAM area (program)
- Protection for extra MRAM area (program)
- Protection for MRAM area (read)
- Protection for register

59.8.4.1 Protection for Code MRAM Area (Program)

This function protects code MRAM from bus access such as CPU, DMAC. For details of protection, see section 51, Security Features.

59.8.4.2 Protection for Extra MRAM Area (Program)

This function protects secure area of extra MRAM from MRAM sequencer command of non-secure developer. The condition of protection depends on extra MRAM sequencer command, access attribution, and product specification.

See Table 59.28 for details of protection for extra MRAM area (program).

Extra MRAM area only allows privileged access.

Table 59.28 Protection for extra MRAM area (program)

MACI command		Target area		Issuing MACI command by non-secure access	Issuing MACI command by secure access
Program	Extra MRAM	MRAM and OTP area (Non-secure area)	Secure alias*1	X	X
			Non-secure alias*2	O	O
		MRAM and OTP area (Secure area)	Secure alias*1	X	O
			Non-secure alias*2	X	X
Configuration Set	Extra MRAM	Configuration area (Non-secure area)	Secure alias*1	X	X
			Non-secure alias*2	O	O
		Configuration area (Secure area)	Secure alias*1	X	O
			Non-secure alias*2	X	X
Increment Counter Read Counter	Extra MRAM	Anti-rollback counter area (Secure area only)	—	X	O

Note 1. The secure alias is determined when the address of MSADDR[28] is 0.
Note 2. The non-secure alias is determined when the address of MSADDR[28] is 1.

Remarks:

O: MACI command operation is not prohibited.

X: MACI command operation is prohibited.

Error occurs when the area is selected, and MACI command is executed.

The Configuration area has two types of secure region mapping for security-enabled product and security-disabled product. See Figure 59.21 for details of non-secure/secure region of configuration area.

The extra MRAM sequencer determines that the target area is a secure region from the MSADDR register setting and product information.

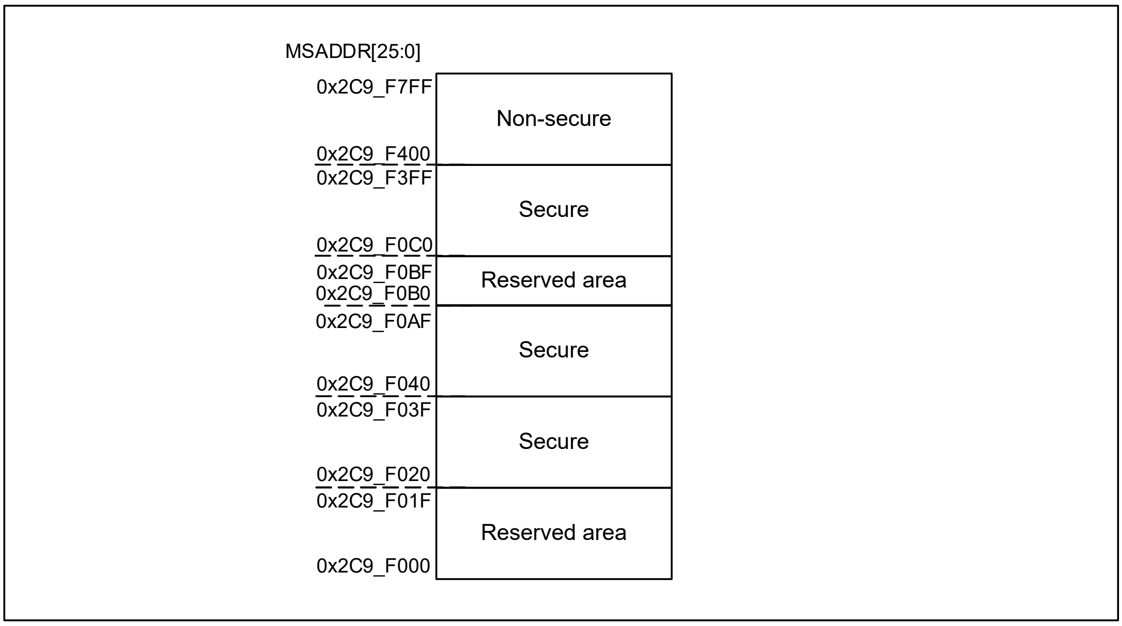


Figure 59.21 Secure/non-secure region in configuration area

59.8.4.3 Protection for MRAM Area (Read)

This function protects code MRAM and extra MRAM from bus access (such as CPU, DMAC). For details of protection, see the User's Manual: Section xx, Security Features.

59.8.4.4 Protection for Register

The MRAM registers have the write access protection against the non-secure access. See Table 59.29 for details of the protected MRAM registers. MRAM registers can be accessed with privileges.

Table 59.29 Protected MRAM registers for TrustZone (1 of 2)

Protection target registers	Secure attribute setting	Remarks
MRCEECC	SA register setting (MSAR.MRBCPSA)	See section 59.5.40. MRCEECC : Code MRAM ECC Encoder Control Register. For details of MSAR register, see section 59.5.13. MSAR : MRAM Security Attribution Register.
MRPSC and MREEECC	SA register setting (MSAR.MRBEPSA)	See section 59.5.31. MRPSC : MRAM Program Speed Control Register and MREEECC : Extra MRAM ECC Encoder Control Register. For details of MSAR register, see section 59.5.13. MSAR : MRAM Security Attribution Register.
MRCPC0, MRCBPROT0, MRCPS, MRCPEAINT, and MRCPEA	SA register setting (MSAR.MRCPSA)	See section 59.5.32. MRCPC0 : Code MRAM Program Control Register, section 59.5.34. MRCBPROT0 : Code MRAM Block Protection Register, section 59.5.36. MRCPS : Code MRAM Program Status Register, section 59.5.37. MRCPAEINT : Code MRAM Program Access Error Interrupt Enable Register and section 59.5.38. MRCPEA : Code MRAM Program Error Address Register. For details of MSAR register, see section 59.5.13. MSAR : MRAM Security Attribution Register.
MSAR	Always Secure	See section 59.5.13. MSAR : MRAM Security Attribution Register.

Table 59.29 Protected MRAM registers for TrustZone (2 of 2)

Protection target registers	Secure attribute setting	Remarks
MCTRCNTR, MCTRLSR, and MCTRSTATR	SA register setting (MSAR.MACITRSA)	See section 59.5.22. MCTRCNTR : MRAM Configuration Update Transfer Control Register, section 59.5.23. MCTRLSR : MRAM Configuration Update Transfer List Select Register and section 59.5.24. MCTRSTATR : MRAM Configuration Update Transfer Status Register. For details of MSAR register, see section 59.5.13. MSAR : MRAM Security Attribution Register.
MRAM access area	SA register setting (MSAR.MACICOMISA)	For details of MSAR register, see section 59.5.13. MSAR : MRAM Security Attribution Register.
MACI*1	SA register setting (MSAR.MACICOMRSA)	For details of MSAR register, see section 59.5.13. MSAR : MRAM Security Attribution Register.
MRCPFB	SA register setting (MSAR.MPFBENSA)	See section 59.5.1. MRCPFB : Code MRAM Prefetch Buffer Enable Register. For details of MSAR register, see section 59.5.13. MSAR : MRAM Security Attribution Register.
MRCFREQ	SA register setting (MSAR.MRCFREQSA)	See section 59.5.2. MRCFREQ : Code MRAM Frequency Notifications Register. For details of MSAR register, see section 59.5.13. MSAR : MRAM Security Attribution Register.
MRCDECC, MRCRAEINT, MRCRAES, MRCRTEA, and MRCRDEA	SA register setting (MSAR.MRCECCSA)	See section 59.5.4. MRCDECC : Code MRAM ECC Decoder Control Register, section 59.5.5. MRCRAEINT : Code MRAM Read Access Error Interrupt Enable Register, section 59.5.6. MRCRAES : Code MRAM Read Access Error Status Register, section 59.5.7. MRCRTEA : Code MRAM TED Error Address Register, and section 59.5.8. MRCRDEA : Code MRAM DEC Error Address Register. For details of MSAR register, see section 59.5.13. MSAR : MRAM Security Attribution Register.
MREFREQ	SA register setting (MSAR.MREFREQSA)	See section 59.5.3. MREFREQ : Extra MRAM Frequency Notifications Register. For details of MSAR register, see section 59.5.13. MSAR : MRAM Security Attribution Register.
MREDECC, MRERAEINT, MRERAES, MRERTEA, and MRERDEA	SA register setting (MSAR.MREECCSA)	See MREDECC : Extra MRAM ECC Decoder Control Register, section 59.5.9. MRERAEINT : Extra MRAM Read Access Error Interrupt Enable Register, section 59.5.10. MRERAES : Extra MRAM Read Access Error Status Register, section 59.5.11. MRERTEA : Extra MRAM TED Error Address Register, and section 59.5.12. MRERDEA : Extra MRAM DEC Error Address Register. For details of MSAR register, see section 59.5.13. MSAR : MRAM Security Attribution Register.
MREZS, MREZC, MCNTSELR, MCNTDATAR0 to 1, MRCPC1, MRCBPROT1, and MSUACR	Always Secure	See section 59.5.14. MREZS : Extra MRAM Zeroization Status Register, section 59.5.15. MREZC : Extra MRAM Zeroization Control Register, section 59.5.20. MCNTSELR : MRAM Counter Select Register, section 59.5.21. MCNTDTRn : MRAM Counter Data Register (n = 0, 1), section 59.5.33. MRCPC1 : Code MRAM Program Control Register for Secure, section 59.5.35. MRCBPROT1 : Code MRAM Block Protection Register for Secure, and section 59.5.30. MSUACR : MRAM Startup Area Control Register.

Note 1. The target MACI registers are MASTAT, MPAEINT, MRDYIE, MSADDR, MSTATR, MENTRYR, MSUINITR, MCMDR, and MSUASMON.

59.8.5 Anti-Rollback Counter

The anti-rollback counter cannot be decremented and is used in the firmware update. Only secure applications are available. Anti-rollback counter shifts one bit to the left with the Increment Counter command from the initial value of 0. The least significant bit (bit 0) is set to 1. When the anti-rollback counter is of maximum value, the extra MRAM sequencer does not change the counter by the Increment Counter command (error does not occur).

There are three commands that control the counters. See section 59.7.4.9. Increment Counter Command and section 59.7.4.10. Read Counter Command for details.

Figure 59.22 shows anti-rollback counter increment image when increment counter command is issued.

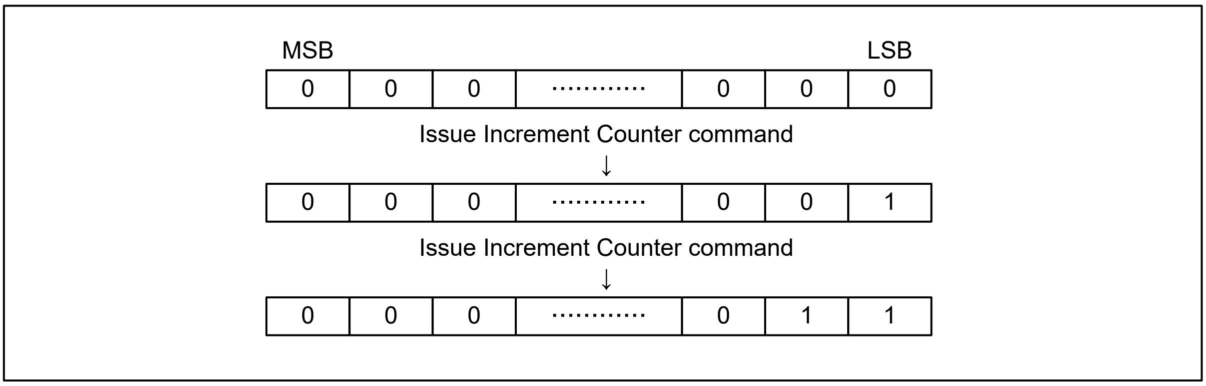


Figure 59.22 Increment image of anti-rollback counter

Anti-rollback counter has three types of counters (ARC_SEC, ARC_NSEC, and ARC_OEMBL) and lock bit (ARCSEC_LK, ARCNSEC_LK0 to 3, and ARCOEMBL_LK) for protection of each counter.

ARC_NSEC can be configured as 1 × 256 or 4 × 64 bits by the configuration area setting (CNF_ARCNS1 and CNF_ARCNS0) and cannot be updated in invalid setting.

ARC_OEMBL is protected by the security engine and cannot be updated until the update validation of the OEM boot loader is successful. After successful validation, ARC_OEMBL can be updated to the current OEM boot loader version (OEM BL Ver.) output by the security engine. ARC_OEMBL cannot be updated to a value that exceeds the OEM boot loader version (OEM BL Ver.).

59.8.6 HUK Protection

Zeroization of W-HUK is triggered when one of ZHUKEN[3:0] bits (zeroization enable bit) is 0 and anti-tamper is detected or 0x07 is set to WHUKWE[2:0] bits in MREZC register. Zeroing W-HUK disables the generation of HUK. Table 59.30 shows the combinations of enable bits and request signals. If a zeroing request is received during code MRAM or extra MRAM program, the executing program is forcibly terminated, and the zeroing process is started. If a code MRAM program request is received during zeroing, the code MRAM program is started after zeroing is completed. MACI commands are not accepted during zeroing. During W-HUK zeroing, all MACI commands are not accepted and extra MRAM cannot be read.

Table 59.30 Combinations of request signals and enable bits (1 of 2)

Zeroization request signal				Zeroization enable bit				Function
WHUKZE [2:0]	RTCIC			ZHUKEN				
	2	1	0	3	2	1	0	
0x7	—	—	—	0	—	—	—	O
0x7	—	—	—	1	—	—	—	Oe
Except 0x7	—	—	—	—	—	—	—	X
—	1	—	—	—	0	—	—	O
—	1	—	—	—	1	—	—	Oe
—	0	—	—	—	—	—	—	X
—	—	1	—	—	—	0	—	O
—	—	1	—	—	—	1	—	Oe
—	—	0	—	—	—	—	—	X
—	—	—	1	—	—	—	0	O
—	—	—	1	—	—	—	1	Oe

Table 59.30 Combinations of request signals and enable bits (2 of 2)

Zeroization request signal				Zeroization enable bit				Function
WHUKZE [2:0]	RTCIC			ZHUKEN				
	2	1	0	3	2	1	0	
—	—	—	0	—	—	—	—	X

Remarks:

O: W-HUK zeroing

Oe: W-HUK Pseudo-zeroing

X: Zeroization of W-HUK does not execute

—: Don't care

Table 59.31 shows differences of W-HUK pseudo-zeroing and W-HUK zeroing described in Table 59.30. If the zeroization enable bit is 1 and tamper is detected, the MREZS.WHUKZF bit is set to 0 to complete zeroing without physically zeroing W-HUK (W-HUK pseudo-zeroing).

Pseudo means the zeroization tamper event has happened, but the W-HUK zeroization response/action was not enabled and therefore not physically executed. The flag of zeroized W-HUK in the MREZS register is set, but because it is a pseudo tamper, it is restored at the next power cycle. W-HUK usage is disabled until the next power cycle. If the zeroization enable bit is 0 and a tamper is detected, physical zeroization of W-HUK is performed and the Zeroized Monitor register is set to 0 (W-HUK zeroing). When the zeroization function is enabled and after a tamper or software trigger, it is permanent (not pseudo). The flag of zeroized W-HUK in the MREZS register is cleared (set to 0 and remain at 0 permanently). See section 12.1.7. Tamper Detection Function for details of the tamper detection.

Table 59.31 Internal state at the time of zeroing request

Table 59.30 function	Operation name	Zeroization enable bits	W-HUK	Zeroized monitor register (MREZS)
Oe	W-HUK Pseudo-zeroing	Disable(1)	Not zeroed	Clear
O	W-HUK zeroing	Enable(0)	Zeroed	Clear flag after zeroization finished

59.8.7 OEM ROOT KEY Protection

OEM_ROOT_PKn can be revoked by the REVOKE setting in Option-Setting Memory. See section 51, Security Features for details of how OEM_ROOT_PKs are used when updating OEM_BL.

59.9 Protection Function

59.9.1 Software Protection

Software protection function disables MRAM programming operation according to register and configuration area settings.

59.9.1.1 Code MRAM Program Protection

When the MRCPC0 register is set to 0x0000, programming of non-secure aliases in code MRAM is prohibited. When the MRCPC1 register is set to 0x0000, programming of secure aliases in code MRAM is prohibited. Code MRAM program protection has a higher priority than block protection and cannot be canceled by MRCBPROT0 or MRCBPROT1 register.

59.9.1.2 Protection by Block Protect Setting

Each block in the user area has the block protect setting (BPS or BPS_SEC). When the MRCBPROT0 or MRCBPROT1 register is 0x0000 and the block protect bit is 0, program access to the non-secure user area of code MRAM is prohibited. When the MRCBPROT1 register is 0x0000 and the block protect bit is 0, program access to the secure user area of code MRAM is prohibited. To program the sector whose block protect bit is 0, set MRCBPROT0 or MRCBPROT1 register to 0x0001.

The block protect setting can be locked by the permanent block protect setting (PBPS or PBPS_SEC). When the permanent block protect setting and the block protect setting are 0, program access cannot be performed.

Effective block protect setting (BPS or BPS_SEC) and permanent block protect setting (PBPS or PBPS_SEC) depends on code MRAM boundary setting.

See section 59.8.3. Permanent Block Protect Setting for details of the permanent block protect setting and block protect setting.

See section 59.5.34. MRCBPROT0 : Code MRAM Block Protection Register and section 59.5.35. MRCBPROT1 : Code MRAM Block Protection Register for Secure details of the MRCBPROT0 and MRCBPROT1 registers.

For details of block protect setting (BPS or BPS_SEC), permanent block protect setting (PBPS or PBPS_SEC), and code MRAM boundary setting, see the User’s Manual: Section xx, Option-Setting Memory.

Table 59.32 to Table 59.35 show the relation of user area and the block protect setting in each function setting.

- BPS[0] to BPS[n] or BPS_SEC[0] to BPS_SEC[n] are assigned to the block of user area (for example, address is BASE_MC + 0x0000 0000 to the last block address).
- BPS[0]/BPS_SEC[0] and BPS[1]/BPS_SEC[1] are assigned to the block of user area depending on the startup area select setting (SAS.BTFLG bit) and startup area size select setting (SAS.BTSIZE bits). See section 59.9.3. Startup Program Protection for details.

Table 59.32 to Table 59.35 show example of the block protect setting.

Table 59.32 Example of block protect setting when SAS.BTFLG is 1 and all sizes

Address (offset address: BASE_MC)	Block size	Block protect setting	User area block number	Remarks
The last block address	2 KB	BPS[n] or BPS_SEC[n]	Block n	
⋮	⋮	⋮	⋮	
0x0000 3800 to 0x0000 3FFF	2 KB	BPS[7] or BPS_SEC[7]	Block 7	
0x0000 3000 to 0x0000 37FF	2 KB	BPS[6] or BPS_SEC[6]	Block 6	
0x0000 2800 to 0x0000 2FFF	2 KB	BPS[5] or BPS_SEC[5]	Block 5	
0x0000 2000 to 0x0000 27FF	2 KB	BPS[4] or BPS_SEC[4]	Block 4	
0x0000 1800 to 0x0000 1FFF	2 KB	BPS[3] or BPS_SEC[3]	Block 3	
0x0000 1000 to 0x0000 17FF	2 KB	BPS[2] or BPS_SEC[2]	Block 2	
0x0000 0800 to 0x0000 0FFF	2 KB	BPS[1] or BPS_SEC[1]	Block 1	
0x0000 0000 to 0x0000 07FF	2 KB	BPS[0] or BPS_SEC[0]	Block 0	

Table 59.33 Example of block protect setting when SAS.BTFLG is 0 and SAS.BTSIZE is 11 (1 of 2)

Address (offset address: BASE_MC)	Block size	Block protect setting	User area block number	Remarks
The last block address	2 KB	BPS[n] or BPS_SEC[n]	Block n	—
⋮	⋮	⋮	⋮	—
0x0000 4000 to 0x0000 47FF	2 KB	BPS[8] or BPS_SEC[8]	Block 8	—
0x0000 3800 to 0x0000 3FFF	2 KB	BPS[7] or BPS_SEC[7]	Block 3	Swap block 3 and block 7 in this startup area select setting
0x0000 3000 to 0x0000 37FF	2 KB	BPS[6] or BPS_SEC[6]	Block 2	Swap block 2 and block 6 in this startup area select setting
0x0000 2800 to 0x0000 2FFF	2 KB	BPS[5] or BPS_SEC[5]	Block 1	Swap block 1 and block 5 in this startup area select setting
0x0000 2000 to 0x0000 27FF	2 KB	BPS[4] or BPS_SEC[4]	Block 0	Swap block 0 and block 4 in this startup area select setting
0x0000 1800 to 0x0000 1FFF	2 KB	BPS[3] or BPS_SEC[3]	Block 7	Swap block 3 and block 7 in this startup area select setting
0x0000 1000 to 0x0000 17FF	2 KB	BPS[2] or BPS_SEC[2]	Block 6	Swap block 2 and block 6 in this startup area select setting

Table 59.33 Example of block protect setting when SAS.BTFLG is 0 and SAS.BTSIZE is 11 (2 of 2)

Address (offset address: BASE_MC)	Block size	Block protect setting	User area block number	Remarks
0x0000 0800 to 0x0000 0FFF	2 KB	BPS[1] or BPS_SEC[1]	Block 5	Swap block 1 and block 5 in this startup area select setting
0x0000 0000 to 0x0000 07FF	2 KB	BPS[0] or BPS_SEC[0]	Block 4	Swap block 0 and block 4 in this startup area select setting

Table 59.34 Example of block protect setting when SAS.BTFLG is 0 and SAS.BTSIZE is 10

Address (offset address: BASE_MC)	Block size	Block protect setting	User area block number	Remarks
The last block address	2 KB	BPS[n] or BPS_SEC[n]	Block n	—
⋮	⋮	⋮	⋮	—
0x0000 8000 to 0x0000 87FF	2 KB	BPS[16] or BPS_SEC[16]	Block 16	—
0x0000 7800 to 0x0000 7FFF	2 KB	BPS[15] or BPS_SEC[15]	Block 7	Swap block 7 and block 15 in this startup area select setting
0x0000 7000 to 0x0000 77FF	2 KB	BPS[14] or BPS_SEC[14]	Block 6	Swap block 6 and block 14 in this startup area select setting
⋮	⋮	⋮	⋮	—
0x0000 4800 to 0x0000 4FFF	2 KB	BPS[9] or BPS_SEC[9]	Block 1	Swap block 1 and block 9 in this startup area select setting
0x0000 4000 to 0x0000 47FF	2 KB	BPS[8] or BPS_SEC[8]	Block 0	Swap block 0 and block 8 in this startup area select setting
0x0000 3800 to 0x0000 3FFF	2 KB	BPS[7] or BPS_SEC[7]	Block 15	Swap block 7 and block 15 in this startup area select setting
0x0000 3000 to 0x0000 37FF	2 KB	BPS[6] or BPS_SEC[6]	Block 14	Swap block 6 and block 14 in this startup area select setting
⋮	⋮	⋮	⋮	—
0x0000 0800 to 0x0000 0FFF	2 KB	BPS[1] or BPS_SEC[1]	Block 9	Swap block 1 and block 9 in this startup area select setting
0x0000 0000 to 0x0000 07FF	2 KB	BPS[0] or BPS_SEC[0]	Block 8	Swap block 0 and block 8 in this startup area select setting

Table 59.35 Example of block protect setting when SAS.BTFLG is 0 and SAS.BTSIZE is 01 (1 of 2)

Address (offset address: BASE_MC)	Block size	Block protect setting	User area block number	Remarks
The last block address	2 KB	BPS[n] or BPS_SEC[n]	Block n	—
⋮	⋮	⋮	⋮	—
0x0001 0000 to 0x0001 07FF	2 KB	BPS[32] or BPS_SEC[32]	Block 32	—
0x0000 F800 to 0x0000 FFFF	2 KB	BPS[31] or BPS_SEC[31]	Block 15	Swap block 15 and block 31 in this startup area select setting
0x0000 F000 to 0x0000 F7FF	2 KB	BPS[30] or BPS_SEC[30]	Block 14	Swap block 14 and block 30 in this startup area select setting
⋮	⋮	⋮	⋮	—
0x0000 8800 to 0x0000 8FFF	2 KB	BPS[17] or BPS_SEC[17]	Block 1	Swap block 1 and block 17 in this startup area select setting
0x0000 8000 to 0x0000 87FF	2 KB	BPS[16] or BPS_SEC[16]	Block 0	Swap block 0 and block 16 in this startup area select setting
0x0000 7800 to 0x0000 7FFF	2 KB	BPS[15] or BPS_SEC[15]	Block 31	Swap block 15 and block 31 in this startup area select setting
0x0000 7000 to 0x0000 77FF	2 KB	BPS[14] or BPS_SEC[14]	Block 30	Swap block 14 and block 30 in this startup area select setting

Table 59.35 Example of block protect setting when SAS.BTFLG is 0 and SAS.BTSIZE is 01 (2 of 2)

Address (offset address: BASE_MC)	Block size	Block protect setting	User area block number	Remarks
⋮	⋮	⋮	⋮	—
0x0000 0800 to 0x0000 0FFF	2 KB	BPS[1] or BPS_SEC[1]	Block 17	Swap block 1 and block 17 in this startup area select setting
0x0000 0000 to 0x0000 07FF	2 KB	BPS[0] or BPS_SEC[0]	Block 16	Swap block 0 and block 16 in this startup area select setting

59.9.2 Error Protection

Error protection detects the issuing of illegal MACI commands, illegal access, and extra MRAM sequencer malfunction. MACI command acceptance is disabled (command-locked state) in response to the detection of these errors. The extra MRAM cannot be programmed while the extra MRAM sequencer is in the command-locked state. For release from the command-locked state, issue the Status Clear or Forced Stop command. The Status Clear command can only be used while the MRDY bit in the MSTATR register is 1. The Forced Stop command can be used regardless of the value of the MRDY bit. While the CMDLKIE bit in the MPAEINT register is 1, an extra MRAM access error (MRAM_MREPR) interrupt is generated if the extra MRAM sequencer enters the command-locked state (the CMDLK bit in the MASTAT register is set to 1).

If the extra MRAM sequencer enters the command-locked state in response to a command, the extra MRAM sequencer continues the processing for programming. If a command is issued in the command-locked state, the ILGLERR bit becomes 1 and the other bits retain the values set due to previous error detection.

Table 59.36 shows error protection types and status bit values after error detection.

Table 59.36 Error protection type (1 of 2)

Error type	Description	ILGCOMERR	SECERR	OTERR	TZFERR	ILGLERR	PRGERR	CFGPRG	MREAE*1
Illegal command error	An undefined size has been specified in the first cycle of command (not byte write)	1	0	0	0	1	0	0	0
	An undefined code is written in the first access of the MACI command	1	0	0	0	1	0	0	0
	The value specified in the last access of the multiple access MACI command is not 0xD0	1	0	0	0	1	0	0	0
	The value (N) specified in the second write access of the MACI command in the Program command or Configuration Set command is wrong	1	0	0	0	1	0	0	0
	Increment Counter or Read Counter command has been issued with an inconsistent MCNTSELR[2:0] setting. (Include CNF_ARCNS[1:0] setting is the mismatch).	1	0	0	0	1	0	0	0
	The MACI command not acceptable in each mode has been issued (see Table 59.13)	1	0	0	0	1	0	0	0
	The MACI command has been issued when command acceptance conditions are not satisfied (see Table 59.14)	0/1	0/1	0/1	0	1	0/1	0/1	0/1
	Increment Counter command has been issued against the anti-rollback area protected by lock bit setting	1	0	0	0	1	0	0	0
	Increment Counter command against ARC_OEMBL is issued when the OEM boot loader version (OEM BL Ver.) output from the security engine is the same or smaller than the ARC_OEMBL value (see section 59.8.5. Anti-Rollback Counter)	1	0	0	0	1	0	0	0
	Increment Counter command against ARC_OEMBL is issued when the previous command is not the Read Counter command against ARC_OEMBL (see section 59.8.5. Anti-Rollback Counter)	1	0	0	0	1	0	0	0
	A command has been issued against extra MRAM sequencer command illegal status (see Table 59.11)	0/1	0/1	0/1	0	1	0/1	0/1	0/1
Program error	An error occurs during programming (when the OTP area data changes from 0 to 1 or verifies the error)	0	0	0	0	0	1	0	0
Configuration set error	An error has occurred during the configuration set	0	0	0	0	0	0	1	0
Access error	Program command has been issued to reserved area in extra MRAM program mode	0/1	0	0	0	1	0	0	1
	Configuration Set command has been issued to reserved configuration area in program mode	0/1	0	0	0	1	0	0	1
	Program command or Configuration Set command has been issued to the TrustZone protection area in extra MRAM program mode	0/1	0	0	1	1	0	0	1
	Configuration Set command has been issued to the operation mode protection area in extra MRAM program mode	0/1	0	0	0	1	0	0	1

Table 59.36 Error protection type (2 of 2)

Error type	Description	ILGCOMERR	SECERR	OTERR	TZFERR	ILGLERR	PRGERR	CFGPRG	MREAE*1
Security error	Configuration Set command for the BTFLG bit setting has been issued when the FSPR bit is 0 (see section 59.7.4.8. Configuration Set Command)	0	1	0	0	1	0	0	0
Others	The MACI command issuing area has been accessed in read mode	0	0	1	0	1	0	0	0
	The MACI command issuing area has been read in program mode	0	0	1	0	1	0	0	0

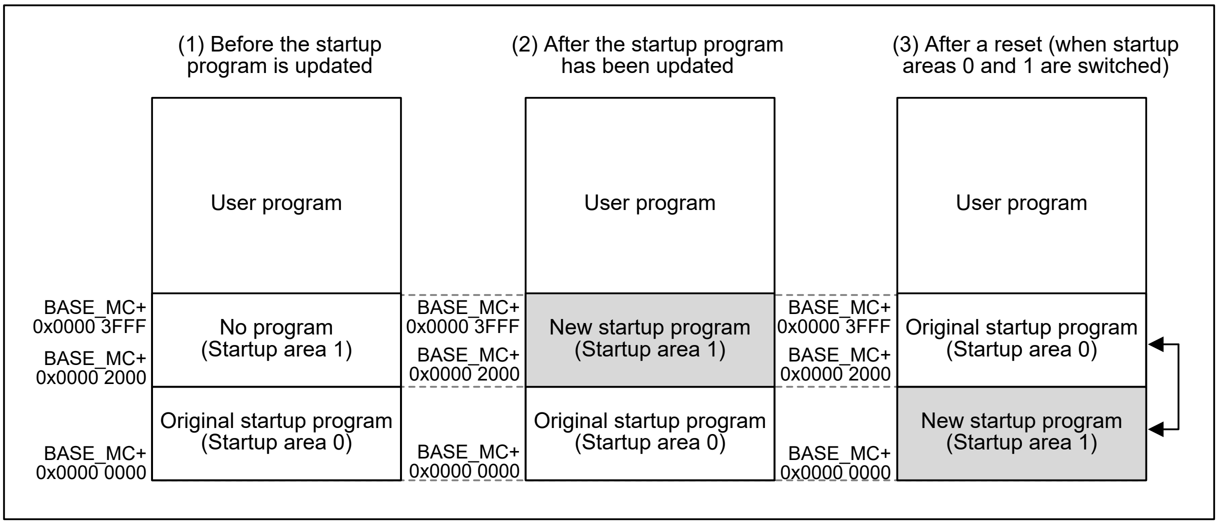
Note 1. MREAE value depends on the MSADDR setting.

59.9.3 Startup Program Protection

Protection of the startup program is for protection of the program to be started after a reset (the startup program). This function provides a way to safely update the startup program when rewriting is suspended during a reset.

The startup area is 8 KB, 16 KB, or 32 KB in size and is assigned to the user area in the code MRAM. The size of the startup area can be changed by the BTSIZE bit. When resizing the startup area, be careful not to destroy the software in use. This function uses the values of the SAS.BTFLG bit, the SAS.BTSIZE bits, and the MSUACR.SAS[1:0] bits to change the area where the startup program is stored in block units (see Figure 59.23 to Figure 59.26).

In protection of the startup program, the state of the selection of the startup area can be fixed by the POFSPS.POFSPS[13] bit. However, the POFSPS.POFSPS[13] bit is not restored to 1 once the flag is set to 0. Exercise extra caution when handling the POFSPS.POFSPS[13] bit.



Note: The addresses are used for reading, programming, and erasure. Startup area 0 is erased when the Block Erase command is issued for the address BASE_CF + 0x0000_0000 with the allocation shown under (3).

1. Before the startup program is updated
   When SAS.BTFLG = 1 and MSUACR.SAS[1:0] = 0xb, the address range from BASE_MC + 0x0000_2000 to BASE_MC + 0x0000_3FFF is selected as startup area 1 and the address range from BASE_MC + 0x0000_0000 to BASE_MC + 0x0000_1FFF is selected as startup area 0.
2. After the startup program has been updated
   A new startup program can be written to startup area 1. At this time, startup area 0 still contains the previous startup program, so rewriting by starting up from startup area 0 remains possible if rewriting to the startup area 1 fails.
3. After a reset (when startup areas 0 and 1 are switched)
   When rewriting startup area 1 is successfully completed, switch the assignment of startup areas 0 and 1 by rewriting the startup area selection bit (the SAS.BTFLG bit) with the Configuration Set command. Afterwards, the program is started up from startup area 1 after a reset.

Figure 59.23 Concept of protection of the startup program

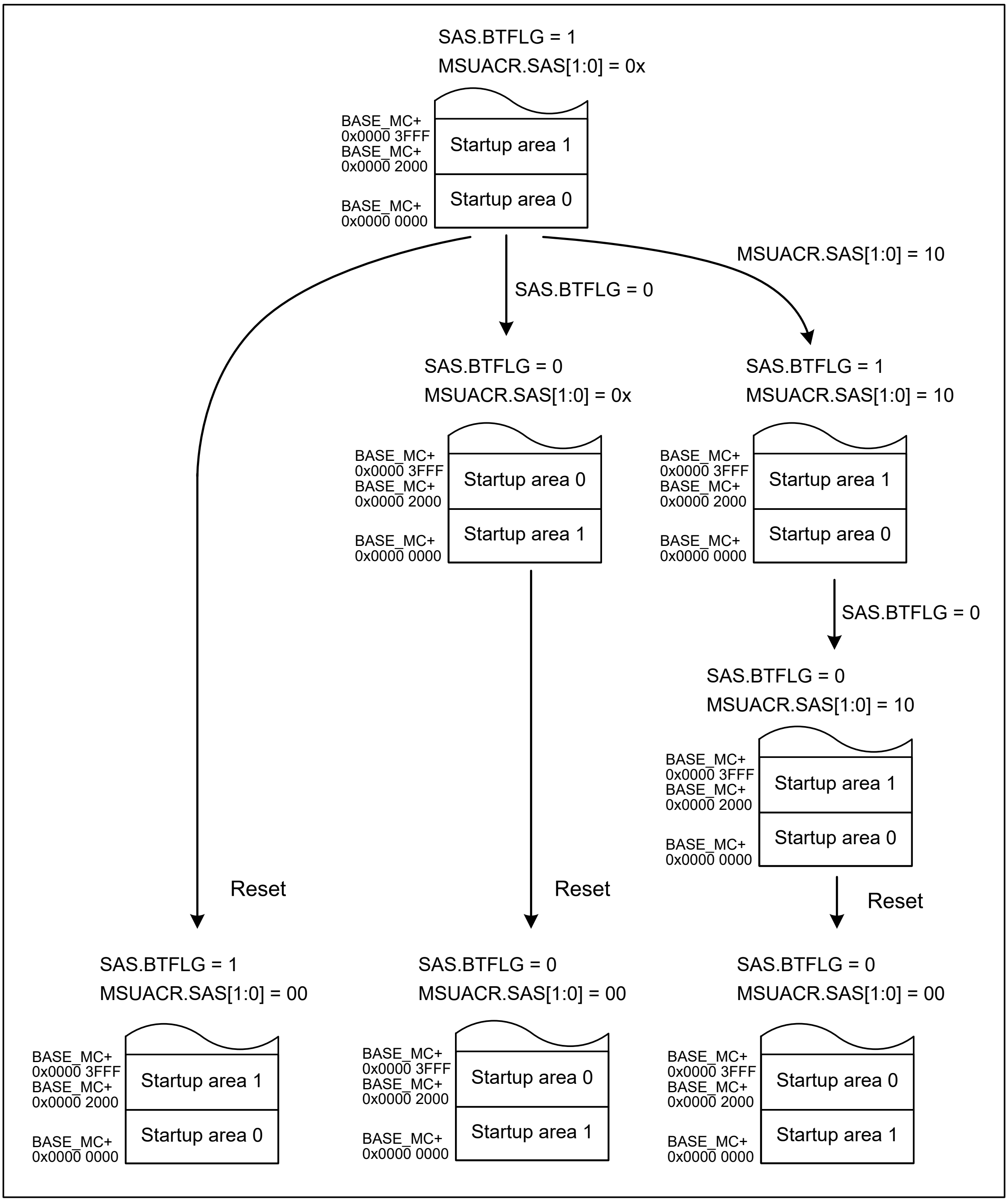


Figure 59.24 Example 1 of transitions for startup program protection settings

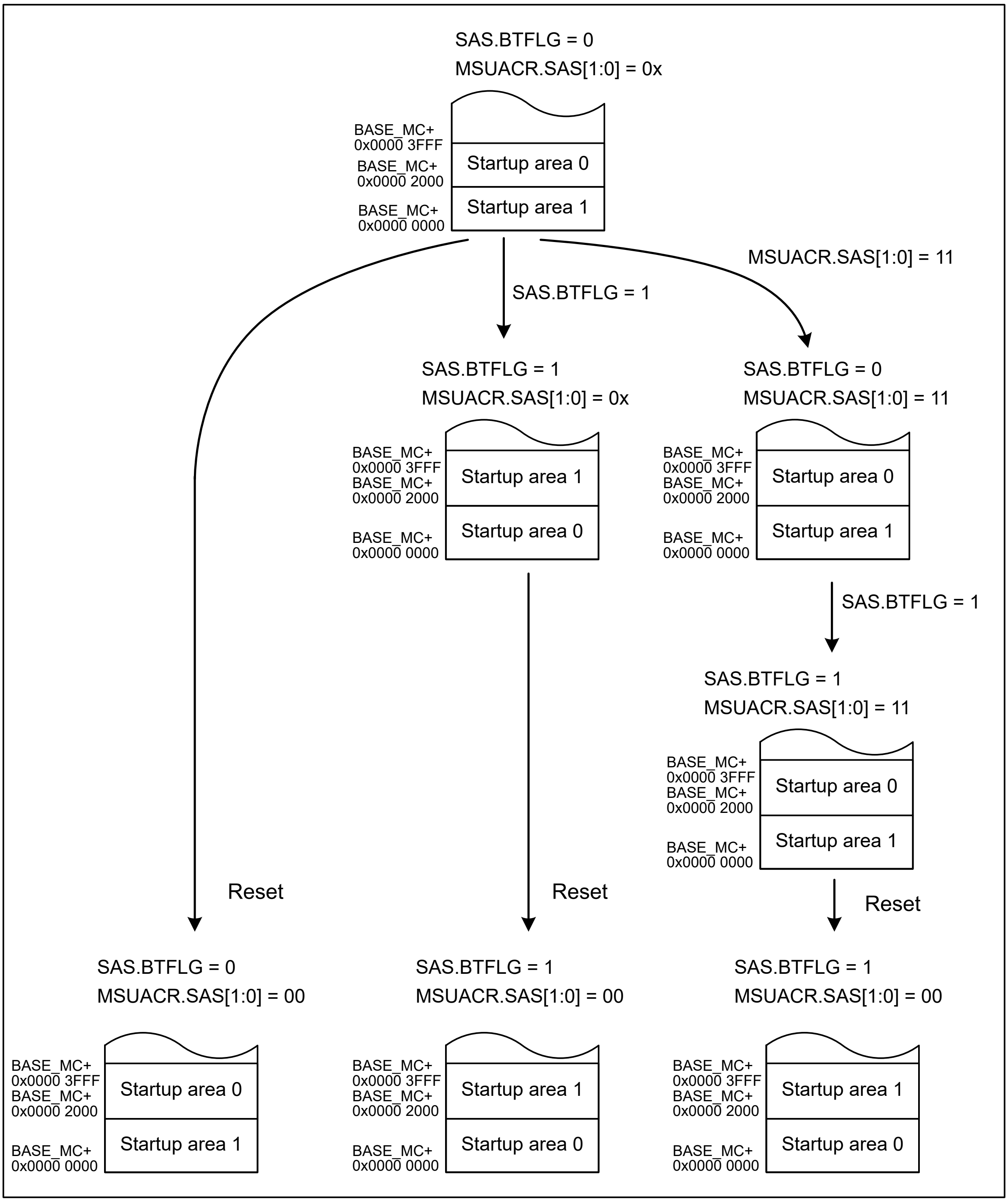


Figure 59.25 Example 2 of transitions for startup program protection settings

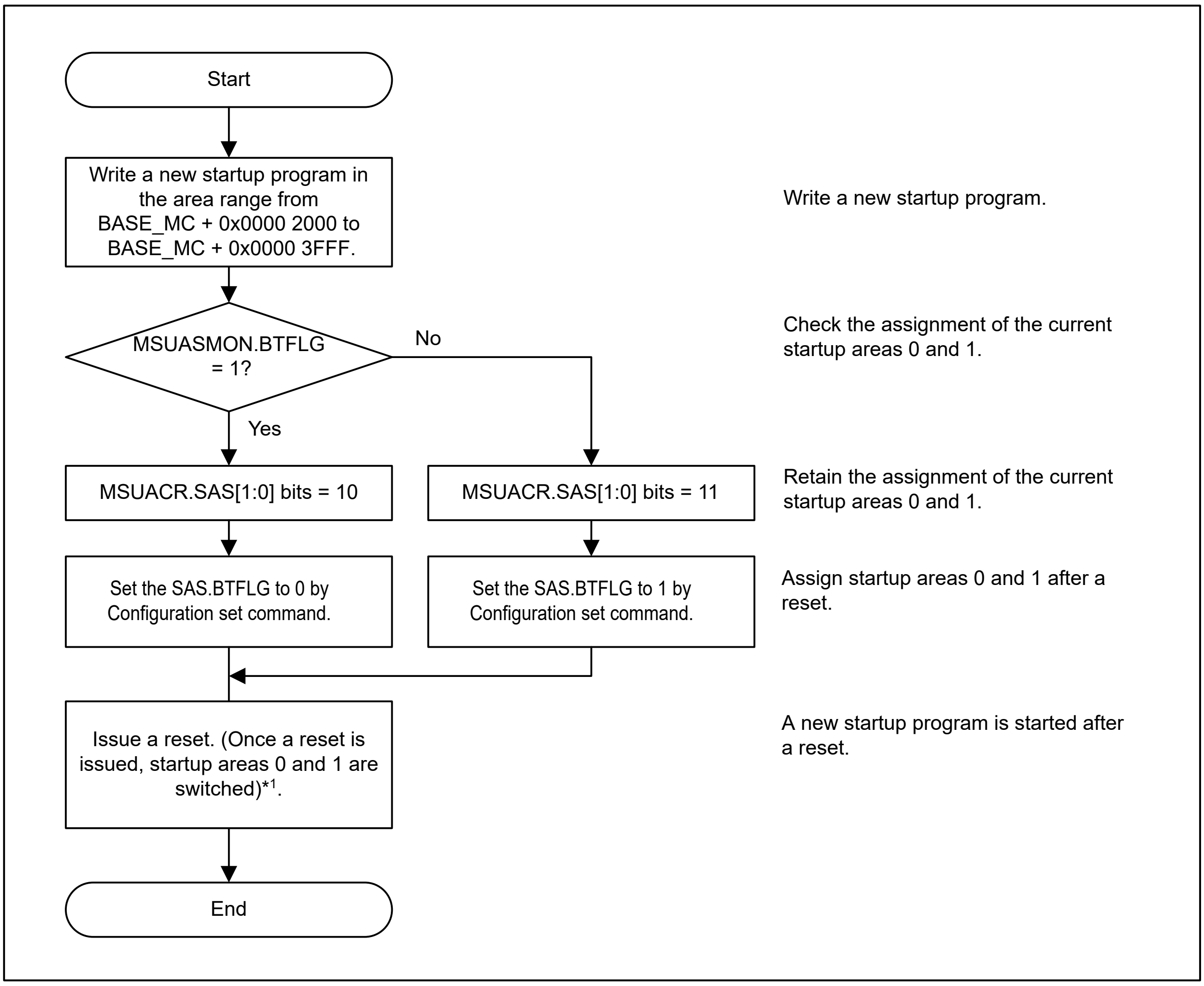


Figure 59.26 Concept of protection of the startup program

59.9.4 Fault Correction and Detection

MRAM systems are equipped with ECC circuitry to correct and detect errors. DEC-TED (double-bit error correction and triple-bit error detection) ECC corrects errors in 128-bit data units. Code MRAM uses two ECC circuits for 256-bit readout, and extra MRAM uses one ECC circuit for 128-bit readout.

If there is a DEC error in the code MRAM reading, error correction is performed and the MRCRAES.DECERRC bit is set to 1. If there is a TED error, the MRCRAES.TEDERRC bit is set to 1, but error correction is not possible. When an error is detected, the error address is stored in the MRCRDEA register for DEC error and in the MRCRTEA register for TED error. If a DEC error occurs, read the data at the error address, and write back. If a TED error occurs, update all 32 bytes of data and overwrite with the correct data.

If there is a DEC error in the extra MRAM reading, error correction is performed and the MRERAES.DECERRE bit is set to 1. If there is a TED error, the MRERAES.TEDERRE bit is set to 1, but error correction is not possible. When an error is detected, the error address is stored in the MRERDEA register for DEC errors and in the MRERTEA register for TED errors. If a DEC error occurs, read the data at the error address, and write back. If a TED error occurs, update all 32 bytes of data and overwrite with the correct data.

59.9.4.1 Test Method for ECC Circuits

The ECC circuit can be tested by setting the program data and ECC data to desired values. A block diagram of the ECC test circuit is shown in Figure 59.27.

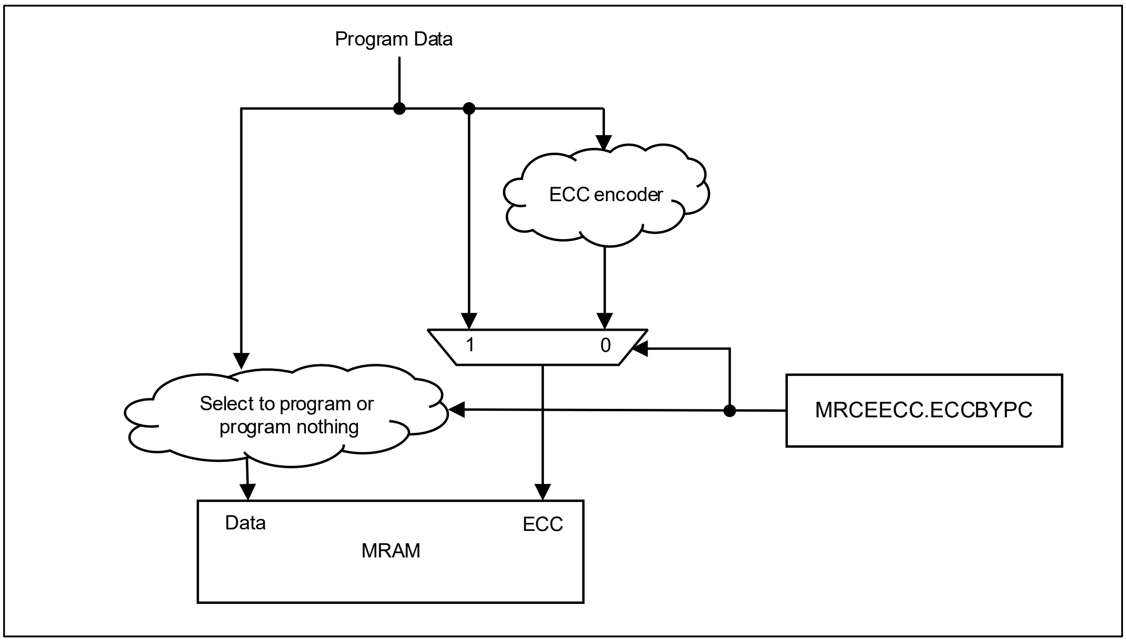


Figure 59.27 Block diagram of the ECC test circuit

Table 59.37 Changes in data allocation depending on the value of MRCEECC.ECCBYPC

MRCEECC.ECCBYPC	Code MRAM	
	Data[255:0]	ECC[33:0]
0	Program data[255:0]	ECC code generated from program data[127:0]
1	Program nothing	{Program data[144:128], Program data[16:0]}

Figure 59.28 shows the ECC decoder test flow of the code MRAM. The code MRAM has two ECC decoders and ECC errors are merged, so when testing the ECC decoder, test the ECC decoder circuits individually.

[Notation of Figure 59.28]

Note 1. To eliminate ECC errors in instruction fetching.

Note 2. If you get an error, try the test again at a different address.

Note 3. Invert the bits of the code according to the error condition you want to check (1 to 3)

Replace the bits shown in Table 59.37 with the code you created and write to the same address as DATA.

Note 4. See Table 59.38.

Table 59.38 Check results

Check results	Expected value		
	1-bit invert	2-bit invert	3-bit invert
MRCRAES.DECERRC	0	1	0
MRCRAES.TEDERRC	0	0	1

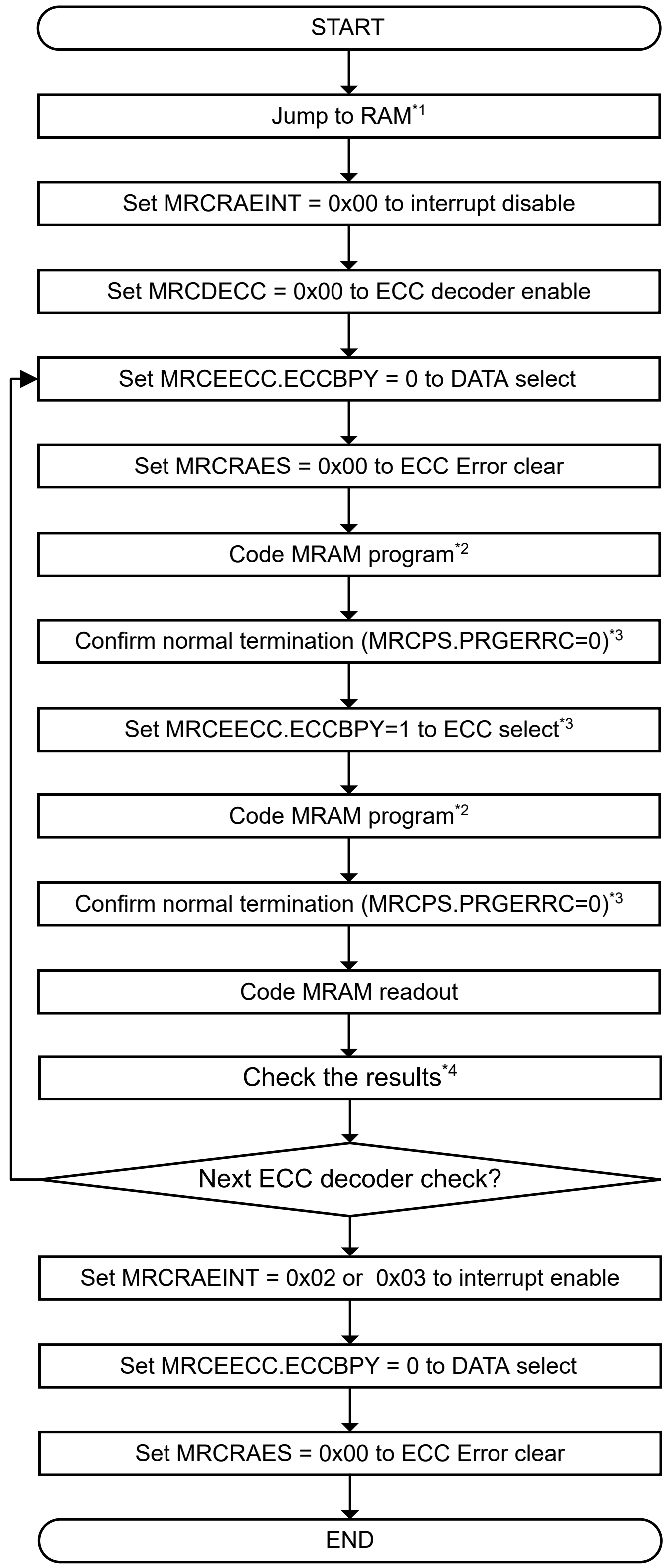


Figure 59.28 Code MRAM ECC decoder test flow

59.10 Configuration Update Transfer

After Program or Configuration Set command operation, the system status can be updated by configuration update transfer without system reset. MACI automatically determines a valid area and executes multidata transfer with the MCTRLSR register setting. With the TRTRG bit in MCTRCNTR written as 1 in the read mode (MENTRYR is 0x0000 and the WHUKEXE bit in MREZS register is 0), MACI transfers data from extra MRAM. MCTRSTATR register can monitor the processing status of configuration update transfer. Extra MRAM cannot be read during configuration data transfer.

Figure 59.29 shows configuration update transfer usage.

If the WHUKEXE bit in the MREZS register is set to 1 in the configuration update transfer flow, the configuration update transfer must be reexecuted after the W-HUK zeroing is finished.

See section 59.5.22. MCTRCNTR : MRAM Configuration Update Transfer Control Register, section 59.5.23. MCTRLSR : MRAM Configuration Update Transfer List Select Register, and section 59.5.24. MCTRSTATR : MRAM Configuration Update Transfer Status Register for details of the associated registers specifications (MCTRCNTR, MCTRLSR, and MCTRSTATR registers).

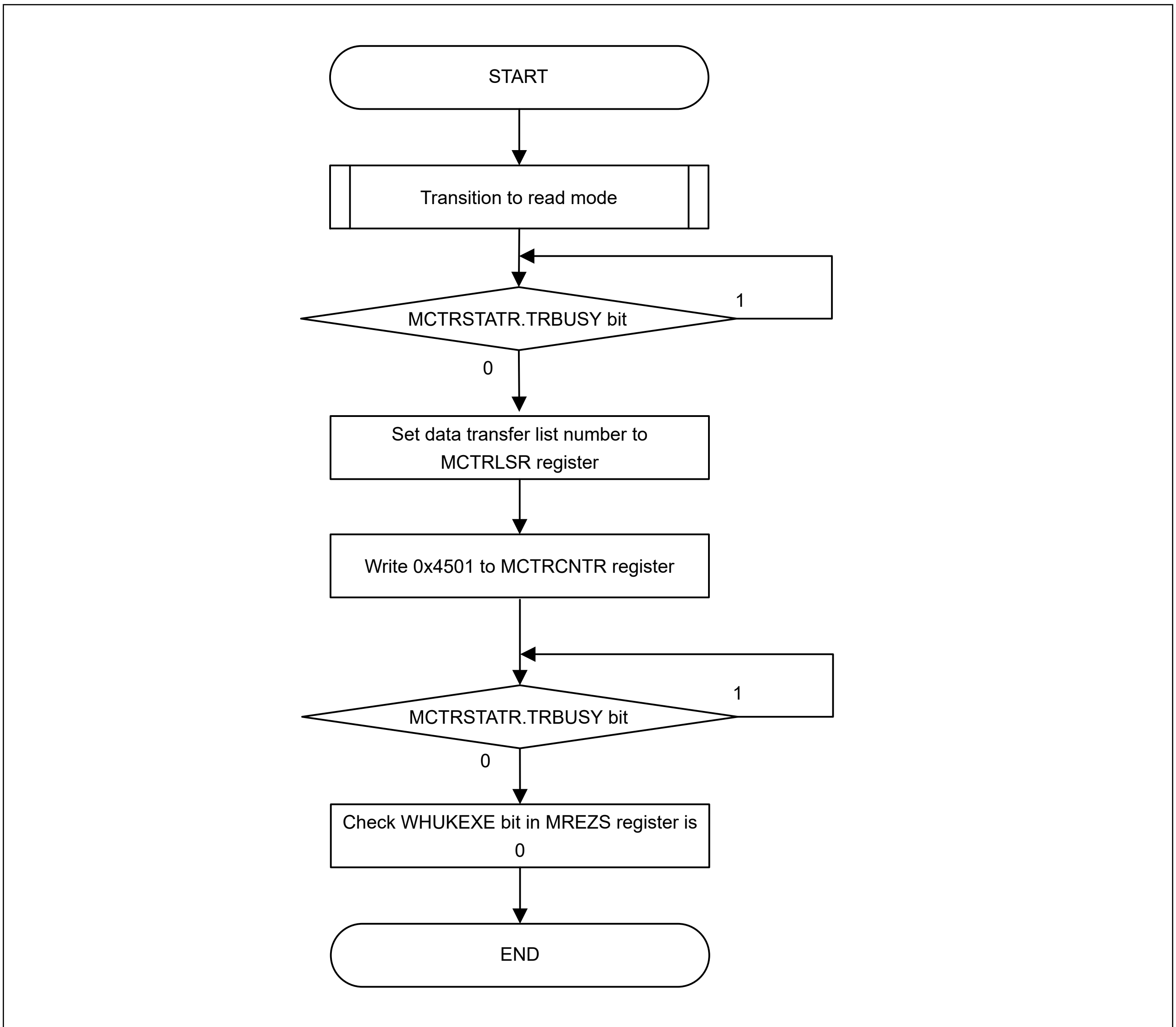


Figure 59.29 Configuration update transfer usage

59.11 Boot Mode

There are two serial programming modes:

- The boot mode (for the SCI interface) with SCI9
- The boot mode (for the USB interface) with USBFS.

Table 59.39 lists the I/O pins used in boot mode. Table 59.40 lists the available communication interface used in the boot mode.

Table 59.39 I/O pins used in boot mode

Pin name	I/O	Mode to be used	Use
MD	Input	Boot mode (for the SCI interface) Boot mode (for the USB interface)	Selection of operating mode
P208/RXD9	Input	Boot mode (for the SCI interface)	For host communication (to receive data through SCI)
P209/TXD9	Output		For host communication (to transmit data through SCI)
USB_DP, USB_DM	I/O	Boot mode (for the USB interface)	Data input/output of USB
USB_VBUS	Input		Detection of connection and disconnection of USB cables

Table 59.40 Available communication interface used in boot mode

Main clock oscillator or external clock is connected	Yes	No	No
Sub-clock oscillator is connected*1	Yes or No	Yes	No
Available interface	SCI or USB	SCI or USB	SCI
Tool connection time*2	Up to 1 second	Up to 2 seconds	Up to 3 seconds

Note 1. The drive capability of the sub-clock oscillator is set to standard by the SOMCR.SODRV bit. If you use the crystal that corresponds to the low drive capability on your board, the crystal may not oscillate in the boot mode.
Note 2. See the boot firmware manual for detail of the connection time.

59.11.1 Boot Mode (for the SCI Interface)

In boot mode (for the SCI interface), the host sends control commands and data for programming, and the MRAM is programmed accordingly. An on-chip SCI handles transfer between the host and this MCU in asynchronous mode. Tools for transmission of control commands and the data for programming must be prepared in the host.

When this MCU is activated in boot mode (for the SCI interface), the program on the dedicated area of the MCU is executed. The boot program automatically adjusts the bit rate of the SCI and controls programming/erasure by receiving control commands from the host.

Figure 59.30 shows the system configuration for operations in boot mode (for the SCI interface).

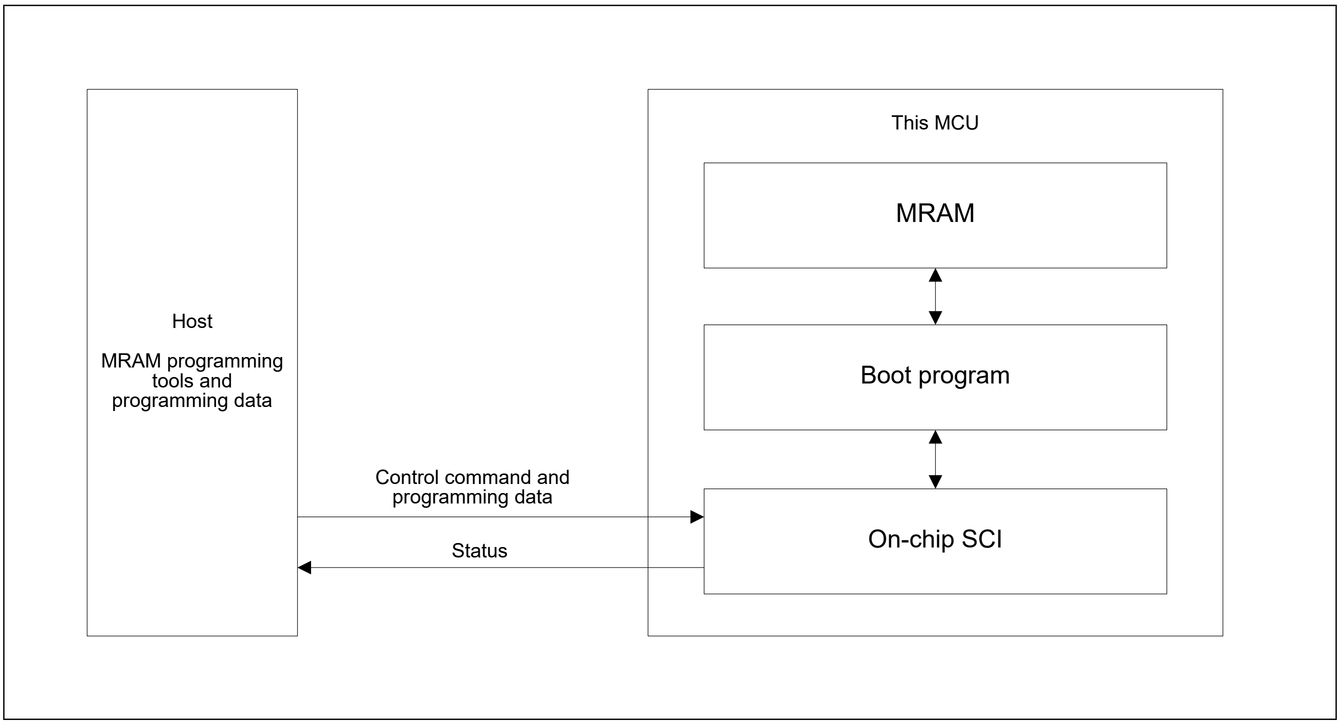


Figure 59.30 System configuration for operations in boot mode (for the SCI interface)

59.11.2 Boot Mode (for the USB Interface)

In boot mode (with the USB interface), the MRAM can be programmed by sending control commands and program data from the host. An on-chip USB is used for communications between the host and this MCU. The host requires tools for sending control commands and data for programming. Figure 59.31 shows the configuration of a system for use in boot mode (for the USB interface). The USB cable must be connected on reset release.

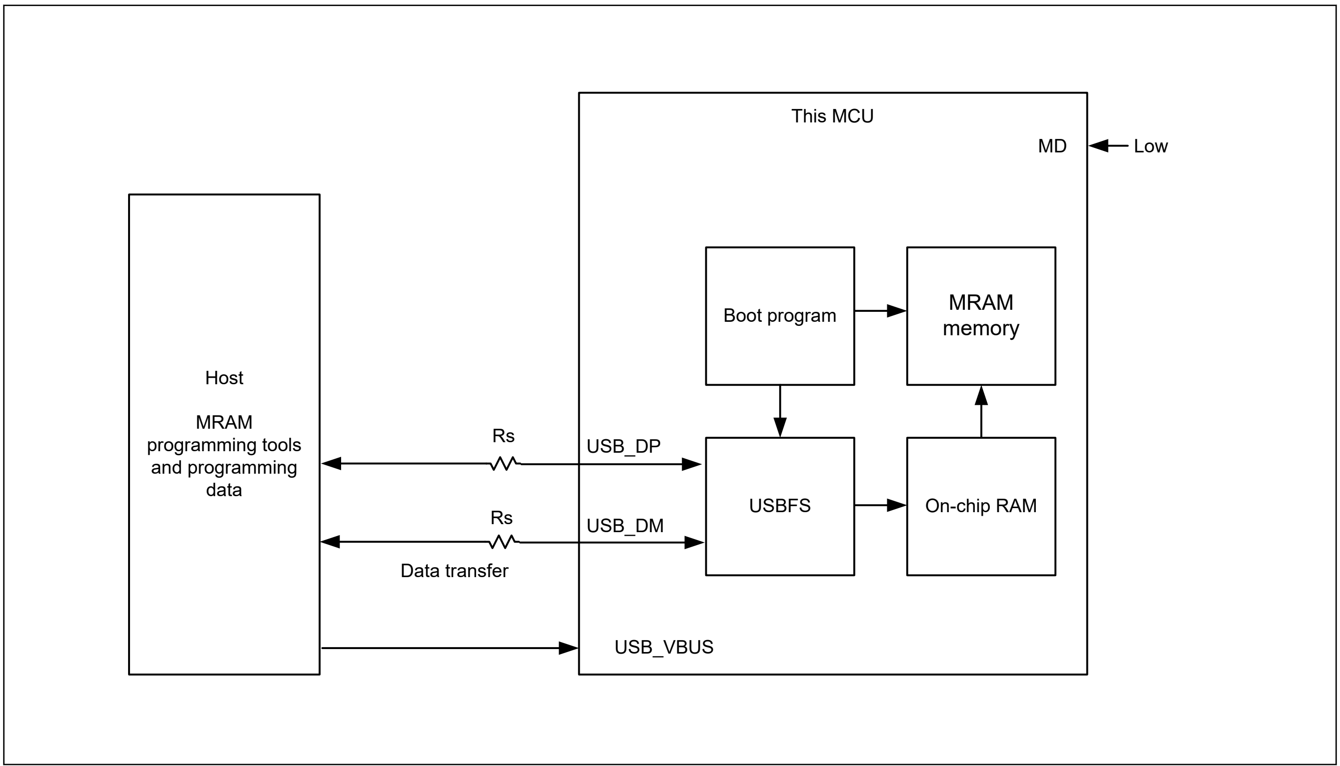


Figure 59.31 System configuration in boot mode (for the USB interface)

59.12 Using the Serial Programmer for Rewriting

A serial programmer can be used to rewrite MRAM in boot mode.

(1) Serial Programming

This MCU is mounted on the system board at the time of serial programming. Providing a connector to the board enables rewriting of this MCU by the serial programmer to proceed.

59.12.1 Environments for Serial Programming

The recommended environments for rewriting the MRAM of the MCU with data are described in Figure 59.32.

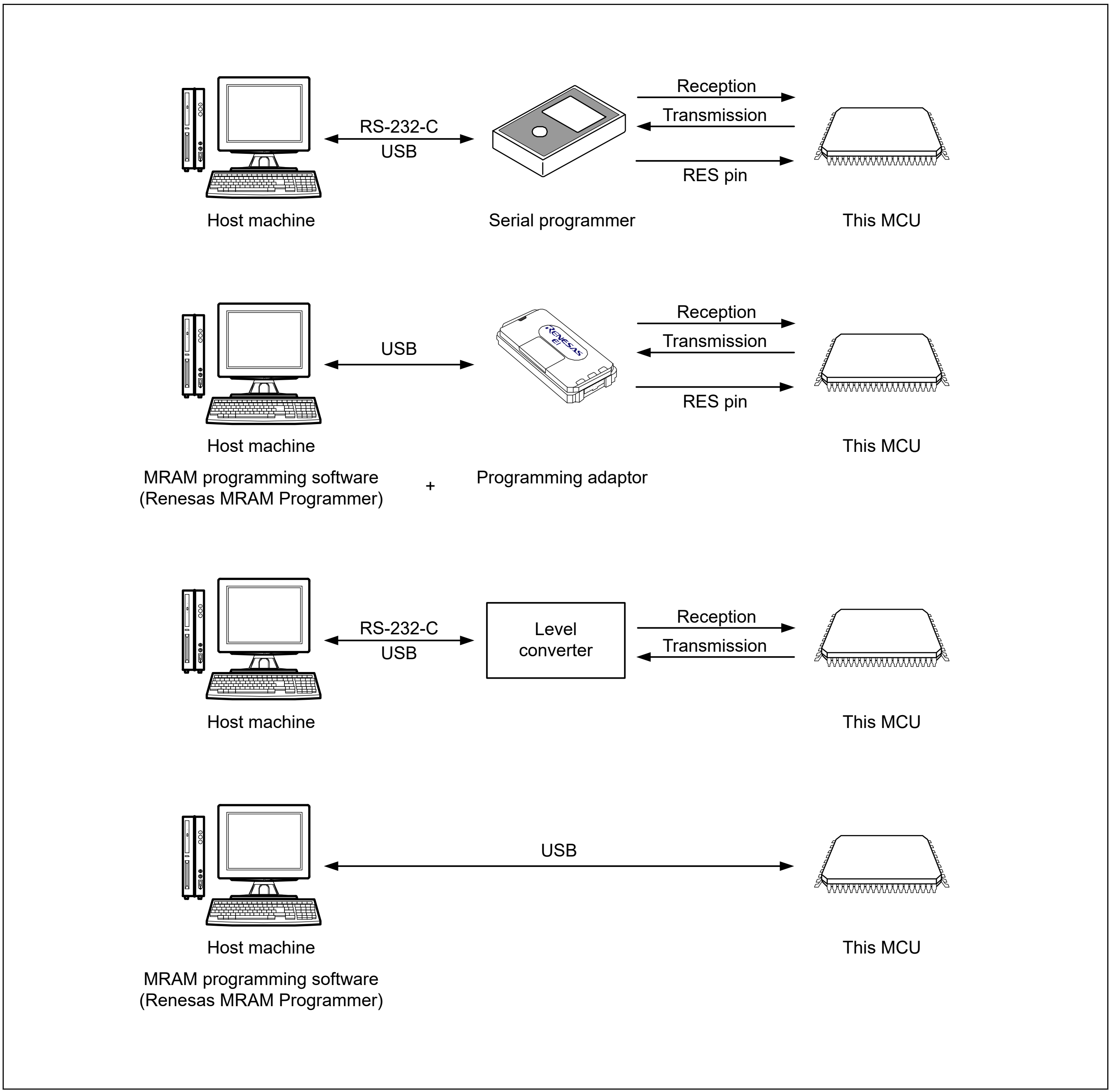


Figure 59.32 Environments for rewriting the MRAM

59.13 Programming Through Self-Programming

59.13.1 Overview

This MCU supports programming of the MRAM by the user program itself. The write page buffer or the MACI commands can be used with user programs for writing to the MRAM. This allows upgrading of user programs and rewriting of constant data fields.

The program for rewriting must be transferred to the internal RAM or external memory in advance when the BGO is not available or when rewriting the option-setting memory.

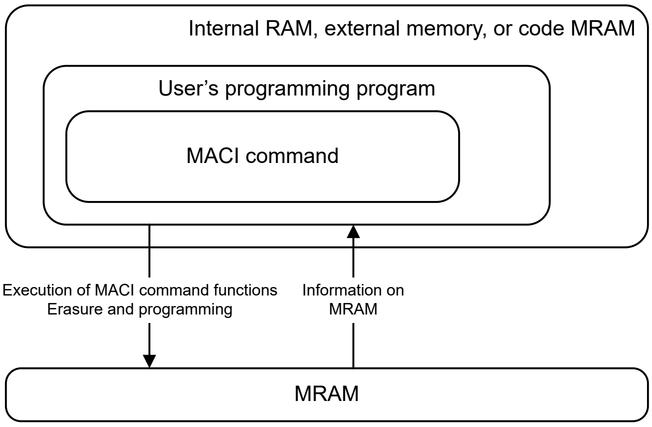


Figure 59.33 Schematic view of self-programming

For comprehensive information on the self-programming, see section 59.7. MACI Commands.

59.13.2 Parallel Accessibility

Parallel read and program accesses are possible in conditions shown in Table 59.41.

Read while program (Background operation: BGO) is possible between different MRAM banks. Parallel read to different MRAM banks is possible. Only one MRAM bank can be programmed at the same time. W-HUK zeroing has the highest priority among all MRAM programming requests. See section 59.8.6. HUK Protection.

Programming the code MRAM and the Extra MRAM cannot be executed at the same time, so they are arbitrated. The program executed later is forced to wait until the program executed first is completed. If the program for the code MRAM is started while MSTATR.MRDY = 0, the program will start after MSTATR.MRDY = 1. And if the program for the extra MRAM (issue the Program, Configuration set, or Increment Counter command) is started while MRCPS.PRGBSYC = 1, the program will start after MRCPS.PRGBSYC = 0. Therefore, in this case, the program time is the sum of the code and the extra MRAM program time.

There is also arbitration between read and program operations for the same MRAM macro, and they cannot be executed simultaneously. The values listed in the MRAM characteristics are for independent operation time without arbitration.

Table 59.41 Parallel accessibility

		Code MRAM		Extra MRAM	
		Read	Program	Read	Program
Code MRAM	Read	X	X	O*1	O
	Program	X	X	O	X
Extra MRAM	Read	O*1	O	X	X
	Program	O	X	X	X

Note: O: Executable
X: Not executable

Note 1. Bus masters cannot read extra MRAM during configuration update transfer. See section 59.10. Configuration Update Transfer.

59.13.3 Reading MRAM

Special settings are not required to read MRAM after release from the reset state. Data can simply be read out through access to addresses in the MRAM.

When reading MRAM that has not yet been programmed (that is, in the nonprogrammed state), all bits are read as 1.

59.14 Usage Notes

(1) Area Where Programming was Interrupted by W-HUK Zeroing

The data stored in the area where programming was suspended due to zeroing of the W-HUK is undefined. To prevent malfunctions due to reading undefined data, care should be taken not to fetch instructions or read data from the area where writing has been suspended.

(2) Actions Prohibited During Programming

- Do not reduce the power to less than the operating voltage.

(3) Prohibition of Additional Programming

An OTP area that uses ECC cannot be programmed twice with different data.

(4) Resets During Programming

For a reset due to the signal on the RES# pin during programming of the MRAM, wait for at least t_{RESW} (see section 62, Electrical Characteristics) of the reset input period once the operating voltage is within the range stipulated in the electrical characteristics, then release the device from the reset state.

(5) Allocation of Vectors for Interrupts and Other Exceptions During Programming

Generation of an interrupt or other exception during programming may lead to fetching of the vector from the code MRAM. The vectors cannot be read from code MRAM until the program is finished. Code MRAM requires t_{PMC} (see section 62, Electrical Characteristics) to program. Under conditions where BGO cannot be used, set the address of the vector to an address that is not in the code MRAM. Alternatively, make sure that no handling of interrupts or exceptions proceeds during programming.

(6) Synchronization of Code MRAM Program Completion and Subsequent Instruction Generation

When the CPU writes to code MRAM, there is a time lag from execution of MRAM programming by code MRAM program sequencer. For a barrier instruction that requires memory access completion, appropriate processing for synchronization is required to execute subsequent instructions. The processing for synchronization is shown as follows:

1. Code MRAM programming (for example, STR instruction).
2. Barrier instruction (for example, DMB instruction).
3. Flush the code MRAM programming data (see Figure 59.2).
4. Synchronization (wait for the flushing process to finish).
5. Subsequent instruction.

If it is necessary to avoid CPU stall in Step 4., completion of code MRAM programming should be checked by the PRGBSYC bit in the MRCPS register instead of dummy reading in Step 3..

(7) Constraints During Programming of MRAM

Frequency change, transition to software standby, and transition to deep standby are prohibited while MRAM is being programmed.

(8) Emulator Connection

Renesas provides the emulator which supports both debugging using SWD or JTAG communication and serial programming using SCI communication.

Table 59.42 shows the pinout of 10-pin or 20-pin socket pinouts when using this emulator. The pinout of SWD and JTAG is ARM standard, and MD, TXD, RXD pins are added for the serial programming using SCI communication.

Table 59.42 Pin assignment for emulator

Pin No.	SWD	JTAG	Serial programming using SCI
1	VCC	VCC	VCC
2	P210/SWDIO	P210/TMS	NC
4	P211/SWCLK	P211/TCK	P201/MD
6	P209/SWO	P209/TDO	P209/TXD9
8	P208	P208/TDI	P208/RXD9
9	GNDdetect	GNDdetect	GNDdetect
10	nRESET	nRESET	nRESET
12	P308/TCLK	P308/TCLK	NC
14	P307/TDATA[0]	P307/TDATA[0]	NC
16	P306/TDATA[1]	P306/TDATA[1]	NC
18	P305/TDATA[2]	P305/TDATA[2]	NC
20	P304/TDATA[3]	P304/TDATA[3]	NC
3, 5, 15, 17, 19	GND	GND	GND
7	NC	NC	NC
11, 13	NC	NC	NC

60. Capture Engine Unit (CEU)

60.1 Overview

The Capture Engine Unit (CEU) is a capture module that fetches image data externally input and transfers it to the memory. The CEU is connected to the system bus via bus bridge modules.

Lists the features of CEU as follows.

(1) Image data fetch

- Captures an image output from an external module and writes YCbCr data to the memory with it separated into Y data and CbCr data.
- Fetches image data other than YCbCr data, for example, JPEG data, RGB565, from an externally connected module, such as a camera, and sequentially writes the image data to the memory.
- Fetches an interlace source image in both-field units or one-field units and writes it to the memory. In both-field capture, an image can be stored in the memory as a frame image.

(2) Filter processing

- Performs scale-down and removal of high-frequency components (only in the horizontal direction) for an image using internal filters. Note that the scaled-down image must not exceed VGA. The filter processing can be applied to only YCbCr input data.

(3) Format conversion

- Converts image data input in the YCbCr422 format into the YCbCr420 format and writes it to the memory. Note that the conversion algorithm is simple thinning in which the chrominance component (CbCr) of the evennumbered lines is thinned.

The functional overview of the CEU is shown in Table 60.1 and the main functions and their details are shown in Table 60.2.

Table 60.1 Functional overview of CEU (1 of 2)

Classification	Item	Function	Description	Note
Connectable camera	Size*1	5 megapixels	2,560 pixels × 1,920 lines	Image capture mode or data synchronous fetch mode Horizontal : 8-pixel units Vertical : 4-line units Data enable fetch mode 32-byte units The range of the image size that can be input is as follows. Horizontal: 2,560 pixels to 128 pixels Vertical: 1,920 pixels lines to 96 lines
		3 megapixels	2,048 pixels × 1,536 lines	
		2 megapixels	1,632 pixels × 1,224 lines	
		UXGA	1,600 pixels × 1,200 lines	
		SXGA (1)	1,280 pixels × 1,024 lines	
		SXGA (2)	1,280 pixels × 960 lines	
		WXGA	1,280 pixels × 768 lines	
		XGA	1,024 pixels × 768 lines	
		SVGA	800 pixels × 600 lines	
		WVGA	800 pixels × 480 lines	
		VGA	640 pixels × 480 lines	
		CIF	352 pixels × 288 lines	
		WQVGA	480 pixels × 240 lines	
		QVGA	320 pixels × 240 lines, 240 pixels × 320 lines	
		QCIF	176 pixels × 144 lines	
		QQVGA	160 pixels × 120 lines	
		Sub-QCIF	128 pixels × 96 lines	
	Input format	YCbCr422 8 bits	Cb_0, Y_0, Cr_0, Y_1…	Supports clock ratio of 1:1
			Cr_0, Y_0, Cb_0, Y_1…	
			Y_0, Cb_0, Y_1, Cr_0…	
			Y_0, Cr_0, Y_1, Cb_0…	
		YCbCr422 16 bits	{Y_0, Cb_0}, {Y_1, Cr_0}, …	
			{Y_0, Cr_0}, {Y_1, Cb_0}, …	
		Binary data	Specified amount to be fetched on edges of the sync signal	Written sequentially
			Data is fetched with the horizontal sync signal as an enable signal	
	Horizontal and vertical sync signal polarities	Arbitrary	High-active and low-active	
	Capture start location	Arbitrary	Can be specified in camera input clock units	Horizontal: 1-cycle units Vertical: 1-HD (horizontal sync signal) units
	Number of captured pixels	Arbitrary	Can be specified in 8-pixel units horizontally and in 4-line units vertically	
	Interlace	Both-field capture	Stored as a field image	Capture: 2-VD (vertical sync signal) units
			Stored as a frame image	
		One-field capture	Top field or bottom field can be specified	Capture: 1-VD units
Memory write	Output format	YCbCr422 YCbCr420	YCbCr420 is realized by simple thinning	

Table 60.1 Functional overview of CEU (2 of 2)

Classification	Item	Function	Description	Note
Filter function	No scaling or scale-down	Scale-down of captured display	Desired scaling factor from 1/16 to 1 (scaled-down display must not exceed VGA)	
	Low-pass filter		Removal of high-frequency components	Only in the horizontal direction
Module-stop function	Module-stop state can be set to reduce power consumption.			
TrustZone Filter	Security and Privilege attribution can be set.			

Note 1. The range of the image size that can be input depends on the AC characteristics and the frame rate of the connected device, and the transfer speed to the destination RAM.

Table 60.2 Main functions of CEU and their details

Main function	Detailed description
Image data fetch	• Captures an image output from an external module and writes YCbCr data to the memory with it separated into Y data and CbCr data. • Fetches image data other than YCbCr data, e.g. JPEG data, from an externally connected module, such as a camera, and sequentially writes the image data to the memory. • Fetches an interlace source image in both-field units or one-field units and writes it to the memory. In both-field capture, an image can be stored in the memory as a frame image.
Filter processing	Performs scale-down and removal of high-frequency components (only in the horizontal direction) for an image using internal filters. Note that the scaled-down image must not exceed VGA. The filter processing can be applied to only YCbCr input data.
Format conversion	Converts image data input in the YCbCr422 format into the YCbCr420 format and writes it to the memory. Note that the conversion algorithm is simple thinning in which the chrominance component (CbCr) of the even-numbered lines is thinned.

Figure 60.1 shows a block diagram of the CEU.

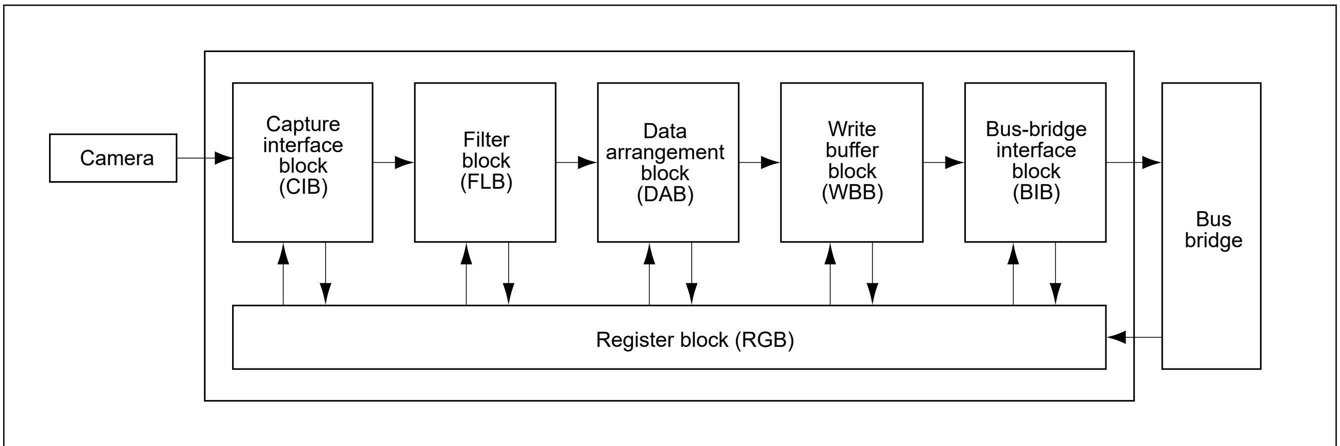


Figure 60.1 Block diagram of CEU

The pin configuration of the CEU is shown in Table 60.3.

Table 60.3 Pin configuration of CEU (1 of 2)

Pin name	Function	I/O	Description
VIO_D15 to VIO_D0*1	CEU data bus	Input	Camera image data input to the CEU
VIO_CLK	CEU clock	Input	Camera clock input to the CEU
VIO_VD	CEU vertical sync	Input	Camera vertical sync signal input to the CEU

Table 60.3 Pin configuration of CEU (2 of 2)

Pin name	Function	I/O	Description
VIO_HD	CEU horizontal sync	Input	Camera horizontal sync signal input to the CEU
VIO_FLD*2	Field signal	Input	Field identification signal to the CEU

Note 1. When the distinction according to the bus width for the data bus is not needed, VIO_D is used in this manual. Otherwise, VIO_D15 to VIO_D0 are used.
Note 2. The VIO_FLD pin is valid only in interlace input mode (CAIFR.IFS=1).

60.2 Register Descriptions

The register configuration of the CEU is shown in Table 60.4.

Most CEU registers have a 2-plane configuration (plane A and plane B). The CEU switches the planes when using these 2-plane registers. A mirror address, which is an address that can always access the register on the unused plane, is provided for each 2-plane register. Figure 60.2 shows the timing to switch the register planes. The CEU switches the register planes at the same time a VD interrupt is asserted.

In the following register descriptions, "during operation" indicates the period that begins when the CEU is activated by the CE bit in the capture start register (CAPSR) and ends when a capture end interrupt (CPE) of the capture event flag clear register (CETCR) occurs. In the read-only bits in each register, the write value should always be 0. If a value other than 0 is written to any of these bits, correct operation cannot be guaranteed.

Table 60.4 Register configuration of CEU (1 of 2)

Register name	Abbr.	R/W	Addresses			Access size
			Address (Plane A)	Address (Plane B)	Mirror Address	
Capture Start Register	CAPSR	R/W	0x4034_8000	—	—	32
Capture Control Register	CAPCR	R/W	0x4034_8004	—	—	32
Capture Interface Control Register*1	CAMCR	R/W	0x4034_8008	—	—	32
Capture Interface Cycle Register*1	CMCYR	R/W	0x4034_800C	—	—	32
Capture Interface Offset Register	CAMOR	R/W	0x4034_8010	0x4034_9010	0x4034_A010	32
Capture Interface Width Register	CAPWR	R/W	0x4034_8014	0x4034_9014	0x4034_A014	32
Capture Interface Input Format Register	CAIFR	R/W	0x4034_8018	—	—	32
CEU Register Control Register	CRCNTR	R/W	0x4034_8028	—	—	32
CEU Register Forcible Control Register	CRCMPR	R/W	0x4034_802C	—	—	32
Capture Filter Control Register	CFLCR	R/W	0x4034_8030	0x4034_9030	0x4034_A030	32
Capture Filter Size Clip Register	CFSZR	R/W	0x4034_8034	0x4034_9034	0x4034_A034	32
Capture Destination Width Register	CDWDR	R/W	0x4034_8038	0x4034_9038	0x4034_A038	32
Capture Data Address Y Register	CDAYR	R/W	0x4034_803C	0x4034_903C	0x4034_A03C	32
Capture Data Address C Register	CDACR	R/W	0x4034_8040	0x4034_9040	0x4034_A040	32
Capture Data Bottom-Field Address Y Register	CDBYR	R/W	0x4034_8044	0x4034_9044	0x4034_A044	32
Capture Data Bottom-Field Address C Register	CDBCR	R/W	0x4034_8048	0x4034_9048	0x4034_A048	32
Capture Bundle Destination Size Register	CBDSR	R/W	0x4034_804C	0x4034_904C	0x4034_A04C	32
Firewall Operation Control Register	CFWCR	R/W	0x4034_805C	—	—	32
Capture Low-Pass Filter Control Register	CLFCR	R/W	0x4034_8060	0x4034_9060	0x4034_A060	32
Capture Data Output Control Register	CDOCR	R/W	0x4034_8064	0x4034_9064	0x4034_A064	32
Capture Event Interrupt Enable Register	CEIER	R/W	0x4034_8070	—	—	32
Capture Event Flag Clear Register	CETCR	R/W	0x4034_8074	—	—	32

Table 60.4 Register configuration of CEU (2 of 2)

Register name	Abbr.	R/W	Addresses			Access size
			Address (Plane A)	Address (Plane B)	Mirror Address	
Capture Status Register	CSTSR	R	0x4034_807C	—	—	32
Capture Data Size Register	CDSSR	R/W	0x4034_8084	—	—	32
Capture Data Address Y Register 2	CDAYR2	R/W	0x4034_8090	0x4034_9090	0x4034_A090	32
Capture Data Address C Register 2	CDACR2	R/W	0x4034_8094	0x4034_9094	0x4034_A094	32
Capture Data Bottom-Field Address Y Register 2	CDBYR2	R/W	0x4034_8098	0x4034_9098	0x4034_A098	32
Capture Data Bottom-Field Address C Register 2	CDBCR2	R/W	0x4034_809C	0x4034_909C	0x4034_A09C	32
CEU Bufferable Write Enable Register	CBWER	R/W	0x4034_80A0	—	—	32

Note: The address shown in this table are Secure region. The address corresponding to the Non-secure region is when b28 of address is set to 1.

Note 1. After changing the setting of a register (CAMCR or CMCYR) that is determined by the external module characteristics, do not start capture until at least 10 external input clock cycles have elapsed.

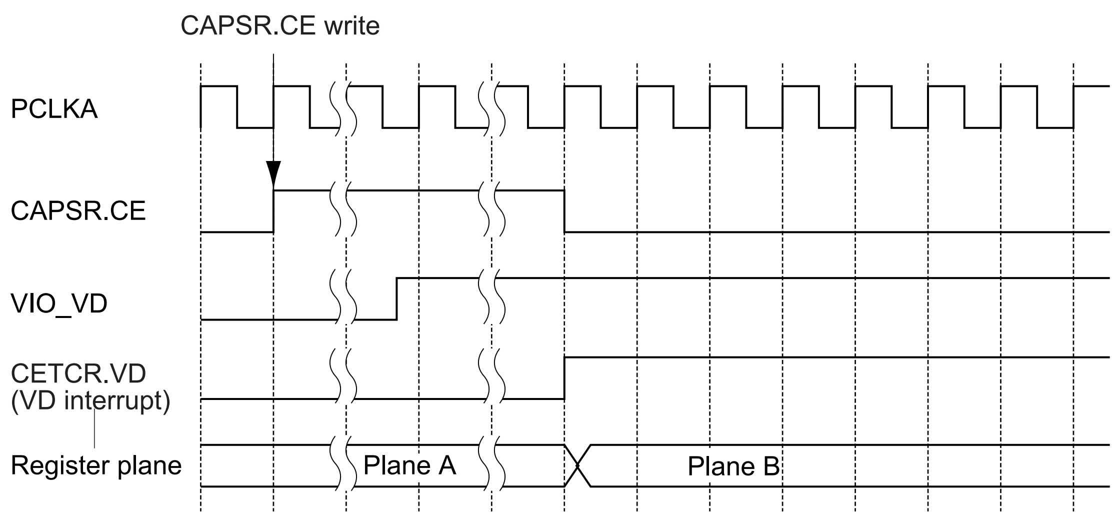


Figure 60.2 Register plane switching timing (VD polarity is high-active in data enable fetch mode)

60.2.1 CAPSR : Capture Start Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPKIL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CE	Capture enable ● Single-capture This bit reserves capturing of the next frame. When 1 is written to this bit, the capture of one frame starts from the next VD input, and stops when the one-frame capture end interrupt (CPE bit in CETCR) is asserted (Figure 60.7). To perform capture again, write 1 to this bit. After the VD or HD polarity is changed, do not write 1 to this bit until the next VD interrupt is asserted. As this bit indicates the capture reserve state, this bit is read as 1 after it is set to 1 and until VD is input. When VD is input, this bit returns to 0 and so is read as 0. The capture end is determined by the one-frame capture end interrupt (CPE bit). This is similar in data fetch mode. Registers should be set before the VD interrupt of the frame where capture starts next. The new register settings take effect at the next VD input. When registers are modified during capturing, the register settings take effect from the capture operations of the next VD input. If a setting register to which writing during capturing is prohibited is modified during capturing, an interrupt source (IGRW bit in CETCR) is generated. For details on the interrupt source, see the description on CETCR. ● Continuous capture When this bit is set to 1 while the CTNCP bit in CAPCR is set to 1, continuous capture starts from the next frame (Figure 60.8). Note that this bit is not cleared to 0 but remains as 1. To stop capturing, clear this bit to 0; capturing stops after the current frame is completed. Continuous capture operations are possible in only image capture mode. The start address of the memory to which the captured data is written to must be set for each frame. 0: Stops capturing 1: Starts capturing	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W
16	CPKIL	Software reset of capturing Write 1 to this bit to perform a software reset of capturing. At a software reset, capturing ends immediately without completing capture operation until the end of a frame. Clear the CE bit to 0 when writing 1 to this bit. Processing of the capture software reset is indicated by this bit being set to 1. When CPKIL = 1, do not start capturing since reset processing is in progress. When restarting capture operations, after referring to the CPTON bit in CSTSR to ensure that the CEU is halted (in the idle state), wait until this bit is cleared to 0. The timing of restarting capture operations is shown in Figure 60.6. When a software reset is generated by this bit, a capture end interrupt (CPE bit in CETCR) may be output immediately after the software reset. However, such kind of interrupt should be ignored. Also, even if the capture end interrupt is not output, the interrupt source (CPE bit) must be cleared before capturing of the next frame. 0: Normal state 1: Software reset of capturing	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CAPSR captures data input to the CEU from an external module.

When both the VD (vertical sync signal) and HD (horizontal sync signal) polarities are high-active, one frame is defined as a period from a VD rising edge to the next VD rising edge, and one line as a period from an HD rising edge to the next HD rising edge. Figure 60.3 shows the timing of one frame (when both the VD and HD polarities are high-active).

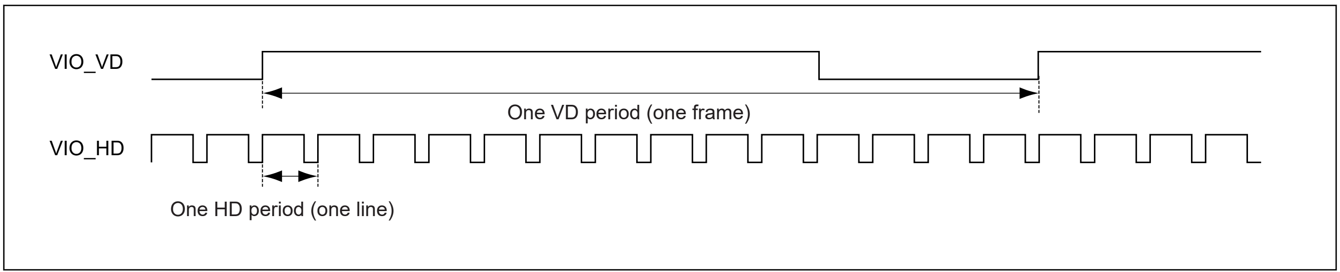


Figure 60.3 Frame timing

When both the VD and HD polarities are high-active, similar to one frame, one field is defined as follows:

- Period from a VD rising edge to the next VD rising edge
- One line is a period from an HD rising edge to the next HD rising edge

The field identification signal FLD should be fixed for at least 1-HD period from a VD input. Figure 60.4 shows the timing of one field (when both the VD and HD polarities are high-active).

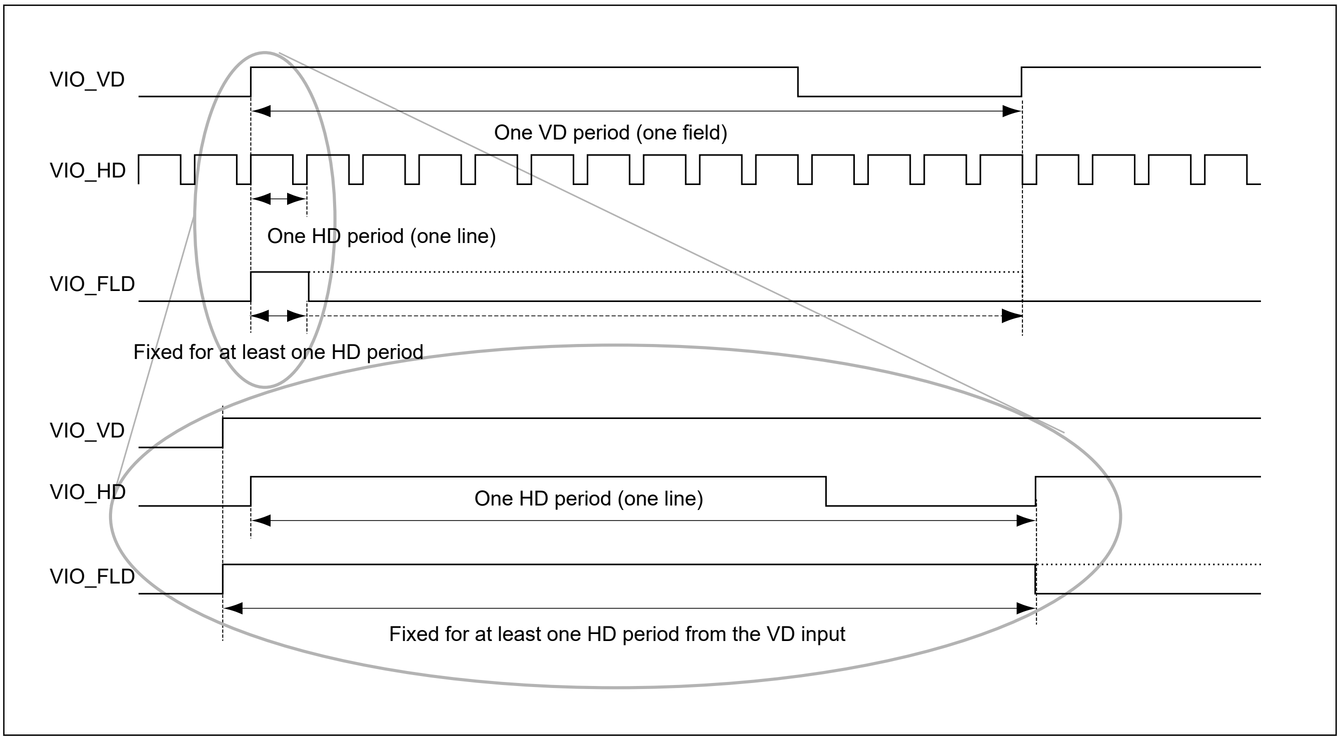


Figure 60.4 One field timing

In data enable fetch, one frame is defined as a period from a VD rising edge to the VD falling edge. With the HD as an enable signal (positive polarity), data of a cycle in which the HD is asserted is fetched while the VD is high. Figure 60.5 shows the timing of one frame for data enable fetch.

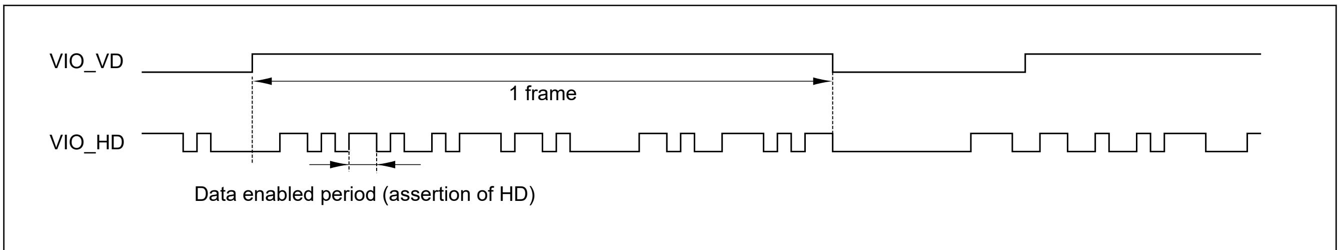


Figure 60.5 Frame timing (data enable fetch)

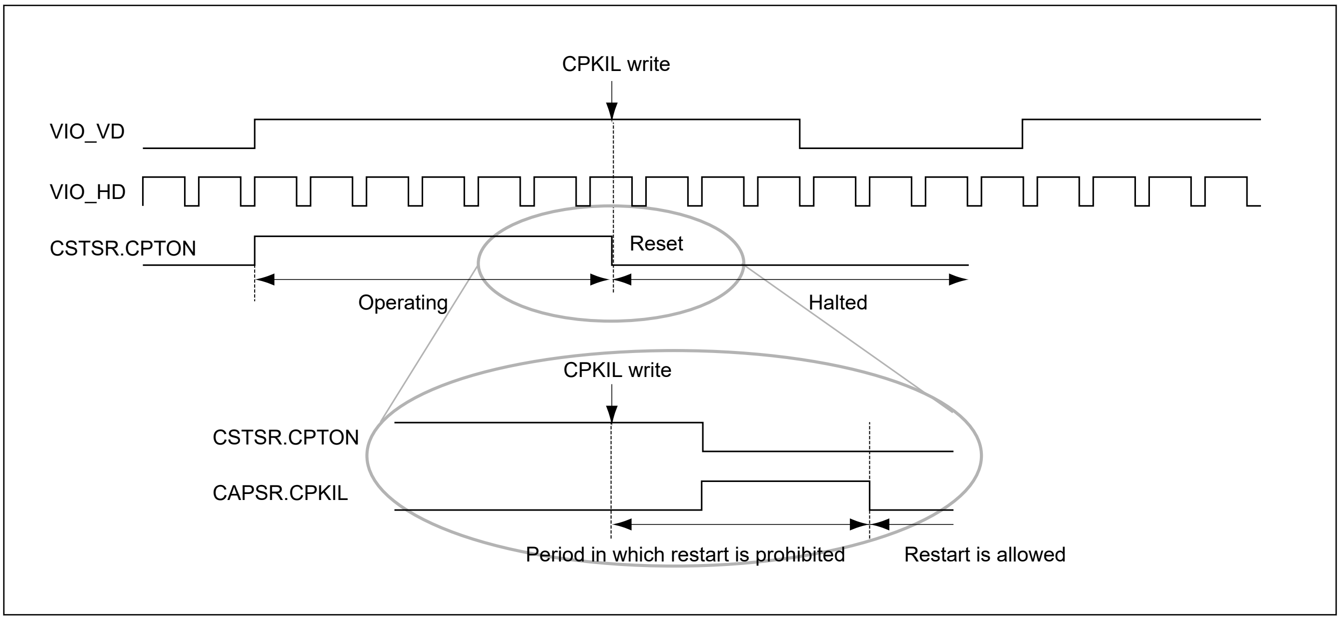


Figure 60.6 Timing of software reset and restart of capturing

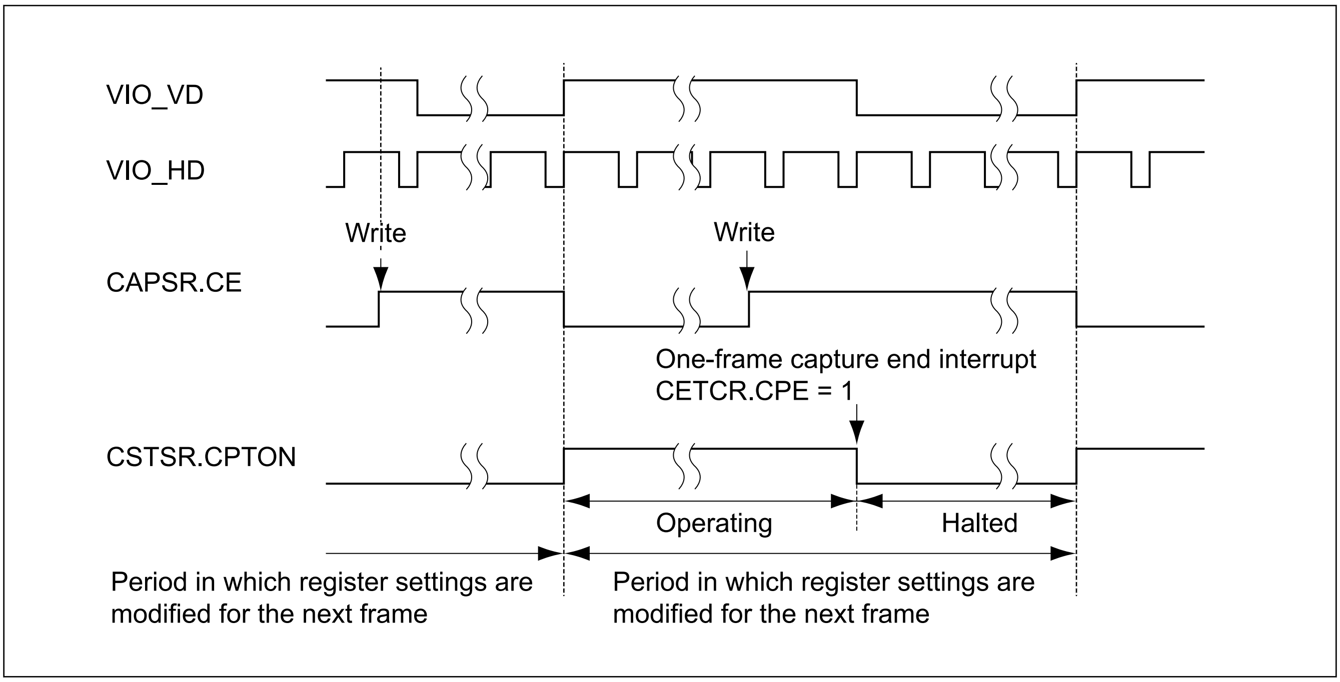


Figure 60.7 Timing of modifying CE bit and register setting in one frame capture

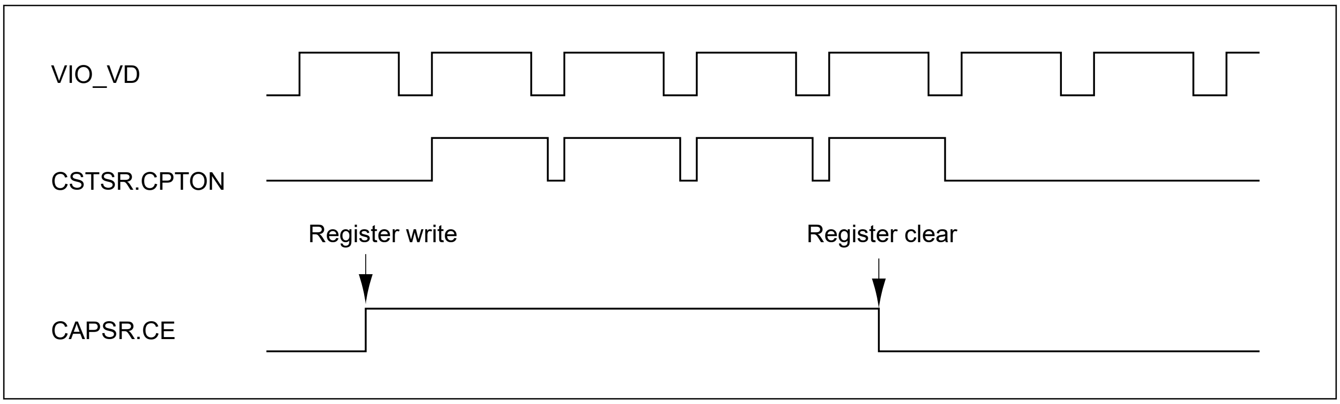


Figure 60.8 Continuous-frame capture

60.2.2 CAPCR : Capture Control Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	FDRP[7:0]								—	—	MTCM[1:0]		—	—	—	CTNCP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
16	CTNCP	Continuous capture When capturing is started with this bit set to 1, capturing continues until the CE bit in CAPSR is cleared to 0 or a software reset is initiated by the CPKIL bit in CAPSR (see Figure 60.8). Continuous capture must be set before capturing is started. This bit is modified only after 0 is written to the CE bit to stop capturing. If this bit is modified during capturing, correct operation cannot be guaranteed. In data fetch mode, clear this bit to 0. 0: One-frame capture when CAPSR.CE = 1 1: Continuous capture until CAPSR.CE = 0	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
21:20	MTCM[1:0]	Specify the unit for transferring data to a bus bridge module. The efficiency of writing image data can be improved by continuously accessing the addresses. To improve the write efficiency, set these bits to 11. The setting of these bits appear to be unchanged from the outside. 0 0: Image capture: Y data and C data are transferred in 32-byte units Data fetch : Data is transferred in 32-byte units 0 1: Image capture: Y data and C data are transferred in 64-byte units Data fetch : Data is transferred in 64-byte 1 0: Image capture: Y data and C data are transferred in 128-byte units Data fetch : Data is transferred in 128-byte 1 1: Image capture: Y data and C data are transferred in 256-byte units Data fetch : Data is transferred in 256-byte	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
31:24	FDRP[7:0]	Set the frame drop interval in continuous-frame capture. When these bits are cleared to 0, frame drop is not performed, and all frames are captured. Figure 60.9 shows the value set in these bits and the timing of captured frames. The frame drop interval unit differs according to the capture setting. Table 60.5 shows the relationship between the capture setting and frame drop interval unit. The image of the frame drop timing for each capture setting when these bits are set to 2 is shown in Figure 60.10. In both-field capture, capturing is performed continuously for 2-VD periods, regardless of whether the second field is the top field or bottom field. In addition, in both-field capture, the frame drop counter is incremented when the first field has been identified as the top field or bottom field, regardless of whether the second field is the top field or bottom field. When 0 is written to the CE bit in CAPSR, capturing terminates after the current frame has been captured for a capture frame. However, for a drop frame, capturing is forcibly terminated in the CEU so no capture end interrupt (CPE bit in CETCR) is output. While CAPSR.CE = 1, do not change the setting of these bits. Note: Do not change the setting of these bits during continuous capture operations. To change the setting of these bits, stop continuous capture (CAPSR.CE = 0), clear the CTNCP bit (continuous capture) in CAPCR to 0, and then restart continuous capture. Continuous capture is performed during the period of CAPSR.CE = 1 shown in Figure 60.9.	R/W

Note: S-TYPE-3, P-TYPE-3

CAPCR sets continuous-frame capture and the frame drop intervals.

Do not modify this register during operation. If this register is modified during operation, correct operation cannot be guaranteed. In addition, the IGRW bit (interrupt source) in CETCR is set to 1.

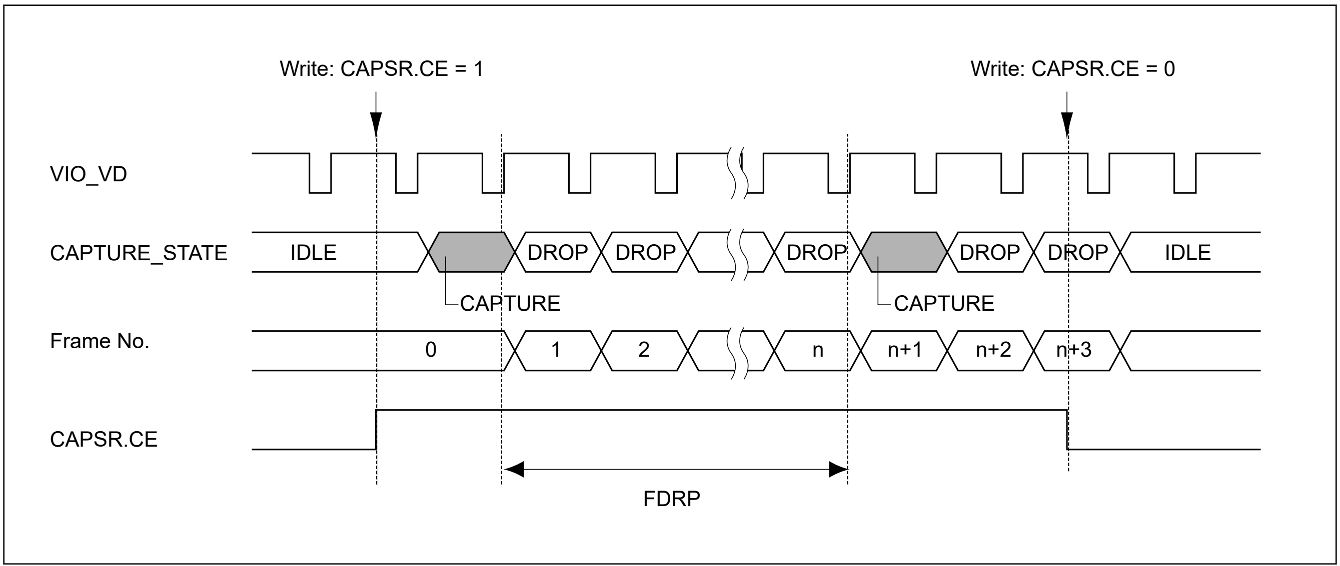


Figure 60.9 Setting of FDRP bits and frame drop timing

Table 60.5 Relationship between capture setting and frame drop interval unit

Input Mode	Captured Image	First Captured Image	Frame Drop Interval Unit	Capture Setting
Progressive	Frame	Frame immediately after capture start	Frame	A
Interlace	Both-field (2-VD capture)	Field immediately after capture start	2 fields (first capture field count)	B
				C
		Top field	2 fields (top-field count)	D
		Bottom field	2 fields (bottom-field count)	E
	One-field (1-VD capture)	Field immediately after capture start	First capture field	F
				G
		Top field	Top field	H
		Bottom field	Bottom field	I

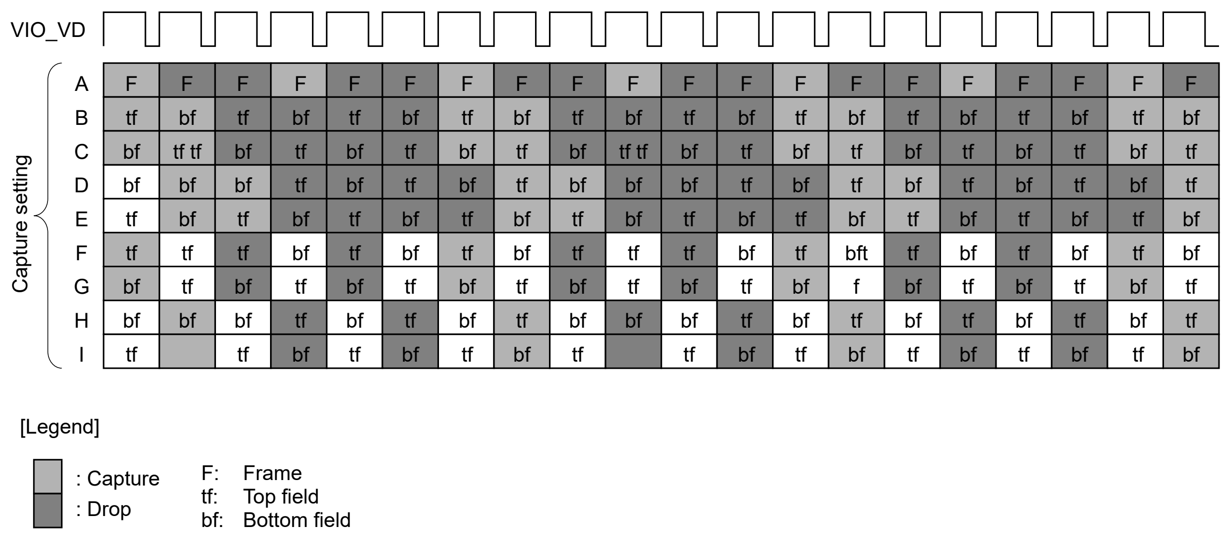


Figure 60.10 Image of frame drop timing for capture settings (FDRP[7:0] = 0x2)

60.2.3 CAMCR : Capture Interface Control Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	VDSEL	HDSEL	FLDSEL	DSEL	—	—	—	—	—	—	—	FLDPOL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DTIF	—	—	DTARY[1:0]		—	—	JPG[1:0]		—	—	VDPOL	HDPOL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	HDPOL	Sets the polarity for detection of the horizontal sync signal (HD) input from an external module. Figure 60.16 shows the relationship between the HD and HD interrupt when high-active is selected. Since an HD interrupt may occur when this bit value is modified, the HD bit in CETCR must always be cleared to 0 when this bit value is changed. In data enable fetch mode, set this bit to 0. 0: Horizontal sync signal (HD) from an external module is detected as high active 1: Horizontal sync signal (HD) from an external module is detected as low active	R/W
1	VDPOL	Sets the polarity for detection of the vertical sync signal (VD) input from an external module. Figure 60.14 and Figure 60.15 show the relationship between the VD, HD and VD interrupt when high-active is selected. Since a VD interrupt may occur when this bit value is modified, the VD bit in CETCR must always be cleared to 0 when this bit value is changed. In data enable fetch mode, this bit is not used and the sense for detection is always active high. 0: Vertical sync signal (VD) from an external module is detected as high active 1: Vertical sync signal (VD) from an external module is detected as low active	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
5:4	JPG[1:0]	These bits select the fetched data type. 0 0: Image capture mode (input data are separated into the luminance component data (Y) and the chrominance component data (CbCr) for output to the memory) 0 1: Data synchronous fetch mode (specified size of input data are output to the specified memory addresses in order of input and in synchronization with the sync signal) 1 0: Data enable fetch mode (input data are fetched with HD as an enable signal and output to the specified addresses in memory in order of input) 1 1: Setting prohibited	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
9:8	DTARY[1:0]	Set the input order of the luminance component and chrominance component. The order in which Y and CbCr are input from an external module differs among modules. The CEU supports the input orders shown in Figure 60.12. Set the corresponding value in these bits. In data fetch mode, set these bits to 00. 0 0: 8-bit interface: Image input data is fetched in the order of Cb_0, Y_0, Cr_0, and Y_1 16-bit interface: Image input data is fetched in the order of {Cb_0, Y_0} and {Cr_0, Y_1} 0 1: 8-bit interface: Image input data is fetched in the order of Cr_0, Y_0, Cb_0, and Y_1 16-bit interface: Image input data is fetched in the order of {Cr_0, Y_0} and {Cb_0, Y_1} 1 0: 8-bit interface: Image input data is fetched in the order of Y_0, Cb_0, Y_1, and Cr_0 16-bit interface: Image input data is fetched in the order of {Y_0, Cb_0} and {Y_1, Cr_0} 1 1: 8-bit interface: Image input data is fetched in the order of Y_0, Cr_0, Y_1, and Cb_0 16-bit interface: Image input data is fetched in the order of {Y_0, Cr_0} and {Y_1, Cb_0}	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	DTIF	Sets the digital image input pins from which data is to be captured. 0: Data input to 8-bit digital image input pins is captured 1: Data input to 16-bit digital image input pins is captured	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
16	FLDPOL	Sets the polarity of the field identification signal (FLD) from an external module. 0: When the FLD signal is high-active, the field is detected as the top field and when low-active, the field is detected as the bottom field. 1: When the FLD signal is low-active, the field is detected as the top field and when high-active, the field is detected as the bottom field.	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W
24	DSEL	Sets the edge for fetching the image data (D15 to D0) from an external module. 0: D15 to D0 are fetched at the rising edge of the camera clock. 1: D15 to D0 are fetched at the falling edge of the camera clock.	R/W
25	FLDSEL	Sets the edge for capturing FLD from an external module. 0: FLD is captured at the rising edge of the camera clock. 1: FLD is captured at the falling edge of the camera clock.	R/W
26	HDSEL	Sets the edge for capturing HD from an external module. 0: HD is captured at the rising edge of the camera clock. 1: HD is captured at the falling edge of the camera clock.	R/W
27	VDSEL	Sets the edge for capturing VD from an external module. 0: VD is captured at the rising edge of the camera clock. 1: VD is captured at the falling edge of the camera clock.	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CAMCR sets the capture interface.

The following items are set by CAMCR.

- Selection between the rising edge or falling edge of the camera clock for signal capture or data fetch operation
- Selection between image capture operation or data fetch operation
- Polarities of the vertical and horizontal sync signals
- Input order of image data components (Y, Cb, and Cr) (only for image capture mode)
- Selection of digital image input pins (8 bits or 16 bits)

- Polarity of the field identification signal

CAMCR must be set according to the module connected. In data fetch mode, set the DTARY bits to 0. Do not modify this register during operation. If this register is modified during operation, correct operation cannot be guaranteed. In addition, the IGRW bit (interrupt source) in CETCR is set to 1.

Note: After changing the setting of this register, do not start capture until at least 10 external input clock cycles have elapsed.

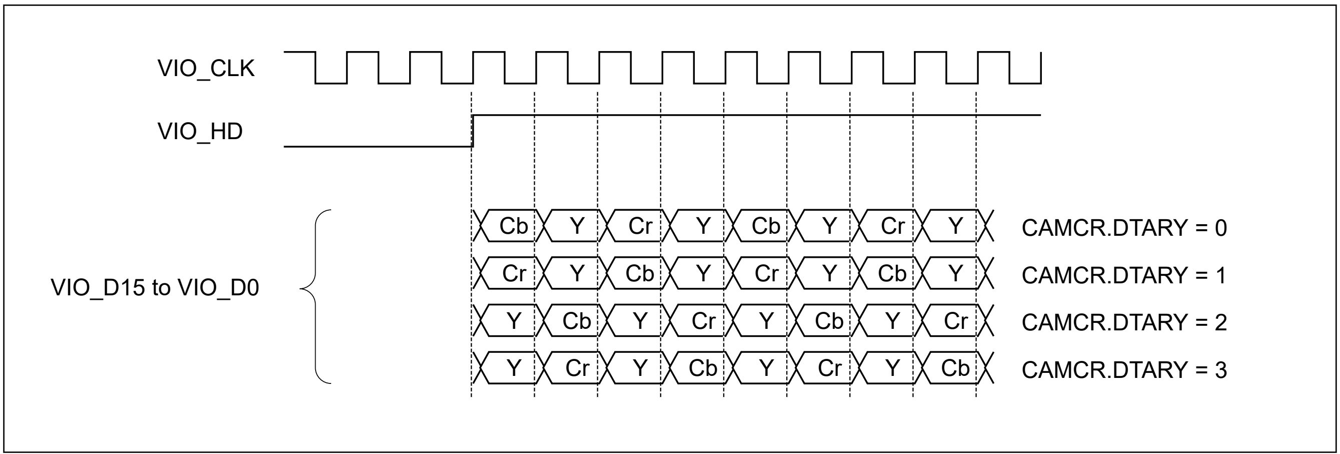


Figure 60.11 Input order of image data

The JPG bit in CAMCR selects whether digital image data is fetched or data such as JPEG is fetched. In addition, when data such as JPEG is fetched, select whether the specified amount of data is continuously fetched in synchronization with the sync signal or data is fetched while the horizontal sync signal is enabled.

In data enable fetch mode, one frame is defined as a period from the rising edge to the falling edge of the vertical sync signal (VD) for data fetching. The horizontal sync signal (HD) is enabled only when the VD is high and treated as an enable signal. Data input in the cycle in which the HD is asserted (high) is fetched and output to the memory continuously.

This module starts fetching data at the rising edge of the VD and stops fetching data at the falling edge of the VD in data enable fetch mode. Thus, if the VD remains high and does not go low, end processing does not start. In addition, if the VD remains high and the HD also remains asserted, data continues to be fetched.

Figure 60.12 and Figure 60.13 show the interface timing in data enable fetch mode.

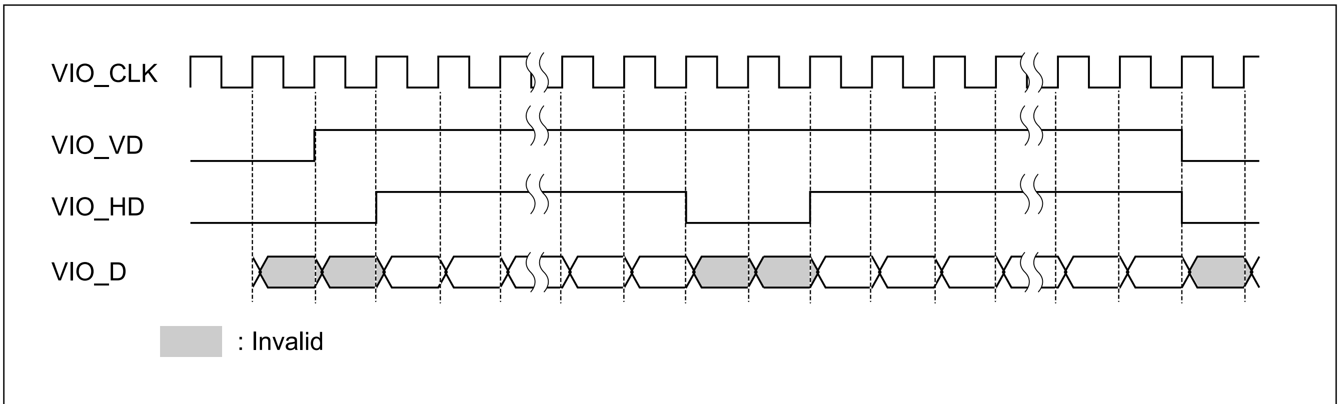


Figure 60.12 Data enable fetch timing (HD asserted (high) while VD is high)

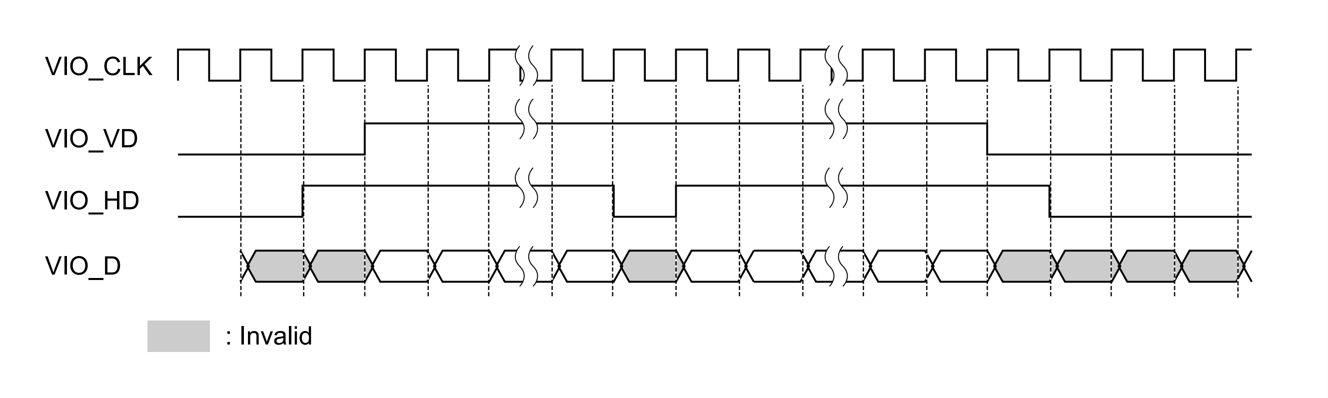


Figure 60.13 Data enable fetch timing (HD asserted (high) when VD is not high)

In data enable fetch mode, this module generates a VD interrupt in response to detection of the active level of VD. In image capture mode and data synchronous fetch mode, this module generates a VD interrupt in response to the first detection of the active level of HD following detection of the active level of VD. Note that, when VD and HD are asserted and detected at the same time, this module generates a VD interrupt at that time.

Figure 60.14 to Figure 60.16 show the relationships between the VIO_VD signal and the VD interrupt, the VIO_VD and VIO_HD signals and the VD interrupt, and the VIO_HD signal and the HD interrupt.

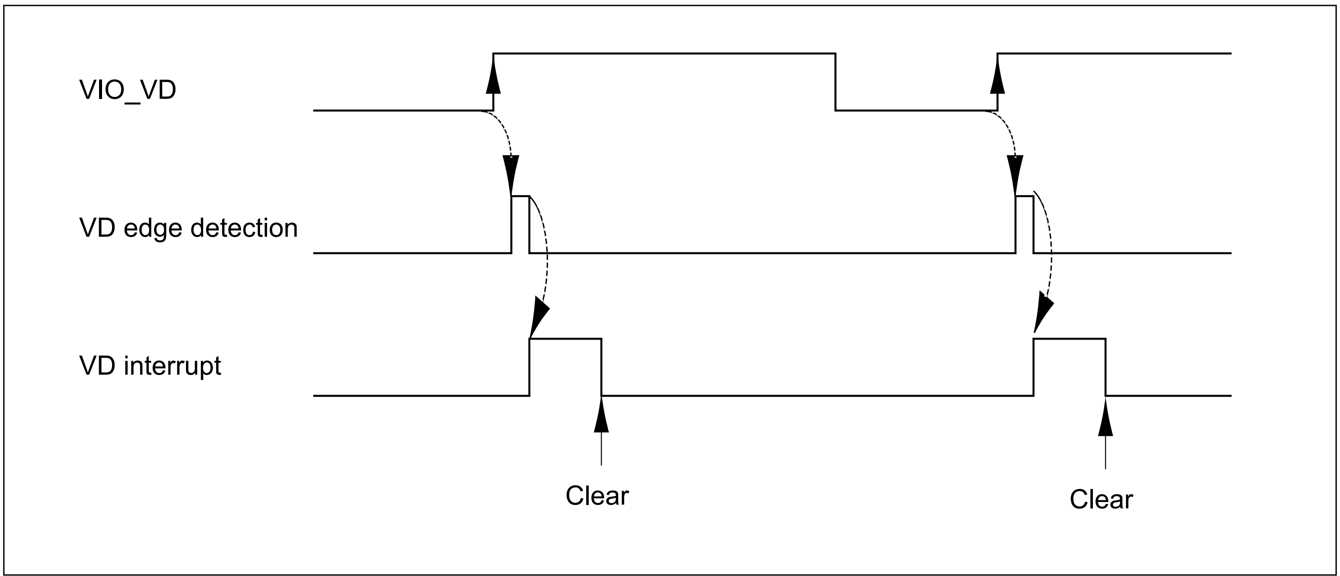


Figure 60.14 Relationship between VIO_VD and VD interrupt when VD is high-active (in data enable fetch mode)

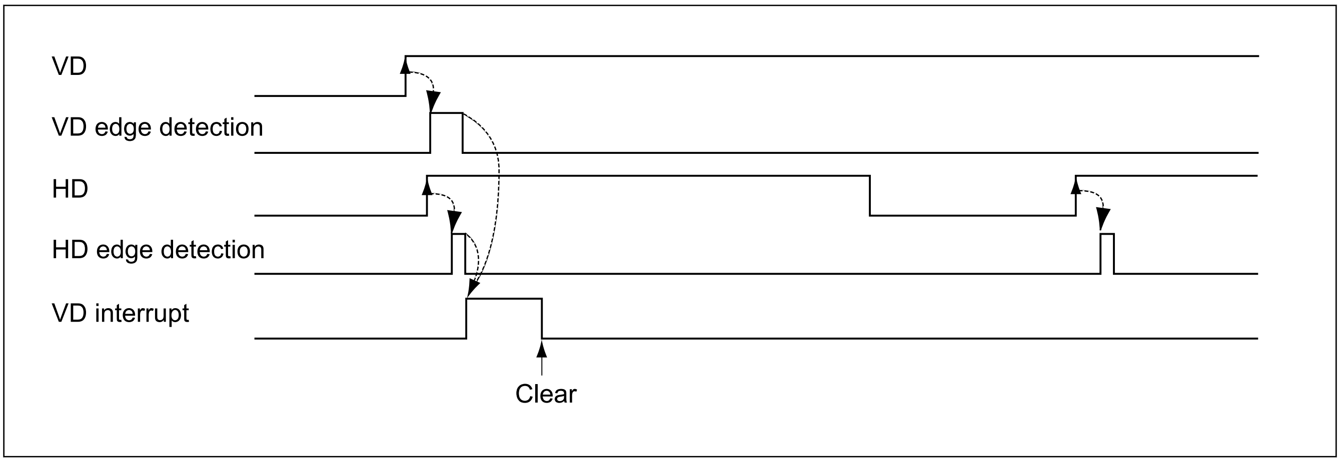


Figure 60.15 Relationship between the VIO_VD and VIO_HD signals and the VD Interrupt when VD and HD are high-active (in Image capture mode or data synchronous fetch mode)

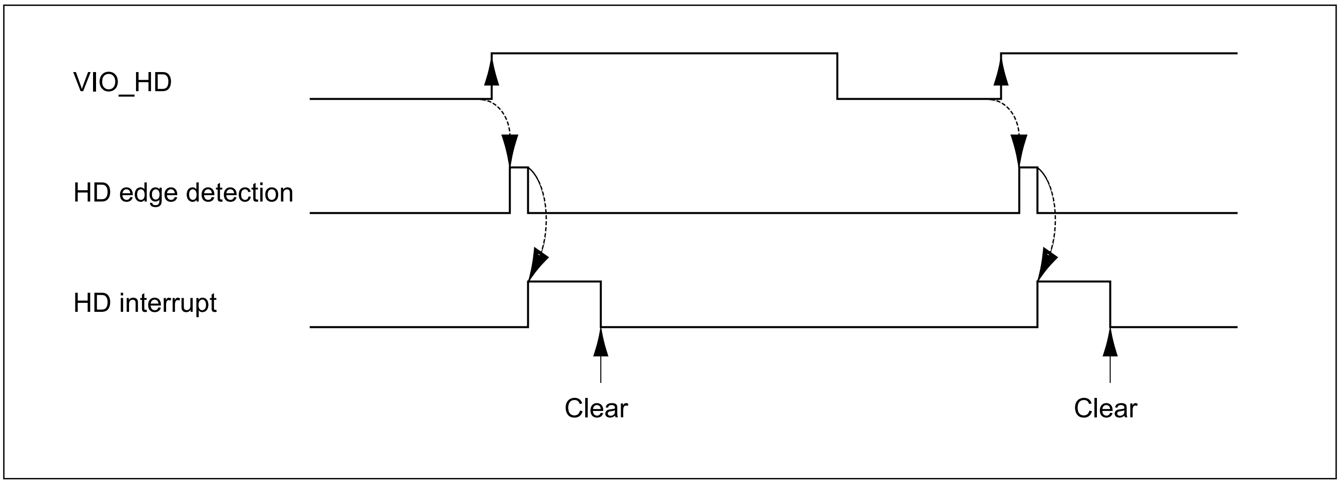


Figure 60.16 Relationship between VIO_HD and HD interrupt when HD is high-active

60.2.4 CMCYR : Capture Interface Cycle Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x000C

Bit position:	31	30	29 to 16	15	14	13 to 0
Bit field:	—	—	VCYL[13:0]	—	—	HCYL[13:0]
Value after reset:	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
13:0	HCYL[13:0]	Horizontal Cycle Count of External Module These bits set the number of HD cycles of an external module. The interrupt source bit IGHS in CETCR is set to 1 when the actual number of HD cycles input from the external module differs from this setting. Set these bits for detecting an illegal HD. When these bits are all cleared to 0, the interrupt source bit IGHS in CETCR is not set to 1. Though the interrupt source bit IGHS in CETCR may be set to 1 after the HDPOL bit (HD polarity) in CAMCR is changed, this interrupt should be ignored.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
29:16	VCYL[13:0]	Vertical HD Count of External Module These bits set the number of VD cycles of an external module with the number of HD inputs. The interrupt source bit IGVS in CETCR is set to 1 when the actual number of VD cycles input from the external module differs from this setting. Set these bits for detecting an illegal VD. When these bits are all cleared to 0, the interrupt source bit IGVS in CETCR is not set to 1. Though the interrupt source bit IGVS in CETCR may be set to 1 after the VDPOL bit (VD polarity) in CAMCR is changed, this interrupt should be ignored.	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CMCYR is used to detect an illegal VD and an illegal HD. For HD, the number of cycles from a rising edge of HD to the next rising edge is set (falling edges when low-active is selected for HD). For VD, the number of HD inputs from a rising edge of VD to the next rising edge is set (falling edges when low-active is selected for VD).

Do not modify this register during operation. If this register is modified during operation, correct operation cannot be guaranteed. In addition, the IGRW bit (interrupt source) in CETCR is set to 1.

Set 0 in all bits of this register, during data enable fetch mode.

Note: After changing the setting of this register, do not start capture until at least 10 external input clock cycles have elapsed.

60.2.5 CAMOR, CAMOR_x : Capture Interface Offset Register (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0010 (CAMOR)
0x1010 (CAMOR_B)
0x2010 (CAMOR_M)

Bit position: 31 27 16 12 0

Bit field:	—	—	—	—	VOFST[11:0]	—	—	—	HOFST[12:0]

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
12:0	HOFST[12:0]	Specify the capture start location in terms of the number of clock cycles from a horizontal sync signal (1-cycle units). The blanking period from a horizontal sync signal differs among external modules. Therefore, the horizontal capture start location must be specified by these bits in terms of external input clock cycles from a horizontal sync signal so that an image can be captured from the valid image area. This is similar in data synchronous fetch mode (see Figure 60.19). Some external modules output a horizontal sync signal as a data enable signal. In this case, there is no blanking period so these bits must be cleared to 0 (see Figure 60.20). Note: The first HD (horizontal sync signal) being input simultaneously or after the first VD (vertical sync signal) is the operating condition of the CEU. These inputs are affected by the polarities (set by the VDPOL and HDPOL bits in CAMCR).	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
27:16	VOFST[11:0]	Specify the capture start location in terms of the HD count from a vertical sync signal (1-HD units). The blanking period from a vertical sync signal differs among external modules. Therefore, the vertical capture start location must be specified by these bits in terms of the HD count from a vertical sync signal so that an image can be captured from the valid image area (see Figure 60.17). Some external modules output a vertical sync signal as a data enable signal. In this case, there is no blanking period so these bits must be cleared to 0 (see Figure 60.18).	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CAMOR sets the location to start capturing when capturing images.

Since the number of HD (horizontal sync signal) inputs from a VD (vertical sync signal) input to the start of a valid image period, and the number of clock cycles from an HD input to the start of a valid image period differ among external modules, these must be set in CAMOR. By setting a value greater than the valid image area, part of the image can be clipped for capture. In data synchronous fetch mode, the setting of this register becomes the number of cycles (HD count) up to the start of a valid data period.

This register is not used, during data enable fetch mode.

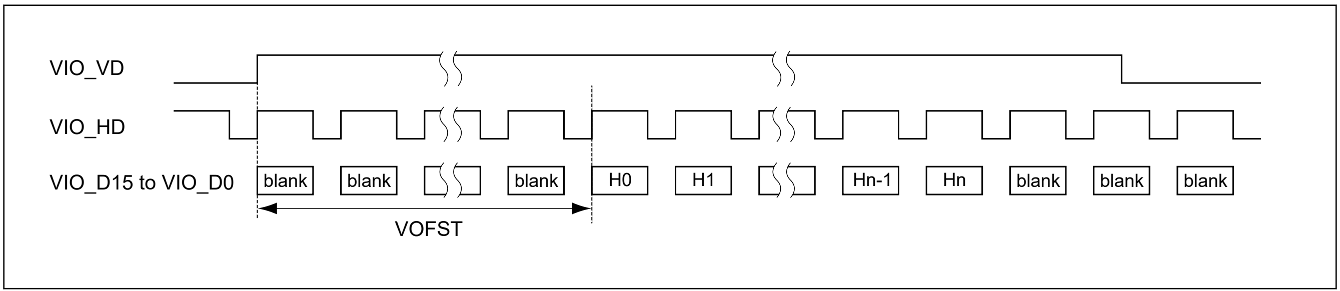


Figure 60.17 Vertical offset

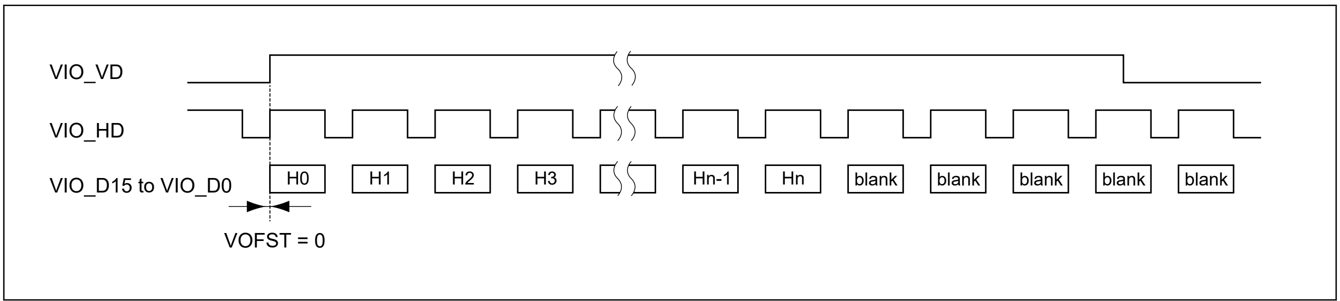


Figure 60.18 Timing when VD is data enable signal

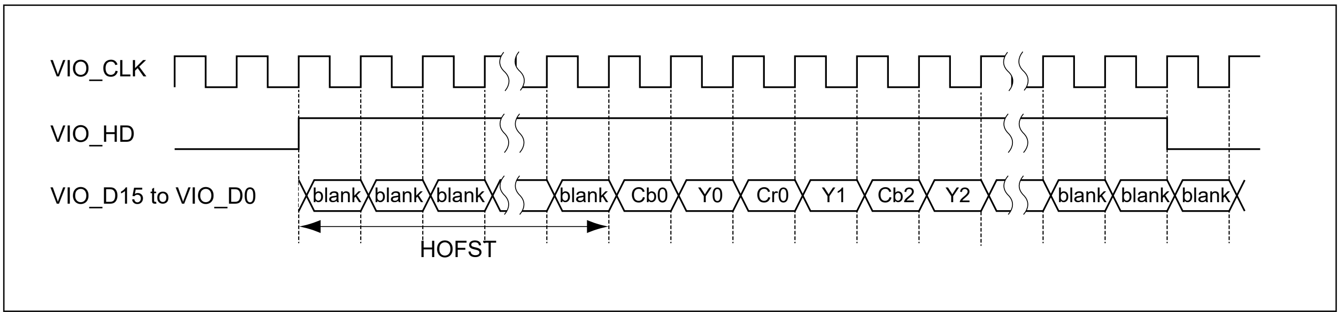


Figure 60.19 Horizontal offset

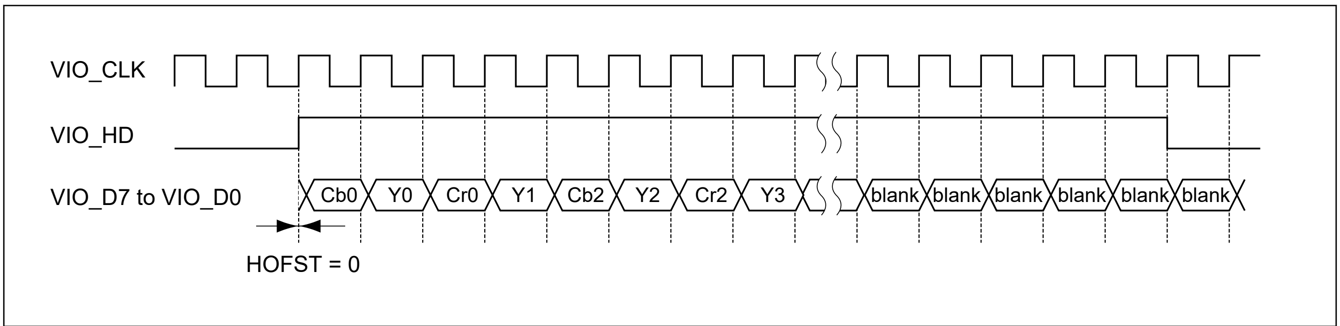


Figure 60.20 Timing when HD is data enable signal (8-bit interface)

60.2.6 CAPWR, CAPWR_x : Capture Interface Width Register (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0014 (CAPWR)
0x1014 (CAPWR_B)
0x2014 (CAPWR_M)

Bit position:	31	30	29	28	27–16	15	14	13	12–0
Bit field:	—	—	—	—	VWDTH[11:0]	—	—	—	HWDTH[12:0]
Value after reset:	0	0	0	0	0 0 0 0 0 0 0 0 0 0 0 0	0	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
12:0	HWDTH[12:0]	Specify the horizontal capture period. Specify the number of cycles to be captured from the location specified by the HOFST bits. Figure 60.22 shows the timing when the horizontal blanking period is 0. The CEU captures for only the number of cycles specified by these bits in the horizontal direction. Make a similar setting for data synchronous fetch. The maximum value to be set is as follows: ● 8-bit interface Image capture (16-cycle units) : 5,120 cycles (2,560 pixels) Data synchronous fetch (8-cycle units) : 2,560 cycles (2,560 bytes) ● 16-bit interface Image capture (8-cycle units) : 2,560 cycles (2,560 pixels) Data synchronous fetch (4-cycle units) : 1,280 cycles (2,560 bytes) The lower 2 bits should be written 0. Note: In data synchronous fetch mode, set CFSZR and CDWDR according to the values set in this register. For details, see the descriptions on CFSZR and CDWDR.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
27:16	VWDTH[11:0]	Specify the vertical capture period (4-HD units). Specify the number of lines (HD count) to be captured from the location specified by the VOFST bits in CAMOR. Figure 60.21 shows the timing when the vertical blanking period is 0. The CEU captures only the number of lines (HD count) specified by these bits in the vertical direction. Make a setting in the same way to obtain data synchronization. The maximum value to be set is 1,920 HD (5 megapixels). The value of writing are ignored on the lower 2 bits, so write 0 to the lower 2 bits.	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CAPWR sets the fetch (capture) cycle width when capturing images.

The cycle width unit differs according to the interface and the data type to be captured. For each setting unit, see Table 60.6.

This register is not used, during data enable fetch mode.

Table 60.6 Unit for setting fetch (capture) cycle width

	Vertical Direction		Horizontal Direction	
Interface	Image Capture	Data Synchronous Fetch	Image Capture	Data Synchronous Fetch
8-bit interface	4 HD	4 HD	16 cycles	8 cycles
16-bit interface	4 HD	4 HD	8 cycles	4 cycles

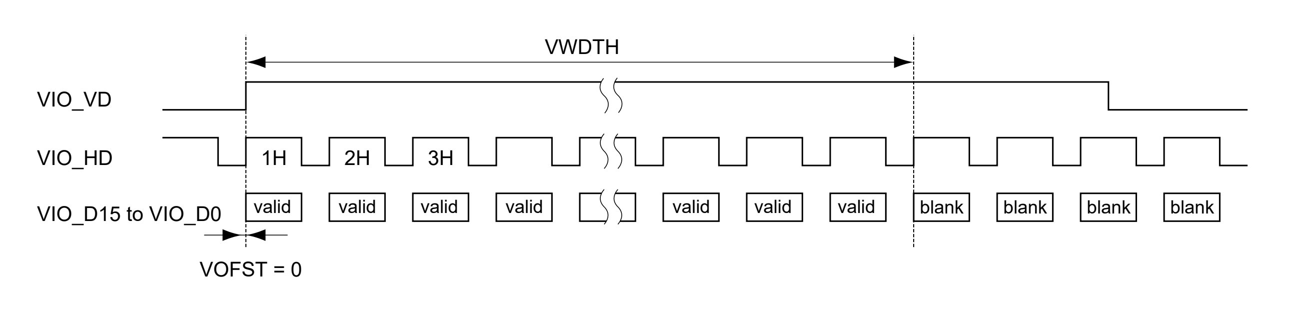


Figure 60.21 Vertical capture timing

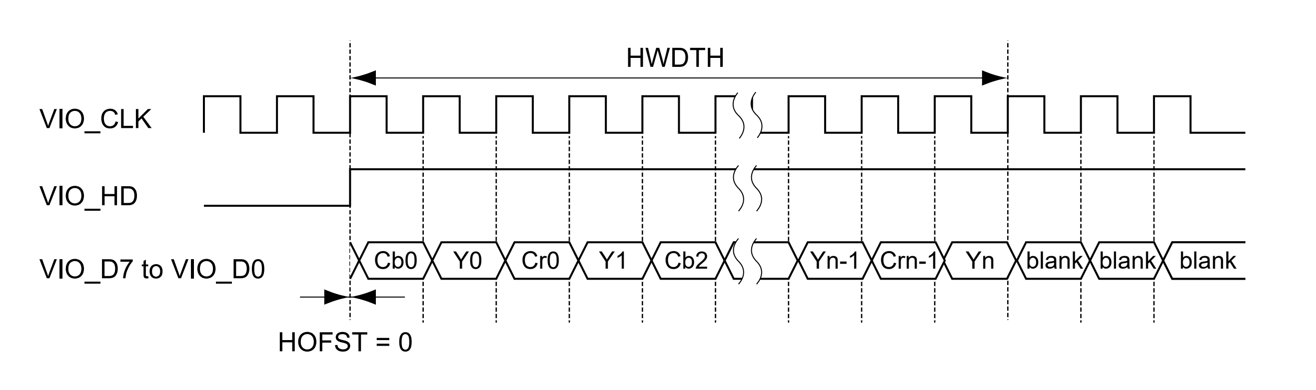


Figure 60.22 Horizontal capture timing (image capture with 8-bit interface)

60.2.7 CAIFR : Capture Interface Input Format Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0018

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	IFS	—	—	—	CIM	—	—	FCI[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	FCI[1:0]	Set the timing to start capturing. The timing to start capturing is set by specifying the image to be captured first. Set these bits to 00 when the input mode is progressive. 0 0: Capture starts from the VD input immediately after the CEU activation regardless of it being a top or bottom field 0 1: After the CEU activation, input of a top-field image is waited, and then capture starts from the top field 1 0: After the CEU activation, input of a bottom-field image is waited, and then capture starts from the bottom field 1 1: Setting prohibited	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
4	CIM	Sets the images to be captured. Clear this bit to 0 when the input mode for image capture is progressive (frame image) or when the input mode for image capture is interlace for continuous capture of both the top and bottom fields. Set this bit to 1 when the input mode for image capture is interlace for capture of only a one-field image. 0: Capture of frame image (1 VD) or both-field image (2 VD) 1: Capture of one-field image (1 VD)	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	IFS	Sets the input mode for capturing images. 0: Progressive 1: Interlace	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CAIFR sets the input mode (progressive or interlace) for capturing images, the images to be captured (frame, both-field, or one-field), the image from which capturing starts (top field or bottom field), etc. CAIFR is not used in data fetch mode.

Do not modify this register during operation. If this register is modified during operation, correct operation cannot be guaranteed. In addition, the IGRW bit (interrupt source) in CETCR is set to 1.

The items set by CAIFR are listed in Table 60.7.

Table 60.7 CAIFR Setting Items

Input Mode	IFS Bit	Captured Image	CIM Bit	Image to Start Capture	FCI Bits
Progressive	0	Frame	0	Frame immediately after activation	00
Interlace	1	Both-field (2-VD capture)	0	Field immediately after activation	00
				Top field	01
				Bottom field	10
				Setting prohibited	11
		One-field (1-VD capture)	1	Field immediately after activation	00
				Top field	01
				Bottom field	10
				Setting prohibited	11

In frame image capture and one-field image capture, a one-frame capture end interrupt occurs when capture for 1 VD finishes. In both-field image capture, a one-field capture end interrupt occurs when capture for 1 VD finishes and an oneframe capture end interrupt occurs when capture for 2 VD finishes. At this time, a one-field capture end interrupt occurs simultaneously with a one-frame capture end interrupt. Figure 60.23 shows the timing of a one-frame capture end interrupt and one-field capture end interrupt in both-field image capture.

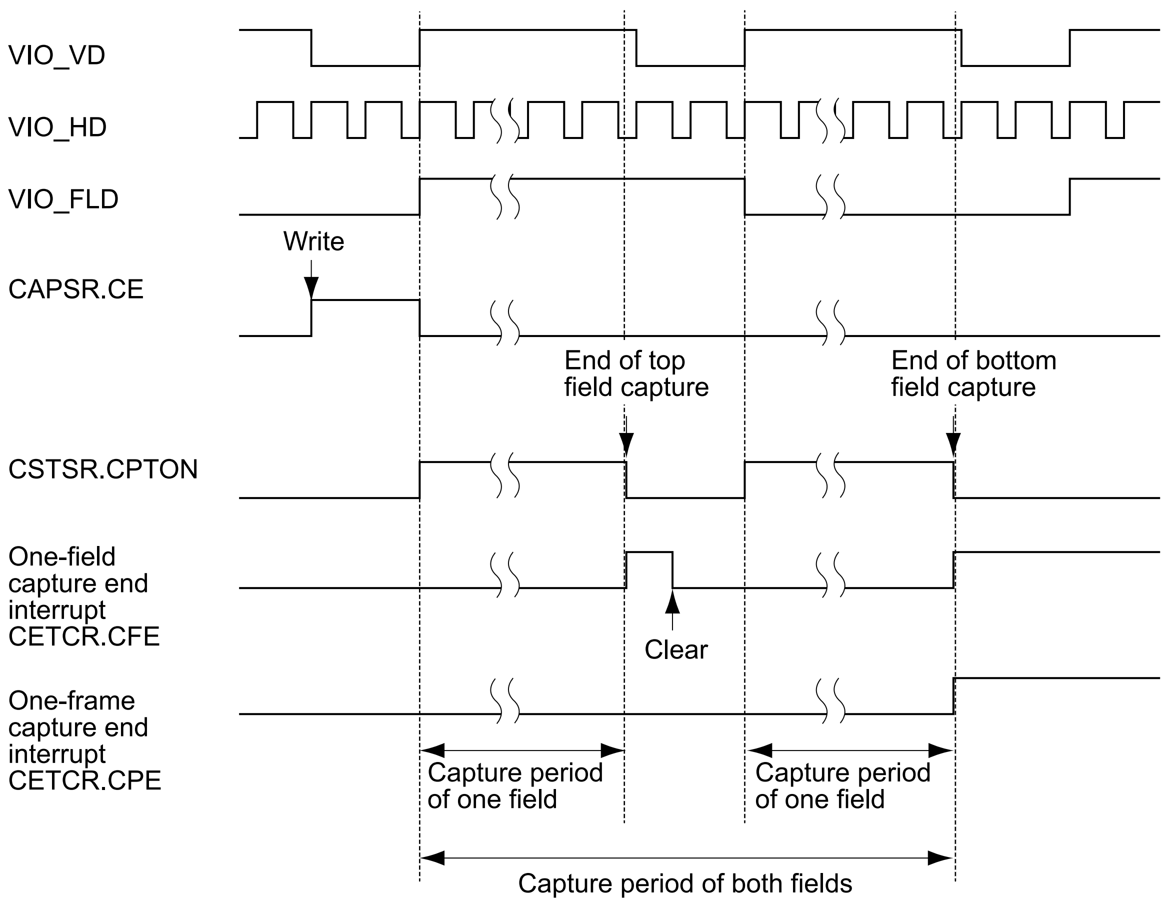


Figure 60.23 One-frame capture end interrupt and one-field capture end interrupt in both-field image capture

A captured frame image or captured one-field image is stored in the memory from the addresses set in CDAYR and CDACR (Figure 60.24). Captured both-field images are stored in different memory areas depending on whether it is a top-field or bottom-field image. A top-field image is stored in the memory from the addresses set in CDAYR and CDACR whereas a bottom-field image is stored in the memory from the addresses set in CDBYR and CDBCR (Figure 60.25).

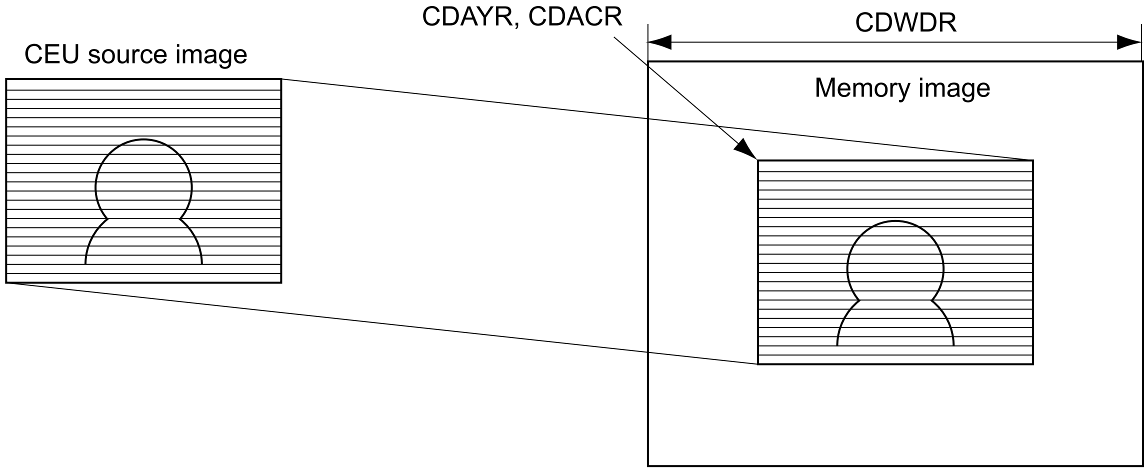


Figure 60.24 Image of storing captured frame image or captured one-field image in memory

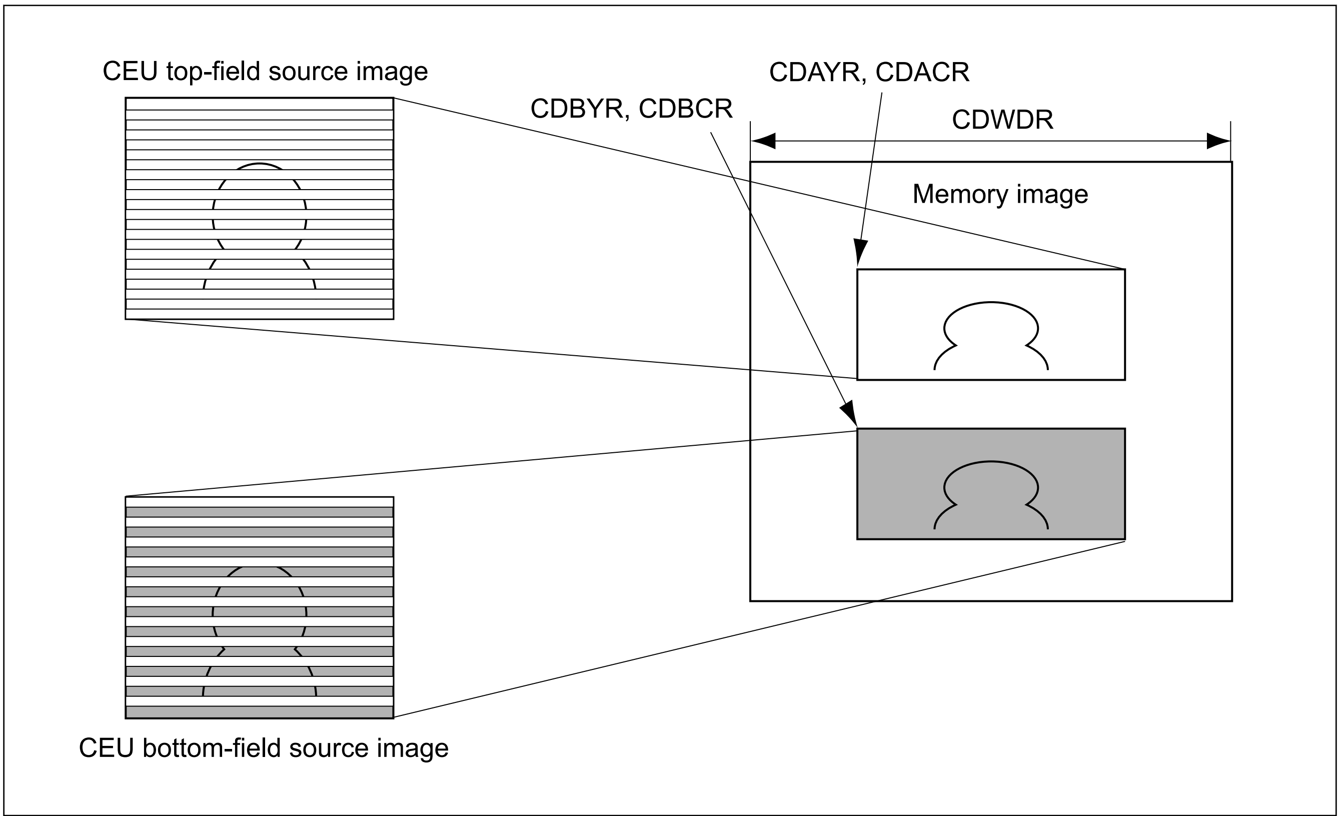


Figure 60.25 Image of storing captured both-field images in memory

If the FCI bits are set to 00b for continuous capture in interlace input mode, images are continuously captured for 2 VD with the first captured field as the reference in both-field image capture (Figure 60.26). In one-field image capture, only the first captured field is continuously captured for 1 VD (Figure 60.27).

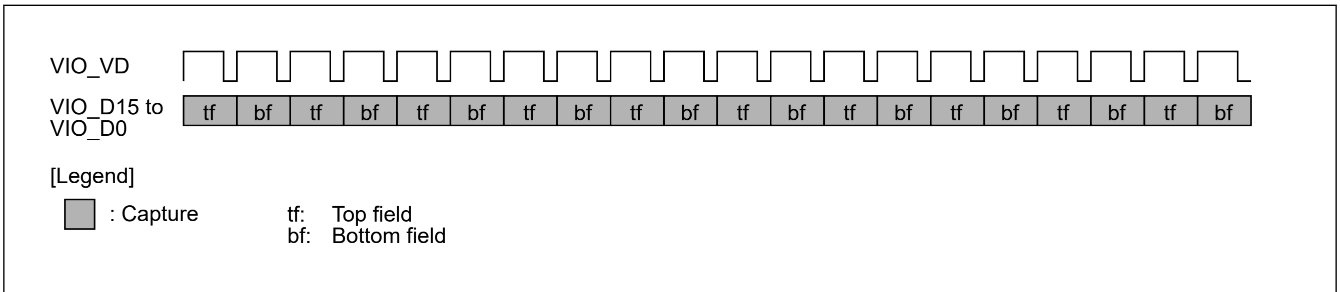


Figure 60.26 Continuous both-field capture in interlace mode (Image Immediately after Activation is Top Field (FCI[1:0] = 00b))

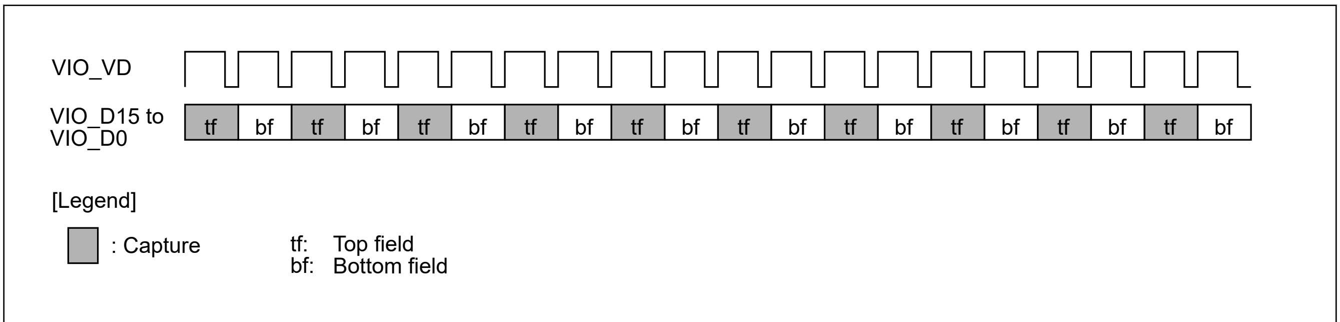


Figure 60.27 Continuous one-field capture in interlace mode (Image Immediately after Activation is Top Field (FCI[1:0] = 00b))

(1) Storage of Interlace Input as Frame Image

The CEU can store an interlace source image in the memory as a frame image. To store an interlace source image as a frame image, make the following register settings:

Input mode: Interlace (IFS = 1)

Capture image: Both-field (CIM = 0)

Image to start capture: Any setting other than the prohibited setting (FCI[1:0] = as desired)

Figure 60.28 shows a memory image of capturing both fields of an interlace input and storing it as a frame image in the memory. Set the start addresses of the memory destination for the captured top-field image in CDAYR and CDACR, and the start addresses of the memory destination for the captured bottom-field image in CDBYR and CDBCR. When storing an interlace image as a frame image in the memory, set the horizontal image size of the memory area in CDWDR with the top-field image and bottom-field image placed next to each other as shown in Figure 60.28. In addition, set the number of captured lines of the field image in the VWDTH bits in CAPWR.

A memory image of folding the horizontal image size of the memory area in Figure 60.28 at CDWDR/2 is shown in Figure 60.29. Setting the registers to form the image in Figure 60.28 enables an interlace image to be stored as a frame image in the memory as shown in Figure 60.29.

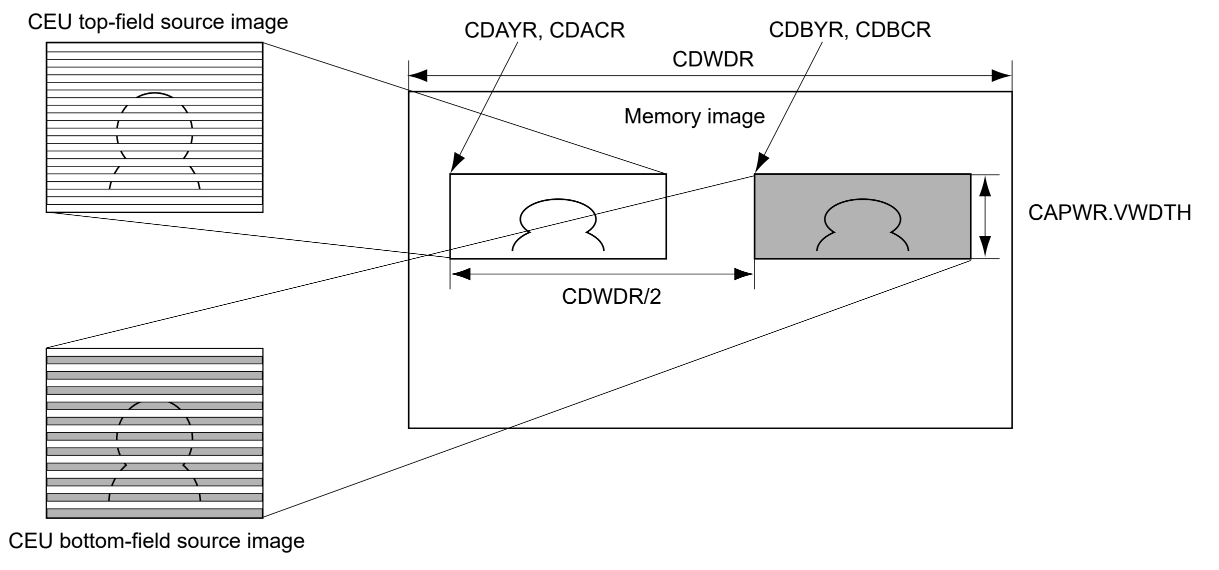


Figure 60.28 Image of storing captured both-fields of interlace input in memory

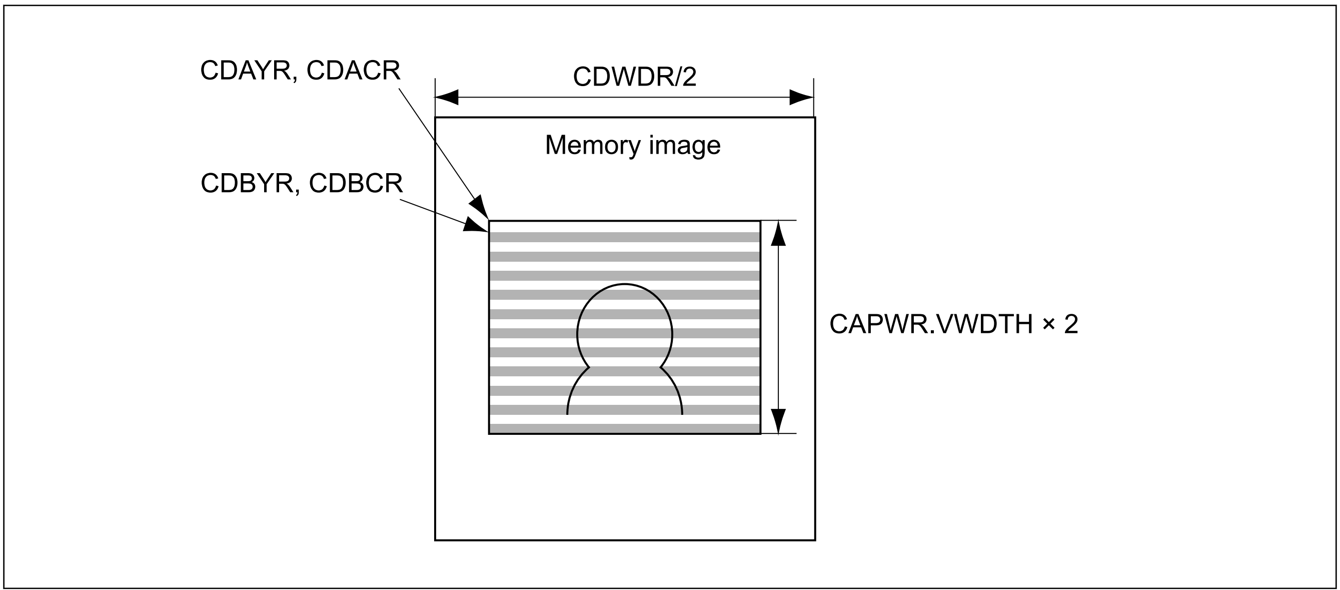


Figure 60.29 Image of storing Interlace input as frame image in memory

60.2.8 CRCNTR : CEU Register Control Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0028

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	RVS	—	—	RS	RC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RC	Specifies switching of the register plane used by the CEU in synchronization with VD. If the register plane is not switched, the register plane specified by the RS bit is used. 0: Uses the specified register plane in synchronization with VD 1: Switches the register plane in synchronization with VD	R/W
1	RS	Specifies which register plane is used by the CEU in synchronization with VD. The setting of this bit is valid only when RC = 0. 0: Uses plane A of the register 1: Uses plane B of the register	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	RVS	Sets the timing to switch the register plane in both-field capture. The setting of this bit is valid only when RC = 1 in both-field capture. 0: Switches the register plane every 2 VD 1: Switches the register plane every 1 VD	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CRCNTR controls switching of the planes of registers with a 2-plane configuration.

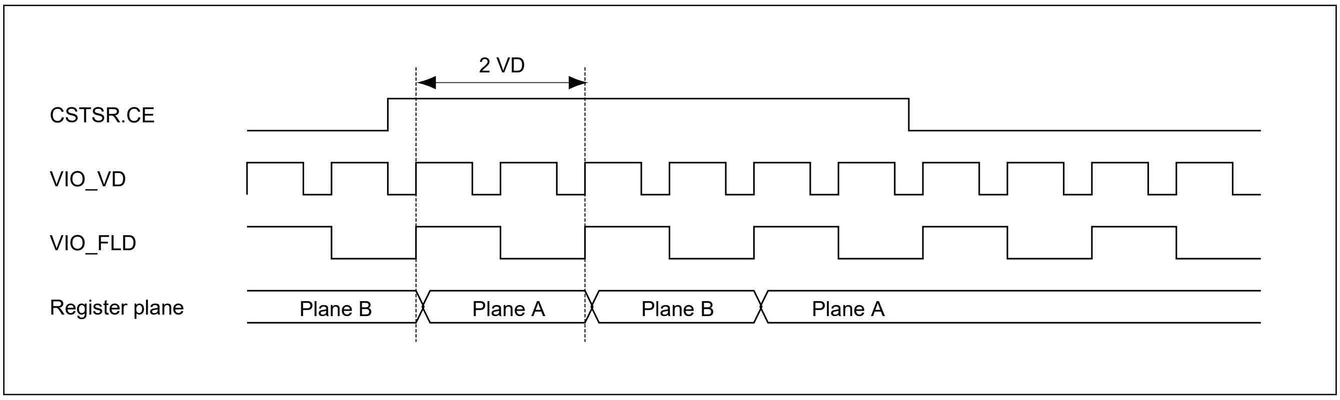


Figure 60.30 Timing for register plane switching when RVS = 0

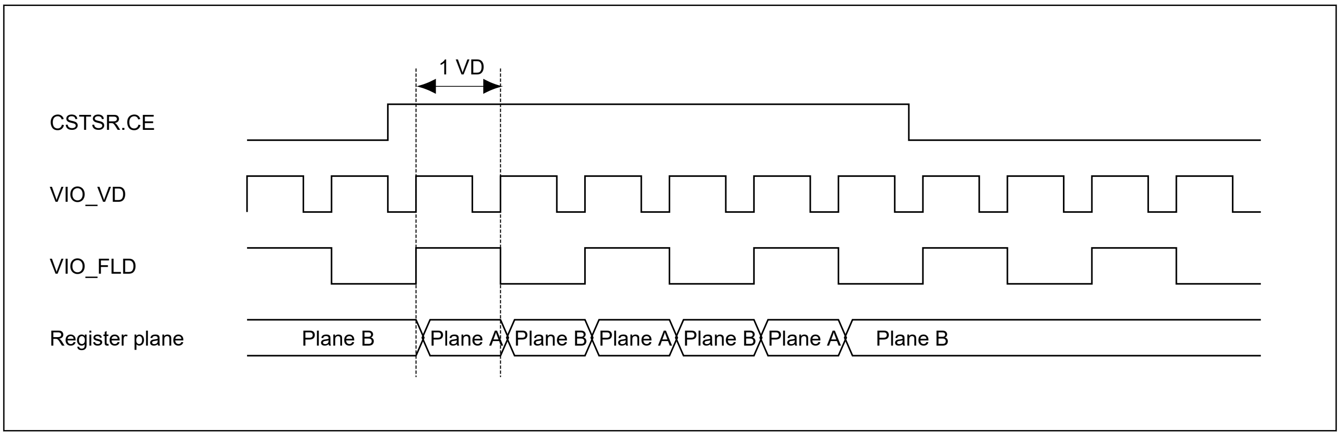


Figure 60.31 Timing for register plane switching when RVS = 1

60.2.9 CRCMPR : CEU Register Forcible Control Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x002C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RA	Indicates the register plane currently specified. This register value automatically changes in synchronization with VD for starting capture. To start capture with plane A of the register when a setting to switch the register plane in synchronization with VD has been made (CRCNTR.RC = 1), specify plane B of the register using this bit. 0: Specifies plane A of the register 1: Specifies plane B of the register	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CRCMPR forcibly controls switching of the planes of registers with a 2-plane configuration. Setting this register enables direct control of register plane switching.

Do not modify this register during operation. If this register is modified during operation, correct operation cannot be guaranteed. In addition, the IGRW bit (interrupt source) in CETCR is set to 1.

60.2.10 CFLCR, CFLCR_x : Capture Filter Control Register (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0030 (CFLCR)
0x1030 (CFLCR_B)
0x2030 (CFLCR_M)

Bit position:	31–28	27–16	15–12	11–0
Bit field:	VMANT[3:0]	VFRAC[11:0]	HMANT[3:0]	HFRAC[11:0]
Value after reset:	0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
11:0	HFRAC[11:0]	Fraction Part of Horizontal Scale-Down Factor The specifiable range is 0x000 to 0xFF8. The fraction of the scale-down factor that cannot be set with only the HMANT bits must be specified with these bits. The value of writing are ignored on the lower 3 bits, so write 0 to the lower 3 bits.	R/W
15:12	HMANT[3:0]	Mantissa Part of Horizontal Scale-Down Factor The specifiable range is 0x0 to 0xF. When 0x0 is set for the HMANT bits and 0x000 is set for the HFRAC bits, the scale-down filter is not used.	R/W
27:16	VFRAC[11:0]	Fraction Part of Vertical Scale-Down Factor The specifiable range is 0x000 to 0xFF8. The fraction of the scale-down factor that cannot be set with only the VMANT bits must be set with these bits. The value of writing are ignored on the lower 3 bits, so write 0 to the lower 3 bits.	R/W
31:28	VMANT[3:0]	Mantissa Part of Vertical Scale-Down Factor The specifiable range is 0x0 to 0xF. When 0x0 is set for the VMANT bits and 0x000 is set for the VFRAC bits, the scale-down filter is not used.	R/W

Note: S-TYPE-3, P-TYPE-3

CFLCR sets the scale-down factor for the filter to scale images down.

The CEU has an image scale-down filter which can be used to scale down the captured images before storing them in the memory. Set CFLCR to 0 when not performing scale-down (same size output). If a value other than 0 is set in CFLCR, scale-down is performed. In data fetch mode, set CFLCR to 0.

When handling an interlace source image as a frame image, set CFLCR to 0 not to use the filter.

An image scale-down filter is installed in the CEU, and the captured images can be scaled down and stored in the memory.

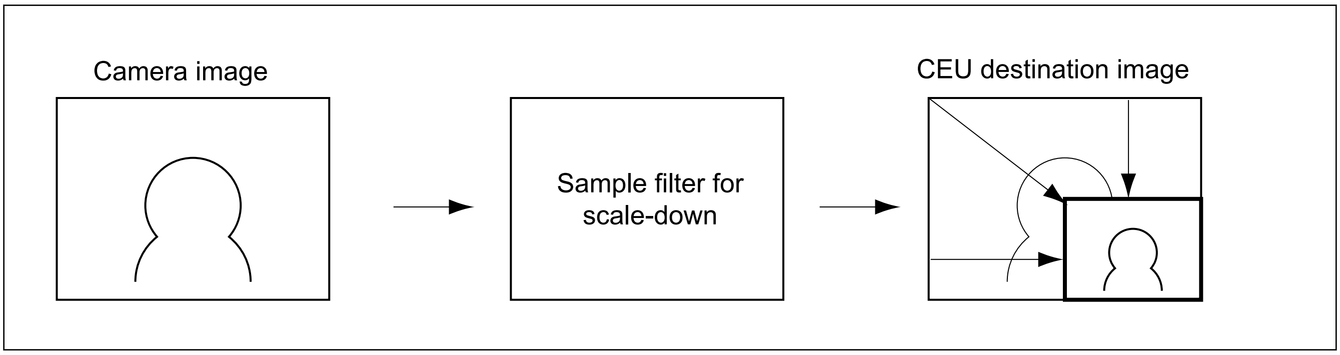


Figure 60.32 Scale-down of captured image

The formulas for obtaining the MANT (VMANT or HMANT) and FRAC (VFRAC or HFRAC) values from the input pixel count and output pixel count of the filter are shown below. Set the MANT and FRAC bits in order to obtain the desired output pixel count from the number of pixels input to the CEU.

First, calculate preliminary MANT and FRAC values. The parameters needed for calculation are defined as follows:

$$\alpha = \mathrm{MANT} \times 4096 + \mathrm{FRAC}$$ …Formula 1

$$\mathrm{SCL\ (scaling\ factor)} = \frac{4096}{\alpha}$$ …Formula 2

Assuming an operator $\lfloor x \rfloor$ which discards fractions of an integer x, the MANT and FRAC values can be temporarily set as follows, according to formula 1 and formula 2.

$$\mathrm{MANT} = \left\lfloor \frac{1}{\mathrm{SCL}} \right\rfloor, \mathrm{FRAC} = \left\lfloor 512 \times \left(\frac{1}{\mathrm{SCL}} - \mathrm{MANT} \right) \right\rfloor \times 8$$

Here, the scaled-down filter output size ($\mathrm{SIZE_D}$) can be calculated using the input image size S_{in} (8-bit interface: half of the CAPWR setting, 16-bit interface: CAPWR setting) in the following formula.

$$\mathrm{SIZE}_D = \left\lfloor 1 + \left(\left\lfloor \frac{1}{2} + \frac{S_{in} - 1}{\mathrm{MANT_{pre}}} \right\rfloor - 1 \right) \times \frac{\mathrm{MANT_{pre}} \times 4096}{\alpha} \right\rfloor$$ …Formula 3

$$\begin{bmatrix} \mathrm{MANT_{pre}} = 1 \rightarrow (0 \leq \mathrm{MANT} < 2) \\ \mathrm{MANT_{pre}} = 2 \rightarrow (2 \leq \mathrm{MANT} < 4) \\ \mathrm{MANT_{pre}} = 4 \rightarrow (4 \leq \mathrm{MANT} < 8) \\ \mathrm{MANT_{pre}} = 8 \rightarrow (8 \leq \mathrm{MANT}) \end{bmatrix}$$

The number of output pixels can be obtained by substituting the temporarily calculated MANT, FRAC, and input image size into these formulas. If the calculated number of output pixels is smaller than the number of output pixels used to obtain the preliminary MANT and FRAC values, recalculate with a smaller FRAC (α) value, and set the MANT and FRAC values in this register so that a pixel value greater than the desired number of output pixels can be obtained.

Example: Scale down 640 pixels to 480 pixels

SCL = 480/640 = 3/4, and the preliminary settings of MANT = 1, MANTpre = 1, and FRAC = 0x550 are made. Substituting these in the following formula results in an output pixel count of 479.

$$\mathrm{SIZE}_D = \left\lfloor 1 + \left(\left\lfloor \frac{1}{2} + \frac{S_{in} - 1}{\mathrm{MANT_{pre}}} \right\rfloor - 1 \right) \times \frac{\mathrm{MANT_{pre}} \times 4096}{\alpha} \right\rfloor$$ …Formula 3

Since this output pixel count is smaller than the desired output pixel count of 480, the formula is recalculated with a FRAC value of 0x548, a value eight less than the previous time. The obtained result of output pixel count = 480 is equal to the desired output pixel count of 480, so this register is set as MANT = 1 and FRAC = 0x548.

Table 60.8 Setting examples for each scale-down filter factor

Scale-down factor	FRAC		MANT	Input pixel count	Output pixel count	Clipping size (CFSZR)
	Decimal	Hexadecimal				
7/8	576	0x240	1	640	560	560
3/4	1352	0x548	1	640	480	480
5/8	2448	0x990	1	640	400	400
1/2	0	0x0	2	640	320	320
3/8	2728	0xAA8	2	640	240	240
1/3	0.0	0x0	3	640	213	212
1/4	0.0	0x0	4	640	160	160
1/5	0.0	0x0	5	640	128	128
1/6	0.0	0x0	6	640	107	104
1/7	0.0	0x0	7	640	91	88
1/8	0.0	0x0	8	640	80	80
1/16	4088	0xFF8	15	640	40	40

Note: This scale-down filter uses a VGA-size line memory for scale-down. Therefore, when an image larger than the VGA size is input for scale-down, settings must be made so that the output image size is equal to or larger than the SubQCIF size and equal to or smaller than the VGA size. When an image is not scaled down (same size output), this restriction does not apply.

60.2.11 CFSZR, CFSZR_x : Capture Filter Size Clip Register (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0034 (CFSZR)
0x1034 (CFSZR_B)
0x2034 (CFSZR_M)

Bit position:	31	30	29	28	27–16	15	14	13	12	11–0
Bit field:	—	—	—	—	VFCLP[11:0]	—	—	—	—	HFCLP[11:0]
Value after reset:	0	0	0	0	0 0 0 0 0 0 0 0 0 0 0 0	0	0	0	0	0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
11:0	HFCLP[11:0]	Specify the horizontal clipping value of the filter output size (8-pixel units). The lower 3 bits should be written 0.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
27:16	VFCLP[11:0]	Specify the vertical clipping value of the filter output size (4-pixel units). The value of writing are ignored on the lower 2 bits, so write 0 to the lower 2 bits.	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CFSZR sets the clipping size for fine adjustment of the image size output from the filter, and must be set in combination with CFLCR. When clipping the output size of the filter, set the clipping size as a number of pixels, and the setting unit should be four pixels. CFSZR must be set even when scale-down is not performed (same size output).

This register is not used, during data enable fetch mode.

In data synchronous fetch mode, set CFSZR according to the setting of CAPWR.

The scale-down filter in the CEU may output an odd number of pixels or lines depending on the settings. To adjust the output size of the filter, the CEU clips the destination image by using the number of pixels specified in CFSZR, as shown in Figure 60.33. The clipping size must be specified vertically in 4-pixel units and horizontally in 8-pixel units.

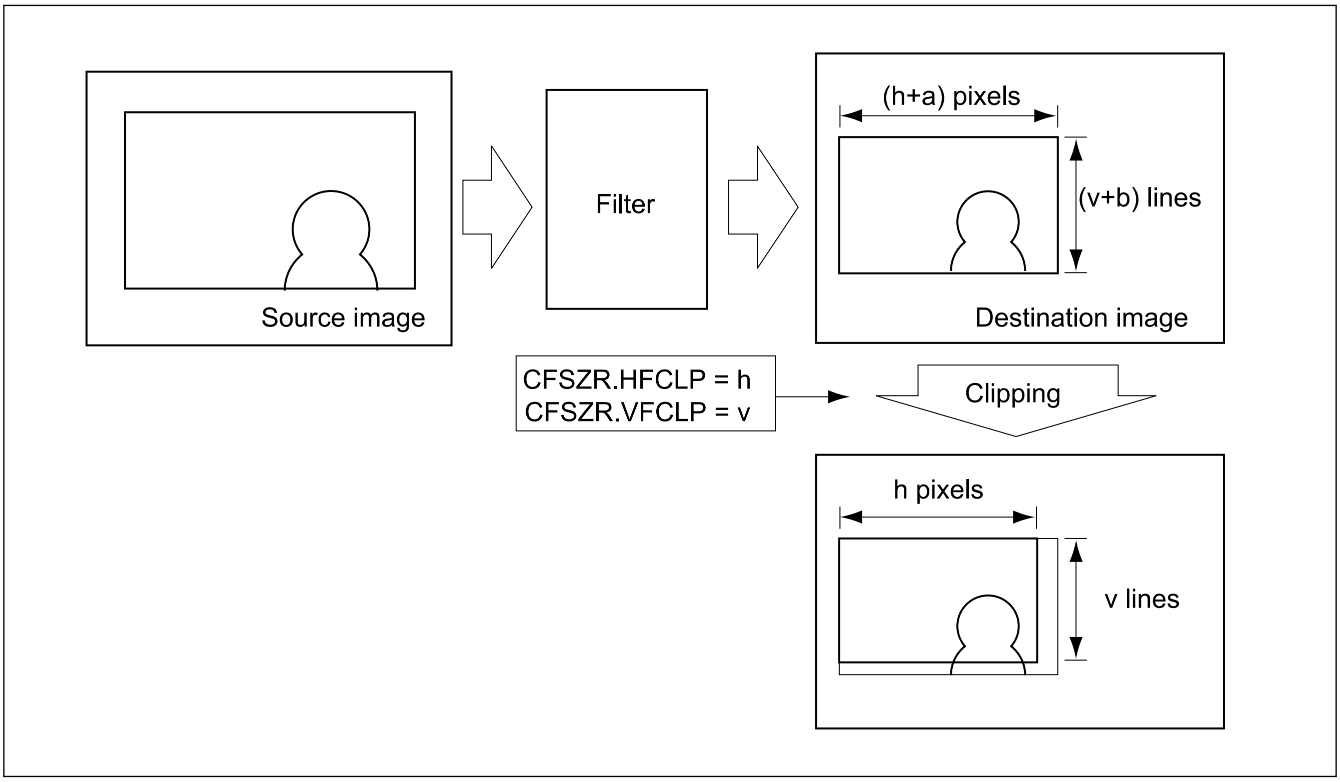


Figure 60.33 Clipping of image output from filter

The pixels to be clipped are counted from the top-left corner of a display. The pixels located to the right of the specified number of pixels or below the specified number of lines are discarded by the clipping function. If the number of pixels specified in CFSZR is larger than that output from the filter, correct operation cannot be guaranteed. To avoid this, the clipping size specified in CFSZR must be equal to or smaller than the number of pixels output from the filter.

Note: In data synchronous fetch mode, the following settings are required. Data cannot be fetched correctly unless the following settings are made.

- 8-bit interface:
  VFCLP = CAPWR.VWDTH
  HFCLP = CAPWR.HWDTH/2
- 16-bit interface:
  VFCLP = CAPWR.VWDTH
  HFCLP = CAPWR.HWDTH

60.2.12 CDWDR, CDWDR_x : Capture Destination Width Register (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0038 (CDWDR)
0x1038 (CDWDR_B)
0x2038 (CDWDR_M)

Bit position:	31																			12												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CHDW[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
12:0	CHDW[12:0]	Specify the horizontal image size in the memory area where the captured image is to be stored (8-byte units). The image data captured by the CEU is stored in the memory. In data synchronous fetch mode, set as follows: 8-bit interface: CHDW = CAPWR.HWDTH 16-bit interface: CHDW = CAPWR.HWDTH × 2 The lower 3 bits should be written 0.	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CDWDR sets the horizontal image size in the memory area where the captured image is to be output in 8-byte units (8-pixel units).

In data synchronous fetch mode, set CDWDR according to the setting of CAPWR.

This register is not used, during data enable fetch mode.

60.2.13 CDAYR, CDAYR_x : Capture Data Address Y Register (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x003C (CDAYR)
0x103C (CDAYR_B)
0x203C (CDAYR_M)

Bit position:	31	0
Bit field:	CAYR[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0

Bit	Symbol	Function	R/W
31:0	CAYR[31:0]	Capture Data Address Y ● Frame image capture: These bits set the address for storing the luminance component data of the captured data (8-pixel units). ● One-field image capture: These bits set the address for storing the luminance component data of the captured data (8-pixel units). ● Both-field image capture: These bits set the address for storing the luminance component data of the captured top-field data (8-pixel units). ● Data fetch: These bits set the address for storing data (8-byte units). ● Data enable fetch bundle write: These bits set the address for storing data (32-byte units). The lower 3 bits should be written 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CDAYR specifies the address where the luminance component of the captured data is to be stored in frame image capture or one-field image capture, the address where the luminance component of the captured top field is to be stored in both-field image capture, and the address where the fetched data is to be stored in data fetch. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. In frame image capture or one-field image capture, set the start address of the memory area where the luminance component of the captured data is to be stored by CDAYR. In both-field image capture, set the start address of the memory area where the luminance component of the captured top-field image is to be stored by CDAYR. In data fetch, set the start address of the memory area where data is to be stored by CDAYR.

Because the address must be specified in 32 bits, the address set by CDAYR must be in longword units.

Set the address of the starting point of the memory area where the fetched data is to be stored in this register, as shown in Figure 60.34.

- Frame image capture: Set the address of the starting point of the memory area where the luminance component of the captured image is to be stored.
- One-field image capture: Set the address of the starting point of the memory area where the luminance component of the captured image is to be stored.

- Both-field image capture: Set the address of the starting point of the memory area where the luminance component of the captured top-field image is to be stored.
- Data fetch: Set the address of the starting point of the memory area where the fetched data is to be stored. In data fetch mode, the data is simply stuffed in order from the start address so the end address becomes as follows:
  End address = CDAYR + number of fetched bytes
- Data enable fetch bundle write: Set the address in 32-byte units.

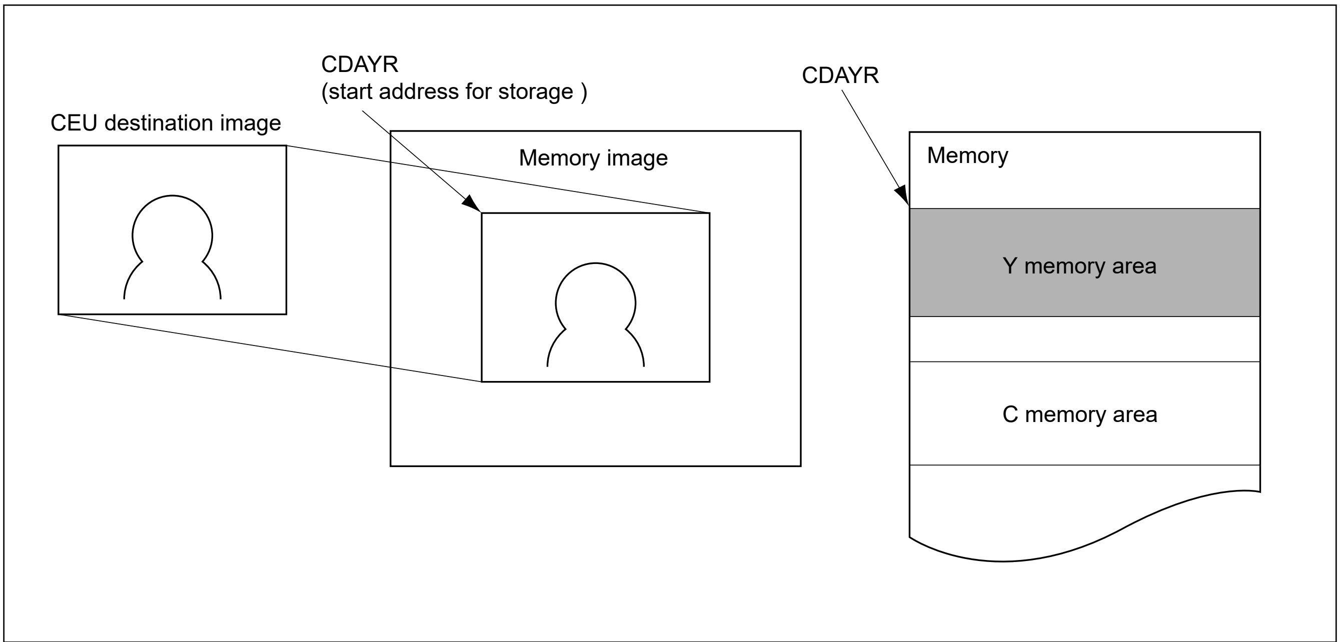


Figure 60.34 Relationship between captured image and luminance component memory area

60.2.14 CDACR, CDACR_x : Capture Data Address C Register (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0040 (CDACR)
0x1040 (CDACR_B)
0x2040 (CDACR_M)

Bit position:	31 … 0
Bit field:	CACR[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	CACR[31:0]	Capture Data Address C ● Frame image capture: These bits set the address for storing the chrominance component data of the captured data (8-pixel units). ● One-field image capture: These bits set the address for storing the chrominance component data of the captured data (8-pixel units). ● Both-field image capture: These bits set the address for storing the chrominance component data of the captured top-field data (8-pixel units). The lower 3 bits should be written 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CDACR specifies the address where the chrominance component of the captured data is to be stored in frame image capture or one-field image capture, and the address where the chrominance component of the captured top field is to be stored in both-field image capture. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. In frame image capture or onefield image capture, set the start address of the memory area where the chrominance component of the captured data is to be stored by

CDACR. In both-field image capture, set the start address of the memory area where the chrominance component of the captured top-field image is to be stored by CDACR. CDACR is not used in data fetch.

Because the address must be specified in 32 bits, the address set by CDACR must be in longword units.

Set the address of the starting point of the memory area where the chrominance component of the captured image is to be stored in this register, as shown in Figure 60.35. The chrominance component has an output data format like that in Figure 60.36, and is saved in the memory in this format.

- Frame image capture: Set the address of the starting point of the memory area where the chrominance component of the captured image is to be stored.
- One-field image capture: Set the address of the starting point of the memory area where the chrominance component of the captured image is to be stored.
- Both-field image capture: Set the address of the starting point of the memory area where the chrominance component of the captured top-field image is to be stored.

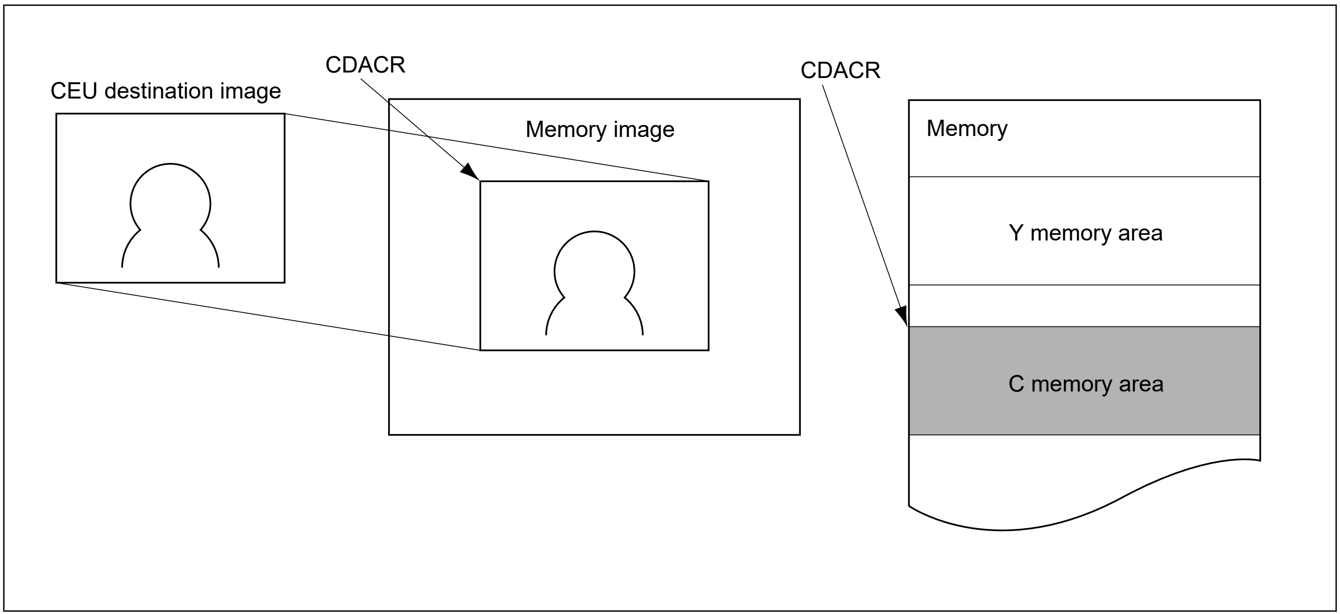


Figure 60.35 Relationship between captured image and chrominance component memory area

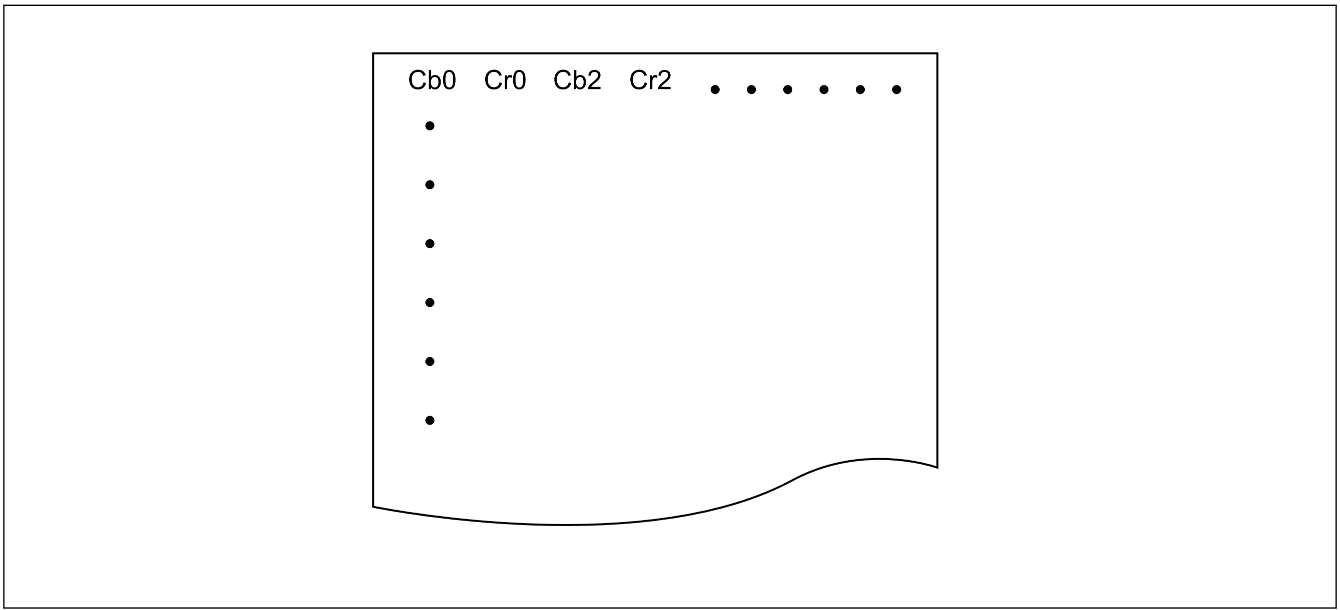


Figure 60.36 Image of storing chrominance components in memory

60.2.15 CDBYR, CDBYR_x : Capture Data Bottom-Field Address Y Register (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0044 (CDBYR)
0x1044 (CDBYR_B)
0x2044 (CDBYR_M)

Bit position:	31 ... 0
Bit field:	CBYR[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	CBYR[31:0]	Set the address for storing the luminance component data of the captured bottom-field data (8-pixel units). The lower 3 bits should be written 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CDBYR specifies the address where the luminance component of the captured bottom-field data is to be stored in both-field image capture. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. Set the start address of the memory area where the luminance component of the captured bottom-field image is to be stored by CDBYR. CDBYR is not used in frame image capture, one-field image capture, or data fetch.

Because the address must be specified in 32 bits, the address set by CDBYR must be in longword units.

Set the address of the starting point of the memory area where the luminance component of the captured bottom-field image is to be stored in this register, as shown in Figure 60.37.

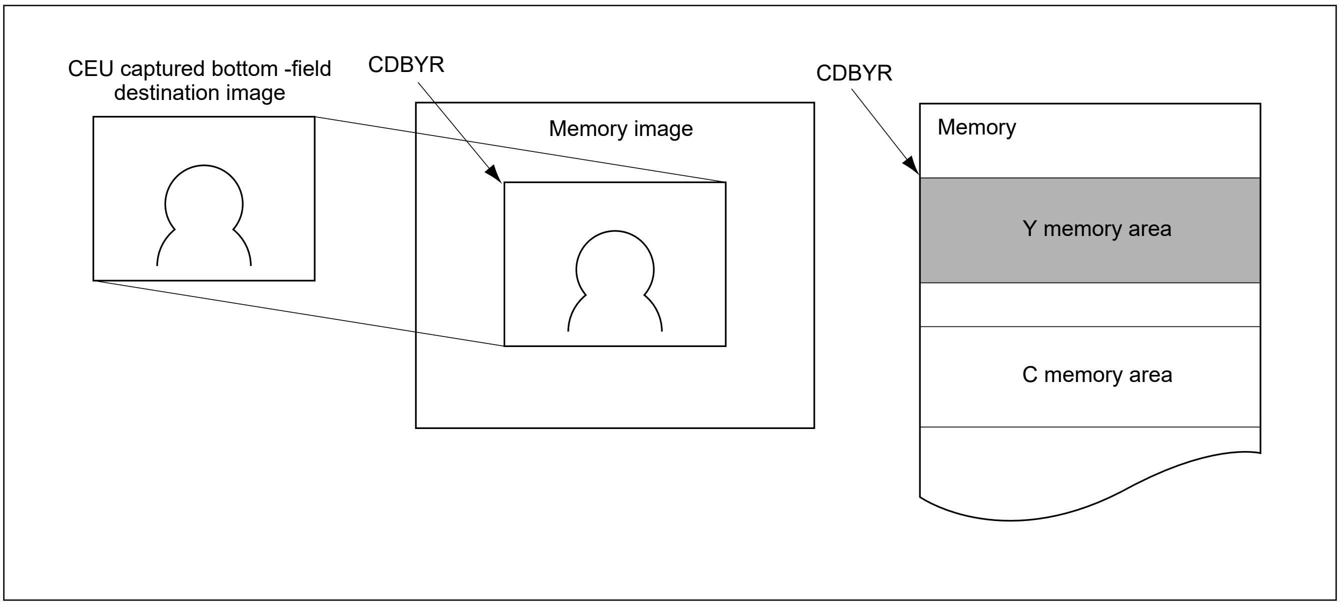


Figure 60.37 Relationship between captured bottom-field image and luminance component memory area

60.2.16 CDBCR, CDBCR_x : Capture Data Bottom-Field Address C Register (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0048 (CDBCR)
0x1048 (CDBCR_B)
0x2048 (CDBCR_M)

Bit position:	31 … 0
Bit field:	CBCR[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	CBCR[31:0]	Set the address for storing the chrominance component data of the captured bottom-field data (8-pixel units). The lower 3 bits should be written 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CDBCR specifies the address where the chrominance component of the captured bottom-field data is to be stored in both-field image capture. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. Set the start address of the memory area where the chrominance component of the captured bottom-field image is to be stored by CDBCR. CDBCR is not used in frame image capture, one-field image capture, or data fetch.

Because the address must be specified in 32 bits, the address set by CDBCR must be in longword units.

Set the address of the starting point of the memory area where the chrominance component of the captured bottom-field image is to be stored in this register, as shown in Figure 60.38. The chrominance component has an output data format like that in Figure 60.39, and is saved in the memory in this format.

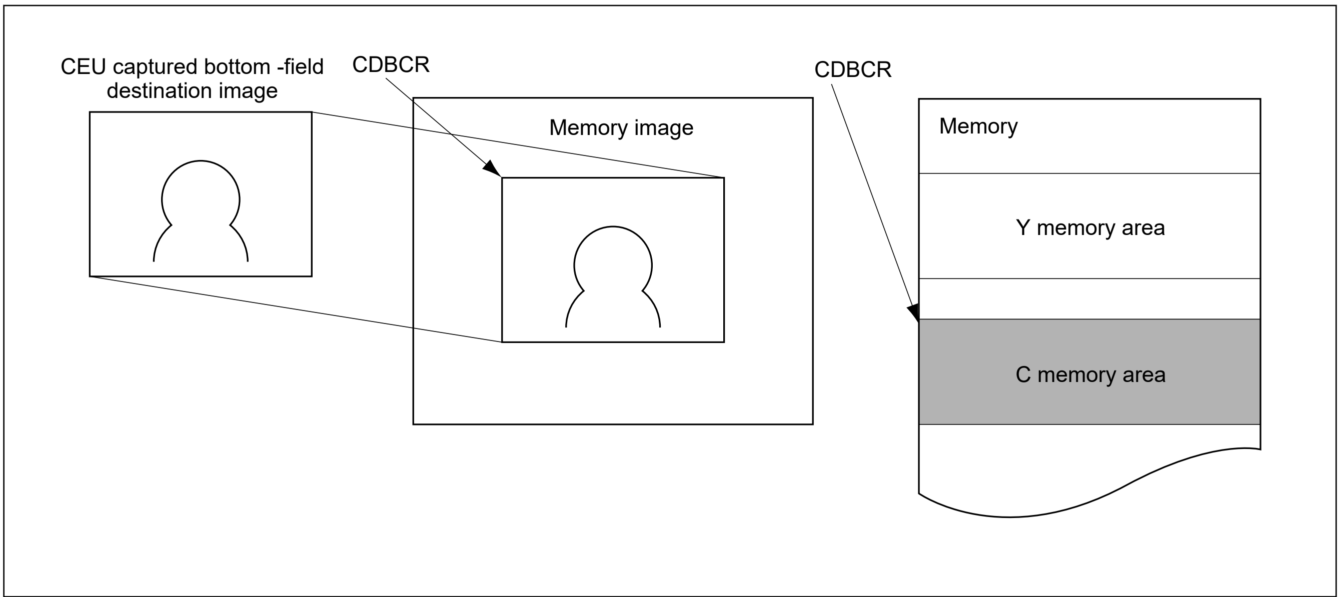


Figure 60.38 Relationship between captured bottom-field image and chrominance component memory area

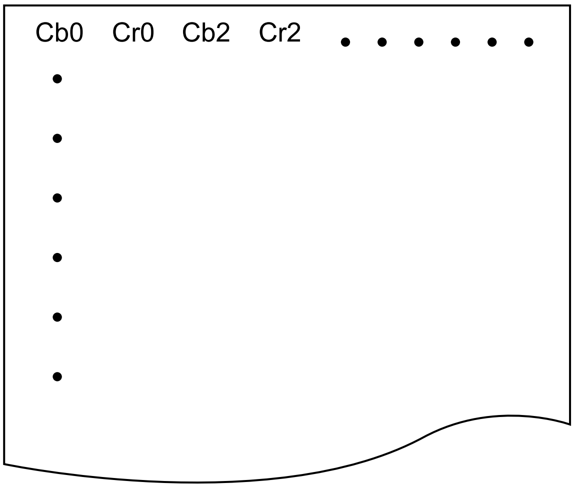


Figure 60.39 Image of storing chrominance components in memory

60.2.17 CBDSR, CBDSR_x : Capture Bundle Destination Size Register (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x004C (CBDSR)
0x104C (CBDSR_B)
0x204C (CBDSR_M)

Bit position:	31	30	29	28	27	26	25	24	23	22 ... 0
Bit field:	—	—	—	—	—	—	—	—	—	CBVS[22:0]
Value after reset:	0	0	0	0	0	0	0	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
22:0	CBVS[22:0]	Select the number of lines or number of bytes for output to the memory in a bundle write. Image capture and data synchronous fetch: Number of lines for output to the memory in a bundle write. Unit: 8 lines, min.: 8 lines, max.: 1,920 lines (0x780) Data enable fetch: Number of bytes for output to the memory in a bundle write. Unit: 32 bytes, min.: 512 bytes, max.: 6291456 bytes (0x600000) The value of writing are ignored on the lower 3 bits, so write 0 to the lower 3 bits.	R/W
31:23	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CBDSR sets the size of output to memory in a bundle write. The number of output lines should be specified for image capture or data synchronous fetch. The number of bytes should be specified for data enable fetch.

(a) Image capture and data synchronous fetch

Set the number of lines of captured data to be written to the memory by a bundle write as a multiple of eight. This register is valid only when CDOCR.CBE = 1. When CDOCR.CBE = 1 and this register cleared to 0, this module operates with the number of lines of captured data to be written to the memory as eight. The maximum number of lines that can be set is 1,920 (0x780). Only bits CBVS[11:3] are valid.

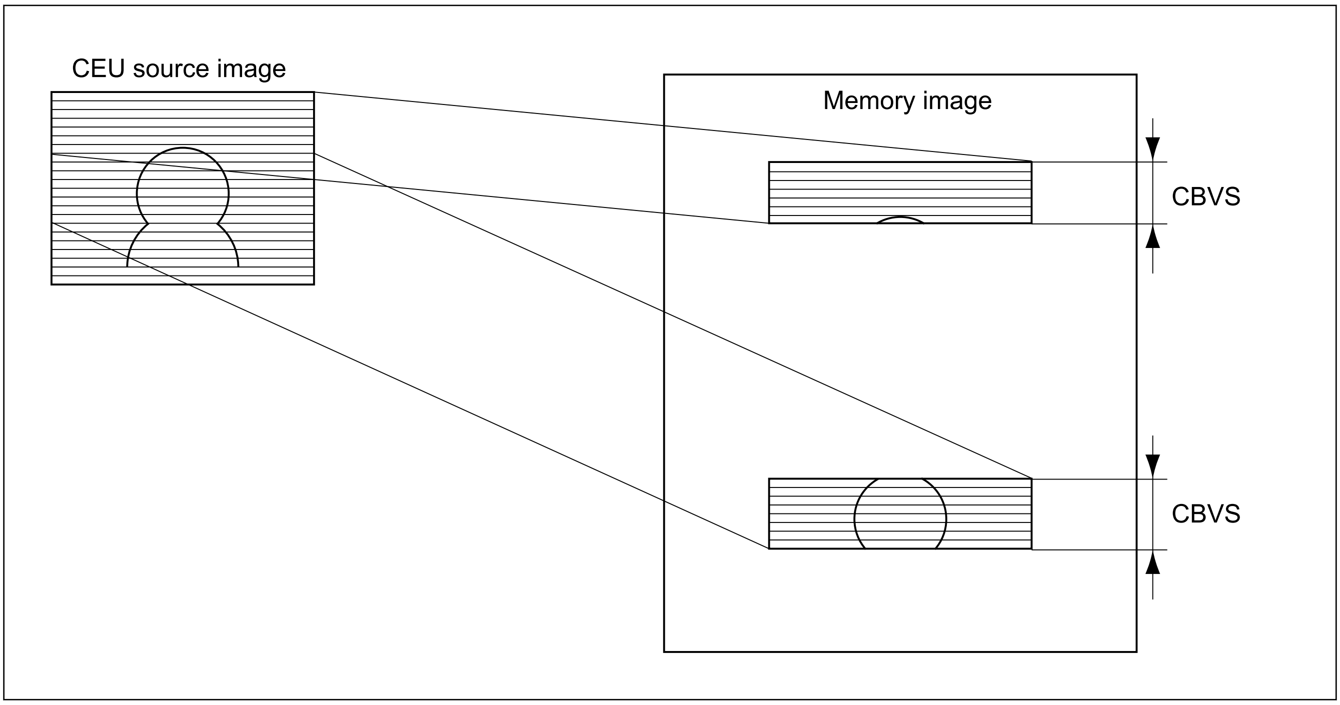


Figure 60.40 Image of storing captured image in memory by bundle write

(b) Data enable fetch

Set the number of bytes of captured data to be written to the memory by a bundle write as a multiple of 32. This register is valid only when CDOCR.CBE = 1. The minimum settable size is 512 bytes. When a number smaller than 512 bytes is specified, operation is not guaranteed.

60.2.18 CFWCR : Firewall Operation Control Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x005C

Bit position:	31 … 5	4	3	2	1	0
Bit field:	FWV[26:0]	—	—	—	—	FWE
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FWE	Firewall Operation With the setting of FWE = 1, when a write address exceeds the value set with FWV, the address is retained and an interrupt source FWF is set. After this, the address is not incremented and data is overwritten on the upper limit address. 0: Firewall is not activated. 1: Firewall is activated.	R/W
4:1	—	These bits are read as 0. The write value should be 0.	R/W
31:5	FWV[26:0]	Specify the upper limit of a write address. Specify the upper 27 bits of the 32-bit address. The upper limit of an address is FWV[26:0]<<5 + 0x1F.	R/W

Note: S-TYPE-3, P-TYPE-3

CFWCR specifies the upper limit of a write address in data enable fetch. When the VD input from an external module dose not go low and end notification is not given, this register can prevent writing to memory from being out of control.

This register is enabled only in data enable fetch.

60.2.19 CLFCR, CLFCR_x : Capture Low-Pass Filter Control Register (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0060 (CLFCR)
0x1060 (CLFCR_B)
0x2060 (CLFCR_M)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LPF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LPF	Enables or disables operation of the low-pass filter. The low-pass filter removes high-frequency components from the destination image in the horizontal direction. Clear this bit to 0 in data fetch mode. 0: Low-pass filter not used 1: Low-pass filter used (only in the horizontal direction)	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CLFCR specifies whether or not to operate the low-pass filter. In data fetch mode, clear the LPF bit to 0.

The characteristic of the low-pass filter installed in the CEU causes the phase location of the image processed by the low-pass filter to be shifted right by one pixel compared to the raw image.

60.2.20 CDOCR, CDOCR_x : Capture Data Output Control Register (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0064 (CDOCR)
0x1064 (CDOCR_B)
0x2064 (CDOCR_M)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CBE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	CDS	—	COLS	COWS	COBS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	COBS	Controls swapping in 8-bit units for data output from the CEU. 0: Data is not swapped in 8-bit units 1: Data is swapped in 8-bit units	R/W
1	COWS	Controls swapping in 16-bit units for data output from the CEU. 0: Data is not swapped in 16-bit units 1: Data is swapped in 16-bit units	R/W
2	COLS	Controls swapping in 32-bit units for data output from the CEU. 0: Data is not swapped in 32-bit units 1: Data is swapped in 32-bit units	R/W

Bit	Symbol	Function	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	CDS	Sets the image format when outputting the image data captured in the YCbCr422 format to the memory. When 0 is written to this bit, only the luminance component (Y) is output and no chrominance components (Cb and Cr) are output for the odd-numbered lines. With an interlace source image, similarly only the luminance component (Y) is output and no chrominance components (Cb and Cr) are output for the odd-numbered lines of the field. In data fetch mode, set this bit to 1. 0: Converts the YCbCr422 format to the YCbCr420 format before outputting data to the memory 1: Outputs data in the YCbCr422 format to the memory without conversion	R/W
15:5	—	These bits are read as 0. The write value should be 0.	R/W
16	CBE	Controls the number of lines of captured data to be written to the memory. ● Image capture This bit controls the number of lines of captured data to be written to the memory. When bundle write is set by this register, captured data is written in line units specified by CBDSR to the addresses specified by CDAYR and CDACR, and CDAYR2 and CDACR2 (CDBYR and CDBCR, and CDBYR2 and CDBCR2 for the bottom field in both-field capture) alternately (Figure 60.41). When captured data has been written for the number of lines set by CBDSR, a write end interrupt corresponding to each address setting register occurs. However, after write for one-frame (one-field) capture ends, a bundle write end interrupt does not occur even when bundle write has finished. ● Data synchronous fetch This bit controls the number of lines of captured data to be written to the memory. When bundle write is set by this register, captured data is written in line units specified by CBDSR to the addresses specified by CDAYR and CDAYR2 alternately. When captured data has been written for the number of lines set by CBDSR, a write end interrupt corresponding to each address setting register occurs. However, after write for one-frame capture ends, a bundle write end interrupt does not occur even when bundle write has finished. ● Data enable fetch This bit controls the number of bytes of captured data to be written to the memory. When bundle write is set by this register, captured data is written in byte units specified by CBDSR to the addresses specified by CDAYR and CDAYR2 alternately. When captured data has been written for the number of bytes set by CBDSR, a write end interrupt corresponding to each address setting register occurs. Also, only in data enable fetch, a bundle write end interrupt occurs when bundle write has finished after write for one-frame capture ends. Table 60.9 shows the correspondence between address setting registers and write end interrupt sources. Figure 60.42 shows the timing of write end interrupts in image capture and data synchronous fetch. Figure 60.43 shows the timing of write end interrupts in data enable fetch. 0: Normal write 1: Bundle write	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CDOCR sets the format for outputting captured data to the memory. In data fetch mode, set the CDS bit to 1.

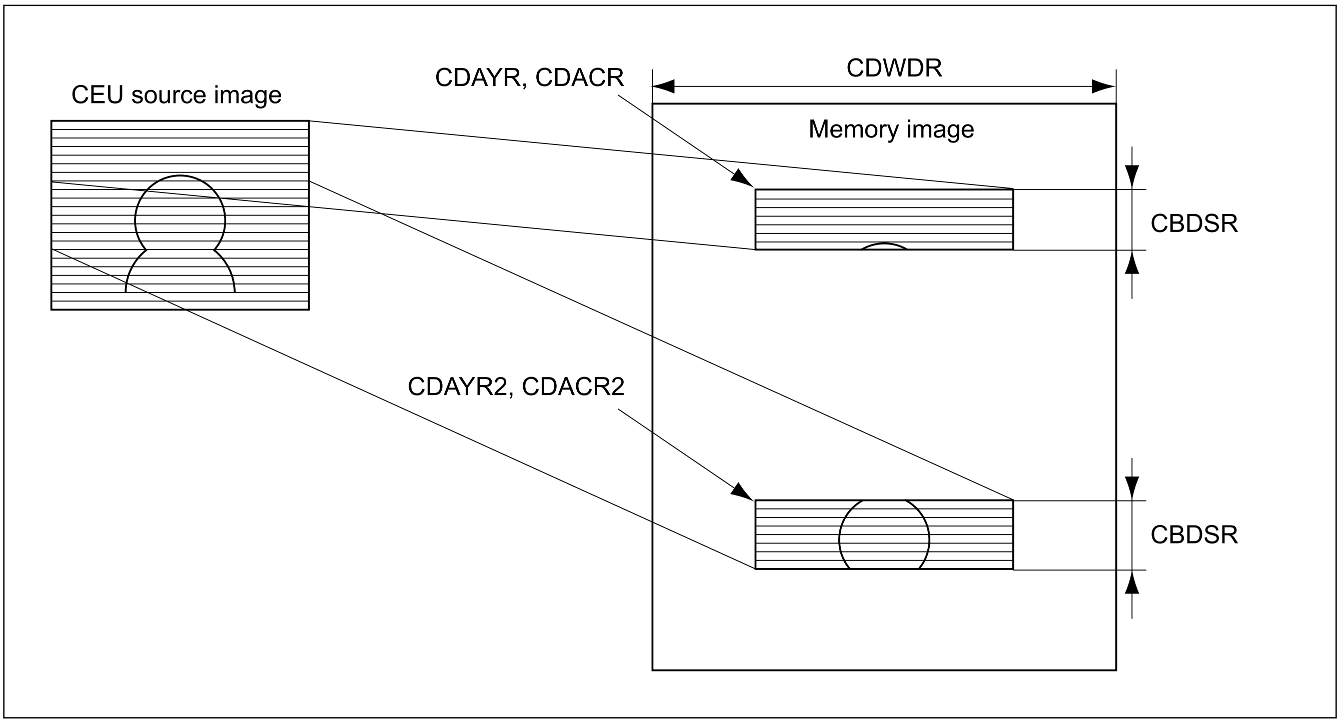


Figure 60.41 Image of bundle write to memory

Table 60.9 Correspondence between address setting registers and write end interrupt sources

Address setting registers	Bundle write end interrupt source
CDAYR, CDACR	CPBE1 bit in CETCR
CDAYR2, CDACR2	CPBE2 bit in CETCR
CDBYR, CDBCR	CPBE3 bit in CETCR
CDBYR2, CDBCR2	CPBE4 bit in CETCR

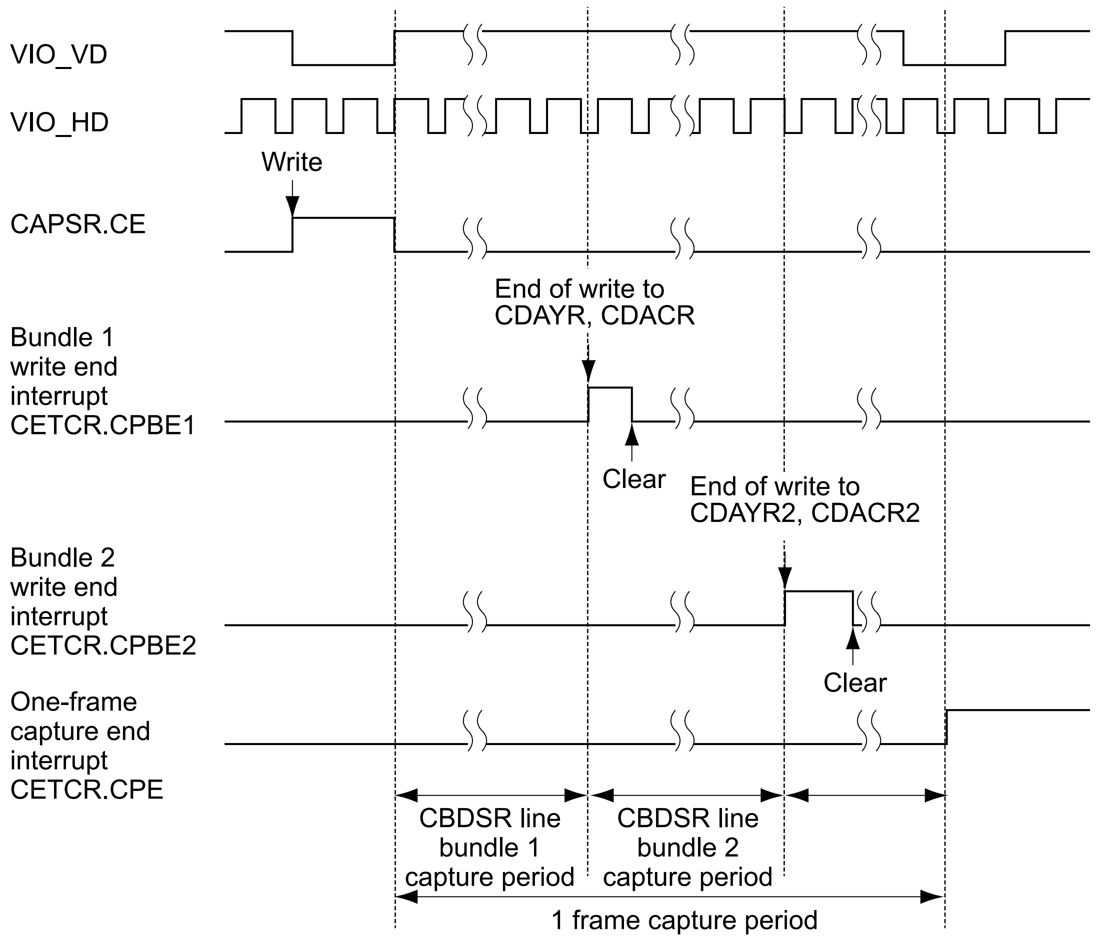


Figure 60.42 Timing of write end interrupts (Image capture, data synchronous fetch)

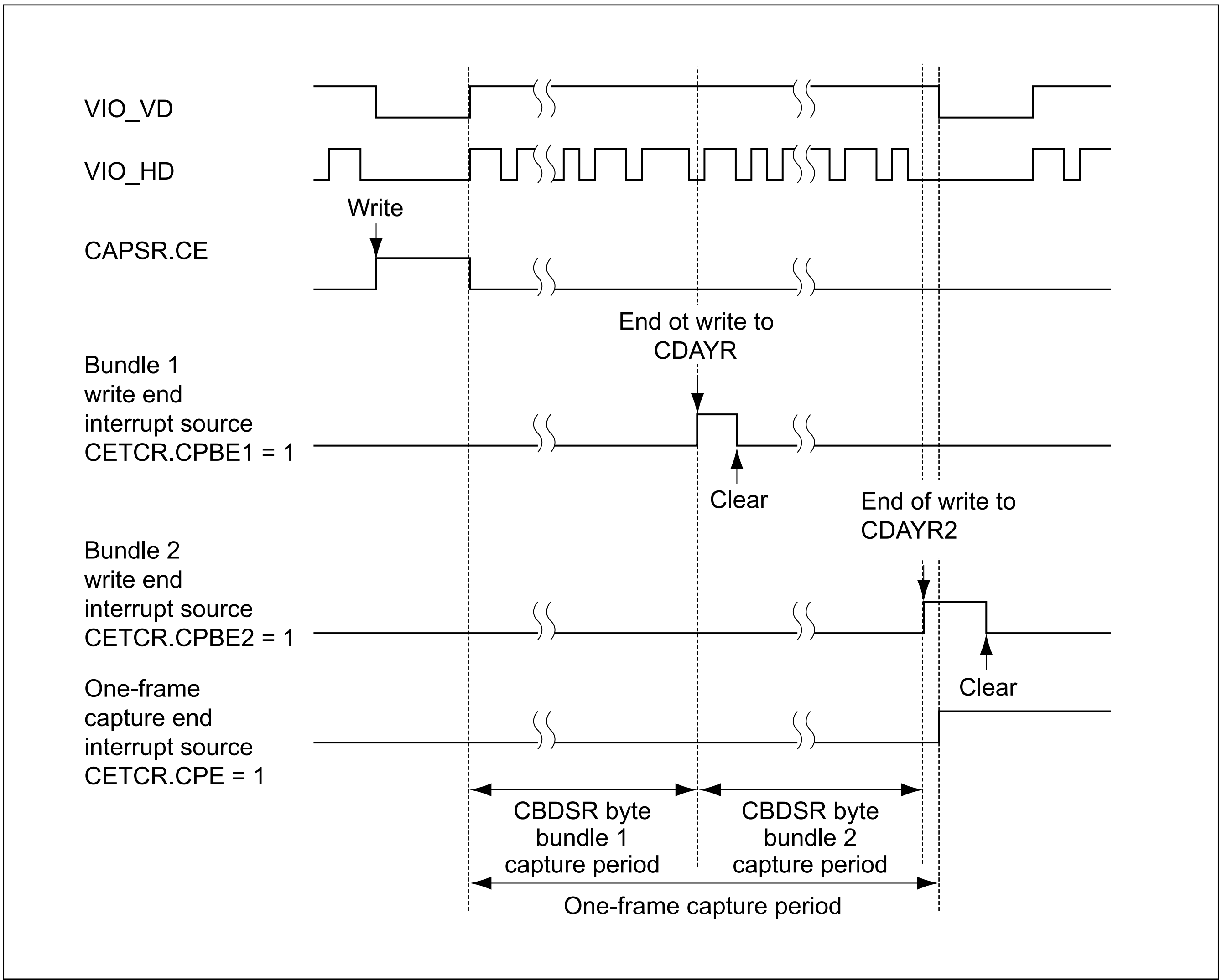


Figure 60.43 Timing of write end interrupts (data enable fetch)

For data output from the CEU, the COLS, COWS, and COBS bits control swapping in 32-bit, 16-bit, and 8-bit units, respectively. Set these bits when data is misaligned because of a difference in endian. The data swapping bits are shown below. These bits can be set similarly in data fetch mode.

Data can be swapped in 8-bit, 16-bit, 32-bit units, or in 32 bits, 16 bits and 8 bits, as shown in Figure 60.44. To enable data swapping, set the corresponding control bit to 1.

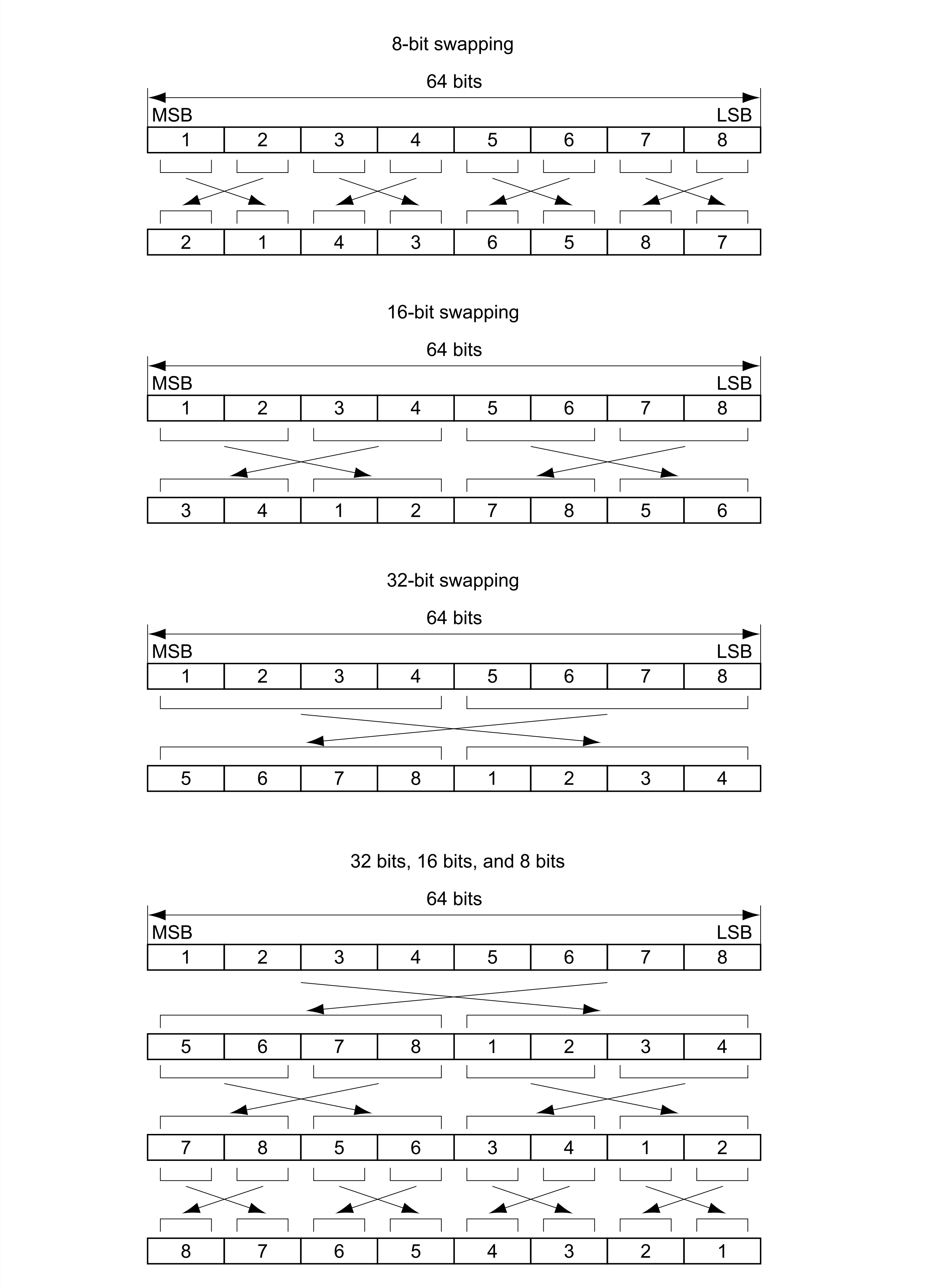


Figure 60.44 Data swapping by data aligner

60.2.21 CEIER : Capture Event Interrupt Enable Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0070

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	NVDIE	NHDIE	FWFIE	—	—	VBPIE	—	IGVSIE	IGHSIE	CDTOFIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CPBE4IE	CPBE3IE	CPBE2IE	CPBE1IE	—	—	VDIE	HDIE	—	—	—	IGRWIE	—	—	CFEIE	CPEIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CPEIE	One-Frame Capture End Interrupt Enable 0: Disables a one-frame capture end interrupt 1: Enables a one-frame capture end interrupt	R/W
1	CFEIE	CFE Interrupt Enable 0: Disables a CFE interrupt 1: Enables a CFE interrupt	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	IGRWIE	Register-Access-During-Capture Interrupt Enable 0: Disables a register-access-during-capture interrupt 1: Enables a register-access-during-capture interrupt	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	HDIE	HD Interrupt Enable 0: Disables an HD interrupt 1: Enables an HD interrupt	R/W
9	VDIE	VD Interrupt Enable 0: Disables a VD interrupt 1: Enables a VD interrupt	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	CPBE1IE	CPBE1 Interrupt Enable 0: Disables a CPBE1 interrupt 1: Enables a CPBE1 interrupt	R/W
13	CPBE2IE	CPBE2 Interrupt Enable 0: Disables a CPBE2 interrupt 1: Enables a CPBE2 interrupt	R/W
14	CPBE3IE	CPBE3 Interrupt Enable 0: Disables a CPBE3 interrupt 1: Enables a CPBE3 interrupt	R/W
15	CPBE4IE	CPBE4 Interrupt Enable 0: Disables a CPBE4 interrupt 1: Enables a CPBE4 interrupt	R/W
16	CDTOFIE	CDTOF Interrupt Enable 0: Disables a CDTOF interrupt 1: Enables a CDTOF interrupt	R/W
17	IGHSIE	IGHS Interrupt Enable 0: Disables an IGHS interrupt 1: Enables an IGHS interrupt	R/W
18	IGVSIE	IGVS Interrupt Enable 0: Disables an IGVS interrupt 1: Enables an IGVS interrupt	R/W

Bit	Symbol	Function	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
20	VBPIE	VBP Interrupt Enable 0: Disables a VBP interrupt 1: Enables a VBP interrupt	R/W
22:21	—	These bits are read as 0. The write value should be 0.	R/W
23	FWFIE	FWF Interrupt Enable 0: Disables a FWF interrupt 1: Enables a FWF interrupt	R/W
24	NHDIE	Non-HD Interrupt Enable Disable this interrupt (NHDIE = 0) for data enable fetch. 0: Disables a non-HD interrupt 1: Enables a non-HD interrupt	R/W
25	NVDIE	Non-VD Interrupt Enable Disable this interrupt (NVDIE = 0) for data enable fetch. 0: Disables a non-VD interrupt 1: Enables a non-VD interrupt	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CEIER enables or disables interrupts of the event flag register that generates CEU interrupts.

60.2.22 CETCR : Capture Event Flag Clear Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0074

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	NVD	NHD	FWF	—	—	VBP	—	IGVS	IGHS	CDTOF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CPBE4	CPBE3	CPBE2	CPBE1	—	—	VD	HD	—	—	—	IGRW	—	—	CFE	CPE
Value after reset:	0	0	0	0	0	0	x	x	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CPE	An interrupt indicating that capturing of one frame from an external module has finished. This interrupt is output when the last captured data has been transferred and the end notification received, regardless of the next VD input. This interrupt indicates that capturing of one frame has finished. This bit is set to 1 when the image of the size set in CAPWR is captured and the last data transfer to the bus finished (see Figure 60.46).	R/W
1	CFE	An interrupt indicating that capturing of one field from an external module has finished. This interrupt is output when the last captured data has been transferred and the end notification received, regardless of the next VD input (see Figure 60.45). This interrupt occurs only in both-field capture mode.	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	IGRW	An interrupt indicating that during capturing, access was attempted to a register to which writing during operation is prohibited. Among the CEU registers, writing during capturing is prohibited for some registers. Table 60.10 shows which registers can/cannot be written to during capturing. This bit is set to 1 when a register to which writing during capturing is prohibited has been written to.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
8	HD	An interrupt indicating that HD (horizontal sync signal) was input from an external module. This bit is set to 1 when an HD input from an external module is detected. Immediately after the HDPOL bit in CAMCR is modified, a pseudo HD is input and this bit is set to 1. The HD interrupt after the HDPOL bit is modified should be ignored.	R/W
9	VD	An interrupt indicating that VD (vertical sync signal) was input from an external module. In data enable fetch mode, this bit is set to 1 when a VD input from an external module is detected. In image capture mode and data synchronous fetch mode, this module generates a VD interrupt in response to the first detection of the active level of HD following detection of the active level of VD from an external module. Note that, when VD and HD are asserted and detected at the same time, this module generates a VD interrupt at that time. Immediately after the VDPOL bit in CAMCR is modified, a pseudo VD is input and this bit is set to 1. The VD interrupt after the VDPOL bit is modified should be ignored.	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	CPBE1	An interrupt indicating that writing to CDAYR and CDACR in a bundle write has finished. This interrupt is output when the last captured data has been transferred and the end notification received, regardless of the next HD input. This bit is set to 1 when data for the number of lines (number of bytes in data enable fetch) set in CBDSR has been captured and the last data transfer to the bus has completed. However, in image capture or data synchronous fetch, this interrupt does not occur when the last captured data in a bundle write is the last captured data of a frame (field).	R/W
13	CPBE2	An interrupt indicating that writing to CDAYR2 and CDACR2 in a bundle write has finished. This interrupt is output when the last captured data has been transferred and the end notification received, regardless of the next HD input. This bit is set to 1 when data for the number of lines (number of bytes in data enable fetch) set in CBDSR has been captured and the last data transfer to the bus has completed. However, in image capture or data synchronous fetch, this interrupt does not occur when the last captured data in a bundle write is the last captured data of a frame (field).	R/W
14	CPBE3	An interrupt indicating that writing to CDBYR and CDBCR in a bundle write has finished. This interrupt is output when the last captured data has been transferred and the end notification received, regardless of the next HD input. This bit is set to 1 when data for the number of lines set in CBDSR has been captured and the last data transfer to the bus has completed. However, this interrupt does not occur when the last captured data in a bundle write is the last captured data of a frame (field).	R/W
15	CPBE4	An interrupt indicating that writing to CDBYR2 and CDBCR2 in a bundle write has finished. This interrupt is output when the last captured data has been transferred and the end notification received, regardless of the next HD input. This bit is set to 1 when data for the number of lines set in CBDSR has been captured and the last data transfer to the bus has completed. However, this interrupt does not occur when the last captured data in a bundle write is the last captured data of a frame (field).	R/W
16	CDTOF	An interrupt indicating that data overflowed in the CRAM of the write buffer Since data is input at realtime from an external module in capture operations, the frame image is overwritten unless the captured data is transferred from the CEU internal buffer to the memory at a certain or higher transfer rate. This bit is set to 1 when writing the data in the CRAM of the CEU internal write buffer to the bus is not performed within time and data has overflowed.	R/W
17	IGHS	An interrupt generated when the number of HD cycles set in CMCYR differ from the number of HD cycles input from an external module. This bit is set to 1 when there is an illegal HD input from an external module. This bit is set to 1 when the number of clock cycles for the HD input to the CEU differs from the value set in the HCYL bits in CMCYR. Note however that when the HCYL bits are cleared to 0, this interrupt does not occur.	R/W
18	IGVS	An interrupt generated when the number of VD cycles set in CMCYR differ from the number of VD cycles input from an external module. This bit is set to 1 when there is an illegal VD input from an external module. This bit is set to 1 when the number of HD cycles for the VD input to the CEU differs from the value set in the VCYL bits in CMCYR. Note however that when the VCYL bits are cleared to 0, this interrupt does not occur.	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
20	VBP	An interrupt indicating that VD has been input while the CEU holds data (insufficient vertical-sync front porch). The conditions for a VBP interrupt to occur are as follows: ● Condition 1 VD is input when there is captured data within the CEU ● Condition 2 The last transfer data cannot be internally detected due to a write buffer overflow or an illegal HD so that the end timing is unclear until the next VD (By generating a VBP interrupt at the VD input timing, capture fail can be announced.) When a VBP interrupt occurs, a capture end interrupt (CPE bit in CETCR) does not occur and the image of that frame is not captured correctly. Though a capture end interrupt (CPE bit) will occur on rare occasions, it should be ignored in this case. Capturing cannot be performed until the next VD (even if the CE bit (capture reservation signal) in CAPSR is 1, capturing does not start). In the case of condition 2, instead of waiting for a VBP interrupt to occur, execute a software reset (CPKIL bit in CAPSR) to stop capturing and then restart capturing. In this case, since capture operation is terminated without waiting for the next VD, a VBP interrupt does not occur and capturing can be performed from the next VD.	R/W
22:21	—	These bits are read as 0. The write value should be 0.	R/W
23	FWF	The interrupt is generated when data is written to the address that exceeds the value specified with CFWCR.FMV. This bit is set to 1 when data is written to the address that exceeds the value specified with CFWCR.FMV while CFWCR.FWE = 1.	R/W
24	NHD	An interrupt indicating that no HD was input. The timing for a non-HD interrupt to occur differs depending on the bit width of the digital image input pins. 8-bit digital image input pins: Occurs when the 11-bit internal counter that is incremented every eight cycles becomes full. Accordingly, this bit is set to 1 when no HD has been input for at least 16,376 cycles since the last HD was input. 16-bit digital image input pins: Occurs when the 12-bit internal counter that is incremented every four cycles becomes full. Accordingly, this bit is set to 1 when no HD has been input for at least 16,380 cycles since the last HD was input. When connecting a camera whose HD is fixed low when VD is low, this bit may be set to 1. Ignore this interrupt during data enable fetch.	R/W
25	NVD	An interrupt indicating that no VD was input. A non-VD interrupt occurs when the 14-bit internal counter becomes full. Accordingly, this bit is set to 1 when no VD has been input for at least 16,383 lines since the last VD was input.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CETCR notifies the CPU of the source of an interrupt that is generated in the CEU. The flags of CETCR can be used as interrupt signals. When the corresponding interrupt is enabled, an interrupt (CEU_CEUI) is generated. To clear the interrupt, clear the bit corresponding to the interrupt source to 0. After several cycles have passed after modifying the bit, the interrupt is cleared.

To clear the bit corresponding to the interrupt source to be cleared to 0 and retain that state, write 1 to that bit. For example, to clear only the CPE bit to 0, write 0xFFFFFFFE to CETCR.

In CETCR, only bits to which 0 is written are cleared. Bits to which 1 is written retain their current values. To clear an interrupt source, write 0 only to the bit corresponding to the interrupt source to be cleared, and 1 to the other bits.

Note: Since the CETCR value becomes undefined in the following cases, clear all bits in CETCR to 0.
- VD and HD bits immediately after system reset or Software Standby mode is entered.
- VD and HD bits after the polarities of the capture interface sync signals are changed.

Table 60.10 Registers that can/cannot be modified during capturing (1 of 2)

Register name	Modification during capturing
CAPSR	Possible
CAPCR	Prohibited
CAMCR	Prohibited

Table 60.10 Registers that can/cannot be modified during capturing (2 of 2)

Register name	Modification during capturing
CMCYR	Prohibited
CAMOR	Possible
CAPWR	Possible
CAIFR	Prohibited
CRCNTR	Possible
CRCMPR	Prohibited
CFLCR	Possible
CFSZR	Possible
CDWDR	Possible
CDAYR	Possible
CDACR	Possible
CDBYR	Possible
CDBCR	Possible
CBDSR	Possible
CFWCR	Possible
CLFCR	Possible
CDOCR	Possible
CEIER	Possible
CETCR	Possible
CSTSR	Prohibited
CDSSR	Prohibited
CDAYR2	Possible
CDACR2	Possible
CDBYR2	Possible
CDBCR2	Possible

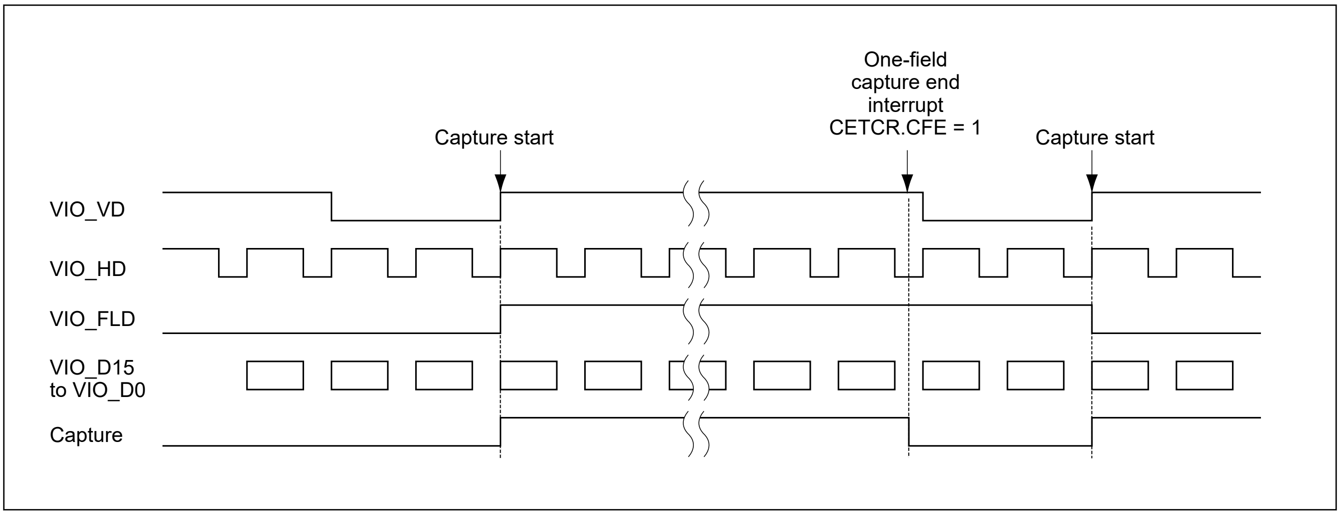


Figure 60.45 CFE generation timing

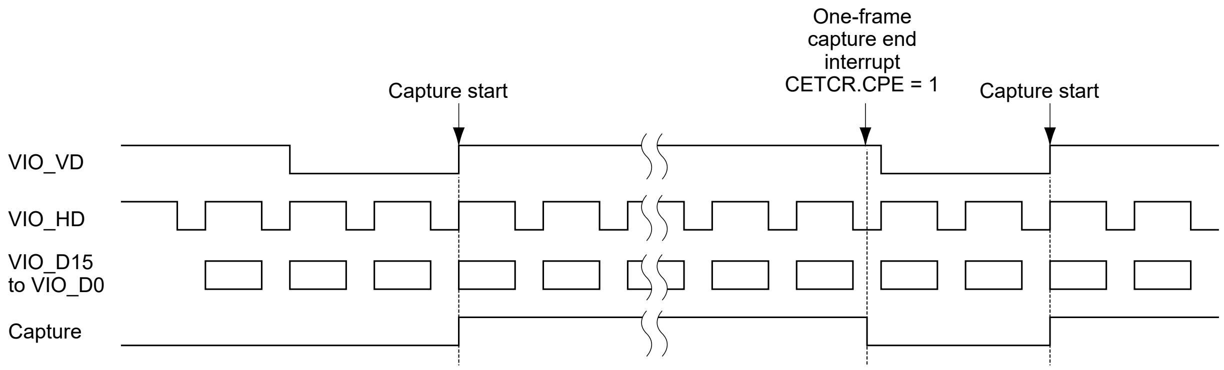


Figure 60.46 CPE generation timing

60.2.23 CSTSR : Capture Status Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x007C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	CRST	—	—	—	—	—	—	—	CPFLD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPTON
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CPTON	Indicates that the CEU is operating. This bit retains 1 during the period that starts from the internal VD at capture start and ends when a one-frame capture end interrupt occurs. Figure 60.47 shows the CEU operating period.	R
15:1	—	These bits are read as 0.	R
16	CPFLD	Indicates which field is being captured. 0: Bottom field is being captured 1: Top field is being captured	R
23:17	—	These bits are read as 0.	R
24	CRST	Indicates which register plane is currently used. 0: Plane A of the register is being used 1: Plane B of the register is being used	R
31:25	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

CSTSR indicates the internal status of the CEU. CSTSR differs from CETCR in that no interrupt is generated for the events indicated in CSTSR.

The CEU operating/halt state can be confirmed using CSTSR. To confirm the halt state of the CEU, make sure that the status bit (bit 0) indicating that the CEU is operating is cleared to 0 for sure.

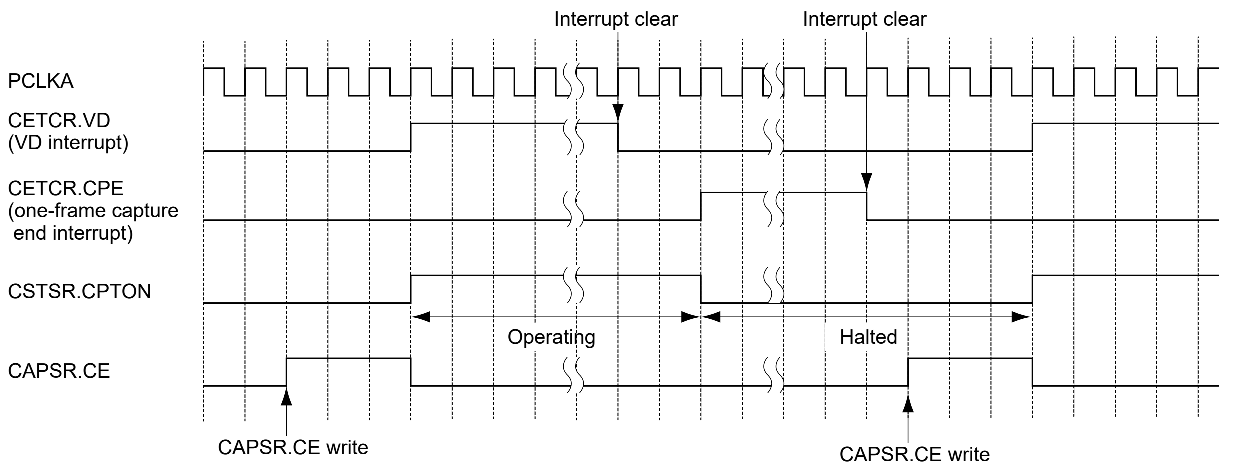


Figure 60.47 Operating status during capturing

60.2.24 CDSSR : Capture Data Size Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0084

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CDSS[31:0]																															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CDSS[31:0]	Indicate the size of data written to the memory in data enable fetch. In a bundle write, size of data written to the selected address at the end of one-frame capture is indicated. In a bundle write, as soon as the number of bytes specified by CBDSR is transferred to the bus, address to which data is written is switched. Therefore, if one-frame capture is completed at the same time as a bundle write is completed, this register indicates 0x00000000. Figure 60.48 and Figure 60.49 show the overall timing of the CDSSR operation in a bundle write.	R

Note: S-TYPE-3, P-TYPE-3

CDSSR indicates the size of data written to the memory in data enable fetch. As this register indicates a correct value at the end of capture, confirm this register when capture is completed.

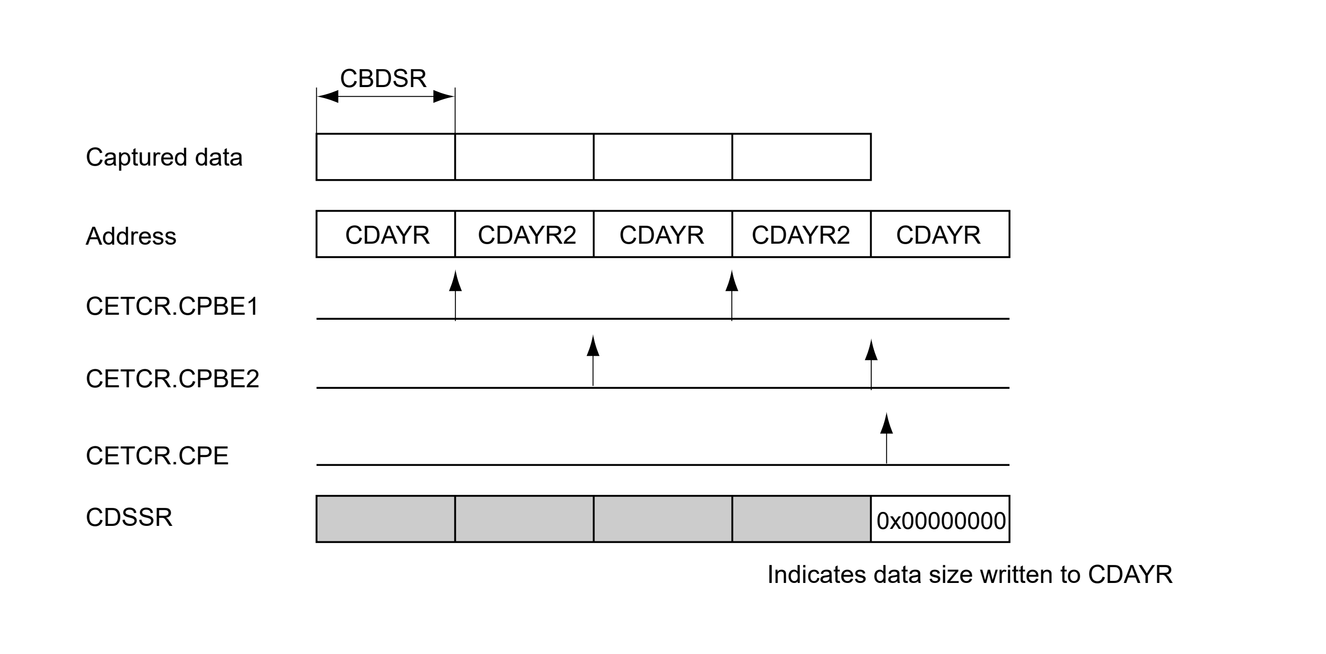


Figure 60.48 Overall timing of CDSSR operation in bundle write (when bundle write end and capture end coincide)

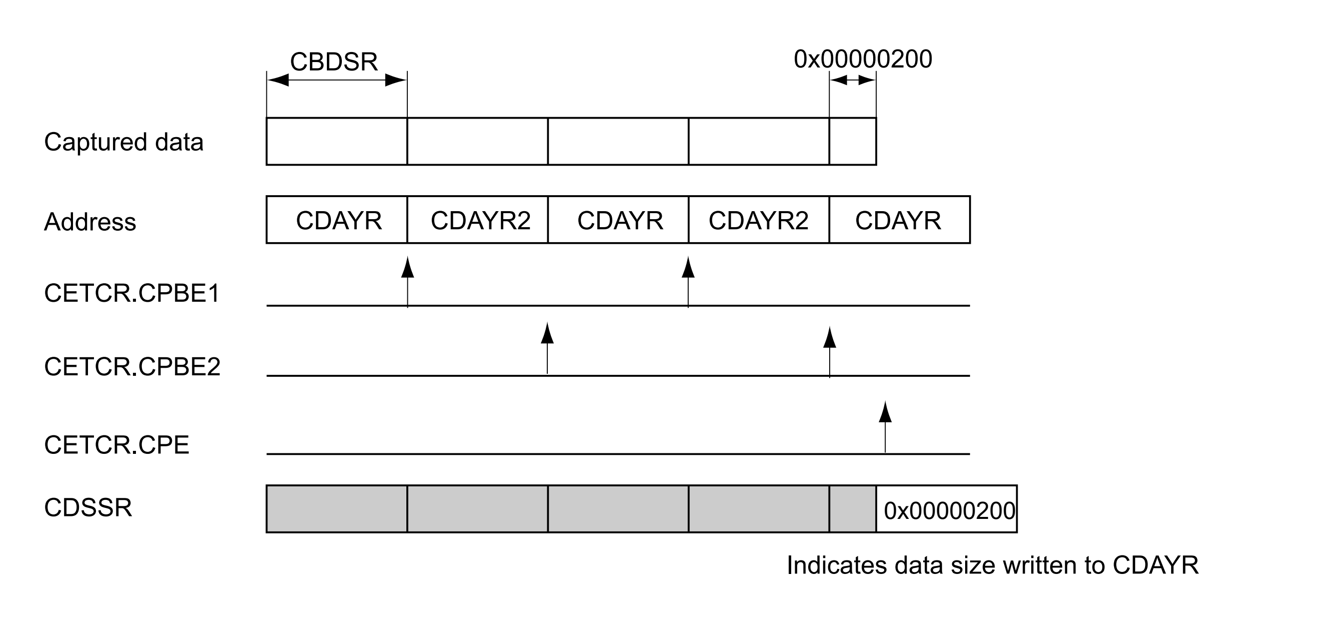


Figure 60.49 Overall timing of CDSSR operation in bundle write (when bundle write end and capture end do not coincide)

60.2.25 CDAYR2, CDAYR2_x : Capture Data Address Y Register 2 (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0090 (CDAYR2)
0x1090 (CDAYR2_B)
0x2090 (CDAYR2_M)

Bit position:	31 … 0
Bit field:	CAYR2[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	CAYR2[31:0]	Capture Data Address Y Frame image capture: These bits set the address for storing the luminance component data of the captured data (8-pixel units). ● One-field image capture: These bits set the address for storing the luminance component data of the captured data (8-pixel units). ● Both-field image capture: These bits set the address for storing the luminance component data of the captured top-field data (8-pixel units). ● Data synchronous fetch: These bits set the address for storing data (8-byte units). ● Data enable fetch: These bits set the address for storing data (32-byte units). The lower 3 bits should be written 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CDAYR2 specifies the address for the luminance component used in a bundle write and the address for data storage in a bundle write in data fetch. CDAYR2 is used only in a bundle write.

CDAYR2 specifies the address where the luminance component of the captured data is to be stored in frame image capture or one-field image capture, the address where the luminance component of the captured top field is to be stored in both-field image capture, and the address where data is to be stored in data fetch. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. In frame image capture or one-field image capture, set the start address of the memory area where the luminance component of the captured data is to be stored by CDAYR2. In both-field image capture, set the start address of the memory area where the luminance component of the captured top-field image is to be stored by CDAYR2. In data fetch, set the start address of the memory area to be used for data storage.

The address specified by this register must be in 32-bit units.

Set the address of the starting point of the memory area where the luminance component of the captured image is to be stored in this register, as shown in Figure 60.50.

- Frame image capture: Set the address of the starting point of the memory area where the luminance component of the captured image is to be stored.
- One-field image capture: Set the address of the starting point of the memory area where the luminance component of the captured image is to be stored.
- Both-field image capture: Set the address of the starting point of the memory area where the luminance component of the captured top-field image is to be stored.
- Data synchronous fetch: Set the address of the starting point of the memory area where the captured data is to be stored.
- Data enable fetch: Set the address of the starting point of the memory area where the captured data is to be stored in 32-byte units.

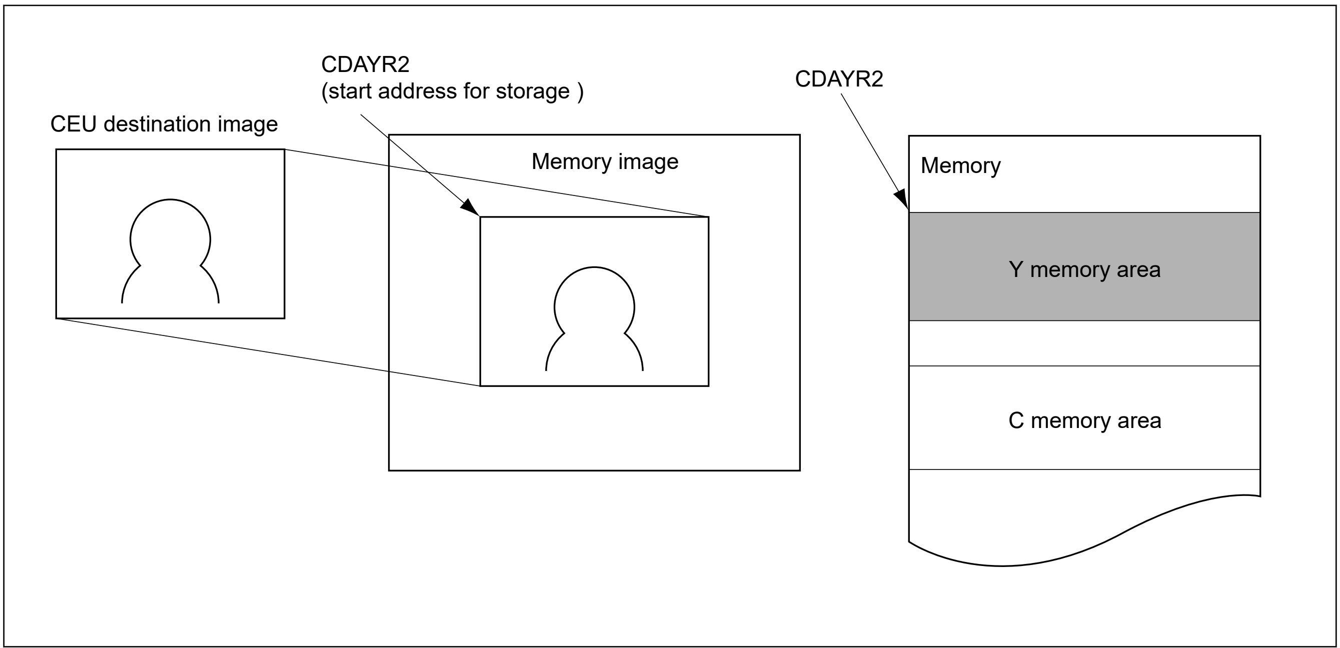


Figure 60.50 Relationship between captured image and luminance component memory area

60.2.26 CDACR2, CDACR2_x : Capture Data Address C Register 2 (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0094 (CDACR2)
0x1094 (CDACR2_B)
0x2094 (CDACR2_M)

Bit position:	31 … 0
Bit field:	CACR2[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	CACR2[31:0]	Capture Data Address C ● Frame image capture: These bits set the address for storing the chrominance component data of the captured data (8-pixel units). ● One-field image capture: These bits set the address for storing the chrominance component data of the captured data (8-pixel units). ● Both-field image capture: These bits set the address for storing the chrominance component data of the captured top-field data (8-pixel units). The lower 3 bits should be written 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CDACR2 specifies the address for the chrominance component used in a bundle write. CDACR2 is used only in a bundle write.

CDACR2 specifies the address where the chrominance component of the captured data is to be stored in frame image capture or one-field image capture, and the address where the chrominance component of the captured top field is to be stored in both-field image capture. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. In frame image capture or one-field image capture, set the start address of the memory area where the chrominance component of the captured data is to be stored by CDACR2. In bothfield image capture, set the start address of the memory area where the chrominance component of the captured top-field image is to be stored by CDACR2. CDACR2 is not used in data fetch.

The address specified by this register must be in 32-bit units.

In this register, set the address of the starting point of the memory area where the captured data is to be stored in a bundle write, as shown in Figure 60.51.

- Frame image capture: Set the address of the starting point of the memory area where the chrominance component of the captured image is to be stored.
- One-field image capture: Set the address of the starting point of the memory area where the chrominance component of the captured image is to be stored.
- Both-field image capture: Set the address of the starting point of the memory area where the chrominance component of the captured top-field image is to be stored.

The chrominance component has an output data format as shown in Figure 60.52, and is saved in the memory in this format.

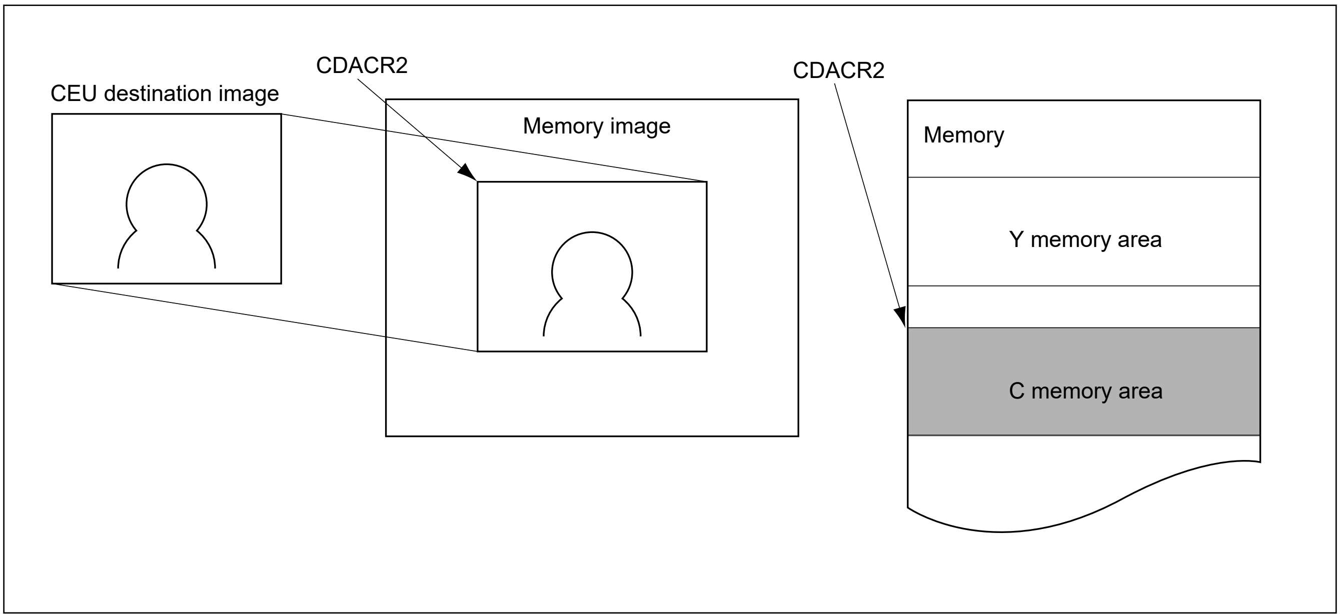


Figure 60.51 Relationship between captured image and chrominance component memory area

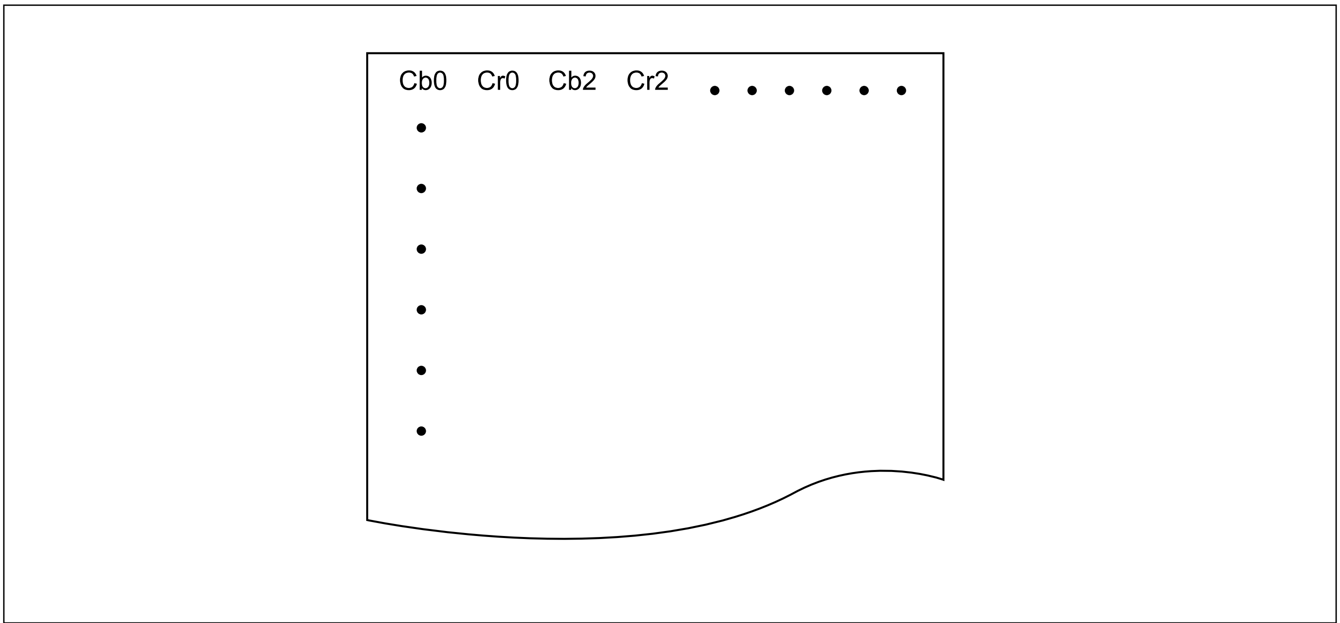


Figure 60.52 Image of storing chrominance components in memory

60.2.27 CDBYR2, CDBYR2_x : Capture Data Bottom-Field Address Y Register 2 (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x0098 (CDBYR2)
0x1098 (CDBYR2_B)
0x2098 (CDBYR2_M)

Bit position:	31 ... 0
Bit field:	CBYR2[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	CBYR2[31:0]	Set the address for storing the luminance component data of the captured bottom-field data (8-pixel units). The lower 3 bits should be written 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CDBYR2 specifies the address for the luminance component of the bottom field used in a bundle write. CDBYR2 is used only in a bundle write.

CDBYR2 specifies the address where the luminance component of the captured bottom-field data is to be stored in both-field image capture. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. Set the start address of the memory area where the luminance component of the bottom-field image captured in both-field image capture is to be stored by CDBYR2. CDBYR2 is not used in frame image capture, one-field image capture, or data fetch.

The address specified by this register must be in 32-bit units.

In this register, set the address of the starting point of the memory area where the luminance component of the captured bottomfield image is to be stored in a bundle write, as shown in Figure 60.53.

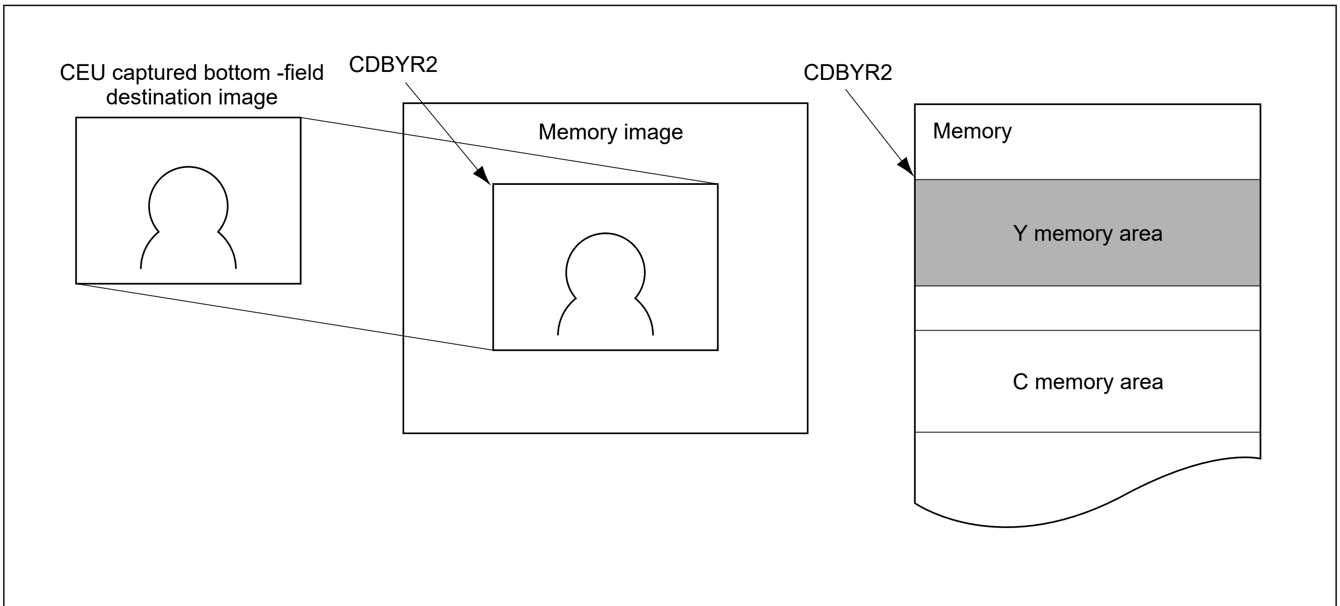


Figure 60.53 Relationship between captured bottom-field image and luminance component memory area

60.2.28 CDBCR2, CDBCR2_x : Capture Data Bottom-Field Address C Register 2 (x = B, M)

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x009C (CDBCR2)
0x109C (CDBCR2_B)
0x209C (CDBCR2_M)

Bit position:	31 – 0
Bit field:	CBCR2[31:0]
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	CBCR2[31:0]	Set the address for storing the chrominance component data of the captured bottom-field data (8-pixel units). The lower 3 bits should be written 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CDBCR2 specifies the address for the chrominance component of the bottom field used in a bundle write. CDBCR2 is used only in a bundle write.

CDBCR2 specifies the address where the chrominance component of the captured bottom-field data is to be stored in both-field image capture. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. Set the start address of the memory area where the chrominance component of the bottom-field image captured in both-field image capture is to be stored by CDBCR2. CDBCR2 is not used in frame image capture, one-field image capture, or data fetch.

The address specified by this register must be in 32-bit units.

In this register, set the address of the starting point of the memory area where the chrominance component of the captured bottomfield image is to be stored in a bundle write, as shown in Figure 60.54. The chrominance component has an output data format as shown in Figure 60.55, and is saved in the memory in this format.

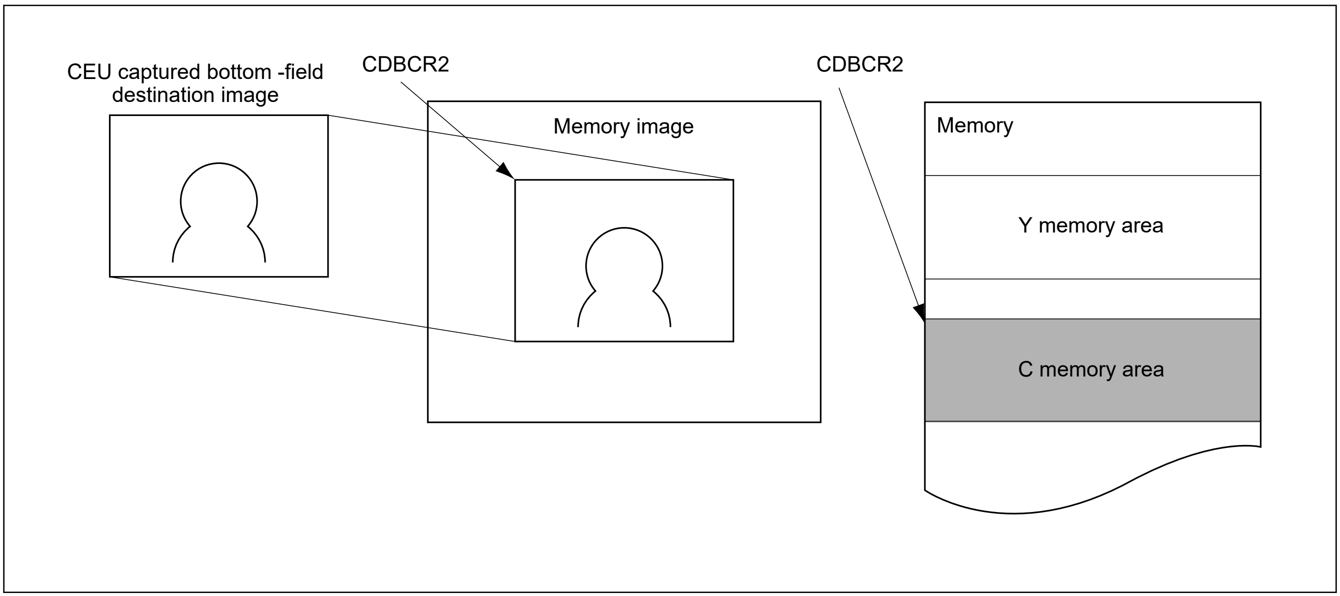


Figure 60.54 Relationship between captured bottom-field image and chrominance component memory area

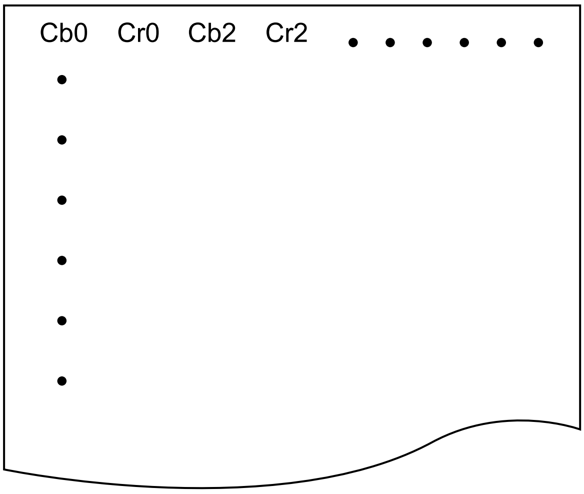


Figure 60.55 Image of storing chrominance components in memory

60.2.29 CBWER : CEU Bufferable Write Enable Register

Base address: CEU = 0x4034_8000
CEU_NS = 0x5034_8000

Offset address: 0x00A0

Bit position:	31																															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BWE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BWE	Bufferable Write Enable 0: Disables Bufferable Write 1: Enables Bufferable Write	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

BWE bit (Bufferable Write Enable)

BWE bit indicates bufferable write is either enabled or disabled.

Note that if this bit is 1, even if the write access as CEU is completed, the actual slave write may not have ended.

Note: When this bit is 1, it requests an early response by writing write data to the temporary buffer if possible to the slave. Therefore, even if write access as CEU is completed, the actual slave write may not be finished. For details on slave groups that support bufferable writes, see section 15.7.2. Operations When a Bus Error Occurs and respective module sections.

60.3 Usage Notes

60.3.1 Settings for the Module-Stop Function

CEU operation can be disabled or enabled using Module Stop Control Register C (MSTPCRC). The CEU is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Mode.

60.3.2 Conditions for Connection to an External Module

(1) Clock Frequency

The external input clock (VIO_CLK) should have a frequency at most the same as the CEU operating clock (PCLKA) frequency, with jitter on both sides included.

$$PCLKA \geq VIO\_CLK$$

(2) Blanking Period

The period from the last valid pixel in each line to the next horizontal sync signal HD must be at least 20 cycles.

(3) Fixed Period of Field Identification Signal

The field identification signal FLD should be fixed for at least 1-HD period since a VD input

60.3.3 Restrictions on Input/Output Functions

Table 60.11 lists the restrictions regarding the CEU input/output functions.

Table 60.11 Restrictions on CEU Input/Output Functions

Item	Restrictions
External module interface	The operating clock of the external module (VIO_CLK) should always have a frequency at most the same as that of the CEU operating (PCLKA), with jitter on both sides included.
	Selecting the interface, or modifying the frequency of the external module operating clock or HD/VD polarity must be done when capture operations are halted for sure.
	The capture horizontal size in image capture must be specified as follows: 8-bit interface: 16-cycle units 16-bit interface: 8-cycle units
	The capture horizontal size in data fetch must be specified as follows: 8-bit interface: 8-cycle units 16-bit interface: 4-cycle units
	The capture vertical size must be specified in 4-line units.
	The maximum number of cycles in the horizontal sync signal period should be 16,375 cycles of external input clock for 8-bit digital image input pins, or 16,379 cycles of external input clock for 16-bit digital image input pins.
	The maximum number of lines (HD count) in the vertical sync signal should be 16,382 lines.
	The minimum number of captured pixels should be sub-QCIF (128 × 96).
	The maximum number of captured pixels should be 5 megapixels (2,560 × 1,920).
	The capture size in data enable fetch must be controlled by VD and HD in 4-byte units. The range of the capture size in data enable fetch that can be input is as follows. Maximum: 6 MB (2,048 × 1,536 × 2) Minimum: 16 bytes
Memory output	The output address must be specified in 32-bit units.
	The horizontal size of the destination image (memory) must be specified in 8-pixel units.
	The number of horizontal output pixels (= horizontal clipping size) must be specified in 8-pixel units.
	The number of vertical output lines (HD) (= vertical clipping size) must be specified in 4-line (HD) units.
	In data enable fetch bundle write, the output address must be specified in 32-byte units.
Internal processing	The filter clipping size must be specified as a value equal to or lower than the actual output size of the filter.

61. Internal Voltage Regulator

61.1 Overview

The MCU supports two power supply providing for core voltage (VDD):

- Switching regulator (DCDC)
- External power supply from external power device (External VDD)

Table 61.1 shows the specifications of internal voltage regulator.

DCDC or External VDD is selected with the OFS setting.

- When OFS2.DCDCEN is 1, DCDC is selected (DCDC mode)
- When OFS2.DCDCEN is 0, External VDD is selected (External VDD mode)

Table 61.1 Power supply selection

OFS setting (OFS2.DCDCEN)	Power mode
1	DCDC mode
0	External VDD mode

Note: External VDD mode is not supported for products in the HLQFP package.

61.2 Operation

61.2.1 DCDC Mode

Table 61.2 lists the DCDC mode pin settings, and Figure 61.1 shows the DCDC mode settings.

In DCDC mode, VDD is supplied from VCL through VLO output and an external inductor and capacitor.

Table 61.2 DCDC mode pin settings

Pins	Setting descriptions
All VCC_n and VCC2_n pins	• Connect each pin to the system power supply. • Connect each pin to the same numbered VSS_n through a 0.1-μF multilayer ceramic capacitor. Place the capacitor close to the pin.
VCC_DCDC pins	• Connect the pin to the system power supply. • Connect the pin to VSS_DCDC through a 22-μF and 0.1-μF multilayer ceramic capacitor in parallel. Place the capacitor close to the pin.
VCLn pins	• Connect each pin to the same numbered VSSn through a 0.22-μF multilayer ceramic capacitor. Place the capacitor close to the pin. • Connect each pin to an external inductor and capacitor. Place the inductor and capacitor close to the pin. See Figure 61.1.
All VLO pins	Connect each pin to an external inductor and capacitor. Place the 2.2-μH inductor and 47-μF capacitor close to the pin. This capacitor must be grounded to VSS_DCDC

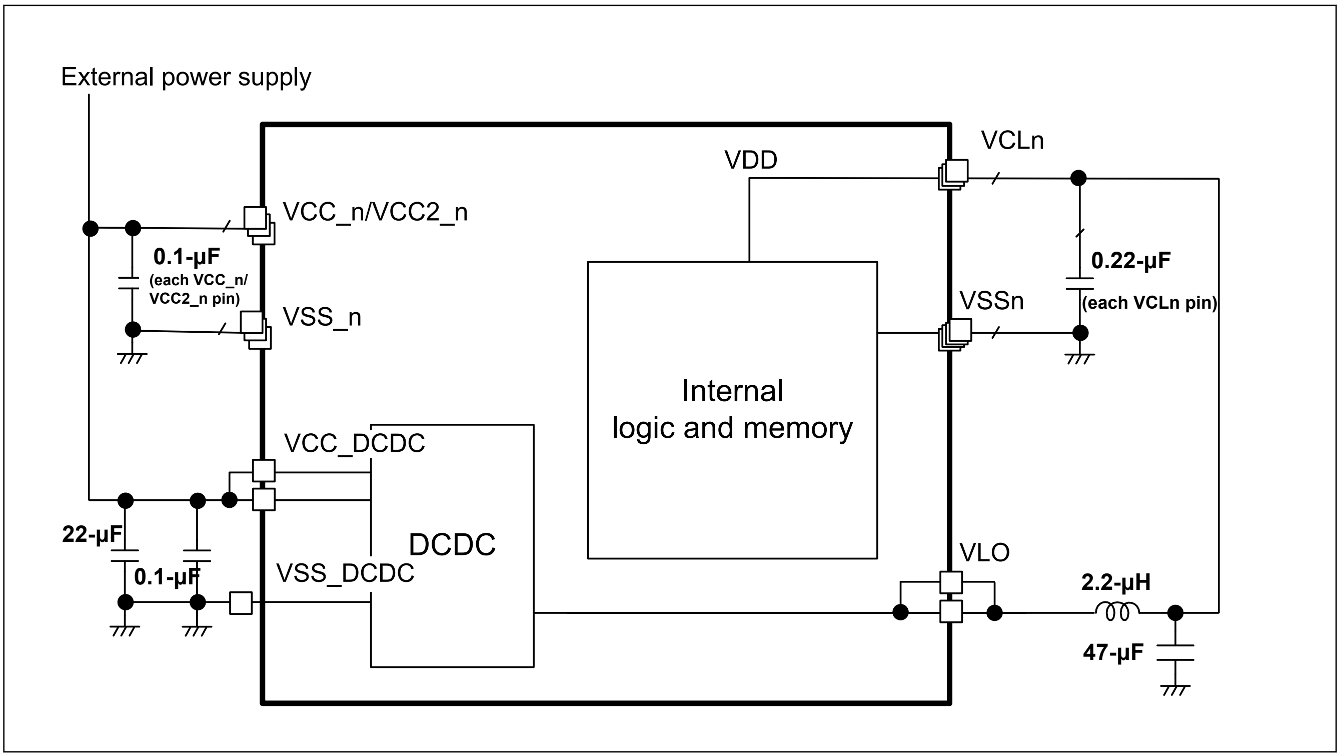


Figure 61.1 DCDC mode setting

61.2.2 External VDD Mode

Table 61.3 lists the External VDD mode pin settings, and Figure 61.2 shows the External VDD mode settings.

VDD is supplied from VCL pins.

Note: Software Standby mode, Deep Software Standby mode 1, 2, 3, Battery Backup Function and Voltage Scaling Control are not supported in this mode.

Table 61.3 Setting descriptions for External VDD mode

Pins	Setting descriptions
All VCC_n and VCC2_n pins	• Connect each pin to the system power supply device for VCC. • Connect each pin to the same numbered VSS_n through a 0.1-μF multilayer ceramic capacitor. Place the capacitor close to the pin.
VCC_DCDC pins	• Connect the pin to VSS_DCDC through 0.1-μF multilayer ceramic capacitor. Place the capacitor close to the pin.
VCLn pins	• Connect each pin to the system power supply device for VDD. • Connect each pin to the same numbered VSSn through a 0.22-μF multilayer ceramic capacitor. Place the capacitor close to the pin.
All VLO pins	Keep pin open

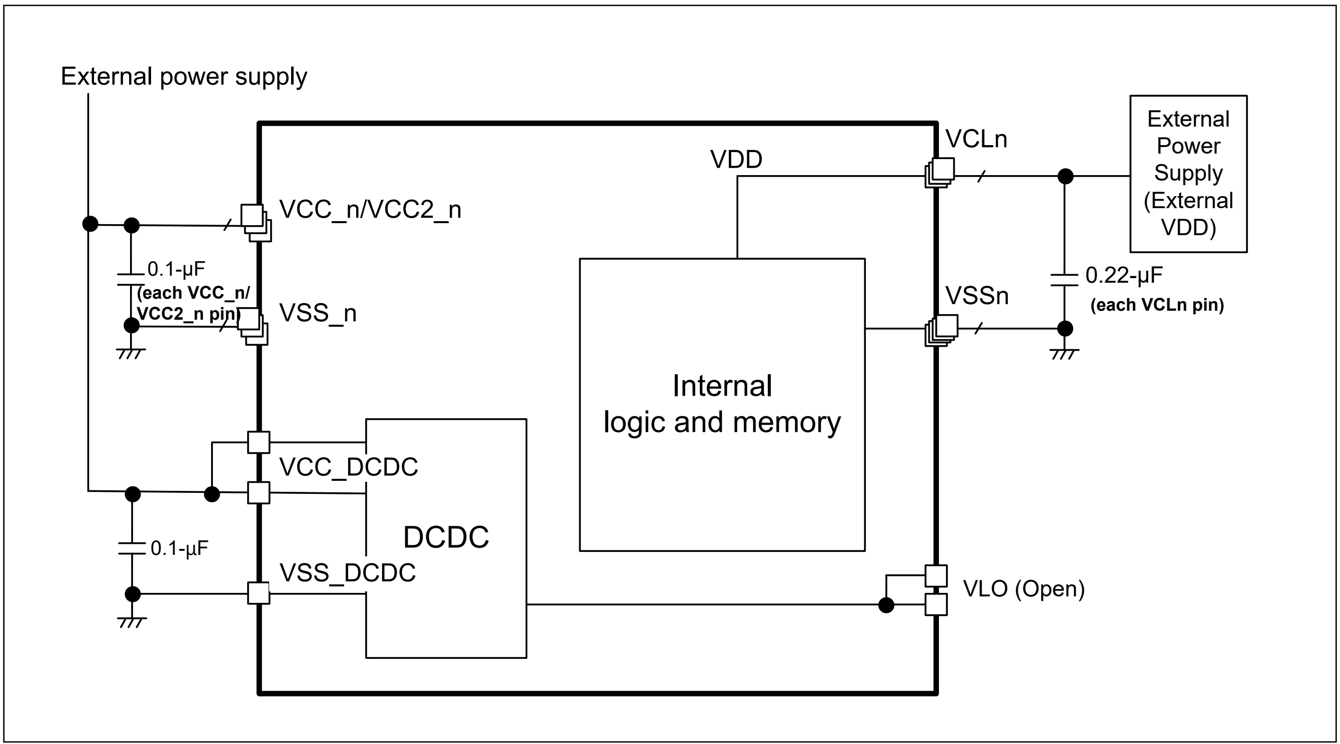


Figure 61.2 External VDD mode setting

61.3 Usage Notes

In DCDC mode, to minimize power loss and maximize efficiency, Renesas recommends using a 2.2-µH inductor with a DC resistance of 100 mΩ or less. Short VCC_n and VCC_DCDC.

In external VDD mode, the voltage of VDD should always be below the voltage of VCC including the power-on and power-off sequence. For details, see section 62, Electrical Characteristics.

62. Electrical Characteristics

Unless otherwise specified, minimum and maximum values are guaranteed by either design simulation, characterization results or test in production.

Supported peripheral functions and pins differ from one product name to another.

Unless otherwise specified, the electrical characteristics of the MCU are defined under the following conditions:

- VCC = VCC_DCDC = VBATT = 1.62 to 3.63 V
- VCC2 = 1.62 to 3.63 V for Standard Product
- VCC2 = 1.70 to 2.00 V for SiP Product
- AVCC0 = 1.62 to 3.63 V
- VCC_USB = VCC_USBHS = AVCC_USBHS = 3.0 to 3.6 V
- VREFH0/VREFH = 1.62 V to AVCC0
- VSS = VSS_DCDC = AVSS0 = VREFL0/VREFL = VSS_USB = VSS1_USBHS = VSS2_USBHS = 0 V
- VCC voltage is lower than 2.7 V : LVOCR.LVO0E = 1, otherwise LVOCR.LVO0E = 0
- VCC2 voltage is lower than 2.7 V : LVOCR.LVO1E = 1, otherwise LVOCR.LVO1E = 0
- $T_j = T_{opj}$
  When not specified otherwise, typical values are measured at room temperature of 25 °C and VCC = VCC_DCDC = VCC_USB = VBATT = VCC_USBHS = AVCC_USBHS = AVCC0 = VREFH0 = VREFH = 3.3V

Figure 62.1 shows the timing conditions.

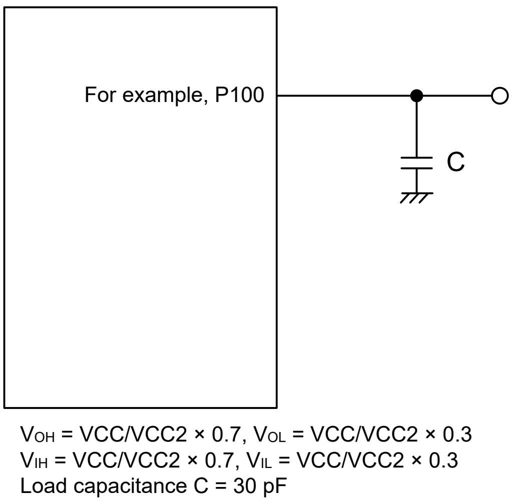


Figure 62.1 Input or output timing measurement conditions

The recommended measurement conditions for the timing specification of each peripheral provided are for the best peripheral operation. Make sure to adjust the driving abilities of each pin to meet your conditions.

62.1 Absolute Maximum Ratings

Table 62.1 Absolute maximum ratings (1 of 2)

Parameter	Symbol		Value	Unit
Power supply voltage	VCC, VCC_DCDC*2		−0.3 to +4.0	V
	VCC2	Standard Product	−0.3 to +4.0	V
		SiP Product	−0.3 to +2.5	V

Table 62.1 Absolute maximum ratings (2 of 2)

Parameter	Symbol	Value	Unit
External power supply voltage [*6]	VCL	−0.3 to +1.2	V
VBATT power supply voltage	VBATT	−0.3 to +4.0	V
Input voltage (except for 5 V-tolerant ports [*1])	V_{in}	−0.3 to VCC + 0.3, −0.3 to VCC2 + 0.3, −0.3 to VCC_USB + 0.3 or −0.3 to VBATT_R + 0.3	V
Input voltage (5 V-tolerant ports [*1])	V_{in}	−0.3 to + VCC + 4.0 (max. 5.8) or −0.3 to + VCC2 + 4.0 (max. 5.8)	V
Reference power supply voltage	VREFH/VREFH0	−0.3 to AVCC0 + 0.3	V
USBFS power supply voltage	VCC_USB	−0.3 to +4.0	V
USBHS power supply voltage	VCC_USBHS	−0.3 to +4.0	V
USBHS analog power supply voltage	AVCC_USBHS	−0.3 to +4.0	V
Analog power supply voltage	AVCC0	−0.3 to +4.0	V
Analog input voltage	V_{AN}	−0.3 to AVCC0 + 0.3	V
Operating junction temperature [*3 *4 *5]	T_{opj}	0 to 95 or −40 to +105 or −40 to +125	°C
Storage temperature	T_{stg}	−55 to +125	°C

Note 1. Ports P204, P205, P303, P407 to P413, P511, P512, P514, P515 and P708 to P715 are 5 V tolerant.
Note 2. Connect VCC_DCDC to VCC.
Note 3. See section 62.2.1. Tj/Ta Definition.
Note 4. Contact a Renesas Electronics sales office for information on derating operation when Tj = +95 °C to +125 °C. Derating is the systematic reduction of load for improved reliability.
Note 5. The lower and upper limit of operating junction temperature depend on the product. For details, see section 1.3. Part Numbering.
Note 6. External VDD mode does not support in HLQFP package.

Caution: Permanent damage to the MCU might result if absolute maximum ratings are exceeded.

Table 62.2 Recommended operating conditions

Parameter	Symbol			Min	Typ	Max	Unit
Power supply voltages	VCC, VCC_DCDC	Other than the following		1.62	—	3.63	V
		When ESWM is used		2.30	—	3.63	V
		When SDRAM is used		3.00	—	3.63	V
	VCC2	Standard Product	Other than the following	1.62	—	3.63	V
			When 32bit SDRAM is used	3.00	—	3.63	V
		SiP Product		1.70	—	2.00	V
	VCL	When external VDD is used*2*3	voltage range 1	0.92	—	0.99	V
			voltage range 2	0.87	—	0.99	V
		When DCDC is used (High-speed mode)	VSCR_1	—	0.95	—	V
			VSCR_2	—	0.925	—	V
		When DCDC is used (Software Standby mode)	SVSCR_1	—	0.95	—	V
			SVSCR_2	—	0.925	—	V
			SVSCR_3	—	0.825	—	V
			SVSCR_4	—	0.765	—	V
			SVSCR_5	—	0.715	—	V
	VSS, VSS_DCDC			—	0	—	V
USB power supply voltages	VCC_USB, VCC_USBHS, AVCC_USBHS	When USB is not used		1.62	—	3.63	V
		When USB is used		3.00	—	3.60	V
	VSS_USB, VSS1_USBHS, VSS2_USBHS			—	0	—	V
VBATT power supply voltage	VBATT			1.62	—	3.63	V
Analog power supply voltages	AVCC0*1		Other than the following	1.62	—	3.63	V
			When Channel dedicated sample-and-hold circuit is used	2.70	—	3.63	V
	AVSS0			—	0	—	V

Note 1. When the A/D converter, the D/A converter and the High-Speed Analog Comparator are not in use, do not leave the AVCC0, VREFH/VREFH0, AVSS0, and VREFL/VREFL0 pins open. Connect the AVCC0 and VREFH/VREFH0 pins to VCC, and the AVSS0 and VREFL/VREFL0 pins to VSS, respectively.

Note 2. VCL voltage must never be higher than VCC voltage.

Note 3. External VDD mode does not support in HLQFP package.

62.2 DC Characteristics

62.2.1 Tj/Ta Definition

Table 62.3 DC characteristics

Parameter	Symbol	Typ	Max	Unit	Test conditions
Permissible operating junction temperature	T_j	—	125*1	°C	High-speed mode

Note: Make sure that $T_j = T_a + \theta_{ja} \times$ total power consumption (W), where total power consumption = $(VCC - V_{OH}) \times \Sigma I_{OH} + V_{OL} \times \Sigma I_{OL} + (I_{CC}max + I_{CC\_DCDC}max) \times VCC$.

Note: Minimum Ambient Temperature (Ta) is -40℃ or 0℃, depending on the product. For details, see section 1.3. Part Numbering.

Note 1. The upper limit of operating junction temperature is 95°C, 105°C or 125°C, depending on the product. For details, see section 1.3. Part Numbering.

62.2.2 I/O V_{IH}, V_{IL}

Table 62.4 I/O V_{IH}, V_{IL} except for Schmitt trigger input pins (1 of 2)

Parameter		VCC/VCC2/ AVCC0/ VCC_USB	Symbol	Min	Typ	Max	Unit
Peripheral function pin	EXTAL (external clock input), WAIT, SPI*1 (except RSPCK)	1.62 V or above	V_{IH}	VCC × 0.8	—	—	V
			V_{IL}	—	—	VCC × 0.2	
	SPI*2 (except RSPCK)	1.62 V or above	V_{IH}	VCC2 × 0.8	—	—	
			V_{IL}	—	—	VCC2 × 0.2	
	OSPI (except OM_0_RSTO1, OM_0_ECSINT 1, OM_1_RSTO1 and OM_1_ECSINT 1)	2.70 V or above	V_{IH}	VCC2 × 0.8	—	—	
			V_{IL}	—	—	VCC2 × 0.2	
		1.62 to 2.00V	V_{IH}	VCC2 × 0.7	—	VCC2 + 0.3	
			V_{IL}	VSS − 0.3	—	VCC2 × 0.3	
	SD*3	2.70 V or above	V_{IH}	VCC × 0.625	—	VCC + 0.3	
			V_{IL}	VSS − 0.3	—	VCC × 0.25	
		1.70 to 1.95 V	V_{IH}	1.27	—	2	
			V_{IL}	VSS − 0.3	—	0.58	
	SD*4	2.70 V or above	V_{IH}	VCC2 × 0.625	—	VCC2 + 0.3	
			V_{IL}	VSS − 0.3	—	VCC2 × 0.25	
		1.70 to 1.95 V	V_{IH}	1.27	—	2	
			V_{IL}	VSS − 0.3	—	0.58	
	MMC*5	2.70 V or above	V_{IH}	VCC × 0.625	—	VCC + 0.3	
			V_{IL}	VSS − 0.3	—	VCC × 0.25	
		1.70 to 1.95 V	V_{IH}	VCC × 0.65	—	VCC + 0.3	
			V_{IL}	VSS − 0.3	—	VCC × 0.35	
	MMC*6	2.70 V or above	V_{IH}	VCC2 × 0.625	—	VCC2 + 0.3	
			V_{IL}	VSS − 0.3	—	VCC2 × 0.25	
		1.70 to 1.95 V	V_{IH}	VCC2 × 0.65	—	VCC2 + 0.3	
			V_{IL}	VSS − 0.3	—	VCC2 × 0.35	
	D00 to D19, TMS, TDI, SWDIO	1.62 V or above	V_{IH}	VCC × 0.7	—	—	
			V_{IL}	—	—	VCC × 0.3	
	D20 to D31	1.62 V or above	V_{IH}	VCC2 × 0.7	—	—	
			V_{IL}	—	—	VCC2 × 0.3	
	DQ00 to DQ19	3.00 V or above	V_{IH}	VCC × 0.7	—	—	
			V_{IL}	—	—	VCC × 0.3	
	DQ20 to DQ31	3.00 V or above	V_{IH}	VCC2 × 0.7	—	—	
			V_{IL}	—	—	VCC2 × 0.3	

Table 62.4 I/O V_{IH}, V_{IL} except for Schmitt trigger input pins (2 of 2)

Parameter		VCC/VCC2/ AVCC0/ VCC_USB	Symbol	Min	Typ	Max	Unit
Peripheral function pin	ESWM*9	2.30 to 3.60 V	V_{IH}	VCC × 0.7	—	—	V
			V_{IL}	—	—	VCC × 0.3	
			V_{IH}	VCC2 × 0.7	—	—	
			V_{IL}	—	—	VCC2 × 0.3	
	ESWM (MII)*10, ESWM (RMII)*11	2.70 to 3.60 V	V_{IH}	2.3	—	—	
			V_{IL}	—	—	VCC × 0.2	
	ESWM (GMII)*10, ESWM (RGMII)*12	3.00 to 3.60 V	V_{IH}	2	—	—	
			V_{IL}	—	—	0.8	
		2.30 to 2.70 V	V_{IH}	1.7	—	—	
			V_{IL}	—	—	0.7	
	IIC (SMBus)*7	2.70 V or above	V_{IH}	2.1	—	VCC + 3.6 (max 5.8)	
			V_{IL}	—	—	0.8	
	IIC (SMBus)*8	2.70 V or above	V_{IH}	2.1	—	VCC2 + 3.6 (max 5.8)	
			V_{IL}	—	—	0.8	
	I3C (SMBus)	2.70 V or above	V_{IH}	2.1	—	VCC + 0.3	
			V_{IL}	—	—	0.8	
	RTCIC0, RTCIC1, RTCIC2, when VCC power supply is selected	1.62 V or above	V_{IH}	0.9	—	3.9	
			V_{IL}	—	—	0.3	
	RTCIC0, RTCIC1, RTCIC2 when VBATT power supply is selected		V_{IH}	0.9	—	3.9	
			V_{IL}	—	—	0.3	
	EXCIN when VCC power supply is selected	1.62 V or above	V_{IH}	0.9	—	VCC	
			V_{IL}	—	—	0.3	
	EXCIN when VBATT power supply is selected		V_{IH}	0.9	—	VBATT	
			V_{IL}	—	—	0.3	

Note 1. SPI0_B, SPI0_C and SPI1_B
Note 2. SPI0_A, SPI1_A
Note 3. SD_A ch0, SD_B ch0, SD_C ch0 and SD_B ch1
Note 4. SD_A ch1
Note 5. MMC_A ch0, MMC_B ch0, MM_C ch0 and MMC_B ch1
Note 6. MMC_A ch1
Note 7. IIC0_A, IIC0_B, IIC1_A, IIC2_A and IIC2_B
Note 8. IIC1_B
Note 9. GPTP_CAPTUREn, ETn_LINKSTA, ETn_MDIO and ETn_INT (n = 0, 1)
Note 10. ETn_RX_CLK, ETn_RX_DV, ETn_RXD7 to ETn_RXD0, ETn_RX_ER and ETn_TX_CLK (n = 0, 1)
Note 11. RMIIn_REF50CK, RMIIn_CRS_DV, RMIIn_RXD1 to RMIIn_RXD0 and RMIIn_RX_ER (n = 0, 1)
Note 12. RGMIIn_RXC, RGMIIn_RX_CTL and RGMIIn_RXD3 to RGMIIn_RXD0 (n = 0, 1)

Table 62.5 I/O V_{IH}, V_{IL} of Schmitt trigger input pins (1 of 2)

Parameter		VCC/VCC2/ AVCC0/ VCC_USB	Symbol	Min	Typ	Max	Unit
Peripheral function pin	IIC (except for SMBus)*7	1.62 V or above	V_{IH}	VCC × 0.7	—	VCC + 3.6 (max 5.8)	V
			V_{IL}	—	—	VCC × 0.3	
			ΔV_T	VCC × 0.05	—	—	
	IIC (except for SMBus)*8	1.62 V or above	V_{IH}	VCC2 × 0.7	—	VCC2 + 3.6 (max 5.8)	
			V_{IL}	—	—	VCC2 × 0.3	
			ΔV_T	VCC2 × 0.05	—	—	
	I3C (except for SMBus)	1.65 V or above	V_{IH}	VCC × 0.7	—	VCC + 0.3	
			V_{IL}	—	—	VCC × 0.3	
			ΔV_T	VCC × 0.1	—	—	
	5 V-tolerant ports*1*6	1.62 V or above	V_{IH}	VCC × 0.8	—	VCC + 3.6 (max 5.8)	
			V_{IL}	—	—	VCC × 0.2	
			ΔV_T	VCC × 0.05	—	—	
	5 V-tolerant ports*2*6	1.62 V or above	V_{IH}	VCC2 × 0.8	—	VCC2 + 3.6 (max 5.8)	
			V_{IL}	—	—	VCC2 × 0.2	
			ΔV_T	VCC2 × 0.05	—	—	
	Other VCC input pins*3	1.62 V or above	V_{IH}	VCC × 0.8	—	—	
			V_{IL}	—	—	VCC × 0.2	
			ΔV_T	VCC × 0.05	—	—	
	Other VCC2 input pins*3	1.62 V or above	V_{IH}	VCC2 × 0.8	—	—	
			V_{IL}	—	—	VCC2 × 0.2	
			ΔV_T	VCC2 × 0.05	—	—	
	Other AVCC0 input pins*3	1.62 V or above	V_{IH}	AVCC0 × 0.8	—	—	
			V_{IL}	—	—	AVCC0 × 0.2	
			ΔV_T	AVCC0 × 0.05	—	—	
	Other VCC_USB input pins*3	1.62 V or above	V_{IH}	VCC_USB × 0.8	—	—	
			V_{IL}	—	—	VCC_USB × 0.2	
			ΔV_T	VCC_USB × 0.05	—	—	
	Other VBATT_R input pins when VCC power supply is selected*3	1.62 V or above	V_{IH}	VCC × 0.8	—	VCC	
			V_{IL}	—	—	VCC × 0.2	
			ΔV_T	VCC × 0.05	—	—	

Table 62.5 I/O V_{IH}, V_{IL} of Schmitt trigger input pins (2 of 2)

Parameter		VCC/VCC2/ AVCC0/ VCC_USB	Symbol	Min	Typ	Max	Unit
Ports	5 V-tolerant port*4*6	1.62 V or above	V_{IH}	VCC × 0.8	—	VCC + 3.6 (max 5.8)	V
			V_{IL}	—	—	VCC × 0.2	
	5 V-tolerant port*2*6	1.62 V or above	V_{IH}	VCC2 × 0.8	—	VCC2 + 3.6 (max 5.8)	
			V_{IL}	—	—	VCC2 × 0.2	
	Other VCC input pins*5	1.62 V or above	V_{IH}	VCC × 0.8	—	—	
			V_{IL}	—	—	VCC × 0.2	
	Other VCC2 input pins*5	1.62 V or above	V_{IH}	VCC2 × 0.8	—	—	
			V_{IL}	—	—	VCC2 × 0.2	
	Other AVCC0 input pins*5	1.62 V or above	V_{IH}	AVCC0 × 0.8	—	—	
			V_{IL}	—	—	AVCC0 × 0.2	
	Other VCC_USB input pins*5	1.62 V or above	V_{IH}	VCC_USB × 0.8	—	—	
			V_{IL}	—	—	VCC_USB × 0.2	
	Other VBATT_R input pins when VCC power supply is selected*5	1.62 V or above	V_{IH}	VCC × 0.8	—	VCC	
			V_{IL}	—	—	VCC × 0.2	

Note 1. RES and peripheral function pins associated with P303, P407 to P413, P511, P512, P514, P515, P708 to P715 (total 21 pins).
Note 2. P204 and P205 (total 2 pins)
Note 3. All input pins except for the peripheral function pins already described in the table. There is an item for each power supply voltage for each port. Refer to the IO chapter for the power supply of the port.
Note 4. P303, P407 to P413, P511, P512, P514, P515, P708 to P715 (total 20 pins).
Note 5. All input pins except for the ports already described in the table. There is an item for each power supply voltage for each port. Refer to the IO chapter for the power supply of the port.
Note 6. When VCC or VCC2 is less than 1.62 V, the input voltage of 5 V-tolerant ports should be less than 3.6 V, otherwise breakdown may occur because 5 V-tolerant ports are electrically controlled so as not to violate the break down voltage.

62.2.3 I/O I_{OH}, I_{OL}

Table 62.6 I/O I_{OH}, I_{OL} (1 of 5)

Parameter			VCC/VCC2/ AVCC0/ VCC_USB	Symbol	Min	Typ	Max	Unit
Permissible output current (average value per pin)	Ports P000 to P015, P201	—	—	I_{OH}	—	—	−2.0	mA
				I_{OL}	—	—	2.0	mA
	Ports P204, P205, P303, P407 to P413, P511, P512, P514, P515, P708 to P715, PA15 (total 23 pins)	Low drive*1	—	I_{OH}	—	—	−2.0	mA
				I_{OL}	—	—	2.0	mA
		Middle drive*2	—	I_{OH}	—	—	−4.0	mA
				I_{OL}	—	—	4.0	mA
		High drive*3	—	I_{OH}	—	—	−16	mA
				I_{OL}	—	—	20.0	mA
		High-speed high drive*4	—	I_{OH}	—	—	−20	mA
				I_{OL}	—	—	20.0	mA
	Other output pins*5	Low drive*1	—	I_{OH}	—	—	−2.0	mA
				I_{OL}	—	—	2.0	mA
		Middle drive*2	—	I_{OH}	—	—	−4.0	mA
				I_{OL}	—	—	4.0	mA
		High drive*3	—	I_{OH}	—	—	−16	mA
				I_{OL}	—	—	16.0	mA
		High-speed high drive*4	—	I_{OH}	—	—	−20	mA
				I_{OL}	—	—	20.0	mA
Permissible output current (max value per pin)	Ports P000 to P015, P201	—	—	I_{OH}	—	—	−4.0	mA
				I_{OL}	—	—	4.0	mA
	Ports P204, P205, P303, P407 to P413, P511, P512, P514, P515, P708 to P715, PA15 (total 23 pins)	Low drive*1	—	I_{OH}	—	—	−4.0	mA
				I_{OL}	—	—	4.0	mA
		Middle drive*2	—	I_{OH}	—	—	−8.0	mA
				I_{OL}	—	—	8.0	mA
		High drive*3	—	I_{OH}	—	—	−32	mA
				I_{OL}	—	—	40.0	mA
		High-speed high drive*4	—	I_{OH}	—	—	−40	mA
				I_{OL}	—	—	40.0	mA
	Other output pins*5	Low drive*1	—	I_{OH}	—	—	−4.0	mA
				I_{OL}	—	—	4.0	mA
		Middle drive*2	—	I_{OH}	—	—	−8.0	mA
				I_{OL}	—	—	8.0	mA
		High drive*3	—	I_{OH}	—	—	−32	mA
				I_{OL}	—	—	32.0	mA
		High-speed high drive*4	—	I_{OH}	—	—	−40	mA
				I_{OL}	—	—	40.0	mA

Table 62.6 I/O I_{OH}, I_{OL} (2 of 5)

Parameter				VCC/VCC2/ AVCC0/ VCC_USB	Symbol	Min	Typ	Max	Unit
Permissible output current (maxvalue of total of all pins)	Maximum of all output pins	VCC I/O	Ports P411 to P415, P511 to P515, P708 to P715, P805 to P807 (total 21 pins)	1.62 V or above	$\Sigma I_{OH\ (max)}$	—	—	−40	mA
			Ports P212, P213, P400 to P410 (total 13 pins)	1.62 V or above		—	—	−40	
			Ports P700 to P707, PB00 to PB04 (total 13 pins)	1.62 V or above		—	—	−40	
			Ports PB05 to PB07, PD06, PD07 (total 5 pins)	1.62 V or above		—	—	−40	
			Ports P207, PD01 to PD05 (total 6 pins)	1.62 V or above		—	—	−40	
			Ports P904, P910 to P913 (total 5 pins)	1.62 V or above		—	—	−40	
			Ports P206, P304 to P312, P902, P903, P905 to P909 (total 17 pins)	1.62 V or above		—	—	−40	
			Ports P108 to P115, P201, P208 to P211, P300 to P303, P609, P914, P915, PA11 (total 21 pins)	1.62 V or above		—	—	−40	
			Ports P610 to P615, P813, PA04 to PA10, PA12 to PA15 (total 18 pins)	1.62 V or above		—	—	−40	
			Ports P503 to P510, P608, PA00 to PA03, PC11 to PC15, PD00 (total 19 pins)	1.62 V or above		—	—	−40	

Table 62.6 I/O I_{OH}, I_{OL} (3 of 5)

Parameter				VCC/VCC2/ AVCC0/ VCC_USB	Symbol	Min	Typ	Max	Unit
Permissible output current (maxvalue of total of all pins)	Maximum of all output pins	VCC2 I/O	Ports PC00 to PC10 (total 11 pins)	1.62 V or above	$\Sigma I_{OH\ (max)}$	—	—	−40	mA
			Ports P204, P205, P600 to P607 (total 10 pins)	1.62 V or above		—	—	−40	
			Ports P202, P203, P313 to P315, P900, P901 (total 7 pins)	1.62 V or above		—	—	−40	
			Ports P100 to P107, P800, P801 (total 10 pins)	1.62 V or above		—	—	−40	
			Ports P500 to P502, P802 to P804, P808 to P812 (total 11 pins)	1.62 V or above		—	—	−40	
		AVCC0 I/O		1.62 V or above		—	—	-33	
		VCC_USB I/O		1.62 V or above		—	—	-33	

Table 62.6 I/O I_{OH}, I_{OL} (4 of 5)

Parameter				VCC/VCC2/ AVCC0/ VCC_USB	Symbol	Min	Typ	Max	Unit
Permissible output current (maxvalue of total of all pins)	Maximum of all output pins	VCC and VCC2 I/O	Ports P411 to P415, P511 to P515, P708 to P715, P805 to P807 (total 21 pins)	1.62 V or above	$\Sigma I_{OL\ (max)}$	—	—	40	mA
			Ports P212, P213, P400 to P410 (total 13 pins)	1.62 V or above		—	—	40	
			Ports P700 to P707, PB00 to PB04 (total 13 pins)	1.62 V or above		—	—	40	
			Ports PB05 to PB07, PD06, PD07 (total 5 pins)	1.62 V or above		—	—	40	
			Ports P207, PD01 to PD05 (total 6 pins)	1.62 V or above		—	—	40	
			Ports P904, P910 to P913 (total 5 pins)	1.62 V or above		—	—	40	
			Ports P206, P304 to P312, P902, P903, P905 to P909 (total 17 pins)	1.62 V or above		—	—	40	
			Ports P108 to P115, P201, P208 to P211, P300 to P303, P609, P914, P915, PA11 (total 21 pins)	1.62 V or above		—	—	40	
			Ports P610 to P615, P813, PA04 to PA10, PA12 to PA15 (total 18 pins)	1.62 V or above		—	—	40	
			Ports P503 to P510, P608, PA00 to PA03, PC11 to PC15, PD00 (total 19 pins)	1.62 V or above		—	—	40	
			Ports PC00 to PC10 (total 11 pins)	1.62 V or above		—	—	40	
			Ports P204, P205, P600 to P607 (total 10 pins)	1.62 V or above		—	—	40	

Table 62.6 I/O I_{OH}, I_{OL} (5 of 5)

Parameter				VCC/VCC2/ AVCC0/ VCC_USB	Symbol	Min	Typ	Max	Unit
Permissible output current (maxvalue of total of all pins)	Maximum of all output pins	VCC and VCC2 I/O	Ports P202, P203, P313 to P315, P900, P901 (total 7 pins)	1.62 V or above	$\Sigma I_{OL\ (max)}$	—	—	40	mA
			Ports P100 to P107, P800, P801 (total 10 pins)	1.62 V or above		—	—	40	
			Ports P500 to P502, P802 to P804, P808 to P812 (total 11 pins)	1.62 V or above		—	—	40	
		AVCC0 I/O		1.62 V or above		—	—	33	
		VCC_USB I/O		1.62 V or above		—	—	33	

Note 1. This is the value when low driving ability is selected in the Port Drive Capability bit in the PmnPFS register. The selected driving ability except for P400 and P401 is retained in Deep Software Standby mode.
Note 2. This is the value when middle driving ability is selected in the Port Drive Capability bit in the PmnPFS register. The selected driving ability except for P400 and P401 is retained in Deep Software Standby mode.
Note 3. This is the value when high driving ability is selected in the Port Drive Capability bit in the PmnPFS register. The selected driving ability except for P400 and P401 is retained in Deep Software Standby mode.
Note 4. This is the value when high-speed high driving ability is selected in the Port Drive Capability in the PmnPFS register. The selected driving ability is retained in Deep Software Standby mode.
Note 5. Except for P200, P214 and P215, which is an input port.

Caution: To protect the reliability of the MCU, the output current values should not exceed the values in this table. The average output current indicates the average value of current measured during 100 µs.

62.2.4 I/O V_{OH}, V_{OL}, and Other Characteristics

Table 62.7 I/O V_{OH}, V_{OL}, and other characteristics (1 of 3)

Parameter		VCC/ VCC2/ AVCC0/ VCC_USB	Symbol	Min	Typ	Max	Unit	Test conditions
Output voltage	IIC	2.70 V or above	V_{OL}	—	—	0.4	V	I_{OL} = 3.0 mA
			V_{OL}	—	—	0.6		I_{OL} = 6.0 mA
		1.62 V to 1.95 V	V_{OL}	—	—	VCC × 0.2		I_{OL} = 2.0 mA
			V_{OL}	—	—	0.4		I_{OL} = 3.0 mA
			V_{OL}	—	—	0.6*4		I_{OL} = 6.0 mA
			V_{OL}	—	—	VCC2 × 0.2		I_{OL} = 2.0 mA
	IIC*1	2.70 V or above	V_{OL}	—	—	0.4		I_{OL} = 15.0 mA (ICFER.FMPE = 1)
			V_{OL}	—	0.4	—		I_{OL} = 20.0 mA (ICFER.FMPE = 1)
		1.62 V to 1.95 V	V_{OL}	—	—	0.4		I_{OL} = 15.0 mA (ICFER.FMPE = 1)
			V_{OL}	—	0.4	—		I_{OL} = 20.0 mA (ICFER.FMPE = 1)
	I3C	2.70 V or above	V_{OL}	—	—	0.4		I_{OL} = 3.0 mA (PRTS.PRTMD = 1, BFCTL.FMPE = 0, BFCTL.HSME = 0)
			V_{OL}	—	—	0.6		I_{OL} = 6.0 mA (PRTS.PRTMD = 1, BFCTL.FMPE = 0, BFCTL.HSME = 0)
			V_{OL}	—	—	0.4		I_{OL} = 15.0 mA (PRTS.PRTMD = 1, BFCTL.FMPE = 1, BFCTL.HSME = 0)
			V_{OL}	—	0.4	—		I_{OL} = 20.0 mA (PRTS.PRTMD = 1, BFCTL.FMPE = 1, BFCTL.HSME = 0)
		3.00 V or above	V_{OL}	—	—	0.4		I_{OL} = 3.0 mA (PRTS.PRTMD = 1, BFCTL.FMPE = 0, BFCTL.HSME = 1)
			V_{OH}	VCC − 0.27	—	—		I_{OH} = 3.0 mA (PRTS.PRTMD = 0, BFCTL.FMPE = 0, BFCTL.HSME = 0)
			V_{OL}	—	—	0.27		I_{OL} = 3.0 mA (PRTS.PRTMD = 0, BFCTL.FMPE = 0, BFCTL.HSME = 0)
		1.65 V to 1.95 V	V_{OL}	—	—	VCC × 0.2		I_{OL} = 2.0 mA (PRTS.PRTMD = 1, BFCTL.FMPE = 0, BFCTL.HSME = 0)
			V_{OL}	—	—	0.4		I_{OL} = 3.0 mA (PRTS.PRTMD = 1, BFCTL.FMPE = 0, BFCTL.HSME = 0)
			V_{OL}	—	—	0.6		I_{OL} = 6.0 mA (PRTS.PRTMD = 1, BFCTL.FMPE = 0, BFCTL.HSME = 0)
			V_{OL}	—	—	VCC × 0.2		I_{OL} = 2.0 mA (PRTS.PRTMD = 1, BFCTL.FMPE = 1, BFCTL.HSME = 0)
			V_{OL}	—	—	0.4		I_{OL} = 15.0 mA (PRTS.PRTMD = 1, BFCTL.FMPE = 1, BFCTL.HSME = 0)
			V_{OL}	—	0.4	—		I_{OL} = 20.0 mA (PRTS.PRTMD = 1, BFCTL.FMPE = 1, BFCTL.HSME = 0)
			V_{OL}	—	—	VCC × 0.2		I_{OL} = 3.0 mA (PRTS.PRTMD = 1, BFCTL.FMPE = 0, BFCTL.HSME = 1)
			V_{OH}	VCC − 0.27	—	—		I_{OH} = 3.0 mA (PRTS.PRTMD = 0, BFCTL.FMPE = 0, BFCTL.HSME = 0)
			V_{OL}	—	—	0.27		I_{OL} = 3.0 mA (PRTS.PRTMD = 0, BFCTL.FMPE = 0, BFCTL.HSME = 0)

Table 62.7 I/O V_{OH}, V_{OL}, and other characteristics (2 of 3)

Parameter		VCC/ VCC2/ AVCC0/ VCC_USB	Symbol	Min	Typ	Max	Unit	Test conditions
Output voltage	ESWM	2.70 V to 3.60 V	V_{OH}	VCC − 0.5	—	—	V	I_{OH} = −1.0 mA
			V_{OL}	—	—	0.4		I_{OL} = 1.0 mA
			V_{OH}	VCC2 − 0.5	—	—		I_{OH} = −1.0 mA
		2.30 V to 2.70 V	V_{OH}	2	—	—		I_{OH} = −1.0 mA
			V_{OL}	—	—	0.4		I_{OL} = 1.0 mA
	SD	2.70 V or above	V_{OH}	VCC × 0.75	—	—		I_{OH} = −2.0 mA
			V_{OL}	—	—	VCC × 0.125		I_{OL} = 3.0 mA
			V_{OH}	VCC2 × 0.75	—	—		I_{OH} = −2.0 mA
			V_{OL}	—	—	VCC2 × 0.125		I_{OL} = 3.0 mA
		1.70 to 1.95 V	V_{OH}	1.4	—	—		I_{OH} = −2.0 mA
			V_{OL}	—	—	0.45		I_{OL} = 2.0 mA
	MMC	2.70 V or above	V_{OH}	VCC × 0.75	—	—		I_{OH} = −0.1 mA (VCC = 2.7 V)
			V_{OL}	—	—	VCC × 0.125		I_{OL} = 0.1 mA (VCC = 2.7 V)
			V_{OH}	VCC2 × 0.75	—	—		I_{OH} = −0.1mA (VCC2 = 2.7 V)
			V_{OL}	—	—	VCC2 × 0.125		I_{OL} = 0.1 mA (VCC2 = 2.7 V)
		1.70 to 1.95 V	V_{OH}	VCC − 0.45	—	—		I_{OH} = −2.0 mA
			V_{OL}	—	—	0.45		I_{OL} = 2.0 mA
			V_{OH}	VCC2 − 0.45	—	—		I_{OH} = −2.0 mA
	Ports P204, P205, P303, P407 to P413, P511, P512, P514, P515, P708 to P715, PA15 (total of 23 pins)*2	—	V_{OH}	VCC − 1.0	—	—		I_{OH} = −16 mA (VCC = 3.3 V)
			V_{OL}	—	—	1		I_{OL} = 20 mA (VCC = 3.3 V)
			V_{OH}	VCC2 − 1.0	—	—		I_{OH} = −16 mA (VCC2 = 3.3 V)
			V_{OL}	—	—	1		I_{OL} = 20 mA (VCC2 = 3.3 V)
	Other output pins	1.62 V or above	V_{OH}	VCC − 0.5	—	—	V	I_{OH} = −1.0 mA
			V_{OL}	—	—	0.5		I_{OL} = 1.0 mA
			V_{OH}	VCC2 − 0.5	—	—		I_{OH} = −1.0 mA
			V_{OL}	—	—	0.5		I_{OL} = 1.0 mA
			V_{OH}	AVCC0 − 0.5	—	—		I_{OH} = −1.0 mA
			V_{OL}	—	—	0.5		I_{OL} = 1.0 mA
			V_{OH}	VCC_USB – 0.5	—	—		I_{OH} = −1.0 mA
			V_{OL}	—	—	0.5		I_{OL} = 1.0 mA

Table 62.7 I/O V_{OH}, V_{OL}, and other characteristics (3 of 3)

Parameter		VCC/ VCC2/ AVCC0/ VCC_USB	Symbol	Min	Typ	Max	Unit	Test conditions
Input leakage current	RES	1.62 V or above	$\|I_{in}\|$	—	—	5	μA	V_{in} = 0 V V_{in} = 5.5 V
	Port P200, P214, P215	1.62 V or above		—	—	1		V_{in} = 0 V V_{in} = VCC
Three-state leakage current (off state)	5 V-tolerant ports	1.62 V or above	$\|I_{TSI}\|$	—	—	5	μA	V_{in} = 0 V V_{in} = 5.5 V
	Other ports (except for port P200, P214, P215)	1.62 V or above		—	—	1		V_{in} = 0 V V_{in} = VCC, VCC2, AVCC0, VCC_USB
Input pull-up MOS current	Ports P0 to PD	2.70 V or above	I_p	−300	—	−10	μA	VCC, VCC2, AVCC0, VCC_USB = 2.7 to 3.63 V V_{in} = 0 V
		1.62 V or above		−300	—	−5		VCC, VCC2, AVCC0, VCC_USB = 1.62 to 3.63 V V_{in} = 0 V
Pull-up current serving as the SCL current source	I3C*3	3.00 to 3.63 V	I_{cs}	3	—	12	mA	VCC = 3.0 to 3.63 V V_{in} = 0.3 × VCC to 0.7 × VCC
		1.65 to 1.95 V						VCC = 1.65 to 1.95 V V_{in} = 0.3 × VCC to 0.7 × VCC
Input capacitance	Ports P014, P015	—	C_{in}	—	—	16	pF	Vbias = 0 V Vamp = 20 mV f = 1 MHz Ta = 25°C
	Ports P814/ USB_DP, P815/ USB_DM	—		—	—	12		
	Ports P400, P401, P409, P410, P511, P512, P708, P709, USBHS_DP, USBHS_DM	—		—	—	10		
	Other input pins	—		—	—	8		

Note 1. SCL0_A, SDA0_A, SCL1_A, SDA1_A, SCL2_A, SDA2_A (total 6 pins).
Note 2. This is the value when high speed high driving ability is selected in the Port Drive Capability bit in the PmnPFS register. The selected driving ability is retained in Deep Software Standby mode.
Note 3. I3C_SCL0 (1 pin). This is the value when IIC high speed mode is selected.
Note 4. This is the value when high speed high driving ability is selected in the Port Drive Capability bit in the PmnPFS register for the following pins: SDA0_B, SCL0_B, SDA1_B, SCL1_B, SDA2_B, SCL2_B.

62.2.5 Operating and Standby Current

Current value of SiP Flash memory is not included in this section, refer to the datasheet of IS25WX064.

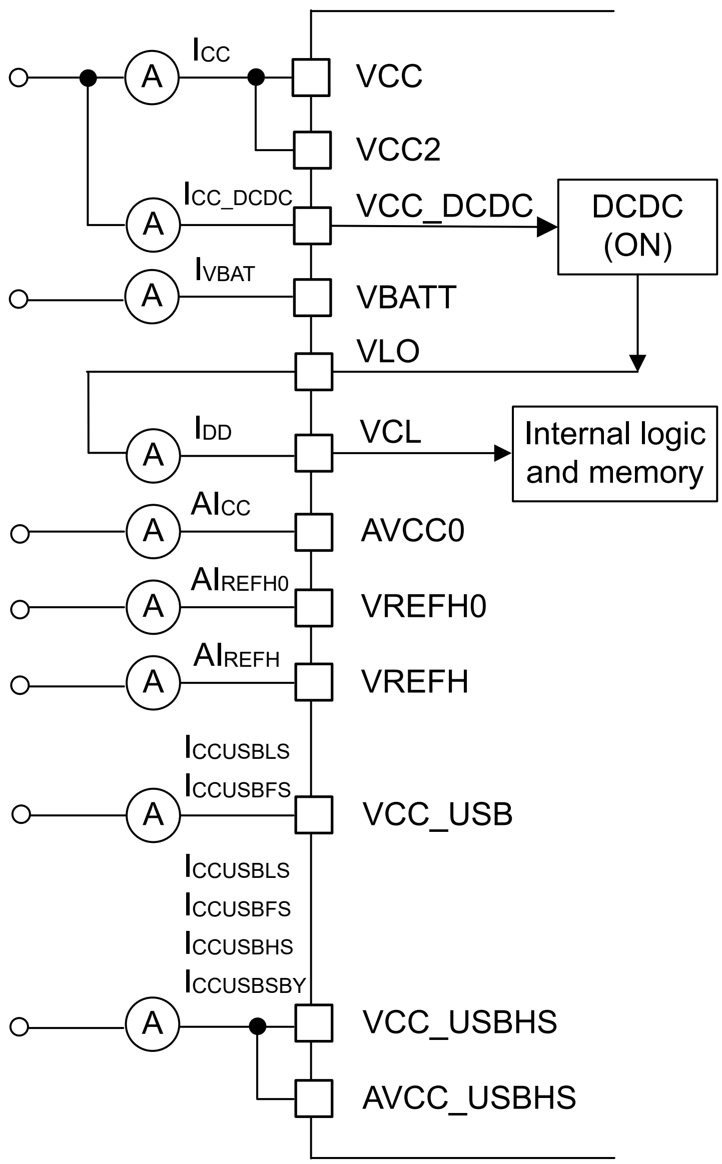


Figure 62.2 Consumption current measurement diagram (DCDC mode)

Table 62.8 Current of high-speed mode, maximum condition (MVE and peripheral operation) (DCDC mode)

Parameter			Symbol	Typ	Max			Unit	Test conditions
					95 °C	105 °C	125 °C		
Supply current *1 *2*6	—		I_{CC}	3.85	6.27	6.69	8.01	mA	—
	CPUCLK0 = 1 GHz CPUCLK1 = 250 MHz VSCR_1	VCC_DCDC ≥ 2.5 V	I_{CC_DCDC} *4	176	340	—	—	mA	VCC_DCDC = 3.3 V MRICLK = 250 MHz, MRPCLK = 125 MHz, ICLK = 250 MHz, BCLK = 125 MHz, PCLKA = 125 MHz, PCLKB = 62.5 MHz, PCLKC = 125 MHz, PCLKD = 250 MHz, PCLKE = 250 MHz
			I_{DD} *3	452	873	—	—		
		VCC_DCDC < 2.5 V	I_{CC_DCDC} *4	343	664	—	—		VCC_DCDC = 1.8 V Clock settings are the same as above
			I_{DD}	452	873	—	—		
	CPUCLK0 = 800 MHz CPUCLK1 = 200 MHz VSCR_1	VCC_DCDC ≥ 2.5 V	I_{CC_DCDC} *4	148	—	356	—	mA	VCC_DCDC = 3.3 V MRICLK = 200 MHz, MRPCLK = 100 MHz, ICLK = 200 MHz, BCLK = 100 MHz, PCLKA = 100 MHz, PCLKB = 50 MHz, PCLKC = 100 MHz, PCLKD = 200 MHz, PCLKE = 200 MHz
			I_{DD} *3	380	—	915	—		
		VCC_DCDC < 2.5 V	I_{CC_DCDC} *4	289	—	696	—		VCC_DCDC = 1.8 V Clock settings are the same as above
			I_{DD}	380	—	915	—		
	CPUCLK0 = 600 MHz CPUCLK1 = 200 MHz VSCR_1	VCC_DCDC ≥ 2.5 V	I_{CC_DCDC} *4	129	—	—	390	mA	VCC_DCDC = 3.3 V MRICLK = 200 MHz, MRPCLK = 100 MHz, ICLK = 200 MHz, BCLK = 100 MHz, PCLKA = 100 MHz, PCLKB = 50 MHz, PCLKC = 100 MHz, PCLKD = 200 MHz, PCLKE = 200 MHz
			I_{DD} *3	330	—	—	1000 *5		
		VCC_DCDC < 2.5 V	I_{CC_DCDC} *4	251	—	—	760		VCC_DCDC = 1.8 V Clock settings are the same as above
			I_{DD}	330	—	—	1000 *5		
	CPUCLK0 = 600 MHz CPUCLK1 = 150 MHz VSCR_2	VCC_DCDC ≥ 2.5 V	I_{CC_DCDC} *4	115	282	309	366	mA	VCC_DCDC = 3.3 V MRICLK = 150 MHz, MRPCLK = 75 MHz, ICLK = 150 MHz, BCLK = 75 MHz, PCLKA = 75 MHz, PCLKB = 37.5 MHz, PCLKC = 75 MHz, PCLKD = 150 MHz, PCLKE = 150 MHz
			I_{DD} *3	301	738	810	959		
		VCC_DCDC < 2.5 V	I_{CC_DCDC} *4	224	550	604	715		VCC_DCDC = 1.8 V Clock settings are the same as above
			I_{DD}	301	738	810	959		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Measured with clocks supplied to the peripheral functions and peripherals being operated. This does not include the BGO operation.
Note 3. I_{DD} depends on f (CPUCLK0 and ICLK) as follows. The ICLK items in the formula include CPU1 current.
I_{DD} Typ. = 0.25 × fCPUCLK0 + 0.77 × fICLK + 21 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_1, 95℃) = 0.22 × fCPUCLK0 + 0.80 × fICLK + 477 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_1, 105℃) = 0.22 × fCPUCLK0 + 0.80 × fICLK + 553 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_1, 125℃) = 0.22 × fCPUCLK0 + 0.80 × fICLK + 709 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_2, 95℃) = 0.22 × fCPUCLK0 + 0.93 × fICLK + 464 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_2, 105℃) = 0.22 × fCPUCLK0 + 0.93 × fICLK + 537 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_2, 125℃) = 0.22 × fCPUCLK0 + 0.93 × fICLK + 691 (unit : mA, fCPUCLK0 and fICLK are MHz)
Note 4. Typical DCDC efficiency and the voltage of the test conditions are applied.
Note 5. Do not let actual consumption current during operation exceed the current value described here.
Note 6. The power consumption is calculated as Power = VCC × I_{cc} + VCC_DCDC × I_{cc_DCDC}.

Table 62.9 Current of high-speed mode, maximum condition (MVE and peripheral operation) (External VDD mode)

Parameter		Symbol	Typ	Max			Unit	Test conditions
				95 °C	105 °C	125 °C		
Supply current *1 *2*5	—	I_{CC}	3.85	6.27	6.69	8.01	mA	—
	CPUCLK0 = 1 GHz CPUCLK1 = 250 MHz VCL = voltage range 1	I_{DD} *3	452	873	—	—	mA	MRICLK = 250 MHz, MRPCLK = 125 MHz, ICLK = 250 MHz, BCLK = 125 MHz, PCLKA = 125 MHz, PCLKB = 62.5 MHz, PCLKC = 125MHz, PCLKD = 250 MHz, PCLKE = 250 MHz
	CPUCLK0 = 800 MHz CPUCLK1 = 200 MHz VCL = voltage range 1	I_{DD} *3	380	—	915	—	mA	MRICLK = 200 MHz, MRPCLK = 100 MHz, ICLK = 200 MHz, BCLK = 100 MHz, PCLKA = 100 MHz, PCLKB = 50 MHz, PCLKC = 100 MHz, PCLKD = 200 MHz, PCLKE = 200 MHz
	CPUCLK0 = 600 MHz CPUCLK1 = 200 MHz VCL = voltage range 1	I_{DD} *3	330	—	—	1000 *4	mA	MRICLK = 200 MHz, MRPCLK = 100 MHz, ICLK = 200 MHz, BCLK = 100 MHz, PCLKA = 100 MHz, PCLKB = 50 MHz, PCLKC = 100 MHz, PCLKD = 200 MHz, PCLKE = 200 MHz
	CPUCLK0 = 600 MHz CPUCLK1 = 150 MHz VCL = voltage range 2	I_{DD} *3	301	738	810	959	mA	MRICLK = 150 MHz, MRPCLK = 75 MHz, ICLK = 150 MHz, BCLK = 75 MHz, PCLKA = 75 MHz, PCLKB = 37.5 MHz, PCLKC = 75 MHz, PCLKD = 150 MHz, PCLKE = 150 MHz

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.

Note 2. Measured with clocks supplied to the peripheral functions and peripherals being operated. This does not include the BGO operation.

Note 3. I_{DD} depends on f (CPUCLK0 and ICLK) as follows. The ICLK items in the formula include CPU1 current.

I_{DD} Typ. = 0.25 × fCPUCLK0 + 0.77 × fICLK + 21 (unit : mA, fCPUCLK0 and fICLK are MHz)

I_{DD} Max.(VCL = voltage range 1, 95℃) = 0.22 × fCPUCLK0 + 0.80 × fICLK + 477 (unit : mA, fCPUCLK0 and fICLK are MHz)

I_{DD} Max.(VCL = voltage range 1, 105℃) = 0.22 × fCPUCLK0 + 0.80 × fICLK + 553 (unit : mA, fCPUCLK0 and fICLK are MHz)

I_{DD} Max.(VCL = voltage range 1, 125℃) = 0.22 × fCPUCLK0 + 0.80 × fICLK + 709 (unit : mA, fCPUCLK0 and fICLK are MHz)

I_{DD} Max.(VCL = voltage range 2, 95℃) = 0.22 × fCPUCLK0 + 0.93 × fICLK + 464 (unit : mA, fCPUCLK0 and fICLK are MHz)

I_{DD} Max.(VCL = voltage range 2, 105℃) = 0.22 × fCPUCLK0 + 0.93 × fICLK + 537 (unit : mA, fCPUCLK0 and fICLK are MHz)

I_{DD} Max.(VCL = voltage range 2, 125℃) = 0.22 × fCPUCLK0 + 0.93 × fICLK + 691 (unit : mA, fCPUCLK0 and fICLK are MHz)

Note 4. Do not let actual consumption current during operation exceed the current value described here.

Note 5. The power consumption is calculated as Power = VCC × I_{cc} + VCL × I_{DD}.

Table 62.10 Current of high-speed mode, maximum condition (MVE and peripheral operation), CPU0 active, CPU1 Deep Sleep (DCDC mode)

Parameter			Symbol	Typ	Max 95 °C	Max 105 °C	Max 125 °C	Unit	Test conditions
Supply current *1*2*5	—		I_{CC}	3.85	6.27	6.69	8.01	mA	—
	CPUCLK0 = 1 GHz CPUCLK1 = 250 MHz VSCR_1	VCC_DCDC ≥ 2.5V	I_{CC_DCDC} *4	168	329	—	—	mA	VCC_DCDC = 3.3 V MRICLK = 250 MHz, MRPCLK = 125 MHz, ICLK = 250 MHz, BCLK = 125 MHz, PCLKA = 125 MHz, PCLKB = 62.5 MHz, PCLKC = 125 MHz, PCLKD = 250 MHz, PCLKE = 250 MHz CPU1 = Deep Sleep
			I_{DD} *3	430	844	—	—		
		VCC_DCDC < 2.5V	I_{CC_DCDC} *4	327	642	—	—		VCC_DCDC = 1.8 V Clock settings are the same as above
			I_{DD}	430	844	—	—		
	CPUCLK0 = 800 MHz CPUCLK1 = 200 MHz VSCR_1	VCC_DCDC ≥ 2.5V	I_{CC_DCDC} *4	141	—	347	—	mA	VCC_DCDC = 3.3 V MRICLK = 200 MHz, MRPCLK = 100 MHz, ICLK = 200 MHz, BCLK = 100 MHz, PCLKA = 100MHz, PCLKB = 50 MHz, PCLKC = 100 MHz, PCLKD = 200 MHz, PCLKE = 200 MHz CPU1 = Deep Sleep
			I_{DD} *3	362	—	891	—		
		VCC_DCDC < 2.5V	I_{CC_DCDC} *4	275	—	678	—		VCC_DCDC = 1.8 V Clock settings are the same as above
			I_{DD}	362	—	891	—		
	CPUCLK0 = 600 MHz CPUCLK1 = 200 MHz VSCR_1	VCC_DCDC ≥ 2.5V	I_{CC_DCDC} *4	122	—	—	380	mA	VCC_DCDC = 3.3 V MRICLK = 200 MHz, MRPCLK = 100 MHz, ICLK = 200 MHz, BCLK = 100 MHz, PCLKA = 100 MHz, PCLKB = 50 MHz, PCLKC = 100 MHz, PCLKD = 200 MHz, PCLKE = 200 MHz CPU1 = Deep Sleep
			I_{DD} *3	312	—	—	976		
		VCC_DCDC < 2.5V	I_{CC_DCDC} *4	237	—	—	742		VCC_DCDC = 1.8 V Clock settings are the same as above
			I_{DD}	312	—	—	976		
	CPUCLK0 = 600 MHz CPUCLK1 = 150 MHz VSCR_2	VCC_DCDC ≥ 2.5V	I_{CC_DCDC} *4	110	274	301	359	mA	VCC_DCDC = 3.3 V MRICLK = 150 MHz, MRPCLK = 75 MHz, ICLK = 150 MHz, BCLK = 75 MHz, PCLKA = 75 MHz, PCLKB = 37.5 MHz, PCLKC = 75 MHz, PCLKD = 150 MHz, PCLKE = 150 MHz CPU1 = Deep Sleep
			I_{DD} *3	289	718	788	940		
		VCC_DCDC < 2.5V	I_{CC_DCDC} *4	216	536	588	701		VCC_DCDC = 1.8 V Clock settings are the same as above
			I_{DD}	289	718	788	940		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Measured with clocks supplied to the peripheral functions and peripherals being operated. This does not include the BGO operation.
Note 3. I_{DD} depends on f (CPUCLK0 and ICLK) as follows.
I_{DD} Typ. = 0.25 × fCPUCLK0 + 0.71 × fICLK + 21 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_1, 95℃) = 0.22 × fCPUCLK0 + 0.70 × fICLK + 474 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_1, 105℃) = 0.22 × fCPUCLK0 + 0.70 × fICLK + 549 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_1, 125℃) = 0.22 × fCPUCLK0 + 0.70 × fICLK + 704 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_2, 95℃) = 0.22 × fCPUCLK0 + 0.83 × fICLK + 461 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_2, 105℃) = 0.22 × fCPUCLK0 + 0.83 × fICLK + 533 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_2, 125℃) = 0.22 × fCPUCLK0 + 0.83 × fICLK + 686 (unit : mA, fCPUCLK0 and fICLK are MHz)
Note 4. Typical DCDC efficiency and the voltage of the test conditions are applied.
Note 5. The power consumption is calculated as Power = VCC × I_{cc} + VCC_DCDC × I_{cc_DCDC}.

Table 62.11 Current of high-speed mode, maximum condition (MVE and peripheral operation), CPU0 active, CPU1 Deep Sleep (External VDD mode)

Parameter		Symbol	Typ	Max 95 °C	Max 105 °C	Max 125 °C	Unit	Test conditions
Supply current *1*2*4	—	I_{CC}	3.85	6.27	6.69	8.01	mA	—
	CPUCLK0 = 1 GHz CPUCLK1 = 250 MHz VCL = voltage range 1	I_{DD} *3	430	844	—	—	mA	MRICLK = 250 MHz, MRPCLK = 125 MHz, ICLK = 250 MHz, BCLK = 125 MHz, PCLKA = 125 MHz, PCLKB = 62.5 MHz, PCLKC = 125 MHz, PCLKD = 250 MHz, PCLKE = 250 MHz CPU1 = Deep Sleep
	CPUCLK0 = 800 MHz CPUCLK1 = 200 MHz VCL = voltage range 1	I_{DD} *3	362	—	891	—	mA	MRICLK = 200 MHz, MRPCLK = 100 MHz, ICLK = 200 MHz, BCLK = 100 MHz, PCLKA = 100 MHz, PCLKB = 50 MHz, PCLKC = 100 MHz, PCLKD = 200 MHz, PCLKE = 200 MHz CPU1 = Deep Sleep
	CPUCLK0 = 600 MHz CPUCLK1 = 200 MHz VCL = voltage range 1	I_{DD} *3	312	—	—	976	mA	MRICLK = 200 MHz, MRPCLK = 100 MHz, ICLK = 200 MHz, BCLK = 100 MHz, PCLKA = 100 MHz, PCLKB = 50 MHz, PCLKC = 100 MHz, PCLKD = 200 MHz, PCLKE = 200 MHz CPU1 = Deep Sleep
	CPUCLK0 = 600 MHz CPUCLK1 = 150 MHz VCL = voltage range 2	I_{DD} *3	289	718	788	940	mA	MRICLK = 150 MHz, MRPCLK = 75 MHz, ICLK = 150 MHz, BCLK = 75 MHz, PCLKA = 75 MHz, PCLKB = 37.5 MHz, PCLKC = 75 MHz, PCLKD = 150 MHz, PCLKE = 150 MHz CPU1 = Deep Sleep

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Measured with clocks supplied to the peripheral functions and peripherals being operated. This does not include the BGO operation.
Note 3. I_{DD} depends on f (CPUCLK0 and ICLK) as follows.
I_{DD} Typ. = 0.25 × fCPUCLK0 + 0.71 × fICLK + 21 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 1, 95℃) = 0.22 × fCPUCLK0 + 0.70 × fICLK + 474 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 1, 105℃) = 0.22 × fCPUCLK0 + 0.70 × fICLK + 549 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 1, 125℃) = 0.22 × fCPUCLK0 + 0.70 × fICLK + 704 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 2, 95℃) = 0.22 × fCPUCLK0 + 0.83 × fICLK + 461 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 2, 105℃) = 0.22 × fCPUCLK0 + 0.83 × fICLK + 533 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 2, 125℃) = 0.22 × fCPUCLK0 + 0.83 × fICLK + 686 (unit : mA, fCPUCLK0 and fICLK are MHz)
Note 4. The power consumption is calculated as Power = VCC × I_{cc} + VCL x I_{DD}.

Table 62.12 Current of high-speed mode, maximum data processing (MVE operation), peripheral clock ON (DCDC mode)

Parameter		Symbol	Typ	Max			Unit	Test conditions
				95 °C	105 °C	125 °C		
Supply current [*1*2]	CPUCLK0 = 1 GHz CPUCLK1 = 250 MHz VSCR_1	I_{CC_DCDC} [*4]	136	322	—	—	mA	VCC_DCDC = 3.3 V[*5]
		I_{DD} [*3]	350	827	—	—		
	CPUCLK0 = 800 MHz CPUCLK1 = 200 MHz VSCR_1	I_{CC_DCDC} [*4]	113	—	340	—		
		I_{DD} [*3]	290	—	874	—		
	CPUCLK0 = 600 MHz CPUCLK1 = 200 MHz VSCR_1	I_{CC_DCDC} [*4]	94	—	—	356		
		I_{DD} [*3]	241	—	—	915		
	CPUCLK0 = 600 MHz CPUCLK1 = 150 MHz VSCR_2	I_{CC_DCDC} [*4]	86	264	305	336		
		I_{DD} [*3]	226	692	798	879		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.

Note 2. Measured with clocks supplied to the peripheral functions and peripherals being operated. This does not include the BGO operation.

Note 3. I_{DD} depends on f (CPUCLK0 and ICLK) as follows. The ICLK items in the formula include CPU1 current.

I_{DD} Typ. = 0.24 × fCPUCLK0 + 0.37 × fICLK + 21 (unit : mA, fCPUCLK0 and fICLK are MHz)

I_{DD} Max.(VSCR_1, 95℃) = 0.27 × fCPUCLK0 + 0.20 × fICLK + 477 (unit : mA, fCPUCLK0 and fICLK are MHz)

I_{DD} Max.(VSCR_1, 105℃) = 0.27 × fCPUCLK0 + 0.20 × fICLK + 553 (unit : mA, fCPUCLK0 and fICLK are MHz)

I_{DD} Max.(VSCR_1, 125℃) = 0.27 × fCPUCLK0 + 0.20 × fICLK + 709 (unit : mA, fCPUCLK0 and fICLK are MHz)

I_{DD} Max.(VSCR_2, 95℃) = 0.27 × fCPUCLK0 + 0.18 × fICLK + 464 (unit : mA, fCPUCLK0 and fICLK are MHz)

I_{DD} Max.(VSCR_2, 105℃) = 0.27 × fCPUCLK0 + 0.18 × fICLK + 537 (unit : mA, fCPUCLK0 and fICLK are MHz)

I_{DD} Max.(VSCR_2, 125℃) = 0.27 × fCPUCLK0 + 0.18 × fICLK + 691 (unit : mA, fCPUCLK0 and fICLK are MHz)

Note 4. Typical DCDC efficiency and the voltage of the test conditions are applied.

Note 5. Same frequency condition is applied as in the maximum condition.

Table 62.13 Current of high-speed mode, maximum data processing (MVE operation), peripheral clock ON (External VDD mode)

Parameter		Symbol	Typ	Max			Unit	Test conditions
				95 °C	105 °C	125 °C		
Supply current *1*2	CPUCLK0 = 1 GHz CPUCLK1 = 250 MHz VCL = voltage range 1	I_{DD} *3	350	827	—	—	mA	*4
	CPUCLK0 = 800 MHz CPUCLK1 = 200 MHz VCL = voltage range 1	I_{DD} *3	290	—	874	—		
	CPUCLK0 = 600 MHz CPUCLK1 = 200 MHz VCL = voltage range 1	I_{DD} *3	241	—	—	915		
	CPUCLK0 = 600 MHz CPUCLK1 = 150 MHz VCL = voltage range 2	I_{DD} *3	226	692	798	879		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.

Note 2. Measured with clocks supplied to the peripheral functions and peripherals being operated. This does not include the BGO operation.

Note 3. I_{DD} depends on f (CPUCLK0 and ICLK) as follows. The ICLK items in the formula include CPU1 current.
I_{DD} Typ. = 0.24 × fCPUCLK0 + 0.37 × fICLK + 21 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 1, 95℃) = 0.27 × fCPUCLK0 + 0.20 × fICLK + 477 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 1, 105℃) = 0.27 × fCPUCLK0 + 0.20 × fICLK + 553 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 1, 125℃) = 0.27 × fCPUCLK0 + 0.20 × fICLK + 709 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 2, 95℃) = 0.27 × fCPUCLK0 + 0.18 × fICLK + 464 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 2, 105℃) = 0.27 × fCPUCLK0 + 0.18 × fICLK + 537 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_2, 125℃) = 0.27 × fCPUCLK0 + 0.18 × fICLK + 691 (unit : mA, fCPUCLK0 and fICLK are MHz)

Note 4. Same frequency condition is applied as in the maximum condition.

Table 62.14 Current of high-speed mode, maximum data processing (MVE operation), CPU0 active, CPU1 Deep Sleep, peripheral clock ON (DCDC mode)

Parameter		Symbol	Typ	Max			Unit	Test conditions
				95 °C	105 °C	125 °C		
Supply current *1*2	CPUCLK0 = 1 GHz CPUCLK1 = 250 MHz VSCR_1	I_{CC_DCDC}*4	131	313	—	—	mA	VCC_DCDC = 3.3 V CPU1 = Deep Sleep *5
		I_{DD}*3	335	804	—	—		
	CPUCLK0 = 800 MHz CPUCLK1 = 200 MHz VSCR_1	I_{CC_DCDC}*4	109	—	332	—		
		I_{DD}*3	279	—	853	—		
	CPUCLK0 = 600 MHz CPUCLK1 = 200 MHz VSCR_1	I_{CC_DCDC}*4	89	—	—	342		
		I_{DD}*3	229	—	—	879		
	CPUCLK0 = 600 MHz CPUCLK1 = 150 MHz VSCR_2	I_{CC_DCDC}*4	83	258	298	328		
		I_{DD}*3	216	676	780	858		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Measured with clocks supplied to the peripheral functions and peripherals being operated. This does not include the BGO operation.
Note 3. I_{DD} depends on f (CPUCLK0 and ICLK) as follows.
I_{DD} Typ. = 0.25 × fCPUCLK0 + 0.30 × fICLK + 21 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_1, 95℃) = 0.27 × fCPUCLK0 + 0.05 × fICLK + 474 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_1, 105℃) = 0.27 × fCPUCLK0 + 0.05 × fICLK + 549 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_1, 125℃) = 0.27 × fCPUCLK0 + 0.05 × fICLK + 704 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_2, 95℃) = 0.27 × fCPUCLK0 + 0.07 × fICLK + 461 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_2, 105℃) = 0.27 × fCPUCLK0 + 0.07 × fICLK + 533 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_2, 125℃) = 0.27 × fCPUCLK0 + 0.07 × fICLK + 686 (unit : mA, fCPUCLK0 and fICLK are MHz)
Note 4. Typical DCDC efficiency and the voltage of the test conditions are applied.
Note 5. Same frequency condition is applied as in the maximum condition.

Table 62.15 Current of high-speed mode, maximum data processing (MVE operation), CPU0 active, CPU1 Deep Sleep, peripheral clock ON (External VDD mode)

Parameter		Symbol	Typ	Max			Unit	Test conditions
				95 °C	105 °C	125 °C		
Supply current *1*2	CPUCLK0 = 1 GHz CPUCLK1 = 250 MHz VCL = voltage range 1	I_{DD}*3	335	804	—	—	mA	CPU1 = Deep Sleep *4
	CPUCLK0 = 800 MHz CPUCLK1 = 200 MHz VCL = voltage range 1	I_{DD}*3	279	—	853	—		
	CPUCLK0 = 600 MHz CPUCLK1 = 200 MHz VCL = voltage range 1	I_{DD}*3	229	—	—	879		
	CPUCLK0 = 600 MHz CPUCLK1 = 150 MHz VCL = voltage range 2	I_{DD}*3	216	676	780	858		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.

Note 2. Measured with clocks supplied to the peripheral functions and peripherals being operated. This does not include the BGO operation.

Note 3. I_{DD} depends on f (CPUCLK0 and ICLK) as follows.

I_{DD} Typ. = 0.25 × fCPUCLK0 + 0.30 × fICLK + 21 (unit : mA, fCPUCLK0 and fICLK are MHz)

I_{DD} Max.(VCL = voltage range 1, 95℃) = 0.27 × fCPUCLK0 + 0.05 × fICLK + 474 (unit : mA, fCPUCLK0 and fICLK are MHz)

I_{DD} Max.(VCL = voltage range 1, 105℃) = 0.27 × fCPUCLK0 + 0.05 × fICLK + 549 (unit : mA, fCPUCLK0 and fICLK are MHz)

I_{DD} Max.(VCL = voltage range 1, 125℃) = 0.27 × fCPUCLK0 + 0.05 × fICLK + 704 (unit : mA, fCPUCLK0 and fICLK are MHz)

I_{DD} Max.(VCL = voltage range 2, 95℃) = 0.27 × fCPUCLK0 + 0.07 × fICLK + 461 (unit : mA, fCPUCLK0 and fICLK are MHz)

I_{DD} Max.(VCL = voltage range 2, 105℃) = 0.27 × fCPUCLK0 + 0.07 × fICLK + 533 (unit : mA, fCPUCLK0 and fICLK are MHz)

I_{DD} Max.(VCL = voltage range 2, 125℃) = 0.27 × fCPUCLK0 + 0.07 × fICLK + 686 (unit : mA, fCPUCLK0 and fICLK are MHz)

Note 4. Same frequency condition is applied as in the maximum condition.

Table 62.16 Current of high-speed mode, maximum data processing, CPU0 Deep Sleep, CPU1 active, peripheral clock ON (DCDC mode)

Parameter		Symbol	Typ	Max			Unit	Test conditions
				95 °C	105 °C	125 °C		
Supply current *1*2	CPUCLK0 = 1 GHz CPUCLK1 = 250 MHz VSCR_1	I_{CC_DCDC} *4	44	175	—	—	mA	VCC_DCDC = 3.3 V CPU0 = Deep Sleep *3
		I_{DD}	112	450	—	—		
	CPUCLK0 = 800 MHz CPUCLK1 = 200 MHz VSCR_1	I_{CC_DCDC} *4	38	—	190	—		
		I_{DD}	99	—	488	—		
	CPUCLK0 = 600 MHz CPUCLK1 = 200 MHz VSCR_1	I_{CC_DCDC} *4	38	—	—	224		
		I_{DD}	96	—	—	545		
	CPUCLK0 = 600 MHz CPUCLK1 = 150 MHz VSCR_2	I_{CC_DCDC} *4	31	155	179	208		
		I_{DD}	82	406	470	545		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Measured with clocks supplied to the peripheral functions and peripherals being operated. This does not include the BGO operation.
Note 3. Same frequency condition is applied as in the maximum condition.
Note 4. Typical DCDC efficiency and the voltage of the test conditions are applied.

Table 62.17 Current of high-speed mode, maximum data processing, CPU0 Deep Sleep, CPU1 active, peripheral clock ON (External VDD mode)

Parameter		Symbol	Typ	Max			Unit	Test conditions
				95 °C	105 °C	125 °C		
Supply current *1*2	CPUCLK0 = 1 GHz CPUCLK1 = 250 MHz VCL = voltage range 1	I_{DD}	112	450	—	—	mA	CPU0 = Deep Sleep *3
	CPUCLK0 = 800 MHz CPUCLK1 = 200 MHz VCL = voltage range 1	I_{DD}	99	—	488	—		
	CPUCLK0 = 600 MHz CPUCLK1 = 200 MHz VCL = voltage range 1	I_{DD}	96	—	—	545		
	CPUCLK0 = 600 MHz CPUCLK1 = 150 MHz VCL = voltage range 2	I_{DD}	82	406	470	545		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Measured with clocks supplied to the peripheral functions and peripherals being operated. This does not include the BGO operation.
Note 3. Same frequency condition is applied as in the maximum condition.

Table 62.18 Current of high-speed mode, maximum data processing (MVE operation), peripheral clock OFF (DCDC mode)

Parameter		Symbol	Typ	Max			Unit	Test conditions
				95 °C	105 °C	125 °C		
Supply current *1*2	CPUCLK0 = 1 GHz CPUCLK1 = 250 MHz VSCR_1	I_{CC_DCDC} *4	121	317	—	—	mA	VCC_DCDC = 3.3 V *5
		I_{DD} *3	310	814	—	—		
	CPUCLK0 = 800 MHz CPUCLK1 = 200 MHz VSCR_1	I_{CC_DCDC} *4	99	—	325	—		
		I_{DD} *3	255	—	833	—		
	CPUCLK0 = 600 MHz CPUCLK1 = 200 MHz VSCR_1	I_{CC_DCDC} *4	80	—	—	358		
		I_{DD} *3	207	—	—	919		
	CPUCLK0 = 600 MHz CPUCLK1 = 150 MHz VSCR_2	I_{CC_DCDC} *4	75	258	292	340		
		I_{DD} *3	197	675	764	889		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.
Note 3. I_{DD} depends on f (CPUCLK0 and ICLK) as follows. The ICLK items in the formula include CPU1 current.
I_{DD} Typ. = 0.24 × fCPUCLK0 + 0.20 × fICLK + 21 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_1, 95℃) = 0.27 × fCPUCLK0 + 0.23 × fICLK + 477 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_1, 105℃) = 0.27 × fCPUCLK0 + 0.23 × fICLK + 553 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_1, 125℃) = 0.27 × fCPUCLK0 + 0.23 × fICLK + 709 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_2, 95℃) = 0.27 × fCPUCLK0 + 0.25 × fICLK + 464 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_2, 105℃) = 0.27 × fCPUCLK0 + 0.25 × fICLK + 537 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_2, 125℃) = 0.27 × fCPUCLK0 + 0.25 × fICLK + 691 (unit : mA, fCPUCLK0 and fICLK are MHz)
Note 4. Typical DCDC efficiency and the voltage of the test conditions are applied.
Note 5. Same frequency condition is applied as in the maximum condition.

Table 62.19 Current of high-speed mode, maximum data processing (MVE operation), peripheral clock OFF (External VDD mode)

Parameter		Symbol	Typ	Max			Unit	Test conditions
				95 °C	105 °C	125 °C		
Supply current *1*2	CPUCLK0 = 1 GHz CPUCLK1 = 250 MHz VCL = voltage range 1	I_{DD}*3	310	814	—	—	mA	*4
	CPUCLK0 = 800 MHz CPUCLK1 = 200 MHz VCL = voltage range 1	I_{DD}*3	255	—	833	—		
	CPUCLK0 = 600 MHz CPUCLK1 = 200 MHz VCL = voltage range 1	I_{DD}*3	207	—	—	919		
	CPUCLK0 = 600 MHz CPUCLK1 = 150 MHz VCL = voltage range 2	I_{DD}*3	197	675	764	889		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.
Note 3. I_{DD} depends on f (CPUCLK0 and ICLK) as follows. The ICLK items in the formula include CPU1 current.
I_{DD} Typ. = 0.24 × fCPUCLK0 + 0.20 × fICLK + 21 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 1, 95℃) = 0.27 × fCPUCLK0 + 0.23 × fICLK + 477 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 1, 105℃) = 0.27 × fCPUCLK0 + 0.23 × fICLK + 553 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 1, 125℃) = 0.27 × fCPUCLK0 + 0.23 × fICLK + 709 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 2, 95℃) = 0.27 × fCPUCLK0 + 0.25 × fICLK + 464 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 2, 105℃) = 0.27 × fCPUCLK0 + 0.25 × fICLK + 537 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 2, 125℃) = 0.27 × fCPUCLK0 + 0.25 × fICLK + 691 (unit : mA, fCPUCLK0 and fICLK are MHz)
Note 4. Same frequency condition is applied as in the maximum condition.

Table 62.20 Current of high-speed mode, maximum data processing (MVE operation), CPU0 active, CPU1 Deep Sleep, peripheral clock OFF (DCDC mode)

Parameter		Symbol	Typ	Max			Unit	Test conditions
				95 °C	105 °C	125 °C		
Supply current *1*2	CPUCLK0 = 1 GHz CPUCLK1 = 250 MHz VSCR_1	I_{CC_DCDC} *4	115	308	—	—	mA	VCC_DCDC = 3.3 V CPU1 = Deep Sleep *5
		I_{DD} *3	295	792	—	—		
	CPUCLK0 = 800 MHz CPUCLK1 = 200 MHz VSCR_1	I_{CC_DCDC} *4	95	—	317	—		
		I_{DD} *3	244	—	813	—		
	CPUCLK0 = 600 MHz CPUCLK1 = 200 MHz VSCR_1	I_{CC_DCDC} *4	76	—	—	348		
		I_{DD} *3	194	—	—	895		
	CPUCLK0 = 600 MHz CPUCLK1 = 150 MHz VSCR_2	I_{CC_DCDC} *4	71	251	285	331		
		I_{DD} *3	187	658	746	868		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.
Note 3. I_{DD} depends on f (CPUCLK0 and ICLK) as follows.
I_{DD} Typ. = 0.25 × fCPUCLK0 + 0.13 × fICLK + 21 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_1, 95℃) = 0.27 × fCPUCLK0 + 0.13 × fICLK + 474 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_1, 105℃) = 0.27 × fCPUCLK0 + 0.13 × fICLK + 549 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_1, 125℃) = 0.27 × fCPUCLK0 + 0.13 × fICLK + 704 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_2, 95℃) = 0.27 × fCPUCLK0 + 0.14 × fICLK + 461 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_2, 105℃) = 0.27 × fCPUCLK0 + 0.14 × fICLK + 533 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_2, 125℃) = 0.27 × fCPUCLK0 + 0.14 × fICLK + 686 (unit : mA, fCPUCLK0 and fICLK are MHz)
Note 4. Typical DCDC efficiency and the voltage of the test conditions are applied.
Note 5. Same frequency condition is applied as in the maximum condition.

Table 62.21 Current of high-speed mode, maximum data processing (MVE operation), CPU0 active, CPU1 Deep Sleep, peripheral clock OFF (External VDD mode)

Parameter		Symbol	Typ	Max			Unit	Test conditions
				95 °C	105 °C	125 °C		
Supply current *1*2	CPUCLK0 = 1 GHz CPUCLK1 = 250 MHz VCL = voltage range 1	I_{DD}*3	295	792	—	—	mA	CPU1 = Deep Sleep *4
	CPUCLK0 = 800 MHz CPUCLK1 = 200 MHz VCL = voltage range 1	I_{DD}*3	244	—	813	—		
	CPUCLK0 = 600 MHz CPUCLK1 = 200 MHz VCL = voltage range 1	I_{DD}*3	194	—	—	895		
	CPUCLK0 = 600 MHz CPUCLK1 = 150 MHz VCL = voltage range 2	I_{DD}*3	187	658	746	868		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.
Note 3. I_{DD} depends on f (CPUCLK0 and ICLK) as follows.
I_{DD} Typ. = 0.25 × fCPUCLK0 + 0.13 × fICLK + 21 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 1, 95℃) = 0.27 × fCPUCLK0 + 0.13 × fICLK + 474 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 1, 105℃) = 0.27 × fCPUCLK0 + 0.13 × fICLK + 549 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 1, 125℃) = 0.27 × fCPUCLK0 + 0.13 × fICLK + 704 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 2, 95℃) = 0.27 × fCPUCLK0 + 0.14 × fICLK + 461 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 2, 105℃) = 0.27 × fCPUCLK0 + 0.14 × fICLK + 533 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 2, 125℃) = 0.27 × fCPUCLK0 + 0.14 × fICLK + 686 (unit : mA, fCPUCLK0 and fICLK are MHz)
Note 4. Same frequency condition is applied as in the maximum condition.

Table 62.22 Current of high-speed mode, maximum data processing, CPU0 Deep Sleep, CPU1 active, peripheral clock OFF (DCDC mode)

Parameter		Symbol	Typ	Max			Unit	Test conditions
				95 °C	105 °C	125 °C		
Supply current *1*2	CPUCLK0 = 1 GHz CPUCLK1 = 250 MHz VSCR_1	I_{CC_DCDC} *4	28	157	—	—	mA	VCC_DCDC = 3.3 V CPU0 = Deep Sleep *3
		I_{DD}	72	404	—	—		
	CPUCLK0 = 800 MHz CPUCLK1 = 200 MHz VSCR_1	I_{CC_DCDC} *4	25	—	174	—		
		I_{DD}	64	—	447	—		
	CPUCLK0 = 600 MHz CPUCLK1 = 200 MHz VSCR_1	I_{CC_DCDC} *4	24	—	—	208		
		I_{DD}	61	—	—	533		
	CPUCLK0 = 600 MHz CPUCLK1 = 150 MHz VSCR_2	I_{CC_DCDC} *4	20	142	166	195		
		I_{DD}	53	372	435	511		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.
Note 3. Same frequency condition is applied as in the maximum condition.
Note 4. Typical DCDC efficiency and the voltage of the test conditions are applied.

Table 62.23 Current of high-speed mode, maximum data processing, CPU0 Deep Sleep, CPU1 active, peripheral clock OFF (External VDD mode)

Parameter		Symbol	Typ	Max			Unit	Test conditions
				95 °C	105 °C	125 °C		
Supply current *1*2	CPUCLK0 = 1 GHz CPUCLK1 = 250 MHz VCL = voltage range 1	I_{DD}	72	404	—	—	mA	CPU0 = Deep Sleep *3
	CPUCLK0 = 800 MHz CPUCLK1 = 200 MHz VCL = voltage range 1	I_{DD}	64	—	447	—		
	CPUCLK0 = 600 MHz CPUCLK1 = 200 MHz VCL = voltage range 1	I_{DD}	61	—	—	533		
	CPUCLK0 = 600 MHz CPUCLK1 = 150 MHz VCL = voltage range 2	I_{DD}	53	372	435	511		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.
Note 3. Same frequency condition is applied as in the maximum condition.

Table 62.24 Current of high-speed mode, CPU Sleep mode (DCDC mode)

Parameter		Symbol	Typ	Max			Unit	Test conditions
				95 °C	105 °C	125 °C		
Supply current *1*3*4	CPUCLK0 = 1 GHz CPUCLK1 = 250 MHz VSCR_1	I_{CC_DCDC} *5	22	200	—	—	mA	VCC_DCDC = 3.3 V
		I_{DD} *2	56	513	—	—		
	CPUCLK0 = 800 MHz CPUCLK1 = 200 MHz VSCR_1	I_{CC_DCDC} *5	20	—	228	—		
		I_{DD} *2	50	—	586	—		
	CPUCLK0 = 600 MHz CPUCLK1 = 200 MHz VSCR_1	I_{CC_DCDC} *5	18	—	—	279		
		I_{DD} *2	46	—	—	716		
	CPUCLK0 = 600 MHz CPUCLK1 = 150 MHz VSCR_2	I_{CC_DCDC} *5	16	185	219	264		
		I_{DD} *2	42	484	573	690		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. I_{DD} depends on f (ICLK) as follows.
I_{DD} Typ. = 0.02 × fCPUCLK0 + 0.06 × fICLK + 57 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_1, 95℃) = 0.02 × fCPUCLK0 + 0.01 × fICLK + 477 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_1, 105℃) = 0.02 × fCPUCLK0 + 0.01 × fICLK + 553 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_1, 125℃) = 0.02 × fCPUCLK0 + 0.01 × fICLK + 709 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_2, 95℃) = 0.02 × fCPUCLK0 + 0.01 × fICLK + 464 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_2, 105℃) = 0.02 × fCPUCLK0 + 0.01 × fICLK + 537 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_2, 125℃) = 0.02 × fCPUCLK0 + 0.01 × fICLK + 691 (unit : mA, fCPUCLK0 and fICLK are MHz)
Note 3. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.
Note 4. MRICLK, MRPCLK, ICLK, PCLKA, PCLKB, PCLKC, PCLKD, PCLKE and BCLK are set to divided by 64.
Note 5. Typical DCDC efficiency and the voltage of the test conditions are applied.

Table 62.25 Current of high-speed mode, CPU Sleep mode (External VDD mode)

Parameter		Symbol	Typ	Max			Unit	Test conditions
				95 °C	105 °C	125 °C		
Supply current *1*3*4	CPUCLK0 = 1 GHz CPUCLK1 = 250 MHz VCL = voltage range 1	I_{DD}*2	56	513	—	—	mA	—
	CPUCLK0 = 800 MHz CPUCLK1 = 200 MHz VCL = voltage range 1	I_{DD}*2	50	—	586	—		
	CPUCLK0 = 600 MHz CPUCLK1 = 200 MHz VCL = voltage range 1	I_{DD}*2	46	—	—	716		
	CPUCLK0 = 600 MHz CPUCLK1 = 150 MHz VCL = voltage range 2	I_{DD}*2	42	484	573	690		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. I_{DD} depends on f (ICLK) as follows.
I_{DD} Typ. = 0.02 × fCPUCLK0 + 0.06 × fICLK + 57 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 1, 95℃) = 0.02 × fCPUCLK0 + 0.01 × fICLK + 477 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 1, 105℃) = 0.02 × fCPUCLK0 + 0.01 × fICLK + 553 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 1, 125℃) = 0.02 × fCPUCLK0 + 0.01 × fICLK + 709 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 2, 95℃) = 0.02 × fCPUCLK0 + 0.01 × fICLK + 464 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 2, 105℃) = 0.02 × fCPUCLK0 + 0.01 × fICLK + 537 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 2, 125℃) = 0.02 × fCPUCLK0 + 0.01 × fICLK + 691 (unit : mA, fCPUCLK0 and fICLK are MHz)
Note 3. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.
Note 4. MRICLK, MRPCLK, ICLK, PCLKA, PCLKB, PCLKC, PCLKD, PCLKE and BCLK are set to divided by 64.

Table 62.26 Current of high-speed mode, CPU0 Sleep, CPU1 Deep Sleep (DCDC mode)

Parameter		Symbol	Typ	Max			Unit	Test conditions
				95 °C	105 °C	125 °C		
Supply current *1*3*4	CPUCLK0 = 1 GHz CPUCLK1 = 250 MHz VSCR_1	I_{CC_DCDC}*5	21	197	—	—	mA	VCC_DCDC = 3.3 V CPU1 = Deep Sleep
		I_{DD}*2	54	505	—	—		
	CPUCLK0 = 800 MHz CPUCLK1 = 200 MHz VSCR_1	I_{CC_DCDC}*5	19	—	229	—		
		I_{DD}*2	48	—	576	—		
	CPUCLK0 = 600 MHz CPUCLK1 = 200 MHz VSCR_1	I_{CC_DCDC}*5	17	—	—	274		
		I_{DD}*2	44	—	—	703		
	CPUCLK0 = 600 MHz CPUCLK1 = 150 MHz VSCR_2	I_{CC_DCDC}*5	15	182	216	259		
		I_{DD}*2	40	477	565	678		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. I_{DD} depends on f (ICLK) as follows.
I_{DD} Typ. = 0.02 × fCPUCLK0 + 0.05 × fICLK + 55 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_1, 95℃) = 0.004 × fCPUCLK0 + 0.01 × fICLK + 474 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_1, 105℃) = 0.004 × fCPUCLK0 + 0.01 × fICLK + 549 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_1, 125℃) = 0.004 × fCPUCLK0 + 0.01 × fICLK + 704 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_2, 95℃) = 0.004 × fCPUCLK0 + 0.01 × fICLK + 461 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_2, 105℃) = 0.004 × fCPUCLK0 + 0.01 × fICLK + 533 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VSCR_2, 125℃) = 0.004 × fCPUCLK0 + 0.01 × fICLK + 686 (unit : mA, fCPUCLK0 and fICLK are MHz)
Note 3. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.
Note 4. MRICLK, MRPCLK, ICLK, PCLKA, PCLKB, PCLKC, PCLKD, PCLKE and BCLK are set to divided by 64.
Note 5. Typical DCDC efficiency and the voltage of the test conditions are applied.

Table 62.27 Current of high-speed mode, CPU0 Sleep, CPU1 Deep Sleep (External VDD mode)

Parameter		Symbol	Typ	Max			Unit	Test conditions
				95 °C	105 °C	125 °C		
Supply current *1*3*4	CPUCLK0 = 1 GHz CPUCLK1 = 250 MHz VCL = voltage range 1	I_{DD}*2	54	505	—	—	mA	CPU1 = Deep Sleep
	CPUCLK0 = 800 MHz CPUCLK1 = 200 MHz VCL = voltage range 1	I_{DD}*2	48	—	576	—		
	CPUCLK0 = 600 MHz CPUCLK1 = 200 MHz VCL = voltage range 1	I_{DD}*2	44	—	—	703		
	CPUCLK0 = 600 MHz CPUCLK1 = 150 MHz VCL = voltage range 2	I_{DD}*2	40	477	565	678		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. I_{DD} depends on f (ICLK) as follows.
I_{DD} Typ. = 0.02 × fCPUCLK0 + 0.05 × fICLK + 55 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 1, 95℃) = 0.004 × fCPUCLK0 + 0.01 × fICLK + 474 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 1, 105℃) = 0.004 × fCPUCLK0 + 0.01 × fICLK + 549 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 1, 125℃) = 0.004 × fCPUCLK0 + 0.01 × fICLK + 704 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 2, 95℃) = 0.004 × fCPUCLK0 + 0.01 × fICLK + 461 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 2, 105℃) = 0.004 × fCPUCLK0 + 0.01 × fICLK + 533 (unit : mA, fCPUCLK0 and fICLK are MHz)
I_{DD} Max.(VCL = voltage range 2, 125℃) = 0.004 × fCPUCLK0 + 0.01 × fICLK + 686 (unit : mA, fCPUCLK0 and fICLK are MHz)
Note 3. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.
Note 4. MRICLK, MRPCLK, ICLK, PCLKA, PCLKB, PCLKC, PCLKD, PCLKE and BCLK are set to divided by 64.

Table 62.28 Current of high-speed mode, CPU0 Deep Sleep, CPU1 Sleep (DCDC mode)

Parameter		Symbol	Typ	Max			Unit	Test conditions
				95 °C	105 °C	125 °C		
Supply current *1*2*3	CPUCLK0 = 1 GHz CPUCLK1 = 250 MHz VSCR_1	I_{CC_DCDC} *4	12	141	—	—	mA	VCC_DCDC = 3.3 V CPU0 = Deep Sleep
		I_{DD}	32	363	—	—		
	CPUCLK0 = 800 MHz CPUCLK1 = 200 MHz VSCR_1	I_{CC_DCDC} *4	12	—	161	—		
		I_{DD}	30	—	414	—		
	CPUCLK0 = 600 MHz CPUCLK1 = 200 MHz VSCR_1	I_{CC_DCDC} *4	11	—	—	196		
		I_{DD}	29	—	—	504		
	CPUCLK0 = 600 MHz CPUCLK1 = 150 MHz VSCR_2	I_{CC_DCDC} *4	10	132	157	184		
		I_{DD}	26	347	411	481		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.
Note 3. MRICLK, MRPCLK, ICLK, PCLKA, PCLKB, PCLKC, PCLKD, PCLKE and BCLK are set to divided by 64.
Note 4. Typical DCDC efficiency and the voltage of the test conditions are applied.

Table 62.29 Current of high-speed mode, CPU0 Deep Sleep, CPU1 Sleep (External VDD mode)

Parameter		Symbol	Typ	Max			Unit	Test conditions
				95 °C	105 °C	125 °C		
Supply current *1*2*3	CPUCLK0 = 1 GHz CPUCLK1 = 250 MHz VCL = voltage range 1	I_{DD}	32	363	—	—	mA	CPU0 = Deep Sleep
	CPUCLK0 = 800 MHz CPUCLK1 = 200 MHz VCL = voltage range 1	I_{DD}	30	—	414	—		
	CPUCLK0 = 600 MHz CPUCLK1 = 200 MHz VCL = voltage range 1	I_{DD}	29	—	—	504		
	CPUCLK0 = 600 MHz CPUCLK1 = 150 MHz VCL = voltage range 2	I_{DD}	26	347	411	481		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.
Note 3. MRICLK, MRPCLK, ICLK, PCLKA, PCLKB, PCLKC, PCLKD, PCLKE and BCLK are set to divided by 64.

Table 62.30 Current of high-speed mode, CPU Deep Sleep mode (DCDC mode)

Parameter		Symbol	Typ	Max			Unit	Test conditions
				95 °C	105 °C	125 °C		
Supply current *1*2*3	CPUCLK0 = 1 GHz CPUCLK1 = 250 MHz VSCR_1	I_{CC_DCDC} *4	13	138	—	—	mA	VCC_DCDC = 3.3 V
		I_{DD}	32	355	—	—		
	CPUCLK0 = 800 MHz CPUCLK1 = 200 MHz VSCR_1	I_{CC_DCDC} *4	12	—	158	—		
		I_{DD}	30	—	405	—		
	CPUCLK0 = 600 MHz CPUCLK1 = 200 MHz VSCR_1	I_{CC_DCDC} *4	11	—	—	192		
		I_{DD}	29	—	—	492		
	CPUCLK0 = 600 MHz CPUCLK1 = 150 MHz VSCR_2	I_{CC_DCDC} *4	11	130	154	182		
		I_{DD}	28	340	402	476		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.
Note 3. MRICLK, MRPCLK, ICLK, PCLKA, PCLKB, PCLKC, PCLKD, PCLKE and BCLK are set to divided by 64.
Note 4. Typical DCDC efficiency and the voltage of the test conditions are applied.

Table 62.31 Current of high-speed mode, CPU Deep Sleep mode (External VDD mode)

Parameter		Symbol	Typ	Max			Unit	Test conditions
				95 °C	105 °C	125 °C		
Supply current *1*2*3	CPUCLK0 = 1 GHz CPUCLK1 = 250 MHz VCL = voltage range 1	I_{DD}	32	355	—	—	mA	—
	CPUCLK0 = 800 MHz CPUCLK1 = 200 MHz VCL = voltage range 1	I_{DD}	30	—	405	—		
	CPUCLK0 = 600 MHz CPUCLK1 = 200 MHz VCL = voltage range 1	I_{DD}	29	—	—	492		
	CPUCLK0 = 600 MHz CPUCLK1 = 150 MHz VCL = voltage range 2	I_{DD}	28	340	402	476		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.
Note 3. MRICLK, MRPCLK, ICLK, PCLKA, PCLKB, PCLKC, PCLKD, PCLKE and BCLK are set to divided by 64.

Table 62.32 Current increase during BGO operation (Programing of MRAM OTP) (DCDC mode and External VDD mode)

Parameter		Symbol	Typ	Max			Unit	Test conditions
				95 °C	105 °C	125 °C		
Supply current*1	normal speed write mode	I_{CC}	—	—	—	20	mA	VCC ≥ 1.62V
		I_{DD}	—	—	—	0.50		
	high speed write mode 0	I_{CC}	—	—	—	25		VCC ≥ 2.5V
		I_{DD}	—	—	—	0.5		
	high speed write mode 1	I_{CC}	—	—	—	80		VCC ≥ 3.0V
		I_{DD}	—	—	—	0.5		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.

Table 62.33 Standby current (DCDC mode) (1 of 4)

Parameter					Symbol	Typ	Max			Unit	Test conditions
							95 °C	105 °C	125 °C		
Supply current*1	Software Standby mode				I_{CC}	0.10	1.11	1.12	1.14	mA	—
		SS2LP_0	SVSCR_1	Data of SRAM and TCM is retained	I_{CC_DCDC}	2.67	54.24	62.23	78.41		VCC_DCDC = 3.3 V PDRAMSCR0.RKEEPn = 1 (n = 0 to 12) PDRAMSCR1.RKEEPn = 1 (n = 0, 1)
				Data of SRAM and TCM is not retained	I_{CC_DCDC}	2.44	52.53	59.86	74.47		VCC_DCDC = 3.3 V PDRAMSCR0.RKEEPn = 0 (n = 0 to 12) PDRAMSCR1.RKEEPn = 0 (n = 0, 1)
			SVSCR_2	Data of SRAM and TCM is retained	I_{CC_DCDC}	2.52	51.58	59.19	74.81		VCC_DCDC = 3.3 V PDRAMSCR0.RKEEPn = 1 (n = 0 to 12) PDRAMSCR1.RKEEPn = 1 (n = 0, 1)
				Data of SRAM and TCM is not retained	I_{CC_DCDC}	2.33	50.03	57.09	71.10		VCC_DCDC = 3.3 V PDRAMSCR0.RKEEPn = 0 (n = 0 to 12) PDRAMSCR1.RKEEPn = 0 (n = 0, 1)
			SVSCR_3	Data of SRAM and TCM is retained	I_{CC_DCDC}	1.68	37.85	43.60	56.69		VCC_DCDC = 3.3 V PDRAMSCR0.RKEEPn = 1 (n = 0 to 12) PDRAMSCR1.RKEEPn = 1 (n = 0, 1)
				Data of SRAM and TCM is not retained	I_{CC_DCDC}	1.60	36.87	42.22	54.07		VCC_DCDC = 3.3 V PDRAMSCR0.RKEEPn = 0 (n = 0 to 12) PDRAMSCR1.RKEEPn = 0 (n = 0, 1)
			SVSCR_4	Data of SRAM and TCM is retained	I_{CC_DCDC}	1.47	32.41	38.22	49.79		VCC_DCDC = 3.3 V PDRAMSCR0.RKEEPn = 1 (n = 0 to 12) PDRAMSCR1.RKEEPn = 1 (n = 0, 1)
				Data of SRAM and TCM is not retained	I_{CC_DCDC}	1.42	31.61	37.07	47.62		VCC_DCDC = 3.3 V PDRAMSCR0.RKEEPn = 0 (n = 0 to 12) PDRAMSCR1.RKEEPn = 0 (n = 0, 1)
			SVSCR_5	Data of SRAM and TCM is retained	I_{CC_DCDC}	1.28	29.69	34.28	44.67		VCC_DCDC = 3.3 V PDRAMSCR0.RKEEPn = 1 (n = 0 to 12) PDRAMSCR1.RKEEPn = 1 (n = 0, 1)
				Data of SRAM and TCM is not retained	I_{CC_DCDC}	1.24	29.03	33.32	42.82		VCC_DCDC = 3.3 V PDRAMSCR0.RKEEPn = 0 (n = 0 to 12) PDRAMSCR1.RKEEPn = 0 (n = 0, 1)

Table 62.33 Standby current (DCDC mode) (2 of 4)

Parameter					Symbol	Typ	Max 95 °C	Max 105 °C	Max 125 °C	Unit	Test conditions
Supply current*1	Software Standby mode	SS2LP_1	SVSCR_2	Data of SRAM and TCM is retained	I_{CC_DCDC}	2.12	43.32	49.71	62.83	mA	VCC_DCDC = 3.3 V PDRAMSCR0.RKEEPn = 1 (n = 0 to 12) PDRAMSCR1.RKEEPn = 1 (n = 0, 1)
				Data of SRAM and TCM is not retained	I_{CC_DCDC}	1.95	42.01	47.94	59.71		VCC_DCDC = 3.3 V PDRAMSCR0.RKEEPn = 0 (n = 0 to 12) PDRAMSCR1.RKEEPn = 0 (n = 0, 1)
			SVSCR_3	Data of SRAM and TCM is retained	I_{CC_DCDC}	1.40	31.70	36.52	47.48		VCC_DCDC = 3.3 V PDRAMSCR0.RKEEPn = 1 (n = 0 to 12) PDRAMSCR1.RKEEPn = 1 (n = 0, 1)
				Data of SRAM and TCM is not retained	I_{CC_DCDC}	1.34	30.88	35.36	45.29		VCC_DCDC = 3.3 V PDRAMSCR0.RKEEPn = 0 (n = 0 to 12) PDRAMSCR1.RKEEPn = 0 (n = 0, 1)
			SVSCR_4	Data of SRAM and TCM is retained	I_{CC_DCDC}	1.22	26.41	31.14	40.57		VCC_DCDC = 3.3 V PDRAMSCR0.RKEEPn = 1 (n = 0 to 12) PDRAMSCR1.RKEEPn = 1 (n = 0, 1)
				Data of SRAM and TCM is not retained	I_{CC_DCDC}	1.18	25.76	30.20	38.80		VCC_DCDC = 3.3 V PDRAMSCR0.RKEEPn = 0 (n = 0 to 12) PDRAMSCR1.RKEEPn = 0 (n = 0, 1)
			SVSCR_5	Data of SRAM and TCM is retained	I_{CC_DCDC}	1.06	24.15	27.89	36.35		VCC_DCDC = 3.3 V PDRAMSCR0.RKEEPn = 1 (n = 0 to 12) PDRAMSCR1.RKEEPn = 1 (n = 0, 1)
				Data of SRAM and TCM is not retained	I_{CC_DCDC}	1.03	23.62	27.11	34.84		VCC_DCDC = 3.3 V PDRAMSCR0.RKEEPn = 0 (n = 0 to 12) PDRAMSCR1.RKEEPn = 0 (n = 0, 1)

Table 62.33 Standby current (DCDC mode) (3 of 4)

Parameter				Symbol	Typ	Max			Unit	Test conditions
						95 °C	105 °C	125 °C		
Supply current*1	Deep Software Standby mode 1			I_{CC}	10.04	207	297	498	μA	—
				I_{CC_DCDC}	0.16	0.85	1.24	2.45		—
		Increase when the function is activated	PVDn (n = 0 to 2, 4, 5) or Battery power supply switch	I_{CC}	See Table 62.36					—
			When the LOCO is in use		2.46	—	—	—		—
			Crystal oscillator and RTC		See Table 62.37					—
			IWDT and ULPT (all units) are operating		1.58	—	—	—		—
Supply current*1	Deep Software Standby mode 2			I_{CC}	3.04	98	122	175	μA	—
				I_{CC_DCDC}	0.16	0.85	1.24	2.45		—
		Increase when the function is activated	PVDn (n = 0 to 2, 4, 5) or Battery power supply switch	I_{CC}	See Table 62.36					—
			Crystal oscillator and RTC		See Table 62.37					—
	Deep Software Standby mode 3			I_{CC}	2.78	97	121	173		—
				I_{CC_DCDC}	0.16	0.85	1.24	2.45		—
		Increase when the function is activated	Crystal oscillator and RTC	I_{CC}	See Table 62.37					—

Table 62.33 Standby current (DCDC mode) (4 of 4)

Parameter				Symbol	Typ	Max			Unit	Test conditions
						95 °C	105 °C	125 °C		
Supply current*1	RTC operating while VCC is off (with the battery backup function, only the RTC operate)	When a crystal oscillator with low power mode 3 is in use		I_{VBAT}	0.53	—	—	—	μA	VBATT = 1.8 V, VCC = 0 V
					0.82	—	—	—		VBATT = 3.3 V, VCC = 0 V
		When a crystal oscillator with low power mode 2 is in use			0.63	—	—	—		VBATT = 1.8 V, VCC = 0 V
					0.94	—	—	—		VBATT = 3.3 V, VCC = 0 V
		When a crystal oscillator with low power mode 1 is in use			0.73	—	—	—		VBATT = 1.8 V, VCC = 0 V
					1.03	—	—	—		VBATT = 3.3 V, VCC = 0 V
		When a crystal oscillator with standard mode is in use			0.99	—	—	—		VBATT = 1.8 V, VCC = 0 V
					1.29	—	—	—		VBATT = 3.3 V, VCC = 0 V
		When EXCIN is in use			0.30	—	—	—		VBATT = 1.8 V, VCC = 0 V
					0.52	—	—	—		VBATT = 3.3 V, VCC = 0 V
		Increase when the function is activated	RTCICn (n = 0 to 2) input is in use per channel		0.01	—	—	—		VBATT = 1.8 V, VCC = 0 V
					0.01	—	—	—		VBATT = 3.3 V, VCC = 0 V

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.

Table 62.34 Coremark and normal mode current, CPU0 active, CPU1 Deep Sleep. (DCDC mode and External VDD mode) (1 of 2)

Parameter				Symbol	Typ	Max			Unit	Test conditions
						95 °C	105 °C	125 °C		
Supply current[*1*2]	CPUCLK0 = 1 GHz VSCR_1 VCL = voltage range 1	Coremark	Cache on	I_{DD}	151	—	—	—	μA/MHz	CPU1 = Deep Sleep ESWM power domain off CPUCLK1 = 250 MHz, MRICLK = 250 MHz, MRPCLK = 15.6 MHz, ICLK = 250 MHz, PCLKA = 15.6 MHz, PCLKB = 15.6 MHz, PCLKC = 15.6 MHz, PCLKD = 15.6 MHz, PCLKE = 15.6 MHz, BCLK = 15.6 MHz
			Cache off, executing from ITCM		143	—	—	—		
			Cache off, executing from SRAM		92	—	—	—		
			Cache off, executing from MRAM		104	—	—	—		
		Normal mode	All peripheral disabled, Cache on, while (1) code		118	—	—	—		
			All peripheral disabled, Cache off, while (1) code executing from MRAM		121	—	—	—		
	CPUCLK0 = 800 MHz VSCR_1 VCL = voltage range 1	Coremark	Cache on		157	—	—	—	μA/MHz	CPU1 = Deep Sleep ESWM power domain off CPUCLK1 = 200 MHz, MRICLK = 200 MHz, MRPCLK = 12.5 MHz, ICLK = 200 MHz, PCLKA = 12.5 MHz, PCLKB = 12.5 MHz, PCLKC = 12.5 MHz, PCLKD = 12.5 MHz, PCLKE = 12.5 MHz, BCLK = 12.5 MHz
			Cache off, executing from ITCM		149	—	—	—		
			Cache off, executing from SRAM		98	—	—	—		
			Cache off, executing from MRAM		111	—	—	—		
		Normal mode	All peripheral disabled, Cache on, while (1) code		124	—	—	—		
			All peripheral disabled, Cache off, while (1) code executing from MRAM		127	—	—	—		

Table 62.34 Coremark and normal mode current, CPU0 active, CPU1 Deep Sleep. (DCDC mode and External VDD mode) (2 of 2)

Parameter				Symbol	Typ	Max			Unit	Test conditions
						95 °C	105 °C	125 °C		
Supply current[*1*2]	CPUCLK0 = 600 MHz CPUCLK1 = 200 MHz VSCR_1 VCL = voltage range 1	Coremark	Cache on	I_{DD}	172	—	—	—	μA/MHz	CPU1 = Deep Sleep ESWM power domain off CPUCLK1 = 200 MHz, MRICLK = 200 MHz, MRPCLK = 25 MHz, ICLK = 200 MHz, PCLKA = 25 MHz, PCLKB = 25 MHz, PCLKC = 25 MHz, PCLKD = 25 MHz, PCLKE = 25 MHz, BCLK = 25 MHz
			Cache off, executing from ITCM		164	—	—	—		
			Cache off, executing from SRAM		118	—	—	—		
			Cache off, executing from MRAM		133	—	—	—		
		Normal mode	All peripheral disabled, Cache on, while (1) code		139	—	—	—		
			All peripheral disabled, Cache off, while (1) code executing from MRAM		142	—	—	—		
	CPUCLK0 = 600 MHz VSCR_2 VCL = voltage range 2	Coremark	Cache on		164	—	—	—	μA/MHz	CPU1 = Deep Sleep ESWM power domain off CPUCLK1 = 150 MHz, MRICLK = 150 MHz, MRPCLK = 9.4 MHz, ICLK = 150 MHz, PCLKA = 9.4 MHz, PCLKB = 9.4 MHz, PCLKC = 9.4 MHz, PCLKD = 9.4 MHz, PCLKE = 9.4 MHz, BCLK = 9.4 MHz
			Cache off, executing from ITCM		156	—	—	—		
			Cache off, executing from SRAM		106	—	—	—		
			Cache off, executing from MRAM		119	—	—	—		
		Normal mode	All peripheral disabled, Cache on, while (1) code		131	—	—	—		
			All peripheral disabled, Cache off, while (1) code executing from MRAM		135	—	—	—		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.

Table 62.35 Coremark and normal mode current, CPU0 Deep Sleep, CPU1 active. (DCDC mode and External VDD mode) (1 of 2)

Parameter				Symbol	Typ	Max 95 °C	Max 105 °C	Max 125 °C	Unit	Test conditions
Supply current[*1] [*2]	CPUCLK1 = 250 MHz VSCR_1 VCL = voltage range 1	Coremark	Cache on	I_{DD}	215	—	—	—	μA/MHz	CPU0 = Deep Sleep ESWM power domain off CPUCLK0 = 1 GHz, MRICLK = 250 MHz, MRPCLK = 15.6 MHz, ICLK = 250 MHz, PCLKA = 15.6 MHz, PCLKB = 15.6 MHz, PCLKC = 15.6 MHz, PCLKD = 15.6 MHz, PCLKE = 15.6 MHz, BCLK = 15.6 MHz
			Cache off, executing from ITCM		197	—	—	—		
			Cache off, executing from SRAM		194	—	—	—		
			Cache off, executing from MRAM		268	—	—	—		
		Normal mode	All peripheral disabled, Cache on, while (1) code		201	—	—	—		
			All peripheral disabled, Cache off, while (1) code executing from MRAM		264	—	—	—		
	CPUCLK1 = 200 MHz VSCR_1 VCL = voltage range 1	Coremark	Cache on		236	—	—	—	μA/MHz	CPU0 = Deep Sleep ESWM power domain off CPUCLK0 = 800 MHz, MRICLK = 200 MHz, MRPCLK = 12.5 MHz, ICLK = 200 MHz, PCLKA = 12.5 MHz, PCLKB = 12.5 MHz, PCLKC = 12.5 MHz, PCLKD = 12.5 MHz, PCLKE = 12.5 MHz, BCLK = 12.5 MHz
			Cache off, executing from ITCM		218	—	—	—		
			Cache off, executing from SRAM		215	—	—	—		
			Cache off, executing from MRAM		299	—	—	—		
		Normal mode	All peripheral disabled, Cache on, while (1) code		222	—	—	—		
			All peripheral disabled, Cache off, while (1) code executing from MRAM		297	—	—	—		

Table 62.35 Coremark and normal mode current, CPU0 Deep Sleep, CPU1 active. (DCDC mode and External VDD mode) (2 of 2)

Parameter				Symbol	Typ	Max			Unit	Test conditions
						95 °C	105 °C	125 °C		
Supply current[*1][*2]	CPUCLK1 = 200 MHz VSCR_1 VCL = voltage range 1	Coremark	Cache on	I_{DD}	226	—	—	—	μA/MHz	CPU0 = Deep Sleep ESWM power domain off CPUCLK0 = 600 MHz, MRICLK = 200 MHz, MRPCLK = 25 MHz, ICLK = 200 MHz, PCLKA = 25 MHz, PCLKB = 25 MHz, PCLKC = 25 MHz, PCLKD = 25 MHz, PCLKE = 25 MHz, BCLK = 25MHz
			Cache off, executing from ITCM		208	—	—	—		
			Cache off, executing from SRAM		205	—	—	—		
			Cache off, executing from MRAM		289	—	—	—		
		Normal mode	All peripheral disabled, Cache on, while (1) code		213	—	—	—		
			All peripheral disabled, Cache off, while (1) code executing from MRAM		287	—	—	—		
	CPUCLK1 = 150 MHz VSCR_2 VCL = voltage range 2	Coremark	Cache on		265	—	—	—	μA/MHz	CPU0 = Deep Sleep ESWM power domain off CPUCLK0 = 600 MHz, MRICLK = 150 MHz, MRPCLK = 9.4 MHz, ICLK = 150 MHz, PCLKA = 9.4 MHz, PCLKB = 9.4 MHz, PCLKC = 9.4 MHz, PCLKD = 9.4 MHz, PCLKE = 9.4 MHz, BCLK = 9.4 MHz
			Cache off, executing from ITCM		247	—	—	—		
			Cache off, executing from SRAM		244	—	—	—		
			Cache off, executing from MRAM		326	—	—	—		
		Normal mode	All peripheral disabled, Cache on, while (1) code		252	—	—	—		
			All peripheral disabled, Cache off, while (1) code executing from MRAM		324	—	—	—		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.

Table 62.36 Increase when the PVD1, PVD2, PVD4, PVD5 or Battery power supply switch is enabled in Deep Software Standby mode 1 and 2.

Parameter		Symbol	Typ	Max			Unit	Test conditions
				95 °C	105 °C	125 °C		
Supply current	Common circuit when enabling PVDn (n=1, 2, 4, 5) or disabling low power consumption function of PVD0 for the battery power supply switch control (OFS1(_SEC).PVDLPSEL=1) in Deep Software Standby mode 1	I_{CC}	4.00	—	—	—	μA	—
	Common circuit when enabling PVDn (n=1, 2, 4, 5) or disabling low power consumption function of PVD0 for the battery power supply switch control (OFS1(_SEC).PVDLPSEL=1) in Deep Software Standby mode 2		4.00	—	—	—		—
	PVD1 enabled		2.00	—	—	—		—
	PVD2 enabled		2.00	—	—	—		—
	PVD4 enabled		2.00		—	—		
	PVD5 enabled		2.00	—	—	—		—
	Battery power supply switch enabled with following conditions.*1 ● Battery power supply switch enable (VBTBPCR1.BPWSWSTP=0), and low power consumption function of PVD0 select at Deep Software Standby mode disable (OFS1(_SEC).PVDLPSEL=1).		2.00	—	—	—		—

Note 1. Consumption current is not increased in other condition.

Table 62.37 Increase when the sub-clock oscillator and RTC are enabled in Deep Software Standby mode 1, 2 and 3.

Parameter			Symbol	Typ	Max			Unit	Test conditions
					95 °C	105 °C	125 °C		
Supply current	When a crystal oscillator is in use	Low Power mode 3	I_{CC}	0.31	—	—	—	μA	—
		Low Power mode 2		0.43	—	—	—		—
		Low Power mode 1		0.52	—	—	—		—
		Standard mode		0.78	—	—	—		—
	RTC is operating			0.30	—	—	—		—

Table 62.38 Inrush current

Parameter				Symbol	Typ	Max			Unit	Test conditions
						95 °C	105 °C	125 °C		
Supply current	Inrush current on Cold Start		Inrush current of VCC_DCDC*1	I_{RUSH}	—	1330	1330	1330	mA	—
	Inrush current on returning from deep software standby mode	DPSBYCR.DCSSMODE=1			—	1270	1270	1270		—
		DPSBYCR.DCSSMODE=2			—	1170	1170	1170		—
		DPSBYCR.DCSSMODE=3			—	1160	1160	1160		—

Note 1. Reference value

Table 62.39 Operating current (Analog) (1 of 2)

Parameter				Symbol	Typ	Max			Unit	Test conditions
						95 °C	105 °C	125 °C		
Supply current*1	Oscillator	Main clock oscillator		I_{CC}	0.65	—	—	—	mA	MOMCR.MODRV0 [2:0] = 000b
					0.76	—	—	—	mA	MOMCR.MODRV0 [2:0] = 011b
					0.88	—	—	—	mA	MOMCR.MODRV0 [2:0] = 101b
	Analog power supply current	During 16-bit A/D conversion	SAR mode, Oversampling mode and Hybrid mode	AI_{CC}	2.4	3.2	3.2	3.2	mA	—
		During 16-bit A/D conversion with S/H amp	SAR mode and Hybrid mode		3.9	5.1	5.1	5.1	mA	—
		ACMPHS(1unit)			99	192	192	192	μA	—
		Temperature sensor			0.1	0.2	0.2	0.2	mA	—
		During D/A conversion (per unit)			1.2	1.6	1.6	1.6	mA	—
		Waiting for A/D, D/A conversion (all units)			3.4	4.1	4.1	4.1	mA	—
		ADC16H, DAC12 in standby modes (all units)*2			1	16.0	22.4	42	μA	—
	Reference power supply current (VREFH0)	During 16-bit A/D conversion (unit 0)	SAR mode	AI_{REFH0}	70	120	120	120	μA	—
			Oversampling mode and Hybrid mode		200	310	310	310	μA	
		Waiting for 16-bit A/D conversion (unit 0)			14.00	14.00	14.00	14.00	μA	—
		ADC16H in standby modes (unit 0)			0.01	0.1	0.1	0.2	μA	—
	Reference power supply current (VREFH)	During 16-bit A/D conversion (unit 1)	SAR mode	AI_{REFH}	70	120	120	120	μA	—
			Oversampling mode and Hybrid mode		200	310	310	310	μA	—
		During D/A conversion (per unit)			29	41.0	41.0	41.0	μA	—
		Waiting for 16-bit A/D (unit 1), D/A (all units) conversion			14	14	14	14	μA	—
		ADC16H in standby modes (unit 1)			0.1	0.1	0.2	0.3	μA	—

Table 62.39 Operating current (Analog) (2 of 2)

Parameter				Symbol	Typ	Max 95 °C	Max 105 °C	Max 125 °C	Unit	Test conditions
Supply current[*1]	USB operating current	Low speed	USBFS	$I_{CCUSBLS}$	2.9	4.0	4.0	4.0	mA	VCC_USB
			USBHS		11.51	14.6	14.6	14.6	mA	VCC_USBHS = AVCC_USBHS (PHYSET.HSEB = 0)
			USBHS		5.04	6.8	6.8	6.8	mA	VCC_USBHS = AVCC_USBHS (PHYSET.HSEB = 1)
		Full speed	USBFS	$I_{CCUSBFS}$	4.0	4.7	4.7	4.7	mA	VCC_USB
			USBHS		12.45	14.7	14.7	14.7	mA	VCC_USBHS = AVCC_USBHS (PHYSET.HSEB = 0)
			USBHS		5.98	6.9	6.9	6.9	mA	VCC_USBHS = AVCC_USBHS (PHYSET.HSEB = 1)
		High speed	USBHS	$I_{CCUSBHS}$	45.71	55.3	55.3	55.3	mA	VCC_USBHS = AVCC_USBHS
		Standby mode (direct power down)	USBHS	$I_{CCUSBSBY}$	0.89	11.4	11.4	11.4	μA	VCC_USBHS = AVCC_USBHS

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
Note 2. When the MCU is in Software Standby mode or the MSTPCRD.MSTPD21 (16-Bit A/D Converter Module Stop bit) is in the module-stop state.

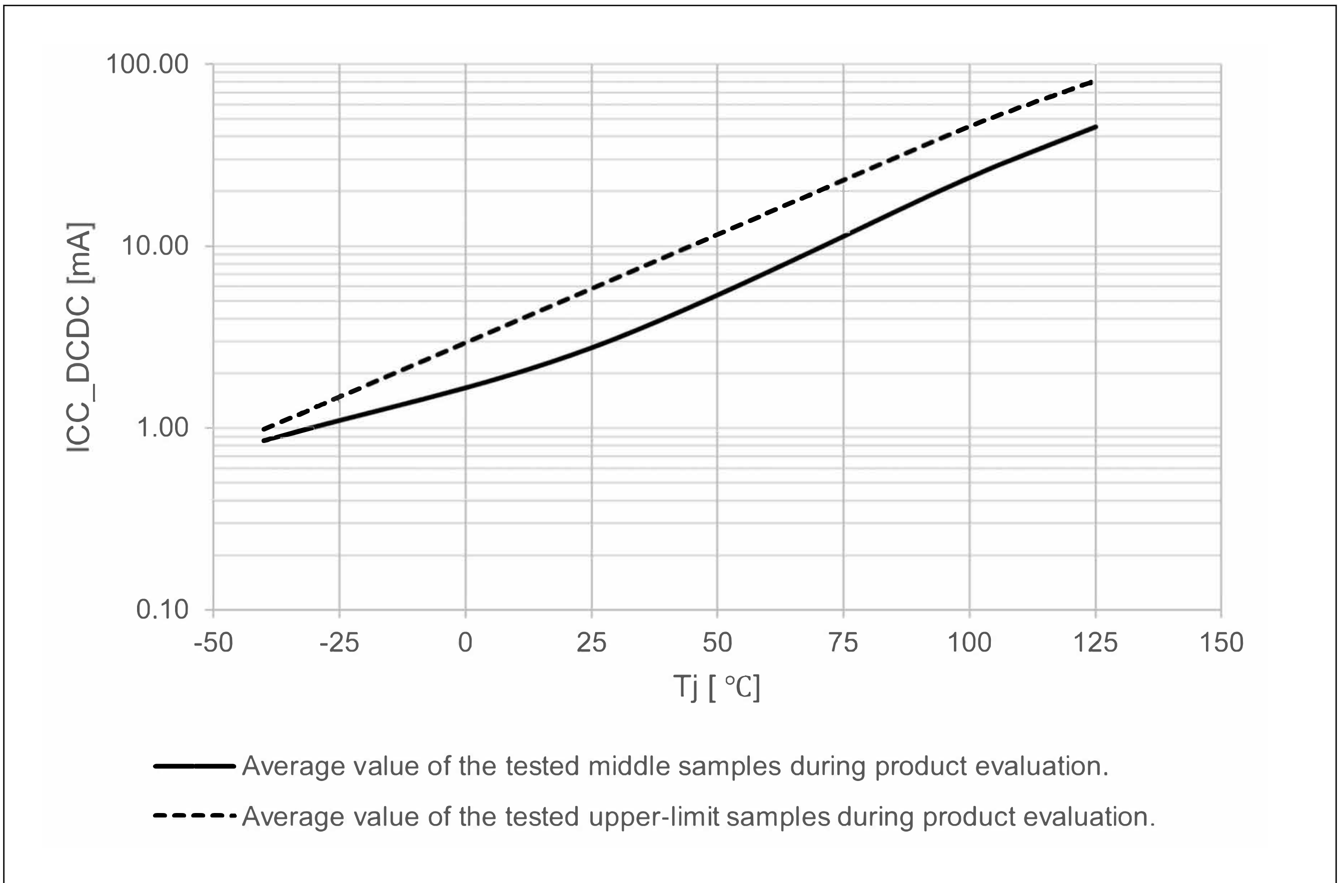


Figure 62.3 Temperature dependency in Software Standby mode (ICC_DCDC, SS2LP_0, SVSCR_1) (reference data)

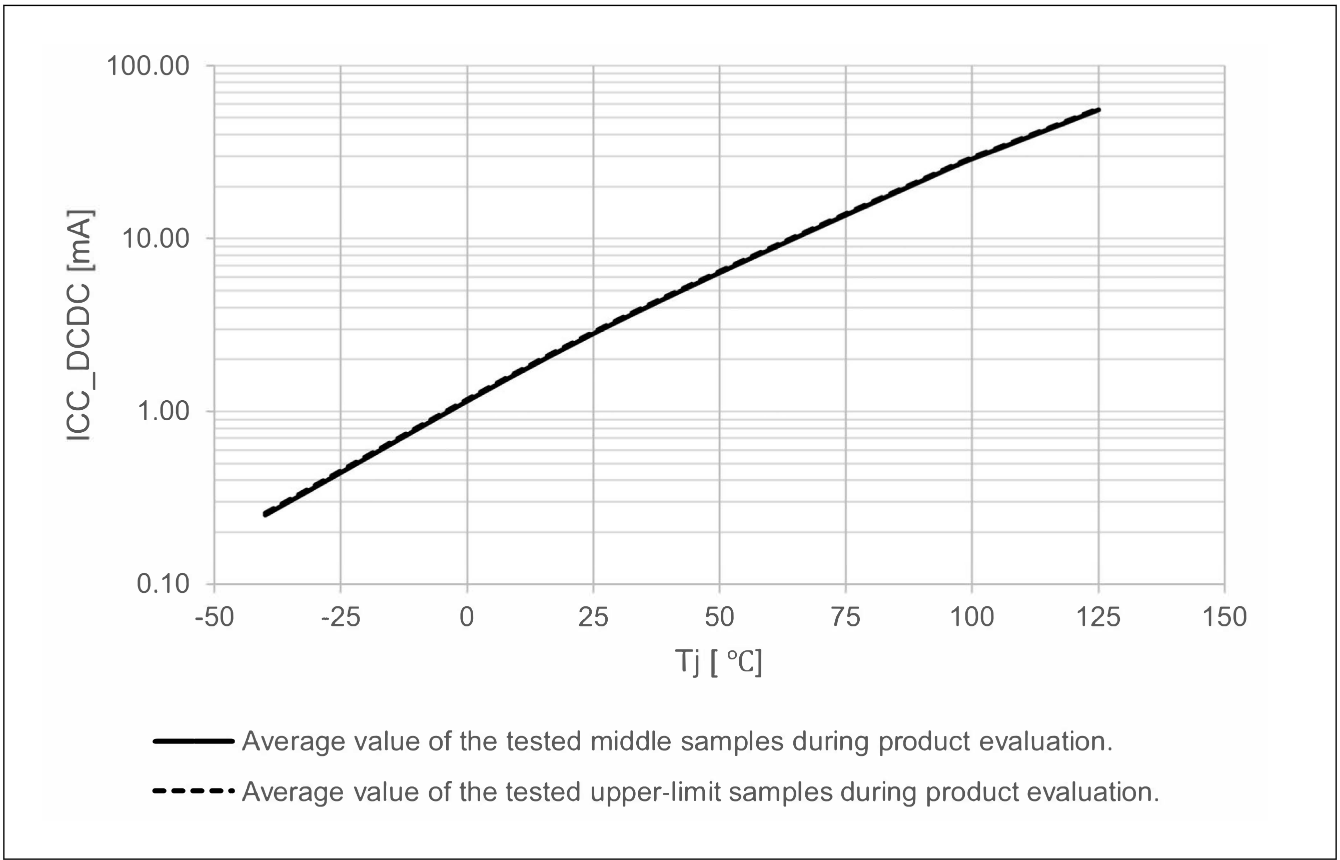


Figure 62.4 Temperature dependency in Software Standby mode (ICC_DCDC, SS2LP_0, SVSCR_5) (reference data)

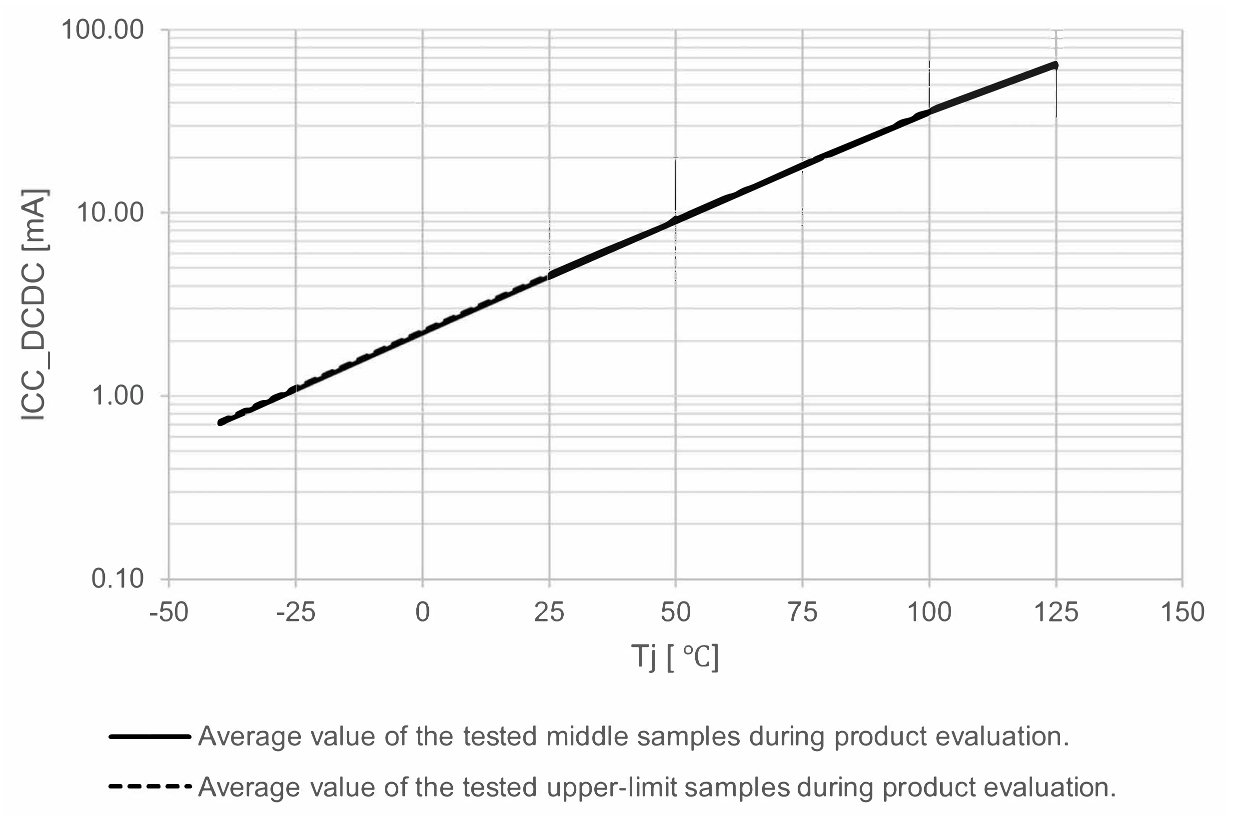


Figure 62.5 Temperature dependency in Software Standby mode (ICC_DCDC, SS2LP_1, SVSCR_2) (reference data)

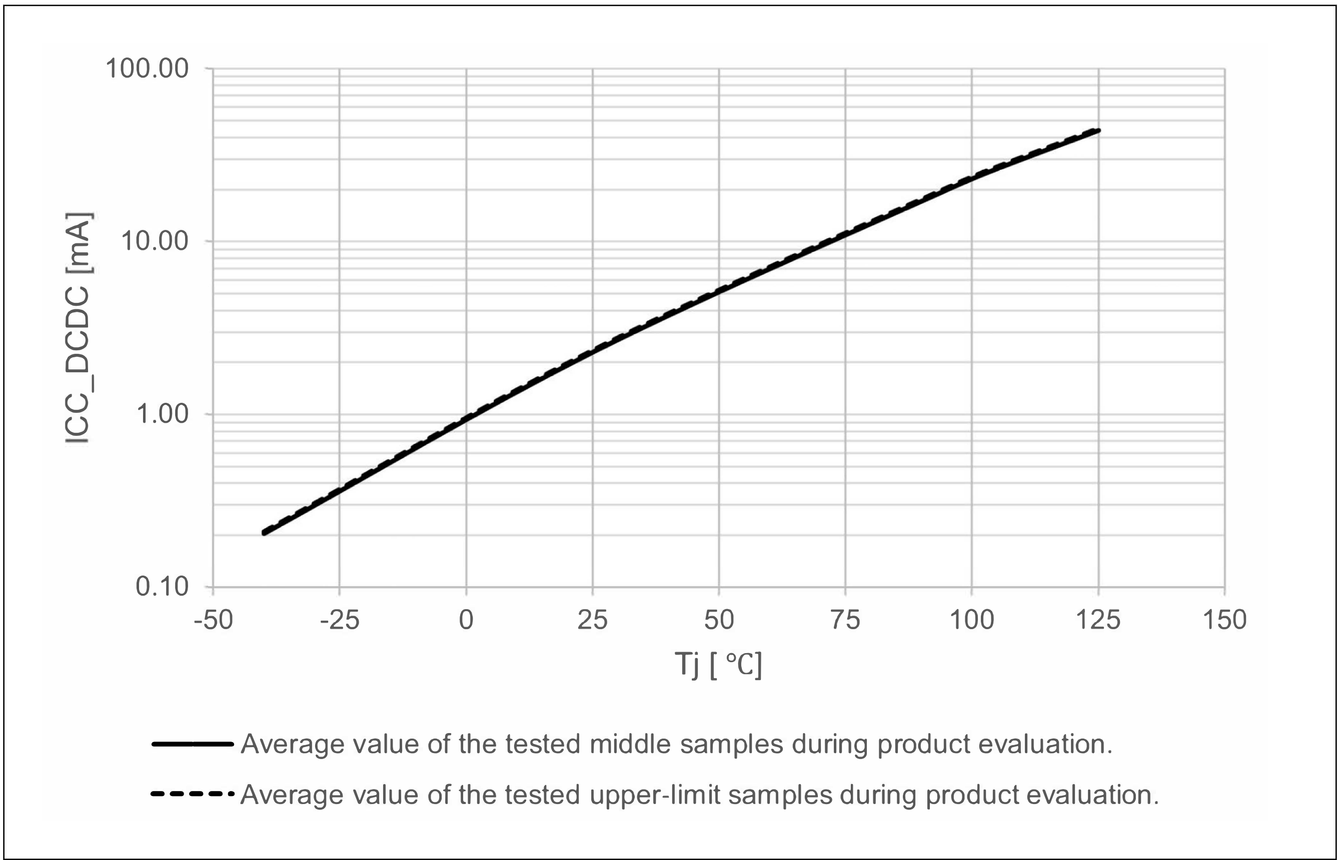


Figure 62.6 Temperature dependency in Software Standby mode (ICC_DCDC, SS2LP_1, SVSCR_5) (reference data)

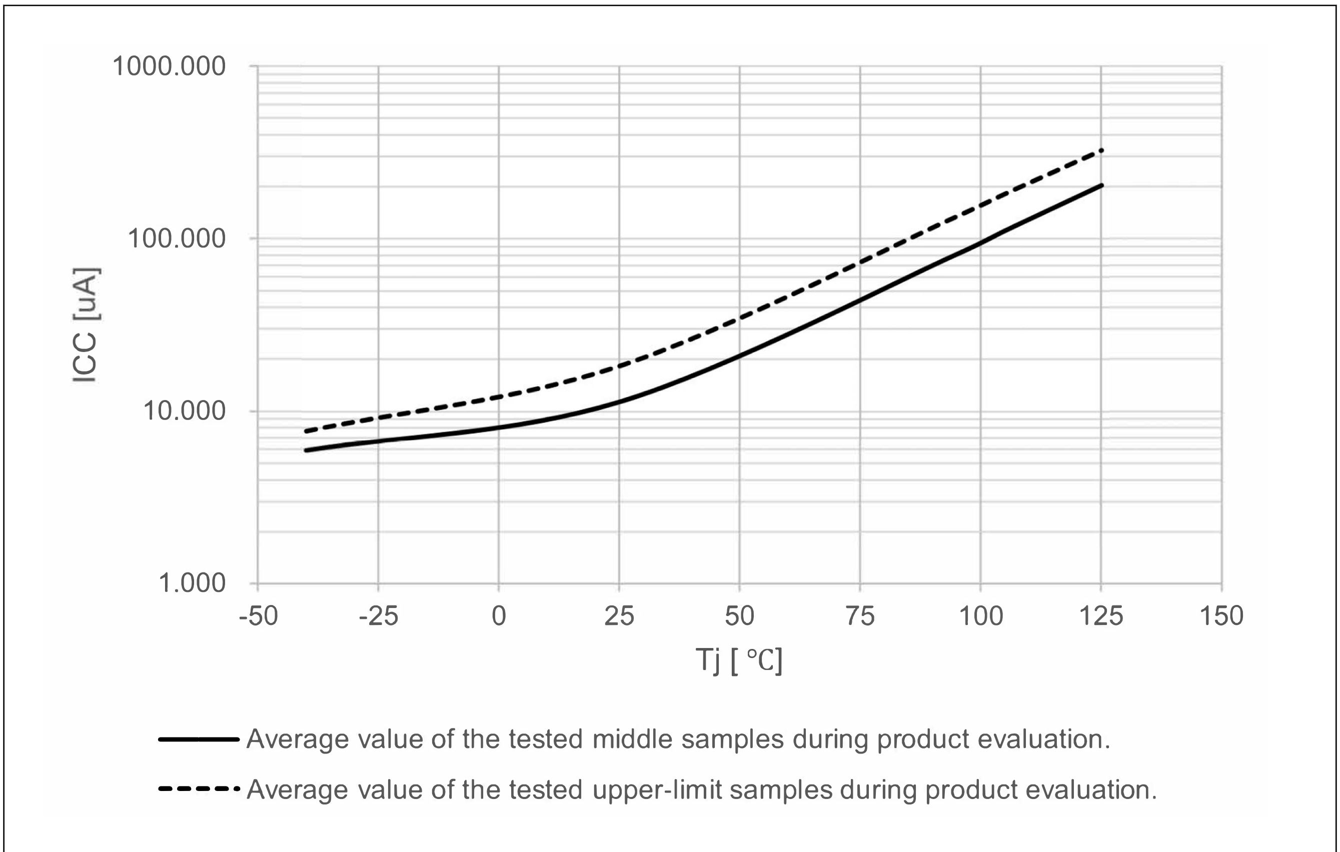


Figure 62.7 Temperature dependency in Deep Software Standby mode 1 (reference data)

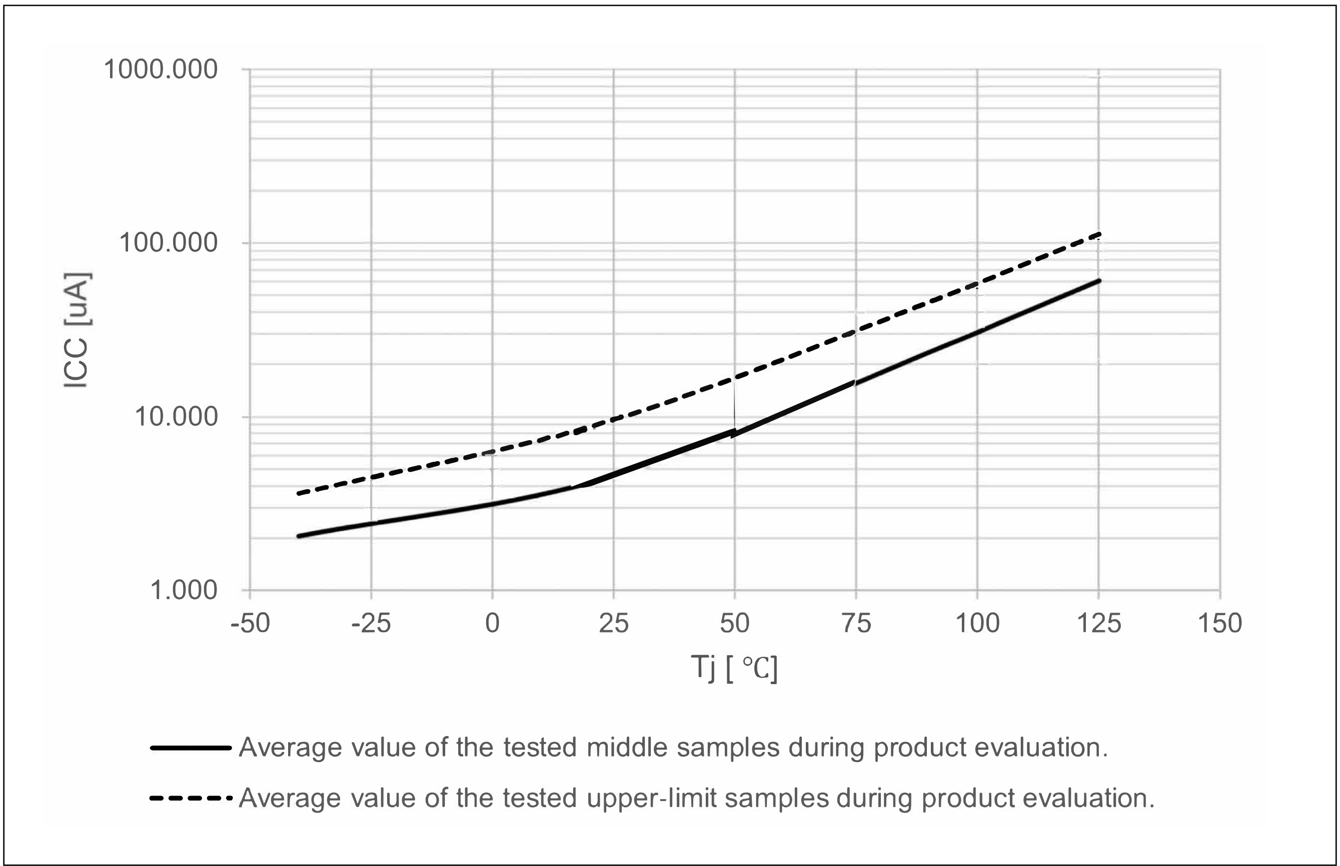


Figure 62.8 Temperature dependency in Deep Software Standby mode 2 (reference data)

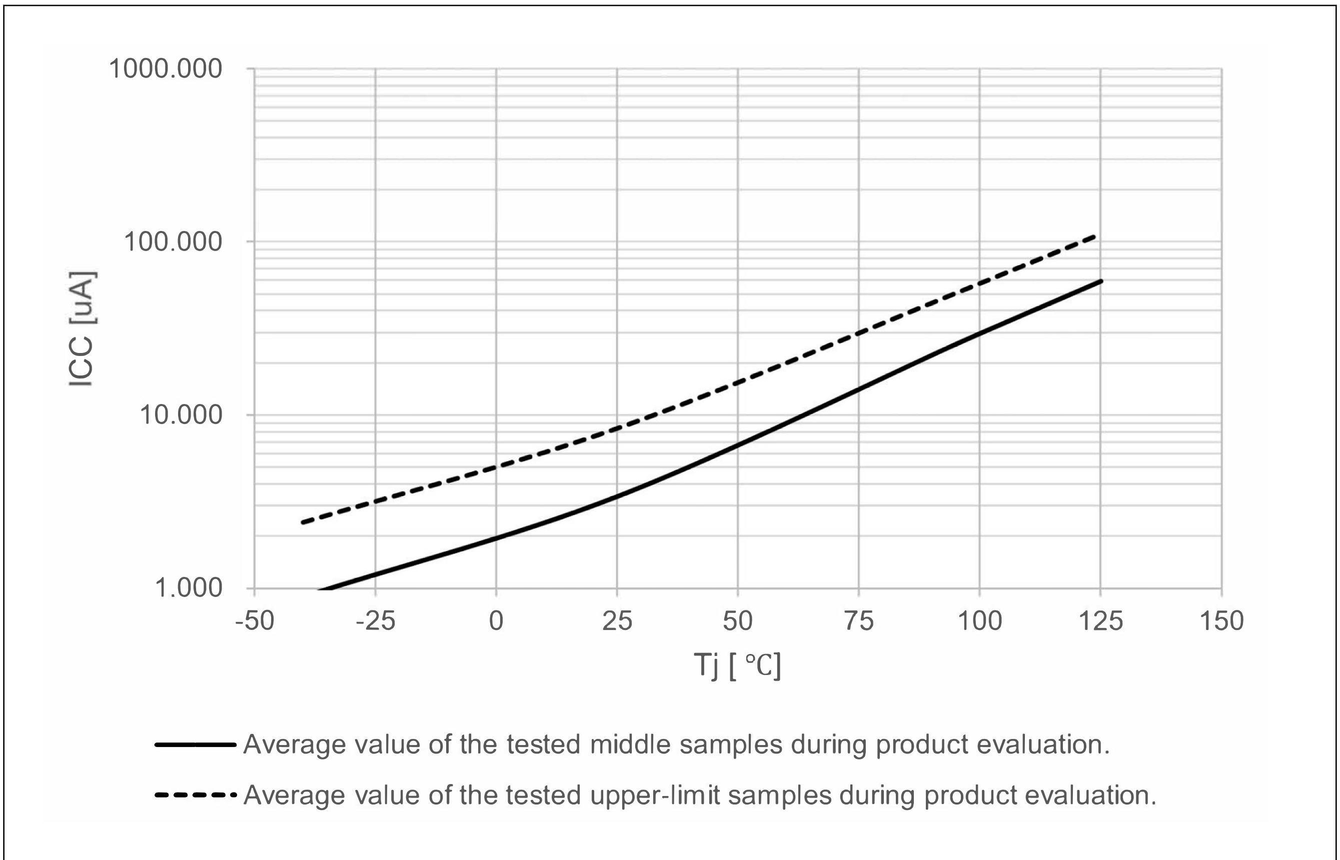


Figure 62.9 Temperature dependency in Deep Software Standby mode 3 (reference data)

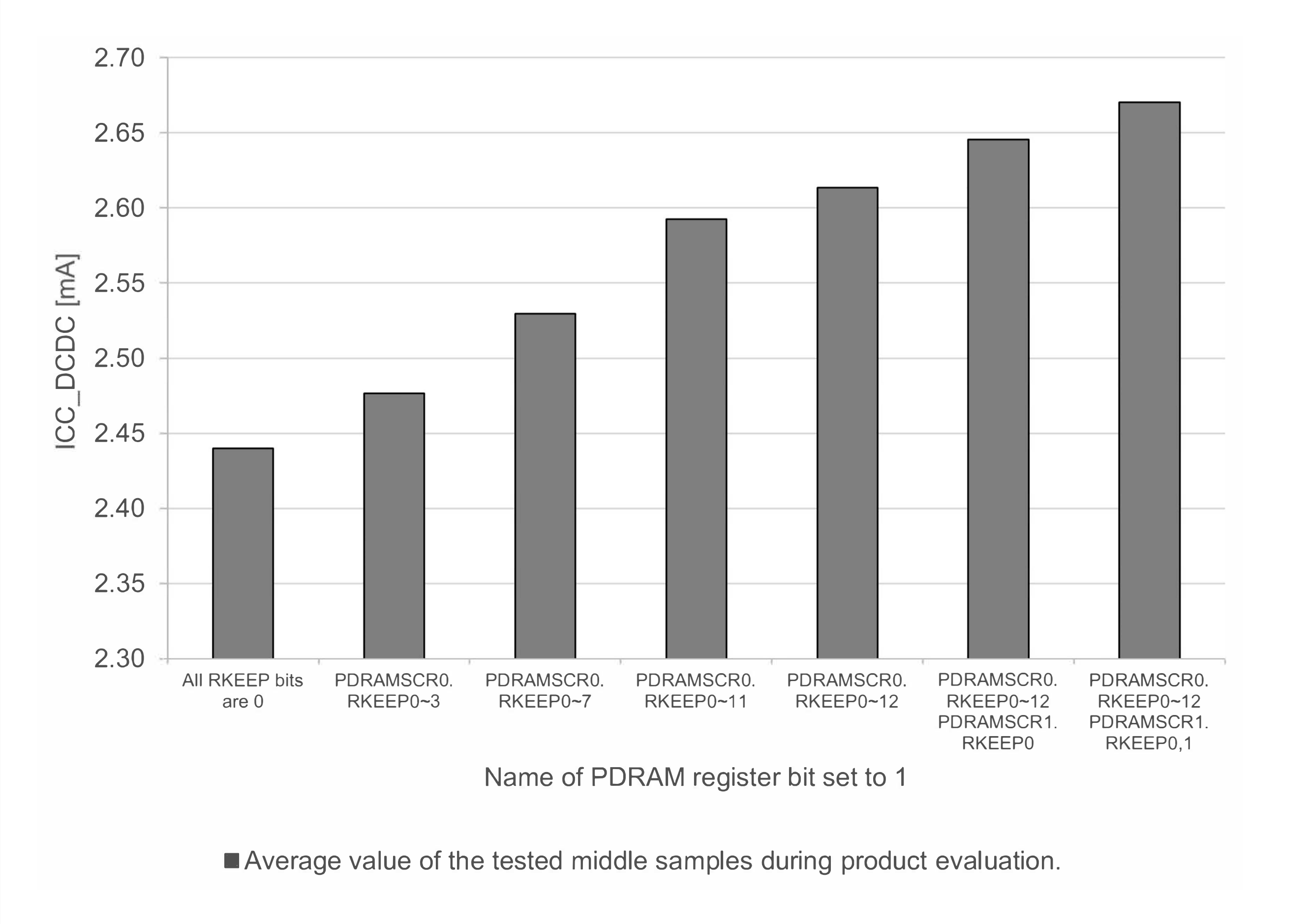


Figure 62.10 Software Standby current per SRAM state (ICC_DCDC, SS2LP_0, SVSCR_1) (reference data)

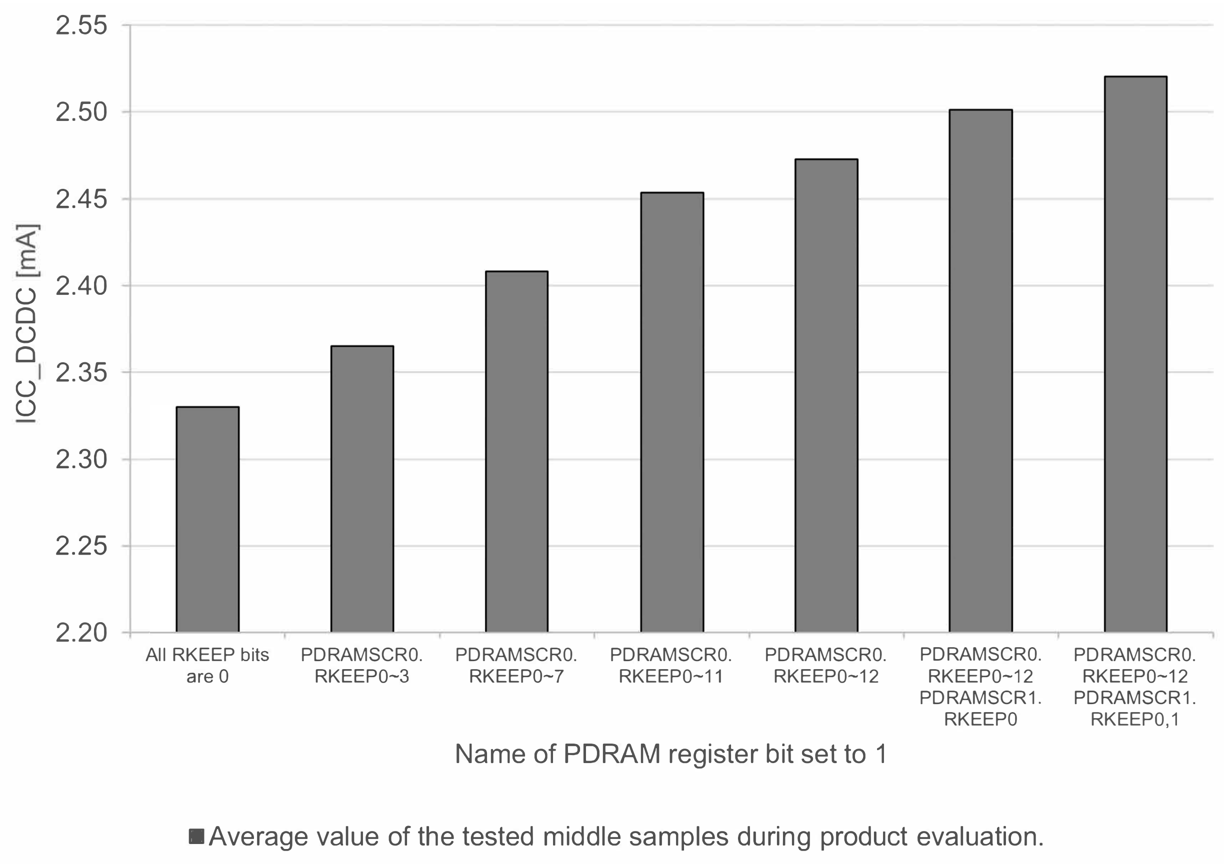


Figure 62.11 Software Standby current per SRAM state (ICC_DCDC, SS2LP_0, SVSCR_2) (reference data)

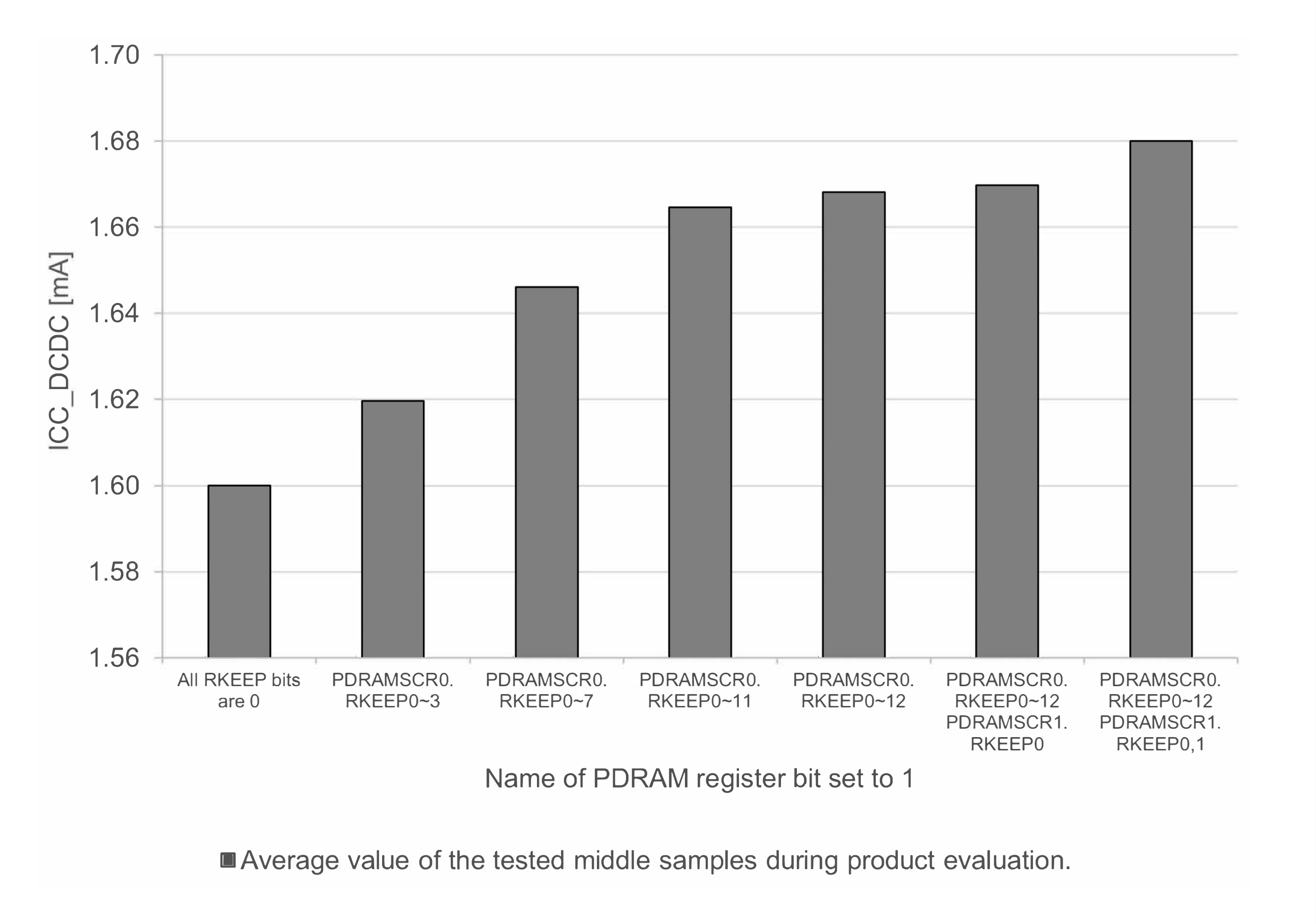


Figure 62.12 Software Standby current per SRAM state (ICC_DCDC, SS2LP_0, SVSCR_3) (reference data)

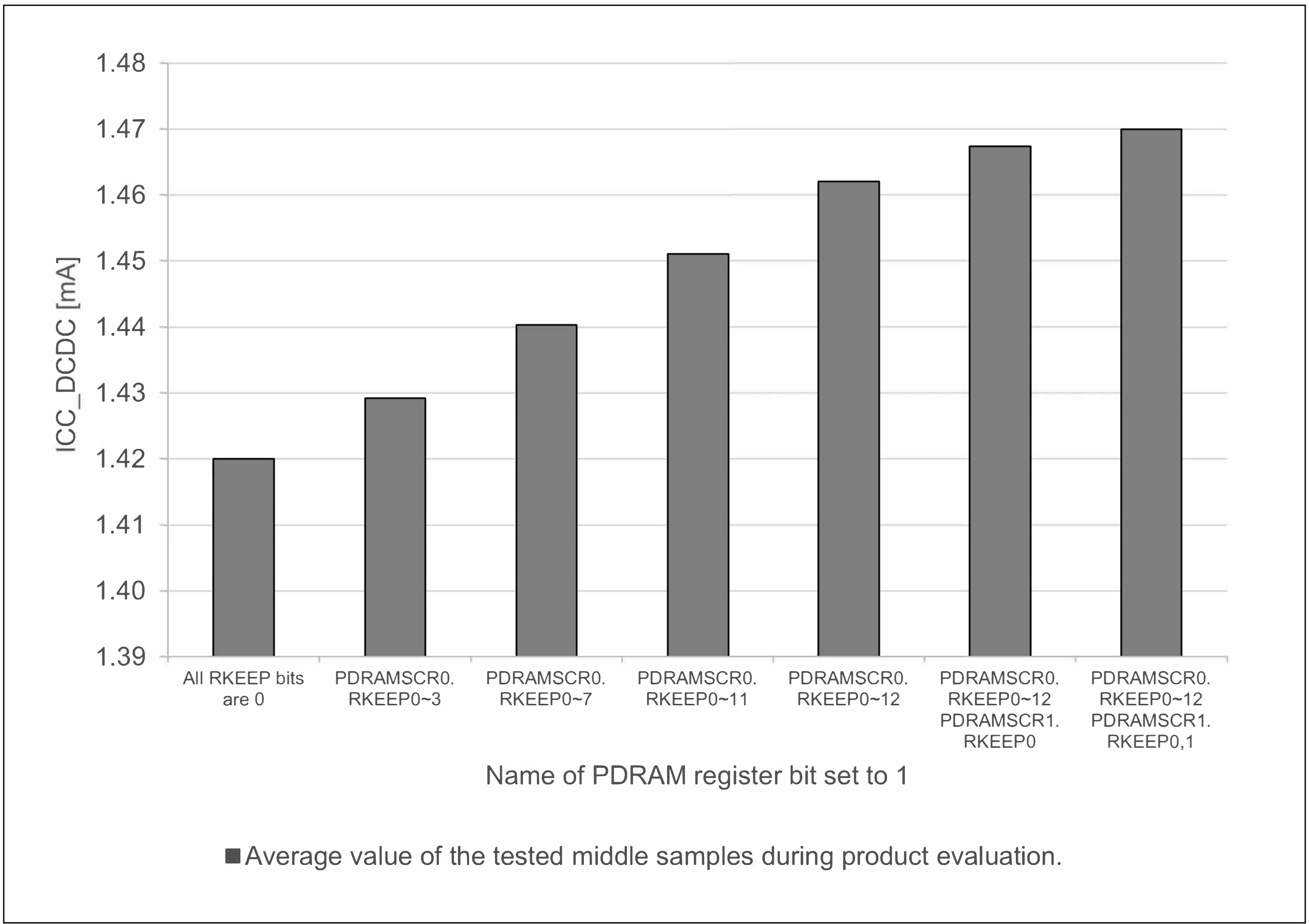


Figure 62.13 Software Standby current per SRAM state (ICC_DCDC, SS2LP_0, SVSCR_4) (reference data)

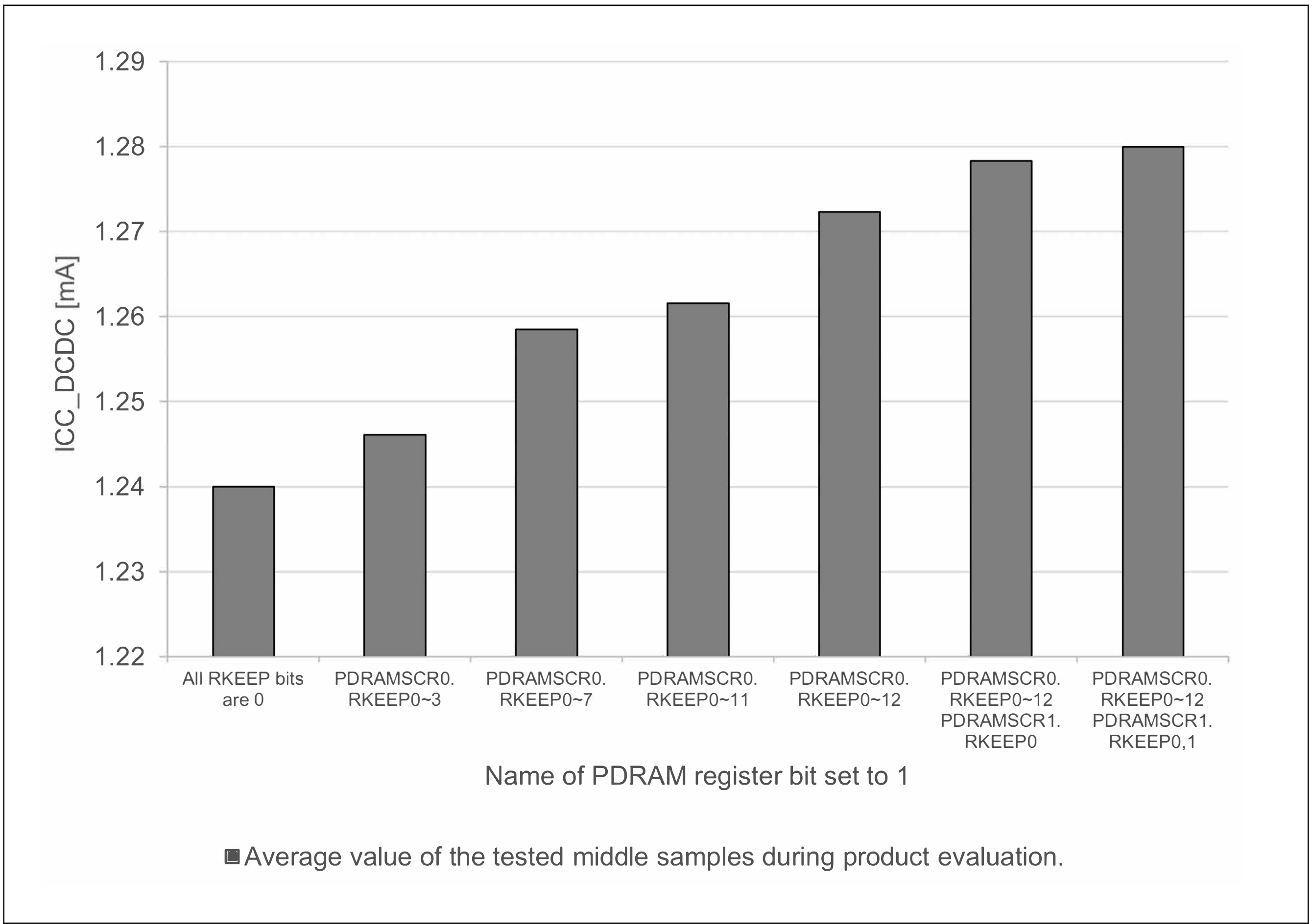


Figure 62.14 Software Standby current per SRAM state (ICC_DCDC, SS2LP_0, SVSCR_5) (reference data)

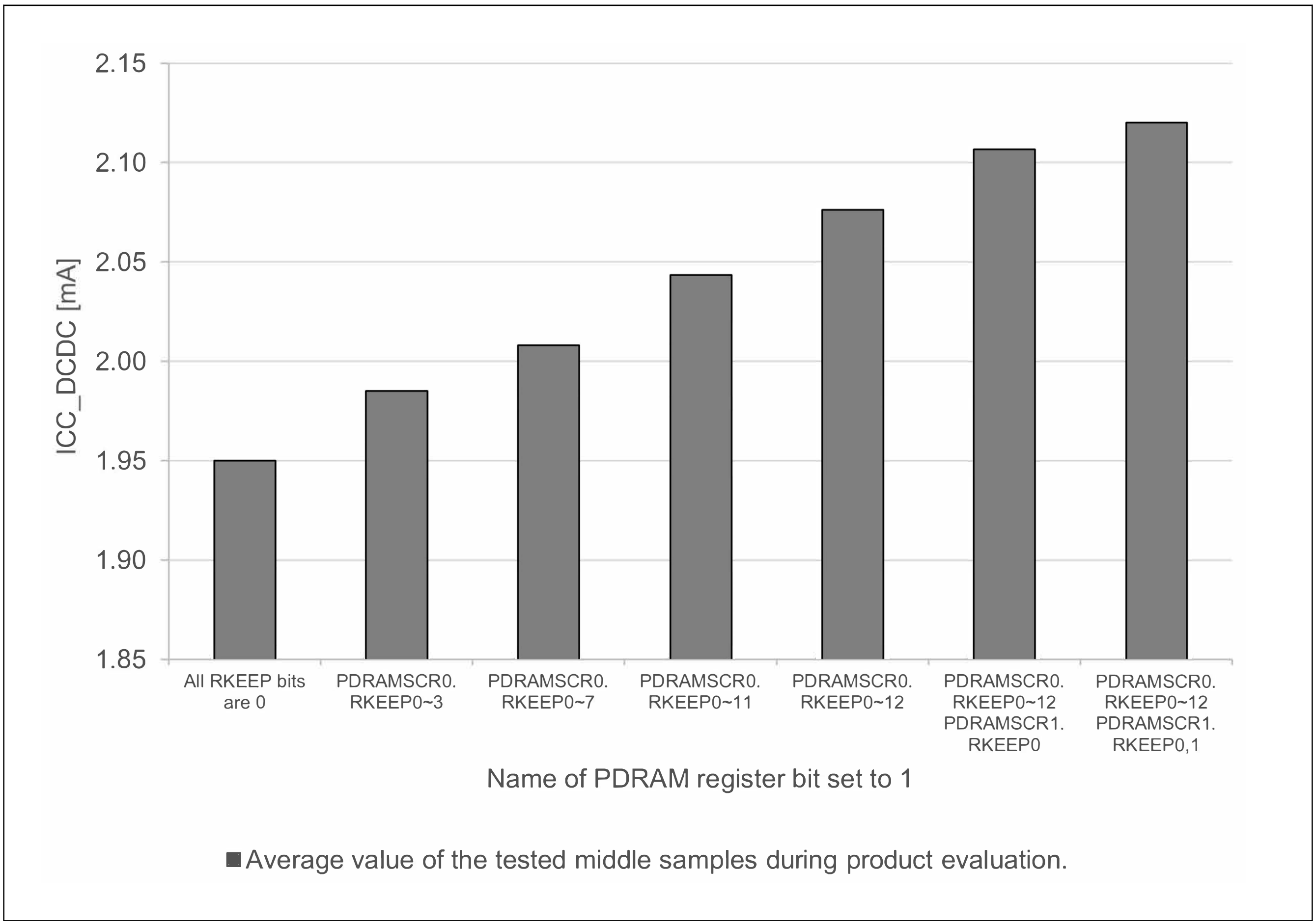


Figure 62.15 Software Standby current per SRAM state (ICC_DCDC, SS2LP_1, SVSCR_2) (reference data)

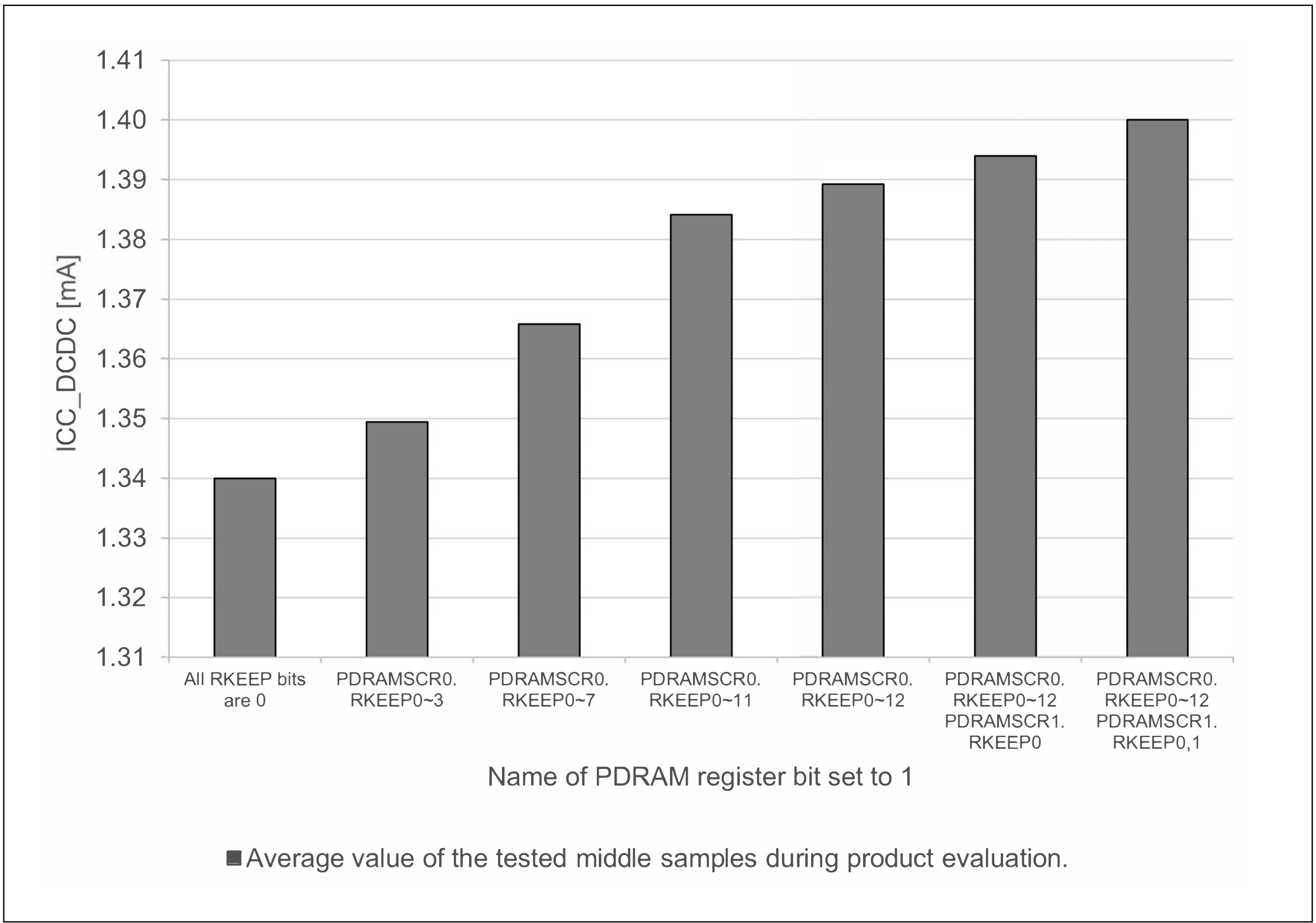


Figure 62.16 Software Standby current per SRAM state (ICC_DCDC, SS2LP_1, SVSCR_3) (reference data)

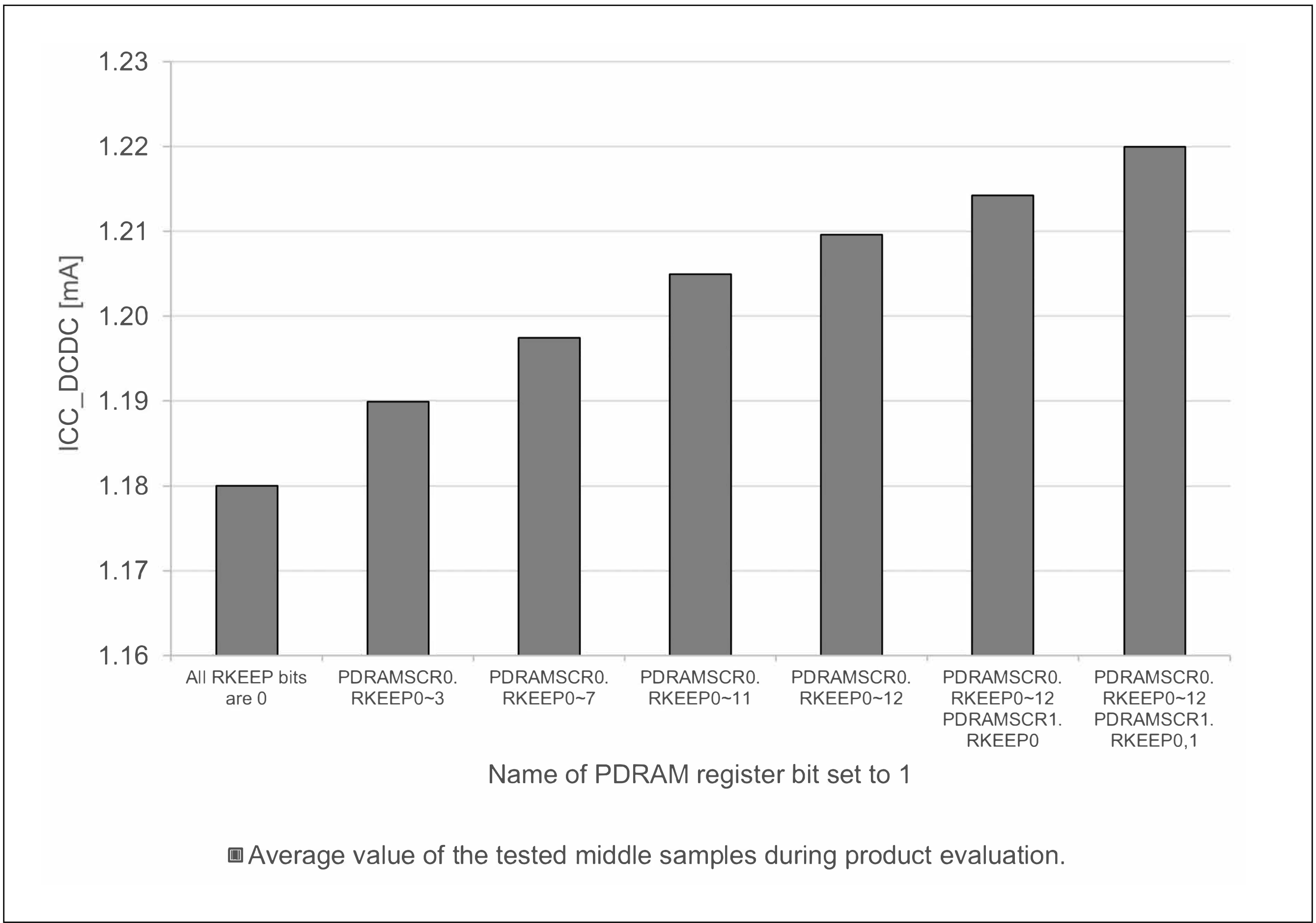


Figure 62.17 Software Standby current per SRAM state (ICC_DCDC, SS2LP_1, SVSCR_4) (reference data)

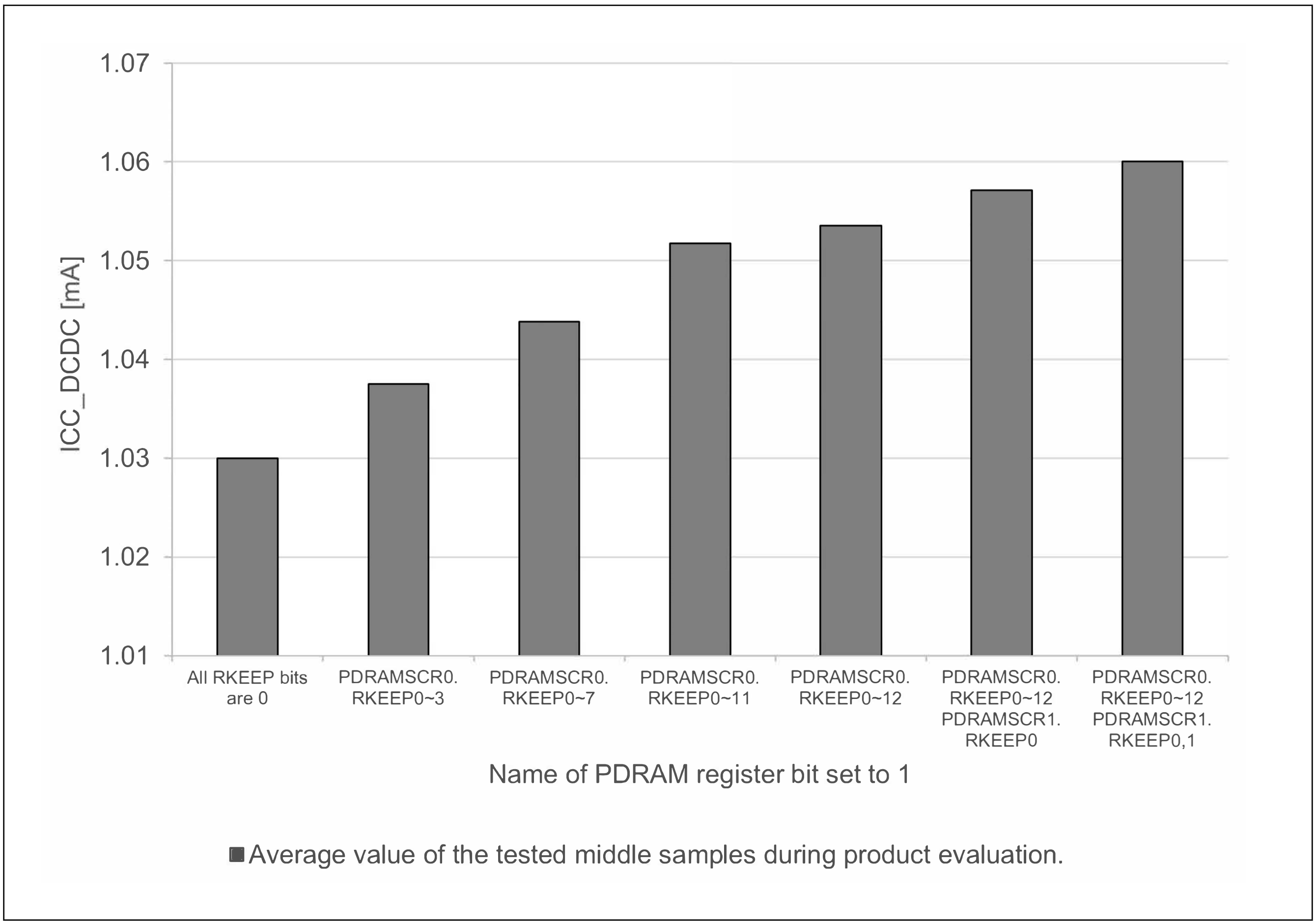


Figure 62.18 Software Standby current per SRAM state (ICC_DCDC, SS2LP_1, SVSCR_5) (reference data)

The more practical I_{CC_DCDC} value can be obtained with the following formula.

I_{CC_DCDC} = (I_{DD} × VCL) / (VCC_DCDC × efficiency)

Where: VCL and VCC are the voltage of VCL pin and VCC pin respectively, and efficiency is shown in the following figures.

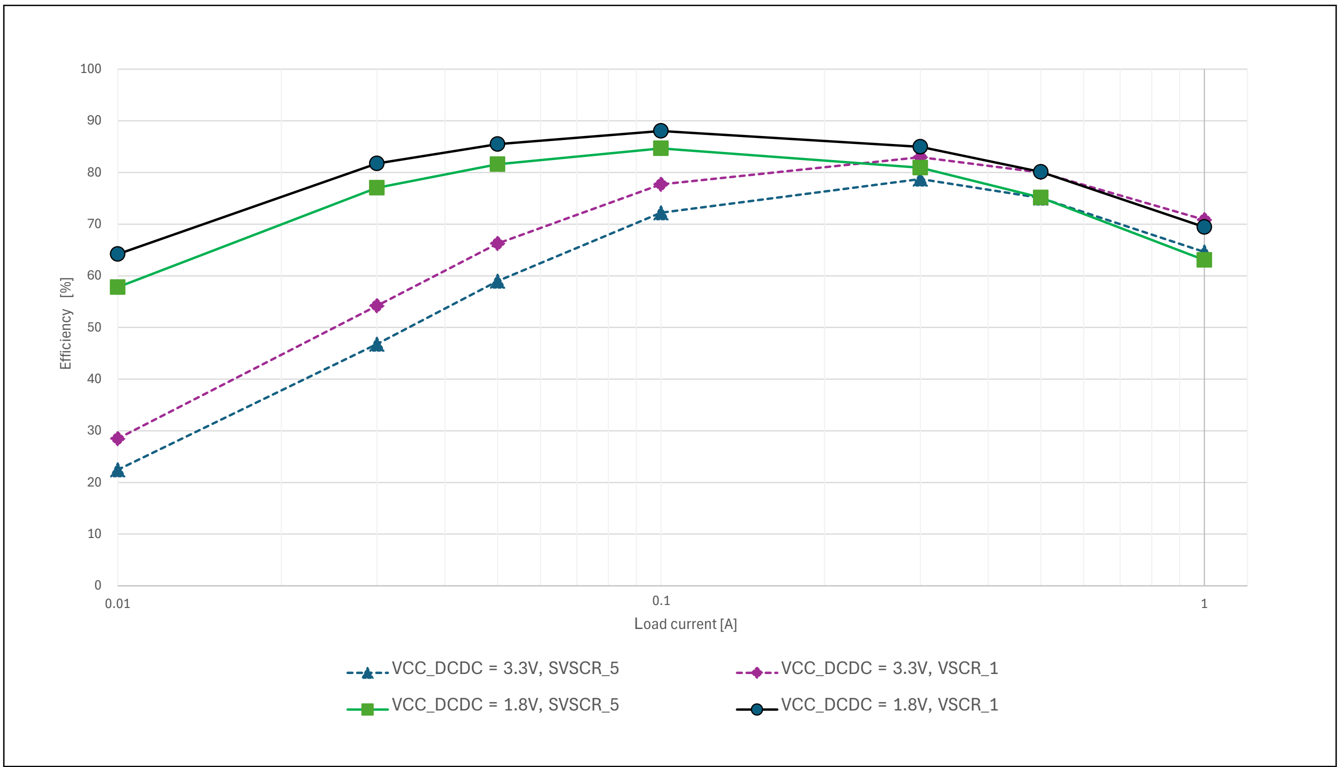


Figure 62.19 Typical DCDC efficiency (%) vs load current (A) in High-speed mode and Software Standby mode (SSCR1.SS2LP = SS2LP_1), Tj = 25 °C

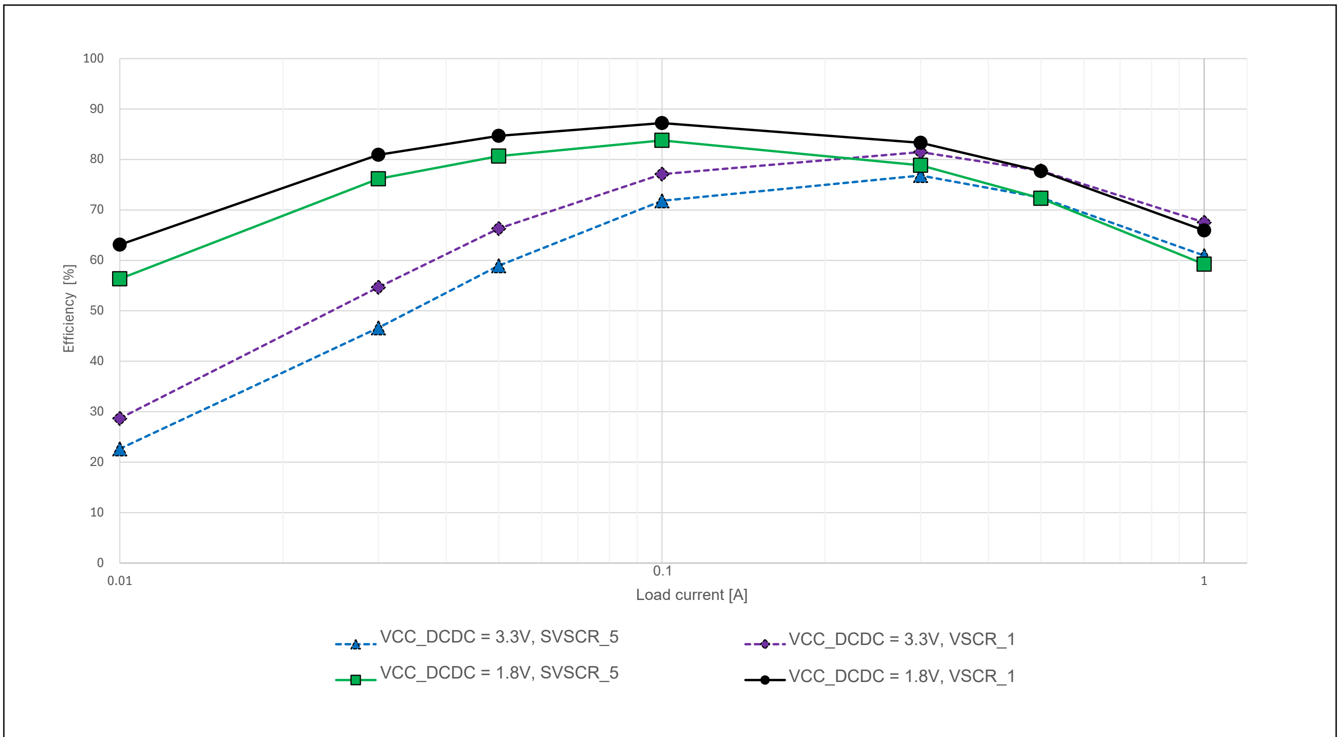


Figure 62.20 Typical DCDC efficiency (%) vs load current (A) in High-speed mode and Software Standby mode (SSCR1.SS2LP = SS2LP_0), Tj = 125 °C

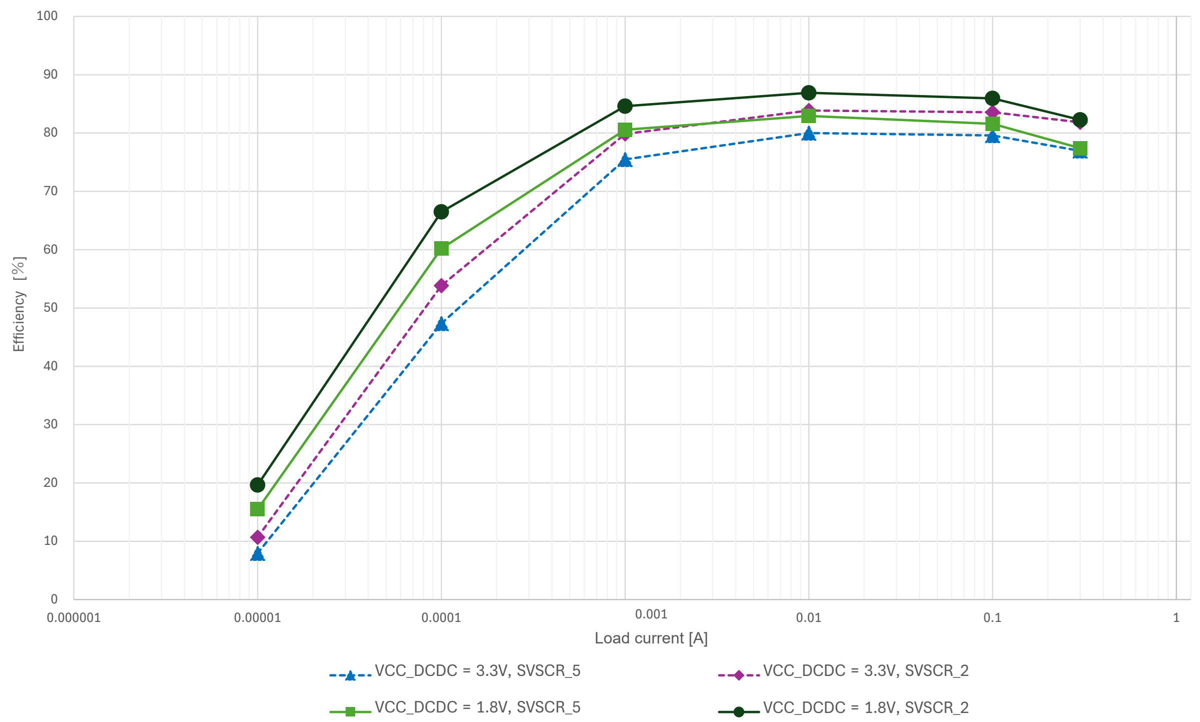


Figure 62.21 Typical DCDC efficiency (%) vs load current (A) in Software Standby mode (SSCR1.SS2LP = SS2LP_1), Tj = 25 °C

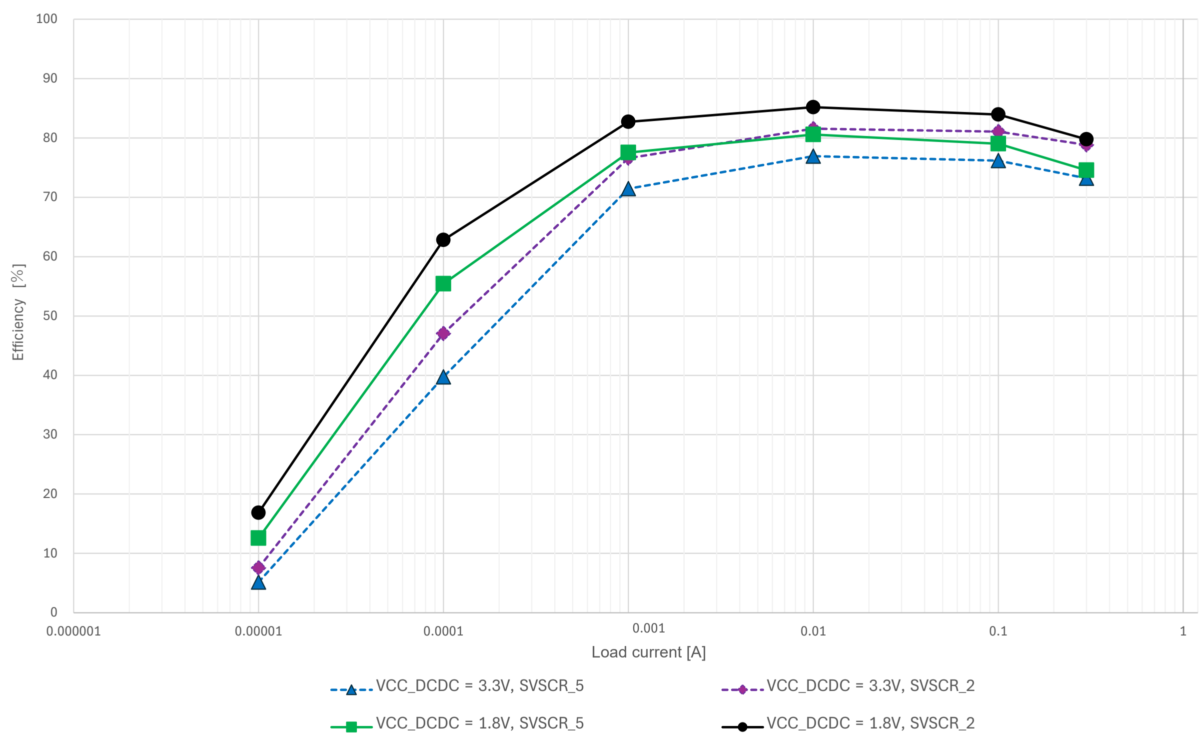


Figure 62.22 Typical DCDC efficiency (%) vs load current (A) in Software Standby mode (SSCR1.SS2LP = SS2LP_1), Tj = 125 °C

62.2.6 VCC Rise and Fall Gradient and Ripple Frequency

Table 62.40 VCC rise and fall gradient characteristics at power on/off

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
VCC rising gradient at power on*1		SrVCC	0.0084	—	20	ms/V	—
VCC falling gradient at power off	VBATT function is disabled*1	SfVCC1	0.0084	—	—	ms/V	—
	VBATT function is enabled.	SfVCC2	1.0000	—	—		—

Note 1. In case the VCC voltage crosses V_{POR1}

Table 62.41 VCC ripple frequency and gradient characteristics during operation

The ripple voltage must meet the allowable ripple frequency $f_{r(VCC)}$ within the range between the VCC upper limit (3.63 V) and lower limit (1.62 V). When the VCC change exceeds VCC ±10%, the allowable voltage change rising and falling gradient dt/dVCC must be met.

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Allowable ripple frequency	$f_{r(VCC)}$	—	—	10	kHz	Figure 62.23 $V_{r(VCC)} \le VCC \times 0.2$
		—	—	1	MHz	Figure 62.23 $V_{r(VCC)} \le VCC \times 0.08$
		—	—	10	MHz	Figure 62.23 $V_{r(VCC)} \le VCC \times 0.06$
Allowable voltage change rising and falling gradient	dt/dVCC*1	1.0	—	—	ms/V	When VCC change exceeds VCC ±10%

Note 1. In case the VCC voltage does not cross $_{VPOR1}$.

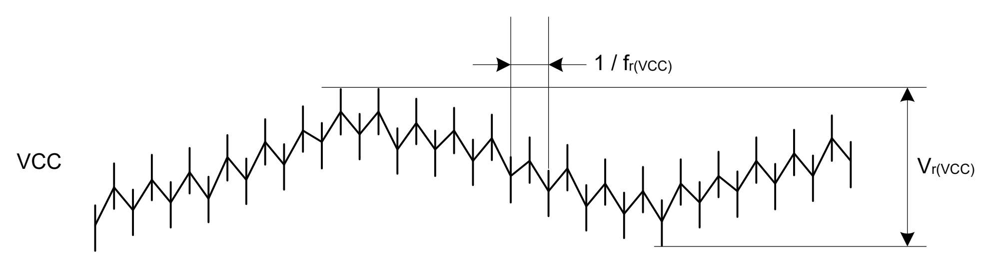


Figure 62.23 Ripple waveform

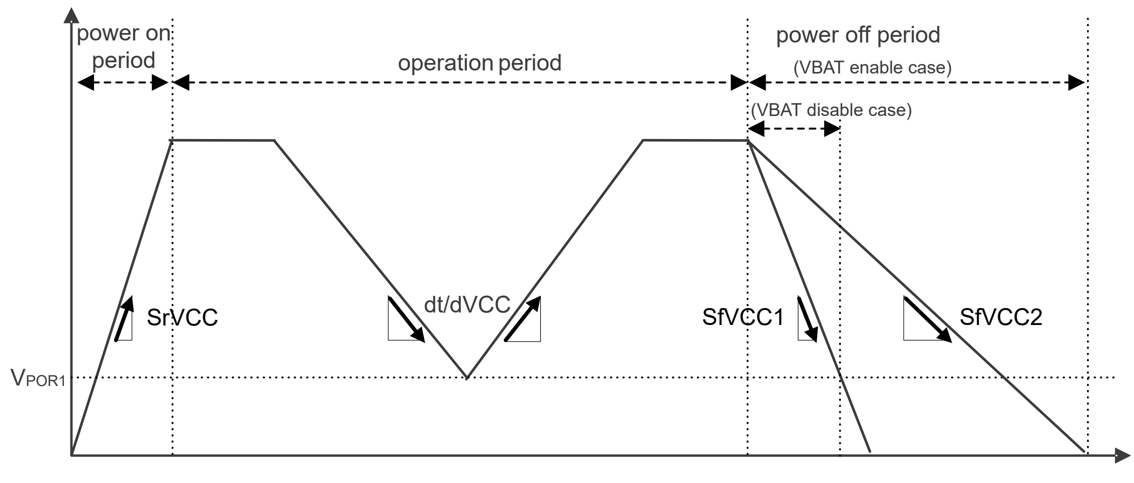


Figure 62.24 VCC rising and falling waveform

62.2.7 Power Supply Rise Gradient

Table 62.42 VCC_USB rise gradient characteristics at power on

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
VCC_USB rising gradient at power on	SrVCC_USB	8.4	—	—	μs/V	—

Table 62.43 VCC_USBHS and AVCC_USBHS rise gradient characteristics at power on

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
VCC_USBHS (=AVCC_USBHS) rising gradient at power on	SrVCC_USBHS	8.4	—	—	μs/V	—

62.2.8 Thermal Characteristics

Maximum value of junction temperature (Tj) must not exceed the value of section 62.2.1. Tj/Ta Definition.

Tj is calculated by either of the following equations.

- Tj = Ta + θja × Total power consumption
- Tj = Tt + Ψjt × Total power consumption
  - Tj : Junction Temperature (°C)
  - Ta : Ambient Temperature (°C)
  - Tt : Top Center Case Temperature (°C)
  - θja : Thermal Resistance of "Junction"-to-"Ambient" (°C/W)
  - Ψjt : Thermal Resistance of "Junction"-to-"Top Center Case" (°C/W)
- Total power consumption = Voltage × (Leakage current + Dynamic current)
- Leakage current of IO = Σ (I_{OL} × V_{OL}) /Voltage + Σ (|I_{OH}| × |VCC – V_{OH}|) /Voltage
- Dynamic current of IO = Σ IO (C_{in} + C_{load}) × IO switching frequency × Voltage
  - C_{in}: Input capacitance
  - C_{load}: Output capacitance

Regarding θja and Ψjt, see Table 62.44.

Table 62.44 Thermal Resistance

Parameter	Package	Symbol	Value*1	Unit	Test conditions
Thermal Resistance	176-pin HLQFP (PLQP0176K?-A)	θja	15	°C/W	JESD 51-2 and 51-7 compliant
	224-pin BGA (PLBG0224J?-A)		21		JESD 51-2 and 51-9 compliant
	289-pin BGA (PLBG0289J?-A)		20		
	303-pin BGA (PLBG0303G?-A)		17		
	176-pin HLQFP (PLQP0176K?-A)	Ψjt	0.3	°C/W	JESD 51-2 and 51-7 compliant
	224-pin BGA (PLBG0224J?-A)		0.3		JESD 51-2 and 51-9 compliant
	289-pin BGA (PLBG0289J?-A)		0.3		
	303-pin BGA (PLBG0303G?-A)		0.5		

Note 1. The values are reference values when the 4-layer board is used. Thermal resistance depends on the number of layers or size of the board. For details, see the JEDEC standards.

62.2.8.1 Calculation Guide of Maximum Current

Table 62.45 Power consumption of each unit (DCDC mode) (1 of 3)

Dynamic current/ Leakage current	MCU Domain	Category	Item	Symbol	Frequency [MHz]	Current [μA/MHz]	Current [mA]	Condition
Leakage current	Analog	Regulator and Leak*1	Tj = 95°C	I_{CC}	—	—	2.51	—
			Tj = 105°C		—	—	2.91	
			Tj = 115°C		—	—	3.45	
			Tj = 125°C		—	—	4.15	
			Tj = 95°C	I_{CC_DCDC}	—	—	186	VCC_DCDC = 3.3 V, VSCR_1, PDCTRESWM.PDDE = 0
			Tj = 105°C		—	—	216	
			Tj = 115°C		—	—	243	
			Tj = 125°C		—	—	277	
			Tj = 95°C		—	—	363	VCC_DCDC = 1.8 V, VSCR_1, PDCTRESWM.PDDE = 0
			Tj = 105°C		—	—	421	
			Tj = 115°C		—	—	474	
			Tj = 125°C		—	—	540	
			Tj = 95°C	I_{DD}	—	—	477	VSCR_1, PDCTRESWM.PDDE = 0
			Tj = 105°C		—	—	553	
			Tj = 115°C		—	—	622	
			Tj = 125°C		—	—	709	

Table 62.45 Power consumption of each unit (DCDC mode) (2 of 3)

Dynamic current/ Leakage current	MCU Domain	Category	Item	Symbol	Frequency [MHz]	Current [μA/MHz]	Current [mA]	Condition
Dynamic current	CPU0	Operation with Cache	CoreMark	I_{DD}	1000	145	144	CPUCLK0 = 1000 MHz VSCR_1
	CPU1	Operation with Cache	CoreMark		250	166	41	CPUCLK1 = 250 MHz VSCR_1
	Peripheral Unit	Timer	RTC		62.5	1.229	0.077	VSCR_1
			GPT32 (14ch)*2		125	65.123	8.140	
			POEG (4 Groups)*2		62.5	1.539	0.096	
			PDG (4ch)*2		125	47.465	5.933	
			AGT (2ch)*2		62.5	1.518	0.095	
			ULPT (2ch)*2		62.5	2.373	0.148	
			WDT0		62.5	0.437	0.027	
			WDT1		62.5	0.446	0.028	
			IWDT		62.5	0.014	0.001	

Table 62.45 Power consumption of each unit (DCDC mode) (3 of 3)

Dynamic current/ Leakage current	MCU Domain	Category	Item	Symbol	Frequency [MHz]	Current [μA/MHz]	Current [mA]	Condition
Dynamic current	Peripheral Unit	Communication interfaces	ESWM	I_{DD}	125	294.026	36.753	VSCR_1
			USBFS		62.5	7.495	0.468	
			USBHS		125	67.424	8.428	
			SCI (10ch)*2		125	32.336	4.042	
			IIC (3ch)*2		62.5	3.722	0.233	
			I3C		125	9.883	1.235	
			CANFD (2ch)*2		125	6.025	0.753	
			SPI (2ch)*2		125	11.36	1.420	
			OSPI (2ch)*2		62.5	100.8	6.300	
			SSIE (2ch)*2		62.5	7.89	0.493	
			SDHI (2ch)*2		62.5	9.858	0.616	
			PDMIF		62.5	1.939	0.121	
		Analog	ADC16H (2 units)*2		125	66.267	8.283	VSCR_1
			DAC12 (2ch)*2		62.5	0.325	0.020	
			TSN		62.5	0.115	0.007	
			ACMPHS (4ch)*2		62.5	0.173	0.011	
		Human machine interfaces	CEU		125	18.383	2.298	VSCR_1
		Event link	ELC		62.5	5.075	0.317	VSCR_1
		Security	RSIP-E50D		125	302.444	37.806	VSCR_1
			DOTF (2ch)*2		62.5	131.817	8.239	
		Data processing	CRC		125	1.455	0.182	VSCR_1
			DOC		125	0.241	0.030	
		System	CAC		62.5	0.946	0.059	VSCR_1
		DMA	DMAC0 (per 1ch)		250	7.278	1.819	VSCR_1
			DMAC1 (per 1ch)		250	6.858	1.715	
			DTC0		250	9.077	2.269	
			DTC1		250	8.716	2.179	
Dynamic current	FSBL operation				250	—	79.9	FSBLCLK[2:0] = 111
					200	—	66.1	FSBLCLK[2:0] = 110
					150	—	51.0	FSBLCLK[2:0] = 101
					133	—	46.6	FSBLCLK[2:0] = 100

Note 1. Regulator and Leak are Internal voltage regulator’s current and MCU’s leakage current.
It is selected according to the temperature of Tj.

Note 2. To determine the current consumption per channel or unit, divide Current [mA] by the number of channels, groups or units.

Table 62.46 Power consumption of each unit (External VDD mode) (1 of 2)

Dynamic current/ Leakage current	MCU Domain	Category	Item	Symbol	Frequency [MHz]	Current [μA/MHz]	Current [mA]	Condition
Leakage Current	Analog	Regulator and Leak*1	Tj = 95℃	I_{CC}	—	—	2.51	—
			Tj = 105℃		—	—	2.91	
			Tj = 115℃		—	—	3.45	
			Tj = 125℃		—	—	4.15	
			Tj = 95℃	I_{DD}	—	—	477	VCL = voltage range 1, PDCTRESWM.PDDE = 0
			Tj = 105℃		—	—	553	
			Tj = 115℃		—	—	622	
			Tj = 125℃		—	—	709	
Dynamic Current	CPU0	Operation with Cache	CoreMark	I_{DD}	1000	145	144	CPUCLK0 = 1000 MHz, VCL = voltage range 1
	CPU1	Operation with Cache	CoreMark		250	166	41	CPUCLK1 = 250 MHz, VCL = voltage range 1
Dynamic Current	Peripheral Unit	Timer	RTC	I_{DD}	62.5	1.229	0.077	VCL = voltage range 1
			GPT32 (14ch)*2		125	65.123	8.140	
			POEG (4 Groups)*2		62.5	1.539	0.096	
			PDG (4ch)*2		125	47.465	5.933	
			AGT (2ch)*2		62.5	1.518	0.095	
			ULPT (2ch)*2		62.5	2.373	0.148	
			WDT0		62.5	0.437	0.027	
			WDT1		62.5	0.446	0.028	
			IWDT		62.5	0.014	0.001	
		Communication interfaces	ESWM	I_{DD}	125	294.026	36.753	VCL = voltage range 1
			USBFS		62.5	7.495	0.468	
			USBHS		125	67.424	8.428	
			SCI (10ch)*2		125	32.336	4.042	
			IIC (3ch)*2		62.5	3.722	0.233	
			I3C		125	9.883	1.235	
			CANFD (2ch)*2		125	6.025	0.753	
			SPI (2ch)*2		125	11.36	1.420	
			OSPI (2ch) *2		62.5	100.8	6.300	
			SSIE (2ch)*2		62.5	7.89	0.493	
			SDHI (2ch)*2		62.5	9.858	0.616	
			PDMIF		62.5	1.939	0.121	

Table 62.46 Power consumption of each unit (External VDD mode) (2 of 2)

Dynamic current/ Leakage current	MCU Domain	Category	Item	Symbol	Frequency [MHz]	Current [μA/MHz]	Current [mA]	Condition
Dynamic Current	Peripheral Unit	Analog	ADC16H (2 units)*2	I_{DD}	125	66.267	8.283	VCL = voltage range 1
			DAC12 (2ch)*2		62.5	0.325	0.020	
			TSN		62.5	0.115	0.007	
			ACMPHS (4ch)*2		62.5	0.173	0.011	
		Human machine interfaces	CEU	I_{DD}	125	18.383	2.298	VCL = voltage range 1
		Event link	ELC	I_{DD}	62.5	5.075	0.317	VCL = voltage range 1
		Security	RSIP-E50D	I_{DD}	125	302.444	37.806	VCL = voltage range 1
			DOTF (2ch)*2		62.5	131.817	8.239	
		Data processing	CRC	I_{DD}	125	1.455	0.182	VCL = voltage range 1
			DOC		125	0.241	0.030	
		System	CAC	I_{DD}	62.5	0.946	0.059	VCL = voltage range 1
		DMA	DMAC0 (per 1ch)	I_{DD}	250	7.278	1.819	VCL = voltage range 1
			DMAC1 (per 1ch)		250	6.858	1.715	
			DTC0		250	9.077	2.269	
			DTC1		250	8.716	2.179	
Dynamic Current	FSBL operation			I_{DD}	250	—	79.9	FSBLCLK[2:0] = 111
					200	—	66.1	FSBLCLK[2:0] = 110
					150	—	51.0	FSBLCLK[2:0] = 101
					133	—	46.6	FSBLCLK[2:0] = 100

Note 1. Regulator and Leak are Internal voltage regulator’s current and MCU’s leakage current.
It is selected according to the temperature of Tj.

Note 2. To determine the current consumption per channel or unit, divide Current [mA] by the number of channels, groups or units.

Table 62.47 Outline of operation for each unit (1 of 2)

Peripheral	Outline of operation
RTC	RTC is operating with LOCO.
GPT	Operating modes is set to saw-wave PWM mode. GPT is operating with PCLKD
POEG	Only clear module stop bit.
PDG	PDG is applying delay of 1/128 times GTCLK period.
AGT	AGT is operating with PCLKB.
ULPT	ULPT is operating with LOCO.
WDT	WDT is operating with PCLKB.
IWDT	IWDT is operating with IWDTCLK.

Table 62.47 Outline of operation for each unit (2 of 2)

Peripheral	Outline of operation
ESWM	Communication mode is set to 1 Gbps, MAC Loop back. gPTP timer is enabled. ESWM is operating continuous transmission and reception on two ports simultaneously.
USBFS	Transfer types are set to bulk transfer. USBFS is operating using Full-speed transfer (12 Mbps).
USBHS	Transfer types are set to bulk transfer. USBHS is operating using High-speed transfer.
SCI	SCI is transmitting data in clock synchronous mode.
IIC	Communication format is set to I2C-bus format. IIC is transmitting data in master mode.
I3C	Communication format is set to I3C SDR format. I3C is transmitting data in master mode (12.5MHz).
CANFD	CANFD is transmitting and receiving data in self-test mode 1.
SPI	SPI mode is set to SPI operation (4-wire method). SPI master/slave mode is set to master mode. SPI is transmitting 32-bit width data.
OSPI	OSPI is issuing memory write command to HyperRAM.
SSIE	Communication mode is set to Master. System word length is set to 32 bits. Data word length is set to 20 bits. SSIE is transmitting data using I2S format.
SDHI	Transfer bus mode is set to 8-bit wide bus mode. SDHI is issuing CMD24 (single-block write).
PDMIF	PDMIF is detecting sound activity for 3 channels.
ADC16H	Resolution is set to 16-bit accuracy. Conversion Data Operation Control B Register is set to 16 times average mode. ADC is converting the analog input in continuous scan mode. ADC is operating with ADCCLK.
DAC12	DAC12 is outputting the conversion result while updating the value of data register.
TSN	TSN is operating.
ACMPHS	ACMPHS is operating.
CEU	CEU is capturing data and transferring to the SRAM.
ELC	Only clear module stop bit.
RSIP-E50D	RSIP is doing self-test operation.
DOTF	DOTF is doing decryption with AES.
CRC	CRC is generating CRC code using 32-bit CRC32-C polynomial.
DOC	DOC is operating in data comparison mode.
CAC	Measurement target clocks is set to PCLKB. Measurement reference clocks is set to PCLKB. CAC is measuring the clock frequency accuracy.
DMAC	Bit length of transfer data is set to 32 bits. Transfer mode is set to block transfer mode. DMAC is transferring data from SRAM0 to SRAM0.
DTC	Bit length of transfer data is set to 32 bits. Transfer mode is set to block transfer mode. DTC is transferring data from SRAM0 to SRAM0.

62.2.8.2 Example of Tj Calculation

Assumption :

- Package 289-pin BGA : θja = 20 °C/W
- Ta = 65 °C
- $I_{CC} + I_{CC\_DCDC}$ = 320 mA
- VCC = 3.5 V (VCC = VCC2 = AVCC0 = AVCC_USBHS = VCC_USB = VCC_USBHS)
- I_{OH} = 1 mA, V_{OH} = VCC – 0.5 V, 12 Outputs
- I_{OL} = 20 mA, V_{OL} = 1.0 V, 8 Outputs
- I_{OL} = 1 mA, V_{OL} = 0.5 V, 12 Outputs

- C_{in} = 8 pF, 32 pins, Input frequency = 10 MHz
- C_{load} = 30 pF, 32 pins, Output frequency = 10 MHz

Static current of IO = Σ (VOL × IOL) / Voltage + Σ ((VCC - VOH) × IOH) / Voltage

= (20 mA × 1 V) × 8 / 3.5 V + (1 mA × 0.5 V) × 12 / 3.5 V + ((VCC - (VCC - 0.5 V)) × 1 mA) × 12 / 3.5 V

= 45.7 mA + 1.71 mA + 1.71 mA

= 49.1 mA

Dynamic current of IO = Σ IO (Cin + Cload) × IO switching frequency × Voltage

= ((8 pF × 32) × 10 MHz + (30 pF × 32) × 10 MHz) × 3.5 V

= 42.6 mA

Total power consumption = Voltage × (Static current + Dynamic current)

= (320 mA × 3.5 V) + (49.1 mA + 42.6 mA) × 3.5 V

= 1441 mW (1.441 W)

Tj = Ta + θja × Total power consumption

= 65 °C + 20 °C/W × 1.441W

= 93.82 °C

62.3 AC Characteristics

62.3.1 Frequency

Table 62.48 Operation frequency value in high-speed mode (1 of 8)

Parameter			Symbol	Min	Typ	Max	Unit
Operation frequency	PLL1 output clock P (PLL1P)	BGA package, 0 °C ≤ Tj ≤ 95 °C (Product group A), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)	f	—	—	1000	MHz
		BGA package, –40 °C ≤ Tj ≤ 105 °C (Product group B), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	800	
		BGA package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	600	
		BGA package, 0 °C ≤ Tj ≤ 95 °C (Product group A), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	600	
		BGA package, –40 °C ≤ Tj ≤ 105 °C (Product group B), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	600	
		BGA package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	600	
		HLQFP package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_1 (DCDC mode)		—	—	600	
		HLQFP package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_2 (DCDC mode)		—	—	600	
	Other PLL output clock (PLL1Q, PLL1R, PLL2P, PLL2Q, PLL2R)	BGA package, 0 °C ≤ Tj ≤ 95 °C (Product group A), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	1200	
		BGA package, –40 °C ≤ Tj ≤ 105 °C (Product group B), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	1200	
		BGA package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	1200	
		BGA package, 0 °C ≤ Tj ≤ 95 °C (Product group A), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	1200	
		BGA package, –40 °C ≤ Tj ≤ 105 °C (Product group B), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	1200	
		BGA package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	1200	
		HLQFP package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_1 (DCDC mode)		—	—	1200	
		HLQFP package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_2 (DCDC mode)		—	—	1200	

Table 62.48 Operation frequency value in high-speed mode (2 of 8)

Parameter			Symbol	Min	Typ	Max	Unit
Operation frequency	CPU0 clock (CPUCLK0)*1	BGA package, 0 °C ≤ Tj ≤ 95 °C (Product group A), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)	f	—	—	1000	MHz
		BGA package, –40 °C ≤ Tj ≤ 105 °C (Product group B), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	800	
		BGA package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	600	
		BGA package, 0 °C ≤ Tj ≤ 95 °C (Product group A), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	600	
		BGA package, –40 °C ≤ Tj ≤ 105 °C (Product group B), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	600	
		BGA package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	600	
		HLQFP package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_1 (DCDC mode)		—	—	600	
		HLQFP package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_2 (DCDC mode)		—	—	600	
	CPU1 clock (CPUCLK1)*1	BGA package, 0 °C ≤ Tj ≤ 95 °C (Product group A), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	250	
		BGA package, –40 °C ≤ Tj ≤ 105 °C (Product group B), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	200	
		BGA package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	200	
		BGA package, 0 °C ≤ Tj ≤ 95 °C (Product group A), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	150	
		BGA package, –40 °C ≤ Tj ≤ 105 °C (Product group B), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	150	
		BGA package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	150	
		HLQFP package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_1 (DCDC mode)		—	—	200	
		HLQFP package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_2 (DCDC mode)		—	—	150	

Table 62.48 Operation frequency value in high-speed mode (3 of 8)

Parameter			Symbol	Min	Typ	Max	Unit
Operation frequency	System clock (ICLK)*1	BGA package, 0 °C ≤ Tj ≤ 95 °C (Product group A), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)	f	—	—	250	MHz
		BGA package, –40 °C ≤ Tj ≤ 105 °C (Product group B), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	200	
		BGA package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	200	
		BGA package, 0 °C ≤ Tj ≤ 95 °C (Product group A), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	150	
		BGA package, –40 °C ≤ Tj ≤ 105 °C (Product group B), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	150	
		BGA package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	150	
		HLQFP package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_1 (DCDC mode)		—	—	200	
		HLQFP package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_2 (DCDC mode)		—	—	150	

Table 62.48 Operation frequency value in high-speed mode (4 of 8)

Parameter			Symbol	Min	Typ	Max	Unit
Operation frequency	MRAM bus clock (MRICLK)*1	BGA package, 0 °C ≤ Tj ≤ 95 °C (Product group A), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)	f	—	—	250	MHz
		BGA package, –40 °C ≤ Tj ≤ 105 °C (Product group B), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	200	
		BGA package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	200	
		BGA package, 0 °C ≤ Tj ≤ 95 °C (Product group A), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	150	
		BGA package, –40 °C ≤ Tj ≤ 105 °C (Product group B), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	150	
		BGA package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	150	
		HLQFP package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_1 (DCDC mode)		—	—	200	
		HLQFP package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_2 (DCDC mode)		—	—	150	
	MRAM clock (MRPCLK)*1	BGA package, 0 °C ≤ Tj ≤ 95 °C (Product group A), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	125	
		BGA package, –40 °C ≤ Tj ≤ 105 °C (Product group B), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	100	
		BGA package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	100	
		BGA package, 0 °C ≤ Tj ≤ 95 °C (Product group A), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	75	
		BGA package, –40 °C ≤ Tj ≤ 105 °C (Product group B), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	75	
		BGA package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	75	
		HLQFP package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_1 (DCDC mode)		—	—	100	
		HLQFP package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_2 (DCDC mode)		—	—	75	

Table 62.48 Operation frequency value in high-speed mode (5 of 8)

Parameter			Symbol	Min	Typ	Max	Unit
Operation frequency	Peripheral module clock (PCLKA)*1	BGA package, 0 °C ≤ Tj ≤ 95 °C (Product group A), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)	f	—	—	125	MHz
		BGA package, –40 °C ≤ Tj ≤ 105 °C (Product group B), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	100	
		BGA package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	100	
		BGA package, 0 °C ≤ Tj ≤ 95 °C (Product group A), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	75	
		BGA package, –40 °C ≤ Tj ≤ 105 °C (Product group B), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	75	
		BGA package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	75	
		HLQFP package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_1 (DCDC mode)		—	—	100	
		HLQFP package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_2 (DCDC mode)		—	—	75	
	Peripheral module clock (PCLKB)*1	BGA package, 0 °C ≤ Tj ≤ 95 °C (Product group A), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	62.5	
		BGA package, –40 °C ≤ Tj ≤ 105 °C (Product group B), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	50	
		BGA package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	50	
		BGA package, 0 °C ≤ Tj ≤ 95 °C (Product group A), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	37.5	
		BGA package, –40 °C ≤ Tj ≤ 105 °C (Product group B), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	37.5	
		BGA package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	37.5	
		HLQFP package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_1 (DCDC mode)		—	—	50	
		HLQFP package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_2 (DCDC mode)		—	—	37.5	

Table 62.48 Operation frequency value in high-speed mode (6 of 8)

Parameter			Symbol	Min	Typ	Max	Unit
Operation frequency	Peripheral module clock (PCLKD)*1	BGA package, 0 °C ≤ Tj ≤ 95 °C (Product group A), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)	f	—	—	250	MHz
		BGA package, –40 °C ≤ Tj ≤ 105 °C (Product group B), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	200	
		BGA package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	200	
		BGA package, 0 °C ≤ Tj ≤ 95 °C (Product group A), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	150	
		BGA package, –40 °C ≤ Tj ≤ 105 °C (Product group B), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	150	
		BGA package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	150	
		HLQFP package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_1 (DCDC mode)		—	—	200	
		HLQFP package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_2 (DCDC mode)		—	—	150	
	Peripheral module clock (PCLKE)*1	BGA package, 0 °C ≤ Tj ≤ 95 °C (Product group A), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	250	
		BGA package, –40 °C ≤ Tj ≤ 105 °C (Product group B), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	200	
		BGA package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	200	
		BGA package, 0 °C ≤ Tj ≤ 95 °C (Product group A), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	150	
		BGA package, –40 °C ≤ Tj ≤ 105 °C (Product group B), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	150	
		BGA package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	150	
		HLQFP package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_1 (DCDC mode)		—	—	200	
		HLQFP package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_2 (DCDC mode)		—	—	150	

Table 62.48 Operation frequency value in high-speed mode (7 of 8)

Parameter				Symbol	Min	Typ	Max	Unit
Operation frequency	External bus clock (BCLK)*1	VCC ≥ 2.7 V	BGA package, 0 °C ≤ Tj ≤ 95 °C (Product group A), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)	f	—	—	125	MHz
			BGA package, –40 °C ≤ Tj ≤ 105 °C (Product group B), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	100	
			BGA package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_1 (DCDC mode), voltage range 1 (External VDD mode)		—	—	100	
			BGA package, 0 °C ≤ Tj ≤ 95 °C (Product group A), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	75	
			BGA package, –40 °C ≤ Tj ≤ 105 °C (Product group B), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	75	
			BGA package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_2 (DCDC mode), voltage range 2 (External VDD mode)		—	—	75	
			HLQFP package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_1 (DCDC mode)		—	—	100	
			HLQFP package, –40 °C ≤ Tj ≤ 125 °C (Product group C), VSCR_2 (DCDC mode)		—	—	75	
		VCC ≥ 1.62 V			—	—	60	
	EBCLK pin output	VCC ≥ 2.7 V			—	—	60	
		VCC ≥ 1.62 V			—	—	30	
	SDCLK pin output	VCC ≥ 3.0 V			—	—	133	

Table 62.48 Operation frequency value in high-speed mode (8 of 8)

Parameter			Symbol	Min	Typ	Max	Unit
Operation frequency	SCI clock (SCICLK)		f	—	—	120	MHz
	SPI clock (SPICLK)			—	—	333	
	Octal SPI clock (OCTACLK)			—	—	333	
	CANFD core clock (CANFDCLK)			—	—	80	
	ADC clock (ADCCLK)			25	—	120	
	GPT clock (GPTCLK)			—*2	—	300	
	USB clock (USBCLK)			—	—	48	
	USB clock (USB60CLK)			—	—	60	
	I3C clock (I3CCLK)			—	—	200	
	Asynchronous external bus clock (BCLKA)			—	—	133	
	EtherSW clock (ESWCLK)	BGA package		—	—	250	
		HLQFP package		—	—	125	
	EtherSW-PHY clock (ESWPHYCLK)			—	—	500	

Note 1. See section 9, Clock Generation Circuit for the relationship between the CPUCLK0, CPUCLK1, ICLK, MRICLK, MRPCLK, PCLKA, PCLKB, PCLKC, PCLKD, PCLKE, and BCLK frequencies.

Note 2. When the GPTCLK is used for A/D conversion clock, the GPTCLK frequency must be at least 25 MHz.

62.3.2 Clock Timing

Table 62.49 Clock timing except for sub-clock oscillator (1 of 3)

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
EBCLK pin output cycle time	VCC = 2.70 V or above	t_{Bcyc}	16.6	—	—	ns	Figure 62.25
	VCC = 1.62 V or above		33.3	—	—		
EBCLK pin output high pulse width	VCC = 2.70 V or above	t_{CH}	3.3	—	—	ns	
	VCC = 1.62 V or above		9.6	—	—		
EBCLK pin output low pulse width	VCC = 2.70 V or above	t_{CL}	3.3	—	—	ns	
	VCC = 1.62 V or above		9.6	—	—		
EBCLK pin output rise time	VCC = 2.70 V or above	t_{Cr}	—	—	5.0	ns	
	VCC = 1.62 V or above		—	—	7.0		
EBCLK pin output fall time	VCC = 2.70 V or above	t_{Cf}	—	—	5.0	ns	
	VCC = 1.62 V or above		—	—	7.0		
SDCLK pin output cycle time		t_{SDcyc}	7.52	—	—	ns	
SDCLK pin output high pulse width		t_{CH}	1.0	—	—	ns	
SDCLK pin output low pulse width		t_{CL}	1.0	—	—	ns	
SDCLK pin output rise time		t_{Cr}	—	—	2.7	ns	
SDCLK pin output fall time		t_{Cf}	—	—	2.7	ns	
EXTAL external clock input cycle time		t_{EXcyc}	20.80	—	—	ns	Figure 62.26
EXTAL external clock input high pulse width		t_{EXH}	5.30	—	—	ns	
EXTAL external clock input low pulse width		t_{EXL}	5.30	—	—	ns	
EXTAL external clock rise time		t_{EXr}	—	—	3.0	ns	
EXTAL external clock fall time		t_{EXf}	—	—	3.0	ns	
Main clock oscillator frequency		f_{MAIN}	8	—	48	MHz	—

Table 62.49 Clock timing except for sub-clock oscillator (2 of 3)

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Main clock oscillation stabilization wait time (crystal)*1		$t_{MAINOSCWT}$	—	—	—*1	ms	Figure 62.27
LOCO clock oscillation frequency		f_{LOCO}	29.4912	32.768	36.0448	kHz	—
LOCO clock oscillation stabilization wait time		t_{LOCOWT}	—	—	26.0	µs	Figure 62.28
MOCO clock oscillation frequency		F_{MOCO}	7.2	8.0	8.8	MHz	—
MOCO clock oscillation stabilization wait time		t_{MOCOWT}	—	—	3.5	µs	—
HOCO clock oscillator oscillation frequency	Without FLL (Standard Product)	f_{HOCO16}	15.776	16.000	16.224	MHz	–20 ≤ Tj ≤ 105 °C
		f_{HOCO18}	17.748	18.000	18.252		
		f_{HOCO20}	19.720	20.000	20.280		
		f_{HOCO32}	31.552	32.000	32.448		
		f_{HOCO48}	47.328	48.000	48.672		
		f_{HOCO16}	15.712	16.000	16.288		–40 ≤ Tj ≤ 125 °C
		f_{HOCO18}	17.676	18.000	18.324		
		f_{HOCO20}	19.640	20.000	20.360		
		f_{HOCO32}	31.424	32.000	32.576		
		f_{HOCO48}	47.136	48.000	48.864		
	Without FLL (SiP Product)	f_{HOCO16}	15.744	16.000	16.256		–20 ≤ Tj ≤ 105 °C
		f_{HOCO18}	17.712	18.000	18.288		
		f_{HOCO20}	19.680	20.000	20.320		
		f_{HOCO32}	31.488	32.000	32.512		
		f_{HOCO48}	47.232	48.000	48.768		
		f_{HOCO16}	15.680	16.000	16.320		–40 ≤ Tj ≤ 125 °C
		f_{HOCO18}	17.640	18.000	18.360		
		f_{HOCO20}	19.600	20.000	20.400		
		f_{HOCO32}	31.360	32.000	32.640		
		f_{HOCO48}	47.040	48.000	48.960		
	With FLL	f_{HOCO16}	15.960	16.000	16.040		–40 ≤ Tj ≤ 125 °C Sub-clock frequency accuracy is ±50 ppm.
		f_{HOCO18}	17.955	18.000	18.045		
		f_{HOCO20}	19.950	20.000	20.050		
		f_{HOCO32}	31.920	32.000	32.080		
		f_{HOCO48}	47.880	48.000	48.120		
HOCO clock oscillation stabilization wait time*2		t_{HOCOWT}	—	—	15.0	µs	—
HOCO stop width time		$t_{HOCOSTP}$	1	—	—	µs	Figure 62.31
HOCO period jitter		—	-3	—	3	ps	—
FLL stabilization wait time (Standard Product)		t_{FLLWT}	—	—	1.92	ms	—
FLL stabilization wait time (SiP Product)		t_{FLLWT}	—	—	2.15	ms	—
PLL1/PLL2 clock frequency		f_{PLL}	60	—	1200	MHz	—
PLL1/PLL2 clock oscillation stabilization wait time		t_{PLLWT}	—	—	50	µs	Figure 62.29
PLL1/PLL2 period jitter		—	—	±52	—	ps	—

Table 62.49 Clock timing except for sub-clock oscillator (3 of 3)

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
PLL1/PLL2 long term jitter	—	—	±300	—	ps	Term: 1 µs, 10 µs

Note 1. When setting up the main clock oscillator, ask the oscillator manufacturer for an oscillation evaluation, and use the results as the recommended oscillation stabilization time. Set the MOSCWTCR register to a value equal to or greater than the recommended value.
After changing the setting in the MOSCCR.MOSTP bit to start main clock operation, read the OSCSF.MOSCSF flag to confirm that it is 1, and then start using the main clock oscillator.

Note 2. This is the time from release from reset state until the HOCO oscillation frequency (f_{HOCO}) reaches the range for guaranteed operation.

Table 62.50 Clock timing for the sub-clock oscillator

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Sub-clock frequency	f_{SUB}	—	32.768	—	kHz	—
Sub-clock oscillation stabilization wait time	$t_{SUBOSCWT}$	—	—	—*1	s	Figure 62.30

Note 1. When setting up the sub-clock oscillator, ask the oscillator manufacturer for an oscillation evaluation and use the results as the recommended oscillation stabilization time.
After changing the setting in the SOSCCR.SOSTP bit to start sub-clock operation, only start using the sub-clock oscillator after the sub-clock oscillation stabilization time elapses with an adequate margin. A value that is two times the value shown is recommended.

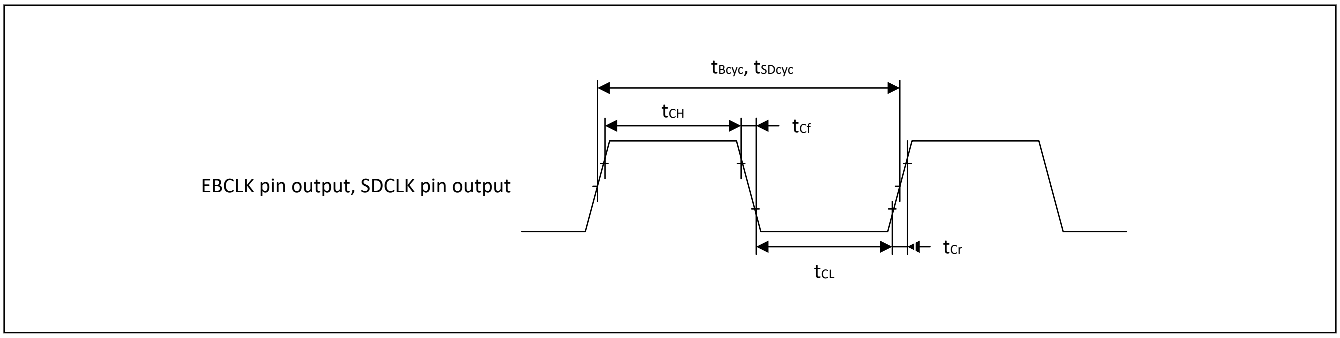


Figure 62.25 EBCLK and SDCLK output timing

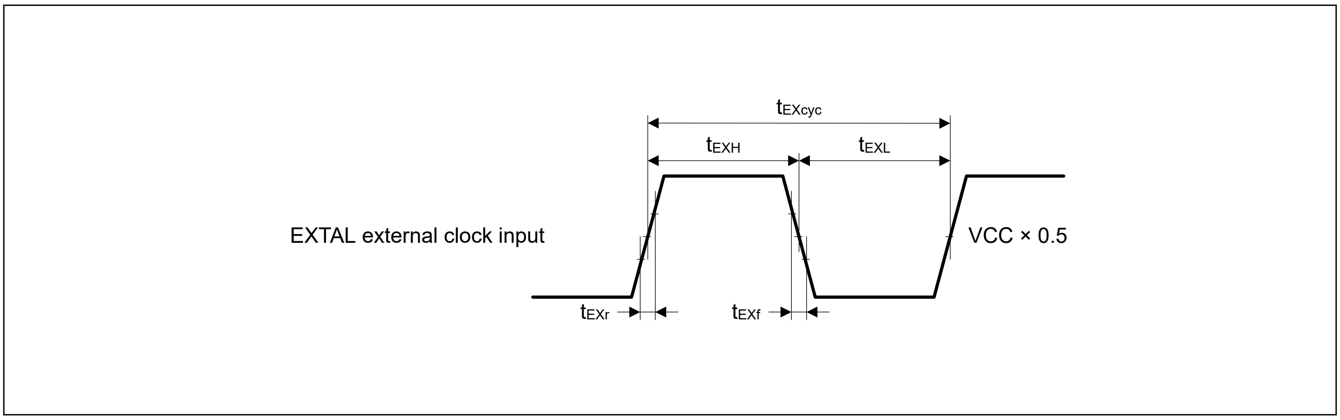


Figure 62.26 EXTAL external clock input timing

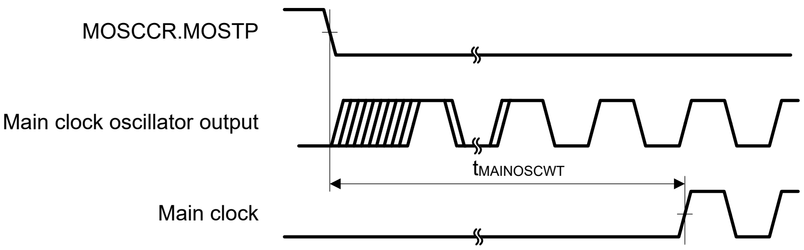


Figure 62.27 Main clock oscillation start timing

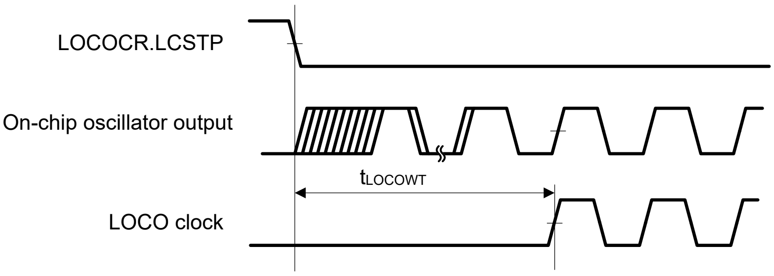
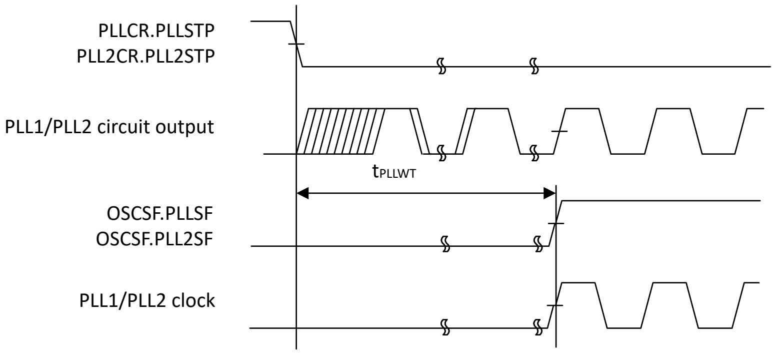


Figure 62.28 LOCO clock oscillation start timing



Note: Only operate the PLL is operated after main clock oscillation has stabilized.

Figure 62.29 PLL1/PLL2 clock oscillation start timing

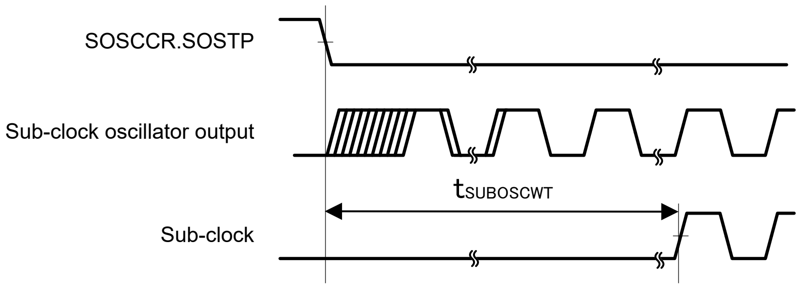


Figure 62.30 Sub-clock oscillation start timing

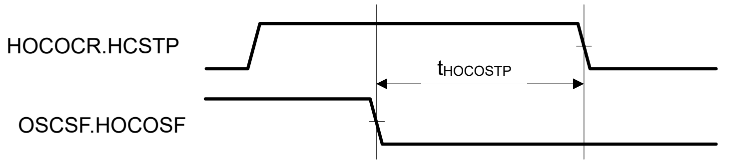


Figure 62.31 HOCO stop width time

62.3.3 Reset Timing

Table 62.51 Reset timing

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
RES pulse width	Power-on	—	t_{RESWP}	2.4	—	—	ms	Figure 62.32
	Deep Software Standby Mode 1	DPSBYCR.DCSSMODE = 01	t_{RESWD}	0.51	—	—	ms	Figure 62.33
		DPSBYCR.DCSSMODE = 10		0.67	—	—		
		DPSBYCR.DCSSMODE = 11		1.00	—	—		
	Deep Software Standby Mode 2	DPSBYCR.DCSSMODE = 01		0.51	—	—		
		DPSBYCR.DCSSMODE = 10		0.67	—	—		
		DPSBYCR.DCSSMODE = 11		1.00	—	—		
	Deep Software Standby Mode 3	DPSBYCR.DCSSMODE = 01		0.68	—	—		
		DPSBYCR.DCSSMODE = 10		0.84	—	—		
		DPSBYCR.DCSSMODE = 11		1.20	—	—		
	Software Standby Mode		t_{RESWS}	0.55	—	—	ms	
	CPU Deep Sleep mode (Subosc operation)		$t_{RESWSODS}$	0.16	—	—	ms	
	CPU Deep Sleep mode (Other than SOSC operation)		t_{RESWDS}	0.04	—	—	ms	
	SOSC operation	PGSCR.PGS = 1	t_{RESWSO}	0.27	—	—	ms	
		PGSCR.PGS = 0		0.30	—	—		
	Other than above	PGSCR.PGS = 1	t_{RESW}	0.15	—	—	ms	
		PGSCR.PGS = 0		0.18	—	—		
Wait time after RES cancellation			t_{RESWT}	—	78.7	79.1	µs	Figure 62.32
Wait time after internal reset cancellation (IWDT reset, WDT0/1 reset, CPU0/1 Lockup reset, Bus Error reset, Common Memory Error reset, Software reset, Local Memory 0/1 error reset, Temperature monitor reset)			t_{RESW2}	—	78.7	79.1	µs	—

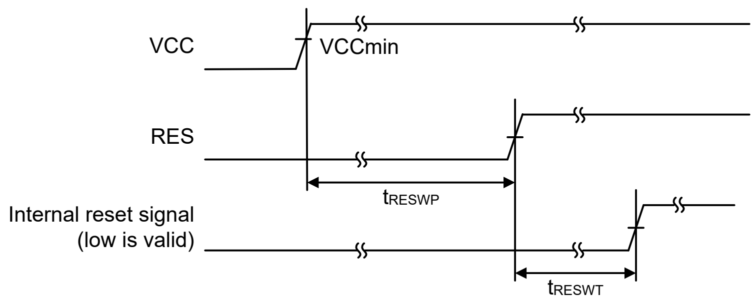


Figure 62.32 RES pin input timing under the condition that VCC exceeds V_{POR} voltage threshold

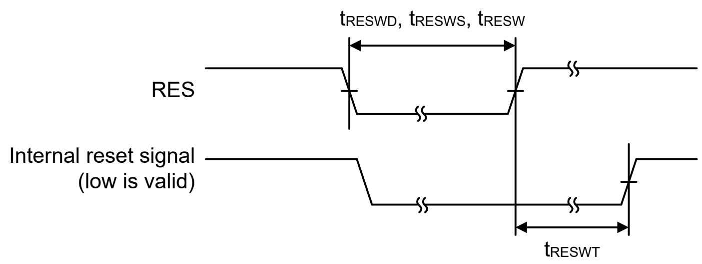


Figure 62.33 Reset input timing

62.3.4 Wakeup Timing

Table 62.52 Timing of recovery from low power modes (1 of 3)

Parameter		Fast return function*9	Symbol	Min	Typ	Max	Unit	Test conditions
Recovery time from CPU Deep Sleep mode	CPU0 Deep Sleep mode	—	t_{DSLP}*11	—	6.14	9.45	µs	—
	CPU1 Deep Sleep mode	—		—	7.71	15.66	µs	

Table 62.52 Timing of recovery from low power modes (2 of 3)

Parameter			Fast return function*9	Symbol	Min	Typ	Max	Unit	Test conditions
Recovery time from Software Standby mode*12	Crystal resonator connected to main clock oscillator	System clock source is main clock oscillator*1 MOSCSCR.MOSCSOKP = 0	Enabled	t_{SBYMC} *10	—	2.09	2.14	ms	Figure 62.34 The division ratio of all oscillators is 1.
		System clock source is main clock oscillator*1 MOSCSCR.MOSCSOKP = 1	Enabled		—	44.9	94.6	µs	
		System clock source is PLL1P with main clock oscillator*2 MOSCSCR.MOSCSOKP = 0	Enabled	t_{SBYPC} *10	—	2.21	2.27	ms	
		System clock source is PLL1P with main clock oscillator*2 MOSCSCR.MOSCSOKP = 1	Enabled		—	135	197	µs	
	External clock input to main clock oscillator	System clock source is main clock oscillator*3	Enabled	t_{SBYEX} *10	—	44.9	94.6	µs	
		System clock source is PLL1P with main clock oscillator*4	Enabled	t_{SBYPE} *10	—	135	197		
	System clock source is sub-clock oscillator*5		Enabled	t_{SBYSC} *10	—	480	481	µs	
	System clock source is HOCO clock oscillator*6		Enabled	t_{SBYHO} *10	—	46.3	96.0	µs	
	System clock source is PLL1P with HOCO*7		Enabled	t_{SBYPH} *10	—	146	208	µs	
	System clock source is MOCO clock oscillator*8		Enabled	t_{SBYMO} *10	—	44.6	87.5	µs	

Table 62.52 Timing of recovery from low power modes (3 of 3)

Parameter			Fast return function*9	Symbol	Min	Typ	Max	Unit	Test conditions
Recovery time from Deep Software Standby mode	Deep Software Standby mode 1	DPSBYCR.DCSSMODE[1:0] = 01	—	t_{DSBY}	—	296	346	µs	Figure 62.35
		DPSBYCR.DCSSMODE[1:0] = 10	—		—	456	506	µs	
		DPSBYCR.DCSSMODE[1:0] = 11	—		—	776	826	µs	
	Deep Software Standby mode 2	DPSBYCR.DCSSMODE[1:0] = 01	—		—	296	346	µs	
		DPSBYCR.DCSSMODE[1:0] = 10	—		—	456	506	µs	
		DPSBYCR.DCSSMODE[1:0] = 11	—		—	776	826	µs	
	Deep Software Standby mode 3	DPSBYCR.DCSSMODE[1:0] = 01	—		—	483	604	µs	
		DPSBYCR.DCSSMODE[1:0] = 10	—		—	643	764	µs	
		DPSBYCR.DCSSMODE[1:0] = 11	—		—	963	1084	µs	
Wait time after cancellation of Deep Software Standby mode			—	t_{DSBYWT}	22.2	—	33.6	µs	

Note 1. When the frequency of the crystal is 48 MHz (Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 0x05) and the greatest value of the internal clock division setting is 1.
Note 2. When the frequency of PLL1P is 1 GHz (Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 0x05) and the greatest value of the internal clock division setting is 16.
Note 3. When the frequency of the external clock is 48 MHz (Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 0x00) and the greatest value of the internal clock division setting is 1.
Note 4. When the frequency of PLL1P is 1 GHz (Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 0x00) and the greatest value of the internal clock division setting is 16.
Note 5. The Sub-clock oscillator frequency is 32.768 kHz and the greatest value of the internal clock division setting is 1.
Note 6. The HOCO frequency is 20 MHz and the greatest value of the internal clock division setting is 1.
Note 7. The PLL frequency is 1 GHz and the greatest value of the internal clock division setting is 16.
Note 8. The MOCO frequency is 8 MHz and the greatest value of the internal clock division setting is 1.
Note 9. For details, see SSCR1.SS2FR bit in the section 11, Low Power Mode.
Note 10. The recovery time can be calculated with the equation of t_{Common} + max(t_{OSCSTB}, t_{PG1}, t_{PGCK}) + max(t_{PG2}, t_{LPW}). And they can be determined with the following values and equations. For n, the greatest value is selected from among the internal clock(CPUCLK0, CPUCLK1, NPUCLK, ICLK, MRICLK, MRPCLK, PCLKm, FCLK, BCLK and EBCLK) division settings (m = A to E). t_{OSCSTB} in the table below means the time when each oscillator is active. When multiple oscillators are active, t_{OSCSTB} is determined by the longest t_{OSCSTB} among the active oscillators.
Note 11. The ICLK frequency is 250 MHz. This recovery time corresponds to t_{PG2}.
Note 12. When Ccyc is 27. See Table 62.54.

Table 62.53 Each element of recovery time

Mode	Wakeup time	Oscillation keep	Fast return function	Typ						Max						Unit
				t_{Common}	t_{OSCSTB}*1	t_{PG1}	t_{PGCK}	t_{PG2}	t_{LPW}	t_{Common}	t_{OSCSTB}*1	t_{PG1}	t_{PGCK}	t_{PG2}	t_{LPW}	
Software Standby mode	t_{SBYMC}	MOSC disabled	Enabled	Ccyc*2/ f_{MOCO} + 2/f_{ICLK}	$t_{MAINOSCWT}$	t_{OSCSTB}/ f_{MOCO} + 208/ f_{MOCO} + 11.6	(10.5 + 2.5n)/ f_{MOCO} + 2.5/ f_{SRCCLK}+ 2/ f_{ICLK}	18/ f_{MOCO} + 9/ f_{ICLK}	2/f_{ICLK} + 2n/ f_{MOSC} + 2/ f_{ICLK}	Ccyc*2/ f_{MOCO} + 2/f_{ICLK}	$t_{MAINOSCWT}$ + 11/0.236	t_{OSCSTB}/ f_{MOCO} + 208/ f_{MOCO} + 51.0	(10.5 + 2.5n)/ f_{MOCO} + 2.5/ f_{SRCCLK}+ 2/ f_{ICLK}	18/ f_{MOCO} + 9/ f_{ICLK}	2/f_{ICLK} + 2n/ f_{MOSC} + 2/ f_{ICLK}	μs
		MOSC enabled	Enabled		3/0.262						14/0.236					μs
	t_{SBYPC}	MOSC disabled	Enabled		$t_{MAINOSCWT}$ +31/0.262				2/f_{ICLK} + 2n/ f_{PLL} + 2/f_{ICLK}		$t_{MAINOSCWT}$ + 42/0.236				2/f_{ICLK} + 2n/ f_{PLL} + 2/f_{ICLK}	μs
		MOSC enabled	Enabled		34/0.262						(14 + 31)/ 0.236					μs
	t_{SBYEX}	—	Enabled		3/0.262				2/f_{ICLK} + 2n/ f_{MOSC} + 2/ f_{ICLK}		14/0.236				2/f_{ICLK} + 2n/ f_{MOSC} + 2/ f_{ICLK}	μs
	t_{SBYPE}	—	Enabled		34/0.262				2/f_{ICLK} + 2n/ f_{PLL} + 2/f_{ICLK}		45/0.236				2/f_{ICLK} + 2n/ f_{PLL} + 2/f_{ICLK}	μs
	t_{SBYSC}	—	Enabled		0				2/f_{ICLK} + 2n/ f_{SOSC} + 2/ f_{ICLK}		0				2/f_{ICLK} + 2n/ f_{SOSC} + 2/ f_{ICLK}	μs
	t_{SBYHO}	—	Enabled		20				2/f_{ICLK} + 2n/ f_{HOCO} + 2/ f_{ICLK}		67				2/f_{ICLK} + 2n/ f_{HOCO} + 2/ f_{ICLK}	μs
	t_{SBYPH}	—	Enabled		140				2/f_{ICLK} + 2n/ f_{PLL} + 2/f_{ICLK}		202				2/f_{ICLK} + 2n/ f_{PLL} + 2/f_{ICLK}	μs
	t_{SBYMO}	—	Enabled		0				2/f_{ICLK} + 2n/ f_{MOCO} + 2/ f_{ICLK}		0				2/f_{ICLK} + 2n/ f_{MOCO} + 2/ f_{ICLK}	μs

Note: The unit of frequency is MHz.
Note 1. If more than one oscillator is operating, the largest value of the operating oscillator in this column is applied.
Note 2. For Ccyc, see Table 62.54.

Table 62.54 Ccyc value

SSCR1.SS2LP [1:0]	VSCR.VSCM[2:0]	SVSCR.SVSCM [2:0]	{PLL1LDOCR.LDOSTP, PLL2LDOCR.LDOSTP, PLL1LDOCR.SKEEP, PLL2LDOCR.SKEEP}	{HOCOLDOCR.LDOSTP, HOCOLDOCR.SKEEP}	Ccyc	Unit
00: SS2LP_0	001 : VSCR_1	001 : SVSCR_1	{1, 1, x, x} or {x, x, 1, 1}	{0, 0}	56	cycle
				Other than above	27	cycle
			Other than above	Don't care	237	cycle
		010 : SVSCR_2	Don't care	Don't care	379	cycle
		011 : SVSCR_3	Don't care	Don't care	591	cycle
		100 : SVSCR_4	Don't care	Don't care	696	cycle
		101 : SVSCR_5	Don't care	Don't care	802	cycle
	010 : VSCR_2	001 : SVSCR_1	Don't care	Don't care	379	cycle
		010 : SVSCR_2	{1, 1, x, x} or {x, x, 1, 1}	{0, 0}	56	cycle
				Other than above	27	cycle
			Other than above	Don't care	237	cycle
		011 : SVSCR_3	Don't care	Don't care	538	cycle
		100 : SVSCR_4	Don't care	Don't care	643	cycle
		101 : SVSCR_5	Don't care	Don't care	749	cycle
01: SS2LP_1	001 : VSCR_1	010 : SVSCR_2	Don't care	Don't care	514	cycle
		011 : SVSCR_3	Don't care	Don't care	726	cycle
		100 : SVSCR_4	Don't care	Don't care	831	cycle
		101 : SVSCR_5	Don't care	Don't care	937	cycle
	010 : VSCR_2	010 : SVSCR_2	{1, 1, x, x} or {x, x, 1, 1}	Don't care	162	cycle
			Other than above	Don't care	327	cycle
		011 : SVSCR_3	Don't care	Don't care	673	cycle
		100 : SVSCR_4	Don't care	Don't care	778	cycle
		101 : SVSCR_5	Don't care	Don't care	884	cycle

Note: x: Don't care

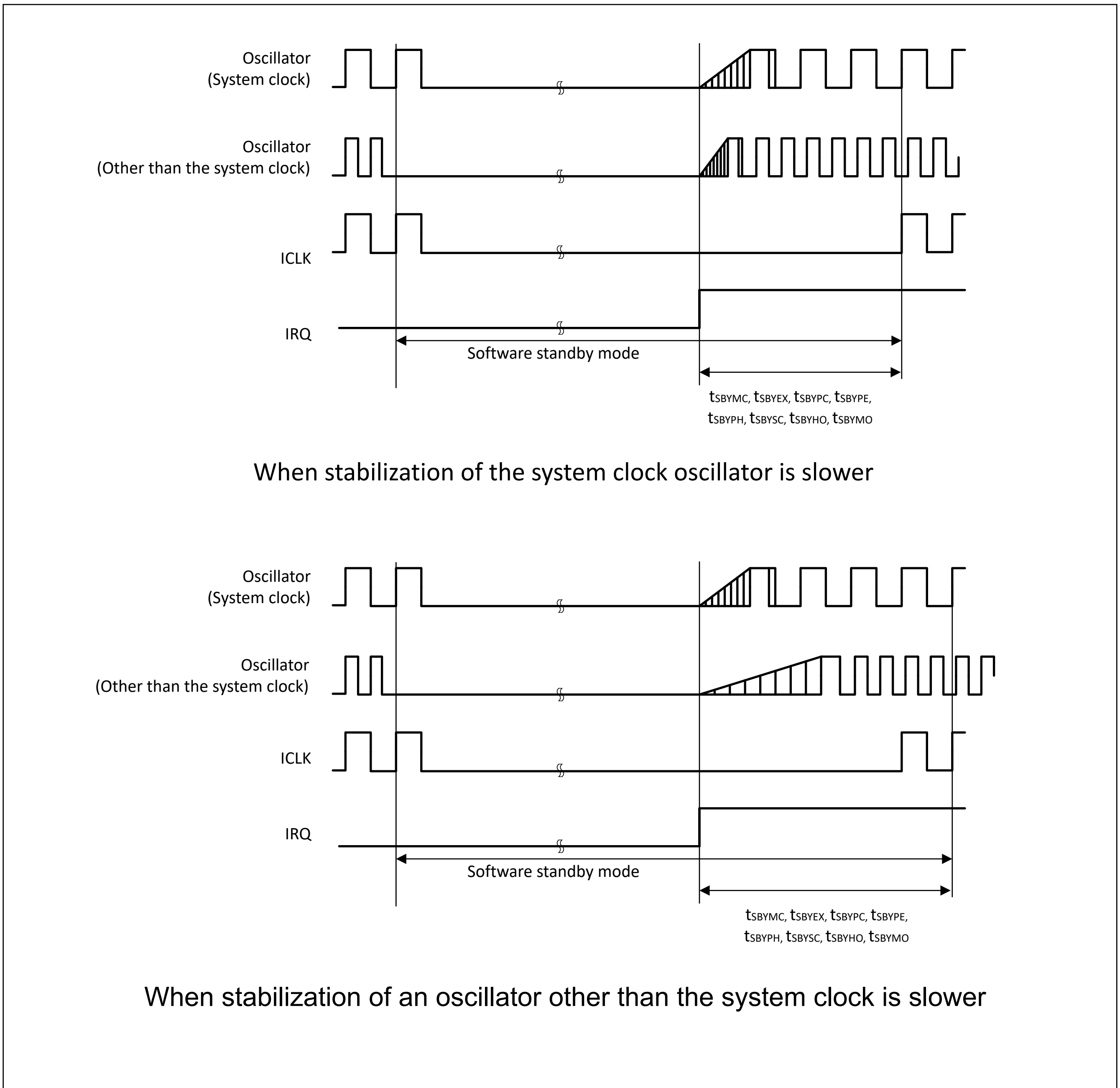


Figure 62.34 Software Standby mode cancellation timing

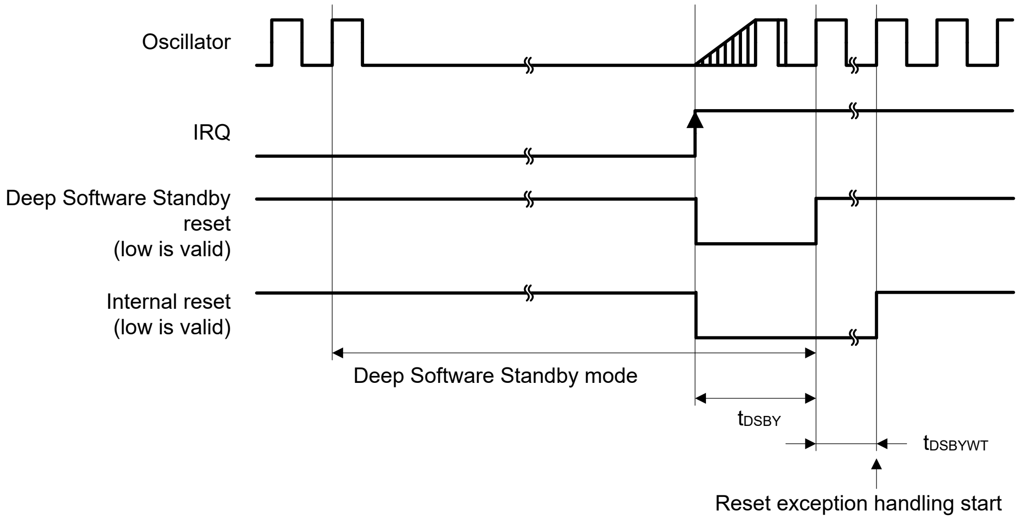


Figure 62.35 Deep Software Standby mode cancellation timing

62.3.5 NMI and IRQ Noise Filter

Table 62.55 NMI and IRQ noise filter

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions	
NMI pulse width	t_{NMIW}	200	—	—	ns	NMI digital filter disabled	$t_{Pcyc} \times 2 \leq 200$ ns
		$t_{Pcyc} \times 2^{*1}$	—	—			$t_{Pcyc} \times 2 > 200$ ns
		200	—	—		NMI digital filter enabled	$t_{NMICK} \times 3 \leq 200$ ns
		$t_{NMICK} \times 3.5^{*2}$	—	—			$t_{NMICK} \times 3 > 200$ ns
IRQ pulse width	t_{IRQW}	200	—	—	ns	IRQ digital filter disabled	$t_{Pcyc} \times 2 \leq 200$ ns
		$t_{Pcyc} \times 2^{*1}$	—	—			$t_{Pcyc} \times 2 > 200$ ns
		200	—	—		IRQ digital filter enabled	$t_{IRQCK} \times 3 \leq 200$ ns
		$t_{IRQCK} \times 3.5^{*3}$	—	—			$t_{IRQCK} \times 3 > 200$ ns

Note: 200 ns minimum in Software Standby mode.
Note: If the system clock source is switched, add 4 clock cycles of the switched source.
Note 1. t_{Pcyc} indicates the PCLKB cycle.
Note 2. t_{NMICK} indicates the cycle of the NMI digital filter sampling clock.
Note 3. t_{IRQCK} indicates the cycle of the IRQi digital filter sampling clock.

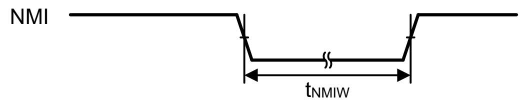


Figure 62.36 NMI interrupt input timing

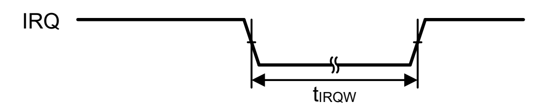


Figure 62.37 IRQ interrupt input timing

62.3.6 Bus Timing

Table 62.56 Bus timing (1 of 3)

Condition 1: When using the CS area controller (CSC).
VCC = VCC_DCDC = VBATT = 1.62 V to 3.6 V, VCC2 = 1.62 V to 3.63 V
BCLK = 8 to 120 MHz, BCLKA = 8 to 120 MHz, EBCLK = 8 to 60 MHz (When VCC = VCC_USB = VBATT = 2.70 to 3.63 V)
BCLK = BCLKA = 8 to 60 MHz, EBCLK = 8 to 30 MHz (When VCC = VCC_USB = VBATT = 1.62 to 3.63 V)
Output load conditions: VOH = VCC × 0.5, VOL = VCC × 0.5, C = 30 pF
EBCLK: High drive output is selected in the port drive capability bit in the PmnPFS register.
Others: Middle drive output is selected in the port drive capability bit in the PmnPFS register.

Condition 2: When using the SDRAM area controller (SDRAMC).
BCLK = SDCLK = 8 to 125 MHz, BCLKA = SDCLK = 8 to 133 MHz
VCC = VCC_DCDC = VBATT = 3.0 to 3.63 V, VCC2 = 1.62 V to 3.63V
Output load conditions: VOH = VCC × 0.5, VOL = VCC × 0.5, C = 15 pF
SDCLK: High-speed high drive output is selected in the port drive capability bit in the PmnPFS register.
Others: High drive output is selected in the port drive capability bit in the PmnPFS register.

Condition 3: When using the SDRAM area controller (SDRAMC) and CS area controller (CSC) simultaneously.
BCLK = SDCLK = 8 to 66 MHz, BCLKA = SDCLK = 8 to 66 MHz
VCC = VCC_DCDC = VBATT = 3.0 to 3.63 V, VCC2 = 1.62 V to 3.63V
Output load conditions: VOH = VCC × 0.5, VOL = VCC × 0.5, C = 15 pF
EBCLK/SDCLK: High drive output is selected in the port drive capability bit in the PmnPFS register.
Others: Middle drive output is selected in the port drive capability bit in the PmnPFS register.

Parameter	Condition	VCC/VCC2	Symbol	Min	Max	Unit	Test conditions
Address delay	Condition1	2.70 V or above	t_{AD}	1.0	12.5	ns	Figure 62.38 to Figure 62.44
		1.62 V or above		1.0	12.5	ns	
	Condition3	3.0 V or above		1.0	10.8	ns	
Byte control delay	Condition1	2.70 V or above	t_{BCD}	1.0	12.5	ns	
		1.62 V or above		1.0	12.5	ns	
	Condition3	3.0 V or above		1.0	10.8	ns	
CS delay	Condition1	2.70 V or above	t_{CSD}	1.0	12.5	ns	
		1.62 V or above		1.0	12.5	ns	
	Condition3	3.0 V or above		1.0	10.8	ns	
ALE delay time	Condition1	2.70 V or above	t_{ALED}	1.0	12.5	ns	
		1.62 V or above		1.0	12.5	ns	
	Condition3	3.0 V or above		1.0	10.8	ns	
RD delay	Condition1	2.70 V or above	t_{RSD}	1.0	12.5	ns	
		1.62 V or above		1.0	12.5	ns	
	Condition3	3.0 V or above		1.0	10.8	ns	
Read data setup time	Condition1	2.70 V or above	t_{RDS}	12.5	—	ns	
		1.62 V or above		20.5	—	ns	
	Condition3	3.0 V or above		10.8	—	ns	
Read data hold time	Condition1	2.70 V or above	t_{RDH}	0	—	ns	
		1.62 V or above		0	—	ns	
	Condition3	3.0 V or above		0	—	ns	
WR/WRn delay	Condition1	2.70 V or above	t_{WRD}	1.0	12.5	ns	
		1.62 V or above		1.0	12.5	ns	
	Condition3	3.0 V or above		1.0	10.8	ns	
Write data delay	Condition1	2.70 V or above	t_{WDD}	—	12.5	ns	
		1.62 V or above		—	12.5	ns	
	Condition3	3.0 V or above		1.0	10.8	ns	

Table 62.56 Bus timing (2 of 3)

Condition 1: When using the CS area controller (CSC).
VCC = VCC_DCDC = VBATT = 1.62 V to 3.6 V, VCC2 = 1.62 V to 3.63 V
BCLK = 8 to 120 MHz, BCLKA = 8 to 120 MHz, EBCLK = 8 to 60 MHz (When VCC = VCC_USB = VBATT = 2.70 to 3.63 V)
BCLK = BCLKA = 8 to 60 MHz, EBCLK = 8 to 30 MHz (When VCC = VCC_USB = VBATT = 1.62 to 3.63 V)
Output load conditions: VOH = VCC × 0.5, VOL = VCC × 0.5, C = 30 pF
EBCLK: High drive output is selected in the port drive capability bit in the PmnPFS register.
Others: Middle drive output is selected in the port drive capability bit in the PmnPFS register.

Condition 2: When using the SDRAM area controller (SDRAMC).
BCLK = SDCLK = 8 to 125 MHz, BCLKA = SDCLK = 8 to 133 MHz
VCC = VCC_DCDC = VBATT = 3.0 to 3.63 V, VCC2 = 1.62 V to 3.63V
Output load conditions: VOH = VCC × 0.5, VOL = VCC × 0.5, C = 15 pF
SDCLK: High-speed high drive output is selected in the port drive capability bit in the PmnPFS register.
Others: High drive output is selected in the port drive capability bit in the PmnPFS register.

Condition 3: When using the SDRAM area controller (SDRAMC) and CS area controller (CSC) simultaneously.
BCLK = SDCLK = 8 to 66 MHz, BCLKA = SDCLK = 8 to 66 MHz
VCC = VCC_DCDC = VBATT = 3.0 to 3.63 V, VCC2 = 1.62 V to 3.63V
Output load conditions: VOH = VCC × 0.5, VOL = VCC × 0.5, C = 15 pF
EBCLK/SDCLK: High drive output is selected in the port drive capability bit in the PmnPFS register.
Others: Middle drive output is selected in the port drive capability bit in the PmnPFS register.

Parameter	Condition	VCC/VCC2	Symbol	Min	Max	Unit	Test conditions
Write data hold time	Condition1	2.70 V or above	t_{WDH}	1.0	—	ns	Figure 62.38 to Figure 62.44
		1.62 V or above		1.0	—	ns	
	Condition3	3.0 V or above		1.0	10.8	ns	
WAIT setup time	Condition1	2.70 V or above	t_{WTS}	12.5	—	ns	
		1.62 V or above		20.5	—	ns	
	Condition3	3.0 V or above		10.8	—	ns	
WAIT hold time	Condition1	2.70 V or above	t_{WTH}	0	—	ns	
		1.62 V or above		0	—	ns	
	Condition3	3.0 V or above		0	—	ns	

Table 62.56 Bus timing (3 of 3)

Condition 1: When using the CS area controller (CSC).
VCC = VCC_DCDC = VBATT = 1.62 V to 3.6 V, VCC2 = 1.62 V to 3.63 V
BCLK = 8 to 120 MHz, BCLKA = 8 to 120 MHz, EBCLK = 8 to 60 MHz (When VCC = VCC_USB = VBATT = 2.70 to 3.63 V)
BCLK = BCLKA = 8 to 60 MHz, EBCLK = 8 to 30 MHz (When VCC = VCC_USB = VBATT = 1.62 to 3.63 V)
Output load conditions: VOH = VCC × 0.5, VOL = VCC × 0.5, C = 30 pF
EBCLK: High drive output is selected in the port drive capability bit in the PmnPFS register.
Others: Middle drive output is selected in the port drive capability bit in the PmnPFS register.

Condition 2: When using the SDRAM area controller (SDRAMC).
BCLK = SDCLK = 8 to 125 MHz, BCLKA = SDCLK = 8 to 133 MHz
VCC = VCC_DCDC = VBATT = 3.0 to 3.63 V, VCC2 = 1.62 V to 3.63V
Output load conditions: VOH = VCC × 0.5, VOL = VCC × 0.5, C = 15 pF
SDCLK: High-speed high drive output is selected in the port drive capability bit in the PmnPFS register.
Others: High drive output is selected in the port drive capability bit in the PmnPFS register.

Condition 3: When using the SDRAM area controller (SDRAMC) and CS area controller (CSC) simultaneously.
BCLK = SDCLK = 8 to 66 MHz, BCLKA = SDCLK = 8 to 66 MHz
VCC = VCC_DCDC = VBATT = 3.0 to 3.63 V, VCC2 = 1.62 V to 3.63V
Output load conditions: VOH = VCC × 0.5, VOL = VCC × 0.5, C = 15 pF
EBCLK/SDCLK: High drive output is selected in the port drive capability bit in the PmnPFS register.
Others: Middle drive output is selected in the port drive capability bit in the PmnPFS register.

Parameter	Condition	VCC/VCC2	Symbol	Min	Max	Unit	Test conditions
Address delay 2 (SDRAM)	Condition 2	3.0 V or above	t_{AD2}	0.8	6.0	ns	Figure 62.45 to Figure 62.51
	Condition 3	3.0 V or above		0.8	10		
CS delay 2 (SDRAM)	Condition 2	3.0 V or above	t_{CSD2}	0.8	6.0	ns	
	Condition 3	3.0 V or above		0.8	10		
DQM delay (SDRAM)	Condition 2	3.0 V or above	t_{DQMD}	0.8	6.0	ns	
	Condition 3	3.0 V or above		0.8	10		
CKE delay (SDRAM)	Condition 2	3.0 V or above	t_{CKED}	0.8	6.0	ns	
	Condition 3	3.0 V or above		0.8	10		
Read data setup time 2 (SDRAM)	Condition 2	3.0 V or above	t_{RDS2}	2.1	—	ns	
	Condition 3	3.0 V or above		6.1	—		
Read data hold time 2 (SDRAM)	Condition 2	3.0 V or above	t_{RDH2}	1.5	—	ns	
	Condition 3	3.0 V or above		1.5	—		
Write data delay 2 (SDRAM)	Condition 2	3.0 V or above	t_{WDD2}	—	6.0	ns	
	Condition 3	3.0 V or above		—	10		
Write data hold time 2 (SDRAM)	Condition 2	3.0 V or above	t_{WDH2}	0.8	—	ns	
	Condition 3	3.0 V or above		0.8	—		
WE delay (SDRAM)	Condition 2	3.0 V or above	t_{WED}	0.8	6.0	ns	
	Condition 3	3.0 V or above		0.8	10		
RAS delay (SDRAM)	Condition 2	3.0 V or above	t_{RASD}	0.8	6.0	ns	
	Condition 3	3.0 V or above		0.8	10		
CAS delay (SDRAM)	Condition 2	3.0 V or above	t_{CASD}	0.8	6.0	ns	
	Condition 3	3.0 V or above		0.8	10		

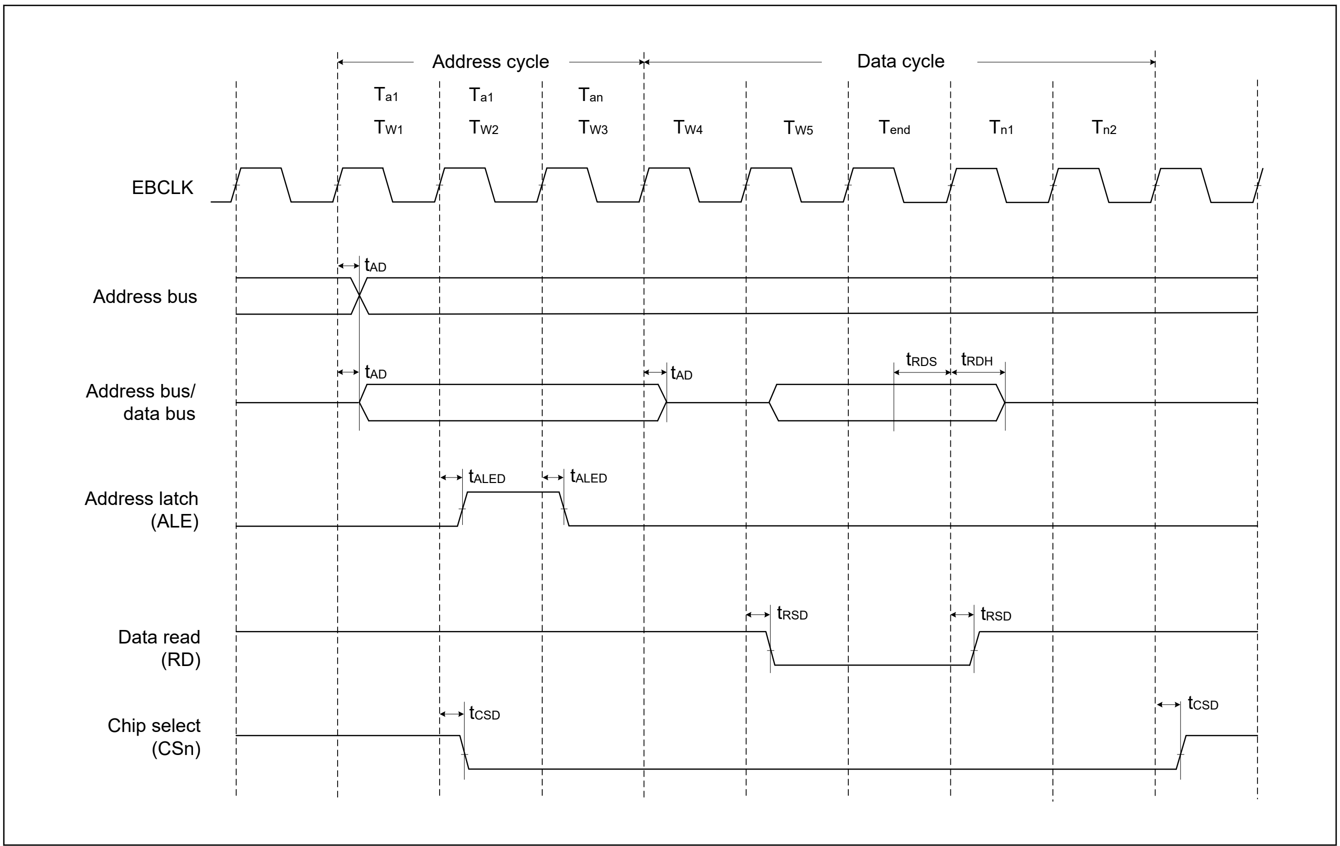


Figure 62.38 Address/data multiplexed bus read access timing

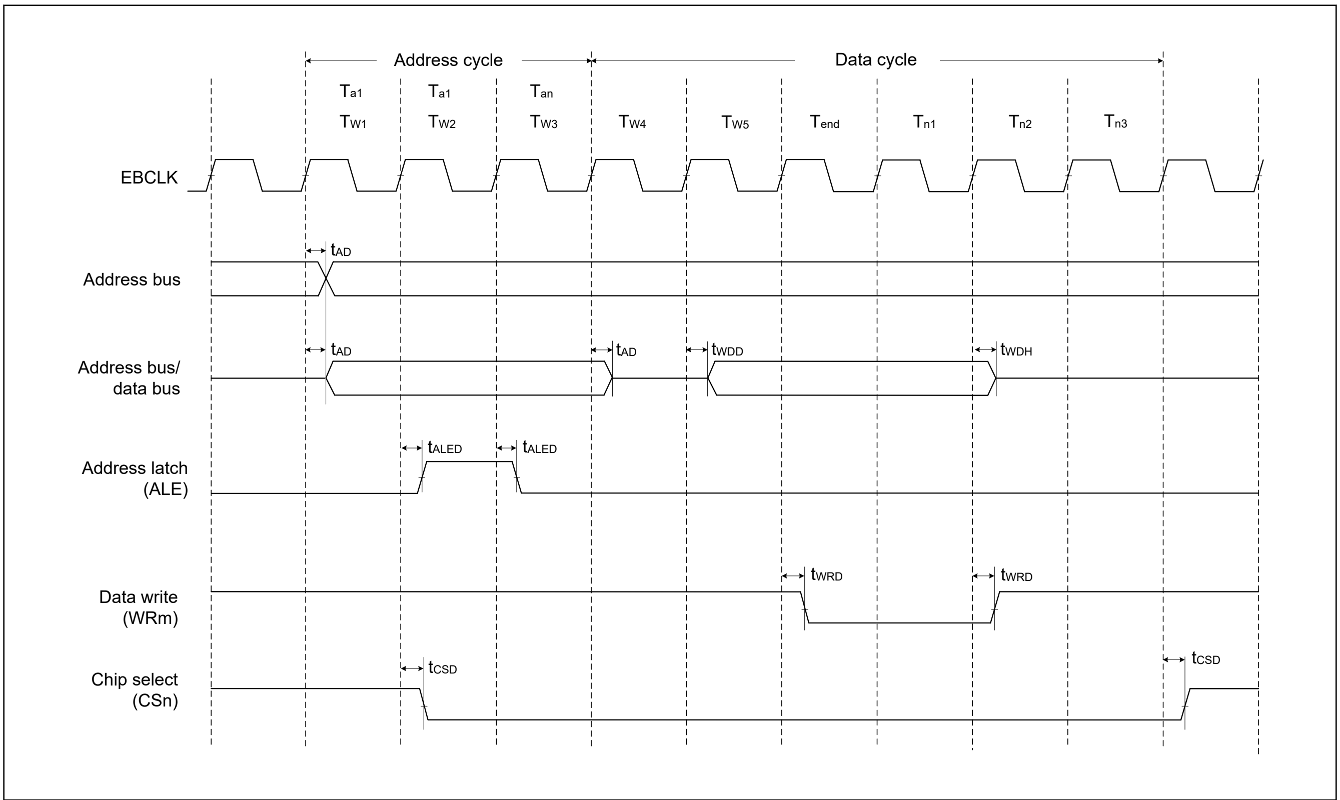


Figure 62.39 Address/data multiplexed bus write access timing

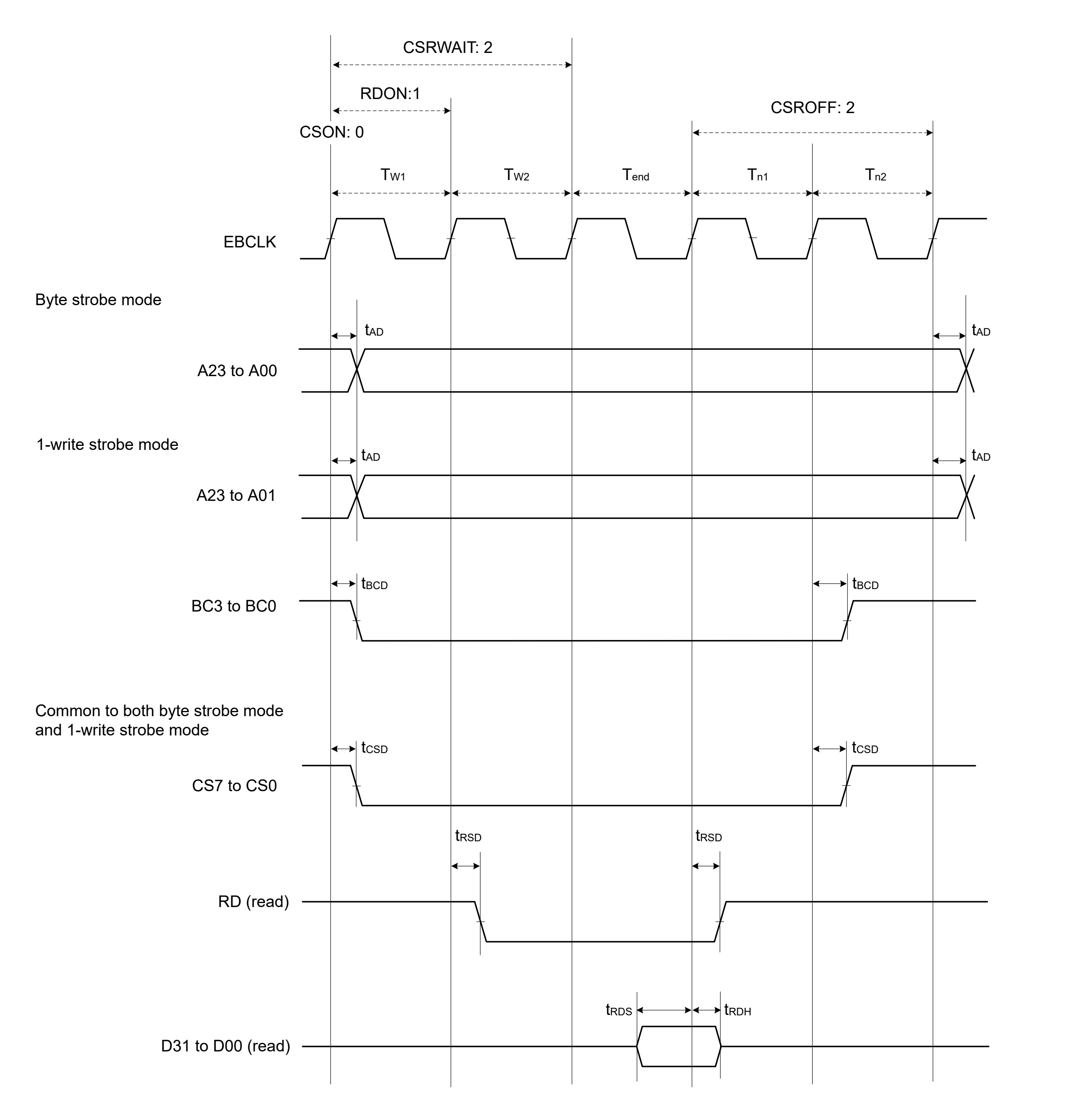
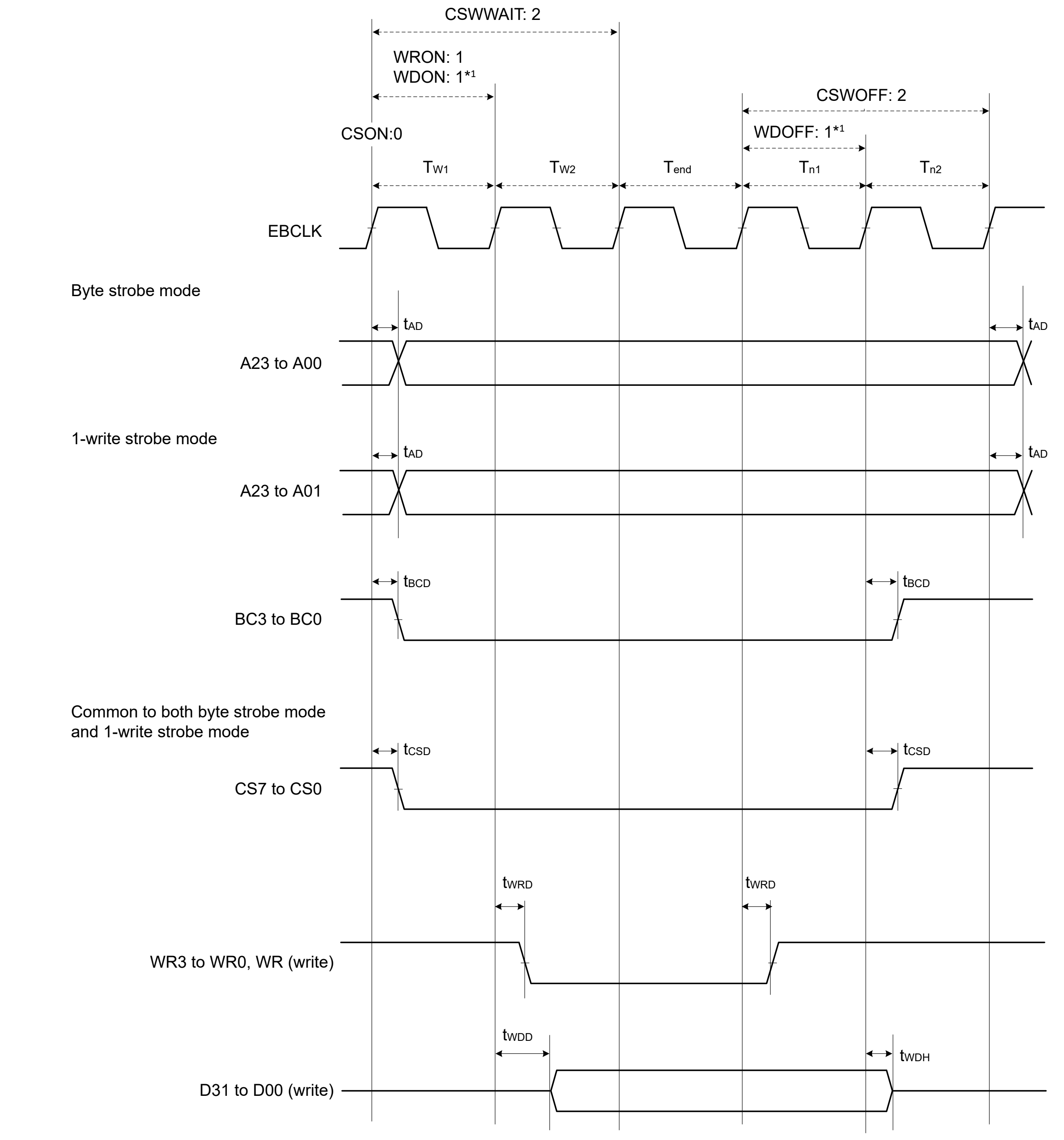


Figure 62.40 External bus timing for normal read cycle with bus clock synchronized



Note 1. Always specify WDON and WDOFF as at least one EBCLK cycle.

Figure 62.41 External bus timing for normal write cycle with bus clock synchronized

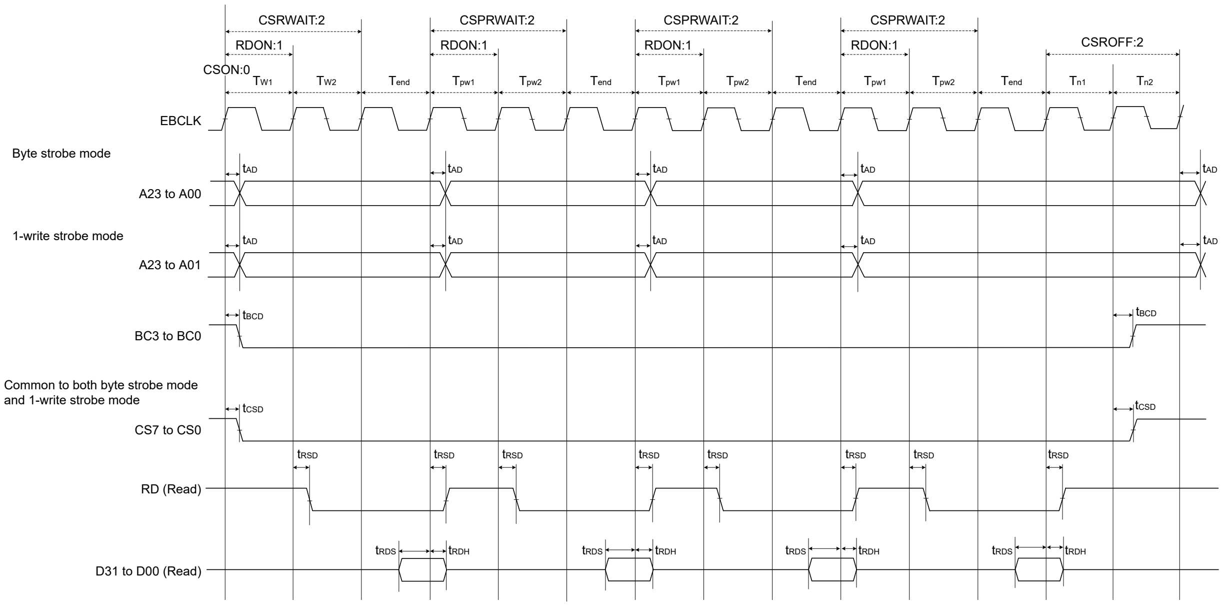
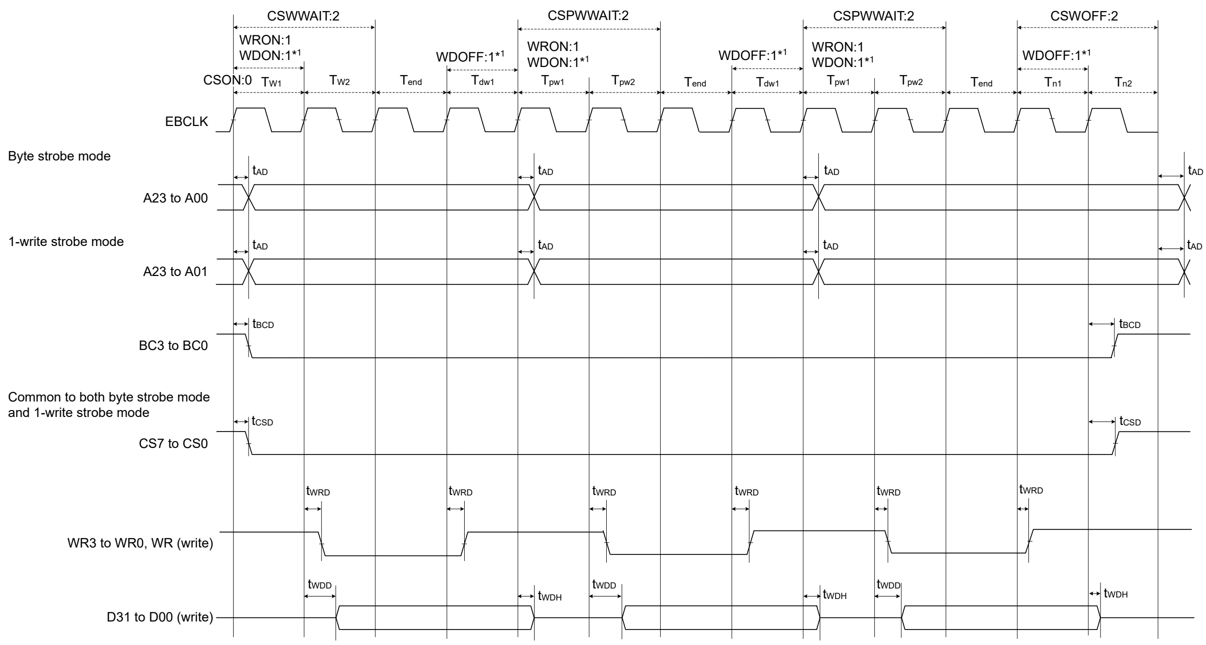


Figure 62.42 External bus timing for page read cycle with bus clock synchronized



Note 1. Always specify WDON and WDOFF as at least one EBCLK cycle.

Figure 62.43 External bus timing for page write cycle with bus clock synchronized

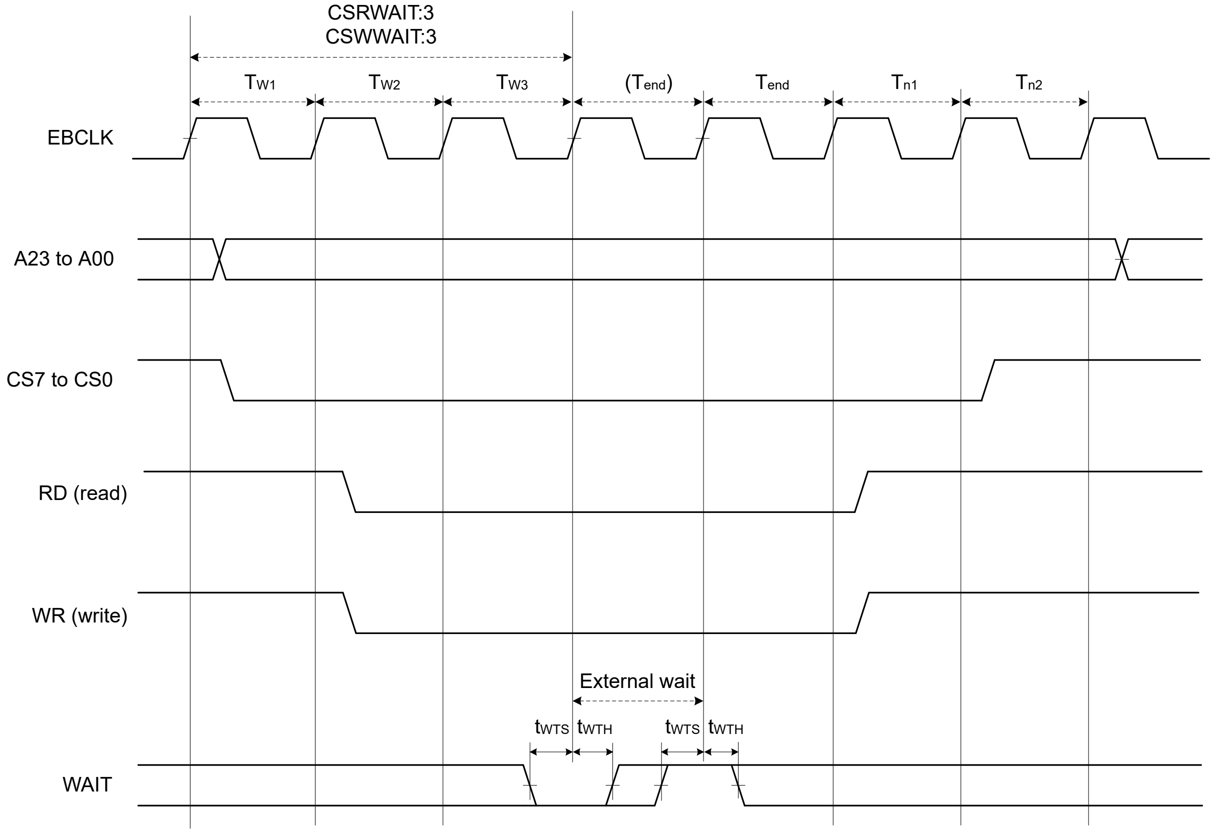


Figure 62.44 External bus timing for external wait control

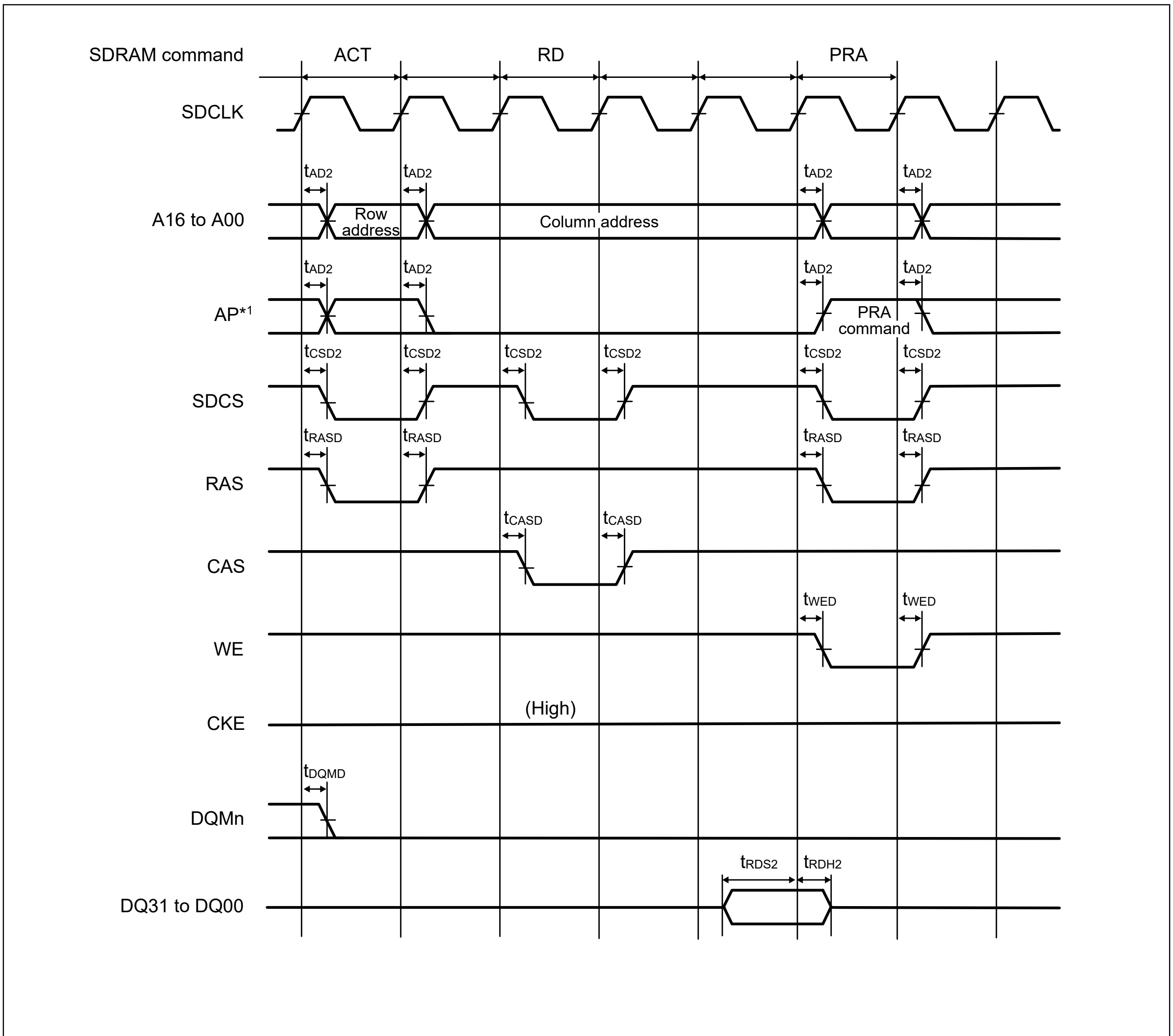


Figure 62.45 SDRAM single read timing

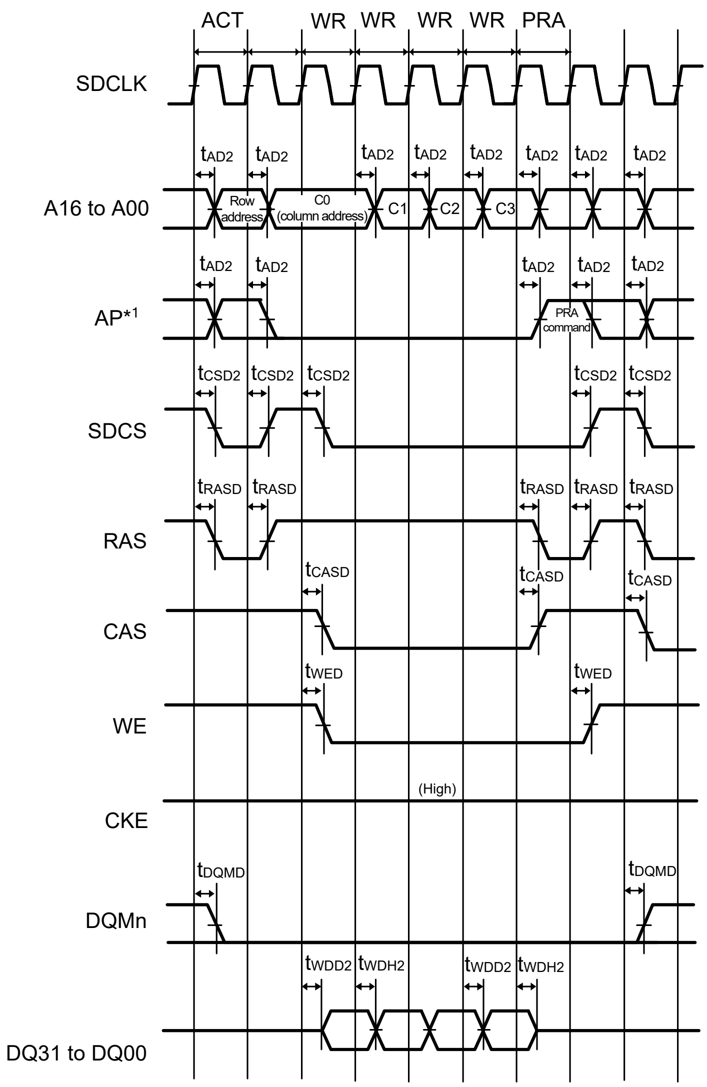


Figure 62.46 SDRAM single write timing

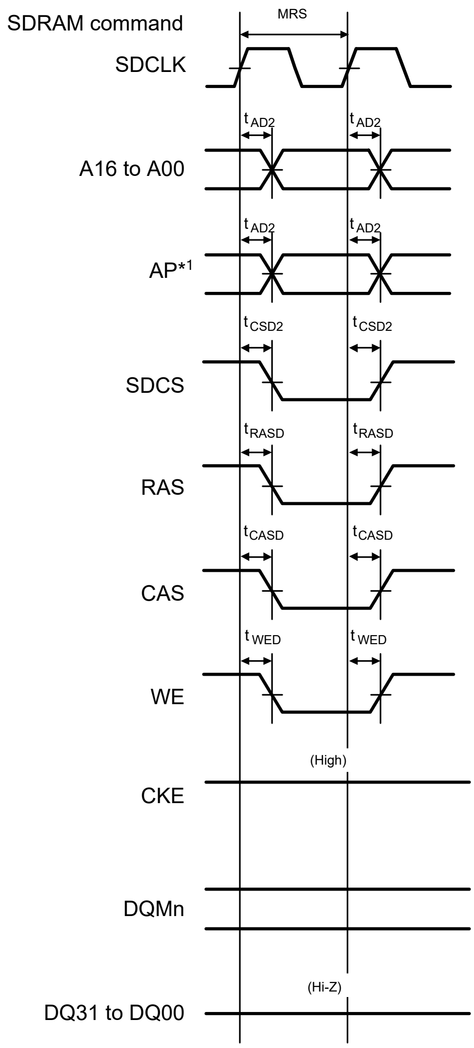


Figure 62.47 SDRAM multiple read timing

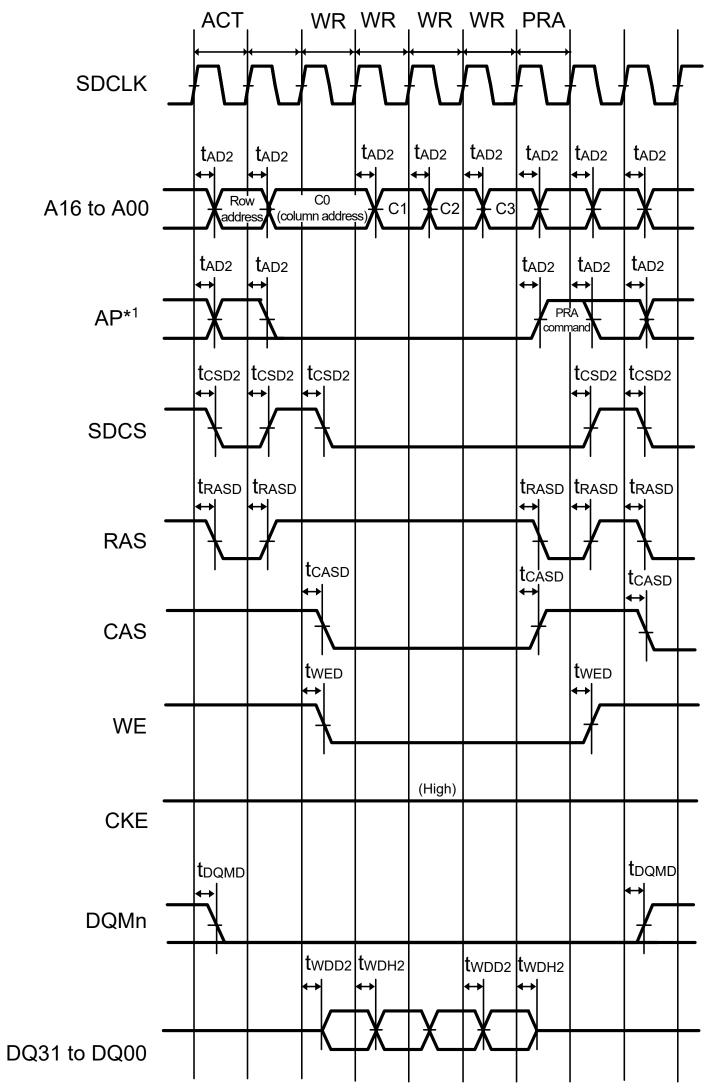


Figure 62.48 SDRAM multiple write timing

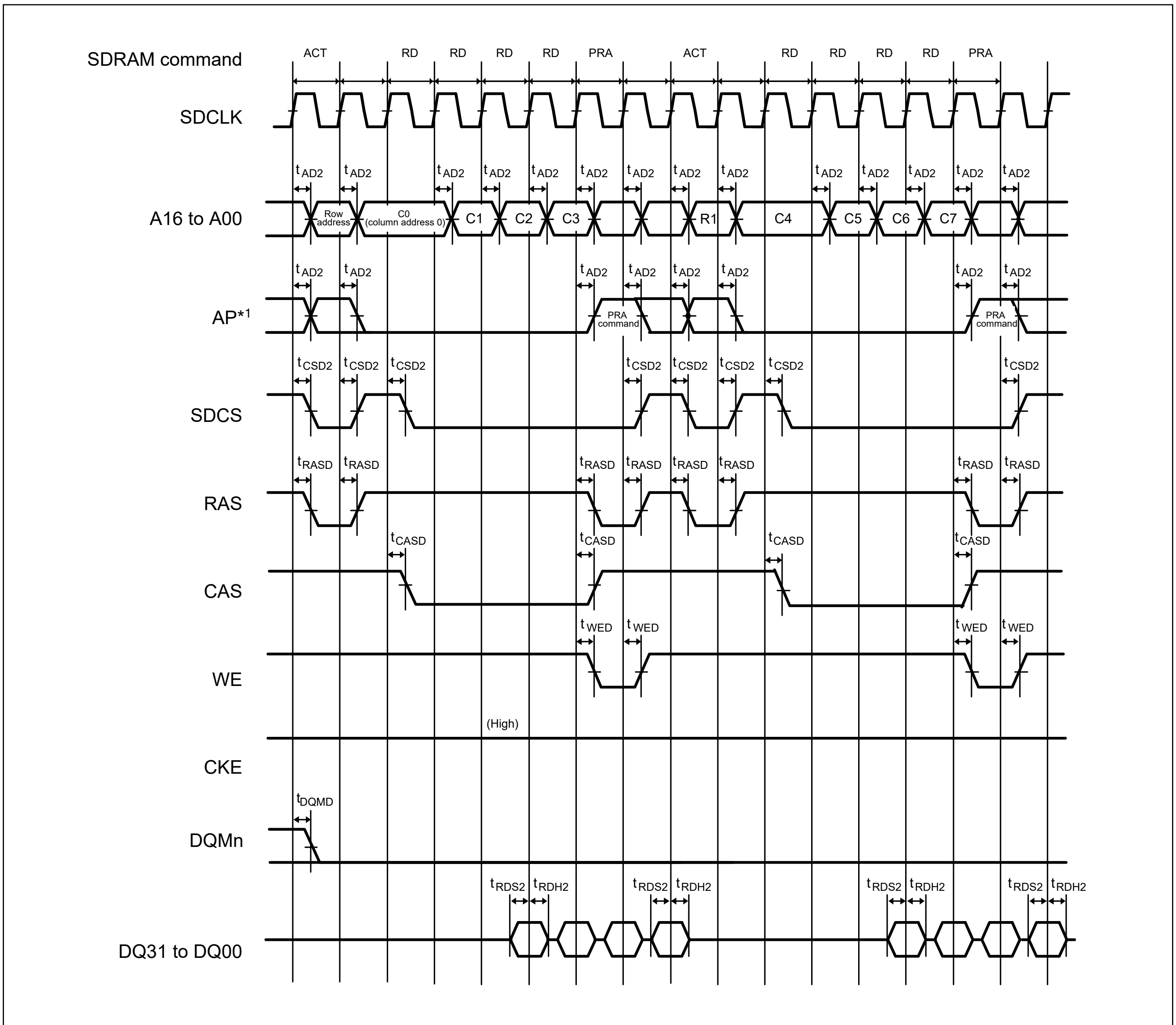


Figure 62.49 SDRAM multiple read line stride timing

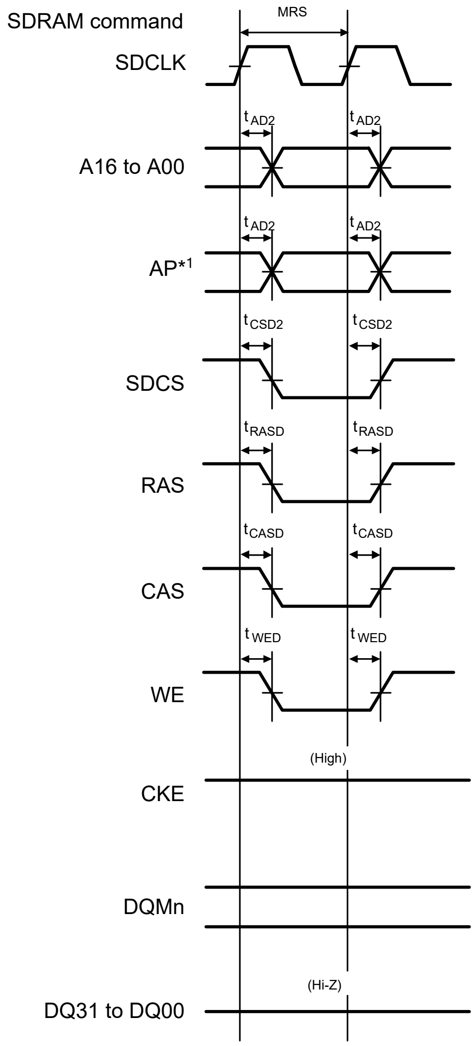


Figure 62.50 SDRAM mode register set timing

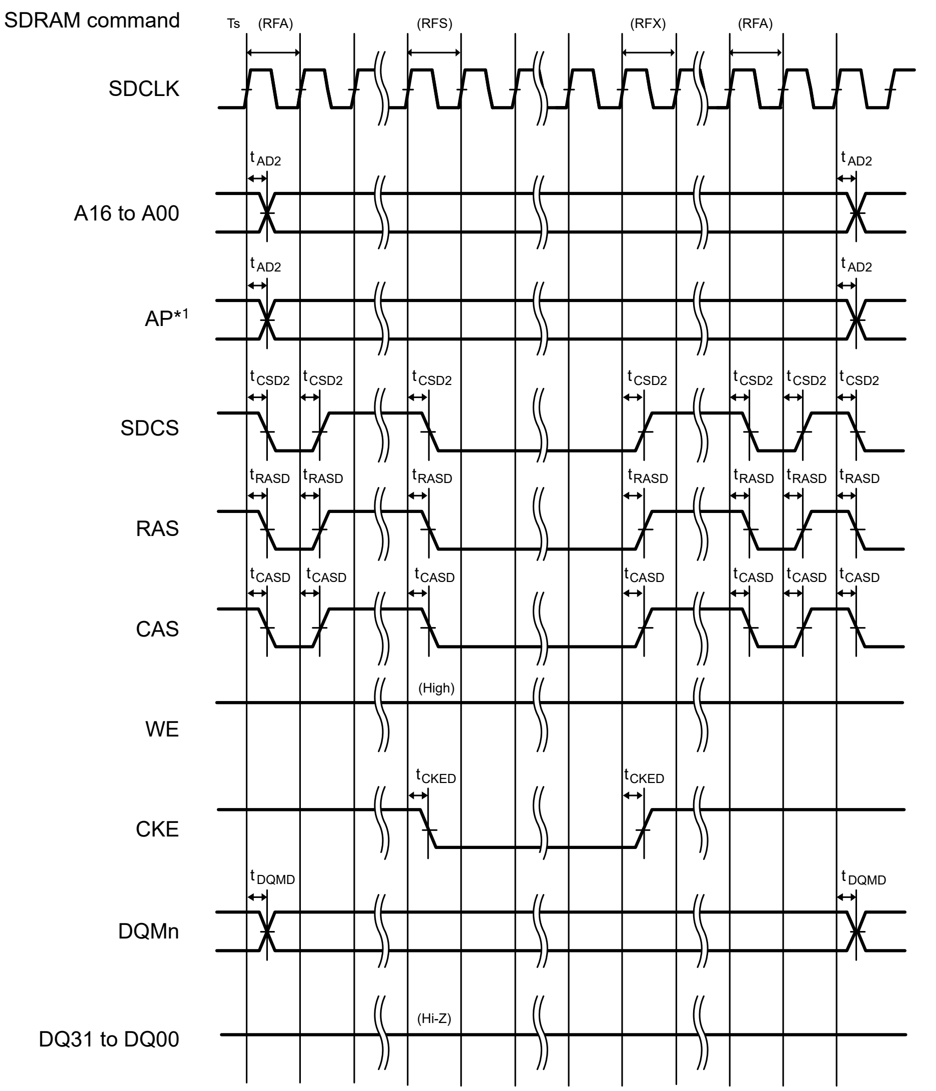


Figure 62.51 SDRAM self-refresh timing

62.3.7 I/O Ports, POEG, GPT, AGT, ULPT and ADC Trigger Timing

Table 62.57 I/O ports, POEG, GPT, AGT, ULPT and ADC trigger timing (1 of 4)

GPT32 Conditions:
Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If GPT pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.
AGT Conditions:
Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter		Symbol	Min	Max	Unit	Test conditions
I/O ports	Input data pulse width	t_{PRW}	5.5	—	t_{Icyc}	Figure 62.52
	EXCIN input frequency	t_{EXCIN}	—	36	kHz	
	RTCICn (n = 0 to 2) input pulse width	t_{RTCICW}	13.89	—	μs	Figure 62.53

Table 62.57 I/O ports, POEG, GPT, AGT, ULPT and ADC trigger timing (2 of 4)

GPT32 Conditions:
Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If GPT pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.
AGT Conditions:
Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter			Symbol	Min	Max	Unit	Test conditions
POEG	POEG input trigger pulse width		t_{POEW}	3	—	t_{Pcyc}	Figure 62.54
	Output disable time	Input level detection of the GTETRGn pin (via flag)	t_{POEGDI}	—	2 PCLKB + 0.34	μs	Figure 62.55 When the digital noise filter is not in use (POEGGn.NFEN = 0 (n = A to D))
		Detection of the output stopping signal from GPT (deadtime error, simultaneous high output, or simultaneous low output)	t_{POEGDE}	—	0.5	μs	Figure 62.56
		Edge detection signal from a comparator	t_{POEGDC}	—	3 PCLKB + 0.5	μs	Figure 62.57 The time is that when the noise filter for ACMPHS is not in use (CMPCTL.CDFS [1:0] = 00b) and excludes the time for detection by ACMPHS.
		Register setting	t_{POEGDS}	—	0.3	μs	Figure 62.58 Time for access to the register is not included.
		Oscillation stop detection	$t_{POEGDOS}$	—	1.3	μs	Figure 62.59
		Level detection signal from a comparator	$t_{POEGDDC}$	—	0.5	μs	Figure 62.60 The time is that when the noise filter for ACMPHS is not in use (CMPCTL.CDFS [1:0] = 00b) and excludes the time for detection by ACMPHS.

Table 62.57 I/O ports, POEG, GPT, AGT, ULPT and ADC trigger timing (3 of 4)

GPT32 Conditions:
Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If GPT pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.
AGT Conditions:
Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter				Symbol	Min	Max	Unit	Test conditions
GPT	Input capture pulse width (Cycle)		Single edge	t_{GTICW} *1	1.5	—	t_{PDcyc}	Figure 62.61
			Dual edge		2.5	—		
	Input capture pulse width (Times)		2.70 V or above	t_{GTICW} *1	8.3	—	ns	
			1.62 V or above		10.0	—		
	GTIOCxY output skew (x = 0 to 3, Y = A or B)	Middle drive output	2.70 V or above	t_{GTISK}	—	4	ns	Figure 62.62
			1.62 V or above		—	6		
		High drive output	2.70 V or above			3.5		
			1.62 V or above			4.5		
	GTIOCxY output skew (x = 4 to 13, Y = A or B)	Middle drive output	2.70 V or above		—	4		
			1.62 V or above		—	6		
		High drive output	2.70 V or above		—	3.5		
			1.62 V or above		—	4.5		
	GTIOCxY output skew (x = 0 to 13, Y = A or B)	Middle drive output	2.70 V or above		—	6		
			1.62 V or above		—	7		
		High drive output	2.70 V or above		—	3.5		
			1.62 V or above		—	5		
	OPS output skew GTOUUP, GTOULO, GTOVUP, GTOVLO, GTOWUP, GTOWLO	Middle drive output	2.70 V or above	t_{GTOSK}	—	5	ns	Figure 62.63
			1.62 V or above		—	6		
GPT (PWM Delay Generation Circuit)	GTIOCxY output skew (x = 0 to 3, Y = A or B)	Middle drive output	2.70 V or above	t_{HRSK}	—	4	ns	Figure 62.64
			1.62 V or above		—	6		
		High drive output	2.70 V or above		—	3.5		
			1.62 V or above		—	5		

Table 62.57 I/O ports, POEG, GPT, AGT, ULPT and ADC trigger timing (4 of 4)

GPT32 Conditions:
Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If GPT pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.
AGT Conditions:
Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter			Symbol	Min	Max	Unit	Test conditions
AGT	AGTIO, AGTEE input cycle	2.70 V or above	t_{ACYC}*2	100	—	ns	Figure 62.65
		1.62 V or above		100	—		
	AGTIO, AGTEE input high width, low width	2.70 V or above	t_{ACKWH}, t_{ACKWL}	40	—	ns	
		1.62 V or above		40	—		
	AGTIO, AGTO, AGTOA, AGTOB output cycle	2.70 V or above	t_{ACYC2}	62.5	—	ns	
		1.62 V or above		62.5	—		
ULPT	ULPTEE, ULPTEVI input cycle	2.70 V or above	t_{ULCYC}*3	32	—	μs	Figure 62.66
		1.62 V or above		32	—		
	ULPTEE, ULPTVI input high width, low width	2.70 V or above	t_{ULCKWH}, t_{ULCKWL}	12	—	μs	
		1.62 V or above		12	—		
	ULPTO, ULPTOA, ULPTOB output cycle	2.70 V or above	t_{ULCYC2}	64	—	μs	
		1.62 V or above		64	—		
ADC	ADC trigger input pulse width	2.70 V or above	t_{TRGW}	1.5	—	t_{ADcyc}	Figure 62.67
		1.62 V or above		3.0	—		

Note: t_{Icyc}: ICLK cycle, t_{Pcyc}: PCLKB cycle, t_{PDcyc}: GTCLK cycle, $t_{ULPTLCLK}$: ULPTLCLK cycle, t_{ADcyc}: ADCLK cycle.
Note 1. For Cycle and Time, the longer time characteristics are applied.
Note 2. Constraints on input cycle:
When not switching the source clock: $t_{Pcyc} \times 2 < t_{ACYC}$ should be satisfied.
When switching the source clock: $t_{Pcyc} \times 6 < t_{ACYC}$ should be satisfied.
Note 3. Constraints on input cycle:
ULPTEVI: $t_{Pcyc} \times 2 < t_{ULCYC}$ should be satisfied.
ULPTEE: $t_{ULPTLCLK} \times 2 < t_{ULCYC}$ should be satisfied.

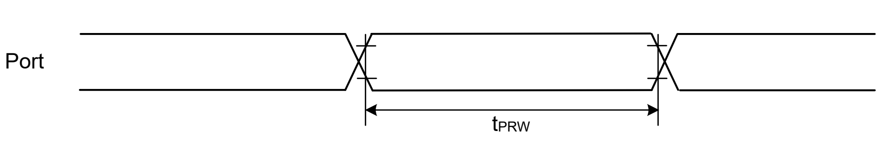


Figure 62.52 I/O ports input timing

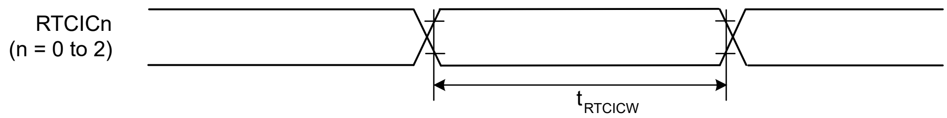


Figure 62.53 RTCICn input timing

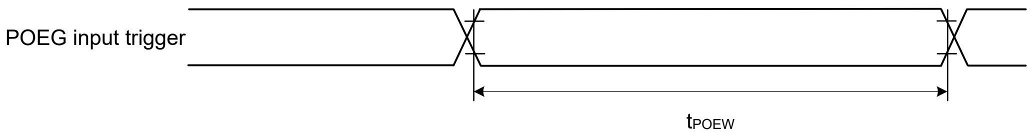


Figure 62.54 POEG input trigger timing

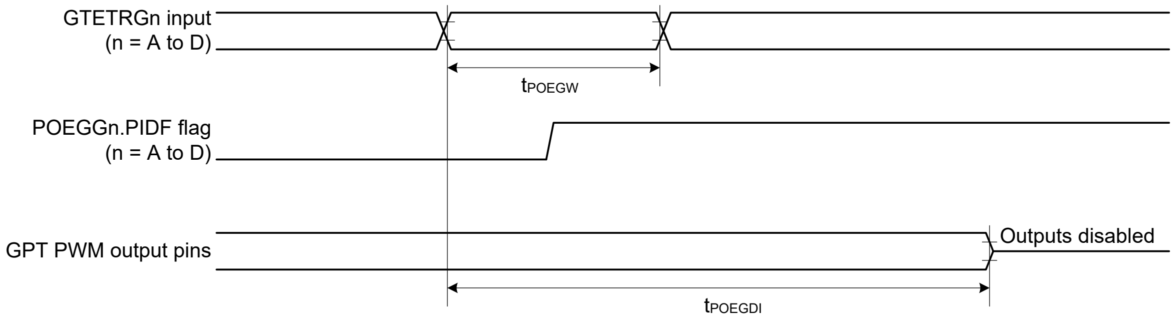
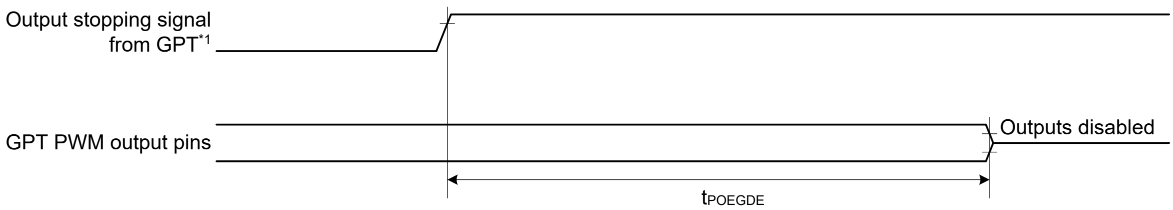


Figure 62.55 Output Disable Time for POEG via Detection Flag in Response to the Input Level Detection of the GTETRGn pin



Note 1. GPT32n.GTST.DTEF (dead time error flag), GPT32n.GTST.OABLF (simultaneous low output flag), or GPT32n.GTST.OABHF (simultaneous high output flag)

Figure 62.56 Output Disable Time for POEG in Response to Detection of the Output Stopping Signal from GPT

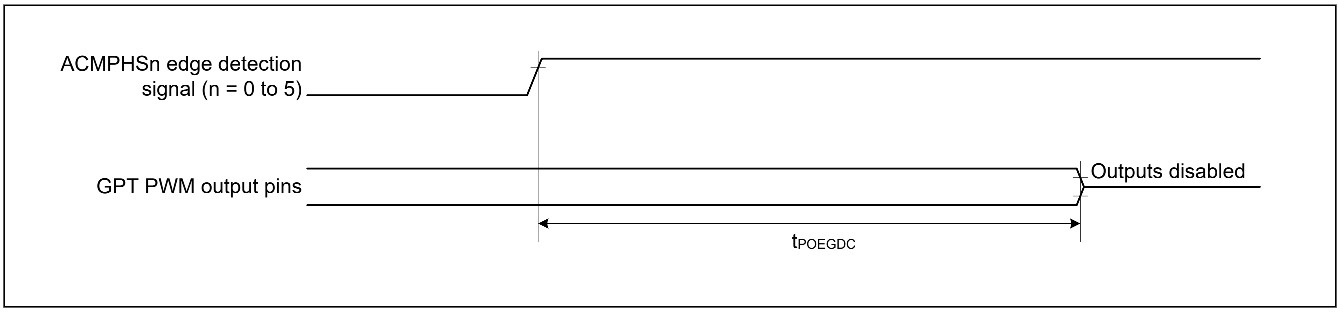


Figure 62.57 Output Disable Time for POEG in Response to Edge Detection Signal from a Comparator

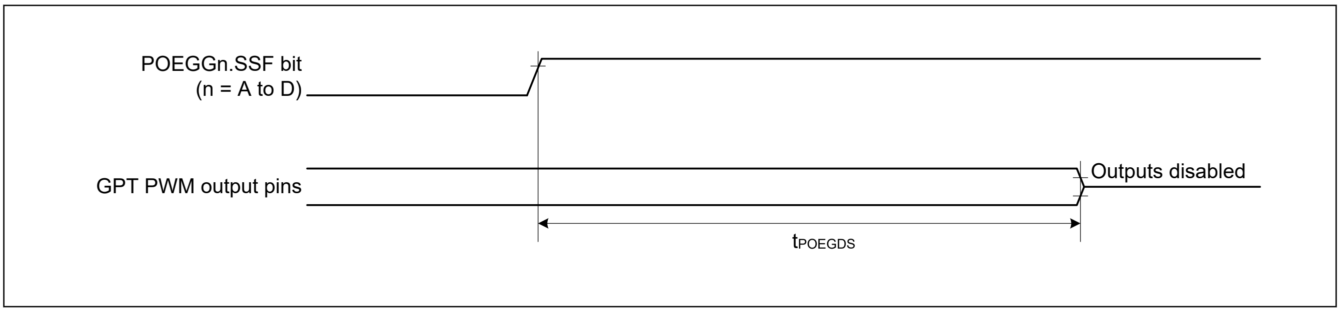


Figure 62.58 Output Disable Time for POEG in Response to the Register Setting

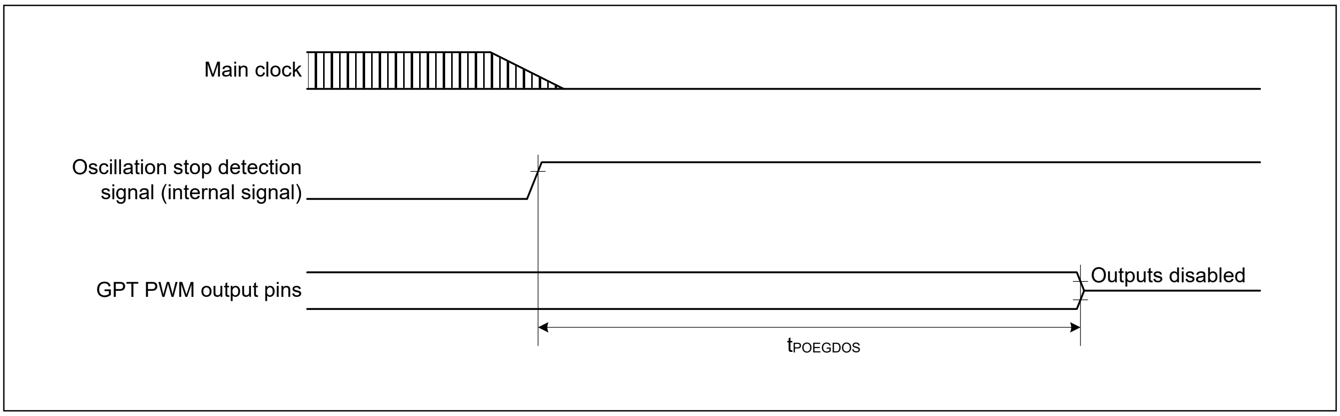


Figure 62.59 Output Disable Time of POEG in Response to the Oscillation Stop Detection

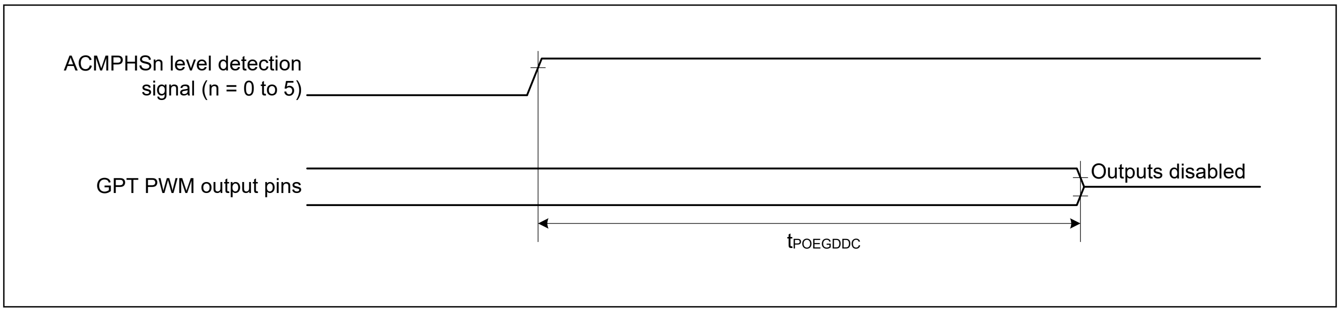


Figure 62.60 Output Disable Time for POEG in Response to Level Detection Signal from a Comparator

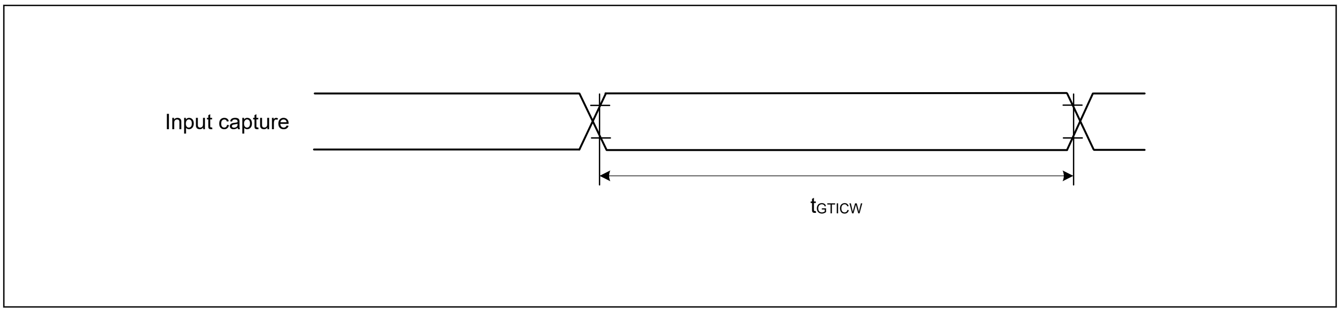


Figure 62.61 GPT input capture timing

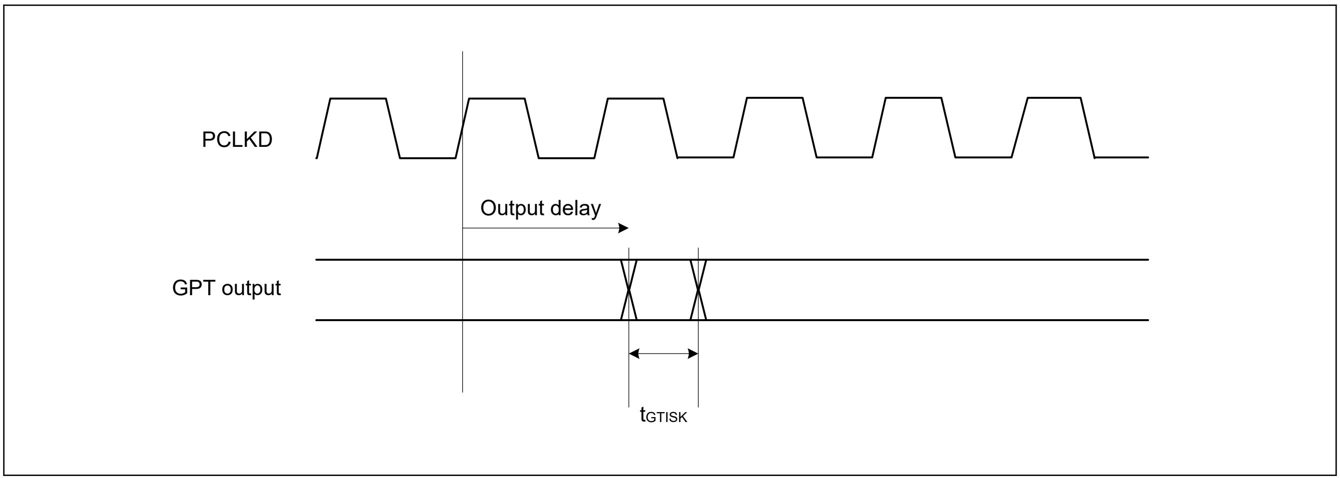


Figure 62.62 GPT output delay skew

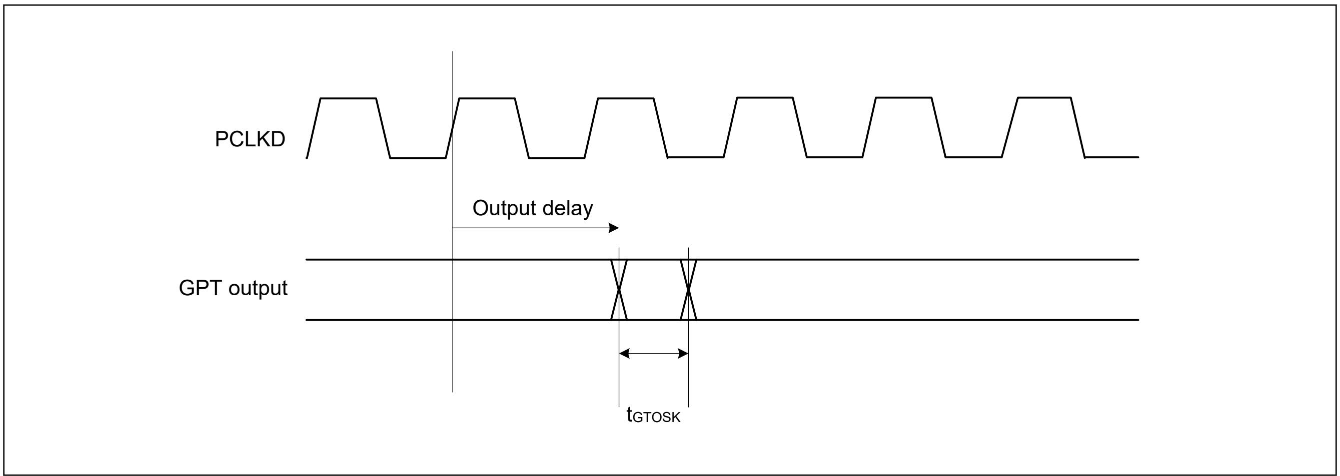


Figure 62.63 GPT output delay skew for OPS

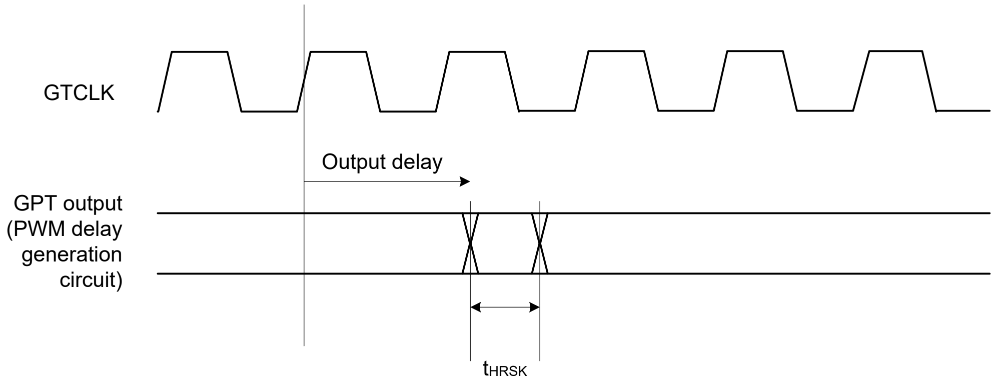


Figure 62.64 GPT (PDG) output delay skew

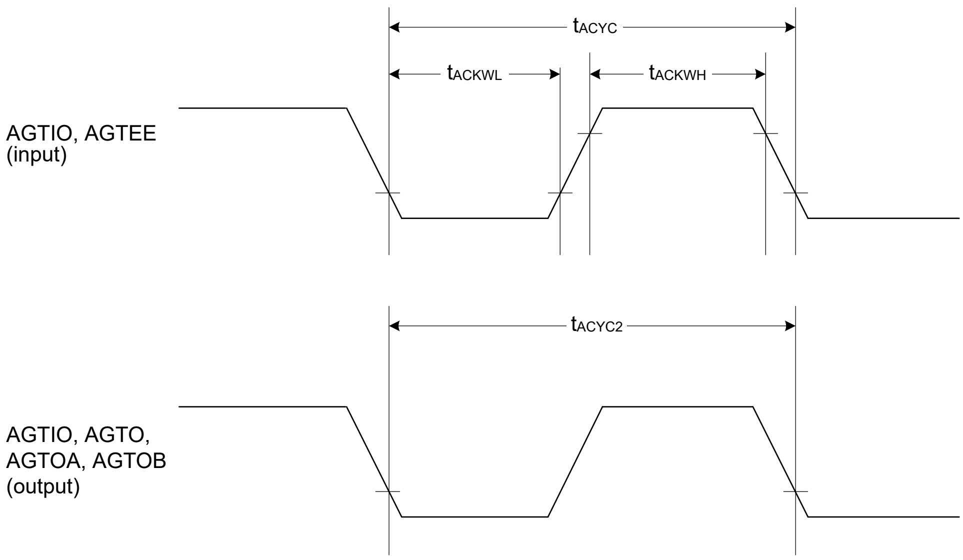


Figure 62.65 AGT input/output timing

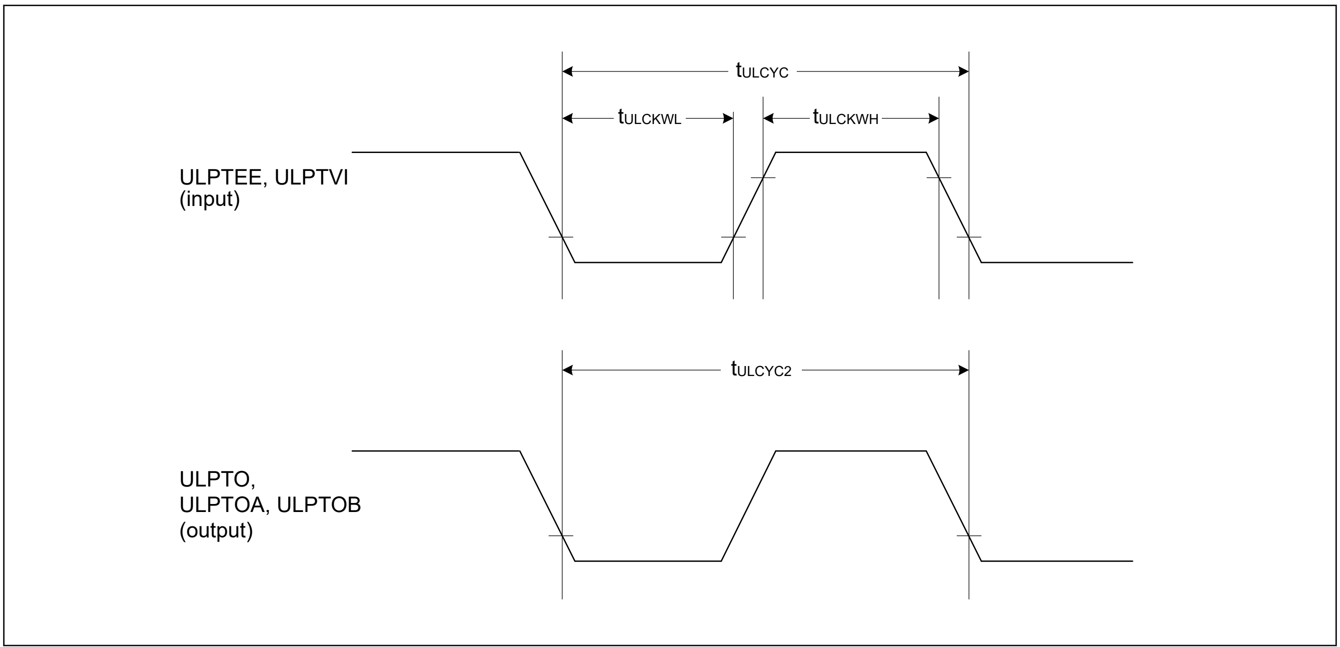


Figure 62.66 ULPT input/output timing

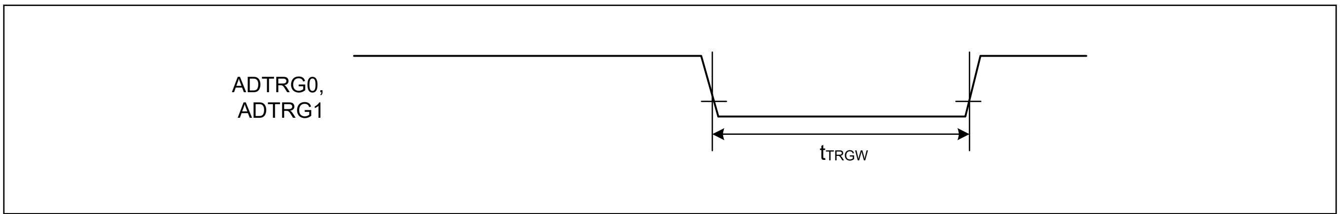


Figure 62.67 ADC trigger input timing

62.3.8 CAC Timing

Table 62.58 CAC timing

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
CAC	CACREF input pulse width	$t_{PBcyc} \leq t_{cac}$ *1	t_{CACREF}	$4.5 \times t_{cac} + 3 \times t_{PBcyc}$	—	—	ns	—
		$t_{PBcyc} > t_{cac}$ *1		$5 \times t_{cac} + 6.5 \times t_{PBcyc}$	—	—	ns	

Note: t_{PBcyc}: PCLKB cycle.
Note 1. t_{cac}: CAC count clock source cycle.

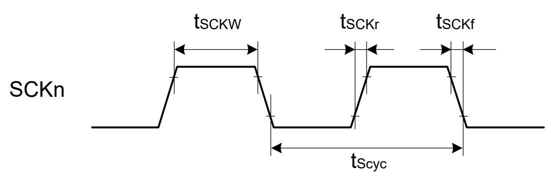
62.3.9 SCI Timing

Table 62.59 SCI timing (Asynchronous mode)

Conditions:
High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.

Parameter	VCC/VCC2	Symbol	Min	Max	Unit	Note
Input clock cycle	1.62 V or above	t_{Scyc}	4.0	—	t_{Tcyc}	Figure 62.68
Input clock pulse width	1.62 V or above	t_{SCKW}	0.4	0.6	t_{Scyc}	
Input clock rise time	1.62 V or above	t_{SCKr}	—	0.1[*1]	t_{Scyc}	
Input clock fall time	1.62 V or above	t_{SCKf}	—	0.1[*1]	t_{Scyc}	
Output clock cycle	1.62 V or above	t_{Scyc}	6.0	—	t_{Tcyc}	
Output clock pulse width	1.62 V or above	t_{SCKW}	0.4	0.6	t_{Scyc}	
Output clock rise time	2.70 V or above	t_{SCKr}	—	3.3	ns	
	1.62 V or above		—	6.6		
Output clock fall time	2.70 V or above	t_{SCKf}	—	3.3	ns	
	1.62 V or above		—	6.6		

Note: tT_{cyc}: TCLK cycle.
Note 1. 1 µs at the longest



Note: n = 0 to 4, 9

Figure 62.68 SCK clock input/output timing

Table 62.60 SCI timing (Simple SPI) (1 of 3)

Condition 1: VCC/VCC2 = 2.70 V or above
High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.
Condition 2: VCC/VCC2 = 1.62 V or above
Following pins have high-speed high-drive output selected in the Port Drive Capability bit in the PmnPFS register: SCK1_A, SCK1_C, SCK3_A, SCK4_A, SCK4_B, SCK4_C and SCK6_B
Other pins have high drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC=VCC2.

Parameter		High Speed/ Default	VCC/VCC2	Symbol	Min	Max	Unit	Note
SCK clock cycle output	Master	—	2.70 V or above	t_{SPcyc}	2 (TCLK ≤ 120MHz) 4 (TCLK > 120MHz)	65536	t_{Tcyc}	Figure 62.69
			1.62 V or above		2 (TCLK ≤ 60MHz) 4 (TCLK ≤ 120MHz) 8 (TCLK > 120MHz)	65536		
SCK clock cycle input	Slave	—	2.70 V or above		2	—		
			1.62 V or above		2 (TCLK ≤ 100MHz) 4 (TCLK > 100MHz)	—		
SCK clock high pulse width	Master	—	1.62 V or above	t_{SPCKWH}	0.4	—	t_{SPcyc}	
	Slave	—						
SCK clock low pulse width	Master	—	1.62 V or above	t_{SPCKWL}	0.4	—	t_{SPcyc}	
	Slave	—						
SCK clock rise and fall time	Output	—	2.70 V or above	t_{SPCKr}, t_{SPCKf}	—	3.3	ns	
			1.62 V or above		—	6.6		
	Input	—	2.70 V or above		—	0.1*3	t_{SPcyc}	
			1.62 V or above		—	0.1*3		

Table 62.60 SCI timing (Simple SPI) (2 of 3)

Condition 1: VCC/VCC2 = 2.70 V or above
High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.
Condition 2: VCC/VCC2 = 1.62 V or above
Following pins have high-speed high-drive output selected in the Port Drive Capability bit in the PmnPFS register: SCK1_A, SCK1_C, SCK3_A, SCK4_A, SCK4_B, SCK4_C and SCK6_B
Other pins have high drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC=VCC2.

Parameter		High Speed/ Default	VCC/VCC2	Symbol	Min	Max	Unit	Note
Data input setup time	Master	High Speed*1	2.70 V or above	t_{SU}	-1.5	—	ns	Figure 62.70, Figure 62.71
			1.62 V or above		-1.5	—		
		Default*2	2.70 V or above		2.0	—		
			1.62 V or above		2.0	—		
	Slave	Default*2	2.70 V or above		2.5	—		
			1.62 V or above		4.5	—		
Data input hold time	Master	High Speed*1	2.70 V or above	t_H	7.5	—	ns	
			1.62 V or above		9.5	—		
		Default*2	2.70 V or above		7.5	—		
			1.62 V or above		9.5	—		
	Slave	Default*2	2.70 V or above		2.5	—		
			1.62 V or above		4.5	—		
Data output delay	Master	High Speed*1	2.70 V or above	t_{OD}	—	3.0	ns	
			1.62 V or above		—	4.5		
		Default*2	2.70 V or above		—	3.5		
			1.62V or above		—	5.5		
	Slave	High Speed*1	2.70 V or above		—	12.5		
			1.62 V or above		—	20.5		
		Default*2	2.70 V or above		—	18.5		
			1.62 V or above		—	26.5		
Data output hold time	Master	High Speed*1	2.70 V or above	t_{OH}	−3.0	—	ns	
			1.62 V or above		−4.5	—		
		Default*2	2.70 V or above		−3.5	—		
			1.62 V or above		−5.5	—		
	Slave	Default*2	2.70 V or above		0.0	—		
			1.62 V or above		0.0	—		
Data rise and fall time	Output	—	2.70 V or above	t_{Dr}, t_{Df}	—	3.3	ns	
			1.62 V or above		—	6.6		
	Input	—	2.70 V or above		—	1		
			1.62 V or above		—	1		
SS input setup time		—	1.62 V or above	t_{LEAD}	1.0	—	t_{SPcyc}	Figure 62.72, Figure 62.73
SS input hold time		—	1.62 V or above	t_{LAG}	1.0	—	t_{SPcyc}	
SS input rise and fall time		—	1.62 V or above	t_{SSLr}, t_{SSLf}	—	1	µs	—

Table 62.60 SCI timing (Simple SPI) (3 of 3)

Condition 1: VCC/VCC2 = 2.70 V or above
High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.
Condition 2: VCC/VCC2 = 1.62 V or above
Following pins have high-speed high-drive output selected in the Port Drive Capability bit in the PmnPFS register: SCK1_A, SCK1_C, SCK3_A, SCK4_A, SCK4_B, SCK4_C and SCK6_B
Other pins have high drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC=VCC2.

Parameter	High Speed/ Default	VCC/VCC2	Symbol	Min	Max	Unit	Note
Slave access time	—	2.70 V or above	t_{SA}	—	3 × t_{Tcyc} + 25	ns	Figure 62.72, Figure 62.73
		1.62 V or above		—	3 × t_{Tcyc} + 32		
Slave output release time	—	2.70 V or above	t_{REL}	—	3 × t_{Tcyc} + 25	ns	
		1.62 V or above		—	3 × t_{Tcyc} + 32		

Note: t_{Tcyc}: TCLK cycle.
Note 1. Must use pins that have a letter appended to their name, for instance _A, _B, _C, to indicate group membership. SCI0, SCI1, SCI2, SCI3 and SCI9 are instance _A, SCI4 and SCI5 are instance _B, SCI6, SCI7 and SCI8 are instance _C.
Note 2. All pins of group membership can be used.
Note 3. 1 µs at the longest

Table 62.61 SCI timing (Clock synchronous mode) (1 of 2)

Condition 1: VCC/VCC2 = 2.70 V or above
High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.
Condition 2: VCC/VCC2 = 1.62 V or above
Following pins have high-speed high-drive output selected in the Port Drive Capability bit in the PmnPFS register: SCK1_A, SCK1_C, SCK3_A, SCK4_A, SCK4_B, SCK4_C and SCK6_B
Other pins have high drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC=VCC2.

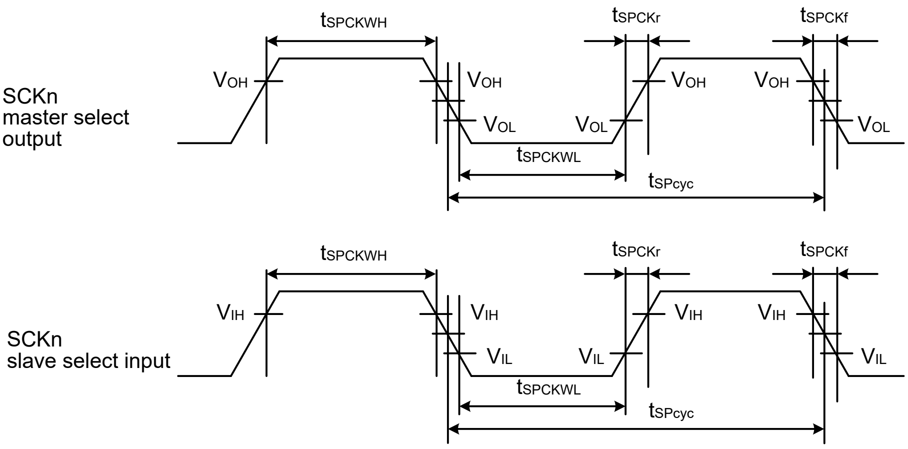
Parameter		High Speed/ Default	VCC/VCC2	Symbol	Min	Max	Unit	Note
SCK clock cycle output	Master	—	2.70 V or above	t_{Scyc}	2 (TCLK ≤ 120MHz) 4 (TCLK > 120MHz)	—	t_{Tcyc}	
			1.62 V or above		2 (TCLK ≤ 60MHz) 4 (TCLK ≤ 120MHz) 8 (TCLK > 120MHz)	—		
SCK clock cycle input	Slave	—	2.70 V or above		2	—		
			1.62 V or above		2 (TCLK ≤ 100MHz) 4 (TCLK > 100MHz)	—		
SCK clock high pulse width	Master	—	1.62 V or above	t_{SCKWH}	0.4	0.6	t_{Scyc}	
	Slave		1.62 V or above					
SCK clock low pulse width	Master	—	1.62 V or above	t_{SCKWL}	0.4	0.6	t_{Scyc}	
	Slave		1.62 V or above					

Table 62.61 SCI timing (Clock synchronous mode) (2 of 2)

Condition 1: VCC/VCC2 = 2.70 V or above
High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.
Condition 2: VCC/VCC2 = 1.62 V or above
Following pins have high-speed high-drive output selected in the Port Drive Capability bit in the PmnPFS register: SCK1_A, SCK1_C, SCK3_A, SCK4_A, SCK4_B, SCK4_C and SCK6_B
Other pins have high drive output is selected in the Port Drive Capability bit in the PmnPFS register.
If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC=VCC2.

Parameter		High Speed/ Default	VCC/VCC2	Symbol	Min	Max	Unit	Note
SCK clock rise and fall time	Output	—	2.70 V or above	t_{SCKr}, t_{SCKf}	—	3.3	ns	
			1.62 V or above		—	6.6		
	Input	—	1.62 V or above		—	0.1*3	t_{Scyc}	
Data input setup time	Master	High Speed*1	2.70 V or above	t_{SU}	2.6	—	ns	
			1.62 V or above		2.6	—		
		Default*2	2.70 V or above		2.8	—		
			1.62 V or above		2.8	—		
	Slave	Default*2	2.70 V or above		3.3	—		
			1.62 V or above		5.3	—		
Data input hold time	Master	High Speed*1	2.70 V or above	t_H	7.5	—	ns	
			1.62 V or above		9.5	—		
		Default*2	2.70 V or above		7.5	—		
			1.62 V or above		9.5	—		
	Slave	Default*2	2.70 V or above		3.0	—		
			1.62 V or above		5.0	—		
Data output delay	Master	High Speed*1	2.70 V or above	t_{OD}	—	5	ns	
			1.62 V or above		—	5		
		Default*2	2.70 V or above		—	7.3		
			1.62 V or above		—	7.3		
	Slave	High Speed*1	2.70 V or above		—	12.5		
			1.62 V or above		—	20.5		
		Default*2	2.70 V or above		—	18.5		
			1.62 V or above		—	26.5		
Data rise and fall time	Output	—	2.70 V or above	t_{Dr}, t_{Df}	—	3.3	ns	
			1.62 V or above		—	6.6		
	Input	—	1.62 V or above		—	1	μs	

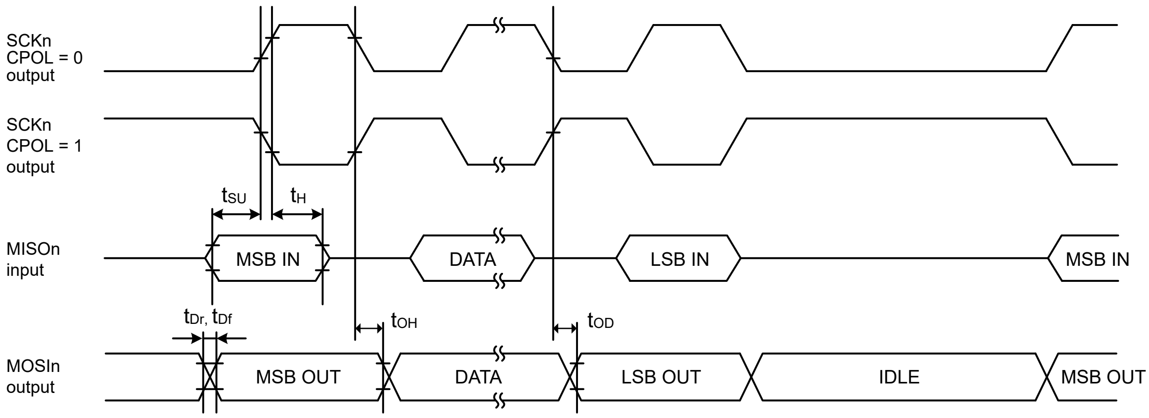
Note: t_{Tcyc}: TCLK cycle.
Note 1. Must use pins that have a letter appended to their name, for instance _A, _B, _C, to indicate group membership. SCI0, SCI1, SCI2, SCI3 and SCI9 are instance _A, SCI4 and SCI5 are instance _B, SCI6, SCI7 and SCI8 are instance _C.
Note 2. All pins of group membership can be used.
Note 3. 1 μs at the longest



V_{OH} = 0.7 × VCC or 0.7 × VCC2, V_{OL} = 0.3 × VCC or 0.3 × VCC2,
V_{IH} = 0.7 × VCC or 0.7 × VCC2, V_{IL} = 0.3 × VCC or 0.3 × VCC2

Note: n = 0 to 4, 9

Figure 62.69 SCI simple SPI mode clock timing



Note: n = 0 to 4, 9

Figure 62.70 SCI simple SPI mode timing for master when CPHA = 0

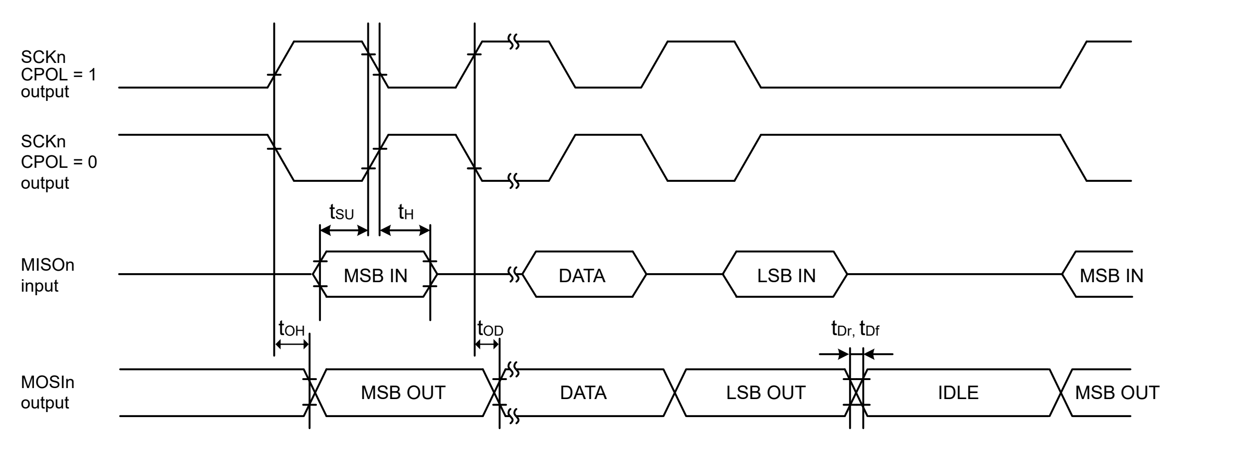
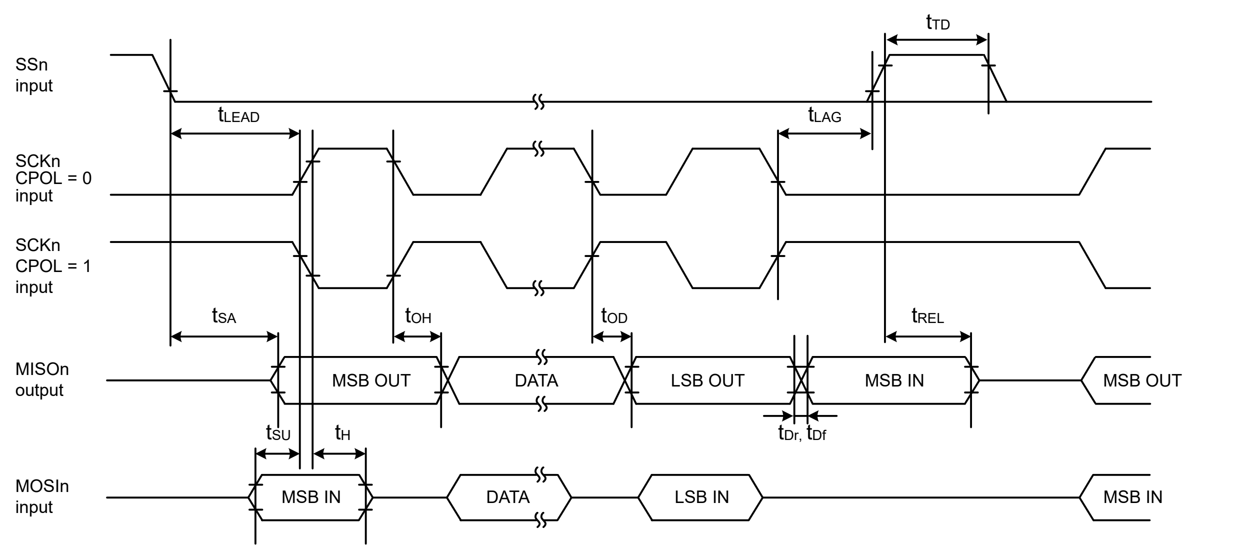
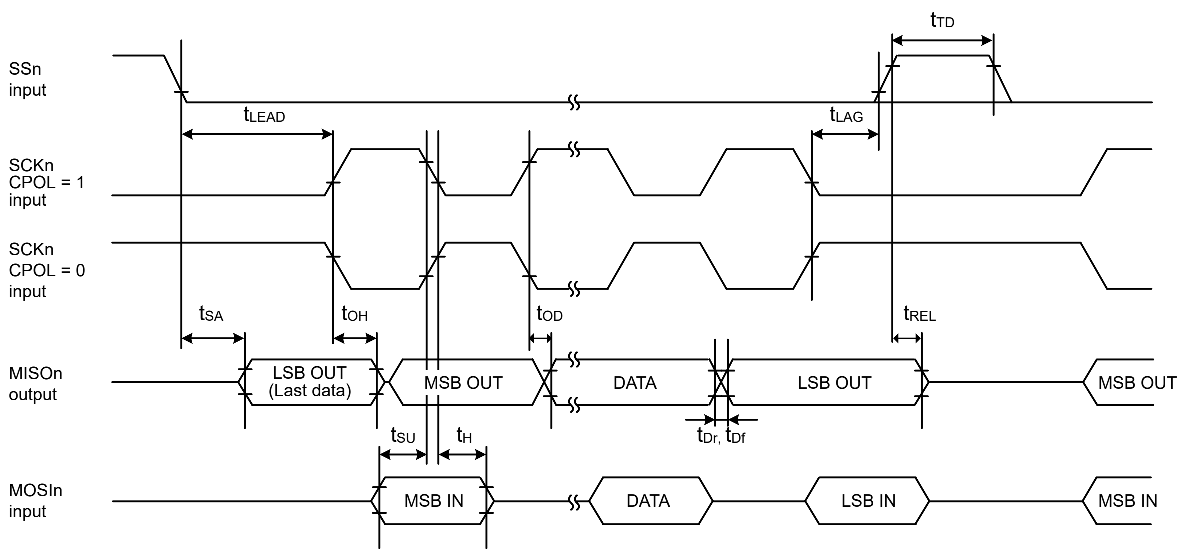


Figure 62.71 SCI simple SPI mode timing for master when CPHA = 1



Note: n = 0 to 4, 9

Figure 62.72 SCI simple SPI mode timing for slave when CPHA = 0



Note: n = 0 to 4, 9

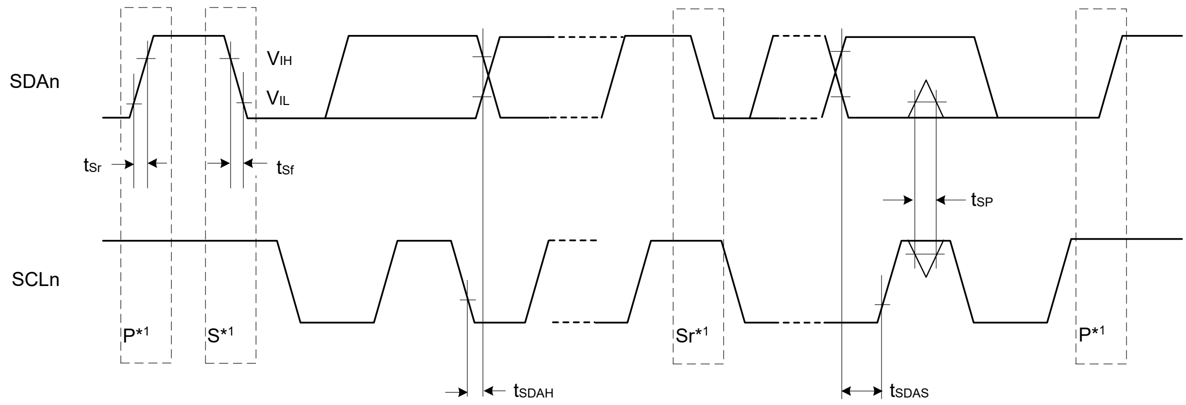
Figure 62.73 SCI simple SPI mode timing for slave when CPHA = 1

Table 62.62 SCI timing (Simple IIC mode)

Conditions:
Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.
VCC/VCC2 : 1.62 V or above
If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.

Parameter		Symbol	Min	Max	Unit	Note
Simple IIC (Standard mode)	SCL, SDA input rise time	t_{Sr}	—	1000	ns	
	SCL, SDA input fall time	t_{Sf}	—	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	4 × t_{Tcyc}	ns	
	Data input setup time	t_{SDAS}	250	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b*1	—	400	pF	
Simple IIC (Fast mode)	SCL, SDA input rise time	t_{Sr}	—	300	ns	
	SCL, SDA input fall time	t_{Sf}	—	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	4 × t_{Tcyc}	ns	
	Data input setup time	t_{SDAS}	100	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b*1	—	400	pF	

Note: t_{Tcyc}: TCLK cycle.
Note 1. C_b indicates the total capacity of the bus line.



Test conditions:
V_{IH} = VCC × 0.7 or VCC2 × 0.7
V_{IL} = VCC × 0.3 or VCC2 × 0.3
V_{OL} = 0.6 V, I_{OL} = 6 mA

Note: n = 0 to 4, 9
Note 1. S, P, and Sr indicate the following conditions:
S: Start condition
P: Stop condition
Sr: Restart condition

Figure 62.74 SCI simple IIC mode timing

62.3.10 SPI Timing

Table 62.63 SPI timing (1 of 5)

Conditions:

1. High-speed high drive output is selected in the Port Drive Capability bit in the PmnPFS register for the following pins: RSPCLKA_B, RSPCLKB_B.
   For other pins, high drive output is selected in the Port Drive Capability bit in the PmnPFS register.
2. Use pins that have a letter appended to their names, for instance "_A" or "_B" to indicate group membership.
3. Load capacitance C = 15pF is applied to the VCC/VCC2 condition "3.00 V or above".

Parameter		High Speed/ Default	VCC/VCC2	Symbol	Min	Max	Unit	Note
RSPCK clock cycle	Master	High Speed*1	3.00 V or above	t_{SPcyc}	2	4096	t_{Tcyc}	Figure 62.75
		High Speed*1	2.70 V or above		2 (TCLK ≤ 166.6MHz) 4 (TCLK > 166.6 MHz)	4096		
		High Speed*1	1.62 V or above		2 (TCLK ≤ 83.3MHz) 4 (TCLK ≤ 166.6 MHz) 8 (TCLK > 166.6 MHz)	4096		
		Default*2	3.00 V or above		2 (TCLK ≤ 166.6 MHz) 4 (TCLK > 166.6 MHz)	4096		
		Default*2	2.70 V or above		2 (TCLK ≤ 120MHz) 4 (TCLK ≤ 240 MHz) 8 (TCLK > 240 MHz)	4096		
		Default*2	1.62 V or above		2 (TCLK ≤ 60 MHz) 4 (TCLK ≤ 120 MHz) 8 (TCLK ≤ 240 MHz) 16 (TCLK > 240 MHz)	4096		
	Slave	High Speed*1	3.00 V or above		2 (TCLK ≤ 266 MHz) 4 (TCLK > 266 MHz)	—		
		High Speed*1	2.70 V or above		2 (TCLK ≤ 166.6 MHz) 4 (TCLK > 166.6 MHz)	—		
		High Speed*1	1.62 V or above		2 (TCLK ≤ 83.3MHz) 4 (TCLK ≤ 166.6 MHz) 8 (TCLK > 166.6 MHz)	—		
		Default*2	3.00 V or above		2 (TCLK ≤ 166.6 MHz) 4 (TCLK > 166.6 MHz)	—		
		Default*2	2.70 V or above		2 (TCLK ≤ 120 MHz) 4 (TCLK ≤ 240 MHz) 8 (TCLK > 240 MHz)	—		
		Default*2	1.62 V or above		2 (TCLK ≤ 60 MHz) 4 (TCLK ≤ 120 MHz) 8 (TCLK ≤ 240 MHz) 16 (TCLK > 240 MHz)	—		

Table 62.63 SPI timing (2 of 5)

Conditions:

1. High-speed high drive output is selected in the Port Drive Capability bit in the PmnPFS register for the following pins: RSPCLKA_B, RSPCLKB_B.
   For other pins, high drive output is selected in the Port Drive Capability bit in the PmnPFS register.
2. Use pins that have a letter appended to their names, for instance "_A" or "_B" to indicate group membership.
3. Load capacitance C = 15pF is applied to the VCC/VCC2 condition "3.00 V or above".

Parameter		High Speed/ Default	VCC/VCC2	Symbol	Min	Max	Unit	Note
RSPCK clock high pulse width	Master	—	3.00 V or above	t_{SPCKWH}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 1$	—	ns	Figure 62.75
		—	2.70 V or above		$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 2$	—		
		—	1.62 V or above		$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 3$	—		
	Slave	—	1.62 V or above		0.4	—	t_{SPcyc}	
RSPCK clock low pulse width	Master	—	3.00 V or above	t_{SPCKWL}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 1$	—	ns	
		—	2.70 V or above		$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 2$	—		
		—	1.62 V or above		$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 3$	—		
	Slave	—	1.62 V or above		0.4	—	t_{SPcyc}	
RSPCK clock rise and fall time	Output	High Speed*1	3.00 V or above	t_{SPCKr}, t_{SPCKf}	—	0.80	ns	
		High Speed*1	2.70 V or above		—	1.40		
		High Speed*1	1.62 V or above		—	2.50		
		Default*2	3.00 V or above		—	1.66		
		Default*2	2.70 V or above		—	3.30		
		Default*2	1.62 V or above		—	6.60		
	Input	—	3.00 V or above		—	0.1*3	μs	
		—	2.70 V or above		—	0.1*3		
		—	1.62 V or above		—	0.1*3		

Table 62.63 SPI timing (3 of 5)

Conditions:

1. High-speed high drive output is selected in the Port Drive Capability bit in the PmnPFS register for the following pins: RSPCLKA_B, RSPCLKB_B.
   For other pins, high drive output is selected in the Port Drive Capability bit in the PmnPFS register.
2. Use pins that have a letter appended to their names, for instance “_A” or “_B” to indicate group membership.
3. Load capacitance C = 15pF is applied to the VCC/VCC2 condition “3.00 V or above”.

Parameter		High Speed/ Default	VCC/VCC2	Symbol	Min	Max	Unit	Note
Data input setup time	Master	—	3.00 V or above	t_{SU}	−2.5	—	ns	Figure 62.76, Figure 62.77
		—	2.70 V or above		0.0	—		
		—	1.62 V or above		0.0	—		
	Slave	High Speed*1	3.00 V or above		1.5	—		
		High Speed*1	2.70 V or above		1.5	—		
		High Speed*1	1.62 V or above		1.5	—		
		Default*2	3.00 V or above		2.5	—		
		Default*2	2.70 V or above		2.5	—		
		Default*2	1.62 V or above		2.5	—		
Data input hold time	Master	—	3.00 V or above	t_H	7.5	—	ns	
		—	2.70 V or above		7.5	—		
		—	1.62 V or above		9.5	—		
	Slave	High Speed*1	3.00 V or above		1.5	—		
		High Speed*1	2.70 V or above		1.5	—		
		High Speed*1	1.62 V or above		1.5	—		
		Default*2	3.00 V or above		2.5	—		
		Default*2	2.70 V or above		2.5	—		
		Default*2	1.62 V or above		5.5	—		
SSL setup time	Master	—	3.00 V or above	t_{LEAD}	1 × t_{SPcyc} − 10	8 × t_{SPcyc} + 10	ns	Figure 62.76, Figure 62.77
		—	2.70 V or above		1 × t_{SPcyc} − 10	8 × t_{SPcyc} + 10		
		—	1.62 V or above		1 × t_{SPcyc} − 10	8 × t_{SPcyc} + 10		
	Slave	—	3.00 V or above		5.0	—	t_{Tcyc}	
		—	2.70 V or above		5.0	—		
		—	1.62 V or above		5.0	—		
SSL hold time	Master	—	3.00 V or above	t_{LAG}	1 × t_{SPcyc} − 10	8 × t_{SPcyc} + 10	ns	
		—	2.70 V or above		1 × t_{SPcyc} − 10	8 × t_{SPcyc} + 10		
		—	1.62 V or above		1 × t_{SPcyc}–- 10	8 × t_{SPcyc} + 10		
	Slave	—	3.00 V or above		5.0	—	t_{Tcyc}	
		—	2.70 V or above		5.0	—		
		—	1.62 V or above		5.0	—		

Table 62.63 SPI timing (4 of 5)

Conditions:

1. High-speed high drive output is selected in the Port Drive Capability bit in the PmnPFS register for the following pins: RSPCLKA_B, RSPCLKB_B.
   For other pins, high drive output is selected in the Port Drive Capability bit in the PmnPFS register.
2. Use pins that have a letter appended to their names, for instance "_A" or "_B" to indicate group membership.
3. Load capacitance C = 15pF is applied to the VCC/VCC2 condition "3.00 V or above".

Parameter		High Speed/ Default	VCC/VCC2	Symbol	Min	Max	Unit	Note
TI SSP SS input setup time	Slave	—	3.00 V or above	t_{TISS}	2.5	—	ns	Figure 62.81
		—	2.70 V or above		2.5	—		
		—	1.62 V or above		2.5	—		
TI SSP SS input hold time	Slave	—	3.00 V or above	t_{TISH}	2.5	—	ns	
		—	2.70 V or above		2.5	—		
		—	1.62 V or above		5.5	—		
TI SSP next-access time	Slave	—	3.00 V or above	t_{TIND}	$2 \times t_{Tcyc}$ + SLNDL $\times t_{Tcyc}$	—	ns	
		—	2.70 V or above		$2 \times t_{Tcyc}$ + SLNDL $\times t_{Tcyc}$	—		
		—	1.62 V or above		$2 \times t_{Tcyc}$ + SLNDL $\times t_{Tcyc}$	—		
TI SSP master SS output delay	Master	—	3.00 V or above	t_{TISSOD}	—	4.0	ns	Figure 62.78
		—	2.70 V or above		—	8.0		
		—	1.62 V or above		—	8.0		
Data output delay time	Master	—	3.00 V or above	t_{OD1}	—	2.0	ns	Figure 62.76, Figure 62.77
		—	2.70 V or above		—	3.0		
		—	1.62 V or above		—	6.0		
		High Speed*1	3.00 V or above	t_{OD2}	—	1.5		
		High Speed*1	2.70 V or above		—	2.5		
		High Speed*1	1.62 V or above		—	4.5		
		Default*2	3.00 V or above		—	2.5		
		Default*2	2.70 V or above		—	2.5		
		Default*2	1.62 V or above		—	4.5		
	Slave	—	3.00 V or above	t_{OD}	—	10.0		
		—	2.70 V or above		—	13.5		
		—	1.62 V or above		—	21.5		
Data output hold time	Master	High Speed*1	3.00 V or above	t_{OH}	−1.5	—	ns	
		High Speed*1	2.70 V or above		−2.5	—		
		High Speed*1	1.62 V or above		−4.5	—		
		Default*2	3.00 V or above		−2.5	—		
		Default*2	2.70 V or above		−2.5	—		
		Default*2	1.62 V or above		−4.5	—		
	Slave	—	3.00 V or above		0.0	—		
		—	2.70 V or above		0.0	—		
		—	1.62 V or above		0.0	—		

Table 62.63 SPI timing (5 of 5)

Conditions:

1. High-speed high drive output is selected in the Port Drive Capability bit in the PmnPFS register for the following pins: RSPCLKA_B, RSPCLKB_B.
   For other pins, high drive output is selected in the Port Drive Capability bit in the PmnPFS register.
2. Use pins that have a letter appended to their names, for instance "_A" or "_B" to indicate group membership.
3. Load capacitance C = 15pF is applied to the VCC/VCC2 condition "3.00 V or above".

Parameter		High Speed/ Default	VCC/VCC2	Symbol	Min	Max	Unit	Note
Successive transmission delay time	Master	—	3.00 V or above	t_{TD}	$t_{SPcyc} + 2 \times t_{Tcyc}$	$8 \times t_{SPcyc} + 2 \times t_{Tcyc}$	ns	Figure 62.76, Figure 62.77
		—	2.70 V or above		$t_{SPcyc} + 2 \times t_{Tcyc}$	$8 \times t_{SPcyc} + 2 \times t_{Tcyc}$		
		—	1.62 V or above		$t_{SPcyc} + 2 \times t_{Tcyc}$	$8 \times t_{SPcyc} + 2 \times t_{Tcyc}$		
	Slave	—	3.00 V or above		t_{Tcyc}	—	ns	
		—	2.70 V or above		t_{Tcyc}	—		
		—	1.62 V or above		t_{Tcyc}	—		
MOSI and MISO rise and fall time	Output	—	3.00 V or above	t_{Dr}, t_{Df}	—	1.66	ns	
		—	2.70 V or above		—	3.30		
		—	1.62V or above		—	6.60		
	Input	—	3.00 V or above		—	1.0	µs	
		—	2.70 V or above		—	1.0		
		—	1.62 V or above		—	1.0		
SSL rise and fall time	Output	—	3.00- V or above	t_{SSLr}, t_{SSLf}	—	1.66	ns	
		—	2.70 V or above		—	3.30		
		—	1.62 V or above		—	6.60		
	Input	—	3.00 V or above		—	1.0	µs	
		—	2.70 V or above		—	1.0		
		—	1.62 V or above		—	1.0		
Slave access time	Slave	—	3.00 V or above	t_{SA}	—	20.0	ns	Figure 62.79, Figure 62.80
		—	2.70 V or above		—	20.0		
		—	1.62 V or above		—	25.0		
Slave output release time	Slave	—	3.00 V or above	t_{REL}	—	20.0	ns	
		—	2.70 V or above		—	20.0		
		—	1.62 V or above		—	25.0		

Note: t_{Tcyc}: TCLK cycle.
Note 1. Must use pins that have a letter appended to their name, for instance _A, _B, to indicate group membership. SPI0 and SPI1 are instance _B.
Note 2. All pins of group membership can be used.
Note 3. 1 µs at the longest

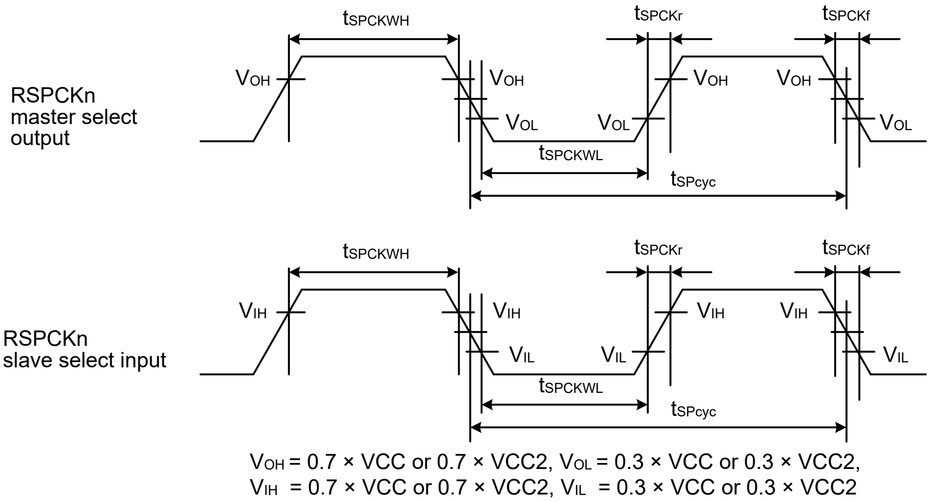


Figure 62.75 SPI clock timing

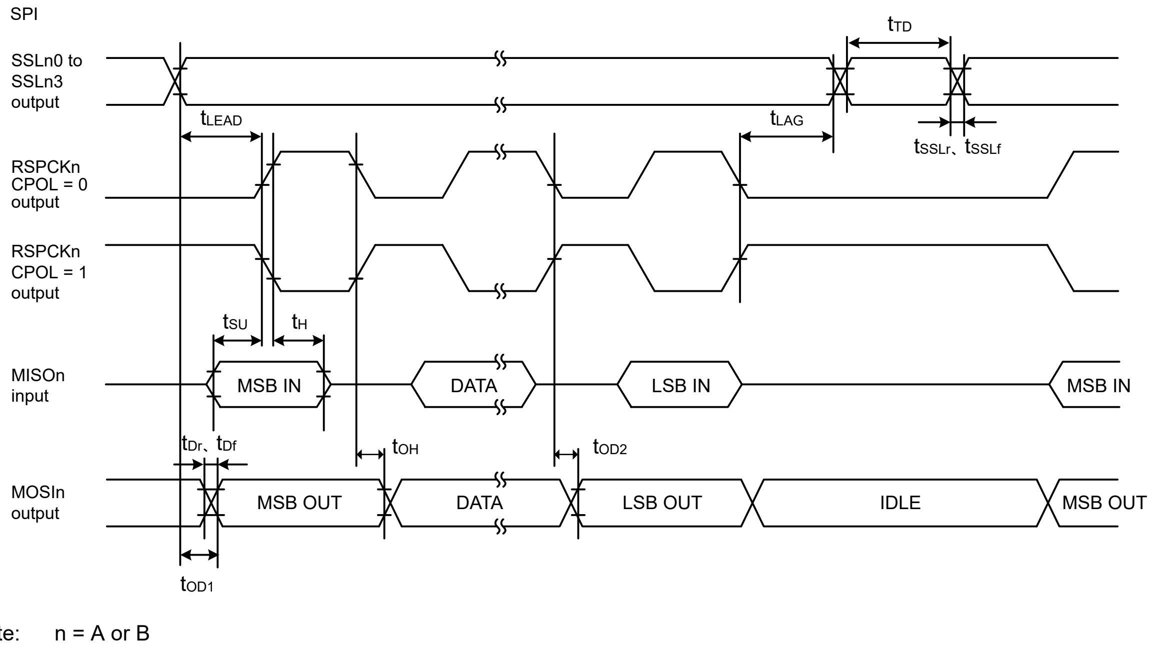


Figure 62.76 SPI timing for Motorola SPI master when CPHA = 0

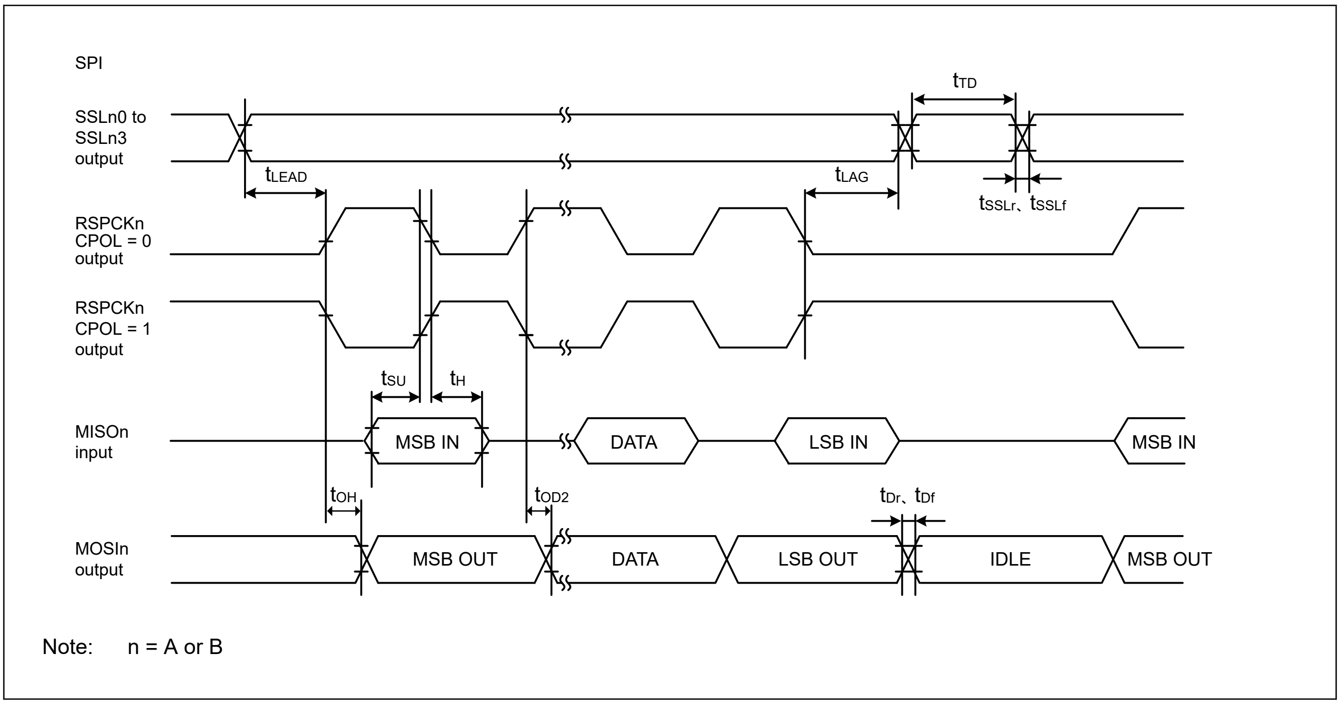


Figure 62.77 SPI timing for Motorola SPI master when CPHA = 1

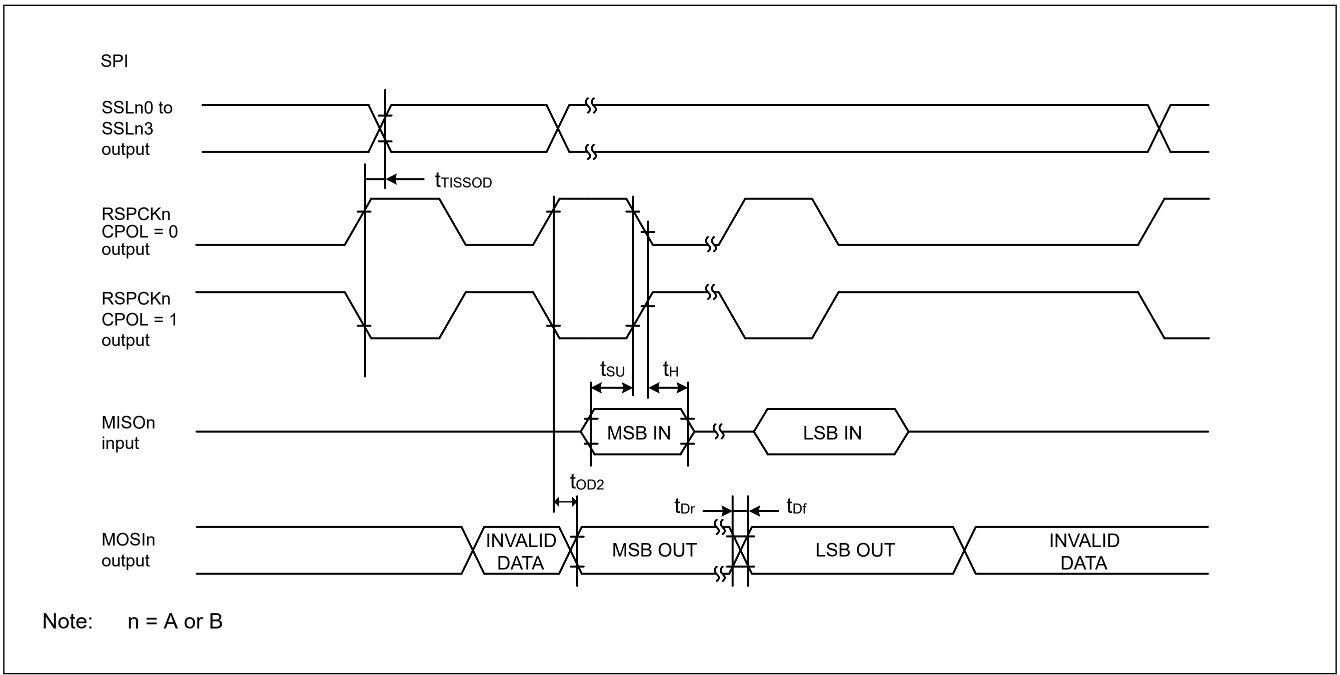


Figure 62.78 SPI timing for TI SSP master

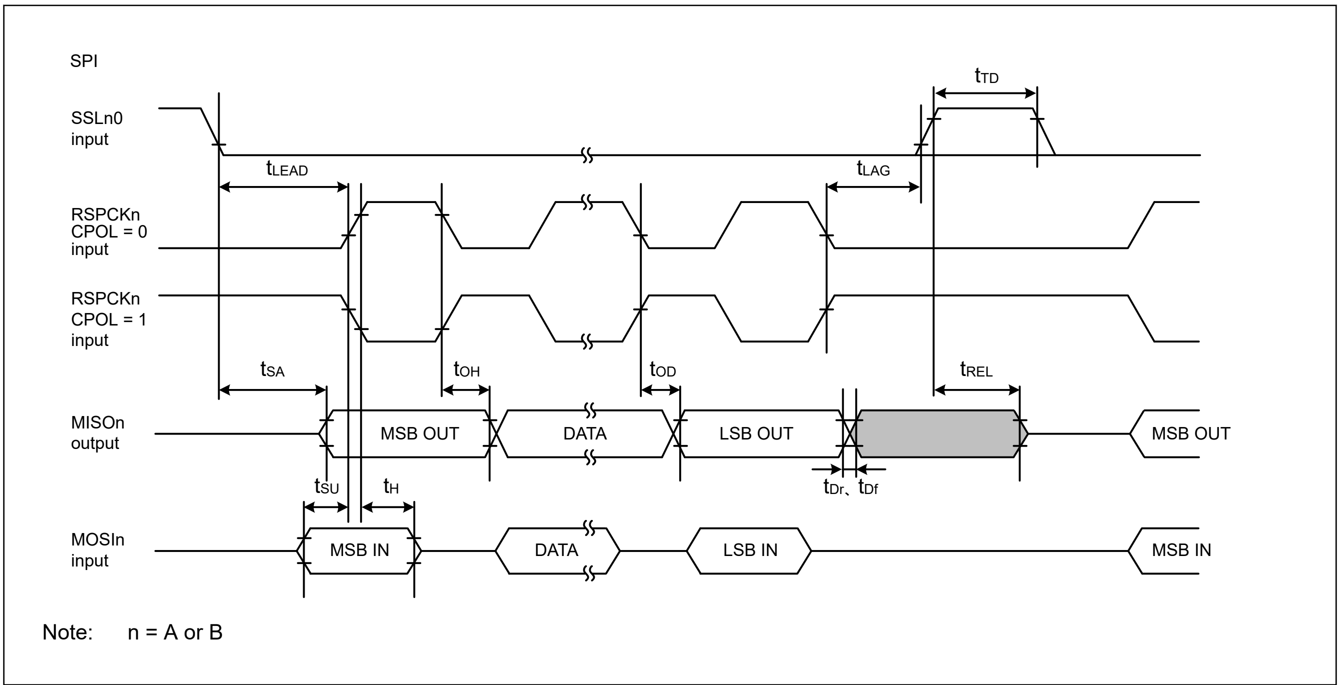


Figure 62.79 SPI timing for Motorola SPI slave when CPHA = 0

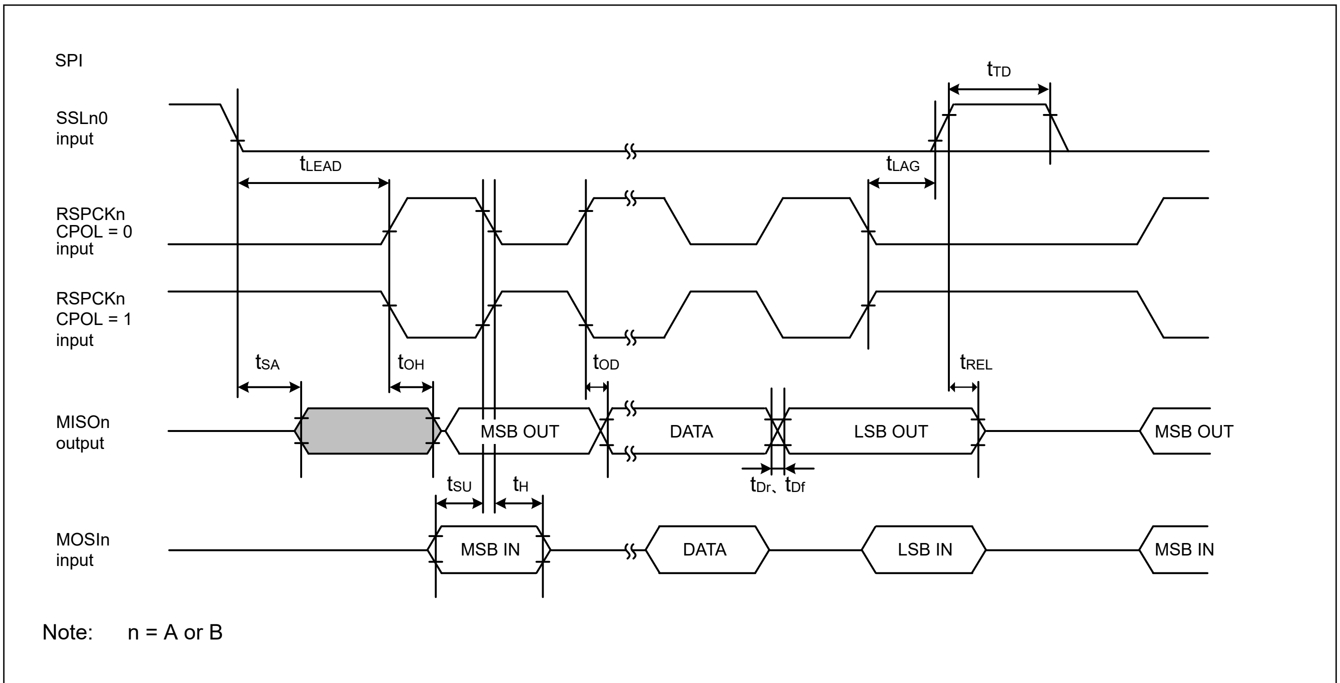


Figure 62.80 SPI timing for Motorola SPI slave when CPHA = 1

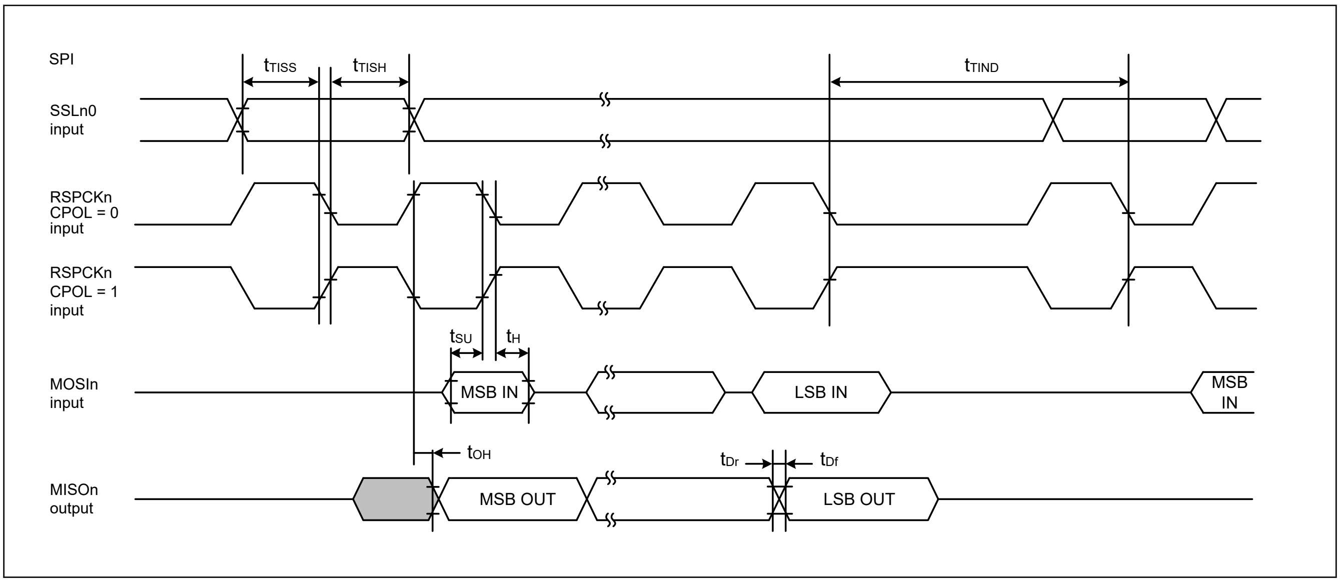


Figure 62.81 SPI timing for TI SSP slave when transmit with delay between frames

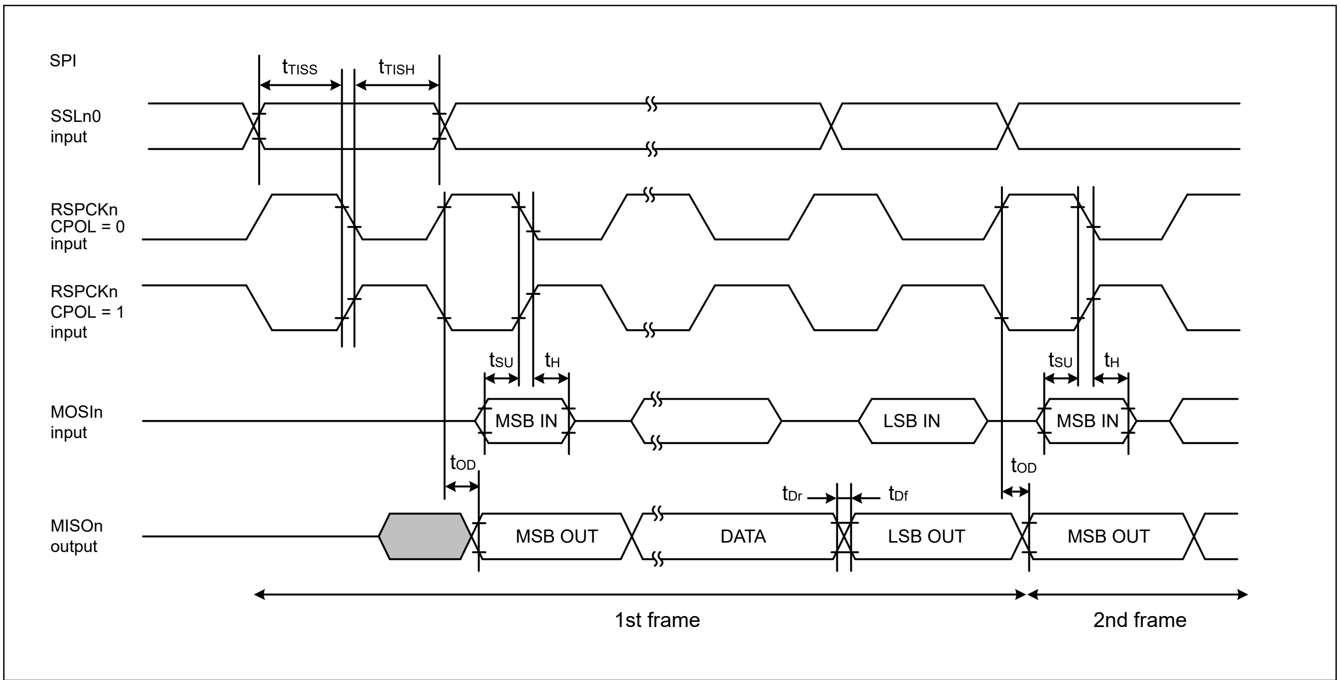


Figure 62.82 SPI timing for TI SSP slave when transmit with no delay between frames

62.3.11 OSPI Timing

Table 62.64 OSPI timing (1 of 9)

Conditions:
High-speed high drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM_n_SCLK, OM_n_SCLKN, OM_n_SIO7-0, OM_n_DQS.
Middle drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM_n_CS0, OM_n_CS1.
Load capacitance C = 15pF

Item		Symbol	VCC/VCC2	VDD	Min	Max	Unit	Note
Cycle time	SDR without OM_DQS	t_{PERIOD}	2.70 V or above	VSCR_1, voltage range 1	16.67	—	ns	Figure 62.83
				VSCR_2, voltage range 2				
			1.62 V to 2.00 V	VSCR_1, voltage range 1	20.00	—		
				VSCR_2, voltage range 2				
	SDR with OM_DQS/DDR	t_{PERIOD}	2.70 V or above	VSCR_1, voltage range 1	6.00	—	ns	
				VSCR_2, voltage range 2	7.50	—		
			1.62 V to 2.00 V	VSCR_1, voltage range 1				
				VSCR_2, voltage range 2				
Clock output slew rate		t_{SRck}	2.70 V or above	VSCR_1, voltage range 1	0.94	—	V/ns	Figure 62.83
				VSCR_2, voltage range 2	0.75	—		
			1.62 V to 2.00 V	VSCR_1, voltage range 1				
				VSCR_2, voltage range 2				
Clock Duty cycle-distortion		t_{CKDCD}	2.70 V or above	VSCR_1, voltage range 1	0	0.3	ns	Figure 62.83
				VSCR_2, voltage range 2	0	0.375		
			1.62 V to 2.00 V	VSCR_1, voltage range 1				
				VSCR_2, voltage range 2				

Table 62.64 OSPI timing (2 of 9)

Conditions:
High-speed high drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM_n_SCLK, OM_n_SCLKN, OM_n_SIO7-0, OM_n_DQS.
Middle drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM_n_CS0, OM_n_CS1.
Load capacitance C = 15pF

Item	Symbol	VCC/VCC2	VDD	Min	Max	Unit	Note
Clock Minimum Pulse width	t_{CKMPW}	2.70 V or above	VSCR_1, voltage range 1	2.7	—	ns	Figure 62.83
			VSCR_2, voltage range 2	3.375	—		
		1.62 V to 2.00 V	VSCR_1, voltage range 1				
			VSCR_2, voltage range 2				
Differential clock crossing voltage	$V_{ox}(AC)$	2.70 V or above	VSCR_1, voltage range 1	0.2 × VCC2	0.6 × VCC2	V	Figure 62.83
			VSCR_2, voltage range 2				
		1.62 V to 2.00 V	VSCR_1, voltage range 1				
			VSCR_2, voltage range 2				
DS Duty cycle distortion	t_{DSDCD}	2.70 V or above	VSCR_1, voltage range 1	0	0.24	ns	Figure 62.83
			VSCR_2, voltage range 2	0	0.3		
		1.62 V to 2.00 V	VSCR_1, voltage range 1				
			VSCR_2, voltage range 2				
DS Minimum Pulse width	t_{DSMPW}	2.70 V or above	VSCR_1, voltage range 1	2.46	—	ns	Figure 62.83
			VSCR_2, voltage range 2	3.075	—		
		1.62 V to 2.00 V	VSCR_1, voltage range 1				
			VSCR_2, voltage range 2				

Table 62.64 OSPI timing (3 of 9)

Conditions:
High-speed high drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM_n_SCLK, OM_n_SCLKN, OM_n_SIO7-0, OM_n_DQS.
Middle drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM_n_CS0, OM_n_CS1.
Load capacitance C = 15pF

Item		Symbol	VCC/VCC2	VDD	Min	Max	Unit	Note
Data input/output slew rate		t_{SR}	2.70 V or above	VSCR_1, voltage range 1	1.72	—	ns	Figure 62.83
				VSCR_2, voltage range 2	1.37	—		
			1.62 V to 2.00 V	VSCR_1, voltage range 1	0.75	—		
				VSCR_2, voltage range 2				
Data input setup time (to OM_SCLK/ OM_SCLKN)	SDR without OM_DQS	t_{SU}	2.70 V or above	VSCR_1, voltage range 1	8.17	—	ns	Figure 62.84
				VSCR_2, voltage range 2				
			1.62 V to 2.00 V	VSCR_1, voltage range 1	13.0	—		
				VSCR_2, voltage range 2				
Data input hold time (to OM_SCLK/ OM_SCLKN)		t_H	2.70 V or above	VSCR_1, voltage range 1	0.5	—	ns	
				VSCR_2, voltage range 2				
			1.62 V to 2.00 V	VSCR_1, voltage range 1	0.5	—		
				VSCR_2, voltage range 2				

Table 62.64 OSPI timing (4 of 9)

Conditions:
High-speed high drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM_n_SCLK, OM_n_SCLKN, OM_n_SIO7-0, OM_n_DQS.
Middle drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM_n_CS0, OM_n_CS1.
Load capacitance C = 15pF

Item		Symbol	VCC/VCC2	VDD	Min	Max	Unit	Note
Data output valid time	SDR without OM_DQS	t_{OV}*2	2.70 V or above	VSCR_1, voltage range 1	—	5.4	ns	Figure 62.84
				VSCR_2, voltage range 2				
			1.62 V to 2.00 V	VSCR_1, voltage range 1	—	6.9		
				VSCR_2, voltage range 2				
Data output hold time		t_{OH}	2.70 V or above	VSCR_1, voltage range 1	−5.4	—	ns	
				VSCR_2, voltage range 2				
			1.62 V to 2.00 V	VSCR_1, voltage range 1	−6.9	—		
				VSCR_2, voltage range 2				
Data output buffer off time		t_{BOFF}	2.70 V or above	VSCR_1, voltage range 1	−5.4	—	ns	
				VSCR_2, voltage range 2				
			1.62 V to 2.00 V	VSCR_1, voltage range 1	−6.9	—		
				VSCR_2, voltage range 2				

Table 62.64 OSPI timing (5 of 9)

Conditions:
High-speed high drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM_n_SCLK, OM_n_SCLKN, OM_n_SIO7-0, OM_n_DQS.
Middle drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM_n_CS0, OM_n_CS1.
Load capacitance C = 15pF

Item		Symbol	VCC/VCC2	VDD	Min	Max	Unit	Note
Data input setup time (to OM_DQS)	SDR with OM_DQS/DDR	t_{SU}	2.70 V or above	VSCR_1, voltage range 1	−0.58	—	ns	Figure 62.85, Figure 62.86
				VSCR_2, voltage range 2	−0.7	—		
			1.62 V to 2.00 V	VSCR_1, voltage range 1				
				VSCR_2, voltage range 2				
Data input hold time (to OM_DQS)		t_H	2.70 V or above	VSCR_1, voltage range 1	1.88	—	ns	
				VSCR_2, voltage range 2	2.375	—		
			1.62 V to 2.00 V	VSCR_1, voltage range 1				
				VSCR_2, voltage range 2				

Table 62.64 OSPI timing (6 of 9)

Conditions:
High-speed high drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM_n_SCLK, OM_n_SCLKN, OM_n_SIO7-0, OM_n_DQS.
Middle drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM_n_CS0, OM_n_CS1.
Load capacitance C = 15pF

Item		Symbol	VCC/VCC2	VDD	Min	Max	Unit	Note
Data output valid time	SDR with OM_DQS/DDR	t_{OV}*2	2.70 V or above	VSCR_1, voltage range 1	—	$t_{PERIOD}/4 + 0.5$	ns	Figure 62.85, Figure 62.86
				VSCR_2, voltage range 2	—	$t_{PERIOD}/4 + 0.6$		
			1.62 V to 2.00 V	VSCR_1, voltage range 1				
				VSCR_2, voltage range 2				
Data output hold time		t_{OH}	2.70 V or above	VSCR_1, voltage range 1	0.7	—	ns	
				VSCR_2, voltage range 2	0.9	—		
			1.62 V to 2.00 V	VSCR_1, voltage range 1				
				VSCR_2, voltage range 2				
Data output buffer off time		t_{BOFF}	2.70 V or above	VSCR_1, voltage range 1	0.7	—	ns	
				VSCR_2, voltage range 2	0.9	—		
			1.62 V to 2.00 V	VSCR_1, voltage range 1				
				VSCR_2, voltage range 2				

Table 62.64 OSPI timing (7 of 9)

Conditions:
High-speed high drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM_n_SCLK, OM_n_SCLKN, OM_n_SIO7-0, OM_n_DQS.
Middle drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM_n_CS0, OM_n_CS1.
Load capacitance C = 15pF

Item	Symbol	VCC/VCC2	VDD	Min	Max	Unit	Note
Clock Low to CS Low	t_{CKLCSL}	2.70 V or above	VSCR_1, voltage range 1	4.8	—	ns	Figure 62.84, Figure 62.85, Figure 62.86
			VSCR_2, voltage range 2	6	—		
		1.62 V to 2.00 V	VSCR_1, voltage range 1				
			VSCR_2, voltage range 2				
CS Low to Clock High	t_{CSLCKH}*3	2.70 V or above	VSCR_1, voltage range 1	4.8	—	ns	
			VSCR_2, voltage range 2	6	—		
		1.62 V to 2.00 V	VSCR_1, voltage range 1				
			VSCR_2, voltage range 2				
Clock Low to CS High	t_{CKLCSH}	2.70 V or above	VSCR_1, voltage range 1	4.8	—	ns	
			VSCR_2, voltage range 2	6	—		
		1.62 V to 2.00 V	VSCR_1, voltage range 1				
			VSCR_2, voltage range 2				
CS High to Clock High	t_{CSHCKH}	2.70 V or above	VSCR_1, voltage range 1	4.8	—	ns	
			VSCR_2, voltage range 2	6	—		
		1.62 V to 2.00 V	VSCR_1, voltage range 1				
			VSCR_2, voltage range 2				

Table 62.64 OSPI timing (8 of 9)

Conditions:
High-speed high drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM_n_SCLK, OM_n_SCLKN, OM_n_SIO7-0, OM_n_DQS.
Middle drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM_n_CS0, OM_n_CS1.
Load capacitance C = 15pF

<table>
<tr><th>Item</th><th>Symbol</th><th>VCC/VCC2</th><th>VDD</th><th>Min</th><th>Max</th><th>Unit</th><th>Note</th></tr>
<tr><td rowspan="4">DS Low output to CS High</td><td rowspan="4">t_{CSHCKH}</td><td rowspan="2">2.70 V or above</td><td>VSCR_1, voltage range 1</td><td>$0.8 \times t_{PERIOD}$</td><td>—</td><td rowspan="4">ns</td><td rowspan="16">Figure 62.87</td></tr>
<tr><td>VSCR_2, voltage range 2</td><td rowspan="3">$0.8 \times t_{PERIOD}$</td><td rowspan="3">—</td></tr>
<tr><td rowspan="2">1.62 V to 2.00 V</td><td>VSCR_1, voltage range 1</td></tr>
<tr><td>VSCR_2, voltage range 2</td></tr>
<tr><td rowspan="4">CS High to DS Tri-State</td><td rowspan="4">t_{CSHDST}</td><td rowspan="2">2.70 V or above</td><td>VSCR_1, voltage range 1</td><td>—</td><td>t_{PERIOD}</td><td rowspan="4">ns</td></tr>
<tr><td>VSCR_2, voltage range 2</td><td rowspan="3">—</td><td rowspan="3">t_{PERIOD}</td></tr>
<tr><td rowspan="2">1.62 V to 2.00V</td><td>VSCR_1, voltage range 1</td></tr>
<tr><td>VSCR_2, voltage range 2</td></tr>
<tr><td rowspan="4">CS Low to DS Low input[*1 *3]</td><td rowspan="4">t_{CSLDSL}</td><td rowspan="2">2.70 V or above</td><td>VSCR_1, voltage range 1</td><td>0</td><td>12.5</td><td rowspan="4">ns</td></tr>
<tr><td>VSCR_2, voltage range 2</td><td>0</td><td>20</td></tr>
<tr><td rowspan="2">1.62 V to 2.00 V</td><td>VSCR_1, voltage range 1</td><td rowspan="2">0</td><td rowspan="2">12.5</td></tr>
<tr><td>VSCR_2, voltage range 2</td></tr>
<tr><td rowspan="4">DS Tri-State to CS Low</td><td rowspan="4">t_{DSTCSL}</td><td rowspan="2">2.70 V or above</td><td>VSCR_1, voltage range 1</td><td>0</td><td>—</td><td rowspan="4">ns</td></tr>
<tr><td>VSCR_2, voltage range 2</td><td rowspan="3">0</td><td rowspan="3">—</td></tr>
<tr><td rowspan="2">1.62 V to 2.00 V</td><td>VSCR_1, voltage range 1</td></tr>
<tr><td>VSCR_2, voltage range 2</td></tr>
</table>

Table 62.64 OSPI timing (9 of 9)

Conditions:
High-speed high drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM_n_SCLK, OM_n_SCLKN, OM_n_SIO7-0, OM_n_DQS.
Middle drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM_n_CS0, OM_n_CS1.
Load capacitance C = 15pF

Item	Symbol	VCC/VCC2	VDD	Min	Max	Unit	Note
Clock High to DQS input*4	t_{CKHDSH}	2.70 V or above	VSCR_1, voltage range 1	—	t_{PERIOD} × (1 + DDRSMPEX [3:0]) − 8.5	ns	Figure 62.85
			VSCR_2, voltage range 2				
		1.62 V to 2.00 V	VSCR_1, voltage range 1	—	t_{PERIOD} × (1 + DDRSMPEX [3:0]) − 12.5		
			VSCR_2, voltage range 2				

Note: n = 0, 1
Note 1. This restriction does not need to be met when using the JESD251 Profile 1.0 memory with an external pull-down attached to the OM_DQS pin.
Note 2. Condition: COMCFG.OEASTEX = 1
Note 3. Condition: LIOCFGCSx.CSASTEX = 1
Note 4. See the datasheet of memory and set DDRSMPEX[3:0] bits to satisfy this value.

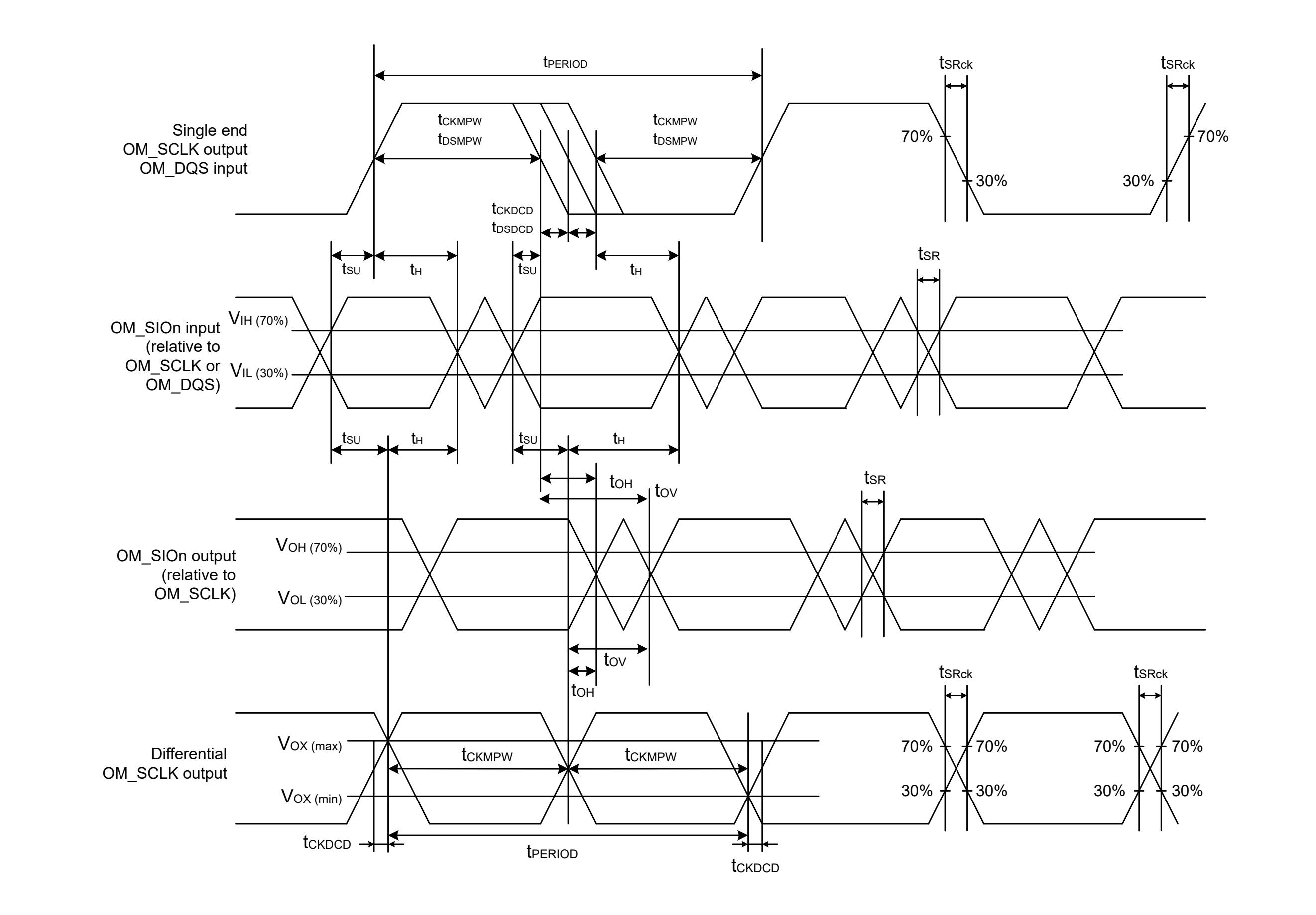


Figure 62.83 OSPI clock / DS timing

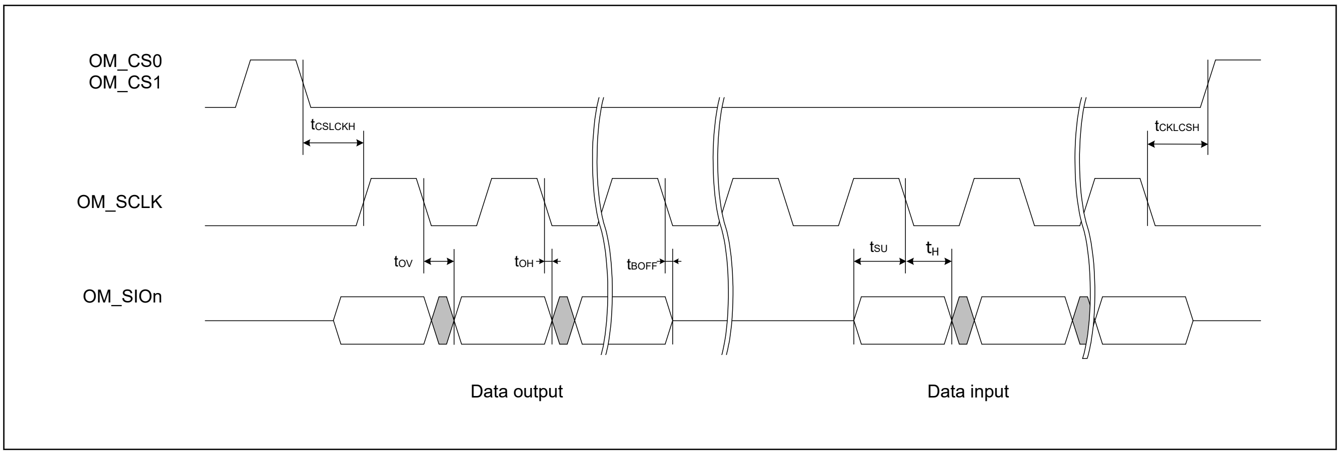


Figure 62.84 SDR transmit/receive timing (1S-1S-1S, 1S-2S-2S, 2S-2S-2S, 1S-4S-4S, 4S-4S-4S)

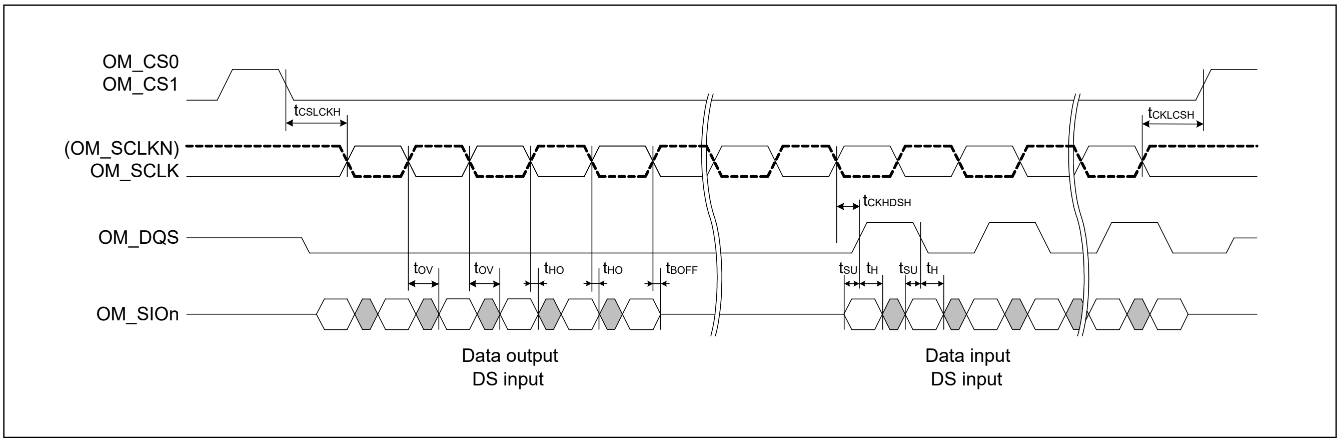


Figure 62.85 DDR transmit/receive timing (4S-4D-4D, 8D-8D-8D)

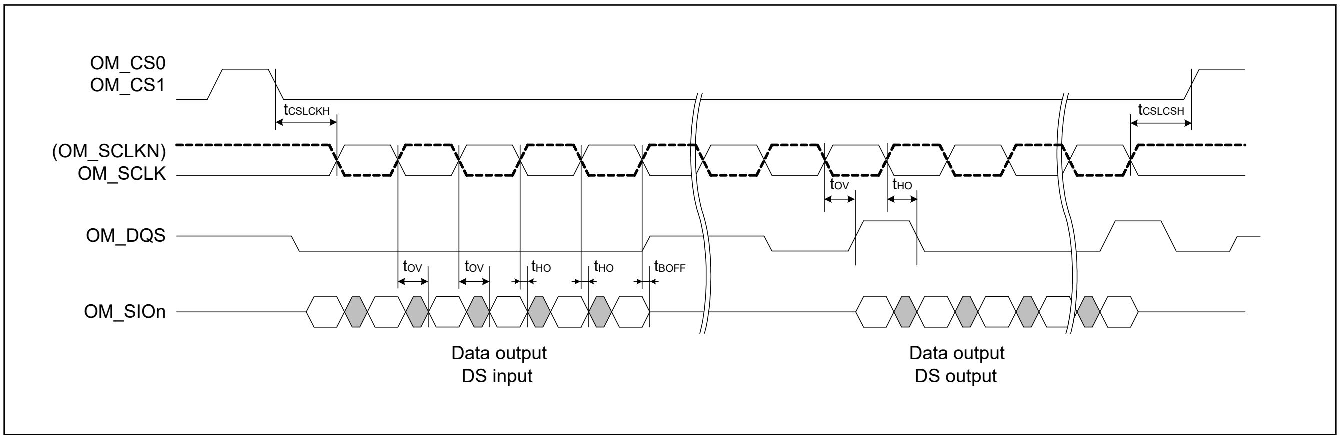


Figure 62.86 DDR transmit/receive timing (HyperRAM write)

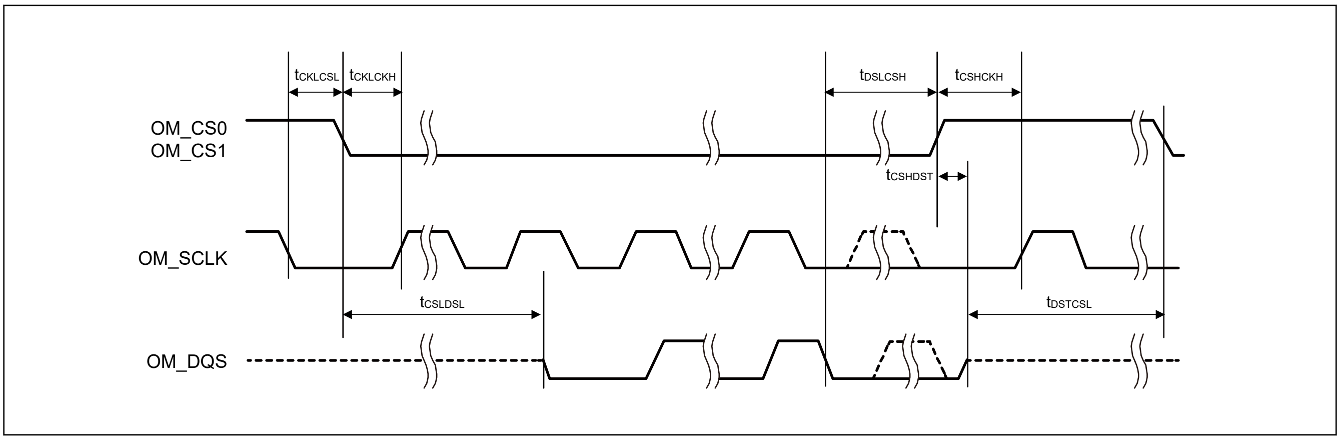


Figure 62.87 DS to CS signal timing

62.3.12 IIC Timing

Table 62.65 IIC timing (1) (1 of 2)

Conditions:

1. Middle drive output is selected when VCC is 2.70 V or above, High drive output is selected when VCC is 1.62 to 1.95 V in the port drive capability bit in the PmnPFS register for the following pins: SDA0_B, SCL0_B, SDA1_B, SCL1_B, SDA2_B, SCL2_B
2. The following pins do not require setting: SCL0_A, SDA0_A. SCL1_A, SDA1_A. SCL2_A, SDA2_A
3. Use pins that have a letter appended to their names, for instance "_A" or "_B", to indicate group membership.

For the IIC interface, the AC portion of the electrical characteristics is measured for each group.

Parameter		Symbol	VCC	Min	Max	Unit	Test conditions
IIC (Standard mode, SMBus) ICFER.FMPE = 0	SCL input cycle time	t_{SCL}	2.70 V or above	6 (12) × t_{IICcyc} + 1300	—	ns	Figure 62.88
			1.62 to 1.95 V				
	SCL input high pulse width	t_{SCLH}	2.70 V or above	3 (6) × t_{IICcyc} + 300	—	ns	
			1.62 to 1.95 V				
	SCL input low pulse width	t_{SCLL}	2.70 V or above	3 (6) × t_{IICcyc} + 300	—	ns	
			1.62 to 1.95 V				
	SCL, SDA rise time	t_{Sr}	2.70 V or above	—	1000	ns	
			1.62 to 1.95 V				
	SCL, SDA fall time	t_{Sf}	2.70 V or above	—	300	ns	
			1.62 to 1.95 V				
	SCL, SDA input spike pulse removal time	t_{SP}	2.70 V or above	0	1 (4) × t_{IICcyc}	ns	
			1.62 to 1.95 V				
	SDA input bus free time when wakeup function is disabled	t_{BUF}	2.70 V or above	3 (6) × t_{IICcyc} + 300	—	ns	
			1.62 to 1.95 V				
	SDA input bus free time when wakeup function is enabled	t_{BUF}	2.70 V or above	3 (6) × t_{IICcyc} + 4 × t_{Pcyc} + 300	—	ns	
			1.62 to 1.95 V				
	START condition input hold time when wakeup function is disabled	t_{STAH}	2.70 V or above	t_{IICcyc} + 300	—	ns	
			1.62 to 1.95 V				
	START condition input hold time when wakeup function is enabled	t_{STAH}	2.70 V or above	1 (5) × t_{IICcyc} + t_{Pcyc} + 300	—	ns	
			1.62 to 1.95 V				
	Repeated START condition input setup time	t_{STAS}	2.70 V or above	1000	—	ns	
			1.62 to 1.95 V				
	STOP condition input setup time	t_{STOS}	2.70 V or above	1000	—	ns	
			1.62 to 1.95 V				
	Data input setup time	t_{SDAS}	2.70 V or above	t_{IICcyc} + 50	—	ns	
			1.62 to 1.95 V				
	Data input hold time	t_{SDAH}	2.70 V or above	0	—	ns	
			1.62 to 1.95 V				
	SCL, SDA capacitive load	C_b	2.70 V or above	—	400	pF	
			1.62 to 1.95 V				

Table 62.65 IIC timing (1) (2 of 2)

Conditions:

1. Middle drive output is selected when VCC is 2.70 V or above, High drive output is selected when VCC is 1.62 to 1.95 V in the port drive capability bit in the PmnPFS register for the following pins: SDA0_B, SCL0_B, SDA1_B, SCL1_B, SDA2_B, SCL2_B
2. The following pins do not require setting: SCL0_A, SDA0_A. SCL1_A, SDA1_A. SCL2_A, SDA2_A
3. Use pins that have a letter appended to their names, for instance "_A" or "_B", to indicate group membership.

For the IIC interface, the AC portion of the electrical characteristics is measured for each group.

Parameter		Symbol	VCC	Min	Max	Unit	Test conditions
IIC (Fast mode) ICFER.FMPE = 0	SCL input cycle time	t_{SCL}	2.70 V or above	6 (12) × t_{IICcyc} + 600	—	ns	Figure 62.88
			1.62 to 1.95 V				
	SCL input high pulse width	t_{SCLH}	2.70 V or above	3 (6) × t_{IICcyc} + 300	—	ns	
			1.62 to 1.95 V				
	SCL input low pulse width	t_{SCLL}	2.70 V or above	3 (6) × t_{IICcyc} + 300	—	ns	
			1.62 to 1.95 V				
	SCL, SDA rise time	t_{Sr}	2.70 V or above	20	300	ns	
			1.62 to 1.95 V				
	SCL, SDA fall time	t_{Sf}	2.70 V or above	20 × (external pullup voltage/5.5 V)*1	300	ns	
			1.62 to 1.95 V				
	SCL, SDA input spike pulse removal time	t_{SP}	2.70 V or above	0	1 (4) × t_{IICcyc}	ns	
			1.62 to 1.95 V				
	SDA input bus free time when wakeup function is disabled	t_{BUF}	2.70 V or above	3 (6) × t_{IICcyc} + 300	—	ns	
			1.62 to 1.95 V				
	SDA input bus free time when wakeup function is enabled	t_{BUF}	2.70 V or above	3 (6) × t_{IICcyc} + 4 × t_{Pcyc} + 300	—	ns	
			1.62 to 1.95 V				
	START condition input hold time when wakeup function is disabled	t_{STAH}	2.70 V or above	t_{IICcyc} + 300	—	ns	
			1.62 to 1.95 V				
	START condition input hold time when wakeup function is enabled	t_{STAH}	2.70 V or above	1 (5) × t_{IICcyc} + t_{Pcyc} + 300	—	ns	
			1.62 to 1.95 V				
	Repeated START condition input setup time	t_{STAS}	2.70 V or above	300	—	ns	
			1.62 to 1.95 V				
	STOP condition input setup time	t_{STOS}	2.70 V or above	300	—	ns	
			1.62 to 1.95 V				
	Data input setup time	t_{SDAS}	2.70 V or above	t_{IICcyc} + 50	—	ns	
			1.62 to 1.95 V				
	Data input hold time	t_{SDAH}	2.70 V or above	0	—	ns	
			1.62 to 1.95 V				
	SCL, SDA capacitive load	C_b	2.70 V or above	—	400	pF	
			1.62 to 1.95 V				

Note: t_{IICcyc}: IIC internal reference clock (IICφ) cycle, t_{Pcyc}: PCLKB cycle.
Note: Values in parentheses apply when ICMR3.NF[1:0] is set to 11b while the digital filter is enabled with ICFER.NFE set to 1.
Note: Must use pins that have a letter appended to their name, for instance "_A", "_B", to indicate group membership. For the IIC interface, the AC portion of the electrical characteristics is measured for each group.
Note 1. Only supported for SCL0_A, SDA0_A, SCL1_A, SDA1_A, SCL2_A, and SDA2_A.

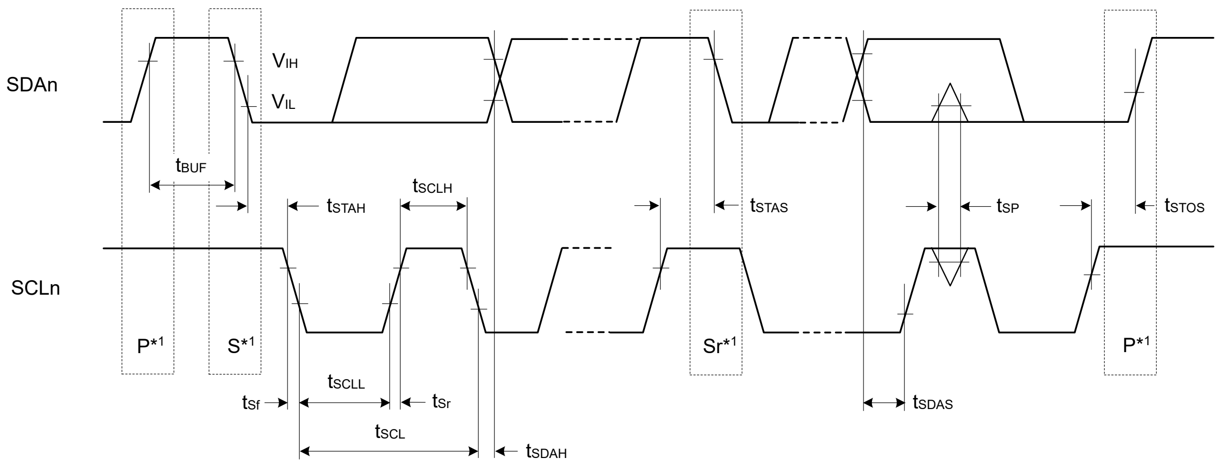
Table 62.66 IIC timing (2)

Setting of the SCL0_A, SDA0_A, SCL1_A, SDA1_A. SCL2_A, SDA2_A pins are not required with the port drive capability bit in the PmnPFS register.

Parameter		Symbol	VCC	Min	Max	Unit	Test conditions
IIC (Fast-mode+) ICFER.FMPE = 1	SCL input cycle time	t_{SCL}	2.70 V or above	6 (12) × t_{IICcyc} + 240	—	ns	Figure 62.88
			1.62 to 1.95V				
	SCL input high pulse width	t_{SCLH}	2.70 V or above	3 (6) × t_{IICcyc} + 120	—	ns	
			1.62 to 1.95V				
	SCL input low pulse width	t_{SCLL}	2.70 V or above	3 (6) × t_{IICcyc} + 120	—	ns	
			1.62 to 1.95V				
	SCL, SDA rise time	t_{Sr}	2.70 V or above	—	120	ns	
			1.62 to 1.95V				
	SCL, SDA fall time	t_{Sf}	2.70 V or above	20 × (external pullup voltage/ 5.5V)	120	ns	
			1.62 to 1.95V				
	SCL, SDA input spike pulse removal time	t_{SP}	2.70 V or above	0	1 (4) × t_{IICcyc}	ns	
			1.62 to 1.95V				
	SDA input bus free time when wakeup function is disabled	t_{BUF}	2.70 V or above	3 (6) × t_{IICcyc} + 120	—	ns	
			1.62 to 1.95V				
	SDA input bus free time when wakeup function is enabled	t_{BUF}	2.70 V or above	3 (6) × t_{IICcyc} + 4 × t_{Pcyc} + 120	—	ns	
			1.62 to 1.95V				
	Start condition input hold time when wakeup function is disabled	t_{STAH}	2.70 V or above	t_{IICcyc} + 120	—	ns	
			1.62 to 1.95V				
	START condition input hold time when wakeup function is enabled	t_{STAH}	2.70 V or above	1 (5) × t_{IICcyc} + t_{Pcyc} + 120	—	ns	
			1.62 to 1.95V				
	Restart condition input setup time	t_{STAS}	2.70 V or above	120	—	ns	
			1.62 to 1.95V				
	Stop condition input setup time	t_{STOS}	2.70 V or above	120	—	ns	
			1.62 to 1.95V				
	Data input setup time	t_{SDAS}	2.70 V or above	t_{IICcyc} + 30	—	ns	
			1.62 to 1.95V				
	Data input hold time	t_{SDAH}	2.70 V or above	0	—	ns	
			1.62 to 1.95V				
	SCL, SDA capacitive load	C_b*1	2.70 V or above	—	550	pF	
			1.62 to 1.95V				

Note: t_{IICcyc}: IIC internal reference clock (IICφ) cycle, tPcyc: PCLKB cycle.

Note: Values in parentheses apply when ICMR3.NF[1:0] is set to 11b while the digital filter is enabled with ICFER.NFE set to 1.
Note 1. Cb indicates the total capacity of the bus line.



Note 1. S, P, and Sr indicate the following conditions:
S: Start condition
P: Stop condition
Sr: Restart condition

Figure 62.88 I2C bus interface input/output timing

62.3.13 I3C Timing

Table 62.67 IIC timing (1)-1

Setting of the I3C_SCL0, I3C_SDA0 pins are not required with the Port Drive Capability bit in the PmnPFS register.

Parameter		Symbol	VCC	Min	Max	Unit
IIC (Standard mode, SMBus) BFCTL.FMPE = 0	SCL input cycle time	t_{SCL}	2.70 V or above, 1.65 to 1.95 V	10 (18) × t_{I3Ccyc} + 1300	—	ns
	SCL input high pulse width	t_{SCLH}	2.70 V or above, 1.65 to 1.95 V	5 (9) × t_{I3Ccyc} + 300	—	ns
	SCL input low pulse width	t_{SCLL}	2.70 V or above, 1.65 to 1.95 V	5 (9) × t_{I3Ccyc} + 300	—	ns
	SCL, SDA rise time	t_{Sr}	2.70 V or above, 1.65 to 1.95 V	—	1000	ns
	SCL, SDA fall time	t_{Sf}	2.70 V or above, 1.65 to 1.95 V	—	300	ns
	SCL, SDA input spike pulse removal time	t_{SP}	2.70 V or above, 1.65 to 1.95 V	0	1 (4) × t_{I3Ccyc}	ns
	SDA input bus free time when wakeup function is disabled	t_{BUF}	2.70 V or above, 1.65 to 1.95 V	5(9) × t_{I3Ccyc} + 300	—	ns
	SDA input bus free time when wakeup function is enabled	t_{BUF}	2.70 V or above, 1.65 to 1.95 V	5(9) × t_{I3Ccyc} + 4 × t_{Tcyc} + 300	—	ns
	START condition input hold time when wakeup function is disabled	t_{STAH}	2.70 V or above, 1.65 to 1.95 V	t_{I3Ccyc} + 300	—	ns
	START condition input hold time when wakeup function is enabled	t_{STAH}	2.70 V or above, 1.65 to 1.95 V	1(5) × t_{I3Ccyc} + t_{Tcyc} + 300	—	ns
	Repeated START condition input setup time	t_{STAS}	2.70 V or above, 1.65 to 1.95 V	1000	—	ns
	STOP condition input setup time	t_{STOS}	2.70 V or above, 1.65 to 1.95 V	1000	—	ns
	Data input setup time	t_{SDAS}	2.70 V or above, 1.65 to 1.95 V	t_{I3Ccyc} + 50	—	ns
	Data input hold time	t_{SDAH}	2.70 V or above, 1.65 to 1.95 V	0	—	ns
	SCL, SDA capacitive load	C_b*1	2.70 V or above, 1.65 to 1.95 V	—	400	pF

Note: t_{I3Ccyc}: I3C internal reference clock (I3Cφ) cycle, t_{Tcyc}: TCLK cycle.

Note: Values in parentheses apply when INCTL.DNFS[3:0] is set to 0011b while the digital filter is enabled with INCTL.DNFE set to 1.

Note 1. C_b indicates the total capacity of the bus line.

Table 62.68 IIC timing (1)-2

Setting of the I3C_SCL0, I3C_SDA0 pins are not required with the Port Drive Capability bit in the PmnPFS register.

Parameter		Symbol	VCC	Min	Max	Unit
IIC (Fast-mode)	SCL input cycle time	t_{SCL}	2.70 V or above, 1.65 to 1.95 V	10 (18) × t_{I3Ccyc} + 600	—	ns
	SCL input high pulse width	t_{SCLH}	2.70 V or above, 1.65 to 1.95 V	5 (9) × t_{I3Ccyc} + 300	—	ns
	SCL input low pulse width	t_{SCLL}	2.70 V or above, 1.65 to 1.95 V	5 (9) × t_{I3Ccyc} + 300	—	ns
	SCL, SDA rise time	t_{Sr}	2.70 V or above, 1.65 to 1.95 V	20	300	ns
	SCL, SDA fall time	t_{Sf}	2.70 V or above, 1.65 to 1.95 V	20 × (external pull-up voltage/3.6 V)	300	ns
	SCL, SDA input spike pulse removal time	t_{SP}	2.70 V or above, 1.65 to 1.95 V	0	1 (4) × t_{I3Ccyc}	ns
	SDA input bus free time when wakeup function is disabled	t_{BUF}	2.70 V or above, 1.65 to 1.95 V	5 (9) × t_{I3Ccyc} + 300	—	ns
	SDA input bus free time when wakeup function is enabled		2.70 V or above, 1.65 to 1.95 V	5(9) × t_{I3Ccyc} + 4 × t_{Tcyc} + 300	—	ns
	START condition input hold time when wakeup function is disabled	t_{STAH}	2.70 V or above, 1.65 to 1.95 V	t_{I3Ccyc} + 300	—	ns
	START condition input hold time when wakeup function is enabled		2.70 V or above, 1.65 to 1.95 V	1(5) × t_{I3Ccyc} + t_{Tcyc} + 300	—	ns
	Repeated START condition input setup time	t_{STAS}	2.70 V or above, 1.65 to 1.95 V	300	—	ns
	STOP condition input setup time	t_{STOS}	2.70 V or above, 1.65 to 1.95 V	300	—	ns
	Data input setup time	t_{SDAS}	2.70 V or above, 1.65 to 1.95 V	t_{I3Ccyc} + 50	—	ns
	Data input hold time	t_{SDAH}	2.70 V or above, 1.65 to 1.95 V	0	—	ns
	SCL, SDA capacitive load	C_b*1	2.70 V or above, 1.65 to 1.95 V	—	400	pF

Note: t_{I3Ccyc}: I3C internal reference clock (I3Cφ) cycle, t_{Tcyc}: TCLK cycle.

Note: Values in parentheses apply when INCTL.DNFS[3:0] is set to 0011b while the digital filter is enabled with INCTL.DNFE set to 1.

Note 1. C_b indicates the total capacity of the bus line.

Table 62.69 IIC timing (1)-3

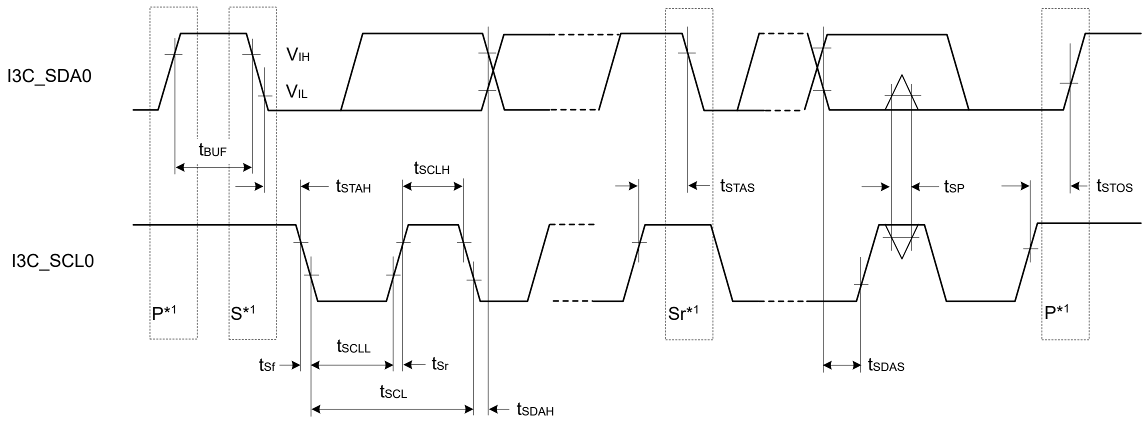
Setting of the I3C_SCL0, I3C_SDA0 pins are not required with the Port Drive Capability bit in the PmnPFS register.

Parameter		Symbol	VCC	Min	Max	Unit
IIC (Fast-mode +) BFCTL.FMPE = 1	SCL input cycle time	t_{SCL}	2.70 V or above, 1.65 to 1.95 V	10 (18) × t_{I3Ccyc} + 240	—	ns
	SCL input high pulse width	t_{SCLH}	2.70 V or above, 1.65 to 1.95 V	5 (9) × t_{I3Ccyc} + 120	—	ns
	SCL input low pulse width	t_{SCLL}	2.70 V or above, 1.65 to 1.95 V	5 (9) × t_{I3Ccyc} + 120	—	ns
	SCL, SDA rise time	t_{Sr}	2.70 V or above, 1.65 to 1.95 V	—	120	ns
	SCL, SDA fall time	t_{Sf}	2.70 V or above, 1.65 to 1.95 V	20 × (external pull-up voltage/3.3V)	120	ns
	SCL, SDA input spike pulse removal time	t_{SP}	2.70 V or above, 1.65 to 1.95 V	0	1 (4) × t_{I3Ccyc}	ns
	SDA input bus free time when wakeup function is disabled	t_{BUF}	2.70 V or above, 1.65 to 1.95 V	5 (9) × t_{I3Ccyc} + 120	—	ns
	SDA input bus free time when wakeup function is enabled			5(9) × t_{I3Ccyc} + 4 × t_{Tcyc} + 120		ns
	START condition input hold time when wakeup function is disabled	t_{STAH}	2.70 V or above, 1.65 to 1.95 V	t_{I3Ccyc} + 120	—	ns
	START condition input hold time when wakeup function is enabled			1(5) × t_{I3Ccyc} + t_{Tcyc} + 120		ns
	Restart condition input setup time	t_{STAS}	2.70 V or above, 1.65 to 1.95 V	120	—	ns
	Stop condition input setup time	t_{STOS}	2.70 V or above, 1.65 to 1.95 V	120	—	ns
	Data input setup time	t_{SDAS}	2.70 V or above, 1.65 to 1.95 V	t_{I3Ccyc} + 30	—	ns
	Data input hold time	t_{SDAH}	2.70 V or above, 1.65 to 1.95 V	0	—	ns
	SCL, SDA capacitive load	C_b*1	2.70 V or above, 1.65 to 1.95 V	—	550	pF

Note: t_{I3Ccyc}: I3C internal reference clock (I3Cφ) cycle, t_{Tcyc}: TCLK cycle.

Note: Values in parentheses apply when INCTL.DNFS[3:0] is set to 0011b while the digital filter is enabled with INCTL.DNFE set to 1.

Note 1. C_b indicates the total capacity of the bus line.



Test conditions:
V_{IH} = VCC × 0.7, V_{IL} = VCC × 0.3
V_{OL} = 0.6 V, I_{OL} = 6 mA (BFCTL.FMPE = 0)
V_{OL} = 0.4 V, I_{OL} = 15 mA (BFCTL.FMPE = 1)

Note 1. S, P, and Sr indicate the following conditions:
S: Start condition
P: Stop condition
Sr: Restart condition

Figure 62.89 I²C bus interface input/output timing

Table 62.70 IIC timing (2)

Setting of the I3C_SCL0, I3C_SDA0 pins are not required with the Port Drive Capability bit in the PmnPFS register.

Parameter			Symbol	VCC	Min	Max	Unit
IIC (Hs-mode) BFCTL.HSME = 1	SCL input cycle time		t_{SCL}	3.00 V or above	46 (48) × t_{I3Ccyc}	—	ns
				1.65 to 1.95 V	46 (48) × t_{I3Ccyc}	—	
	SCL input high pulse width	Cb = 400 pF	t_{SCLH}	3.00 V or above	29 (30) × t_{I3Ccyc}	—	ns
				1.65 to 1.95 V	29 (30) × t_{I3Ccyc}	—	
		Cb = 100 pF		3.00 V or above	13 (14) × t_{I3Ccyc}	—	
				1.65 to 1.95 V	13 (14) × t_{I3Ccyc}	—	
	SCL input low pulse width	Cb = 400 pF	t_{SCLL}	3.00 V or above	69 (70) × t_{I3Ccyc}	—	ns
				1.65 to 1.95 V	69 (70) × t_{I3Ccyc}	—	
		Cb = 100 pF		3.00 V or above	33 (34) × t_{I3Ccyc}	—	
				1.65 to 1.95 V	33 (34) × t_{I3Ccyc}	—	
	SCL rise time	Cb = 400 pF	t_{SrCL}	3.00 V or above	—	80	ns
				1.65 to 1.95 V	—	80	
		Cb = 100 pF		3.00 V or above	—	40	
				1.65 to 1.95 V	—	40	
	SDA rise time	Cb = 400 pF	t_{SrDA}	3.00 V or above	—	160	ns
				1.65 to 1.95 V	—	160	
		Cb = 100 pF		3.00 V or above	—	80	
				1.65 to 1.95 V	—	80	
	SCL fall time	Cb = 400 pF	t_{SfCL}	3.00 V or above	—	80	ns
				1.65 to 1.95 V	—	80	
		Cb = 100 pF		3.00 V or above	—	40	
				1.65 to 1.95 V	—	40	
	SDA fall time	Cb = 400 pF	t_{SfDA}	3.00 V or above	—	160	ns
				1.65 to 1.95 V	—	160	
		Cb = 100 pF		3.00 V or above	—	80	
				1.65 to 1.95 V	—	80	
	SCL, SDA input spike pulse removal time		t_{SP}	3.00 V or above	0	1 (1) × t_{I3Ccyc}	ns
				1.65 to 1.95 V	0	1 (1) × t_{I3Ccyc}	
	Repeated START condition input setup time		t_{STAS}	3.00 V or above	40	—	ns
				1.65 to 1.95 V	40	—	
	STOP condition input setup time		t_{STOS}	3.00 V or above	40	—	ns
				1.65 to 1.95 V	40	—	
	Data input setup time		t_{SDAS}	3.00 V or above	10	—	ns
				1.65 to 1.95 V	10	—	
	Data input hold time	Cb = 400 pF	t_{SDAH}	3.00 V or above	0	150	ns
				1.65 to 1.95 V	0	150	
		Cb = 100 pF		3.00 V or above	0	70	
				1.65 to 1.95 V	0	70	
	SCL, SDA capacitive load		C_b*1	3.00 V or above	—	400	pF
				1.65 to 1.95 V	—	400	

Note: t_{I3Ccyc}: I3C internal reference clock (I3Cφ) cycle.
Note: Values in parentheses apply when INCTL.DNFS[3:0] is set to 0011b while the digital filter is enabled with INCTL.DNFE set to 1.
Note 1. C_b indicates the total capacity of the bus line.

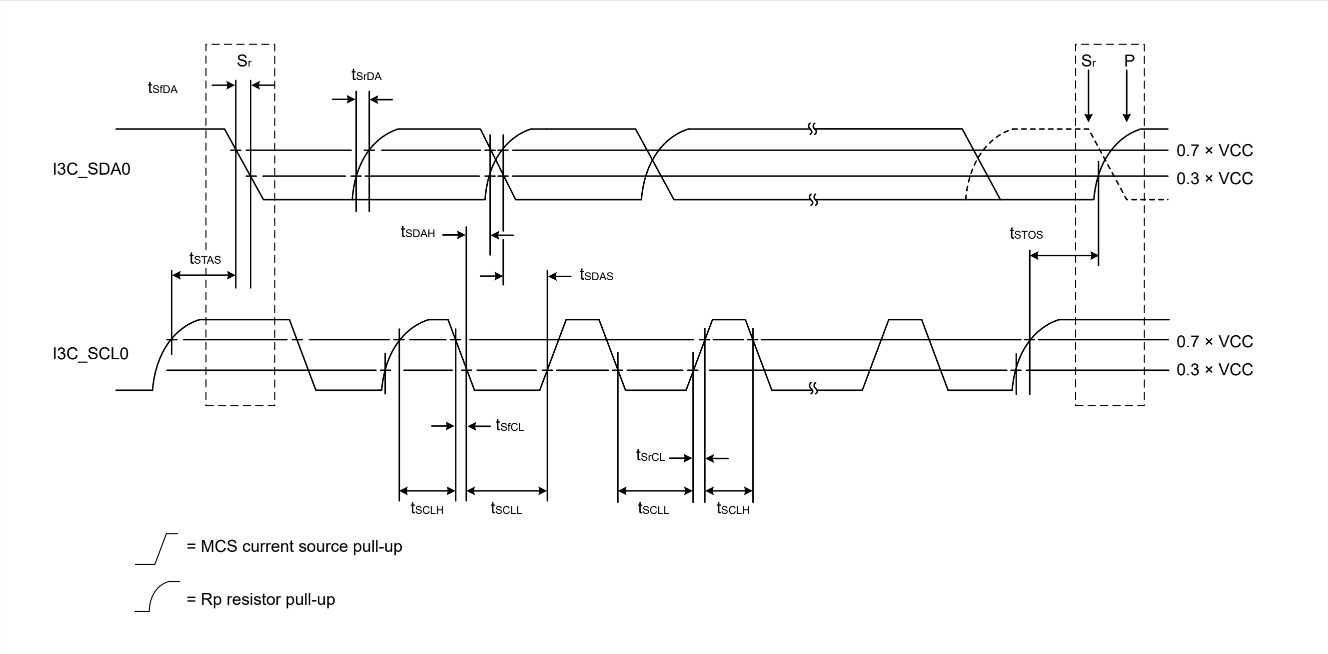


Figure 62.90 **I2C bus interface input/output timing (Hs-mode)**

Table 62.71 I3C timing (Open Drain Timing Parameters)

Setting of the I3C_SCL0, I3C_SDA0 pins are not required with the Port Drive Capability bit in the PmnPFS register.

Parameter		Symbol	VCC	Min	Max	Unit	Test conditions
I3C Open Drain Timing Parameters	SCL Clock Low Period	t_{LOW_OD} *1 *2	3.00 V or above	200	—	ns	Figure 62.93
			1.65 to 1.95 V	200	—		
		$t_{DIG\_OD\_L}$	3.00 V or above	$t_{LOW\_ODmin} + t_{fDA\_ODmin}$	—	ns	Figure 62.93
			1.65 to 1.95 V	$t_{LOW\_ODmin} + t_{fDA\_ODmin}$	—		
	SCL Clock High Period	t_{HIGH} *3 *4	3.00 V or above	—	41	ns	Figure 62.91
			1.65 to 1.95 V	—	41		
		t_{DIG_H}	3.00 V or above	—	$t_{HIGH} + t_{CF}$	ns	Figure 62.91
			1.65 to 1.95 V	—	$t_{HIGH} + t_{CF}$		
	SDA Signal Fall Time	t_{fDA_OD}	3.00 V or above	t_{CF}	12	ns	Figure 62.93
			1.65 to 1.95 V	t_{CF}	12		
	SDA Data Setup Time Open Drain Mode	t_{SU_OD} *1	3.00 V or above	12	—	ns	Figure 62.92
			1.65 to 1.95 V	18	—		
	Clock After START (S) Condition	t_{CAS} *5 *6	3.00 V or above	38.4 nano	For ENAS0: 1 μ	seconds	Figure 62.93
					For ENAS1: 100 μ		
					For ENAS2: 2 milli		
					For ENAS3: 50 milli		
			1.65 to 1.95 V	38.4 nano	For ENAS0: 1 μ		
					For ENAS1: 100 μ		
					For ENAS2: 2 milli		
					For ENAS3: 50 milli		
	Clock Before STOP (P) Condition	t_{CBP}	3.00 V or above	t_{CASmin} / 2	—	seconds	Figure 62.94
			1.65 to 1.95 V	t_{CASmin} / 2	—		
	Current Master to Secondary Master Overlap time during handoff	$t_{MMOverlap}$	3.00 V or above	$t_{DIG\_OD\_Lmin}$	—	ns	Figure 62.100
			1.65 to 1.95 V	$t_{DIG\_OD\_Lmin}$	—		
	Bus Available Condition	t_{AVAL} *7	3.00 V or above	1	—	μs	—
			1.65 to 1.95 V	1	—		
	Bus Idle Condition	t_{IDLE}	3.00 V or above	1	—	ms	—
			1.65 to 1.95 V	1	—		
	Time Internal Where New Master Not Driving SDA Low	t_{MMLock}	3.00 V or above	$t_{AVALmin}$	—	μs	Figure 62.100
			1.65 to 1.95 V	$t_{AVALmin}$	—		

Note 1. This is approximately equal to $t_{LOWmin} + t_{DS\_ODmin} + t_{rDA\_ODtyp} + t_{SU\_Odmin}$.

Note 2. The Master may use a shorter Low period if it knows that this is safe, i.e., that SDA is already above VIH

Note 3. Based on t_{SPIKE}, rise and fall times, and interconnect

Note 4. This maximum High period may be exceeded when the signals can be safely seen by Legacy I^{2}C Devices, and/or in consideration of the interconnect (e.g., a short Bus).
As a product specification, if this Max value cannot be guaranteed, change this Max value and specify that it cannot be used in the Mixed Bus.

Note 5. On a Legacy Bus where I^2C Devices need to see Start
Note 6. Slaves that do not support the optional ENTASx CCCs shall use the t_{CAS} Max value shown for ENTAS3
Note 7. On a Mixed Bus with Fm Legacy I^2C Devices, t_{AVAL} is 300 ns shorter than the Fm Bus Free Condition time (t_{BUF})

Table 62.72 I3C timing (Push-Pull Timing Parameters for SDR and HDR-DDR Modes)

Setting of the I3C_SCL0, I3C_SDA0 pins are not required with the Port Drive Capability bit in the PmnPFS register.

Parameter			Symbol	VCC	Min	Max	Unit	Test conditions
I3C Push-Pull Timing Parameters for SDR and HDR-DDR Modes	SCL Clock Frequency		f_{SCL}*1	3.00 V or above	0.01	12.5	MHz	—
				1.65 to 1.95 V	0.01	12.5		
	SCL Clock Low Period		t_{LOW}	3.00 V or above	27	—	ns	Figure 62.91
				1.65 to 1.95 V	32	—		
			t_{DIG_L}*2 *4	3.00 V or above	35	—	ns	Figure 62.91
				1.65 to 1.95 V	40	—		
	SCL Clock High Period for Mixed Bus		t_{HIGH_MIXED}	3.00 V or above	24	—	ns	Figure 62.91
				1.65 to 1.95 V	27	—		
			$t_{DIG\_H\_MIXED}$*2 *3	3.00 V or above	32	45	ns	Figure 62.91
				1.65 to 1.95 V	35	45		
	SCL Clock High Period		t_{HIGH}	3.00 V or above	24	—	ns	Figure 62.91
				1.65 to 1.95 V	27	—		
			t_{DIG_H}*2	3.00 V or above	32	—	ns	Figure 62.91
				1.65 to 1.95 V	35	—		
	Clock in to Data Out for Slave		t_{SCO}	3.00 V or above	—	12	ns	Figure 62.96
				1.65 to 1.95 V	—	12		
	SCL Clock Rise Time		t_{CR}	3.00 V or above	—	150 × 1 / f_{SCL} (capped at 60)	ns	Figure 62.91
				1.65 to 1.95 V	—	150 × 1 / f_{SCL} (capped at 60)		
	SCL Clock Fall Time		t_{CF}	3.00 V or above	—	150 × 1 / f_{SCL} (capped at 60)	ns	Figure 62.91
				1.65 to 1.95 V	—	150 × 1 / f_{SCL} (capped at 60)		
	SDA Signal Data Hold in Push-Pull Mode	Master	t_{HD_PP}*4*5	3.00 V or above	t_{CR} + 3 and t_{CF} + 3	—	—	Figure 62.95
				1.65 to 1.95 V	t_{CR} + 3 and t_{CF} + 3	—		
		Slave	t_{HD_PP}*5	3.00 V or above	0	—	—	Figure 62.95
				1.65 to 1.95 V	0	—		
	SDA Signal Data Setup in Push-Pull Mode		t_{SU_PP}	3.00 V or above	12	N/A	ns	Figure 62.97
				1.65 to 1.95 V	18	N/A		
	Clock After Repeated START (Sr)		t_{CASr}	3.00 V or above	t_{CASmin}	N/A	ns	Figure 62.99
				1.65 to 1.95 V	t_{CASmin}	N/A		
	Clock Before Repeated START (Sr)		t_{CBSr}	3.00 V or above	t_{CASmin} / 2	N/A	ns	Figure 62.99
				1.65 to 1.95 V	t_{CASmin} / 2	N/A		
	Capacitive Load per Bus Line (SDA/SCL)		C_b	3.00 V or above	—	50	pF	—
				1.65 to 1.95 V	—	50		

Note 1. $f_{SCL} = 1 / (t_{DIG\_L} + t_{DIG\_H})$
Note 2. t_{DIG_L} and t_{DIG_H} are the clock Low and High periods as seen at the receiver end of the I3C Bus using V_{IL} and V_{IH}.
Note 3. When communicating with an I3C Device on a mixed Bus, the $t_{DIG\_H\_MIXED}$ period must be constrained in order to make sure that I^2C Devices do not interpret I3C signaling as valid I^2C signaling.
Note 4. As both edges are used, the hold time needs to be satisfied for the respective edges; i.e., t_{CF} + 3 for falling edge clocks, and t_{CR} + 3 for rising edge clocks.
Note 5. In SDR Mode the Hold time parameter is referred to as t_{HD_SDR}, and in DDR Mode it is referred to as t_{HD_DDR}.

Table 62.73 I3C timing (Push-Pull Timing Parameters for HDR-TSP and HDR-TSL Modes)

Setting of the I3C_SCL0, I3C_SDA0 pins are not required with the Port Drive Capability bit in the PmnPFS register.

Parameter		Symbol	VCC	Min	Max	Unit	Test conditions
I3C Push-Pull Timing Parameters for HDR-TSP and HDR-TSL Modes	Edge-to-Edge Period	t_{EDGE} *1 *2	3.00 V or above	t_{DIG_H}	—	ns	Figure 62.101
			1.65 to 1.95 V	t_{DIG_H}	—		
	Allowed Difference Between Signals for ‘Simultaneous’ Change	t_{SKEW}	3.00 V or above	—	11	ns	
			1.65 to 1.95 V	—	11		
	Stable Condition Between Symbols	t_{EYE}	3.00 V or above	12	—	ns	
			1.65 to 1.95 V	12	—		
	Time Between Successive Symbols	t_{SYMBOL}	3.00 V or above	$t_{EDGE\ Min}$	—	ns	
			1.65 to 1.95 V	$t_{EDGE\ Min}$	—		
	Symbol Clock	t_{CLOCK}	3.00 V or above	1 / f_{SCL} (Max)	—	—	
			1.65 to 1.95 V	1 / f_{SCL} (Max)	—		

Note 1. Edges occur at the rate of 1 / (t_{EDGE} × 2)
Note 2. In a Mixed Bus, HDR-TSL shall respect the maximum $t_{DIG\_H\_MIXED}$ shown in Figure 62.94.

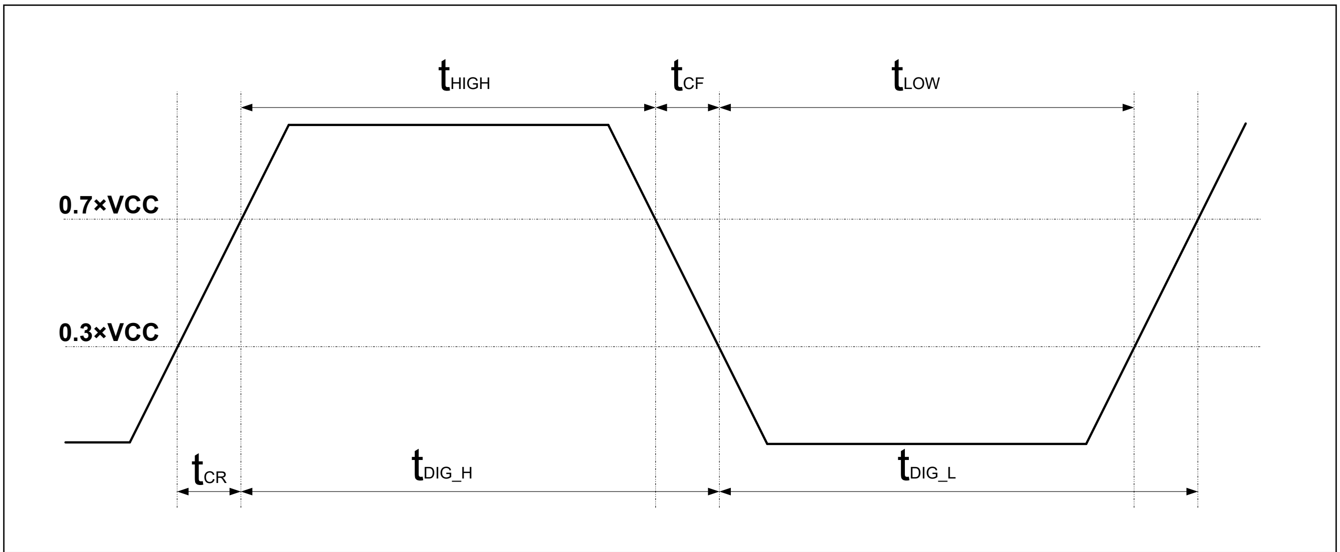
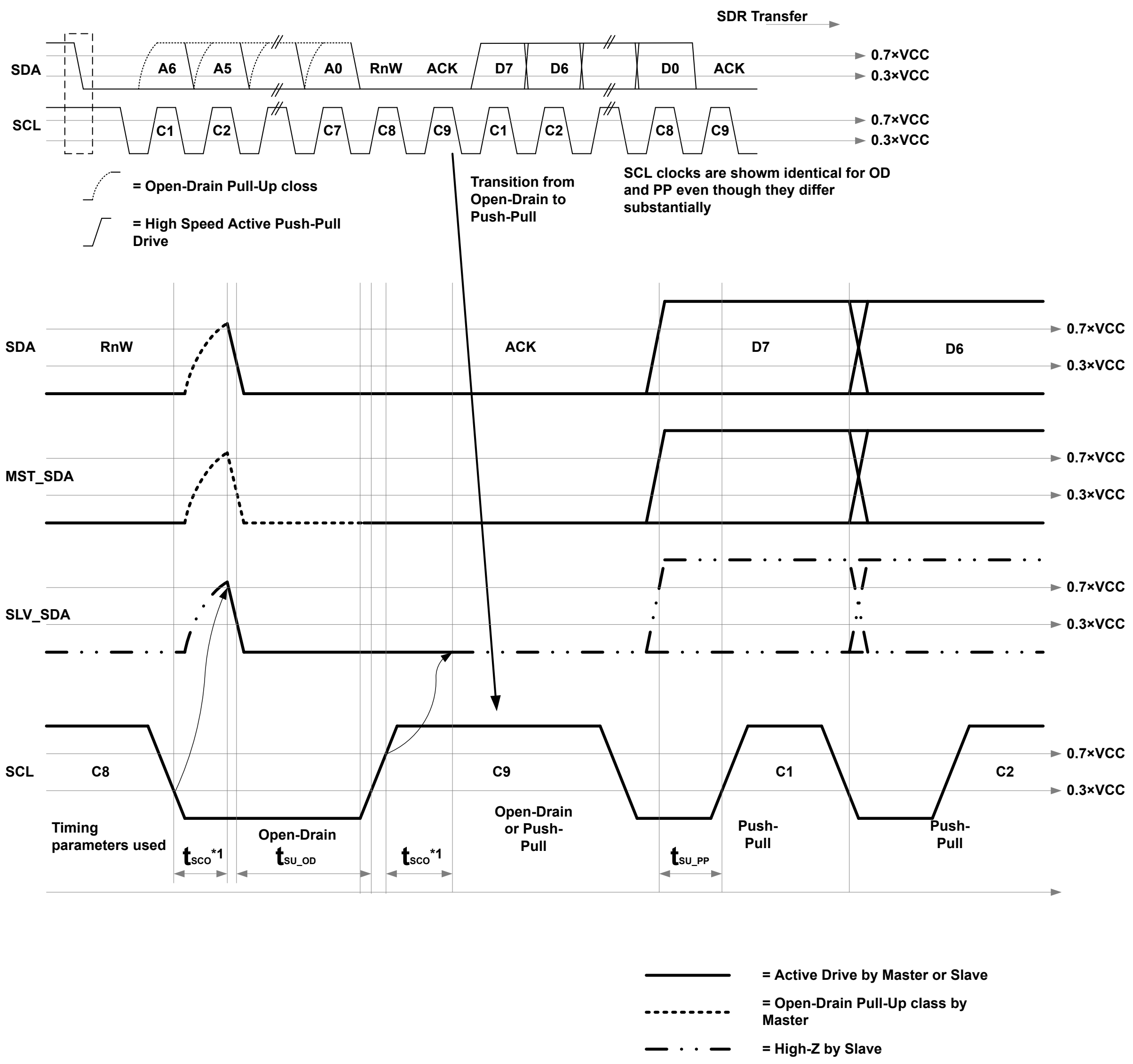


Figure 62.91 t_{DIG_H} and t_{DIG_L}



Note: Open-Drain: Master releases SDA after a suitable delay from SCL edges
Push-Pull: Master drives SDA after a suitable delay from SCL edges

Note 1. t_{SCO} is depicted for informative purposes only

Figure 62.92 I3C Data Transfer – ACK by Slave

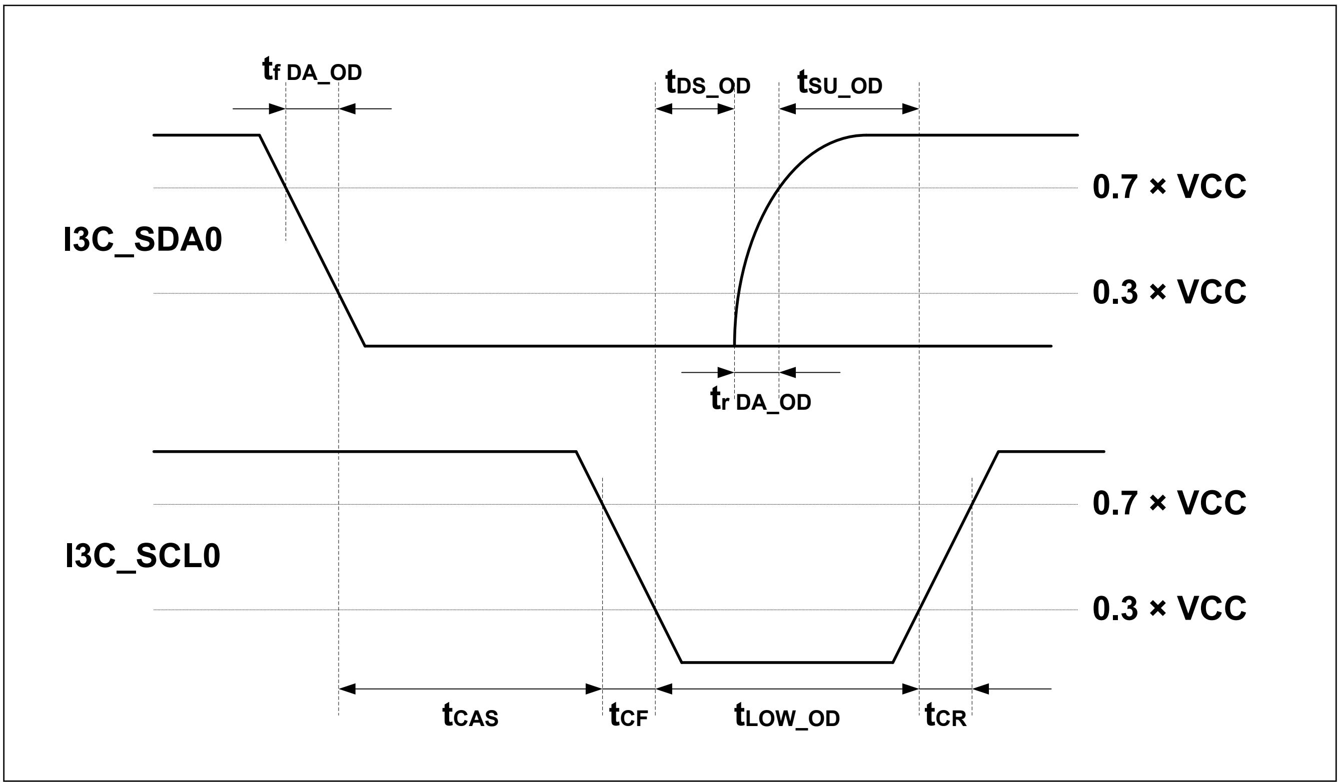


Figure 62.93 I3C START condition Timing

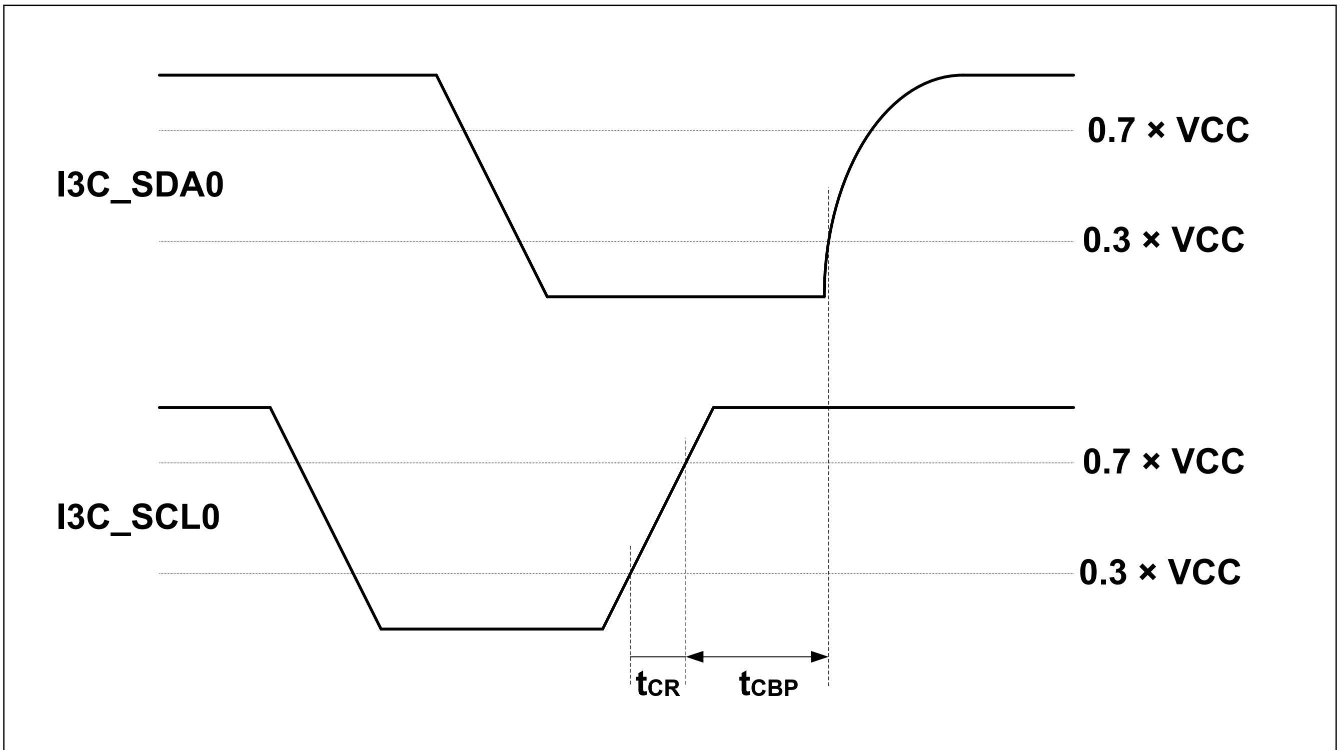


Figure 62.94 I3C STOP condition Timing

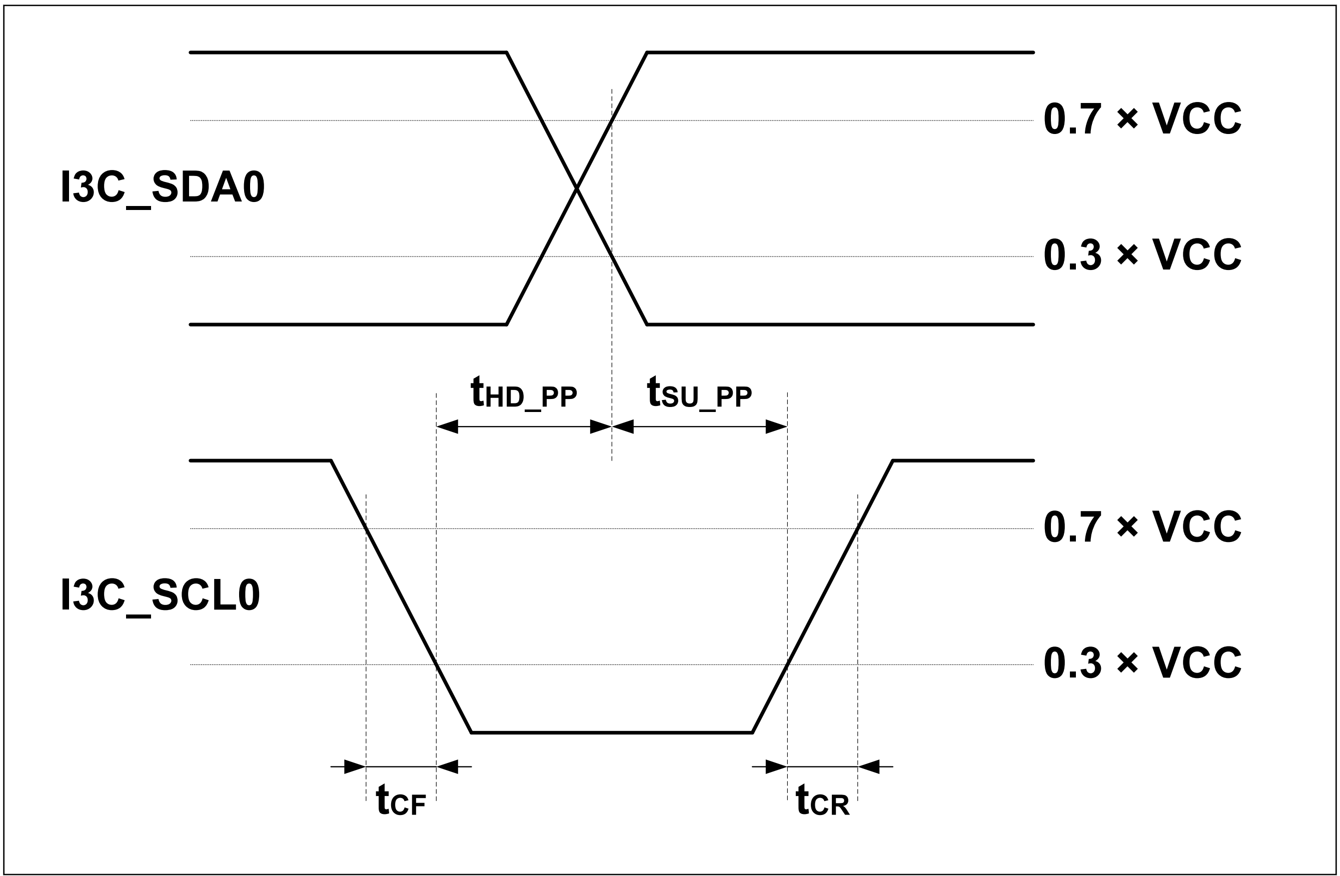


Figure 62.95 I3C Master Out Timing

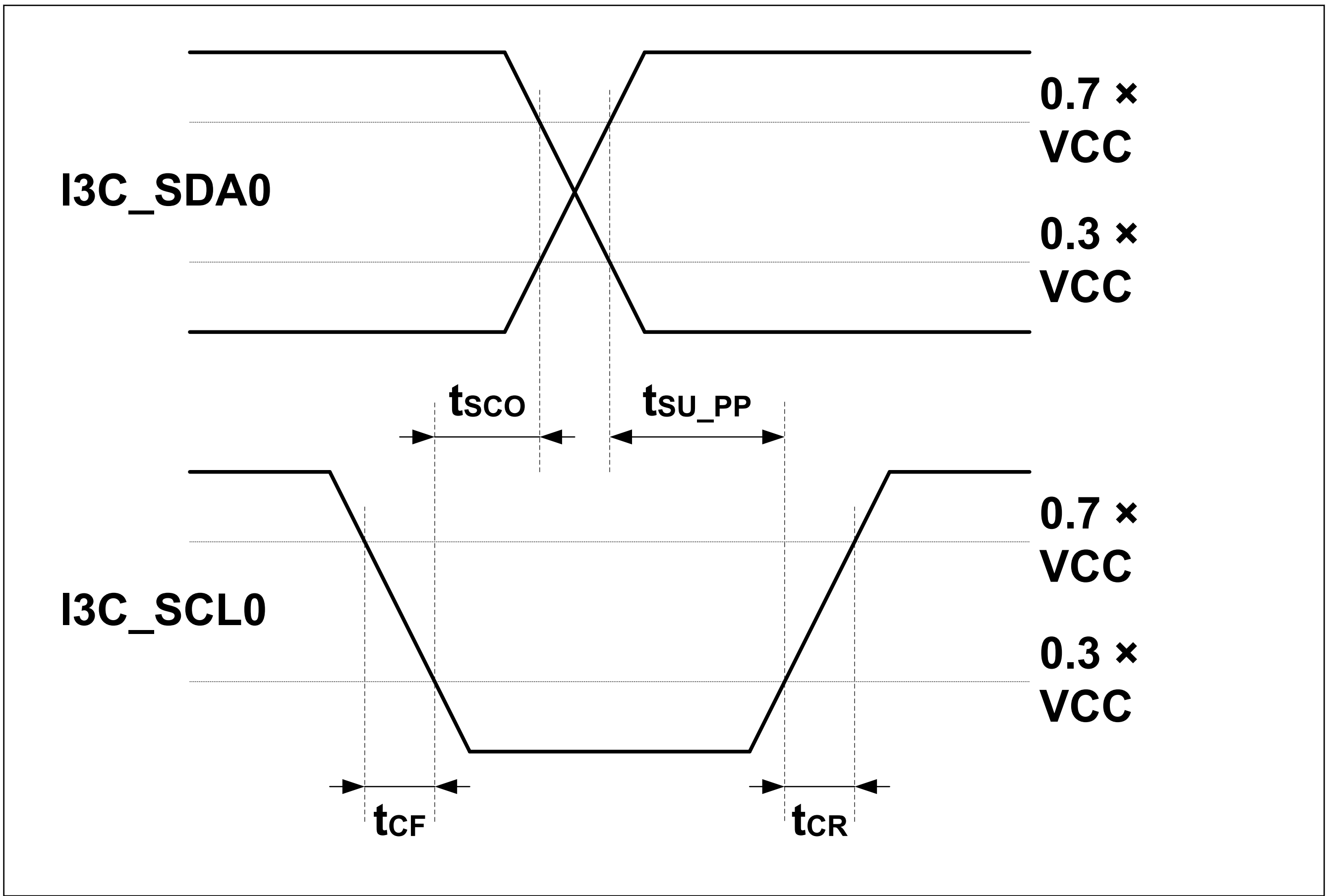


Figure 62.96 I3C Slave Out Timing

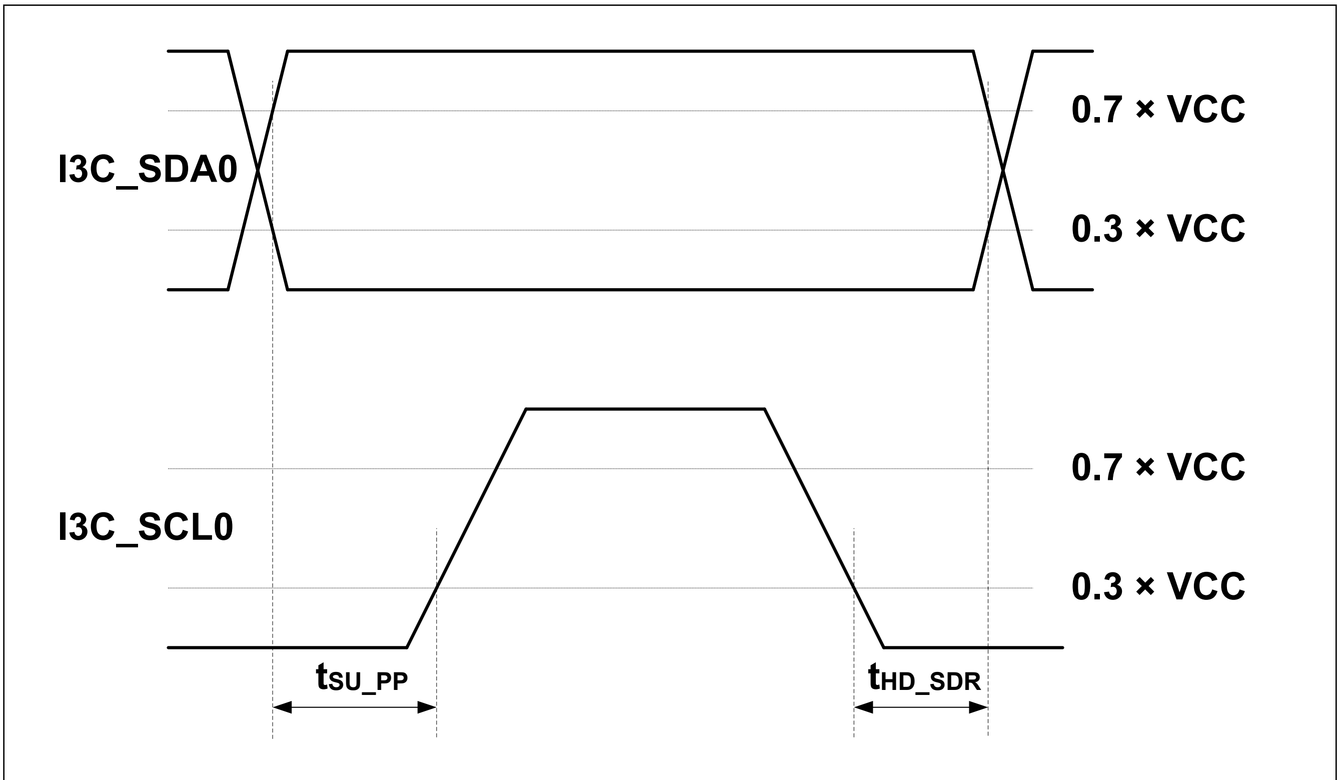


Figure 62.97 Master SDR Timing

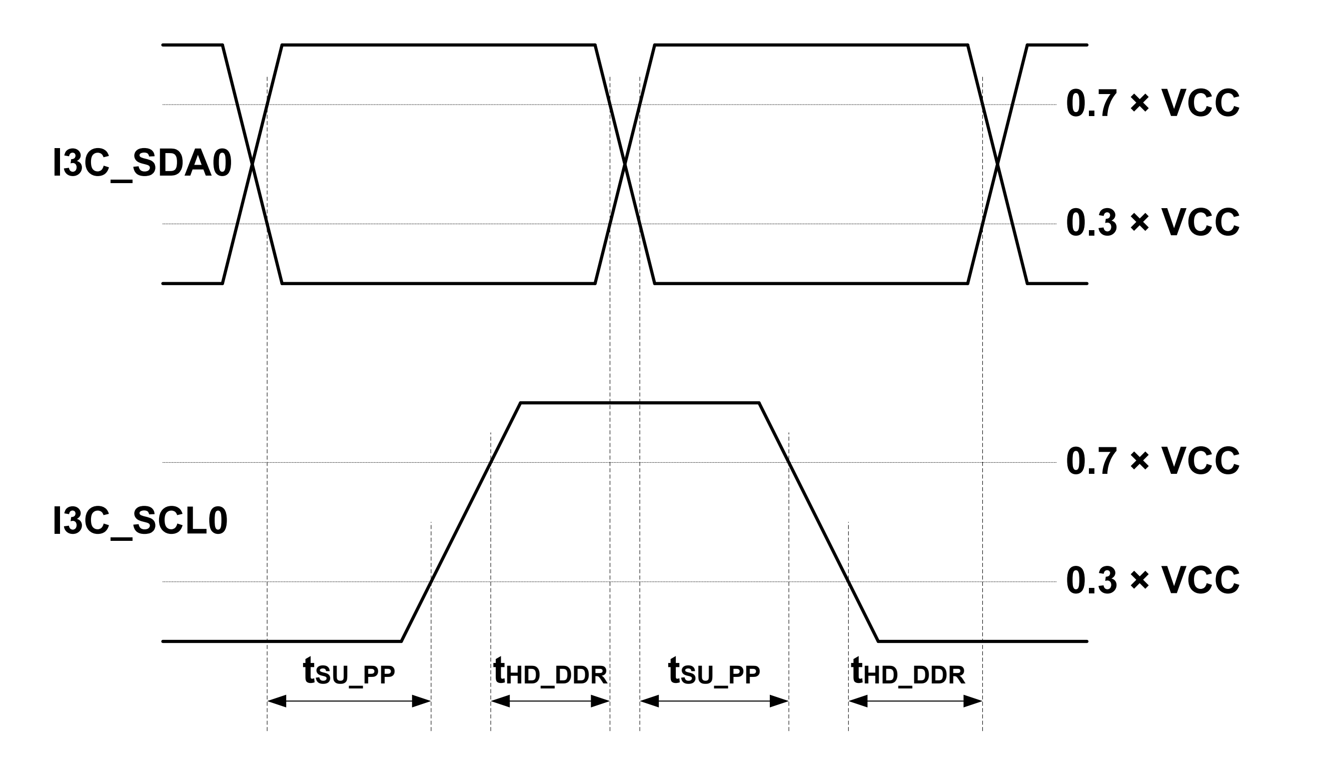
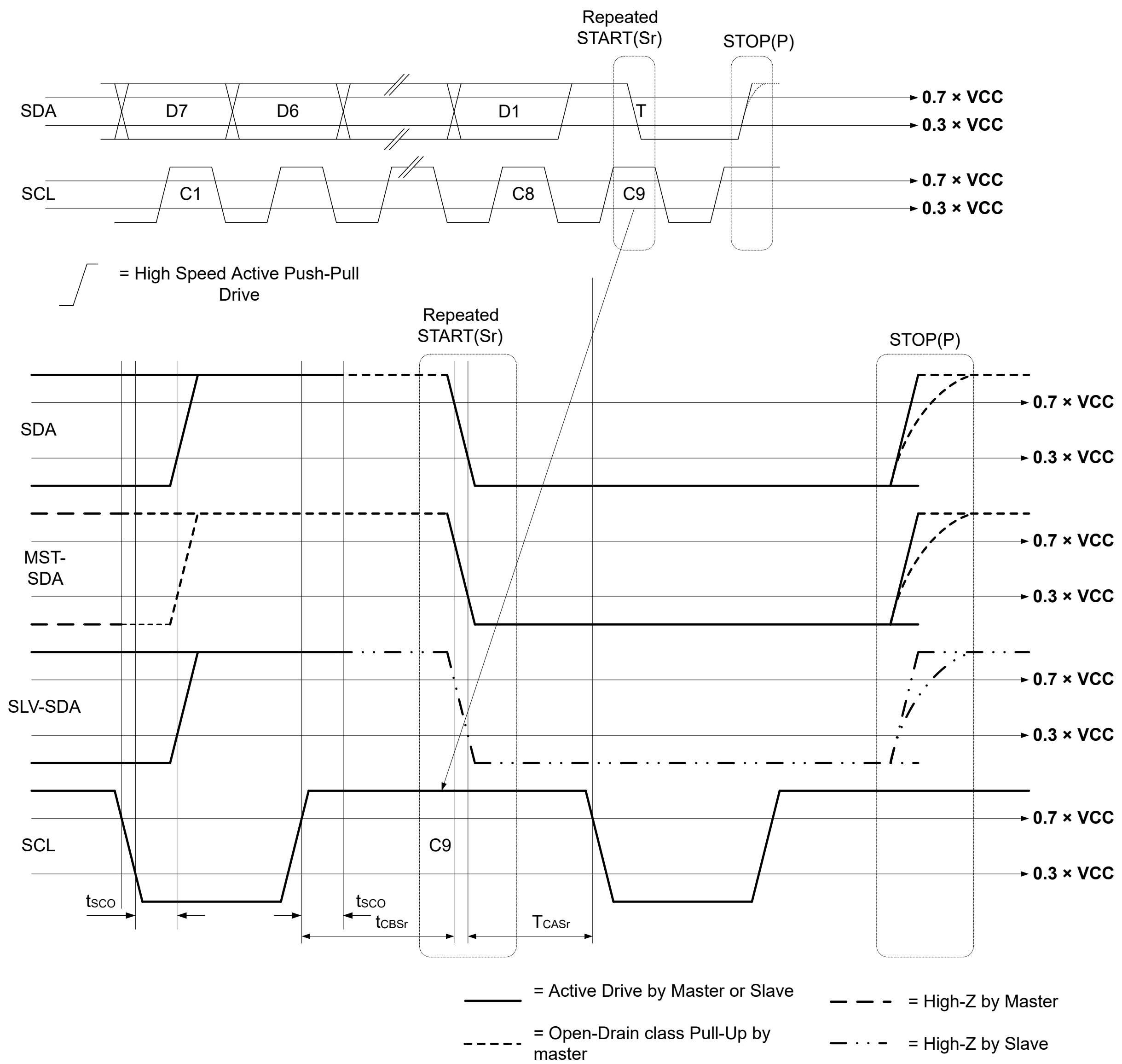


Figure 62.98 Master DDR Timing



Note: Sr immediately followed by P is illegal for I2C devices, however many devices are not adversely affected by it. The condition is not illegal for I3C devices.

Figure 62.99 T-Bit When Master Ends Read with Repeated START and STOP

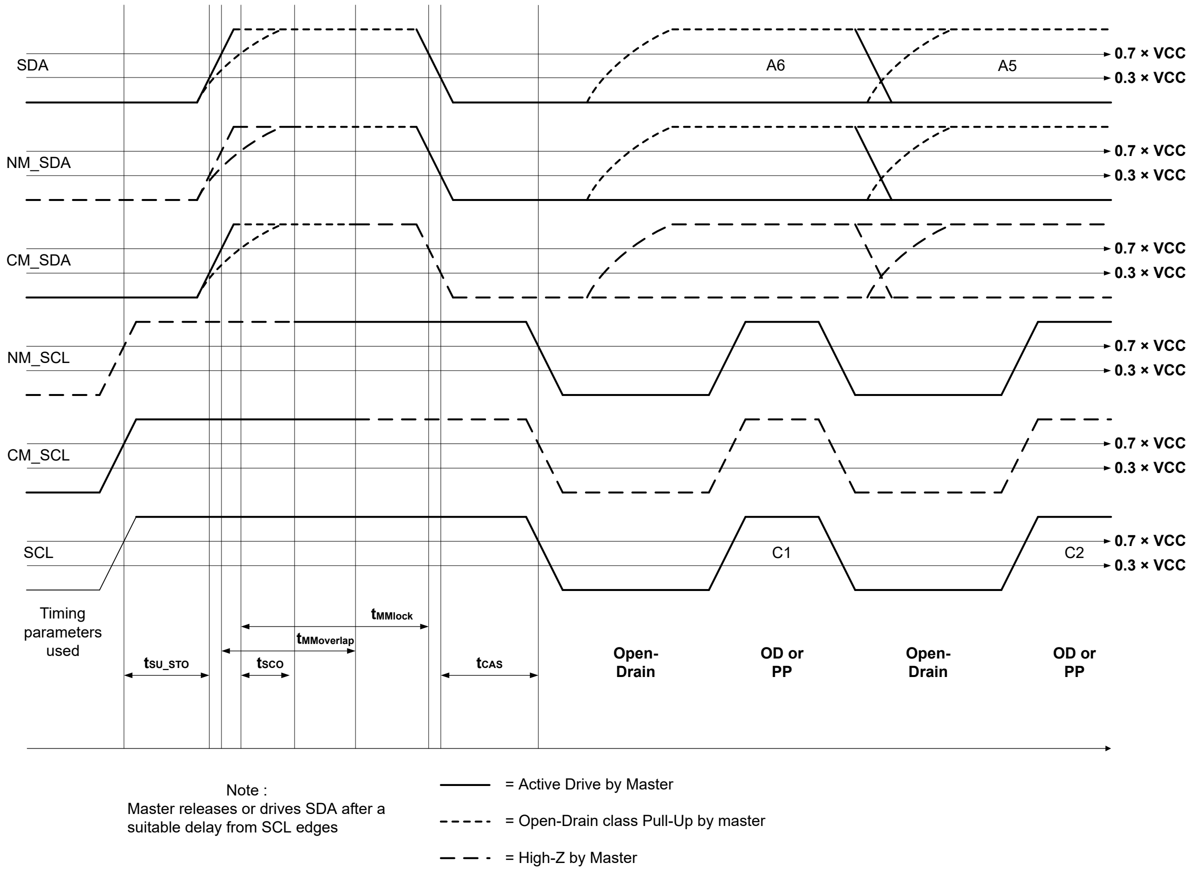
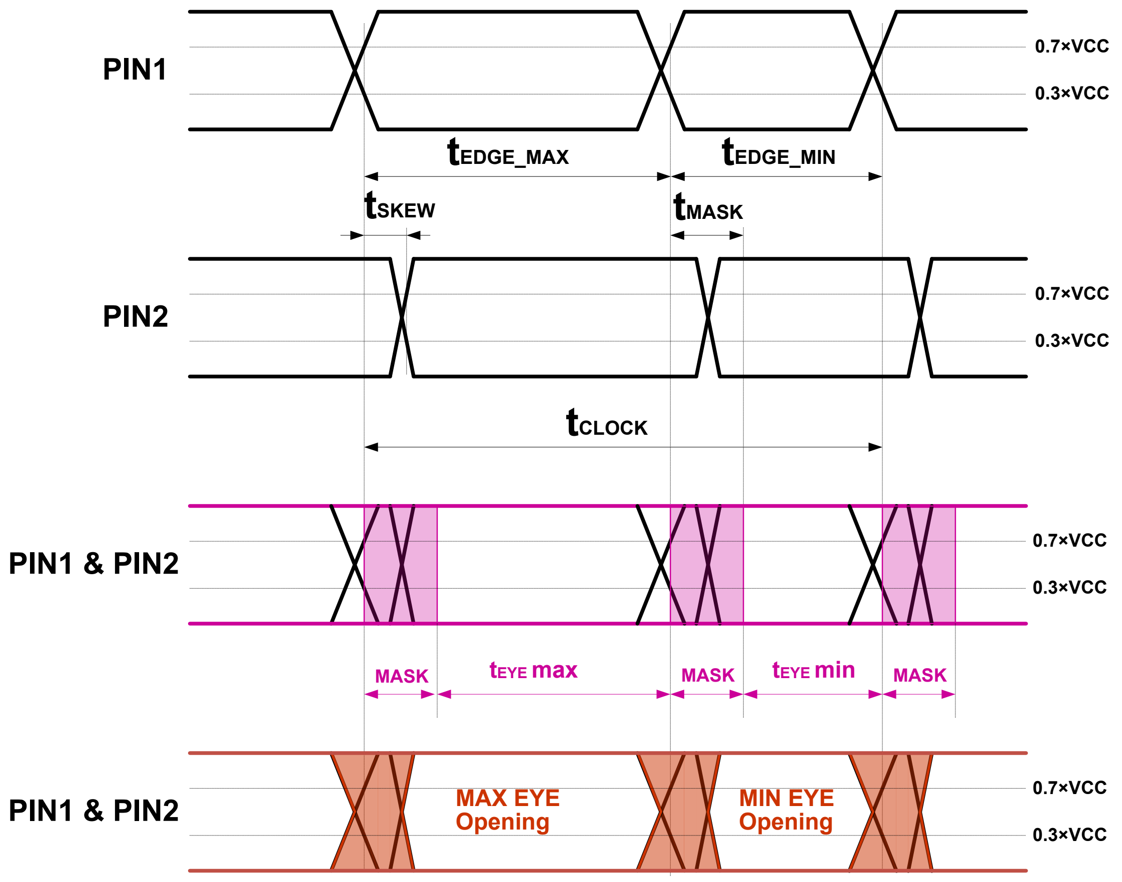


Figure 62.100 Master to Master Bus Handoff



Design Conditions

1. The The differences of load magnitude and drive strength between the SDA and SCL lines result in different slopes, as shown
2. MASK period ≧ MAX t_{SKEW} + MAX rise/fall time between 0.3 × V_{DD} and 0.7 × V_{DD}, respectively, decreased by the time crossing the hysteresis window.

Simplifying Assumptions

1. t_{SKEW} is the same for both edges
2. Base CLOCK (e.g. 12.5MHz) is asymmetrical
3. Falling edges are equal rising edges
4. The MASK is shown as starting, from the last possible moment, whereas in reality it could be shifted left (towards the start point) delayed by the hysteresis window

Figure 62.101 Ternary Protocol Timing

62.3.14 SSIE Timing

Table 62.74 SSIE timing

(1) High drive output is selected with the Port Drive Capability bit in the PmnPFS register.
(2) Use pins that have a letter appended to their names, for instance "_A", "_B" or "_C" to indicate group membership. For the SSIE interface, the AC portion of the electrical characteristics is measured for each group.

Parameter			Symbol	VCC	Min.	Max.	Unit	Comments
SSIBCK	Cycle	Master	t_O	2.70V or above	80	—	ns	Figure 62.102
				1.62V or above	80	—		
		Slave	t_I	2.70V or above	80	—	ns	
				1.62V or above	80	—		
	High level/ low level	Master	t_{HC}/t_{LC}	2.70V or above	0.35	—	t_O	
				1.62V or above	0.35	—		
		Slave		2.70V or above	0.35	—	t_I	
				1.62V or above	0.35	—		
	Rising time/ falling time	Master	t_{RC}/t_{FC}	2.70V or above	—	0.15	t_O / t_I	
				1.62V or above	—	0.15		
		Slave		2.70V or above	—	0.15	t_O / t_I	
				1.62V or above	—	0.15		
SSILRCK/ SSIFS, SSITXD0, SSIRXD0, SSIDATA1	Input set up time	Master	t_{SR}	2.70V or above	12	—	ns	Figure 62.104, Figure 62.105
				1.62V or above	20	—		
		Slave		2.70V or above	12	—	ns	
				1.62V or above	12	—		
	Input hold time	Master	t_{HR}	2.70V or above	8	—	ns	
				1.62V or above	8	—		
		Slave		2.70V or above	15	—	ns	
				1.62V or above	15	—		
	Output delay time	Master	t_{DTR}	2.70V or above	−10	5	ns	
				1.62V or above	−10	7		
		Slave		2.70V or above	0	20	ns	
				1.62V or above	0	25		
	Output delay time from SSILRCK/ SSIFS change	Slave	t_{DTRW}	2.70V or above	—	20	ns	Figure 62.106*1
				1.62V or above	—	25		
GTIOC2A, AUDIO_CLK	Cycle		t_{EXcyc}	2.70V or above	20	—	ns	Figure 62.103
				1.62V or above	40	—		
	High level/ low level		t_{EXL}/t_{EXH}	2.70V or above	0.4	—	t_{EXcyc}	
				1.62V or above	0.4	—		
	Rising time/falling time		t_{EXf}/t_{EXr}	2.70V or above	—	0.1*2	t_{EXcyc}	
				1.62V or above	—	0.1*2		

Note 1. For slave-mode transmission, SSIE has a path, through which the signal input from the SSILRCK/SSIFS pin is used to generate transmit data, and the transmit data is logically output to the SSITXD0 or SSIDATA1 pin.
Note 2. 1μs at the longest.

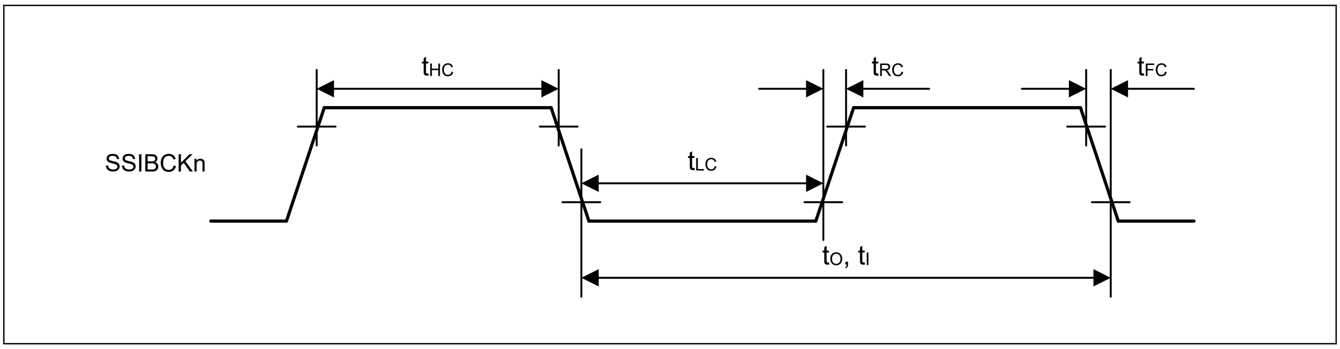


Figure 62.102 SSIE clock input/output timing

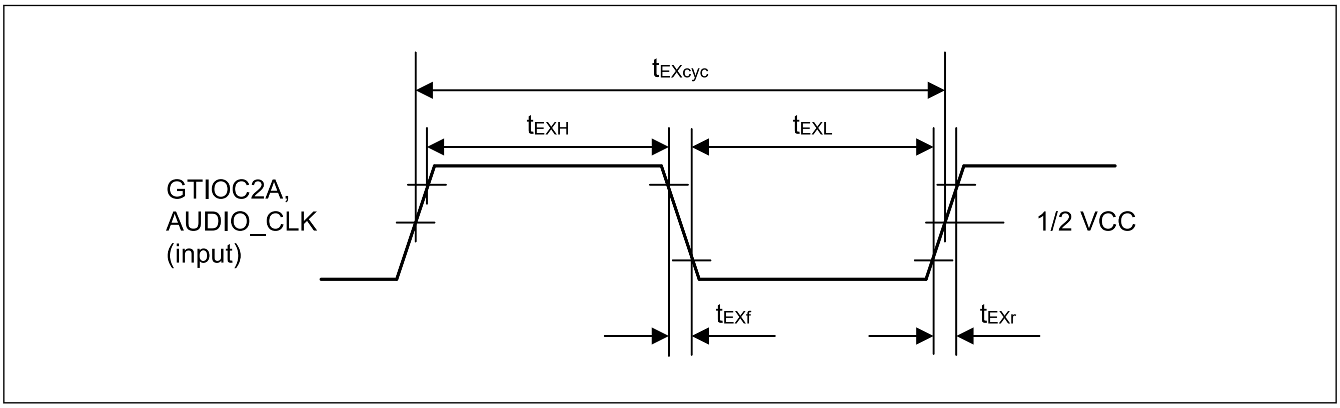


Figure 62.103 Clock input timing

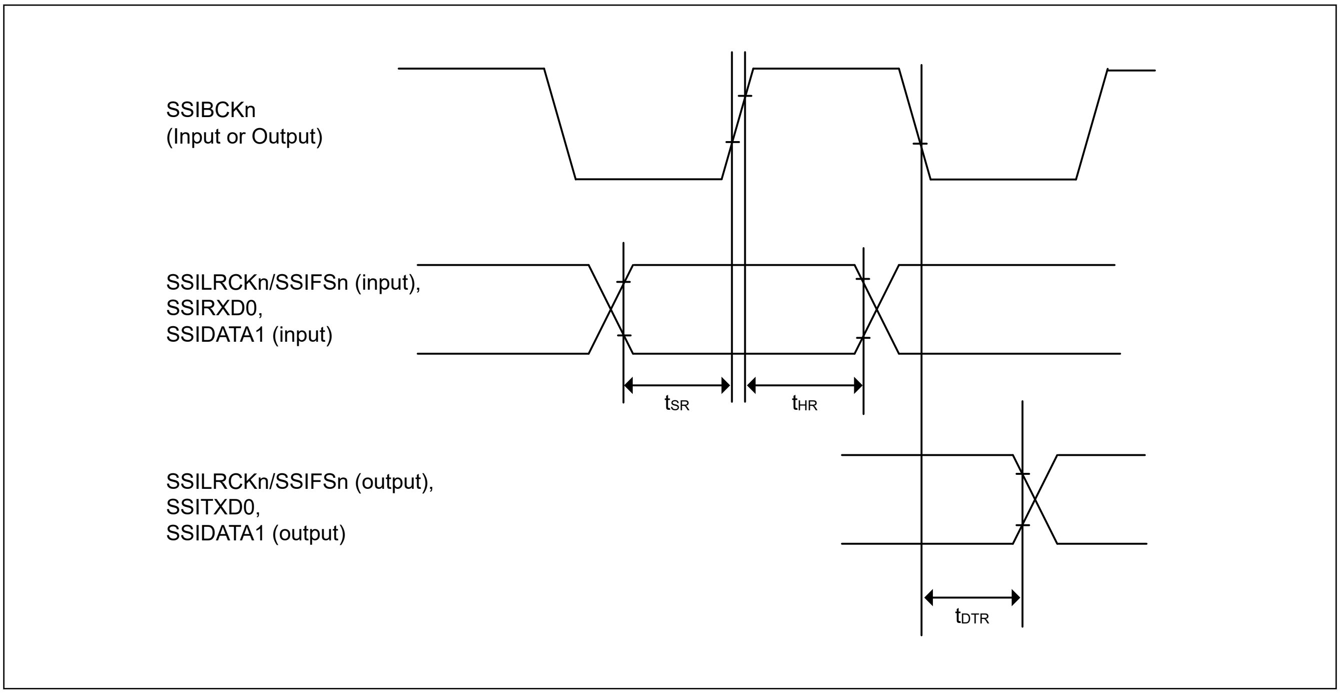


Figure 62.104 SSIE data transmit and receive timing when SSICR.BCKP = 0

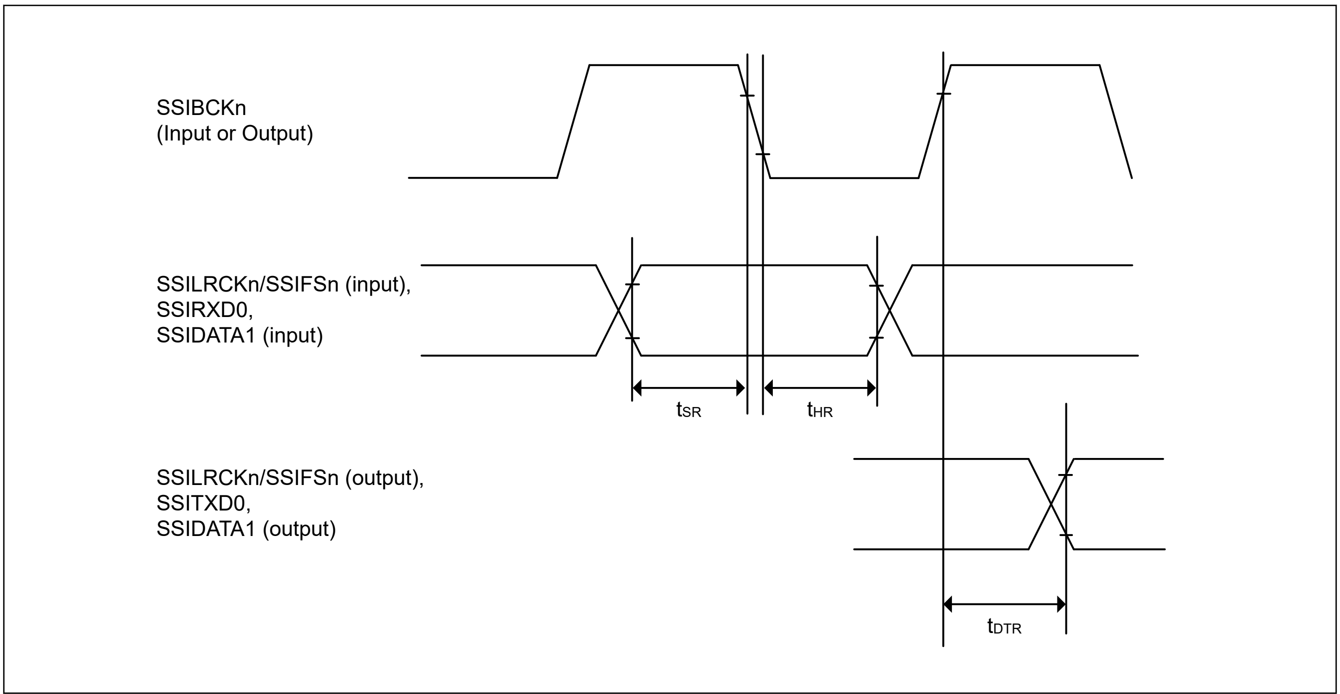


Figure 62.105 SSIE data transmit and receive timing when SSICR.BCKP = 1

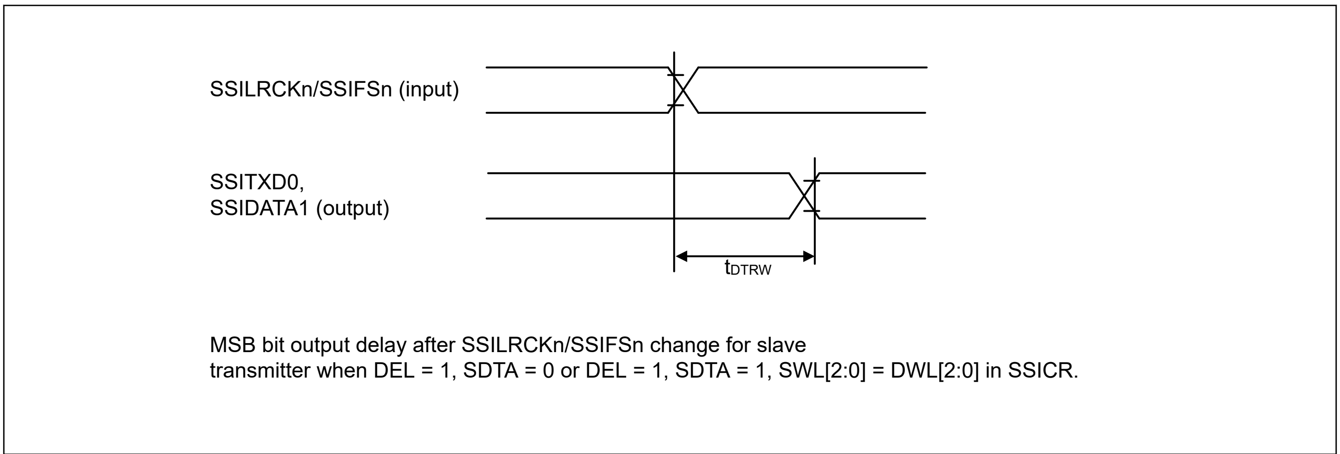


Figure 62.106 SSIE data output delay after SSILRCK0/SSIFS0 change

62.3.15 SD/MMC Host Interface Timing

Table 62.75 SD/MMC Host Interface signal timing

Conditions:
High-speed high drive output is selected in the Port Drive Capability bit in the PmnPFS register for the following pins:
SD0CLK_A, SD0CLK_B, SD0CLK_C, SD1CLK_A, SD1CLK_B
For other pins, high drive output is selected in the Port Drive Capability bit in the PmnPFS register.
Clock duty ratio is 50%.

Parameter	Symbol	VCC/VCC2	Min	Max	Unit	Test conditions
SDCLK clock cycle	t_{SDCYC}	2.70 V or above	20	—	ns	Figure 62.107
		1.70 to1.95 V*1	20	—		
		1.70 to1.95 V	40	—		
SDCLK clock high pulse width	t_{SDWH}	2.70 V or above	6.5	—	ns	
		1.70 to1.95 V*1	6.5	—		
		1.70 to1.95 V	13.0	—		
SDCLK clock low pulse width	t_{SDWL}	2.70 V or above	6.5	—	ns	
		1.70 to1.95 V*1	6.5	—		
		1.70 to1.95 V	13.0	—		
SDCLK clock rise time	t_{SDLH}	2.70 V or above	—	3	ns	
		1.70 to1.95 V*1	—	4		
		1.70 to1.95 V	—	8		
SDCLK clock fall time	t_{SDHL}	2.70 V or above	—	3	ns	
		1.70 to1.95 V*1	—	4		
		1.70 to1.95 V	—	8		
SDCMD/SDDAT output data delay	t_{SDODLY}	2.70 V or above	–7.0	4.0	ns	
		1.70 to1.95 V*1	–7.0	7.0		
		1.70 to1.95 V	–15.0	15.0		
SDCMD/SDDAT input data setup	t_{SDIS}	2.70 V or above	4.5	—	ns	
		1.70 to1.95 V*1	4.5	—		
		1.70 to1.95 V	20.0	—		
SDCMD/SDDAT input data hold	t_{SDIH}	2.70 V or above	1.5	—	ns	
		1.70 to1.95 V	1.5	—		

Note: Must use pins that have a letter appended to their name, for instance "_A", "_B", to indicate group membership. For the SD/MMC Host interface, the AC portion of the electrical characteristics is measured for each group.
Note: If SD1DAT4_A to SD1DAT7_A are used, characteristics above is guaranteed only when VCC = VCC2.
Note 1. Only supported for Ch0 group B ("SD0*_B") and Ch1 group A ("SD1*_A")

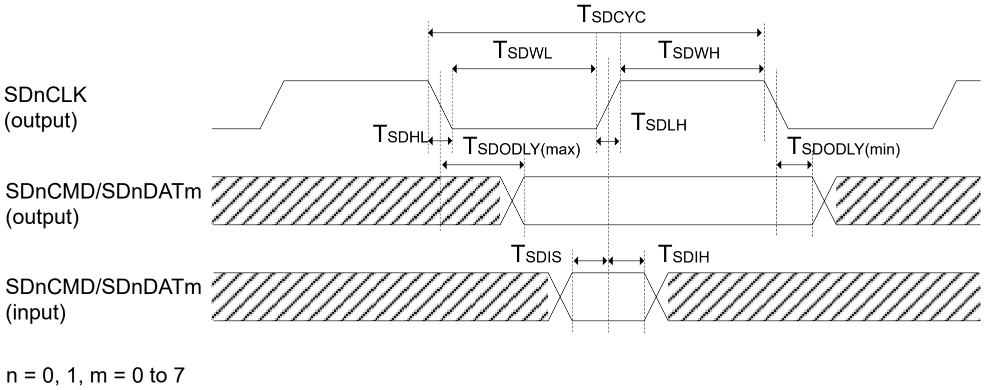


Figure 62.107 SD/MMC Host Interface signal timing

62.3.16 CEU Timing

Table 62.76 Capture Engine Unit Signal Timing

Parameter	Symbol	VCC	Min	Max	Unit	Test conditions
Vertical sync (VIO_VD) setup time (Camera clock rising)	t_{VVDS}	2.70 V or above	2.0	—	ns	Figure 62.108 Figure 62.109
		1.62 V or above	4.5	—		
Vertical sync (VIO_VD) setup time (Camera clock falling)	t_{VVDS}	2.70 V or above	2.5	—	ns	
		1.62 V or above	4.5	—		
Vertical sync (VIO_VD) hold time	t_{VVDH}	2.70 V or above	3.5	—	ns	
		1.62 V or above	5.5	—		
Horizontal sync (VIO_HD) setup time (Camera clock rising)	t_{VHDS}	2.70 V or above	2.0	—	ns	
		1.62 V or above	4.5	—		
Horizontal sync (VIO_HD) setup time (Camera clock falling)	t_{VHDS}	2.70 V or above	2.5	—	ns	
		1.62 V or above	4.5	—		
Horizontal sync (VIO_HD) hold time	t_{VHDH}	2.70 V or above	3.5	—	ns	
		1.62 V or above	5.5	—		
Capture image data (VIO_D) setup time (Camera clock rising)	t_{VDTS}	2.70 V or above	2.0	—	ns	
		1.62 V or above	4.5	—		
Capture image data (VIO_D) setup time (Camera clock falling)	t_{VDTS}	2.70 V or above	2.5	—	ns	
		1.62 V or above	4.5	—		
Capture image data (VIO_D) hold time	t_{VDTH}	2.70 V or above	3.5	—	ns	
		1.62 V or above	5.5	—		
Camera clock cycle	t_{VCYC}	2.70 V or above	11.5	—	ns	
		1.62 V or above	23.0	—		
Camera clock high level width	t_{VHW}	2.70 V or above	$0.4 \times t_{VCYC}$	—	ns	
		1.62 V or above	$0.4 \times t_{VCYC}$	—		
Camera clock low level width	t_{VLW}	2.70 V or above	$0.4 \times t_{VCYC}$	—	ns	
		1.62 V or above	$0.4 \times t_{VCYC}$	—		
Field identification signal (VIO_FLD) setup time (Camera clock rising)	t_{VFDS}	2.70 V or above	2.0	—	ns	
		1.62 V or above	4.5	—		
Field identification signal (VIO_FLD) setup time (Camera clock falling)	t_{VFDS}	2.70 V or above	2.5	—	ns	
		1.62 V or above	4.5	—		
Field identification signal (VIO_FLD) hold time	t_{VFDH}	2.70 V or above	3.5	—	ns	
		1.62 V or above	5.5	—		

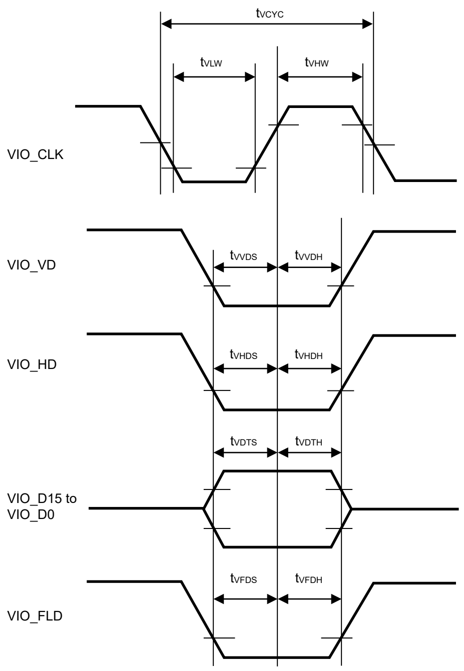


Figure 62.108 Capture Engine Unit Module Signal Timing of data capturing on the rising edge of VIO_CLK

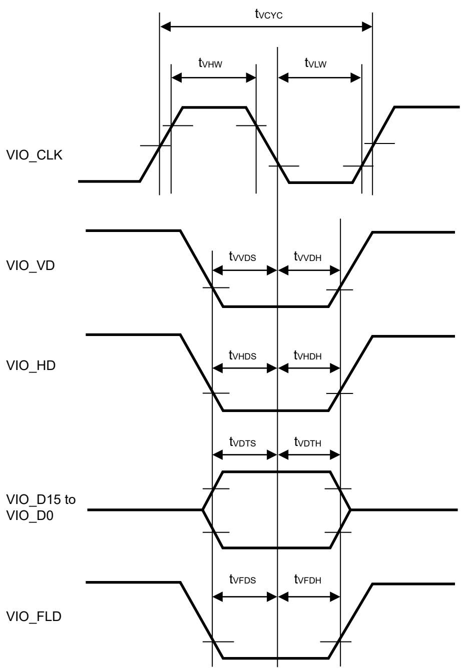


Figure 62.109 Capture Engine Unit Module Signal Timing of data capturing on the falling edge of VIO_CLK

62.3.17 CANFD Timing

Table 62.77 CANFD interface timing

Conditions: Low drive output is selected when VCC is 2.70V or above. Middle drive output is selected when VCC is 1.62V or above.

Parameter	Symbol	VCC/VCC2	Min	Max	Unit	Test conditions
Internal delay time	t_{node}	2.70 V or above	—	50	ns	Figure 62.110
		1.62 V or above	—	50		
Transmission rate		2.70 V or above	—	8	Mbps	
		1.62 V or above	—	8		

Note: Internal delay time (t_{node}) = Internal transfer delay time (t_{output}) + Internal receive delay time (t_{input})

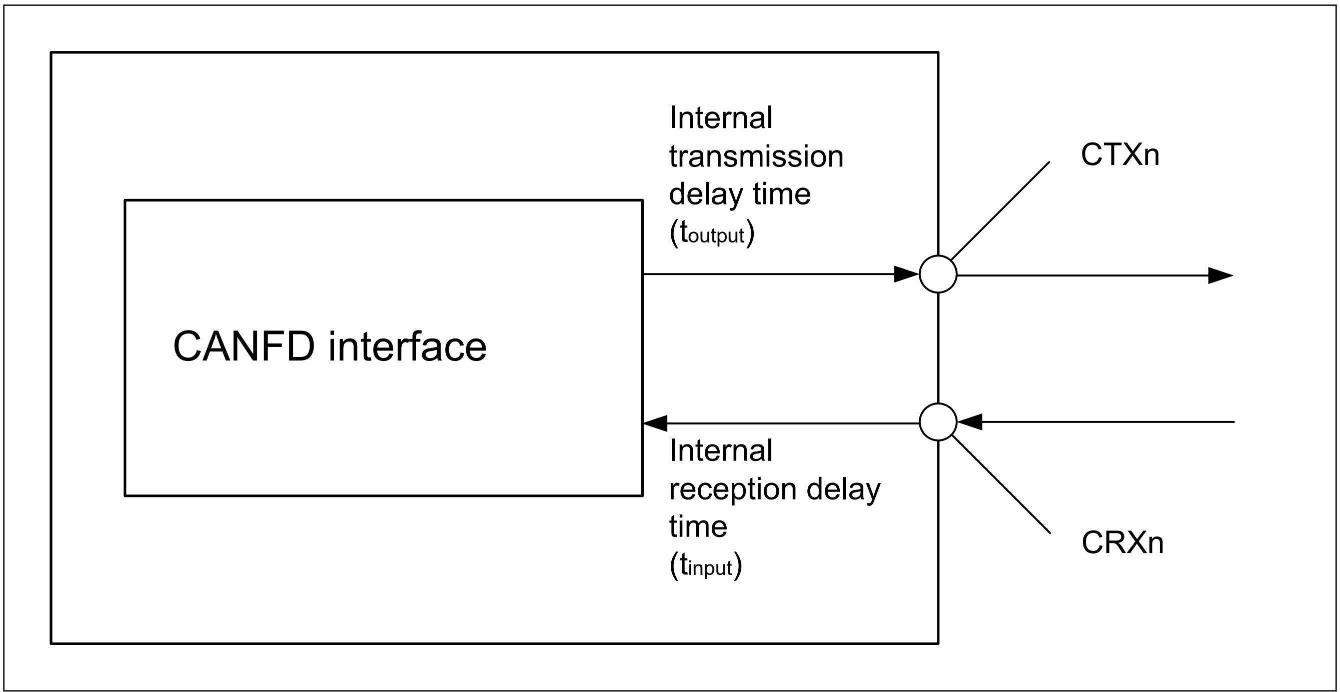


Figure 62.110 CANFD interface condition

62.3.18 PDG Timing

Table 62.78 PDG Timing

Parameter	Min	Typ	Max	Unit	Test conditions
Operation frequency	80	—	300	MHz	—
Resolution	—	48.8	—	ps	GTDLYCR.FRANGE [1:0] = 00 and GPTCLK = 160 MHz
	—	52.1	—	ps	GTDLYCR.FRANGE [1:0] = 01 and GPTCLK = 300 MHz
DNL[*1]	—	±2.0	—	LSB	—

Note 1. This value normalizes the differences between lines in 1-LSB resolution.

62.3.19 ESWM Timing

Table 62.79 ESWM timing (RMII)

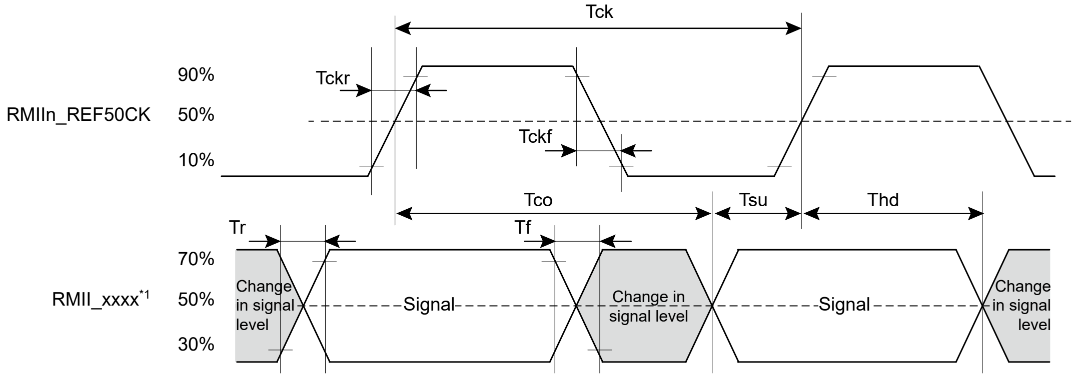
Conditions:

1. Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register for the following pins: ETn_MDC, ETn_MDIO.
2. The drive capacity selection and voltage range for pins other than ETn_MDC, ETn_MDIO.

- RMII use: High drive select, VCC = 2.7 to 3.6

Parameter		Symbol	VCC	Min	Max	Unit	Test conditions
ESWM (RMII)	RMIIn_REF50CK cycle time	Tck	2.70V or above	20	—	ns	Figure 62.112
	RMIIn_REF50CK frequency, typical 50 MHz	—		—	50 + 100 ppm	MHz	
	RMIIn_REF50CK duty	—		35	65	%	
	RMIIn_REF50CK rise/fall time	Tckr/ckf		0.5	3.5	ns	
	RMIIn_xxxx*1 output delay	Tco		2.5	12	ns	
	RMIIn_xxxx*2 setup time	Tsu		3	—	ns	
	RMIIn_xxxx*2 hold time	Thd		1	—	ns	
	RMIIn_xxxx*1 *2 rise/fall time	Tr/Tf		0.5	4	ns	

Note 1. RMIIn_TX_EN, RMIIn_TXD1, RMIIn_TXD0.
Note 2. RMIIn_CRS_DV, RMIIn_RXD1, RMIIn_RXD0, RMIIn_RX_ER.



Note 1. RMIIn_TX_EN, RMIIn_TXD1, RMIIn_TXD0, RMIIn_CRS_DV, RMIIn_RXD1, RMIIn_RXD0, RMIIn_RX_ER

Figure 62.112 RMIIn_REF50CK and RMII signal timing

Table 62.80 ESWM timing (MII)

Conditions:

1. Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register for the following pins: ETn_MDC, ETn_MDIO.
2. The drive capacity selection and voltage range for pins other than ETn_MDC, ETn_MDIO.

- MII use only: Middle drive select, VCC = 2.7 to 3.6
- GMII and MII use: RGMII 2.50 V drive select, VCC = 2.3 to 2.7
- GMII and MII use: RGMII 3.30 V drive select, VCC = 3.0 to 3.6

Parameter		Symbol	VCC	Min	Max	Unit	Test conditions
ESWM (MII)	ETn_TX_CLK cycle time	t_{Tcyc}	2.30 V or above	40	—	ns	—
	ETn_TX_EN output delay	t_{TENd}		1	20	ns	Figure 62.113
	ETn_ETXD0 to ETn_ETXD3 output delay	t_{MTDd}		1	20	ns	
	ETn_RX_CLK cycle time	t_{TRcyc}		40	—	ns	—
	ETn_RX_DV setup time	t_{RDVs}		10	—	ns	Figure 62.114
	ETn_RX_DV hold time	t_{RDVh}		10	—	ns	
	ETn_ERXD0 to ETn_ERXD3 setup time	t_{MRDs}		10	—	ns	
	ETn_ERXD0 to ETn_ERXD3 hold time	t_{MRDh}		10	—	ns	
	ETn_RX_ER setup time	t_{RERs}		10	—	ns	
	ETn_RX_ER hold time	t_{RESh}		10	—	ns	

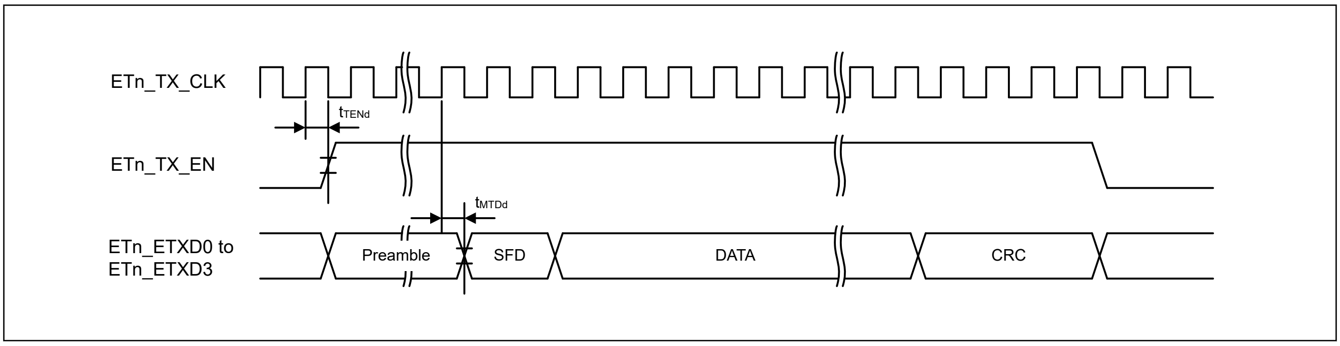


Figure 62.113 MII transmission timing in normal operation

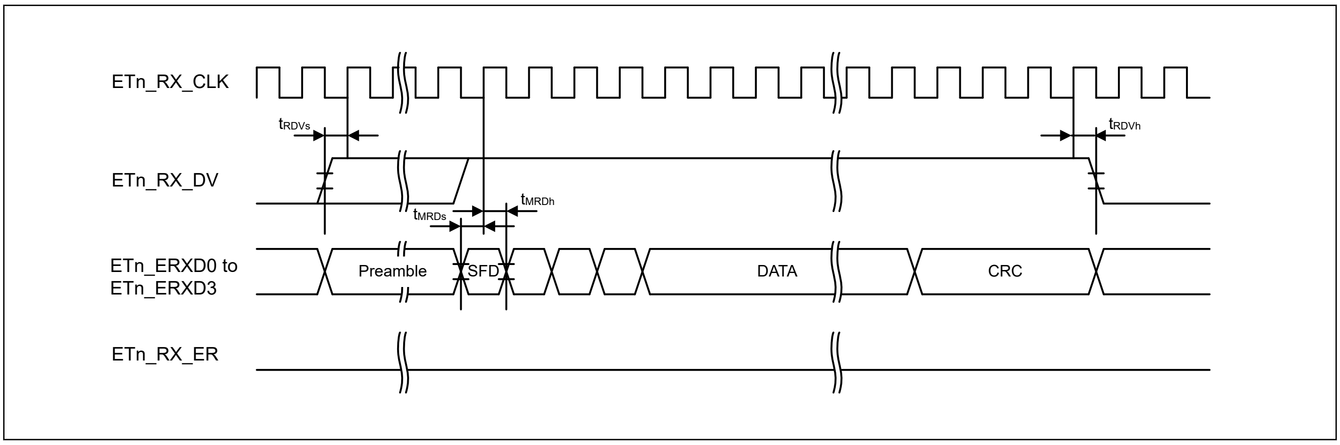


Figure 62.114 MII reception timing in normal operation

Table 62.81 ESWM timing (GMII)

Conditions:

1. Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register for the following pins: ETn_MDC, ETn_MDIO.
2. The drive capacity selection and voltage range for pins other than ETn_MDC, ETn_MDIO.

- GMII use at 2.5 V: RGMII 2.50V drive select, VCC = 2.3 to 2.7
- GMII use at 3.3 V: RGMII 3.30V drive select, VCC = 3.0 to 3.6

Parameter		Symbol	VCC	Min	Max	Unit	Test conditions
ESWM (GMII)	ETn_GTX_CLK Frequency	t_{FREQ}	2.30 V to 3.60 V	125 - 100 ppm	125 + 100 ppm	MHz	Figure 62.115 Figure 62.116
	ETn_GTX_CLK Period	t_{PERIOD}		7.5	8.5	ns	
	ETn_RX_CLK Period	t_{PERIOD}		7.5	—	ns	
	ETn_GTX_CLK, ETn_RX_CLK Time High	t_{HIGH}		2.5	—	ns	
	ETn_GTX_CLK, ETn_RX_CLK Time Low	t_{LOW}		2.5	—	ns	
	ETn_GTX_CLK, ETn_RX_CLK Rise Time	t_R		—	1	ns	
	ETn_GTX_CLK, ETn_RX_CLK Fall Time	t_F		—	1	ns	
	Magnitude of ETn_GTX_CLK, ETn_RX_CLK Slew Rate (rising)*1	—		0.6	—	V/ns	
	Magnitude of ETn_GTX_CLK, ETn_RX_CLK Slew Rate (falling)*1	—		0.6	—	V/ns	
	ETn_TXD, ETn_TX_EN, ETn_TX_ER Setup to ↑ETn_GTX_CLK and ETn_RXD, ETn_RX_DV, ETn_RX_ER Setup to ↑ ETn_RX_CLK	t_{SETUP}		2.5	—	ns	
	ETn_TXD, ETn_TX_EN, ETn_TX_ER Hold from ↑ ETn_GTX_CLK and ETn_RXD, ETn_RX_DV, ETn_RX_ER Hold from ↑ ETn_RX_CLK	t_{HOLD}		0.5	—	ns	
	ETn_TXD, ETn_TX_EN, ETn_TX_ER Setup to ↑ ETn_GTX_CLK and ETn_RXD, ETn_RX_DV, ETn_RX_ER Setup to ↑ ETn_RX_CLK	t_{SETUP}(RCVR)		2	—	ns	
	TXD, TX_EN, TX_ER, ETn_TX_ER Hold from ↑ GTX_CLK and RXD, RX_DV, RX_ER Hold from ↑ RX_CLK	t_{HOLD}(RCVR)		0	—	ns	

Note 1. Clock Skew rate is the instantaneous rate of change of the clock potential with respect to time (dV/dt), not an average value over the entire rise or fall time interval. Conformance with this specification guarantees that the clock signals will rise and fall monotonically through the switching region.

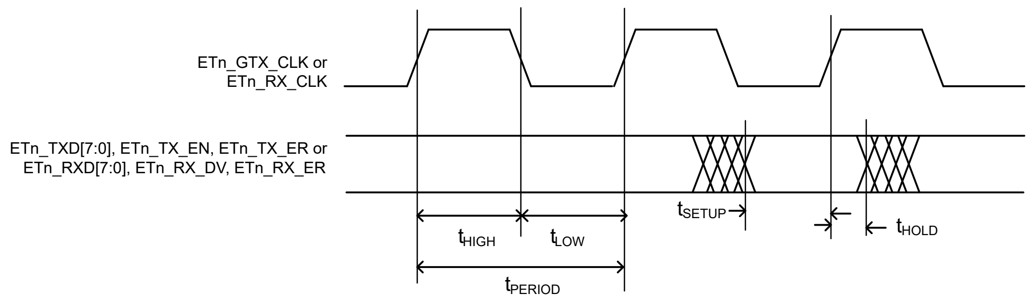


Figure 62.115 GMII timing

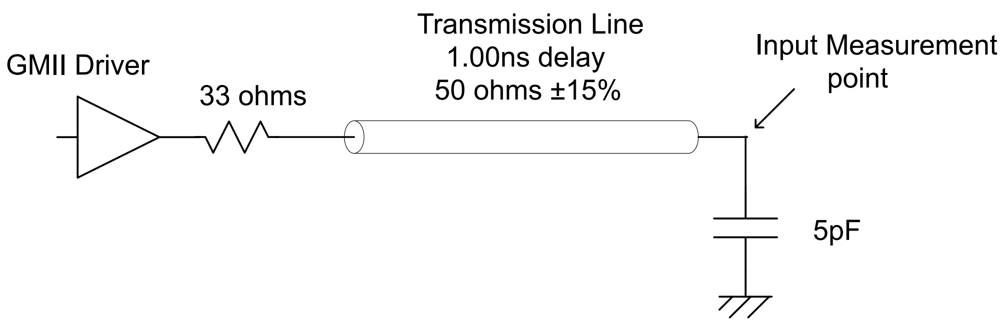


Figure 62.116 GMII output timing measurement conditions

Table 62.82 ESWM timing (RGMII)

Conditions:

1. Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register for the following pins: ETn_MDC, ETn_MDIO.
2. The drive capacity selection and voltage range for pins other than ETn_MDC, ETn_MDIO.

- RGMII use at 2.5 V: RGMII 2.50V drive select, VCC = 2.3 to 2.7
- RGMII use at 3.3 V: RGMII 3.30V drive select, VCC = 3.0 to 3.6

Parameter		Symbol	VCC	Min	Max	Unit	Test conditions
ESWM (RGMII)	Data to Clock output Skew (at Transmitter)*1	T_{skewT}	2.30 V to 3.60 V	−500	500	ps	Figure 62.117 Figure 62.118 Figure 62.119
	Data to Clock input Skew (at Receiver)*1	T_{skewR}		1	2.6	ns	
	Data to Clock output Setup (at Transmitter-integrated delay)	T_{setupT}		1.2	—	ns	
	Clock to Data output Hold (at Transmitter–integrated delay)	T_{holdT}		1.2	—	ns	
	Data to Clock input Setup (at Receiver–integrated delay)	T_{setupR}		1	—	ns	
	Data to Clock input Hold (at Receiver–integrated delay)	T_{holdR}		1	—	ns	
	Clock Cycle Duration*2	T_{cyc}		7.2	8.8	ns	
	Duty Cycle for Gigabit*3	Duty_G		45	55	%	
	Duty Cycle for 10/100T*3	Duty_T		40	60	%	
	Rise / Fall Time (20-80%)	T_r / T_f		—	0.75	ns	

Note 1. This implies that PC board design will require clocks to be routed such that an additional trace delay of greater than 1.5ns and less than 2.0ns will be added to the associated clock signal.
Note 2. For 10Mbps and 100Mbps, Tcyc will scale to 400ns+-40ns and 40ns+-4ns respectively.
Note 3. Duty cycle may be stretched/shrunk during speed changes or while transitioning to a received packet's clock domain as long as minimum duty cycle is not violated and stretching occurs for no more than three Tcyc of the lowest speed transitioned between.

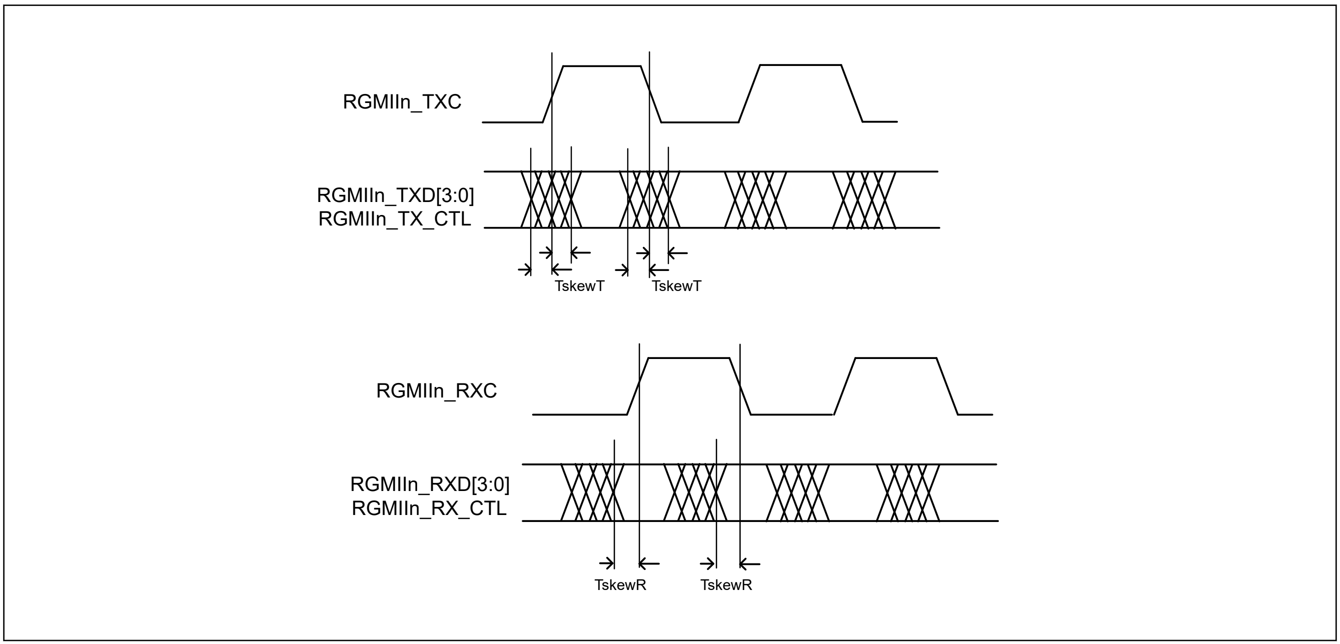


Figure 62.117 RGMII timing

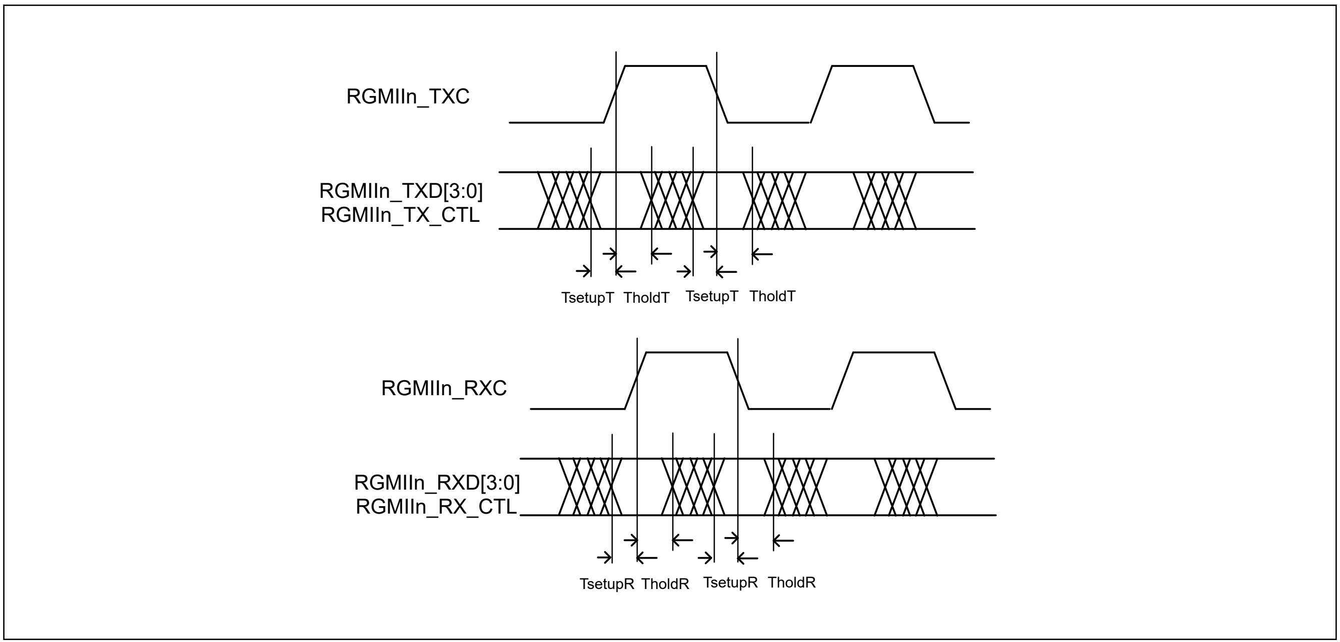


Figure 62.118 RGMII timing (RGMII-ID)

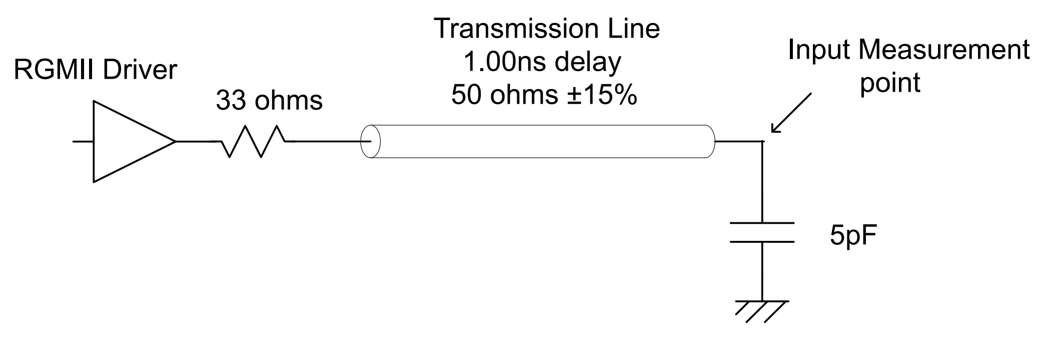


Figure 62.119 RGMII output timing measurement conditions

Table 62.83 ESWM timing (ETn_MDIO, ETn_MDC)

Conditions:

1. Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter		Symbol	VCC	Min	Max	Unit	Test conditions
ESWM (ETn_MDIO, ETn_MDC)	ETHn_MDC output cycle	t_{MDC}	2.70 V or above	80	—	ns	Figure 62.120
			2.30 V or above	160	—	ns	
	ETHn_MDIO setup time (relative to ETHn_MDC↑)	t_{SMDIO}	2.70 V or above	20	—	ns	
			2.30 V or above	40	—	ns	
	ETHn_MDIO hold time (relative to ETHn_MDC↑)	t_{HMDIO}	2.70 V or above	0	—	ns	
			2.30 V or above	0	—	ns	
	ETHn_MDIO output delay time (relative to ETHn_MDC↑)	t_{DMDIO}	2.70 V or above	0	20	ns	
			2.30 V or above	0	40	ns	

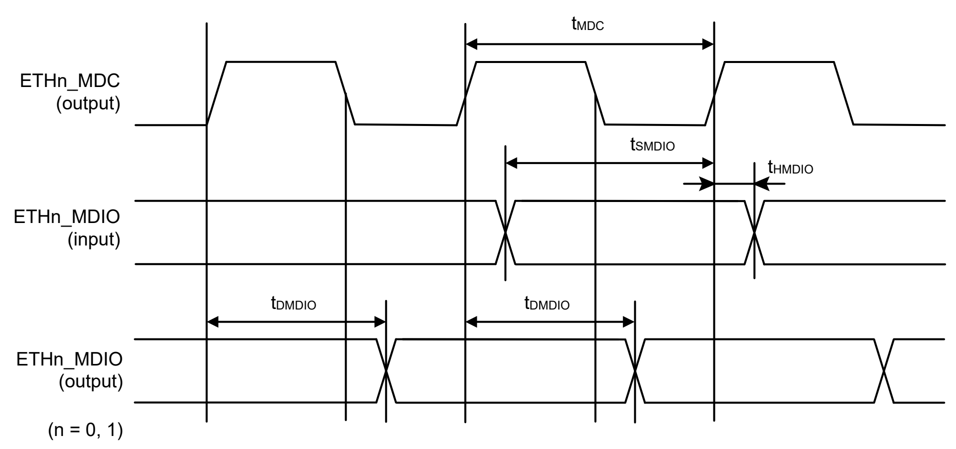


Figure 62.120 ETn_MDIO, ETn_MDC timing

62.3.20 PDMIF Timing

Table 62.84 PDMIF timing

Conditions: High drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter	Symbol	VCC	Min	Max	Unit	Test conditions
Clock period	t_{Psync}	2.70 V or above	250	4000	ns	Figure 62.120
		1.62 V or above	500	4000		
Clock high level period	t_{PDCKWH}	2.70 V or above	t_{PSYNC} × 0.45	t_{PSYNC} × 0.55	ns	
		1.62 V or above	t_{PSYNC} × 0.45	t_{PSYNC} × 0.55		
Clock low level period	t_{PDCKWL}	2.70 V or above	t_{PSYNC} × 0.45	t_{PSYNC} × 0.55	ns	
		1.62 V or above	t_{PSYNC} × 0.45	t_{PSYNC} × 0.55		
Clock rise time	t_{PDCKr}	2.70 V or above	—	3	ns	
		1.62 V or above	—	5		
Clock fall time	t_{PDCKf}	2.70 V or above	—	3	ns	
		1.62 V or above	—	5		
Setup time	t_{SU}	2.70 V or above	15	—	ns	Figure 62.121 Figure 62.122
		1.62 V or above	30	—		
Hold time	t_H	2.70 V or above	0	—	ns	
		1.62 V or above	0	—		

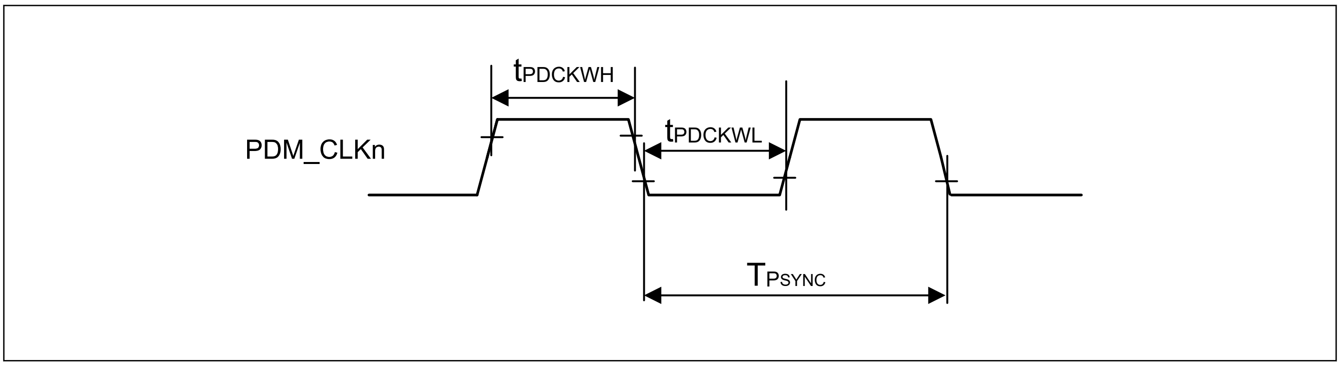


Figure 62.120 Timing of clock output (PDM_CLKn)

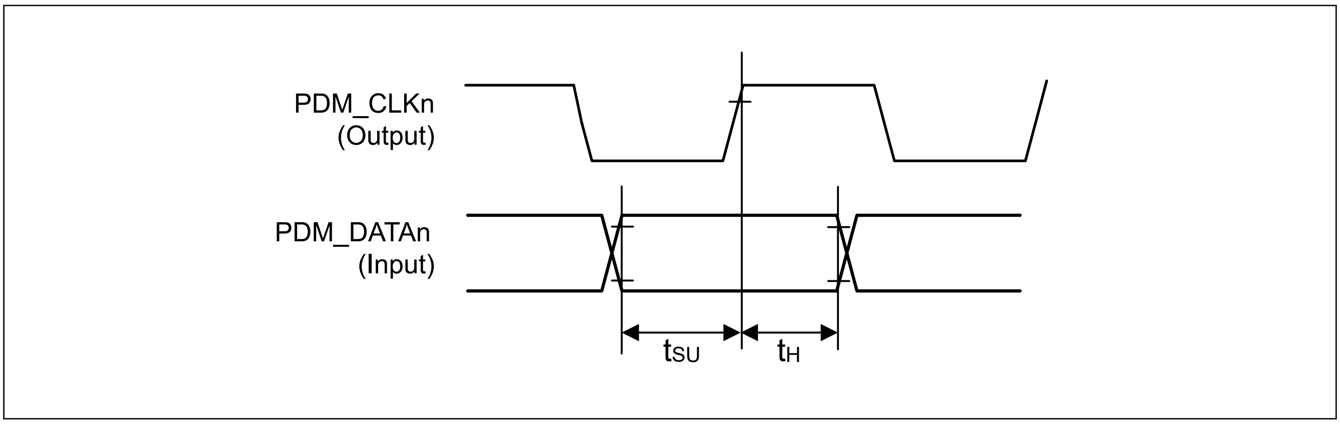


Figure 62.121 Receive Timing (Synchronized with the rise of PDM_CLKn)

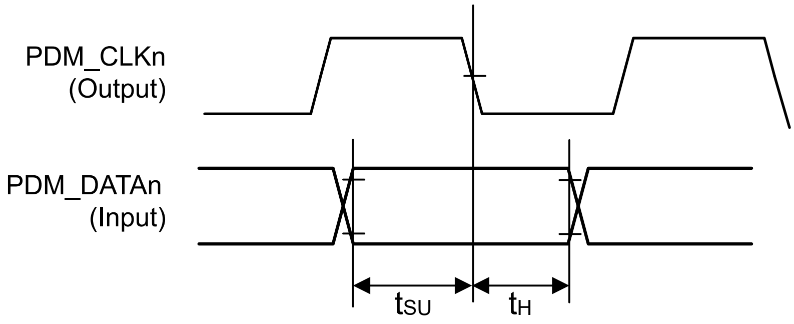


Figure 62.122 Receive Timing (Synchronized with the fall of PDM_CLKn)

62.4 USB Characteristics

62.4.1 USBFS Timing

Table 62.85 USBFS low-speed characteristics for host only (USB_DP and USB_DM pin characteristics)

Conditions: VCC = VCC_USB = 3.0 to 3.6 V, USBCLK = 48 MHz

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Input characteristics	Input high voltage	V_{IH}	2.0	—	—	V	—
	Input low voltage	V_{IL}	—	—	0.8	V	—
	Differential input sensitivity	V_{DI}	0.2	—	—	V	| USB_DP - USB_DM |
	Differential common-mode range	V_{CM}	0.8	—	2.5	V	—
Output characteristics	Output high voltage	V_{OH}	2.8	—	3.6	V	I_{OH} = –200 μA
	Output low voltage	V_{OL}	0.0	—	0.3	V	I_{OL} = 2 mA
	Cross-over voltage	V_{CRS}	1.3	—	2.0	V	Figure 62.123
	Rise time	t_{LR}	75	—	300	ns	
	Fall time	t_{LF}	75	—	300	ns	
	Rise/fall time ratio	t_{LR} / t_{LF}	80	—	125	%	t_{LR}/ t_{LF}
Pull-up and pull-down characteristics	USB_DP and USB_DM pull-down resistance in host controller mode	R_{pd}	14.25	—	24.80	kΩ	—

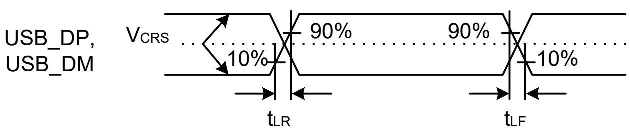


Figure 62.123 USB_DP and USB_DM output timing in low-speed mode

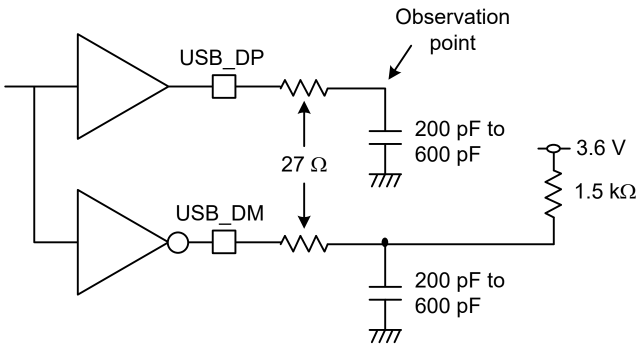


Figure 62.124 Test circuit in low-speed mode

Table 62.86 USBFS full-speed characteristics (USB_DP and USB_DM pin characteristics)

Conditions: VCC = VCC_USB = 3.0 to 3.6 V, USBCLK = 48 MHz

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Input characteristics	Input high voltage	V_{IH}	2.0	—	—	V	—
	Input low voltage	V_{IL}	—	—	0.8	V	—
	Differential input sensitivity	V_{DI}	0.2	—	—	V	| USB_DP - USB_DM |
	Differential common-mode range	V_{CM}	0.8	—	2.5	V	—
Output characteristics	Output high voltage	V_{OH}	2.8	—	3.6	V	I_{OH} = –200 μA
	Output low voltage	V_{OL}	0.0	—	0.3	V	I_{OL} = 2 mA
	Cross-over voltage	V_{CRS}	1.3	—	2.0	V	Figure 62.125
	Rise time	t_{LR}	4	—	20	ns	
	Fall time	t_{LF}	4	—	20	ns	
	Rise/fall time ratio	t_{LR} / t_{LF}	90	—	111.11	%	t_{FR}/ t_{FF}
	Output resistance	Z_{DRV}	28	—	44	Ω	USBFS: Rs = 27 Ω included
Pull-up and pull-down characteristics	DM pull-up resistance in device controller mode	R_{pu}	0.900	—	1.575	kΩ	During idle state
			1.425	—	3.090	kΩ	During transmission and reception
	USB_DP and USB_DM pull-down resistance in host controller mode	R_{pd}	14.25	—	24.80	kΩ	—

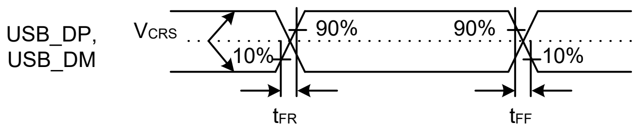


Figure 62.125 USB_DP and USB_DM output timing in full-speed mode

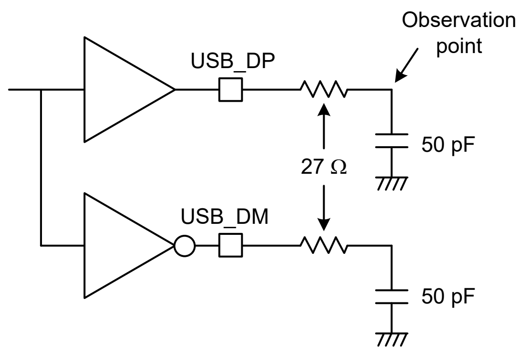


Figure 62.126 Test circuit in full-speed mode

62.4.2 USBHS Timing

Table 62.87 USBHS low-speed characteristics for host only (USB_DP and USB_DM pin characteristics)

Conditions: USBHS_RREF = 2.2 kΩ ± 1%, USBMCLK = 12/20/24/48 MHz, USBCLK = 48MHz, USB60CLK = 60MHz

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Input characteristics	Input high voltage	V_{IH}	2.0	—	—	V	—
	Input low voltage	V_{IL}	—	—	0.8	V	—
	Differential input sensitivity	V_{DI}	0.2	—	—	V	| USB_DP - USB_DM |
	Differential common-mode range	V_{CM}	0.8	—	2.5	V	—
Output characteristics	Output high voltage	V_{OH}	2.8	—	3.6	V	I_{OH} = –200 μA
	Output low voltage	V_{OL}	0.0	—	0.3	V	I_{OL} = 2 mA
	Cross-over voltage	V_{CRS}	1.3	—	2.0	V	Figure 62.127
	Rise time	t_{LR}	75	—	300	ns	
	Fall time	t_{LF}	75	—	300	ns	
	Rise/fall time ratio	t_{LR} / t_{LF}	80	—	125	%	t_{LR}/ t_{LF}
Pull-up and pull-down characteristics	USB_DP and USB_DM pull-down resistance in host controller mode	R_{pd}	14.25	—	24.80	kΩ	—

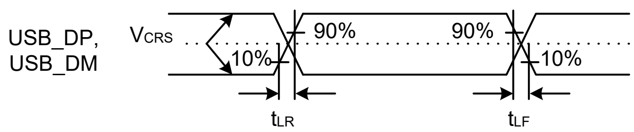


Figure 62.127 USB_DP and USB_DM output timing in low-speed mode

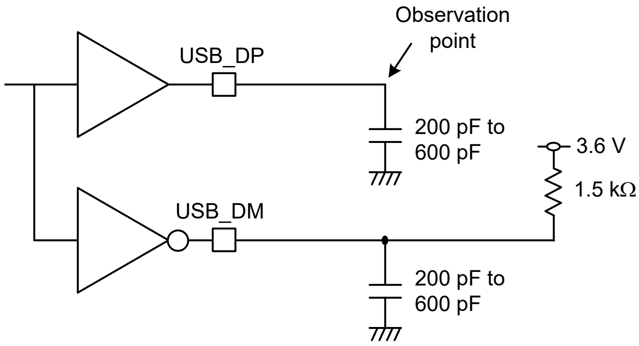


Figure 62.128 Test circuit in low-speed mode

Table 62.88 USBHS full-speed characteristics (USB_DP and USB_DM pin characteristics)

Conditions: USBHS_RREF = 2.2 kΩ ± 1%, USBMCLK = 12/20/24/48 MHz, USBCLK = 48MHz, USB60CLK = 60MHz

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Input characteristics	Input high voltage	V_{IH}	2.0	—	—	V	—
	Input low voltage	V_{IL}	—	—	0.8	V	—
	Differential input sensitivity	V_{DI}	0.2	—	—	V	| USB_DP - USB_DM |
	Differential common-mode range	V_{CM}	0.8	—	2.5	V	—
Output characteristics	Output high voltage	V_{OH}	2.8	—	3.6	V	I_{OH} = –200 μA
	Output low voltage	V_{OL}	0.0	—	0.3	V	I_{OL} = 2 mA
	Cross-over voltage	V_{CRS}	1.3	—	2.0	V	Figure 62.129
	Rise time	t_{LR}	4	—	20	ns	
	Fall time	t_{LF}	4	—	20	ns	
	Rise/fall time ratio	t_{LR} / t_{LF}	90	—	111.11	%	t_{FR}/ t_{FF}
	Output resistance	Z_{DRV}	40.5	—	49.5	Ω	Rs Not used (PHYSET.REPSEL[1:0] = 01b and PHYSET. HSEB = 0)
Pull-up and pull-down characteristics	DM pull-up resistance in device controller mode	R_{pu}	0.900	—	1.575	kΩ	During idle state
			1.425	—	3.090	kΩ	During transmission and reception
	USB_DP and USB_DM pull-down resistance in host controller mode	R_{pd}	14.25	—	24.80	kΩ	—

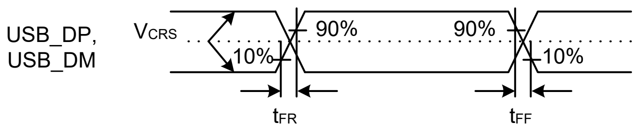


Figure 62.129 USB_DP and USB_DM output timing in full-speed mode

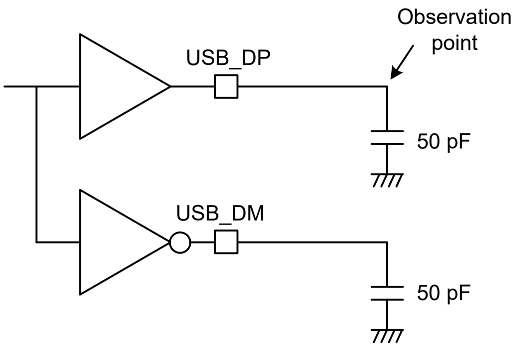


Figure 62.130 Test circuit in full-speed mode

Table 62.89 USB High Speed Characteristics (USB_DP and USB_DM Pin Characteristics)

Conditions: USBHS_RREF = 2.2 kΩ ± 1%, USBMCLK = 12/20/24/48 MHz

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Input characteristics	Squelch detect sensitivity	V_{HSSQ}	100	—	150	mV	Figure 62.131
	Disconnect detect sensitivity	V_{HSDSC}	525	—	648	mV	Figure 62.132
	Common mode voltage	V_{HSCM}	-50	—	500	mV	—
Output characteristics	Idle state	V_{HSOI}	-10	—	10	mV	—
	Output high level voltage	V_{HSOH}	360	—	440	mV	—
	Output low level voltage	V_{HSOL}	-10	—	10	mV	—
	Chirp J output voltage (difference)	V_{CHIRPJ}	700	—	1100	mV	—
	Chirp K output voltage (difference)	V_{CHIRPK}	-900	—	-500	mV	—
AC characteristics	Rise time	t_{HSR}	500	—	—	ps	—
	Fall time	t_{HSF}	500	—	—	ps	Figure 62.133
	Output resistance	Z_{HSDRV}	40.5	—	49.5	Ω	—

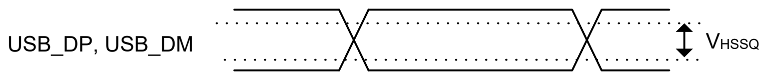


Figure 62.131 USB_DP and USB_DM squelch detect sensitivity (high-speed)

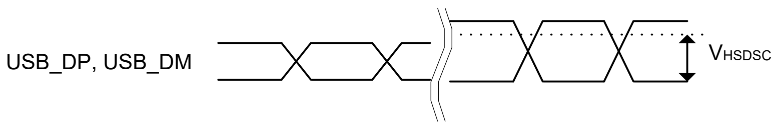


Figure 62.132 USB_DP and USB_DM disconnect detect sensitivity (high-speed)

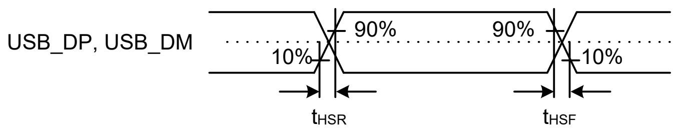


Figure 62.133 USB_DP and USB_DM output timing (high-speed)

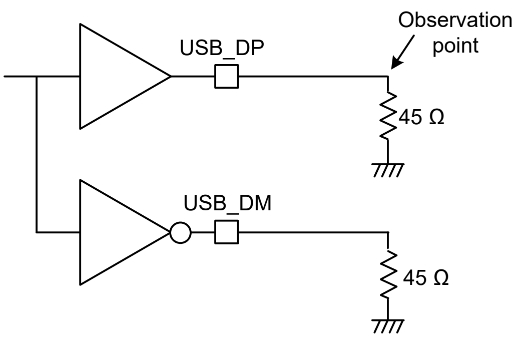


Figure 62.134 Test circuit (high-speed)

Table 62.90 USBHS high-speed characteristics (USB_DP and USB_DM pin characteristics)

Conditions: USBHS_RREF = 2.2 kΩ ± 1%, USBMCLK = 12/20/24/48 MHz

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Battery Charging Specification	D+ sink current	I_{DP_SINK}	25	—	175	μA	—
	D- sink current	I_{DM_SINK}	25	—	175	μA	—
	DCD source current	I_{DP_SRC}	7	—	13	μA	—
	Data detection voltage	V_{DAT_REF}	0.25	—	0.4	V	—
	D+ source voltage	V_{DP_SRC}	0.5	—	0.7	V	Output current = 250 μA
	D- source voltage	V_{DM_SRC}	0.5	—	0.7	V	Output current = 250 μA

62.5 ADC Characteristics

Table 62.91 A/D conversion characteristics (common) (1 of 3)

Conditions: AVCC: 2.7 to 3.63 V, VCC: 2.7 to 3.63 V, VREFH0/VREFH: 2.7 V to AVCC

Parameter	Min	Typ	Max	Unit	Test conditions
A/D conversion clock frequency (ADCLK)	25	50	60	MHz	AVCC: 2.7 to 3.63 V VCC: 2.7 to 3.63 V VREFH0/VREFH: 2.7 V to AVCC
Successive approximation time	100	—	200	ns	

Table 62.91 A/D conversion characteristics (common) (2 of 3)

Conditions: AVCC: 2.7 to 3.63 V, VCC: 2.7 to 3.63 V, VREFH0/VREFH: 2.7 V to AVCC

Parameter						Min	Typ	Max	Unit	Test conditions
A/D sampling time (Normal mode)	Self-calibration				SAR mode	$1 \times t_{ADcyc} + 40$	—	—	ns	AVCC: 2.7 to 3.63 V VCC: 2.7 to 3.63 V VREFH0/VREFH: 2.7 V to AVCC tCmp=100ns
					Oversampling mode	$1 \times t_{ADcyc} + 40$	—	—	ns	
	Self-diagnosis				SAR mode	$1 \times t_{ADcyc} + 40$	—	—	ns	
					Oversampling mode	$1 \times t_{ADcyc} + 40$	—	—	ns	
	A/D conversion	High-speed channels	Without channel dedicated sample and hold circuits	(AN000 to AN005) (AN006 to AN011)	SAR mode	$1 \times t_{ADcyc} + 40$	—	—	ns	
					Oversampling mode (One-channel continuous scan mode)	40	—	—	ns	
					Oversampling mode (Single/ Continuous scan mode)	$1 \times t_{ADcyc} + 40$	—	—	ns	
			With channel-dedicated sample-and-hold circuits	(AN000 to AN005) (AN006 to AN011)	SAR mode	$1 \times t_{ADcyc} + 160$	—	—	ns	
					Hybrid mode	$1 \times t_{ADcyc} + 160$	—	—	ns	
		Middle-speed channels		(AN012 to AN015)	SAR mode	180	—	—	ns	
					Oversampling mode	200	—	—	ns	
		Low-speed channels		(AN016 to AN022)	SAR mode	400	—	—	ns	
					Oversampling mode	400	—	—	ns	
A/D sampling time (High accuracy mode)	Self-calibration				SAR mode	$1 \times t_{ADcyc} + 140^{*1}$	—	—	ns	
					Hybrid mode	$1 \times t_{ADcyc} + 140^{*1}$	—	—	ns	
	Self-diagnosis				SAR mode	$1 \times t_{ADcyc} + 140^{*1}$	—	—	ns	
					Hybrid mode	$1 \times t_{ADcyc} + 140^{*1}$	—	—	ns	
	A/D conversion	High-speed channels	Without channel dedicated sample and hold circuits	(AN000 to AN005) (AN006 to AN011)	SAR mode	$1 \times t_{ADcyc} + 140^{*1}$	—	—	ns	
					Hybrid mode	$1 \times t_{ADcyc} + 140^{*1}$	—	—	ns	
			With channel-dedicated sample-and-hold circuits	(AN000 to AN005) (AN006 to AN011)	SAR mode	$1 \times t_{ADcyc} + 320$	—	—	ns	
					Hybrid mode	$1 \times t_{ADcyc} + 320$	—	—	ns	
		Middle-speed channels		(AN012 to AN015)	SAR mode	400	—	—	ns	
					Hybrid mode	440	—	—	ns	
		Low-speed channels		(AN016 to AN022)	SAR mode	840	—	—	ns	
					Hybrid mode	840	—	—	ns	

Table 62.91 A/D conversion characteristics (common) (3 of 3)

Conditions: AVCC: 2.7 to 3.63 V, VCC: 2.7 to 3.63 V, VREFH0/VREFH: 2.7 V to AVCC

Parameter				Min	Typ	Max	Unit	Test conditions
Channel dedicated sample and hold circuits	Sampling time	Self-calibration		400	—	—	ns	AVCC: 2.7 to 3.63 V VCC: 2.7 to 3.63 V VREFH0/VREFH: 2.7 V to AVCC
		A/D conversion		400	—	—	ns	
	Hold mode switching time			40	—	—	ns	
	Hold time			—	—	5	μs	
Operation stabilization time	A/D start-up time			2	—	—	μs	
	Channel-dedicated sample-and-hold circuits start-up time			2	—	—	μs	
	A/D shut-down time			1	—	—	μs	
Analog input voltage range	Single-ended input voltage	Unit 0	AN000 to AN005, AN012, AN014	VREFL0	—	VREFH0	V	—
			AN016 to AN018	VREFL0	—	VREFH0	V	VCC ≥ VREFH0
				VREFL0	—	VCC	V	VCC < VREFH0
			AN019 to AN022	VREFL0	—	VREFH0	V	VCC2 ≥ VREFH0
				VREFL0	—	VCC2	V	VCC2 < VREFH0
		Unit 1	AN006 to AN011, AN013, AN015	VREFL	—	VREFH	V	—
			AN016 to AN018	VREFL	—	VREFH	V	VCC ≥ VREFH
				VREFL	—	VCC	V	VCC < VREFH
			AN019 to AN022	VREFL	—	VREFH	V	VCC2 ≥ VREFH
				VREFL	—	VCC2	V	VCC2 < VREFH
	Differential input voltage*2	Unit 0	AN000 to AN005	-VREFH0	—	+VREFH0	V	—
		Unit 1	AN006 to AN011	-VREFH	—	+VREFH	V	—

Note: t_{ADcyc}: ADCLK cycle

Note: t_{Cmp}: Successive approximation time

Note 1. If t_{Cmp} is greater than 100ns, the A/D sampling time should be greater than the following equation.

$1 \times t_{ADcyc} + 1.6 \times t_{Cmp}$

Note 2. Differential input voltage is (A_{INP} - A_{INN}).

A/D Converter Unit 0:

- A_{INP} is input voltage of A_{Nx}, and VREFL0 ≤ A_{INP} ≤ VREFH0.
- A_{INN} is input voltage of A_{Ny}, and VREFL0 ≤ A_{INN} ≤ VREFH0.

A/D Converter Unit 1:

- A_{INP} is input voltage of A_{Nx}, and VREFL ≤ A_{INP} ≤ VREFH.
- A_{INN} is input voltage of A_{Ny}, and VREFL ≤ A_{INN} ≤ VREFH.

(x = 2i, y = 2i +1, i = 0, 1, 2… (any integer))

Table 62.92 A/D conversion characteristics (common) (1 of 2)

Conditions: AVCC: 1.62 to 2.7 V, VCC: 1.62 to 2.7 V, VREFH0/VREFH: 1.62 V to AVCC

Parameter						Min	Typ	Max	Unit	Test conditions
A/D conversion clock frequency (ADCLK)						25	50	60	MHz	AVCC: 1.62 to 2.7 V VCC: 1.62 to 2.7 V VREFH0/VREFH: 1.62 V to AVCC
Successive approximation time						200	—	200	ns	
A/D sampling time (Normal mode)	Self-calibration				SAR mode	$1 \times t_{ADcyc} + 420$	—	—	ns	AVCC: 1.62 to 2.7 V VCC: 1.62 to 2.7 V VREFH0/VREFH: 1.62 V to AVCC tCmp=200ns
					Oversampling mode	$1 \times t_{ADcyc} + 420$	—	—	ns	
	Self-diagnosis				SAR mode	$1 \times t_{ADcyc} + 420$	—	—	ns	
					Oversampling mode	$1 \times t_{ADcyc} + 420$	—	—	ns	
	A/D conversion	High-speed channels	Without channel dedicated sample and hold circuits	(AN000 to AN005) (AN006 to AN011)	SAR mode	$1 \times t_{ADcyc} + 420$	—	—	ns	
					Oversampling mode (One-channel continuous scan mode)	440	—	—	ns	
					Oversampling mode (Single/ Continuous scan mode)	$1 \times t_{ADcyc} + 420$	—	—	ns	
		Middle-speed channels		(AN012 to AN015)	SAR mode	560	—	—	ns	
					Oversampling mode	560	—	—	ns	
		Low-speed channels		(AN016 to AN022)	SAR mode	800	—	—	ns	
					Oversampling mode	800	—	—	ns	
A/D sampling time (High accuracy mode)	Self-calibration				SAR mode	$1 \times t_{ADcyc} + 780$	—	—	ns	
					Hybrid mode	$1 \times t_{ADcyc} + 780$	—	—	ns	
	Self-diagnosis				SAR mode	$1 \times t_{ADcyc} + 780$	—	—	ns	
					Hybrid mode	$1 \times t_{ADcyc} + 780$	—	—	ns	
	A/D conversion	High-speed channels	Without channel dedicated sample and hold circuits	(AN000 to AN005) (AN006 to AN011)	SAR mode	$1 \times t_{ADcyc} + 780$	—	—	ns	
					Hybrid mode	$1 \times t_{ADcyc} + 780$	—	—	ns	
		Middle-speed channels		(AN012 to AN015)	SAR mode	1200	—	—	ns	
					Hybrid mode	1200	—	—	ns	
		Low-speed channels		(AN016 to AN022)	SAR mode	1680	—	—	ns	
					Hybrid mode	1680	—	—	ns	
Operation	A/D start-up time					2	—	—	ns	AVCC: 1.62 to 2.7 V VCC: 1.62 to 2.7 V VREFH0/VREFH: 1.62 V to AVCC
	A/D shut-down time					1	—	—	ns	

Table 62.92 A/D conversion characteristics (common) (2 of 2)

Conditions: AVCC: 1.62 to 2.7 V, VCC: 1.62 to 2.7 V, VREFH0/VREFH: 1.62 V to AVCC

Parameter				Min	Typ	Max	Unit	Test conditions
Analog input voltage range	Single-ended input voltage	Unit 0	AN000 to AN005, AN012, AN014	VREFL0	—	VREFH0	V	—
			AN016 to AN018	VREFL0	—	VREFH0	V	VCC ≥ VREFH0
				VREFL0	—	VCC	V	VCC < VREFH0
			AN019 to AN022	VREFL0	—	VREFH0	V	VCC2 ≥ VREFH0
				VREFL0	—	VCC2	V	VCC2 < VREFH0
		Unit 1	AN006 to AN011, AN013, AN015	VREFL	—	VREFH	V	—
			AN016 to AN018	VREFL	—	VREFH	V	VCC ≥ VREFH
				VREFL	—	VCC	V	VCC < VREFH
			AN019 to AN022	VREFL	—	VREFH	V	VCC2 ≥ VREFH
				VREFL	—	VCC2	V	VCC2 < VREFH
	Differential input voltage*1	Unit 0	AN000 to AN005	-VREFH0	—	+VREFH0	V	—
		Unit 1	AN006 to AN011	-VREFH	—	+VREFH	V	—

Note: t_{ADcyc}: ADCLK cycle

Note: t_{Cmp}: Successive approximation time

Note 1. Differential input voltage is (A_{INP} - A_{INN}).

A/D Converter Unit 0:

- A_{INP} is input voltage of A_{Nx}, and VREFL0 ≤ A_{INP} ≤ VREFH0.
- A_{INN} is input voltage of A_{Ny}, and VREFL0 ≤ A_{INN} ≤ VREFH0.

A/D Converter Unit 1:

- A_{INP} is input voltage of A_{Nx}, and VREFL ≤ A_{INP} ≤ VREFH.
- A_{INN} is input voltage of A_{Ny}, and VREFL ≤ A_{INN} ≤ VREFH.

(x = 2i, y = 2i +1, i = 0, 1, 2… (any integer))

Table 62.93 A/D conversion characteristics (SAR mode : DCDC mode) (1 of 6)

Conditions: AVCC: 2.7 to 3.63 V, VCC: 2.7 to 3.63 V, VREFH0/VREFH: 2.7 V to AVCC

Parameter						Min	Typ	Max	Unit	Test conditions
SAR mode	Resolution					—	—	12	bit	—
SAR mode	Single-ended input	Normal mode	High-speed channels (AN000 to AN005) (AN006 to AN011)	Without channel dedicated sample and hold circuits	Conversion time[*1]	0.16	—	—	µs	• ADCLK: 50 MHz • Sampling time: 3 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
					Offset error	—	±3	±6.5	LSB	BGA package
						—	±3	±7.5	LSB	LQFP package
					Full-scale error	—	±3	±6.5	LSB	BGA package
						—	±3	±7.5	LSB	LQFP package
					Absolute accuracy	—	±4	±11	LSB	BGA package
						—	±4	±14	LSB	LQFP package
					DNL differential nonlinearity error [*3]	—	±1	-1 to +1.5	LSB	BGA package
						—	±1	-1 to +2.0	LSB	LQFP package
					INL integral nonlinearity error	—	±2	±3	LSB	BGA package
						—	±2	±5	LSB	LQFP package
			High-speed channels (AN000 to AN005) (AN006 to AN011)	With channel dedicated sample and hold circuits	Conversion time[*2]	1.00	—	—	µs	• ADCLK: 50 MHz • Sampling time of channel-dedicated sample-and-hold circuits: 35 ADCLK • Hold mode switching time of channel-dedicated sample-and-hold circuits: 2 ADCLK • Sampling time: 8 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
					Offset error	—	±1.5	±6.75	LSB	BGA package
						—	±1.5	±7.5	LSB	LQFP package
					Full-scale error	—	±1.5	±6.75	LSB	BGA package
						—	±1.5	±7.5	LSB	LQFP package
					Absolute accuracy	—	±5	±10.5	LSB	BGA package
						—	±5	±11.5	LSB	LQFP package
					DNL differential nonlinearity error [*3]	—	±1	-1 to +1.5	LSB	BGA package
						—	±1	-1 to +2.0	LSB	LQFP package
					INL integral nonlinearity error	—	±2.5	±3.5	LSB	BGA package
						—	±2.5	±5.5	LSB	LQFP package

Table 62.93 A/D conversion characteristics (SAR mode : DCDC mode) (2 of 6)

Conditions: AVCC: 2.7 to 3.63 V, VCC: 2.7 to 3.63 V, VREFH0/VREFH: 2.7 V to AVCC

Parameter						Min	Typ	Max	Unit	Test conditions
SAR mode	Single-ended input	High accuracy mode	High-speed channels (AN000 to AN005) (AN006 to AN011)	Without channel dedicated sample and hold circuits	Conversion time[*1]	0.26	—	—	μs	• ADCLK: 50 MHz • Sampling time: 8 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
					Offset error	—	±1.5	±4.5	LSB	BGA package
						—	±1.5	±5.5	LSB	LQFP package
					Full-scale error	—	±1.5	±4.5	LSB	BGA package
						—	±1.5	±5.5	LSB	LQFP package
					Absolute accuracy	—	±4	±7	LSB	BGA package
						—	±4	±10	LSB	LQFP package
					DNL differential nonlinearity error [*3]	—	±1	-1 to +1.5	LSB	BGA package
						—	±1	-1 to +2.0	LSB	LQFP package
					INL integral nonlinearity error	—	±2	±3	LSB	BGA package
						—	±2	±5	LSB	LQFP package
			High-speed channels (AN000 to AN005) (AN006 to AN011)	With channel dedicated sample and hold circuits	Conversion time[*2]	1.72	—	—	μs	• ADCLK: 50 MHz • Sampling time of channel-dedicated sample-and-hold circuits: 63 ADCLK • Hold mode switching time of channel-dedicated sample-and-hold circuits: 2 ADCLK • Sampling time: 16 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
					Offset error	—	±1.5	±6.75	LSB	BGA package
						—	±1.5	±7.5	LSB	LQFP package
					Full-scale error	—	±1.5	±6.75	LSB	BGA package
						—	±1.5	±7.5	LSB	LQFP package
					Absolute accuracy	—	±4.5	±9	LSB	BGA package
						—	±4.5	±9.5	LSB	LQFP package
					DNL differential nonlinearity error [*3]	—	±1	-1 to +1.5	LSB	BGA package
						—	±1	-1 to +2.0	LSB	LQFP package
					INL integral nonlinearity error	—	±2.5	±3.5	LSB	BGA package
						—	±2.5	±4.5	LSB	LQFP package

Table 62.93 A/D conversion characteristics (SAR mode : DCDC mode) (3 of 6)

Conditions: AVCC: 2.7 to 3.63 V, VCC: 2.7 to 3.63 V, VREFH0/VREFH: 2.7 V to AVCC

Parameter					Min	Typ	Max	Unit	Test conditions
SAR mode	Single-ended input	Normal mode	Middle-speed channels (AN012 to AN015)	Conversion time*1	0.28	—	—	μs	• ADCLK: 50 MHz • Sampling time: 9 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
				Offset error	—	±1.5	±6.5	LSB	BGA package
					—	±1.5	±6.5	LSB	LQFP package
				Full-scale error	—	±1.5	±6.5	LSB	BGA package
					—	±1.5	±6.5	LSB	LQFP package
				Absolute accuracy	—	±4	±11	LSB	BGA package
					—	±4	±15	LSB	LQFP package
				DNL differential nonlinearity error *3	—	±1	-1 to +1.5	LSB	BGA package
					—	±1	-1 to +2.0	LSB	LQFP package
				INL integral nonlinearity error	—	±2	±3	LSB	BGA package
					—	±2	±3	LSB	LQFP package
			Low-speed channels (AN016 to AN022)	Conversion time*1	0.5	—	—	μs	• ADCLK: 50 MHz • Sampling time: 20 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
				Offset error	—	±1.5	±6.5	LSB	—
				Full-scale error	—	±1.5	±6.5	LSB	—
				Absolute accuracy	—	±5.5	±11	LSB	—
				DNL differential nonlinearity error *3	—	±1	-1 to +1.5	LSB	—
				INL integral nonlinearity error	—	±2	±4	LSB	—

Table 62.93 A/D conversion characteristics (SAR mode : DCDC mode) (4 of 6)

Conditions: AVCC: 2.7 to 3.63 V, VCC: 2.7 to 3.63 V, VREFH0/VREFH: 2.7 V to AVCC

Parameter					Min	Typ	Max	Unit	Test conditions
SAR mode	Single-ended input	High accuracy mode	Middle-speed channels (AN012 to AN015)	Conversion time*1	0.5	—	—	μs	• ADCLK: 50 MHz • Sampling time: 20 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
				Offset error	—	±1.5	±4.5	LSB	BGA package
					—	±1.5	±4.5	LSB	LQFP package
				Full-scale error	—	±1.5	±4.5	LSB	BGA package
					—	±1.5	±4.5	LSB	LQFP package
				Absolute accuracy	—	±4	±7	LSB	BGA package
					—	±4	±11	LSB	LQFP package
				DNL differential nonlinearity error *3	—	±1	-1 to +1.5	LSB	BGA package
					—	±1	-1 to +2.0	LSB	LQFP package
				INL integral nonlinearity error	—	±2	±3	LSB	BGA package
					—	±2	±4.5	LSB	LQFP package
			Low-speed channels (AN016 to AN022)	Conversion time*1	0.94	—	—	μs	• ADCLK: 50 MHz • Sampling time: 42 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
				Offset error	—	±1.5	±4.5	LSB	—
				Full-scale error	—	±1.5	±4.5	LSB	—
				Absolute accuracy	—	±5.5	±8	LSB	—
				DNL differential nonlinearity error *3	—	±1	-1 to +1.5	LSB	—
				INL integral nonlinearity error	—	±2	±4	LSB	—

Table 62.93 A/D conversion characteristics (SAR mode : DCDC mode) (5 of 6)

Conditions: AVCC: 2.7 to 3.63 V, VCC: 2.7 to 3.63 V, VREFH0/VREFH: 2.7 V to AVCC

Parameter						Min	Typ	Max	Unit	Test conditions
SAR mode	Differential input	Normal mode	High-speed channels (AN000 to AN005) (AN006 to AN011)	Without channel dedicated sample and hold circuits	Conversion time*1	0.16	—	—	μs	• ADCLK: 50 MHz • Sampling time: 3 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
					Offset error	—	±2	±3.5	LSB	—
					Full-scale error	—	±2	±3.5	LSB	—
					Absolute accuracy	—	±3	±6	LSB	—
					DNL differential nonlinearity error *3	—	±0.75	±1	LSB	—
					INL integral nonlinearity error	—	±1.5	±2	LSB	—
			High-speed channels (AN000 to AN005) (AN006 to AN011)	With channel dedicated sample and hold circuits	Conversion time*2	1.00	—	—	μs	• ADCLK: 50 MHz • Sampling time of channel-dedicated sample-and-hold circuits: 35 ADCLK • Hold mode switching time of channel-dedicated sample-and-hold circuits: 2 ADCLK • Sampling time: 8 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
					Offset error	—	±1.5	±6.75	LSB	—
					Full-scale error	—	±1.5	±6.75	LSB	—
					Absolute accuracy	—	±3.5	±10.5	LSB	—
					DNL differential nonlinearity error *3	—	±1	-1 to +1.5	LSB	—
					INL integral nonlinearity error	—	±2.5	±3.5	LSB	—

Table 62.93 A/D conversion characteristics (SAR mode : DCDC mode) (6 of 6)

Conditions: AVCC: 2.7 to 3.63 V, VCC: 2.7 to 3.63 V, VREFH0/VREFH: 2.7 V to AVCC

Parameter						Min	Typ	Max	Unit	Test conditions
SAR mode	Differential input	High accuracy mode	High-speed channels (AN000 to AN005) (AN006 to AN011)	Without channel dedicated sample and hold circuits	Conversion time*1	0.26	—	—	μs	• ADCLK: 50 MHz • Sampling time: 8 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
					Offset error	—	±1	±2.5	LSB	—
					Full-scale error	—	±1	±2.5	LSB	—
					Absolute accuracy	—	±2	±4	LSB	—
					DNL differential nonlinearity error *3	—	±0.75	±1	LSB	—
					INL integral nonlinearity error	—	±1.5	±2	LSB	—
			High-speed channels (AN000 to AN005) (AN006 to AN011)	With channel dedicated sample and hold circuits	Conversion time*2	1.72	—	—	μs	• ADCLK: 50 MHz • Sampling time of channel-dedicated sample-and-hold circuits: 63 ADCLK • Hold mode switching time of channel-dedicated sample-and-hold circuits: 2 ADCLK • Sampling time: 16 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
					Offset error	—	±1.5	±6.75	LSB	—
					Full-scale error	—	±1.5	±6.75	LSB	—
					Absolute accuracy	—	±3.5	±9	LSB	—
					DNL differential nonlinearity error *3	—	±1	-1 to +1.5	LSB	—
					INL integral nonlinearity error	—	±2.5	±3.5	LSB	—

Note: These specification values are applicable when only one ADC16H is operating, DAC12 and ACMPHS are not operating, and there is no access to the external bus during A/D conversion.
If other ADC unit, DAC12, or ACMPHS is operating or bus access occurs during A/D conversion, values might not fall within the indicated ranges.
The use of ports 0 as digital outputs is not allowed when the ADC16H is used.
The characteristics apply when AVCC0, AVSS0, VREFH0, VREFH, VREFL0, VREFL, and ADC16H input voltage is stable.

Note 1. Without channel dedicated sample and hold circuits; The conversion time is the sum of the sampling time and the successive approximation time. Each of the above state is indicated for the test conditions.

Note 2. With channel-dedicated sample-and-hold circuits; The conversion time is the sum of the sampling time of channel-dedicated sample-and-hold circuits, the hold mode switching time, the sampling time and the successive approximation time. Each of the above state is indicated for the test conditions.

Note 3. DNL is measured using the Histogram Method, so the lower limit value is -1.

Table 62.94 A/D conversion characteristics (SAR mode : External VDD mode) (1 of 6)

Conditions: AVCC: 2.7 to 3.63 V, VCC: 2.7 to 3.63 V, VREFH0/VREFH: 2.7 V to AVCC

Parameter						Min	Typ	Max	Unit	Test conditions
SAR mode	Resolution					—	—	12	bit	—
SAR mode	Single-ended input	Normal mode	High-speed channels (AN000 to AN005) (AN006 to AN011)	Without channel dedicated sample and hold circuits	Conversion time*1	0.16	—	—	μs	• ADCLK: 50 MHz • Sampling time: 3 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
					Offset error	—	±3	±6.5	LSB	—
					Full-scale error	—	±3	±6.5	LSB	—
					Absolute accuracy	—	±4	±11	LSB	—
					DNL differential nonlinearity error *3	—	±1	-1 to +1.5	LSB	—
					INL integral nonlinearity error	—	±2	±3	LSB	—
			High-speed channels (AN000 to AN005) (AN006 to AN011)	With channel dedicated sample and hold circuits	Conversion time*2	1.00	—	—	μs	• ADCLK: 50 MHz • Sampling time of channel-dedicated sample-and-hold circuits: 35 ADCLK • Hold mode switching time of channel-dedicated sample-and-hold circuits: 2 ADCLK • Sampling time: 8 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
					Offset error	—	±1.5	±6.75	LSB	—
					Full-scale error	—	±1.5	±6.75	LSB	—
					Absolute accuracy	—	±5	±10.5	LSB	—
					DNL differential nonlinearity error *3	—	±1	-1 to +1.5	LSB	—
					INL integral nonlinearity error	—	±2.5	±3.5	LSB	—

Table 62.94 A/D conversion characteristics (SAR mode : External VDD mode) (2 of 6)

Conditions: AVCC: 2.7 to 3.63 V, VCC: 2.7 to 3.63 V, VREFH0/VREFH: 2.7 V to AVCC

Parameter						Min	Typ	Max	Unit	Test conditions
SAR mode	Single-ended input	High accuracy mode	High-speed channels (AN000 to AN005) (AN006 to AN011)	Without channel dedicated sample and hold circuits	Conversion time*1	0.26	—	—	μs	• ADCLK: 50 MHz • Sampling time: 8 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
					Offset error	—	±1.5	±4.5	LSB	—
					Full-scale error	—	±1.5	±4.5	LSB	—
					Absolute accuracy	—	±4	±7	LSB	—
					DNL differential nonlinearity error *3	—	±1	-1 to +1.5	LSB	—
					INL integral nonlinearity error	—	±2	±3	LSB	—
			High-speed channels (AN000 to AN005) (AN006 to AN011)	With channel dedicated sample and hold circuits	Conversion time*2	1.72	—	—	μs	• ADCLK: 50 MHz • Sampling time of channel-dedicated sample-and-hold circuits: 63 ADCLK • Hold mode switching time of channel-dedicated sample-and-hold circuits: 2 ADCLK • Sampling time: 16 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
					Offset error	—	±1.5	±6.75	LSB	—
					Full-scale error	—	±1.5	±6.75	LSB	—
					Absolute accuracy	—	±4.5	±9	LSB	—
					DNL differential nonlinearity error *3	—	±1	-1 to +1.5	LSB	—
					INL integral nonlinearity error	—	±2.5	±3.5	LSB	—

Table 62.94 A/D conversion characteristics (SAR mode : External VDD mode) (3 of 6)

Conditions: AVCC: 2.7 to 3.63 V, VCC: 2.7 to 3.63 V, VREFH0/VREFH: 2.7 V to AVCC

Parameter					Min	Typ	Max	Unit	Test conditions
SAR mode	Single-ended input	Normal mode	Middle-speed channels (AN012 to AN015)	Conversion time*1	0.28	—	—	μs	• ADCLK: 50 MHz • Sampling time: 9 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
				Offset error	—	±1.5	±6.5	LSB	—
				Full-scale error	—	±1.5	±6.5	LSB	—
				Absolute accuracy	—	±4	±11	LSB	—
				DNL differential nonlinearity error *3	—	±1	-1 to +1.5	LSB	—
				INL integral nonlinearity error	—	±2	±3	LSB	—
			Low-speed channels (AN016 to AN022)	Conversion time*1	0.5	—	—	μs	• ADCLK: 50 MHz • Sampling time: 20 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
				Offset error	—	±1.5	±6.5	LSB	—
				Full-scale error	—	±1.5	±6.5	LSB	—
				Absolute accuracy	—	±5.5	±11	LSB	—
				DNL differential nonlinearity error *3	—	±1	-1 to +1.5	LSB	—
				INL integral nonlinearity error	—	±2	±4	LSB	—

Table 62.94 A/D conversion characteristics (SAR mode : External VDD mode) (4 of 6)

Conditions: AVCC: 2.7 to 3.63 V, VCC: 2.7 to 3.63 V, VREFH0/VREFH: 2.7 V to AVCC

Parameter					Min	Typ	Max	Unit	Test conditions
SAR mode	Single-ended input	High accuracy mode	Middle-speed channels (AN012 to AN015)	Conversion time*1	0.5	—	—	μs	• ADCLK: 50 MHz • Sampling time: 20 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
				Offset error	—	±1.5	±4.5	LSB	—
				Full-scale error	—	±1.5	±4.5	LSB	—
				Absolute accuracy	—	±4	±7	LSB	—
				DNL differential nonlinearity error *3	—	±1	-1 to +1.5	LSB	—
				INL integral nonlinearity error	—	±2	±3	LSB	—
			Low-speed channels (AN016 to AN022)	Conversion time*1	0.94	—	—	μs	• ADCLK: 50 MHz • Sampling time: 42 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
				Offset error	—	±1.5	±4.5	LSB	—
				Full-scale error	—	±1.5	±4.5	LSB	—
				Absolute accuracy	—	±5.5	±8	LSB	—
				DNL differential nonlinearity error *3	—	±1	-1 to +1.5	LSB	—
				INL integral nonlinearity error	—	±2	±4	LSB	—

Table 62.94 A/D conversion characteristics (SAR mode : External VDD mode) (5 of 6)

Conditions: AVCC: 2.7 to 3.63 V, VCC: 2.7 to 3.63 V, VREFH0/VREFH: 2.7 V to AVCC

Parameter						Min	Typ	Max	Unit	Test conditions
SAR mode	Differential input	Normal mode	High-speed channels (AN000 to AN005) (AN006 to AN011)	Without channel dedicated sample and hold circuits	Conversion time*1	0.16	—	—	μs	• ADCLK: 50 MHz • Sampling time: 3 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
					Offset error	—	±2	±3.5	LSB	—
					Full-scale error	—	±2	±3.5	LSB	—
					Absolute accuracy	—	±3	±6	LSB	—
					DNL differential nonlinearity error *3	—	±0.75	±1	LSB	—
					INL integral nonlinearity error	—	±1.5	±2	LSB	—
			High-speed channels (AN000 to AN005) (AN006 to AN011)	With channel dedicated sample and hold circuits	Conversion time*2	1.00	—	—	μs	• ADCLK: 50 MHz • Sampling time of channel-dedicated sample-and-hold circuits: 35 ADCLK • Hold mode switching time of channel-dedicated sample-and-hold circuits: 2 ADCLK • Sampling time: 8 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
					Offset error	—	±1.5	±6.75	LSB	—
					Full-scale error	—	±1.5	±6.75	LSB	—
					Absolute accuracy	—	±3.5	±10.5	LSB	—
					DNL differential nonlinearity error *3	—	±1	-1 to +1.5	LSB	—
					INL integral nonlinearity error	—	±2.5	±3.5	LSB	—

Table 62.94 A/D conversion characteristics (SAR mode : External VDD mode) (6 of 6)

Conditions: AVCC: 2.7 to 3.63 V, VCC: 2.7 to 3.63 V, VREFH0/VREFH: 2.7 V to AVCC

Parameter						Min	Typ	Max	Unit	Test conditions
SAR mode	Differential input	High accuracy mode	High-speed channels (AN000 to AN005) (AN006 to AN011)	Without channel dedicated sample and hold circuits	Conversion time*1	0.26	—	—	μs	• ADCLK: 50 MHz • Sampling time: 8 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
					Offset error	—	±1	±2.5	LSB	—
					Full-scale error	—	±1	±2.5	LSB	—
					Absolute accuracy	—	±2	±4	LSB	—
					DNL differential nonlinearity error *3	—	±0.75	±1	LSB	—
					INL integral nonlinearity error	—	±1.5	±2	LSB	—
			High-speed channels (AN000 to AN005) (AN006 to AN011)	With channel dedicated sample and hold circuits	Conversion time*2	1.72	—	—	μs	• ADCLK: 50 MHz • Sampling time of channel-dedicated sample-and-hold circuits: 63 ADCLK • Hold mode switching time of channel-dedicated sample-and-hold circuits: 2 ADCLK • Sampling time: 16 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less
					Offset error	—	±1.5	±6.75	LSB	—
					Full-scale error	—	±1.5	±6.75	LSB	—
					Absolute accuracy	—	±3.5	±9	LSB	—
					DNL differential nonlinearity error *3	—	±1	-1 to +1.5	LSB	—
					INL integral nonlinearity error	—	±2.5	±3.5	LSB	—

Note: These specification values are applicable when only one ADC16H is operating, DAC12 and ACMPHS are not operating, and there is no access to the external bus during A/D conversion.
If other ADC unit, DAC12, or ACMPHS is operating or bus access occurs during A/D conversion, values might not fall within the indicated ranges.
The use of ports 0 as digital outputs is not allowed when the ADC16H is used.
The characteristics apply when AVCC0, AVSS0, VREFH0, VREFH, VREFL0, VREFL, and ADC16H input voltage is stable.

Note 1. Without channel dedicated sample and hold circuits; The conversion time is the sum of the sampling time and the successive approximation time. Each of the above state is indicated for the test conditions.

Note 2. With channel-dedicated sample-and-hold circuits; The conversion time is the sum of the sampling time of channel-dedicated sample-and-hold circuits, the hold mode switching time, the sampling time and the successive approximation time. Each of the above state is indicated for the test conditions.

Note 3. DNL is measured using the Histogram Method, so the lower limit value is -1.

Table 62.95 A/D conversion characteristics (SAR mode : DCDC mode) (1 of 4)

Conditions: AVCC: 1.62 to 2.7 V, VCC: 1.62 to 2.7 V, VREFH0/VREFH: 1.62 V to AVCC

Parameter						Min	Typ	Max	Unit	Test conditions
SAR mode	Resolution					—	—	12	bit	—
SAR mode	Single-ended input	Normal mode	High-speed channels (AN000 to AN005) (AN006 to AN011)	Without channel dedicated sample and hold circuits	Conversion time[*1]	0.64	—	—	μs	• ADCLK: 50 MHz • Sampling time: 22 ADCLK • Successive approximation time: 10 ADCLK • Signal source impedance: 50 Ω or less
					Offset error	—	±3	±6.5	LSB	BGA package
						—	±3	±8	LSB	LQFP package
					Full-scale error	—	±3	±6.5	LSB	BGA package
						—	±3	±8	LSB	LQFP package
					Absolute accuracy	—	±5.5	±11	LSB	BGA package
						—	±5.5	±15	LSB	LQFP package
					DNL differential nonlinearity error [*2]	—	±1	-1 to +1.5	LSB	BGA package
						—	±1	-1 to +2.0	LSB	LQFP package
					INL integral nonlinearity error	—	±2	±3	LSB	BGA package
						—	±2	±3.5	LSB	LQFP package
SAR mode	Single-ended input	High accuracy mode	High-speed channels (AN000 to AN005) (AN006 to AN011)	Without channel dedicated sample and hold circuits	Conversion time[*1]	1	—	—	μs	• ADCLK: 50 MHz • Sampling time: 40 ADCLK • Successive approximation time: 10 ADCLK • Signal source impedance: 50 Ω or less
					Offset error	—	±1.5	±4.5	LSB	BGA package
						—	±1.5	±5	LSB	LQFP package
					Full-scale error	—	±1.5	±4.5	LSB	BGA package
						—	±1.5	±5	LSB	LQFP package
					Absolute accuracy	—	±5.0	±8	LSB	BGA package
						—	±5.0	±11	LSB	LQFP package
					DNL differential nonlinearity error [*2]	—	±1	-1 to +1.5	LSB	BGA package
						—	±1	-1 to +1.5	LSB	LQFP package
					INL integral nonlinearity error	—	±2	±3	LSB	BGA package
						—	±2	±3.5	LSB	LQFP package

Table 62.95 A/D conversion characteristics (SAR mode : DCDC mode) (2 of 4)

Conditions: AVCC: 1.62 to 2.7 V, VCC: 1.62 to 2.7 V, VREFH0/VREFH: 1.62 V to AVCC

Parameter					Min	Typ	Max	Unit	Test conditions
SAR mode	Single-ended input	Normal mode	Middle-speed channels (AN012 to AN015)	Conversion time[*1]	0.76	—	—	μs	• ADCLK: 50 MHz • Sampling time: 28 ADCLK • Successive approximation time: 10 ADCLK • Signal source impedance: 50 Ω or less
				Offset error	—	—	±6.5	LSB	BGA package
					—	±1.5	±7	LSB	LQFP package
				Full-scale error	—	±1.5	±6.5	LSB	BGA package
					—	±1.5	±7	LSB	LQFP package
				Absolute accuracy	—	±4	±11	LSB	BGA package
					—	±4	±15	LSB	LQFP package
				DNL differential nonlinearity error [*2]	—	±1	-1 to +1.5	LSB	BGA package
					—	±1	-1 to +2.0	LSB	LQFP package
				INL integral nonlinearity error	—	±2	±3	LSB	BGA package
					—	±2	±4	LSB	LQFP package
			Low-speed channels (AN016 to AN022)	Conversion time[*1]	1	—	—	μs	• ADCLK: 50 MHz • Sampling time: 40 ADCLK • Successive approximation time: 10 ADCLK • Signal source impedance: 50 Ω or less
				Offset error	—	±1.5	±6.5	LSB	—
				Full-scale error	—	±1.5	±6.5	LSB	—
				Absolute accuracy	—	±5.5	±11	LSB	—
				DNL differential nonlinearity error [*2]	—	±1	-1 to +1.5	LSB	—
				INL integral nonlinearity error	—	±2	±4	LSB	—

Table 62.95 A/D conversion characteristics (SAR mode : DCDC mode) (3 of 4)

Conditions: AVCC: 1.62 to 2.7 V, VCC: 1.62 to 2.7 V, VREFH0/VREFH: 1.62 V to AVCC

Parameter					Min	Typ	Max	Unit	Test conditions
SAR mode	Single-ended input	High accuracy mode	Middle-speed channels (AN012 to AN015)	Conversion time*1	1.4	—	—	μs	• ADCLK: 50 MHz • Sampling time: 60 ADCLK • Successive approximation time: 10 ADCLK • Signal source impedance: 50 Ω or less
				Offset error	—	±1.5	±4.5	LSB	BGA package
					—	±1.5	±5	LSB	LQFP package
				Full-scale error	—	±1.5	±4.5	LSB	BGA package
					—	±1.5	±5	LSB	LQFP package
				Absolute accuracy	—	±4	±8	LSB	BGA package
					—	±4	±13	LSB	LQFP package
				DNL differential nonlinearity error *2	—	±1	-1 to +1.5	LSB	BGA package
					—	±1	-1 to +2.0	LSB	LQFP package
				INL integral nonlinearity error	—	±2	±3	LSB	BGA package
					—	±2	±4	LSB	LQFP package
			Low-speed channels (AN016 to AN022)	Conversion time*1	1.88	—	—	μs	• ADCLK: 50 MHz • Sampling time: 84 ADCLK • Successive approximation time: 10 ADCLK • Signal source impedance: 50 Ω or less
				Offset error	—	±1.5	±4.5	LSB	—
				Full-scale error	—	±1.5	±4.5	LSB	—
				Absolute accuracy	—	±5.5	±8	LSB	—
				DNL differential nonlinearity error *2	—	±1	-1 to +1.5	LSB	—
				INL integral nonlinearity error	—	±2	±4	LSB	—
SAR mode	Differential input	Normal mode	High-speed channels (AN000 to AN005) (AN006 to AN011)	Conversion time*1	0.64	—	—	μs	• ADCLK: 50 MHz • Sampling time: 22 ADCLK • Successive approximation time: 10 ADCLK • Signal source impedance: 50 Ω or less
				Offset error	—	±2	±3.5	LSB	—
				Full-scale error	—	±2	±3.5	LSB	—
				Absolute accuracy	—	±4.5	±6	LSB	—
				DNL differential nonlinearity error *2	—	±0.75	±1	LSB	—
				INL integral nonlinearity error	—	±1.5	±2	LSB	—

Table 62.95 A/D conversion characteristics (SAR mode : DCDC mode) (4 of 4)

Conditions: AVCC: 1.62 to 2.7 V, VCC: 1.62 to 2.7 V, VREFH0/VREFH: 1.62 V to AVCC

Parameter						Min	Typ	Max	Unit	Test conditions
SAR mode	Differential input	High accuracy mode	High-speed channels (AN000 to AN005) (AN006 to AN011)	Without channel dedicated sample and hold circuits	Conversion time*1	1	—	—	μs	• ADCLK: 50 MHz • Sampling time: 40 ADCLK • Successive approximation time: 10 ADCLK • Signal source impedance: 50 Ω or less
					Offset error	—	±1	±2.5	LSB	—
					Full-scale error	—	±1	±2.5	LSB	—
					Absolute accuracy	—	±3.5	±4.5	LSB	—
					DNL differential nonlinearity error *2	—	±0.75	±1	LSB	—
					INL integral nonlinearity error	—	±1.5	±2	LSB	—

Note: These specification values are applicable when only one ADC16H is operating, DAC12 and ACMPHS are not operating, and there is no access to the external bus during A/D conversion.
If other ADC unit, DAC12, or ACMPHS is operating or bus access occurs during A/D conversion, values might not fall within the indicated ranges.
The use of ports 0 as digital outputs is not allowed when the ADC16H is used.
The characteristics apply when AVCC0, AVSS0, VREFH0, VREFH, VREFL0, VREFL, and ADC16H input voltage is stable.

Note 1. Without channel dedicated sample and hold circuits; The conversion time is the sum of the sampling time and the successive approximation time. Each of the above state is indicated for the test conditions.

Note 2. DNL is measured using the Histogram Method, so the lower limit value is -1.

Table 62.96 A/D conversion characteristics (SAR mode : External VDD mode) (1 of 4)

Conditions: AVCC: 1.62 to 2.7 V, VCC: 1.62 to 2.7 V, VREFH0/VREFH: 1.62 V to AVCC

Parameter						Min	Typ	Max	Unit	Test conditions
SAR mode	Resolution					—	—	12	bit	—
SAR mode	Single-ended input	Normal mode	High-speed channels (AN000 to AN005) (AN006 to AN011)	Without channel dedicated sample and hold circuits	Conversion time*1	0.64	—	—	μs	• ADCLK: 50 MHz • Sampling time: 22 ADCLK • Successive approximation time: 10 ADCLK • Signal source impedance: 50 Ω or less
					Offset error	—	±3	±6.5	LSB	—
					Full-scale error	—	±3	±6.5	LSB	—
					Absolute accuracy	—	±5.5	±11	LSB	—
					DNL differential nonlinearity error *2	—	±1	-1 to +1.5	LSB	—
					INL integral nonlinearity error	—	±2	±3	LSB	—
SAR mode	Single-ended input	High accuracy mode	High-speed channels (AN000 to AN005) (AN006 to AN011)	Without channel dedicated sample and hold circuits	Conversion time*1	1	—	—	μs	• ADCLK: 50 MHz • Sampling time: 40 ADCLK • Successive approximation time: 10 ADCLK • Signal source impedance: 50 Ω or less
					Offset error	—	±1.5	±4.5	LSB	—
					Full-scale error	—	±1.5	±4.5	LSB	—
					Absolute accuracy	—	±5.0	±8	LSB	—
					DNL differential nonlinearity error *2	—	±1	-1 to +1.5	LSB	—
					INL integral nonlinearity error	—	±2	±3	LSB	—

Table 62.96 A/D conversion characteristics (SAR mode : External VDD mode) (2 of 4)

Conditions: AVCC: 1.62 to 2.7 V, VCC: 1.62 to 2.7 V, VREFH0/VREFH: 1.62 V to AVCC

Parameter					Min	Typ	Max	Unit	Test conditions
SAR mode	Single-ended input	Normal mode	Middle-speed channels (AN012 to AN015)	Conversion time*1	0.76	—	—	μs	• ADCLK: 50 MHz • Sampling time: 28 ADCLK • Successive approximation time: 10 ADCLK • Signal source impedance: 50 Ω or less
				Offset error	—	±1.5	±6.5	LSB	—
				Full-scale error	—	±1.5	±6.5	LSB	—
				Absolute accuracy	—	±4	±11	LSB	—
				DNL differential nonlinearity error *2	—	±1	-1 to +1.5	LSB	—
				INL integral nonlinearity error	—	±2	±3	LSB	—
			Low-speed channels (AN016 to AN022)	Conversion time*1	1	—	—	μs	• ADCLK: 50 MHz • Sampling time: 40 ADCLK • Successive approximation time: 10 ADCLK • Signal source impedance: 50 Ω or less
				Offset error	—	±1.5	±6.5	LSB	—
				Full-scale error	—	±1.5	±6.5	LSB	—
				Absolute accuracy	—	±5.5	±11	LSB	—
				DNL differential nonlinearity error *2	—	±1	-1 to +1.5	LSB	—
				INL integral nonlinearity error	—	±2	±4	LSB	—

Table 62.96 A/D conversion characteristics (SAR mode : External VDD mode) (3 of 4)

Conditions: AVCC: 1.62 to 2.7 V, VCC: 1.62 to 2.7 V, VREFH0/VREFH: 1.62 V to AVCC

Parameter						Min	Typ	Max	Unit	Test conditions
SAR mode	Single-ended input	High accuracy mode	Middle-speed channels (AN012 to AN015)		Conversion time*1	1.4	—	—	μs	• ADCLK: 50 MHz • Sampling time: 60 ADCLK • Successive approximation time: 10 ADCLK • Signal source impedance: 50 Ω or less
SAR mode	Single-ended input	High accuracy mode	Middle-speed channels (AN012 to AN015)		Offset error	—	±1.5	±4.5	LSB	—
SAR mode	Single-ended input	High accuracy mode	Middle-speed channels (AN012 to AN015)		Full-scale error	—	±1.5	±4.5	LSB	—
SAR mode	Single-ended input	High accuracy mode	Middle-speed channels (AN012 to AN015)		Absolute accuracy	—	±4	±8	LSB	—
SAR mode	Single-ended input	High accuracy mode	Middle-speed channels (AN012 to AN015)		DNL differential nonlinearity error *2	—	±1	-1 to +1.5	LSB	—
SAR mode	Single-ended input	High accuracy mode	Middle-speed channels (AN012 to AN015)		INL integral nonlinearity error	—	±2	±3	LSB	—
SAR mode	Single-ended input	High accuracy mode	Low-speed channels (AN016 to AN022)		Conversion time*1	1.88	—	—	μs	• ADCLK: 50 MHz • Sampling time: 84 ADCLK • Successive approximation time: 10 ADCLK • Signal source impedance: 50 Ω or less
SAR mode	Single-ended input	High accuracy mode	Low-speed channels (AN016 to AN022)		Offset error	—	±1.5	±4.5	LSB	—
SAR mode	Single-ended input	High accuracy mode	Low-speed channels (AN016 to AN022)		Full-scale error	—	±1.5	±4.5	LSB	—
SAR mode	Single-ended input	High accuracy mode	Low-speed channels (AN016 to AN022)		Absolute accuracy	—	±5.5	±8	LSB	—
SAR mode	Single-ended input	High accuracy mode	Low-speed channels (AN016 to AN022)		DNL differential nonlinearity error *2	—	±1	-1 to +1.5	LSB	—
SAR mode	Single-ended input	High accuracy mode	Low-speed channels (AN016 to AN022)		INL integral nonlinearity error	—	±2	±4	LSB	—
SAR mode	Differential input	Normal mode	High-speed channels (AN000 to AN005) (AN006 to AN011)	Without channel dedicated sample and hold circuits	Conversion time*1	0.64	—	—	μs	• ADCLK: 50 MHz • Sampling time: 22 ADCLK • Successive approximation time: 10 ADCLK • Signal source impedance: 50 Ω or less
SAR mode	Differential input	Normal mode	High-speed channels (AN000 to AN005) (AN006 to AN011)	Without channel dedicated sample and hold circuits	Offset error	—	±2	±3.5	LSB	—
SAR mode	Differential input	Normal mode	High-speed channels (AN000 to AN005) (AN006 to AN011)	Without channel dedicated sample and hold circuits	Full-scale error	—	±2	±3.5	LSB	—
SAR mode	Differential input	Normal mode	High-speed channels (AN000 to AN005) (AN006 to AN011)	Without channel dedicated sample and hold circuits	Absolute accuracy	—	±4.5	±6	LSB	—
SAR mode	Differential input	Normal mode	High-speed channels (AN000 to AN005) (AN006 to AN011)	Without channel dedicated sample and hold circuits	DNL differential nonlinearity error *2	—	±0.75	±1	LSB	—
SAR mode	Differential input	Normal mode	High-speed channels (AN000 to AN005) (AN006 to AN011)	Without channel dedicated sample and hold circuits	INL integral nonlinearity error	—	±1.5	±2	LSB	—

Table 62.96 A/D conversion characteristics (SAR mode : External VDD mode) (4 of 4)

Conditions: AVCC: 1.62 to 2.7 V, VCC: 1.62 to 2.7 V, VREFH0/VREFH: 1.62 V to AVCC

Parameter						Min	Typ	Max	Unit	Test conditions
SAR mode	Differential input	High accuracy mode	High-speed channels (AN000 to AN005) (AN006 to AN011)	Without channel dedicated sample and hold circuits	Conversion time*1	1	—	—	μs	• ADCLK: 50 MHz • Sampling time: 40 ADCLK • Successive approximation time: 10 ADCLK • Signal source impedance: 50 Ω or less
					Offset error	—	±1	±2.5	LSB	—
					Full-scale error	—	±1	±2.5	LSB	—
					Absolute accuracy	—	±3.5	±4.5	LSB	—
					DNL differential nonlinearity error *2	—	±0.75	±1	LSB	—
					INL integral nonlinearity error	—	±1.5	±2	LSB	—

Note: These specification values are applicable when only one ADC16H is operating, DAC12 and ACMPHS are not operating, and there is no access to the external bus during A/D conversion.
If other ADC unit, DAC12, or ACMPHS is operating or bus access occurs during A/D conversion, values might not fall within the indicated ranges.
The use of ports 0 as digital outputs is not allowed when the ADC16H is used.
The characteristics apply when AVCC0, AVSS0, VREFH0, VREFH, VREFL0, VREFL, and ADC16H input voltage is stable.

Note 1. Without channel dedicated sample and hold circuits; The conversion time is the sum of the sampling time and the successive approximation time. Each of the above state is indicated for the test conditions.

Note 2. DNL is measured using the Histogram Method, so the lower limit value is -1.

Table 62.97 A/D conversion characteristics (Oversampling mode and Hybrid mode) (1)

Parameter				Min	Typ	Max	Unit	Test conditions
Oversampling mode and Hybrid mode	Resolution			—	—	16	bit	—
	Oversampling period	Oversampling mode		0.16	—	—	μs	• ADCLK: 50 MHz • Sampling time: 3 ADCLK • Successive approximation time: 5 ADCLK • Without disconnection detection assist function • Signal source impedance: 50 Ω or less
		Hybrid mode*2		0.18	—	—	μs	• ADCLK: 50 MHz • Sampling time: 8 ADCLK • Successive approximation time: 5 ADCLK • Without disconnection detection assist function • Signal source impedance: 50 Ω or less
	Digital filter characteristics*1	Sinc filter	Initial delay	—	22	—	/Fos	—
			Group delay	—	11	—		—
			Normalized Cutoff Frequency	—	0.033	—	Fin/Fos	—

Note: Fos is oversampling frequency.
When in Hybrid mode, Fos is 1/ (the sum of the oversampling periods of each analog channel assigned to the scan group).

Note 1. See Figure 62.135.

Note 2. Value per channel.

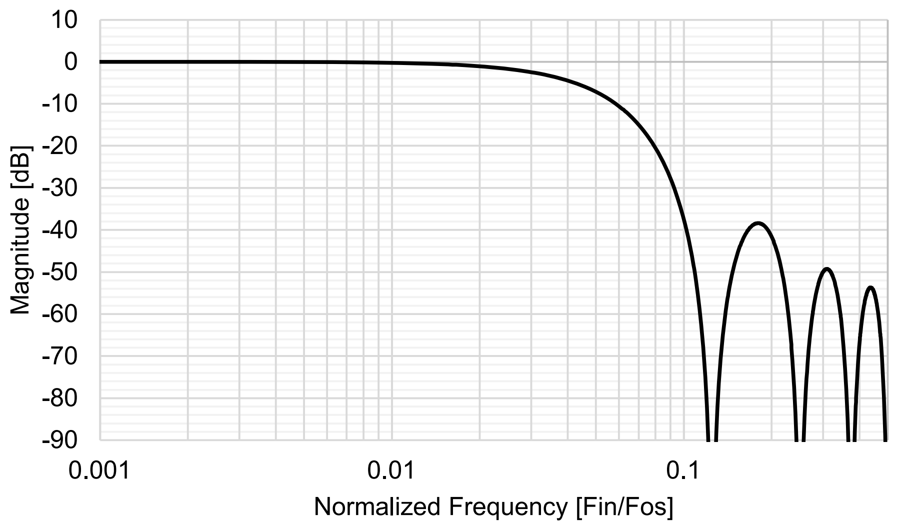


Figure 62.135 Digital filter characteristics (Sinc filter)

Table 62.98 A/D conversion characteristics (Oversampling mode and Hybrid mode) (2)

Conditions: AVCC: 2.7 to 3.63 V, VCC: 2.7 to 3.63 V, VREFH0/VREFH: 2.7 V to AVCC

Parameter				Min	Typ	Max	Unit	Test conditions
Oversampling mode and Hybrid mode	Sinc filter	Single-ended input (AN000 to AN005) (AN006 to AN011) (AN012 to AN015)	SNDR signal-to-noise and distortion ratio	—	80	—	dB	• ADCLK: 50 MHz • Sampling time: – High-speed channels (Oversampling mode): 3 ADCLK – High-speed channels (Hybrid mode): 8 ADCLK – Middle-speed channels (Oversampling mode): 10 ADCLK – Middle-speed channels (Hybrid mode): 22 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less • Input frequency: – Oversampling mode: 5 kHz – Hybrid mode: 5 kHz • Without channel dedicated sample and hold circuits
			ENOB effective number of bits	—	13	—	bit	
		Differential input (AN000 to AN005) (AN006 to AN011)	SNDR signal-to-noise and distortion ratio	—	86	—	dB	
			ENOB effective number of bits	—	14	—	bit	

Table 62.99 A/D conversion characteristics (Oversampling mode and Hybrid mode) (3)

Conditions: AVCC: 1.62 to 2.7 V, VCC: 1.62 to 2.7 V, VREFH0/VREFH: 1.62 V to AVCC

Parameter				Min	Typ	Max	Unit	Test conditions
Oversampling mode and Hybrid mode	Sinc filter	Single-ended input (AN000 to AN005) (AN006 to AN011) (AN012 to AN015)	SNDR signal-to-noise and distortion ratio	—	74	—	dB	• ADCLK: 50 MHz • Sampling time: – High-speed channels (Oversampling mode): 22 ADCLK – High-speed channels (Hybrid mode): 40 ADCLK – Middle-speed channels (Oversampling mode): 28 ADCLK – Middle-speed channels (Hybrid mode): 60 ADCLK • Successive approximation time: 10 ADCLK • Signal source impedance: 50 Ω or less • Input frequency: – Oversampling mode: 5 kHz – Hybrid mode: 5 kHz • Without channel dedicated sample and hold circuits
			ENOB effective number of bits	—	12	—	bit	
		Differential input (AN000 to AN005) (AN006 to AN011)	SNDR signal-to-noise and distortion ratio	—	80	—	dB	
			ENOB effective number of bits	—	13	—	bit	

Table 62.100 A/D conversion characteristics (Oversampling mode)

Conditions: AVCC: 2.7 to 3.63 V, VCC: 2.7 to 3.63 V, VREFH0/VREFH: 2.7 V to AVCC

Parameter				Min	Typ	Max	Unit	Test conditions
Oversampling mode	Single-ended input	High-speed channels (AN000 to AN005) (AN006 to AN011)	Offset error *3	—	±0.5	±4	LSB	• ADCLK: 50 MHz • Sampling time: 3 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less • Digital filter: Sinc filter
			Gain error (Single/ Continuous mode) *3	—	±1	±4	LSB	
			Gain error (One-channel continuous mode) *3	—	±1	±5	LSB	
			DNL differential nonlinearity error*1 *2	—	-1 to +1.5	-1 to +2.5	LSB	
			INL integral nonlinearity error*1	—	±4	±8	LSB	
		Middle-speed channels (AN012 to AN015)	Offset error *3	—	±0.5	±4	LSB	• ADCLK: 50 MHz • Sampling time: 10 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less • Digital filter: Sinc filter
			Gain error *3	—	±1	±4	LSB	
			DNL differential nonlinearity error*1 *2	—	-1 to +2	-1 to +4	LSB	
			INL integral nonlinearity error*1	—	±4	±8	LSB	
		Low-speed channels (AN016 to AN022)	Offset error *3	—	±0.5	±4	LSB	• ADCLK: 50 MHz • Sampling time: 20 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less • Digital filter: Sinc filter
			Gain error *3	—	±1	±4	LSB	
			DNL differential nonlinearity error*1 *2	—	-1 to +2	-1 to +4	LSB	
			INL integral nonlinearity error*1	—	±4	±12	LSB	
	Differential input	High-speed channels (AN000 to AN005) (AN006 to AN011)	Offset error *3	—	±0.25	±2	LSB	• ADCLK: 50 MHz • Sampling time: 3 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less • Digital filter: Sinc filter
			Gain error (Single/ Continuous mode) *3	—	±0.5	±2	LSB	
			Gain error (One-channel continuous mode) *3	—	±0.5	±2.5	LSB	
			DNL differential nonlinearity error*1 *2	—	-1 to +1.5	-1 to +2.0	LSB	
			INL integral nonlinearity error*1	—	±3	±6	LSB	

Note: These specification values are applicable when only one ADC16H is operating, DAC12 and ACMPHS are not operating, and there is no access to the external bus during A/D conversion.
If other ADC unit, DAC12, or ACMPHS is operating or bus access occurs during A/D conversion, values might not fall within the indicated ranges.
The use of ports 0 as digital outputs is not allowed when the ADC16H is used.
The characteristics apply when AVCC0, AVSS0, VREFH0, VREFH, VREFL0, VREFL, and ADC16H input voltage is stable.

Note 1. Test conditions: 0.2% to 99.8% of the analog input voltage range.

Note 2. DNL is measured using the Histogram Method, so the lower limit value is -1.

Note 3. This value is based on a 12-bit resolution.

Table 62.101 A/D conversion characteristics (Oversampling mode)

Conditions: AVCC: 1.62 to 2.7 V, VCC: 1.62 to 2.7 V, VREFH0/VREFH: 1.62 V to AVCC

Parameter				Min	Typ	Max	Unit	Test conditions
Oversampling mode	Single-ended input	High-speed channels (AN000 to AN005) (AN006 to AN011)	Offset error *3	—	±0.5	±4	LSB	• ADCLK: 50 MHz • Sampling time: 22 ADCLK • Successive approximation time: 10 ADCLK • Signal source impedance: 50 Ω or less • Digital filter: Sinc filter
			Gain error (Single/ Continuous mode) *3	—	±1	±4	LSB	
			Gain error (One-channel continuous mode) *3	—	±1	±5	LSB	
			DNL differential nonlinearity error *1 *2	—	-1 to +2	-1 to +2.5	LSB	
			INL integral nonlinearity error *1	—	±4	±8	LSB	
		Middle-speed channels (AN012 to AN015)	Offset error *3	—	±0.5	±4	LSB	• ADCLK: 50 MHz • Sampling time: 28 ADCLK • Successive approximation time: 10 ADCLK • Signal source impedance: 50 Ω or less • Digital filter: Sinc filter
			Gain error *3	—	±1	±4	LSB	
			DNL differential nonlinearity error *1 *2	—	-1 to +2	-1 to +4	LSB	
			INL integral nonlinearity error *1	—	±4	±8	LSB	
		Low-speed channels (AN016 to AN022)	Offset error *3	—	±0.5	±4	LSB	• ADCLK: 50 MHz • Sampling time: 40 ADCLK • Successive approximation time: 10 ADCLK • Signal source impedance: 50 Ω or less • Digital filter: Sinc filter
			Gain error *3	—	±1	±4	LSB	
			DNL differential nonlinearity error *1 *2	—	-1 to +2	-1 to +4	LSB	
			INL integral nonlinearity error *1	—	±4	±12	LSB	
	Differential input	High-speed channels (AN000 to AN005) (AN006 to AN011)	Offset error *3	—	±0.25	±2	LSB	• ADCLK: 50 MHz • Sampling time: 22 ADCLK • Successive approximation time: 10 ADCLK • Signal source impedance: 50 Ω or less • Digital filter: Sinc filter
			Gain error (Single/ Continuous mode) *3	—	±0.5	±2	LSB	
			Gain error (One-channel continuous mode) *3	—	±0.5	±2.5	LSB	
			DNL differential nonlinearity error *1 *2	—	-1 to +2	-1 to +2.5	LSB	
			INL integral nonlinearity error *1	—	±3	±6	LSB	

Note: These specification values are applicable when only one ADC16H is operating, DAC12 and ACMPHS are not operating, and there is no access to the external bus during A/D conversion.
If other ADC unit, DAC12, or ACMPHS is operating or bus access occurs during A/D conversion, values might not fall within the indicated ranges.
The use of ports 0 as digital outputs is not allowed when the ADC16H is used.
The characteristics apply when AVCC0, AVSS0, VREFH0, VREFH, VREFL0, VREFL, and ADC16H input voltage is stable.

Note 1. Test conditions: 0.2% to 99.8% of the analog input voltage range.
Note 2. DNL is measured using the Histogram Method, so the lower limit value is -1.
Note 3. This value is based on a 12-bit resolution.

Table 62.102 A/D conversion characteristics (Hybrid mode)

Conditions: AVCC: 2.7 to 3.63 V, VCC: 2.7 to 3.63 V, VREFH0/VREFH: 2.7 V to AVCC

Parameter					Min	Typ	Max	Unit	Test conditions
Hybrid mode	Without channel dedicated sample and hold circuits	Single-ended input	High-speed channels (AN000 to AN005) (AN006 to AN011)	Offset error *4	—	±0.5	±4	LSB	• ADCLK: 50 MHz • Sampling time: 8 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less • Digital filter: Sinc filter
				Gain error *4	—	±1	±5	LSB	
				DNL differential nonlinearity error *2 *3	—	-1 to +1.5	-1 to +2.5	LSB	
				INL integral nonlinearity error *2	—	±4	±8	LSB	
			Middle-speed channels (AN012 to AN015) *1	Offset error *4	—	±0.5	±4	LSB	• ADCLK: 50 MHz • Sampling time: 22 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less • Digital filter: Sinc filter
				Gain error *4	—	±1	±5	LSB	
				DNL differential nonlinearity error *2 *3	—	-1 to +2	-1 to +4	LSB	
				INL integral nonlinearity error *2	—	±4	±8	LSB	
			Low-speed channels (AN016 to AN022) *1	Offset error *4	—	±0.5	±4	LSB	• ADCLK: 50 MHz • Sampling time: 42 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less • Digital filter: Sinc filter
				Gain error *4	—	±1	±5	LSB	
				DNL differential nonlinearity error *2*3	—	-1 to +2	-1 to +4	LSB	
				INL integral nonlinearity error *2	—	±4	±12	LSB	
		Differential input	High-speed channels (AN000 to AN005) (AN006 to AN011)	Offset error *4	—	±0.25	±2	LSB	• ADCLK: 50 MHz • Sampling time: 8 ADCLK • Successive approximation time: 5 ADCLK • Signal source impedance: 50 Ω or less • Digital filter: Sinc filter
				Gain error *4	—	±0.5	±2.5	LSB	
				DNL differential nonlinearity error *2 *3	—	-1 to +1.5	-1 to +2	LSB	
				INL integral nonlinearity error *2	—	±3	±6	LSB	
	With channel dedicated sample and hold circuits	Single-ended input	High-speed channels (AN000 to AN005) (AN006 to AN011)	Offset error *4	—	±0.5	±4	LSB	• ADCLK: 50 MHz • Sampling time of channel-dedicated sample-and-hold circuits: * ADCLK • Hold mode switching time of channel dedicated sample-and-hold circuits: * ADCLK • Sampling time: * ADCLK • Successive approximation time: * ADCLK • Signal source impedance: 50 Ω or less • Digital filter: Sinc filter
				Gain error *4	—	±0.5	±4	LSB	
				DNL differential nonlinearity error *2 *3	—	±1	-1 to +2	LSB	
				INL integral nonlinearity error *2	—	±12	±16	LSB	
		Differential input	High-speed channels (AN000 to AN005) (AN006 to AN011)	Offset error *4	—	±0.5	±4	LSB	• ADCLK: 50 MHz • Sampling time of channel-dedicated sample-and-hold circuits: * ADCLK • Hold mode switching time of channel dedicated sample-and-hold circuits: * ADCLK • Sampling time: * ADCLK • Successive approximation time: * ADCLK • Signal source impedance: 50 Ω or less • Digital filter: Sinc filter
				Gain error *4	—	±0.5	±4	LSB	
				DNL differential nonlinearity error *2 *3	—	±1	-1 to +2	LSB	
				INL integral nonlinearity error *2	—	±4	±16	LSB	

Note: These specification values are applicable when only one ADC16H is operating, DAC12 and ACMPHS are not operating, and there is no access to the external bus during A/D conversion.
If other ADC unit, DAC12, or ACMPHS is operating or bus access occurs during A/D conversion, values might not fall within the indicated ranges.
The use of ports 0 as digital outputs is not allowed when the ADC16H is used.
The characteristics apply when AVCC0, AVSS0, VREFH0, VREFH, VREFL0, VREFL, and ADC16H input voltage is stable.
Note 1. Channel-dedicated sample-and-hold circuits are not available in these channels.
Note 2. Test conditions: 0.2% to 99.8% of the analog input voltage range.
Note 3. DNL is measured using the Histogram Method, so the lower limit value is -1.
Note 4. This value is based on a 12-bit resolution.

Table 62.103 A/D conversion characteristics (Hybrid mode)

Conditions: AVCC: 1.62 to 2.7 V, VCC: 1.62 to 2.7 V, VREFH0/VREFH: 1.62 V to AVCC

Parameter					Min	Typ	Max	Unit	Test conditions
Hybrid mode	Without channel dedicated sample and hold circuits	Single-ended input	High-speed channels (AN000 to AN005) (AN006 to AN011)	Offset error *3	—	±0.5	±4	LSB	• ADCLK: 50 MHz • Sampling time: 40 ADCLK • Successive approximation time: 10 ADCLK • Signal source impedance: 50 Ω or less • Digital filter: Sinc filter
				Gain error *3	—	±1	±5	LSB	
				DNL differential nonlinearity error *1 *2	—	-1 to +2	-1 to +2.5	LSB	
				INL integral nonlinearity error *1	—	±4	±8	LSB	
			Middle-speed channels (AN012 to AN015)	Offset error *3	—	±0.5	±4	LSB	• ADCLK: 50 MHz • Sampling time: 60 ADCLK • Successive approximation time: 10 ADCLK • Signal source impedance: 50 Ω or less • Digital filter: Sinc filter
				Gain error *3	—	±1	±5	LSB	
				DNL differential nonlinearity error *1 *2	—	-1 to +2	-1 to +4	LSB	
				INL integral nonlinearity error *1	—	±4	±8	LSB	
			Low-speed channels (AN016 to AN022)	Offset error *3	—	±0.5	±4	LSB	• ADCLK: 50 MHz • Sampling time: 84 ADCLK • Successive approximation time: 10 ADCLK • Signal source impedance: 50 Ω or less • Digital filter: Sinc filter
				Gain error *3	—	±1	±5	LSB	
				DNL differential nonlinearity error *1 *2	—	-1 to +2	-1 to +4	LSB	
				INL integral nonlinearity error *1	—	±4	±12	LSB	
		Differential input	High-speed channels (AN000 to AN005) (AN006 to AN011)	Offset error *3	—	±0.25	±2	LSB	• ADCLK: 50 MHz • Sampling time: 40 ADCLK • Successive approximation time: 10 ADCLK • Signal source impedance: 50 Ω or less • Digital filter: Sinc filter
				Gain error *3	—	±0.5	±2.5	LSB	
				DNL differential nonlinearity error *1 *2	—	-1 to +2	-1 to +2.5	LSB	
				INL integral nonlinearity error *1	—	±3	±6	LSB	

Note: These specification values are applicable when only one ADC16H is operating, DAC12 and ACMPHS are not operating, and there is no access to the external bus during A/D conversion.
If other ADC unit, DAC12, or ACMPHS is operating or bus access occurs during A/D conversion, values might not fall within the indicated ranges.
The use of ports 0 as digital outputs is not allowed when the ADC16H is used.
The characteristics apply when AVCC0, AVSS0, VREFH0, VREFH, VREFL0, VREFL, and ADC16H input voltage is stable.
Note 1. Test conditions: 0.2% to 99.8% of the analog input voltage range.
Note 2. DNL is measured using the Histogram Method, so the lower limit value is -1.
Note 3. This value is based on a 12-bit resolution.

Table 62.104 A/D internal reference voltage characteristics (1 of 2)

Parameter	Min	Typ	Max	Unit	Test conditions
A/D internal reference voltage	0.77	0.8	0.84	V	—

Table 62.104 A/D internal reference voltage characteristics (2 of 2)

Parameter	Min	Typ	Max	Unit	Test conditions
Sampling time	4.15	—	—	μs	—

Table 62.105 A/D conversion characteristics of D/A output

Parameter	Min	Typ	Max	Unit	Test conditions
Sampling time	1	—	—	μs	—

62.6 DAC12 Characteristics

Table 62.106 D/A conversion characteristics

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Resolution		—	—	—	12	Bits	—
INL	VREFH ≥ 2.7V	—	—	±2.0	±4.0	LSB	—
	VREFH < 2.7V	—	—	±4.0	±8.0		
DNL	VREFH ≥ 2.7V	—	—	±0.5	±1.0	LSB	—
	VREFH < 2.7V	—	—	±1.0	±2.0		
Conversion time	VREFH ≥ 2.7V	t_{DCONV1}, t_{DCONV2}	—	—	3.5	μs	—
	VREFH < 2.7V		—	—	6		
Output destination switching time	VREFH ≥ 2.7V	$t_{DSLPUP1}$, $t_{DSLPUP2}$	—	—	3.5	μs	—
	VREFH < 2.7V		—	—	6		
Buffer preparation time	VREFH ≥ 2.7V	t_{DISOUT}	—	—	3.5	μs	—
	VREFH < 2.7V		—	—	6		
Setup time		t_{SU}	—	—	4	ns	—
Resistive load		—	5	—	—	kΩ	—
Capacitive load		—	—	—	50	pF	—
Output voltage range	VREFH ≥ 2.7V	—	0.20	—	VREFH – 0.20	V	—
	VREFH < 2.7V	—	0.34	—	VREFH – 0.34		

62.7 TSN Characteristics

Table 62.107 TSN characteristics

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Relative accuracy	—	-1.0	—	1.0	°C	A/D converter error is not included.
Temperature slope	—	—	2.7	—	mV/°C	—
Output voltage (at 25 °C)	—	—	0.83	—	V	—
Temperature sensor stabilization time	t_{TSTBL}	—	—	30	μs	—
Comparator stabilization time	t_{RSTBL}	—	—	30	μs	—
Sampling time	—	4.15	—	—	μs	—

62.8 OSC Stop Detect Characteristics

Table 62.108 Oscillation stop detection circuit characteristics

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Detection time	t_{dr}	—	—	1	ms	Figure 62.136

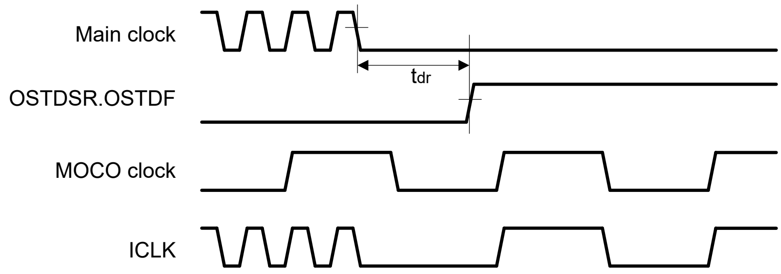


Figure 62.136 Oscillation stop detection timing

Table 62.109 Sub-clock oscillation stop detection circuit characteristics

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Wait time for startup	t_{sosdup}	100	—	—	μs	Figure 62.137
Detection time	t_{dr}	—	—	2	ms	Figure 62.138

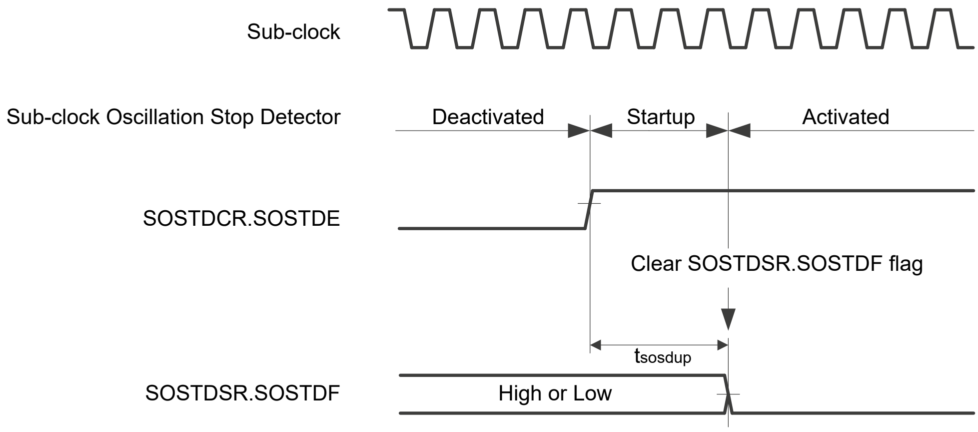


Figure 62.137 Sub-clock Oscillation Stop Detector start-up time

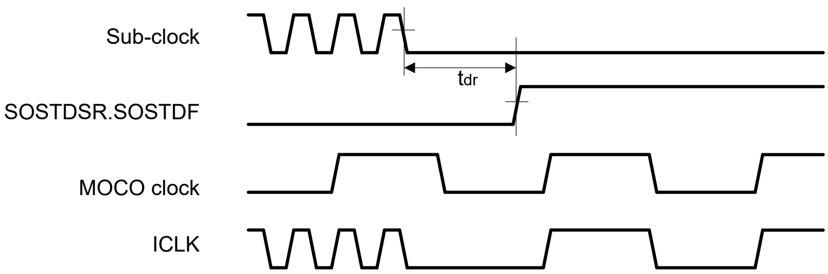


Figure 62.138 Sub-clock oscillation stop detection timing

62.9 POR and PVD Characteristics

Table 62.110 Power-on reset circuit and voltage detection circuit characteristics (1 of 2)

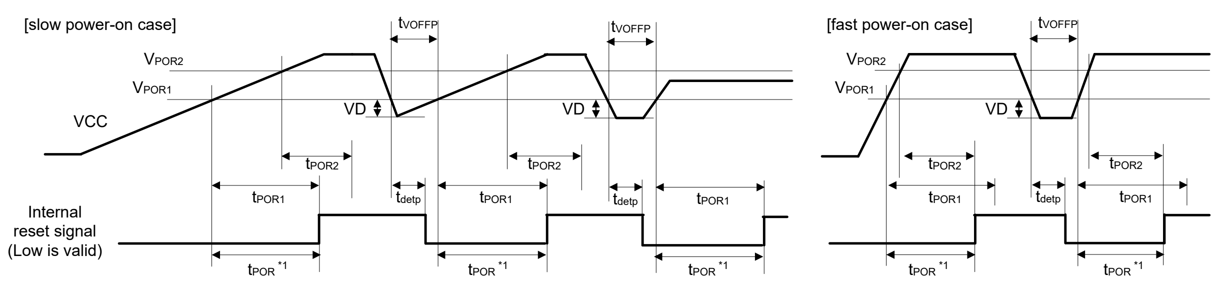
Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Voltage detection level	Power-on reset (POR)	V_{POR1}	1.52	1.56	1.61	V	Figure 62.139
		V_{POR2}	—	—	1.73		
	Voltage detection circuit (PVD0)	V_{det0_0}	2.76	2.85	2.94		Figure 62.140
		V_{det0_1}	2.50	2.58	2.66		
		V_{det0_2}	2.08	2.15	2.22		
		V_{det0_3}	1.93	2.00	2.07		
		V_{det0_4}	1.84	1.90	1.96		
		V_{det0_5}	1.74	1.80	1.86		
		V_{det0_6}	1.62	1.67	1.73		
		V_{det0_7}	1.51	1.56	1.61		
	Voltage detection circuit (PVDn) (n = 1, 2, 4, 5)	$V_{detn\_3\_rise}$	3.78	3.92	4.05		Figure 62.141
		$V_{detn\_3\_fall}$	3.72	3.86	3.99		
		$V_{detn\_4\_rise}$	3.09	3.20	3.30		
		$V_{detn\_4\_fall}$	3.03	3.14	3.24		
		$V_{detn\_5\_rise}$	3.05	3.16	3.26		
		$V_{detn\_5\_fall}$	2.99	3.10	3.20		
		$V_{detn\_6\_rise}$	3.03	3.14	3.24		
		$V_{detn\_6\_fall}$	2.97	3.08	3.18		
		$V_{detn\_7\_rise}$	2.81	2.91	3.00		
		$V_{detn\_7\_fall}$	2.75	2.85	2.94		
		$V_{detn\_8\_rise}$	2.79	2.89	2.98		
		$V_{detn\_8\_fall}$	2.73	2.83	2.92		
		$V_{detn\_9\_rise}$	2.76	2.86	2.95		
		$V_{detn\_9\_fall}$	2.70	2.80	2.89		
		$V_{detn\_10\_rise}$	2.58	2.67	2.75		
		$V_{detn\_10\_fall}$	2.53	2.62	2.70		
		$V_{detn\_11\_rise}$	2.30	2.38	2.46		
		$V_{detn\_11\_fall}$	2.25	2.33	2.41		
		$V_{detn\_12\_rise}$	1.88	1.94	2.00		
		$V_{detn\_12\_fall}$	1.84	1.90	1.96		
		$V_{detn\_13\_rise}$	1.84	1.90	1.96		
		$V_{detn\_13\_fall}$	1.80	1.86	1.92		
		$V_{detn\_14\_rise}$	1.72	1.78	1.84		
		$V_{detn\_14\_fall}$	1.68	1.74	1.80		
		$V_{detn\_15\_rise}$	1.66	1.72	1.77		
		$V_{detn\_15\_fall}$	1.62	1.68	1.73		

Table 62.110 Power-on reset circuit and voltage detection circuit characteristics (2 of 2)

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Internal reset time*1	Power-on reset time	t_{POR1}	—	—	6.7	ms	Figure 62.139
		t_{POR2}	—	—	1.6		
	PVD0 reset time	t_{PVD0}	—	—	*1		Figure 62.139
	PVD1 reset time	t_{PVD1}	—	—	*1		Figure 62.140
	PVD2 reset time	t_{PVD2}	—	—	*1		
	PVD4 reset time	t_{PVD4}	—	—	*1		
	PVD5 reset time	t_{PVD5}	—	—	*1		
Minimum VCC downtime (POR)*2	50mV < VD	t_{VOFFP}	900	—	—	μs	Figure 62.139
	VD ≤ 50mV		2000	—	—		
Minimum VCC downtime (PVD)*2	PVD0	t_{VOFF}	25	—	—	μs	Figure 62.140
	PVD1, PVD2, PVD4, PVD5		25	—	—		
Response delay time (POR)	50mV < VD	t_{detp}	—	—	900	μs	Figure 62.139
	VD ≤ 50mV		—	—	2000		
Response delay time (PVD)	PVD0	t_{det}	—	—	25	μs	Figure 62.140, Figure 62.141
	PVD1, PVD2, PVD4, PVD5		—	—	25		
PVDn operation stabilization time (after PVD is enabled) (n = 1, 2, 4, 5)		$T_{d\ (E-A)}$	—	—	20	μs	Figure 62.141

Note 1. The maximum value of t_{PVD0}, t_{PVD1}, t_{PVD2}, t_{PVD4}, t_{PVD5} are equal to t_{DSBY} because the internal reset time is maximized when returning from Deep Software Standby mode.

Note 2. The minimum VCC downtime indicates the time when VCC is below the minimum value of voltage detection levels V_{POR1}, V_{det0}, V_{det1}, V_{det2}, V_{det4} and V_{det5} for the POR / PVD.



Note 1. t_{POR} is determined by the earlier point in time when the t_{POR1} has passed after crossing V_{POR1} or when the t_{POR2} has passed after crossing V_{POR2}.

Figure 62.139 Power-on reset timing

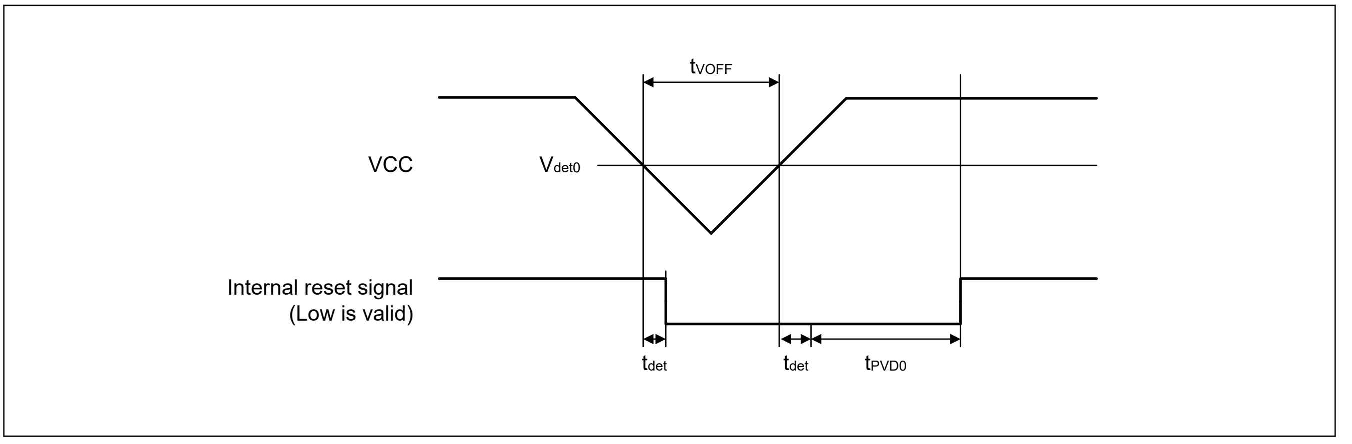


Figure 62.140 Voltage detection circuit timing (V_{det0})

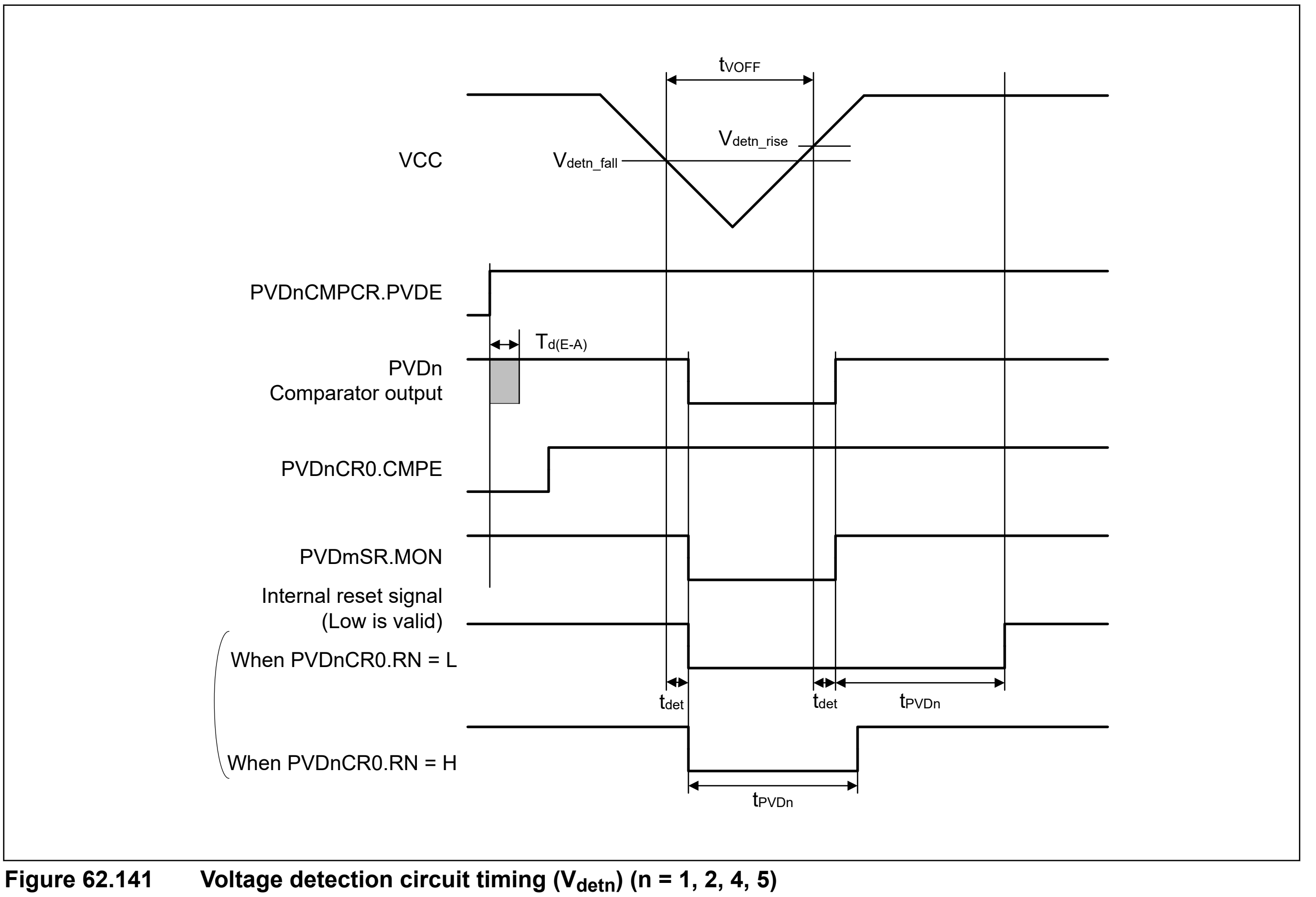


Figure 62.141 Voltage detection circuit timing (V_{detn}) (n = 1, 2, 4, 5)

62.10 External VDD Timing Characteristics

Table 62.111 External VDD timing characteristics

Parameter	Symbol	Min	Typ	Max	Unit	Test condition
Reset hold time at power up of external VDD with using RES pin	t_{EXTVRH}	600.00	—	—	µs	Figure 62.142 Figure 62.143
VDD rise time at power up of external VDD without using RES pin	t_{EXTVDD}	—	—	550	µs	

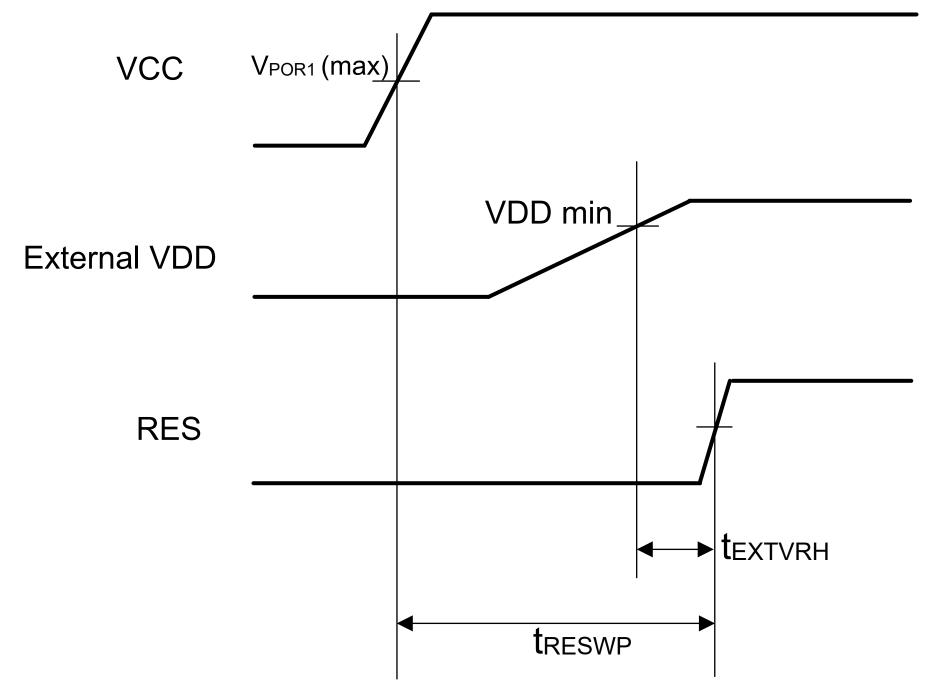


Figure 62.142 Power-up sequence of external VDD mode with using RES pin

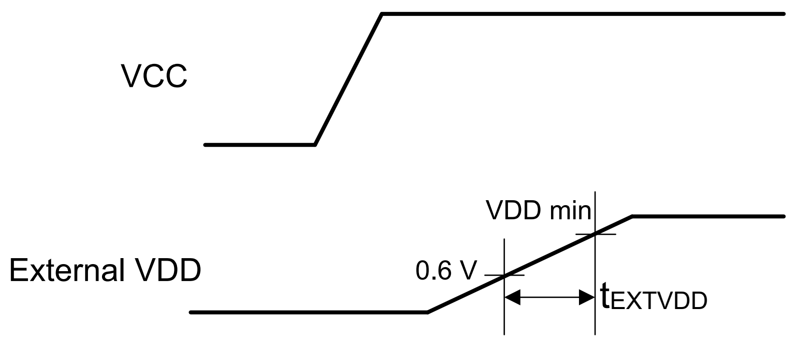


Figure 62.143 Power-up sequence of external VDD mode without using RES pin

62.11 Core Voltage Monitor Reset Characteristics

Table 62.112 Core voltage monitor reset characteristics

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
Voltage detection level	Core voltage monitor reset (CVMR)		V_{det_VDDH}	1.05	1.10	1.15	V	Figure 62.144
			V_{det_VDDL}	0.55	0.58	0.61		
Internal reset time	Core voltage monitor reset time	DCDC mode	t_{CVM}	—	—	0.18	ms	
		External VDD mode		—	—	2.6		
Minimum VDD down /up time (CVMR)			t_{CVMOFF}	45	—	—	µs	
Response delay time (CVMR)			t_{CVMdet}	—	—	45	µs	
Hysteresis width (CVMR)			V_{CVMH}	—	0.225	—	V	

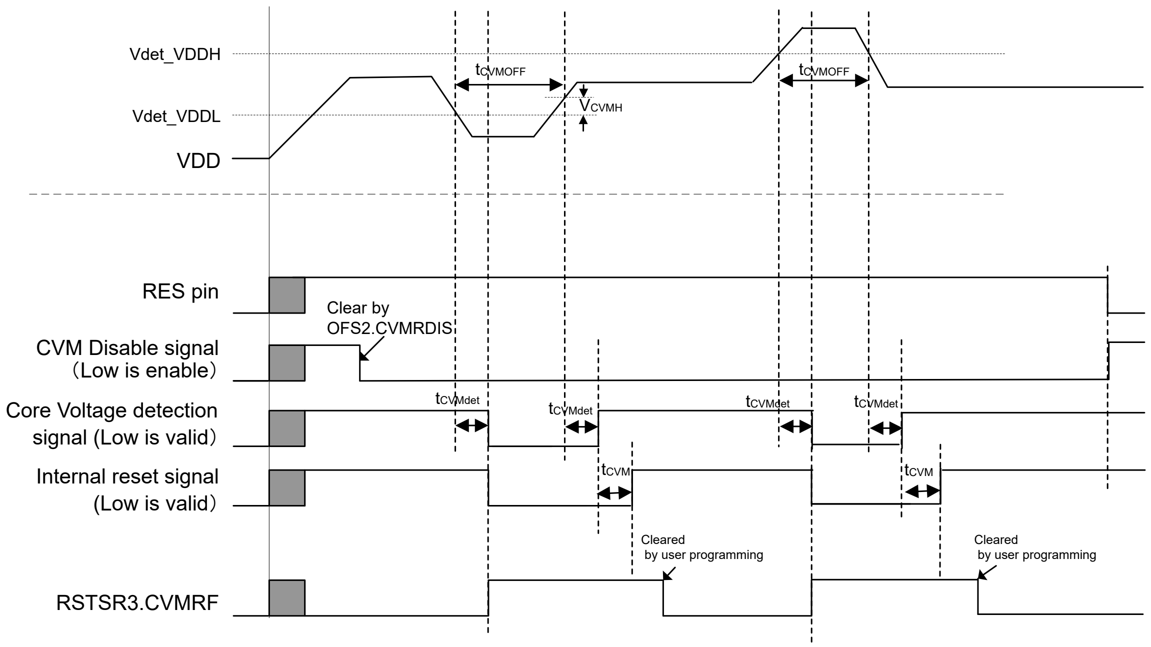


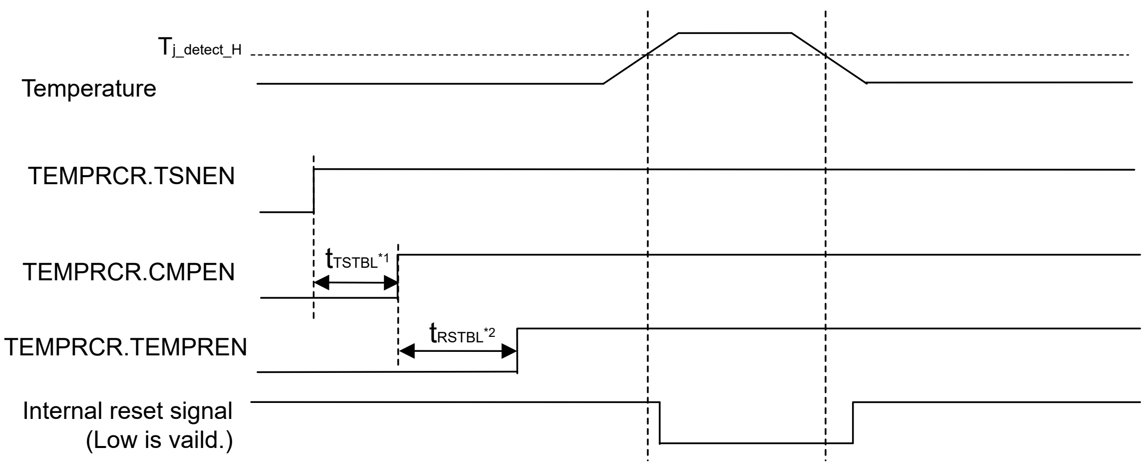
Figure 62.144 Timing of core voltage monitor reset

62.12 Temperature Monitor Reset Characteristics

Table 62.113 Temperature monitor reset characteristics

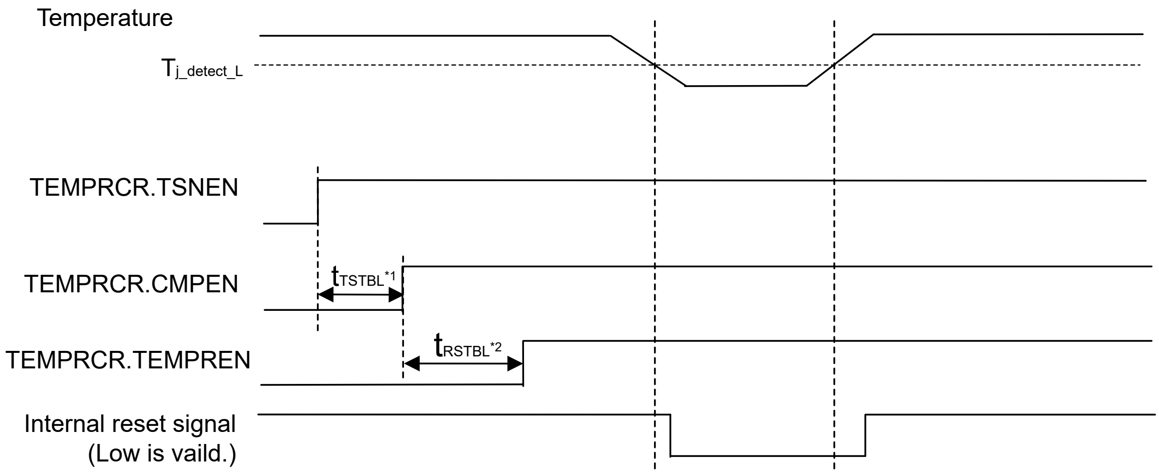
Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
High threshold temperature	$T_{j\_detect\_H}$	105	—	125	°C	Figure 62.145 Figure 62.146
Low threshold temperature	$T_{j\_detect\_L}$	-40	—	-20	°C	

Note: Temperature monitor reset is not supported in 0 to 95 °C product (product group A).



Note 1. t_{TSTBL} is the stabilization wait time of TSN.
Note 2. t_{TSTBL} is the stabilization wait time of the comparator in TSN.

Figure 62.145 Timing of temperature monitor reset (High-temperature detection)



Note 1. t_{TSTBL} is the stabilization wait time of TSN.
Note 2. t_{TSTBL} is the stabilization wait time of the comparator in TSN.

Figure 62.146 Timing of temperature monitor reset (Low-temperature detection)

62.13 VBATT Characteristics

Table 62.114 Battery backup function characteristics (1 of 2)

Conditions: VCC = VCC_DCDC = VCC_USB = 1.62 to 3.63 V, VBATT = 1.62 to 3.63 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Voltage level for switching to battery backup OFS1.PVDAS and PVDLPSEL are 0 in Deep Software Standby mode 1, 2 (VDETVATT_n follows VDSEL[2:0] setting for PVD0)	$V_{DETBATT\_0}$	2.760	2.850	2.940	V	Figure 62.147
	$V_{DETBATT\_1}$	2.500	2.580	2.660		
	$V_{DETBATT\_2}$	2.080	2.150	2.220		
	$V_{DETBATT\_3}$	1.935	2.000	2.065		
	$V_{DETBATT\_4}$	1.840	1.900	1.960		
	$V_{DETBATT\_5}$	1.740	1.800	1.860		
	$V_{DETBATT\_6}$	1.620	1.670	1.730		
Voltage level for switching to battery backup (Other than above)	$V_{DETBATT\_0}$	2.710	2.800	2.890	V	
	$V_{DETBATT\_1}$	2.450	2.530	2.610		
	$V_{DETBATT\_2}$	2.030	2.100	2.170		
	$V_{DETBATT\_3}$	1.885	1.950	2.015		
	$V_{DETBATT\_4}$	1.790	1.850	1.910		
	$V_{DETBATT\_5}$	1.690	1.750	1.810		
VCC drop detection stabilization wait time*2	t_{DETWT}	—	—	20	µs	—
Lower-limit VBATT voltage for power supply switching caused by VCC voltage drop	V_{BATTSW}	1.8	—	—	V	Figure 62.147
VCC-off period for starting power supply switching*1 (OFS1.PVDAS and PVDLPSEL are 0 in Deep Software Standby mode 1, 2)	$t_{VOFFBATT}$	25	—	—	µs	
VCC-off period for starting power supply switching*1 (Other than above)		25	—	—		

Table 62.114 Battery backup function characteristics (2 of 2)

Conditions: VCC = VCC_DCDC = VCC_USB = 1.62 to 3.63 V, VBATT = 1.62 to 3.63 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Backup domain power-down detection level	$V_{PDR\ (BATR)}$	1.43	1.47	1.52	V	Figure 62.148
Time delay in assertion of the reset signal for the backup domain*3	$t_{p\ (PDRL)}$	—	—	2000	µs	
Time delay in negation of the reset signal for the backup domain	$t_{p\ (PDRH)}$	—	—	3000		
VBATT monitor operation stabilization time (after VBATTMNSELR.VBTMNSEL is changed to 1)	t_{MONWT}	—	—	4.2	µs	—
VBATT voltage monitor level	$V_{MONBATT}$	—	VBATT / 6	—	V	—
VBATT current increase (when VBATTMNSELR.VBTTMNSEL is 1 compared to the case that VBATTMNSELR.VBTTMNSEL is 0)	$I_{VBATTSELB}$	—	1.35	2.00	µA	—
VCC current increase (when VBATTMNSELR.VBTTMNSEL is 1 compared to the case that VBATTMNSELR.VBTTMNSEL is 0)	$I_{VBATTSELC}$	—	15	25	µA	—

Note 1. The VCC-off period for starting power supply switching indicates the period in which VCC is below the minimum value of the voltage level for switching to battery backup ($V_{DETBATT}$).
In addition, this period indicates the time tVOFFP when VCC is below the minimum value of voltage detection levels V_{POR1}.

Note 2. Stable time when VBTBPCR2.VDETLVL is changed or VBTBPCR1. BPWSWSTP is changed from 1 to 0.

Note 3. When the VBATT_R recovers within this period, the backup domain reset signal may not be generated.

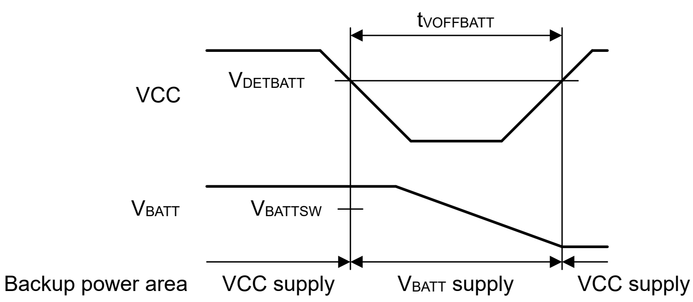


Figure 62.147 Battery backup function characteristics

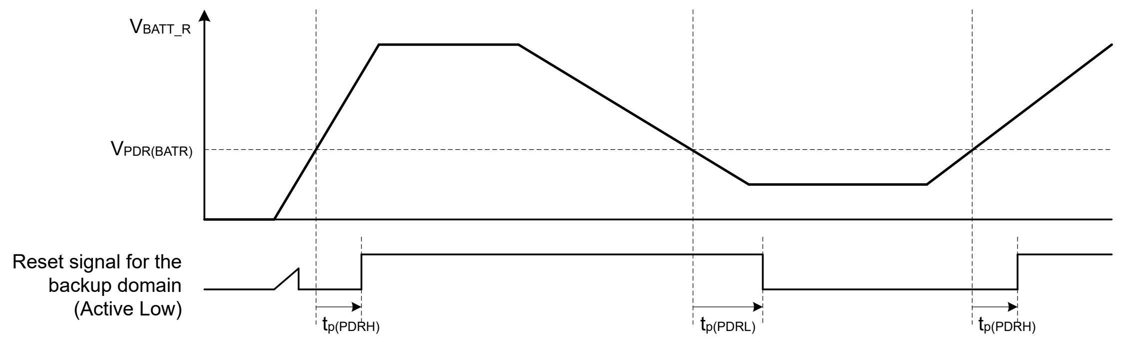


Figure 62.148 Backup domain reset characteristics

62.14 ACMPHS Characteristics

Table 62.115 ACMPHS

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
Reference voltage range			VREF	0	—	AVCC0	V	—
Input voltage range	ACMPHS0, 1	IVCMP1 to IVCMP3	VI	0	—	AVCC0	V	—
		IVCMP0		0	—	AVCC0		VCC ≥ AVCC0
				0	—	VCC		VCC < AVCC0
	ACMPHS2, 3			0	—	AVCC0		—
Output delay*1			Td	—	50	100	ns	VI = VREF ± 100mV
Internal reference voltage			Vref	0.77	0.8	0.84	V	—

Note 1. This value is the internal propagation delay.

62.15 MRAM Characteristics

62.15.1 Code MRAM Characteristics

Table 62.116 Code MRAM Characteristics (1 of 2)

Parameter		Symbol	MRICLK = 250 MHz			MRICLK = 200 MHz			MRICLK = 150 MHz			MRICLK = 133 MHz			Unit	Test conditions
			Min	Typ *4	Max	Min	Typ *4	Max	Min	Typ *4	Max	Min	Typ *4	Max		
Programming time of 32-byte*7 *8	Normal program mode (MRPSC.MHSPEN = 0)	t_{PMC}	—	6.7 *5 *6	83.3 *6	—	6.74 *5 *6	83.6 *6	—	6.92 *5 *6	85.6 *6	—	7.09 *5 *6	87.3 *6	μs	—
	High speed program mode (MRPSC.MHSPEN = 1)	t_{PMC}	—	4.7 *5 *6	81.3 *6	—	4.74 *5 *6	81.6 *6	—	4.92 *5 *6	83.6 *6	—	5.09 *5 *6	85.3 *6	μs	—

Table 62.116 Code MRAM Characteristics (2 of 2)

Parameter	Symbol	MRICLK = 250 MHz			MRICLK = 200 MHz			MRICLK = 150 MHz			MRICLK = 133 MHz			Unit	Test conditions
		Min	Typ *4	Max	Min	Typ *4	Max	Min	Typ *4	Max	Min	Typ *4	Max		
Reprogramming cycle*9	N_{PC}	100000 *1	—	—	100000 *1	—	—	100000 *1	—	—	100000 *1	—	—	Times	—
Data hold time*2	t_{DRP}	10*2 *3	—	—	10*2 *3	—	—	10*2 *3	—	—	10*2 *3	—	—	Years	Tj = +125°C

Note 1. This is the minimum number of times to guarantee all the characteristics after reprogramming. The guaranteed range is from 1 to the minimum value.
Note 2. This indicates the minimum value of the characteristic when reprogramming is performed within the specified range.
Note 3. This result is obtained from reliability testing.
Note 4. The reference value at VCC = 3.3 V and room temperature.
Note 5. Perform rewrite 50% of bits at Typ. condition.
Note 6. If MRPCLK < 125MHz, add the time of MRPCLK 1cycle to the Program time.
Note 7. To calculate the updating time at other frequencies, using approximate formula below listed.
If MRPCLK < 125MHz, add 1/FMRPCLK [us] to the formula. (F_{MRICLK}: frequency of MRICLK [MHz], F_{MRPCLK}: frequency of MRPCLK [MHz])
t_{PMC} (Typ) = 137.8/F_{MRICLK} + 6.452 [μs], t_{PMC} (Max) = 1879/F_{MRICLK} + 78.75 [μs] for normal program mode.
t_{PMC} (Typ) = 137.8/F_{MRICLK} + 4.452 [μs], t_{PMC} (Max) = 1879/F_{MRICLK} + 76.75 [μs] for high speed program mode.
Note 8. Read and program operations for the code MRAM cannot be executed simultaneously. This value is for independent program operation without arbitration between read and program.
Note 9. The reprogramming cycle is the number of programming per 32-byte code MRAM address space. Programming the same data to the same address also increases the reprogramming cycle count by one. For a reprogramming cycle of n times (n = 100,000), programming can be performed n times for every 32-byte code MRAM address space.

62.15.2 Option Setting Memory (configuration area) Characteristics

Table 62.117 Option Setting Memory (configuration area) characteristics

Parameter		Symbol	MRPCLK = 125 MHz			MRPCLK = 100 MHz			MRPCLK = 75 MHz			MRPCLK = 66 MHz			Unit	Test conditions
			Min	Typ *4	Max	Min	Typ *4	Max	Min	Typ *4	Max	Min	Typ *4	Max		
Command time for configuration set*5 *7	Normal program mode (MRPSC.MHSPEN = 0)	t_{PCFG}	—	0.356	8.19 *6	—	0.356	8.3 *6	—	0.372	8.83 *6	—	0.399	9.42 *6	ms	—
	High speed program mode (MRPSC.MHSPEN = 1)		—	0.0665	7.85 *6	—	0.0725	7.96 *6	—	0.0882	8.5 *6	—	0.0978	9.06 *6	ms	—
Update cycle*8		N_{CUPC}	100000 *1	—	—	100000 *1	—	—	100000 *1	—	—	100000 *1	—	—	Times	—
Data hold time*2		t_{DRP}	10*2 *3	—	—	10*2 *3	—	—	10*2 *3	—	—	10*2 *3	—	—	Years	Tj = +125°C

Note 1. This is the minimum number of times to guarantee all the characteristics after update. The guaranteed range is from 1 to the minimum value.
Note 2. This indicates the minimum value of the characteristic when update is performed within the specified range.
Note 3. This result is obtained from reliability testing.
Note 4. The reference value at VCC = 3.3 V and room temperature.
Note 5. To calculate the updating time at other frequencies, using approximate formula below listed. (F_{MRPCLK}: frequency of MRPCLK [MHz])

t_{PCFG} (Typ) = 6.146/F_{MRPCLK} + 0.3133 [msec], t_{PCFG} (Max) = 266.5/F_{MRPCLK} + 6.331 [msec] for normal program mode.
t_{PCFG} (Typ) = 5.184/F_{MRPCLK} + 0.02754 [msec], t_{PCFG} (Max) = 267.5/F_{MRPCLK} + 6.025 [msec] for high speed program mode.

Note 6. Values in this table are calculated based on the times of 5-times programming (16 bytes). Timeout values of Figure 59.16 is about 1.1 times of the maximum value (t_{PCFG} + t_{PMC} + Read time (MRPCLK 40cycle)).

Note 7. Read and program operations for the extra MRAM cannot be executed simultaneously. This value is for independent program operation without arbitration between read and program.

Note 8. The update cycle is the number of a configuration set command that can be issued configuration area. Programming the same data to the same address also increases the update cycle count by one.

62.15.3 Option Setting Memory (OTP area with ECC) Characteristics

Table 62.118 Option setting memory (OTP area with ECC) characteristics

Parameter		Symbol	MRPCLK = 125 MHz			MRPCLK = 100 MHz			MRPCLK = 75 MHz			MRPCLK = 66 MHz			Unit	Test conditions
			Min	Typ *1	Max	Min	Typ *1	Max	Min	Typ *1	Max	Min	Typ *1	Max		
Programming time of 16-byte*4 *5	Normal speed write mode (MWMCR.MWM[1:0] = 00)	t_{OTPE}	—	8.05 *2	113 *3	—	8.05 *2	113 *3	—	8.05 *2	113 *3	—	8.05 *2	113 *3	ms	—
	High speed write mode 0 (MWMCR.MWM[1:0] = 01)		—	4.03 *2	56.7 *3	—	4.03 *2	56.7 *3	—	4.03 *2	56.7 *3	—	4.03 *2	56.7 *3	ms	—
	High speed write mode 1 (MWMCR.MWM[1:0] = 10)		—	0.229*2	3.14 *3	—	0.229*2	3.14 *3	—	0.229*2	3.14 *3	—	0.229*2	3.14 *3	ms	—

Note 1. The reference value at VCC = 3.3 V and room temperature.

Note 2. Perform rewrite 50% of bits at Typ. condition.

Note 3. Timeout values of Figure 59.13 is about 1.1 times of the maximum value. (t_{OTPE} + t_{PMC} + Read time (MRPCLK 40cycle)).

Note 4. To calculate the updating time at other frequencies, using approximate formula below listed. (F_{MRPCLK}: frequency of MRPCLK [MHz])
t_{OTPE} (Typ) = 0.5065/F_{MRPCLK} + 8.123 [ms], t_{OTPE} (Max) = 4.433/F_{MRPCLK} + 114.1 [ms] for normal speed mode.
t_{OTPE} (Typ) = 0.3389/F_{MRPCLK} + 4.067 [ms], t_{OTPE} (Max) = 2.428/F_{MRPCLK} + 57.08 [ms] for high speed mode 0.
t_{OTPE} (Typ) = 0.1458/F_{MRPCLK} + 0.2312 [ms], t_{OTPE} (Max) = 0.3904/F_{MRPCLK} + 3.166 [ms] for high speed mode 1.

Note 5. Read and program operations for the extra MRAM cannot be executed simultaneously. This value is for independent program operation without arbitration between read and program.

62.15.4 Option Setting Memory (OTP area without ECC) Characteristics

Table 62.119 Option Setting Memory (OTP area without ECC) Characteristics

Parameter		Symbol	MRPCLK = 125 MHz			MRPCLK = 100 MHz			MRPCLK = 75 MHz			MRPCLK = 66 MHz			Unit	Test conditions
			Min	Typ *1	Max	Min	Typ *1	Max	Min	Typ *1	Max	Min	Typ *1	Max		
Programming time of 16-byte*4 *5	Normal speed write mode (MWMCR.MWM[1:0] = 00)	t_{OTPENE}	—	14.1 *2	200 *3	—	14.1 *2	200 *3	—	14.1 *2	200 *3	—	14.1 *2	200 *3	ms	—
	High speed write mode 0 (MWMCR.MWM[1:0] = 01)		—	7.07 *2	100 *3	—	7.07 *2	100 *3	—	7.07 *2	100 *3	—	7.07 *2	100 *3	ms	—
	High speed write mode 1 (MWMCR.MWM[1:0] = 10)		—	0.458 *2	6.28 *3	—	0.458 *2	6.28 *3	—	0.458 *2	6.28 *3	—	0.458 *2	6.28 *3	ms	—

Note 1. The reference value at VCC = 3.3 V and room temperature.
Note 2. Perform rewrite 50% of bits at Typ. condition.
Note 3. Timeout values of Figure 59.13 is about 1.1 times of the maximum value. (t_{OTPENE} + t_{PMC} + Read time (MRPCLK 40cycle)).
Note 4. To calculate the updating time at other frequencies, using approximate formula below listed. (F_{MRPCLK}: frequency of MRPCLK [MHz])
t_{OTPENE} (Typ) = 0.8486/F_{MRPCLK} + 14.25 [ms], t_{OTPENE} (Max) = 7.711/F_{MRPCLK} + 201.5 [ms] for normal speed mode.
t_{OTPENE} (Typ) = 0.5479/F_{MRPCLK} + 7.133 [ms], t_{OTPENE} (Max) = 4.148/F_{MRPCLK} + 100.8 [ms] for high speed mode 0.
t_{OTPENE} (Typ) = 0.2571/F_{MRPCLK} + 0.4627 [ms], t_{OTPENE} (Max) = 0.7401/F_{MRPCLK} + 6.333 [ms] for high speed mode 1.
Note 5. Read and program operations for the extra MRAM cannot be executed simultaneously. This value is for independent program operation without arbitration between read and program.

62.15.5 MACI Command Characteristics

Table 62.120 MACI command Characteristics

Parameter	Symbol	MRPCLK = 125 MHz			MRPCLK = 100 MHz			MRPCLK = 75 MHz			MRPCLK = 66 MHz			Unit	Test conditions
		Min	Typ *1	Max	Min	Typ *1	Max	Min	Typ *1	Max	Min	Typ *1	Max		
Command time for forced stop command*2	t_{FS}	—	—	3.35 *3	—	—	3.38 *3	—	—	3.47 *3	—	—	3.7 *3	μs	—
Command time for increment counter*2	t_{INCC}	—	0.252	1.61 *4	—	0.252	1.61 *4	—	0.252	1.61 *4	—	0.252	1.61 *4	ms	—
Command time for read counter*2	t_{RDC}	—	—	0.156*5	—	—	0.182*5	—	—	0.243*5	—	—	0.276*5	μs	—

Note 1. The reference value at VCC = 3.3 V and room temperature.
Note 2. To calculate the updating time at other frequencies, using approximate formula below listed. (F_{MRPCLK}: frequency of MRPCLK [MHz])
t_{FS} (Max) = 38.62/F_{MRPCLK} + 3.155 [us]
t_{INCC} (Typ) = 0.3348/F_{MRPCLK} + 0.2533 [msec], t_{INCC} (Max) = 0.8698/F_{MRPCLK} + 1.62 [msec]

t_{RDC} (Max) = 19.13/F_{MRPCLK} + 0.004099 [usec]

Note 3. Timeout values of Figure 59.15 is about 1.1 times of the maximum value. (t_{FS} + Read time (MRPCLK 40cycle)).

Note 4. Values in this table are calculated based on the times of 2-time programming (32 bytes). Timeout values of Figure 59.17 is about 1.1 times of the maximum value (t_{INCC} + t_{PMC} + Read time (MRPCLK 40cycle)).

Note 5. Counter read time is 100 cycles at MRPCLK. Timeout values of Figure 59.18 is about 1.1 times of the maximum value (t_{RDC} + Read time (MRPCLK 40cycle)).

62.15.6 Zeroizing of W-HUK

Table 62.121 Zeroizing of W-HUK Characteristics

Parameter	Symbol	MRPCLK = 125 MHz			MRPCLK = 100 MHz			MRPCLK = 75 MHz			MRPCLK = 66 MHz			Unit	Test conditions
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
Acton time of Zeroing of W-HUK*1	t_{ZWH}	—	—	793	—	—	793	—	—	793	—	—	793	ms	—

Note 1. To calculate the updating time at other frequencies, using approximate formula below listed. (FMRPCLK: frequency of MRPCLK [MHz])
t_{ZWH} (Max) = 34.52/FMRPCLK + 799 [μs]

62.15.7 MRAM Magnetic Field Immunity Characteristics

Table 62.122 MRAM Magnetic field Immunity Characteristics

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Operation	Write state	G_{max_wr}	—	—	200	Gauss	—
	Read state	G_{max_rd}	—	—	200	Gauss	—
	No access state	G_{max_noac}	—	—	500	Gauss	—
Storage		G_{max_stg}	—	—	500	Gauss	The applied temperature is T_{stg}.

62.16 Boundary Scan

Table 62.123 Boundary scan characteristics (1 of 2)

Parameter	VCC	Symbol	Min	Typ	Max	Unit	Test conditions
TCK clock cycle time	1.62 V or above	t_{TCKcyc}	100	—	—	ns	Figure 62.149
TCK clock high pulse width	1.62 V or above	t_{TCKH}	0.45	—	—	t_{TCKcyc}	
TCK clock low pulse width	1.62 V or above	t_{TCKL}	0.45	—	—	t_{TCKcyc}	
TCK clock rise time	1.62 V or above	t_{TCKr}	—	—	0.05*2	t_{TCKcyc}	
TCK clock fall time	1.62 V or above	t_{TCKf}	—	—	0.05*2	t_{TCKcyc}	

Table 62.123 Boundary scan characteristics (2 of 2)

Parameter	VCC	Symbol	Min	Typ	Max	Unit	Test conditions
TMS setup time	1.62 V or above	t_{TMSS}	20	—	—	ns	Figure 62.150
TMS hold time	1.62 V or above	t_{TMSH}	20	—	—	ns	
TDI setup time	1.62 V or above	t_{TDIS}	20	—	—	ns	
TDI hold time	1.62 V or above	t_{TDIH}	20	—	—	ns	
TDO data delay	1.62 V or above	t_{TDOD}	—	—	40	ns	
Capture register setup time	1.62 V or above	t_{CAPTS}	20	—	—	ns	Figure 62.151
Capture register hold time	1.62 V or above	t_{CAPTH}	20	—	—	ns	
Update register delay time	1.62 V or above	$t_{UPDATED}$	—	—	40	ns	
Boundary scan circuit startup time*1	1.62 V or above	T_{BSSTUP}	t_{RESWP}	—	—	—	Figure 62.152

Note 1. Boundary scan does not function until the power-on reset becomes negative.
Note 2. 1 µs at the longest

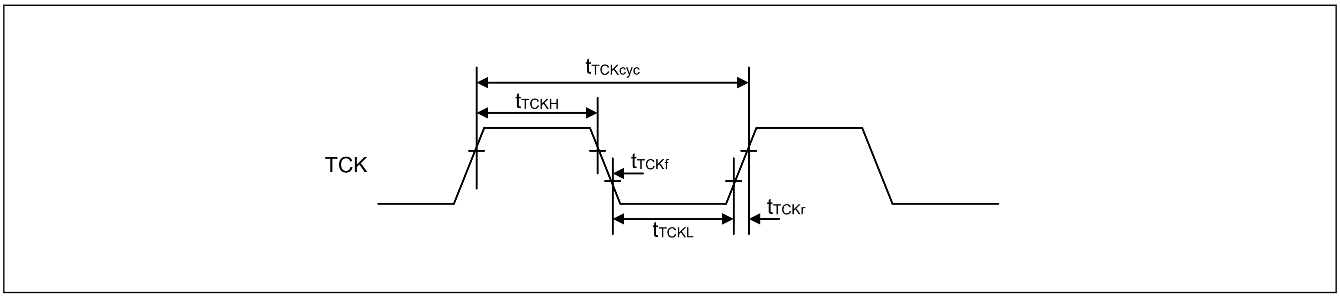


Figure 62.149 Boundary scan TCK timing

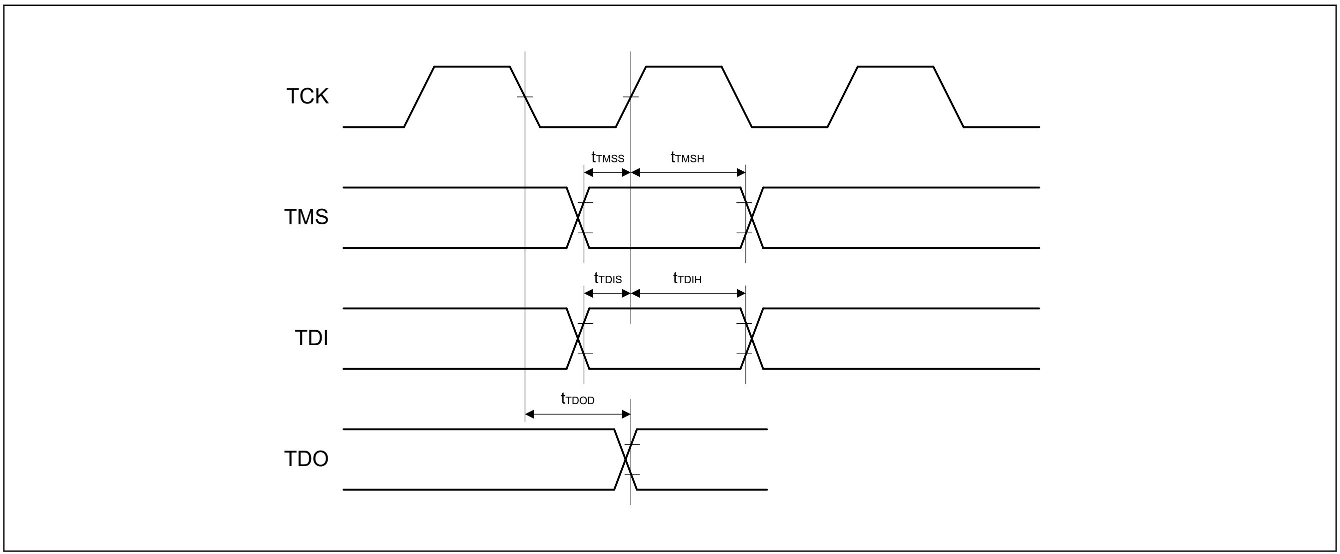


Figure 62.150 Boundary scan input/output timing (1)

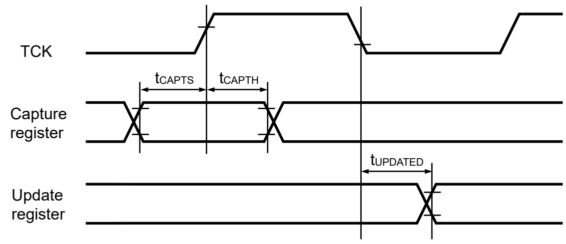


Figure 62.151 Boundary scan input/output timing (2)

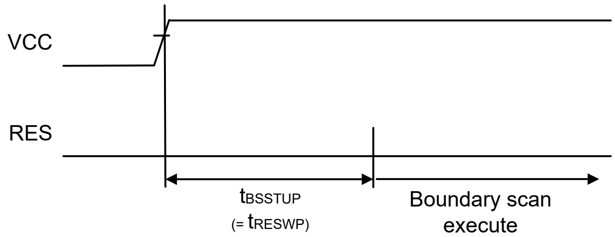


Figure 62.152 Boundary scan circuit startup timing

62.17 Joint European Test Action Group (JTAG)

Table 62.124 JTAG (1 of 2)

Parameter	VCC	Symbol	Min	Typ	Max	Unit	Test conditions
TCK clock cycle time	2.7 V or above	t_{TCKcyc}	40.0	—	—	ns	Figure 62.153
	1.62 V or above		40.0	—	—	ns	
TCK clock high pulse width	2.7 V or above	t_{TCKH}	0.375	—	—	t_{TCKcyc}	
	1.62 V or above		0.375	—	—	t_{TCKcyc}	
TCK clock low pulse width	2.7 V or above	t_{TCKL}	0.375	—	—	t_{TCKcyc}	
	1.62 V or above		0.375	—	—	t_{TCKcyc}	
TCK clock rise time	2.7 V or above	t_{TCKr}	—	—	0.125*1	t_{TCKcyc}	
	1.62 V or above		—	—	0.125*1	t_{TCKcyc}	
TCK clock fall time	2.7 V or above	t_{TCKf}	—	—	0.125*1	t_{TCKcyc}	
	1.62 V or above		—	—	0.125*1	t_{TCKcyc}	

Table 62.124 JTAG (2 of 2)

Parameter	VCC	Symbol	Min	Typ	Max	Unit	Test conditions
TMS setup time	2.7 V or above	t_{TMSS}	8.0	—	—	ns	Figure 62.154
	1.62 V or above		8.0	—	—	ns	
TMS hold time	2.7 V or above	t_{TMSH}	8.0	—	—	ns	
	1.62 V or above		8.0	—	—	ns	
TDI setup time	2.7 V or above	t_{TDIS}	8.0	—	—	ns	
	1.62 V or above		8.0	—	—	ns	
TDI hold time	2.7 V or above	t_{TDIH}	8.0	—	—	ns	
	1.62 V or above		8.0	—	—	ns	
TDO data delay time	2.7 V or above	t_{TDOD}	—	—	20.0	ns	
	1.62 V or above		—	—	28.0	ns	

Note 1. 1 µs at the longest

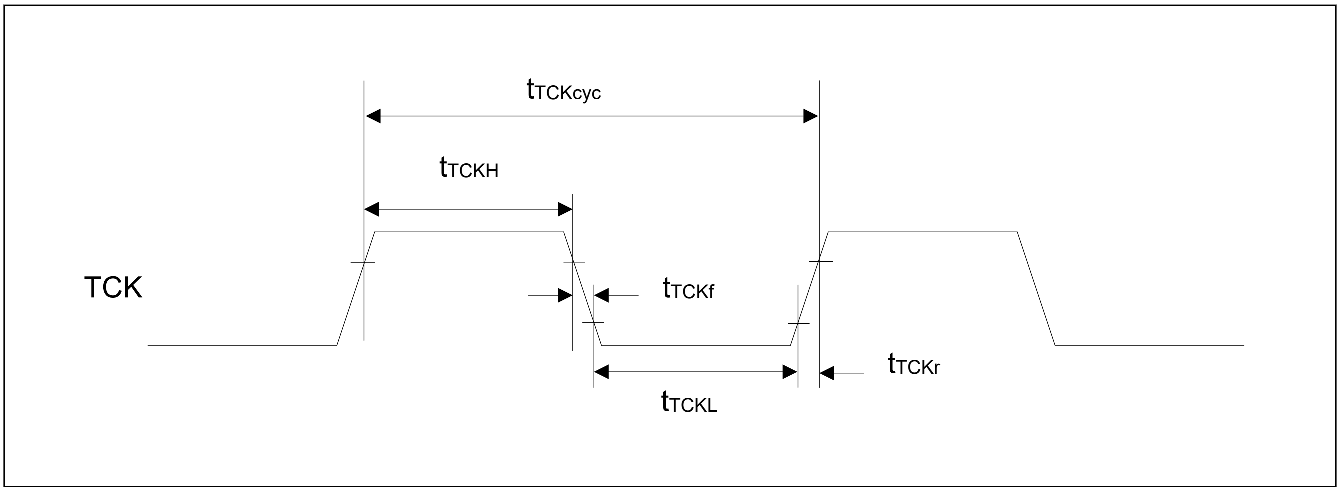


Figure 62.153 JTAG TCK timing

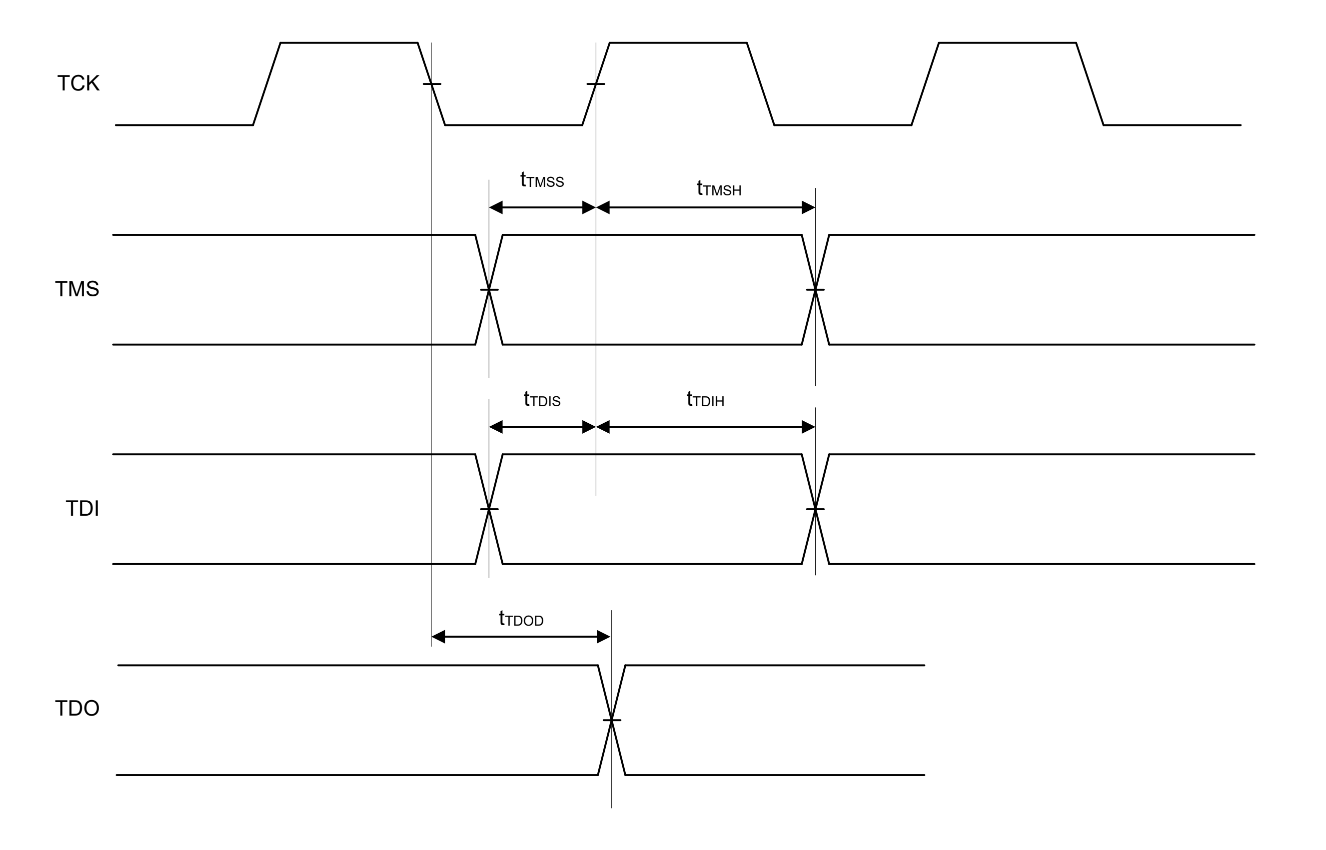


Figure 62.154 JTAG input/output timing

62.18 Serial Wire Debug (SWD)

Table 62.125 SWD (1 of 2)

Parameter	VCC	Symbol	Min	Typ	Max	Unit	Test conditions
SWCLK clock cycle time	2.7 V or above	$t_{SWCKcyc}$	40.0	—	—	ns	Figure 62.155
	1.62 V or above		40.0	—	—	ns	
SWCLK clock high pulse width	2.7 V or above	t_{SWCKH}	0.375	—	—	$t_{SWCKcyc}$	
	1.62 V or above		0.375	—	—	$t_{SWCKcyc}$	
SWCLK clock low pulse width	2.7 V or above	t_{SWCKL}	0.375	—	—	$t_{SWCKcyc}$	
	1.62 V or above		0.375	—	—	$t_{SWCKcyc}$	
SWCLK clock rise time	2.7 V or above	t_{SWCKr}	—	—	0.125*1	$t_{SWCKcyc}$	
	1.62 V or above		—	—	0.125*1	$t_{SWCKcyc}$	
SWCLK clock fall time	2.7 V or above	t_{SWCKf}	—	—	0.125*1	$t_{SWCKcyc}$	
	1.62 V or above		—	—	0.125*1	$t_{SWCKcyc}$	

Table 62.125 SWD (2 of 2)

Parameter	VCC	Symbol	Min	Typ	Max	Unit	Test conditions
SWDIO setup time	2.7 V or above	t_{SWDS}	8.0	—	—	ns	Figure 62.156
	1.62 V or above		8.0	—	—	ns	
SWDIO hold time	2.7 V or above	t_{SWDH}	8.0	—	—	ns	
	1.62 V or above		8.0	—	—	ns	
SWDIO data delay time	2.7 V or above	t_{SWDD}	2.0	—	28.0	ns	
	1.62 V or above		2.0	—	32.0	ns	

Note 1. 1 µs at the longest

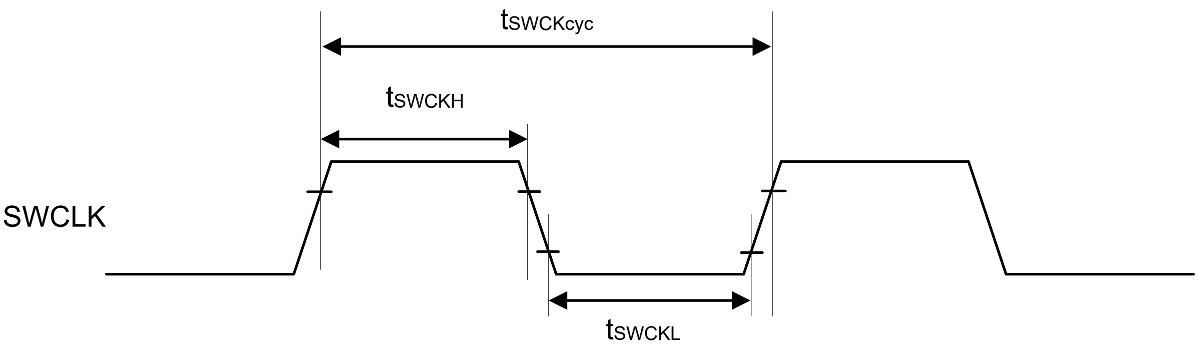


Figure 62.155 SWD SWCLK timing

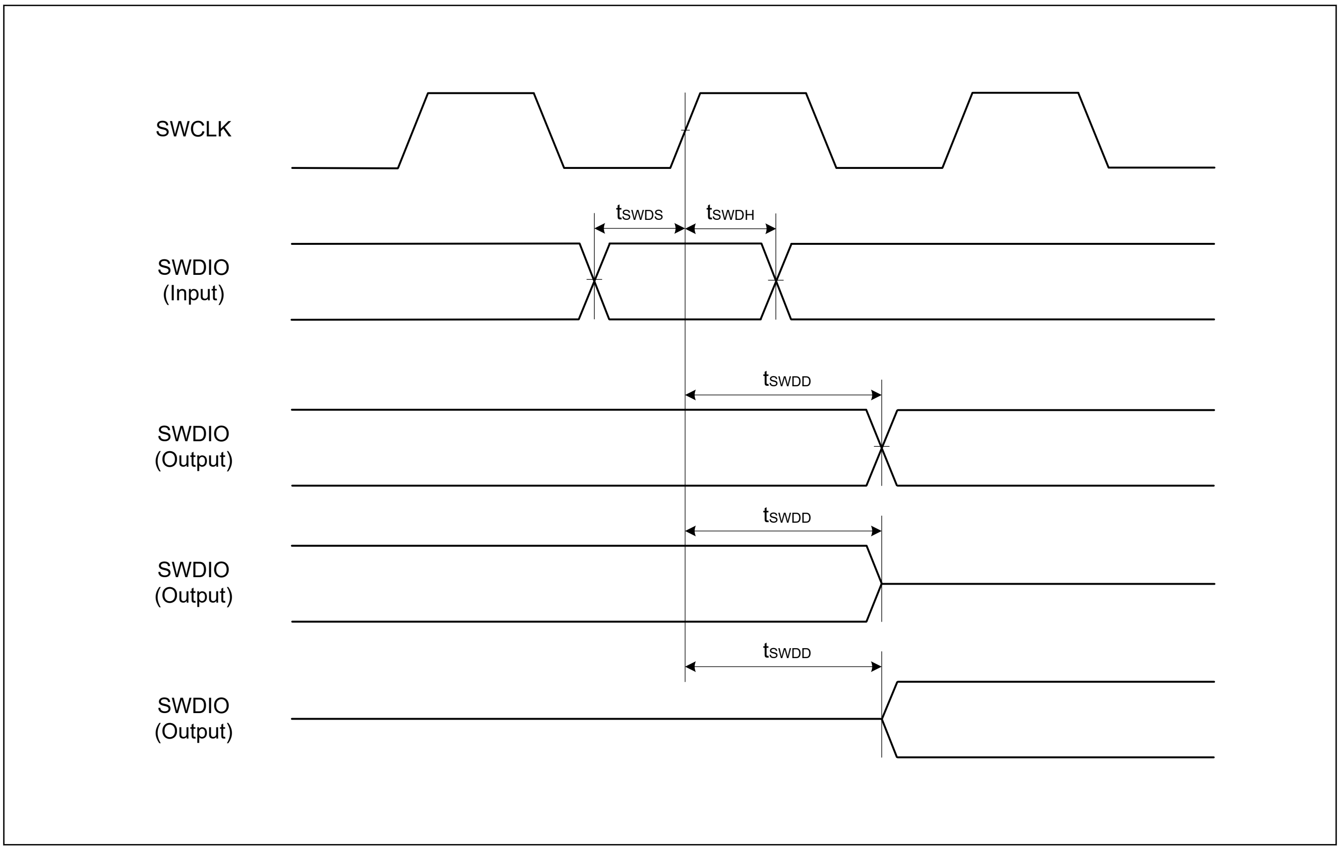


Figure 62.156 SWD input/output timing

62.19 Embedded Trace Macro Interface (ETM)

Table 62.126 ETM (1 of 2)

Conditions: High-speed high drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter	VCC	Symbol	Min	Typ	Max	Unit	Test conditions
TCLK clock cycle time	2.7 V or above	$t_{TCLKcyc}$	16	—	—	ns	Figure 62.157
	1.62 V or above		16	—	—	ns	
TCLK clock high pulse width	2.7 V or above	t_{TCLKH}	7	—	—	ns	
	1.62 V or above		6	—	—	ns	
TCLK clock low pulse width	2.7 V or above	t_{TCLKL}	7	—	—	ns	
	1.62 V or above		6	—	—	ns	
TCLK clock rise time	2.7 V or above	t_{TCLKr}	—	—	1.0	ns	
	1.62 V or above		—	—	2.0	ns	
TCLK clock fall time	2.7 V or above	t_{TCLKf}	—	—	1.0	ns	
	1.62 V or above		—	—	2.0	ns	

Table 62.126 ETM (2 of 2)

Conditions: High-speed high drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter	VCC	Symbol	Min	Typ	Max	Unit	Test conditions
TDATA[3:0] output valid time	2.7 V or above	t_{TRDV}	—	—	$t_{TCLKcyc}$/4 + 1.5	ns	Figure 62.158
	1.62 V or above		—	—	$t_{TCLKcyc}$/4 + 1.5	ns	
TDATA[3:0] output hold time	2.7 V or above	t_{TRDH}	1.5	—	—	ns	
	1.62 V or above		1.5	—	—	ns	

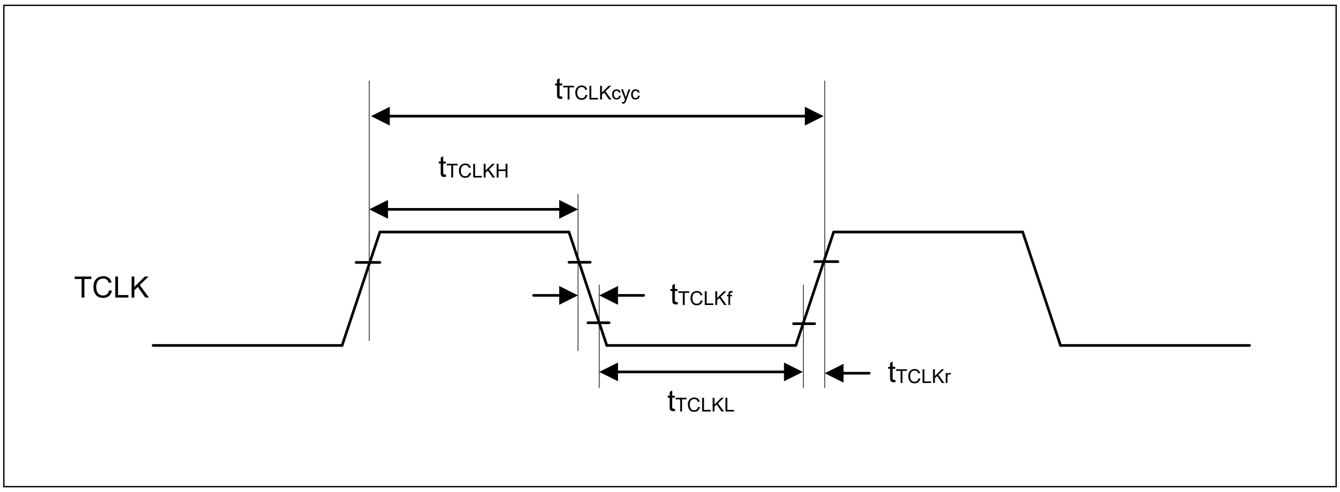


Figure 62.157 ETM TCLK timing

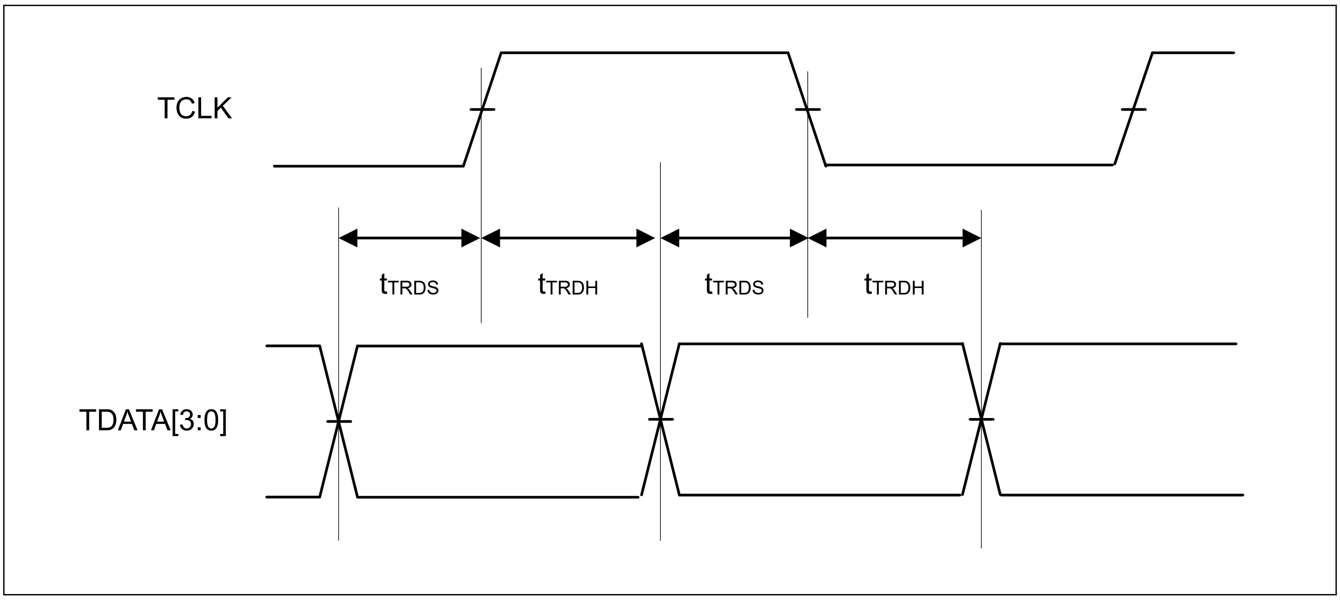


Figure 62.158 ETM output timing

Appendix 1. Port States in Each Processing Mode

Function	Pin function	Reset	Software Standby mode (SSTBY)		Deep Software Standby mode 1, 2, 3 (DSTBY1, 2, 3)		After Deep Software Standby mode is canceled (return to startup mode)	
			OPE = 0	OPE = 1	DSTBY1	DSTBY2/ DSTBY3	IOKEEP = 0	IOKEEP = 1[*1]
Mode	MD	Pull-up	Keep-I		Keep		Pull-up	Keep
JTAG/SWD	TCK/TMS/TDI/SWCLK	Pull-up	TCK/TDI/TMS/SWCLK input		TCK/TDI/TMS/SWCLK input		TCK/TDI/TMS/SWCLK input	
	TDO	Output	TDO output		TDO output		TDO output	
	SWDIO	Pull-up	SWDIO inout		SWDIO inout		SWDIO inout	
Trace	TCLK/TDATAx/SWO	TCLK/ TDATAx/SWO output	TCLK/TDATAx/SWO output		TCLK/TDATAx/SWO output		TCLK/TDATAx/SWO output	
IRQ	IRQx	Hi-Z	Hi-Z[*2]		Keep		Hi-Z	Keep
	IRQx-DS (x: Other than 5)	Hi-Z	Hi-Z[*2]		Keep[*3]		Hi-Z	Keep
	IRQ5-DS	Hi-Z	Hi-Z[*2]		Keep[*3]		Hi-Z	
AGT	AGTIOn	Hi-Z	AGTIOn inout		Keep		Hi-Z	Keep
	AGTOn/AGTOAn/ AGTOBn	Hi-Z	AGTOn/AGTOAn/AGTOBn output		Keep		Hi-Z	Keep
ULPT	ULPTEEn/ULPTEVIn	Hi-Z	ULPTEEn/ULPTEVIn input		Keep		Hi-Z	Keep
	ULPTEEn-DS/ ULPTEVIn-DS	Hi-Z	ULPTEEn-DS/ULPTEVIn-DS input		ULPTEEn-DS/ ULPTEVIn-DS input	Hi-Z	Hi-Z	Keep
	ULPTOn/ ULPTOAn/ ULPTOBn	Hi-Z	ULPTOn/ULPTOAn/ULPTOBn output		Keep		Hi-Z	Keep
	ULPTOn-DS/ ULPTOAn-DS/ ULPTOBn-DS	Hi-Z	ULPTOn/ULPTOAn-DS/ ULPTOBn-DS output		ULPTOn/ULPTOAn-DS/ ULPTOBn-DS output	Keep	Hi-Z	From DSTBY1: ULPTOn/ ULPTOAn-DS/ ULPTOBn-DS output From DSTBY1, 2: Keep
IIC	SCLn/SDAn	Hi-Z	Keep-O[*2]		Keep		Hi-Z	Keep
I3C	I3C_SCL0/I3C_SDA0	Hi-Z	Keep-O[*2]		Hi-Z		Hi-Z	
USBFS	USB_OVRCURx	Hi-Z	Hi-Z[*2]		Keep		Hi-Z	Keep
	USB_OVRCURx-DS/ USB_VBUS	Hi-Z	Hi-Z[*2]		Keep[*3]	Keep	Hi-Z	Keep
	USB_DP/USB_DM	Hi-Z	Keep-O[*4]		Keep[*3]	Keep	Hi-Z	Keep
USBHS	USBHS_OVRCURx	Hi-Z	Hi-Z[*2]		Keep		Hi-Z	Keep
	USBHS_OVRCURx-DS / USBHS_VBUS	Hi-Z	Hi-Z[*2]		Keep[*3]	Keep	Hi-Z	Keep
	USBHS_DP/ USBHS_DM	Hi-Z	Keep-O[*4]		Keep[*5]	Keep	Hi-Z	Keep
RTC	RTCICx	Hi-Z	Hi-Z[*2]		Keep[*3]		Hi-Z	Keep
	RTCOUT	Hi-Z	RTCOUT output		Keep		Hi-Z	Keep
ACMPHS	VCOUT	Hi-Z	VCOUT output		Keep		Hi-Z	Keep
CLKOUT	CLKOUT	Hi-Z	CLKOUT output		Keep		Hi-Z	Keep
DAC	DAn	Hi-Z	D/A output retained		Keep		Hi-Z	Keep
External bus (CS, SDRAM area)	EBCLK/SDCLK	Hi-Z	High-level output		Keep		Hi-Z	Keep
	Dxx/DQxx	Hi-Z	Hi-Z		Hi-Z		Hi-Z	
	Axx/DQMx	Hi-Z	Hi-Z	Keep-O	Keep		Hi-Z	Keep
	BCx/CSx/RD/WRx/WE	Hi-Z	Hi-Z	High-level output	Keep		Hi-Z	Keep
	ALE	Hi-Z	Hi-Z	Low-level output	Keep		Hi-Z	Keep
	CKE/SDCS/RAS/CAS	Hi-Z	Hi-Z	SDSELF.SFEN = 0: High-level output SDSELF.SFEN = 1: Low-level output	Keep		Hi-Z	Keep

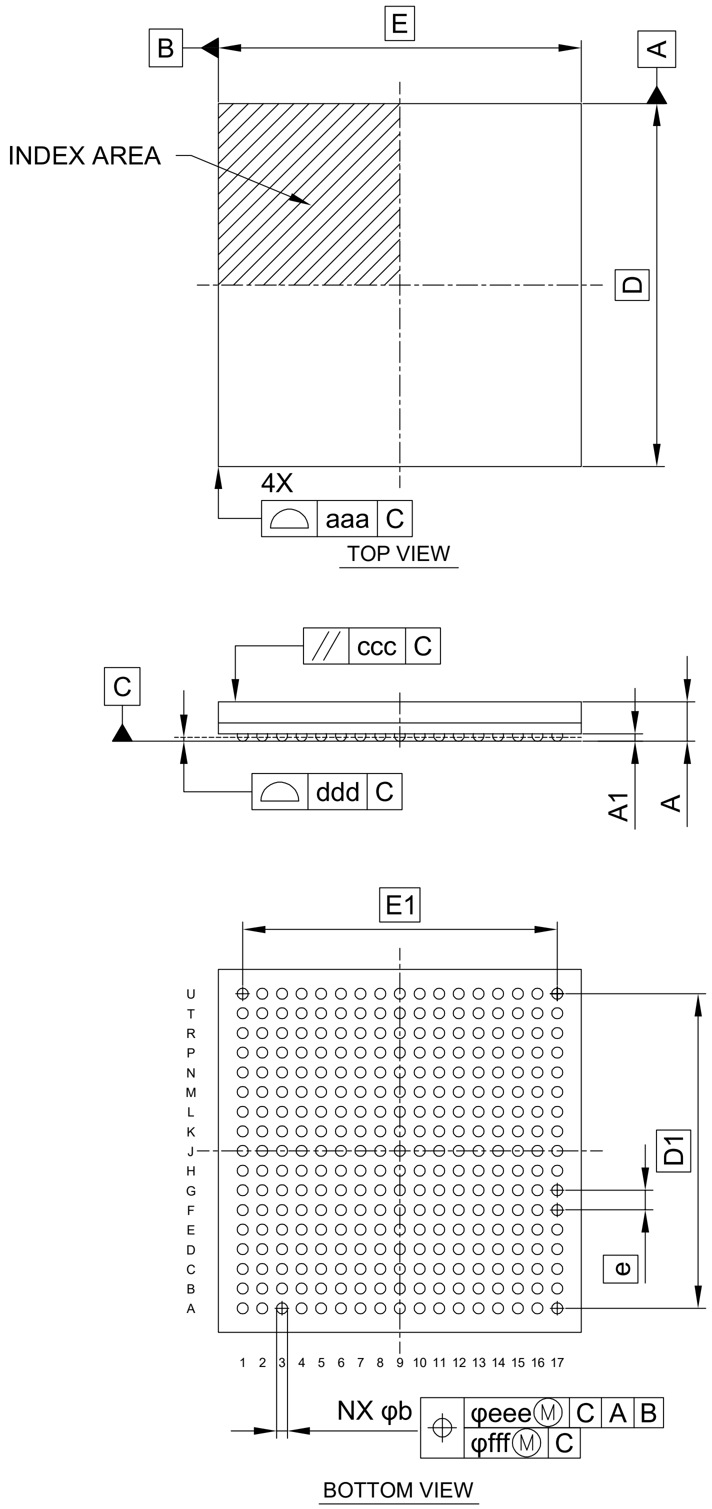
Function	Pin function	Reset	Software Standby mode (SSTBY)		Deep Software Standby mode 1, 2, 3 (DSTBY1, 2, 3)		After Deep Software Standby mode is canceled (return to startup mode)	
			OPE = 0	OPE = 1	DSTBY1	DSTBY2/ DSTBY3	IOKEEP = 0	IOKEEP = 1*1
P400/P401	Other than function IRQ5-DS	Hi-Z	Keep-O*2		Hi-Z		Hi-Z	
PDMIF	PDMCLKn	Hi-Z	PDMCLKn output		Keep		Hi-Z	Keep
	PDMDATAn	Hi-Z	Keep-O*6		Keep		Hi-Z	Keep
Others	—	Hi-Z	Keep-O		Keep		Hi-Z	Keep

Note: Hi-Z: High-impedance
Keep-O: Output pins retain their previous values. Input pins go to high-impedance.
Keep-I: Pin states are retained the same as during periods in Normal mode.
Keep: Pin states are retained the same as during periods in Software Standby mode.
Note 1. Retains the I/O port state until the DPSBYCR.IOKEEP bit is cleared to 0.
Note 2. Input is enabled if the pin is specified as the Software Standby canceling source while it is used as an external interrupt pin.
Note 3. Input is enabled if the pin is specified as the Deep Software Standby canceling source.
Note 4. Input is enabled while the pin is used as an input pin.
Note 5. For host operation, set the USBHS.SYSCFG.DRPD bit to 1 to enable the USBHS_DP and USBHS_DM pull-down resistors.
For device operation, set the USBHS.SYSCFG.DPRPU bit to 1 to enable the DP pull-up resistor.
Note 6. Input is enabled if the PDMIF sound detection interrupt is specified as the Software Standby canceling source.

Appendix 2. Package Dimensions

Information on the latest version of the package dimensions or mountings is displayed in “Packages” on the Renesas Electronics Corporation website.

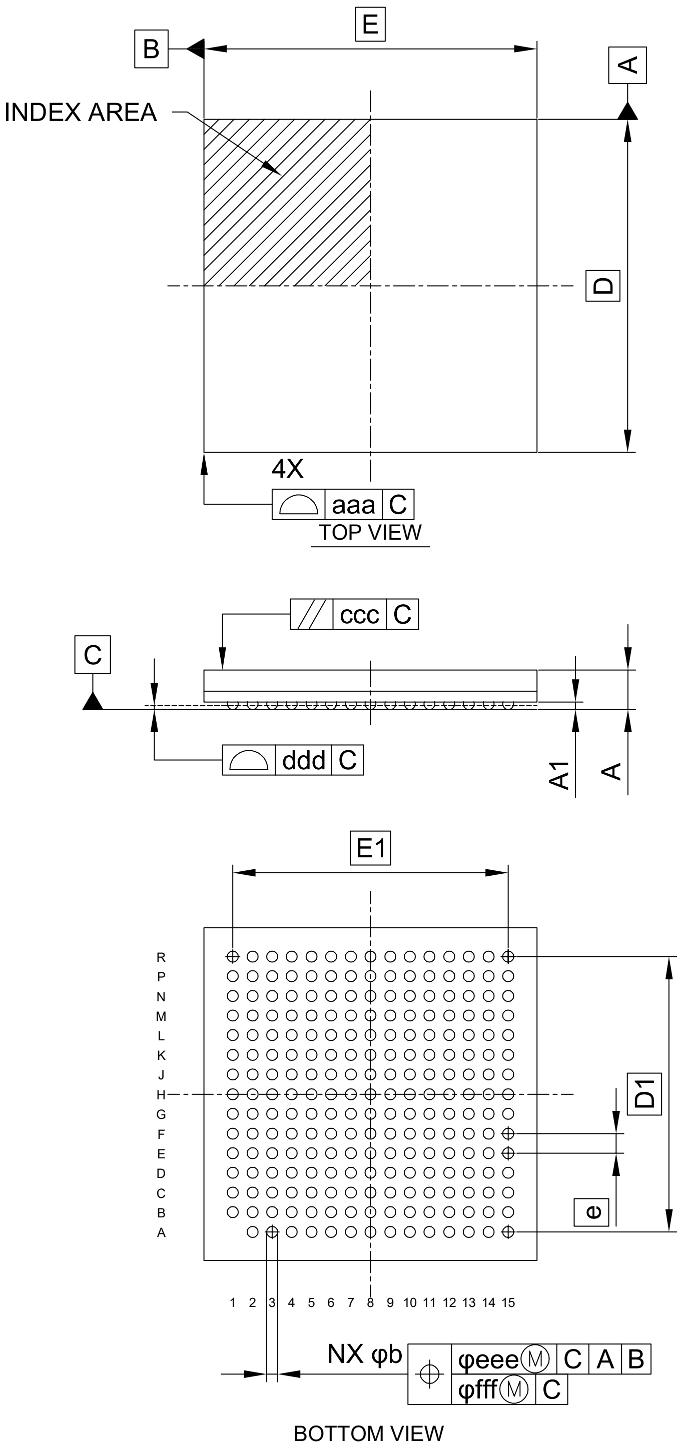
JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-LFBGA289-12x12-0.65	PLBG0289JA-A	0.38



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	—	12.00	—
E	—	12.00	—
D1	—	10.40	—
E1	—	10.40	—
A	—	—	1.38
A1	0.20	—	—
b	0.31	0.36	0.41
e	—	0.65	—
aaa	—	—	0.15
ccc	—	—	0.20
ddd	—	—	0.10
eee	—	—	0.15
fff	—	—	0.08
N	—	289	—

Figure A2.1 BGA 289-pin

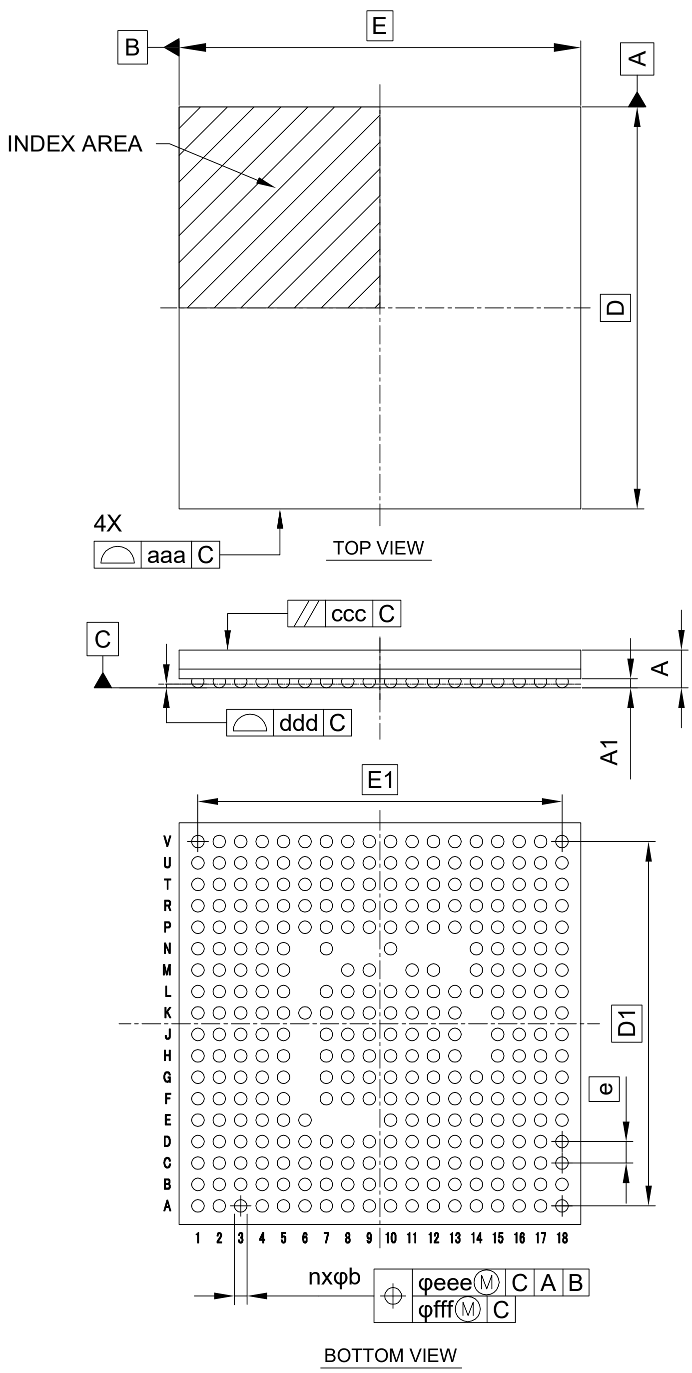
JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-LFBGA224-11x11-0.65	PLBG0224JA-A	0.32



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	—	11.00	—
E	—	11.00	—
D1	—	9.10	—
E1	—	9.10	—
A	—	—	1.38
A1	0.20	—	—
b	0.31	0.36	0.41
e	—	0.65	—
aaa	—	—	0.15
ccc	—	—	0.20
ddd	—	—	0.10
eee	—	—	0.15
fff	—	—	0.08
N	—	224	—

Figure A2.2 BGA 224-pin

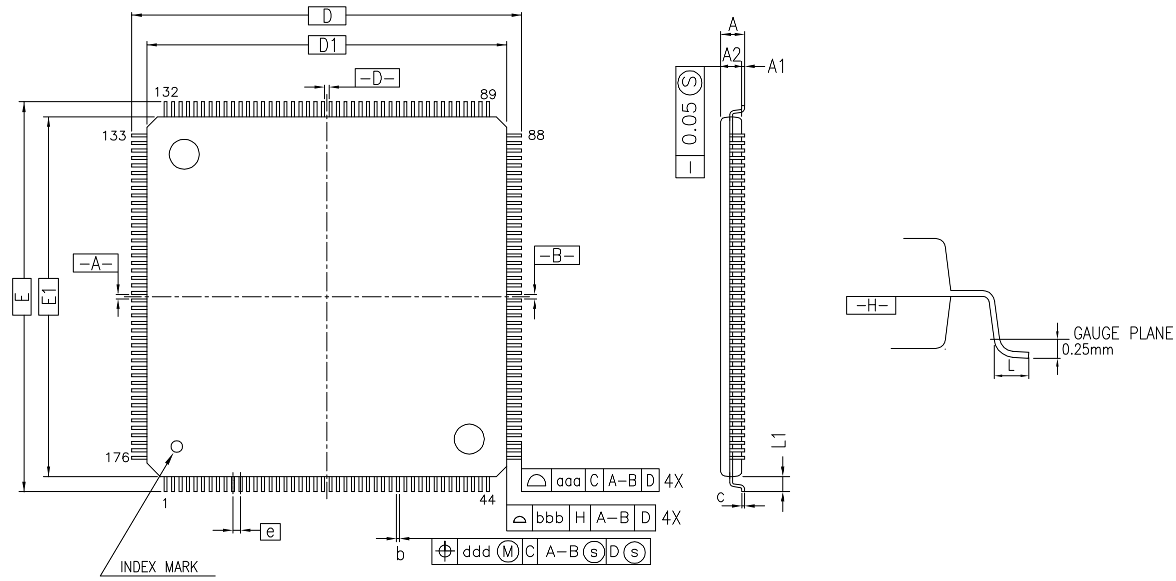
JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-LFBGA303-15x15-0.80	PLBG0303GA-A	0.63

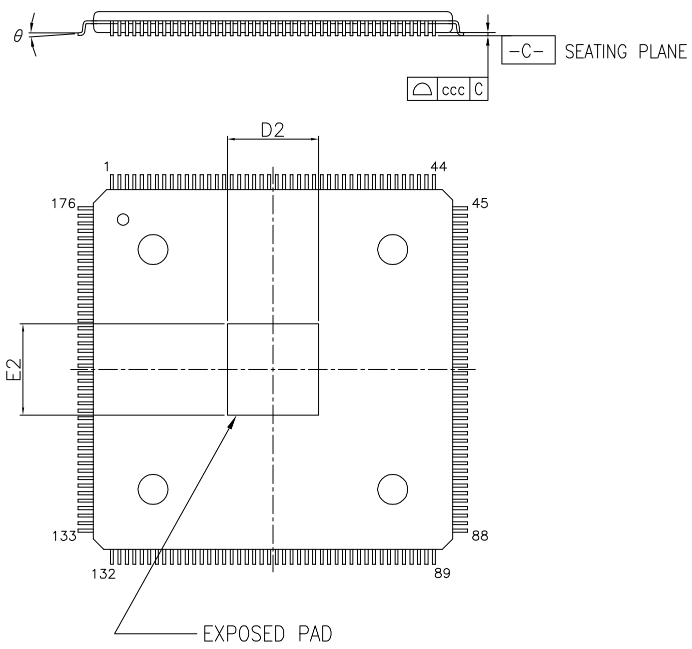


Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	—	15.00	—
E	—	15.00	—
D1	—	13.60	—
E1	—	13.60	—
A	—	—	1.50
A1	0.29	—	—
b	0.42	0.47	0.52
e	—	0.80	—
aaa	—	—	0.15
ccc	—	—	0.20
ddd	—	—	0.20
eee	—	—	0.15
fff	—	—	0.08
N	—	303	—

Figure A2.3 BGA 303-pin

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HLQFP176-24x24-0.50	PLQP0176KK-A	1.90





Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
D	26.00 BSC.		
D1	24.00 BSC.		
E	26.00 BSC.		
E1	24.00 BSC.		
N	–	176	–
e	0.50 BSC.		
b	0.17	0.22	0.27
c	0.09	–	0.20
θ	0°	3.5°	7°
L	0.45	0.60	0.75
L1	1.00 REF.		
D2	5.10	–	–
E2	5.10	–	–
aaa	–	–	0.20
bbb	–	–	0.20
ccc	–	–	0.08
ddd	–	–	0.08

Figure A2.4 HLQFP 176-pin

Appendix 3. I/O Registers

This appendix describes I/O register address and access cycles by function.

3.1 Peripheral Base Addresses

This section provides the base addresses for peripherals described in this manual. Table A3.1 shows the name, description, and the base address of each peripheral.

Table A3.1 Peripheral base address (1 of 4)

Description	Name of secure registers	Base address of secure registers in secure alias region	Name of non-secure registers	Base address of non-secure registers in non-secure alias region
Renesas Memory Protection Unit	RMPU	0x4000_0000	RMPU_NS	0x5000_0000
SRAM Control	SRAM	0x4000_2000	SRAM_NS	0x5000_2000
BUS Control	BUS	0x4000_3000	BUS_NS	0x5000_3000
Common Interrupt Controller	ICU_COMMON	0x4000_6000	ICU_COMMON_NS	0x5000_6000
CPU System Security Control Unit	CPSCU	0x4000_8000	CPSCU_NS	0x5000_8000
Direct Memory Access Controller 00	DMAC00	0x4000_A000	DMAC00_NS	0x5000_A000
Direct Memory Access Controller 01	DMAC01	0x4000_A040	DMAC01_NS	0x5000_A040
Direct Memory Access Controller 02	DMAC02	0x4000_A080	DMAC02_NS	0x5000_A080
Direct Memory Access Controller 03	DMAC03	0x4000_A0C0	DMAC03_NS	0x5000_A0C0
Direct Memory Access Controller 04	DMAC04	0x4000_A100	DMAC04_NS	0x5000_A100
Direct Memory Access Controller 05	DMAC05	0x4000_A140	DMAC05_NS	0x5000_A140
Direct Memory Access Controller 06	DMAC06	0x4000_A180	DMAC06_NS	0x5000_A180
Direct Memory Access Controller 07	DMAC07	0x4000_A1C0	DMAC07_NS	0x5000_A1C0
DMAC Module Activation 0	DMA0	0x4000_A800	DMA0_NS	0x5000_A800
Data Transfer Controller 0	DTC0	0x4000_AC00	DTC0_NS	0x5000_AC00
Interrupt Controller	ICU	0x4000_C000	ICU_NS	0x5000_C000
CPU Control Registers	CPU_CTRL	0x4000_F000	CPU_CTRL_NS	0x5000_F000
On-Chip Debug	CPU_OCD	0x4001_1000	CPU_OCD_NS	0x5001_1000
Debug Function	CPU_DBG	0x4001_B000	CPU_DBG_NS	0x5001_B000
CACHE	CACHE	0x4001_C000	CACHE_NS	0x5001_C000
TCM	TCM	0x4001_C800	TCM_NS	0x5001_C800
System Control	SYSC	0x4001_E000	SYSC_NS	0x5001_E000
Inter-Processor Communication	IPC	0x4002_0000	IPC_NS	0x5002_0000
Temperature Sensor Data	TSD	0x02C1_EDA0	TSD_NS	0x12C1_EDA0
MRAM System Register area	MRAM	0x4013_C000	MRAM_NS	0x5013_C000
Event Link Controller	ELC	0x4020_1000	ELC_NS	0x5020_1000
Realtime Clock	RTC	0x4020_2000	RTC_NS	0x5020_2000
Independent Watchdog Timer	IWDT	0x4020_2200	IWDT_NS	0x5020_2200
Clock Frequency Accuracy Measurement Circuit	CAC	0x4020_2400	CAC_NS	0x5020_2400
Watchdog Timer 0	WDT0	0x4020_2600	WDT0_NS	0x5020_2600
Watchdog Timer 1	WDT1	0x4020_2700	WDT1_NS	0x5020_2700
Module Stop Control A, B, C, D, E	MSTP	0x4020_3000	MSTP_NS	0x5020_3000

Table A3.1 Peripheral base address (2 of 4)

Description	Name of secure registers	Base address of secure registers in secure alias region	Name of non-secure registers	Base address of non-secure registers in non-secure alias region
Peripheral Security Control Unit	PSCU	0x4020_4000	PSCU_NS	0x5020_4000
Port Output Enable Module for GPT	POEG	0x4021_2000	POEG_NS	0x5021_2000
Ultra-Low Power Timer 0	ULPT0	0x4022_0000	ULPT0_NS	0x5022_0000
Ultra-Low Power Timer 1	ULPT1	0x4022_0100	ULPT1_NS	0x5022_0100
Low Power Asynchronous General Purpose Timer 0	AGT0	0x4022_1000	AGT0_NS	0x5022_1000
Low Power Asynchronous General Purpose Timer 1	AGT1	0x4022_1100	AGT1_NS	0x5022_1100
12-bit D/A Converter 0	DAC120	0x4023_3000	DAC120_NS	0x5023_3000
12-bit D/A Converter 1	DAC121	0x4023_3100	DAC121_NS	0x5023_3100
Temperature Sensor	TSN	0x4023_5000	TSN_NS	0x5023_5000
High-Speed Analog Comparator 0	ACMPHS0	0x4023_6000	ACMPHS0_NS	0x5023_6000
High-Speed Analog Comparator 1	ACMPHS1	0x4023_6100	ACMPHS1_NS	0x5023_6100
High-Speed Analog Comparator 2	ACMPHS2	0x4023_6200	ACMPHS2_NS	0x5023_6200
High-Speed Analog Comparator 3	ACMPHS3	0x4023_6300	ACMPHS3_NS	0x5023_6300
USB 2.0 FS Module	USBFS	0x4025_0000	USBFS_NS	0x5025_0000
SD Host Interface 0	SDHI0	0x4025_2000	SDHI0_NS	0x5025_2000
SD Host Interface 1	SDHI1	0x4025_2400	SDHI1_NS	0x5025_2400
Pulse Density Modulator Interface	PDMIF	0x4025_6000	PDMIF_NS	0x5025_6000
Serial Sound Interface Enhanced (SSIE) 0	SSIE0	0x4025_D000	SSIE0_NS	0x5025_D000
Serial Sound Interface Enhanced (SSIE) 1	SSIE1	0x4025_D100	SSIE1_NS	0x5025_D100
Inter-Integrated Circuit 0	IIC0	0x4025_E000	IIC0_NS	0x5025_E000
Inter-Integrated Circuit 0 Wake-up Unit	IIC0WU	0x4025_E014	IIC0WU_NS	0x5025_E014
Inter-Integrated Circuit 1	IIC1	0x4025_E100	IIC1_NS	0x5025_E100
Inter-Integrated Circuit 2	IIC2	0x4025_E200	IIC2_NS	0x5025_E200
Octal Serial Peripheral Interface 0	OSPI0_B	0x4026_8000	OSPI0_B_NS	0x5026_8000
Octal Serial Peripheral Interface 1	OSPI1_B	0x4026_8400	OSPI1_B_NS	0x5026_8400
Decryption On-The-Fly 0	DOTF0	0x4026_8800	DOTF0_NS	0x5026_8800
Decryption On-The-Fly 1	DOTF1	0x4026_8900	DOTF1_NS	0x5026_8900
CRC Calculator	CRC	0x4031_0000	CRC_NS	0x5031_0000
Data Operation Circuit	DOC_B	0x4031_1000	DOC_B_NS	0x5031_1000
General PWM 32-bit Timer 0	GPT320	0x4032_2000	GPT320_NS	0x5032_2000
General PWM 32-bit Timer 1	GPT321	0x4032_2100	GPT321_NS	0x5032_2100
General PWM 32-bit Timer 2	GPT322	0x4032_2200	GPT322_NS	0x5032_2200
General PWM 32-bit Timer 3	GPT323	0x4032_2300	GPT323_NS	0x5032_2300
General PWM 32-bit Timer 4	GPT324	0x4032_2400	GPT324_NS	0x5032_2400
General PWM 32-bit Timer 5	GPT325	0x4032_2500	GPT325_NS	0x5032_2500
General PWM 32-bit Timer 6	GPT326	0x4032_2600	GPT326_NS	0x5032_2600

Table A3.1 Peripheral base address (3 of 4)

Description	Name of secure registers	Base address of secure registers in secure alias region	Name of non-secure registers	Base address of non-secure registers in non-secure alias region
General PWM 32-bit Timer 7	GPT327	0x4032_2700	GPT327_NS	0x5032_2700
General PWM 32-bit Timer 8	GPT328	0x4032_2800	GPT328_NS	0x5032_2800
General PWM 32-bit Timer 9	GPT329	0x4032_2900	GPT329_NS	0x5032_2900
General PWM 32-bit Timer 10	GPT3210	0x4032_2A00	GPT3210_NS	0x5032_2A00
General PWM 32-bit Timer 11	GPT3211	0x4032_2B00	GPT3211_NS	0x5032_2B00
General PWM 32-bit Timer 12	GPT3212	0x4032_2C00	GPT3212_NS	0x5032_2C00
General PWM 32-bit Timer 13	GPT3213	0x4032_2D00	GPT3213_NS	0x5032_2D00
Output Phase Switching Controller	GPT_OPS	0x4032_3F00	GPT_OPS_NS	0x5032_3F00
General PWM Timer Clock Controller	GPT_GTCLK	0x4032_3F10	GPT_GTCLK_NS	0x5032_3F10
PWM Delay Generation Circuit	PDG	0x4032_4000	PDG_NS	0x5032_4000
16-bit A/D Converter	ADC_B	0x4033_8000	ADC_B_NS	0x5033_8000
Capture Engine Unit	CEU	0x4034_8000	CEU_NS	0x5034_8000
USB 2.0 High-Speed Module	USBHS	0x4035_1000	USBHS_NS	0x5035_1000
Serial Communication Interface 0	SCI0_B	0x4035_8000	SCI0_B_NS	0x5035_8000
Serial Communication Interface 1	SCI1_B	0x4035_8100	SCI1_B_NS	0x5035_8100
Serial Communication Interface 2	SCI2_B	0x4035_8200	SCI2_B_NS	0x5035_8200
Serial Communication Interface 3	SCI3_B	0x4035_8300	SCI3_B_NS	0x5035_8300
Serial Communication Interface 4	SCI4_B	0x4035_8400	SCI4_B_NS	0x5035_8400
Serial Communication Interface 5	SCI5_B	0x4035_8500	SCI5_B_NS	0x5035_8500
Serial Communication Interface 6	SCI6_B	0x4035_8600	SCI6_B_NS	0x5035_8600
Serial Communication Interface 7	SCI7_B	0x4035_8700	SCI7_B_NS	0x5035_8700
Serial Communication Interface 8	SCI8_B	0x4035_8800	SCI8_B_NS	0x5035_8800
Serial Communication Interface 9	SCI9_B	0x4035_8900	SCI9_B_NS	0x5035_8900
Serial Peripheral Interface 0	SPI0_B	0x4035_C000	SPI0_B_NS	0x5035_C000
Serial Peripheral Interface 1	SPI1_B	0x4035_C100	SPI1_B_NS	0x5035_C100
I3C Bus Interface	I3C	0x4035_F000	I3C_NS	0x5035_F000
Error correction circuit for MBRAM0	ECCMB0	0x4036_F200	ECCMB0_NS	0x5036_F200
Error correction circuit for MBRAM1	ECCMB1	0x4036_F300	ECCMB1_NS	0x5036_F300
CANFD Module 0	CANFD0	0x4038_0000	CANFD0_NS	0x5038_0000
CANFD Module 1	CANFD1	0x4038_2000	CANFD1_NS	0x5038_2000
Ethernet Message Forwarding Engine	MFWD	0x403C_0000	MFWD_NS	0x503C_0000
Layer 3 Ethernet Switch Module	ESWM	0x403C_8000	ESWM_NS	0x503C_8000
Ethernet Common Agent	COMA	0x403C_9000	COMA_NS	0x503C_9000
Ethernet Agent 0	ETHA0	0x403C_A000	ETHA0_NS	0x503C_A000
Ethernet MAC 0	RMAC0	0x403C_B000	RMAC0_NS	0x503C_B000
Ethernet Agent 1	ETHA1	0x403C_C000	ETHA1_NS	0x503C_C000
Ethernet MAC 1	RMAC1	0x403C_D000	RMAC1_NS	0x503C_D000
Ethernet CPU Agent	GWCA0	0x403C_E000	GWCA0_NS	0x503C_E000
Ethernet Generic PTP Timer	GPTP	0x403E_0000	GPTP_NS	0x503E_0000

Table A3.1 Peripheral base address (4 of 4)

Description	Name of secure registers	Base address of secure registers in secure alias region	Name of non-secure registers	Base address of non-secure registers in non-secure alias region
Port 0 Control Registers	PORT0	0x4040_0000	PORT0_NS	0x5040_0000
Port 1 Control Registers	PORT1	0x4040_0020	PORT1_NS	0x5040_0020
Port 2 Control Registers	PORT2	0x4040_0040	PORT2_NS	0x5040_0040
Port 3 Control Registers	PORT3	0x4040_0060	PORT3_NS	0x5040_0060
Port 4 Control Registers	PORT4	0x4040_0080	PORT4_NS	0x5040_0080
Port 5 Control Registers	PORT5	0x4040_00A0	PORT5_NS	0x5040_00A0
Port 6 Control Registers	PORT6	0x4040_00C0	PORT6_NS	0x5040_00C0
Port 7 Control Registers	PORT7	0x4040_00E0	PORT7_NS	0x5040_00E0
Port 8 Control Registers	PORT8	0x4040_0100	PORT8_NS	0x5040_0100
Port 9 Control Registers	PORT9	0x4040_0120	PORT9_NS	0x5040_0120
Port A Control Registers	PORTA	0x4040_0140	PORTA_NS	0x5040_0140
Port B Control Registers	PORTB	0x4040_0160	PORTB_NS	0x5040_0160
Port C Control Registers	PORTC	0x4040_0180	PORTC_NS	0x5040_0180
Port D Control Registers	PORTD	0x4040_01A0	PORTD_NS	0x5040_01A0
Port E Control Registers	PORTE	0x4040_01C0	PORTE_NS	0x5040_01C0
Port F Control Registers	PORTF	0x4040_01E0	PORTF_NS	0x5040_01E0
Port G Control Registers	PORTG	0x4040_0200	PORTG_NS	0x5040_0200
Pmn Pin Function Control Register	PFS	0x4040_0800	PFS_NS	0x5040_0800

Note: Name = Peripheral name
Description = Peripheral functionality
Base address = Lowest reserved address or address used by the peripheral

3.2 Access Cycles

This section provides access cycle information for the I/O registers described in this manual.

- Registers are grouped by associated module.
- The number of access cycles indicates the number of cycles based on the specified reference clock.
- In the internal I/O area, reserved addresses that are not allocated to registers must not be accessed, otherwise operations cannot be guaranteed.
- The number of I/O access cycles depends on bus cycles of the internal peripheral bus, divided clock synchronization cycles, and wait cycles of each module. Divided clock synchronization cycles differ depending on the frequency ratio between ICLK and PCLK.
- When the frequency of ICLK is equal to that of PCLK, the number of divided clock synchronization cycles is always constant.
- When the frequency of ICLK is greater than that of PCLK, at least 1 PCLK cycle is added to the number of divided clock synchronization cycles.
- The number of write access cycles indicates the number of cycles obtained by non-bufferable write access.

Note: This applies to the number of cycles when access from the CPU does not conflict with the instruction fetching to the external memory or bus access from other bus masters such as DTC or DMAC.

Table A3.2 Access cycles (1 of 3)

Peripheral base address symbol	Address[*1]		Number of access cycles					Related function
			ICLK = PCLK		ICLK > PCLK[*2]		Cycle unit	
	From	To	Read	Write	Read	Write		
RMPU, SRAM, BUS, ICU_COMMON, CPSCU, DMAC0n, DMA0, DTC0, ICU, CPU_CTRL, CPU_OCD, CPU_DBG	0x4000_0000	0x4001_BFFF	3	2	3	2	ICLK	Renesas Memory Protection Unit, SRAM Control, BUS Control, Common Interrupt Controller, CPU System Security Control Unit, Direct memory access controller 0 n, DMAC Module Activation 0, Data Transfer Controller 0, Interrupt Controller, CPU Control Registers, On-Chip Debug, Debug Function
CACHE, TCM	0x4001_C000	0x4001_CFFF	5	4	5	4	ICLK	CM33 Cache, CM33 Tightly Coupled Memory
SYSC	0x4001_E000	0x4001_E9FF	4	3	2 to 4	1 to 3	PCLKB	System Control
SYSC	0x4001_EA00	0x4001_ED7F	7	6	5 to 7	4 to 6	PCLKB	System Control
IPC	0x4002_0000	0x4002_FFFF	3	2	3	2	ICLK	Inter-Processor Communication
ELC, RTC	0x4020_1000	0x4020_21FF	4	3	2 to 4	1 to 3	PCLKB	Event Link Controller, Realtime Clock
IWDT	0x4020_2200	0x4020_22FF	4	65	2 to 4	63 to 65	PCLKB	Independent Watchdog Timer
CAC, WDTn, MSTP, PSCU, POEG	0x4020_2400	0x4021_2FFF	4	3	2 to 4	1 to 3	PCLKB	Clock Frequency Accuracy Measurement Circuit, Watchdog Timer n, Module Stop Control, Peripheral Security Control Unit, Port Output Enable Module for GPT
ULPTn	0x4022_0000	0x4022_01FF	6	65	4 to 6	63 to 65	PCLKB	Ultra-Low Power Timer n
AGTn	0x4022_1000	0x4022_11FF	6	3	4 to 6	1 to 3	PCLKB	Low Power Asynchronous General purpose Timer n
DAC12n, TSN	0x4023_3000	0x4023_5FFF	4	3	2 to 4	1 to 3	PCLKB	12-bit D/A Converter n, Temperature Sensor
ACMPHSn	0x4023_6000	0x4023_63FF	3	3	1 to 3	1 to 3	PCLKB	High-Speed Analog Comparator n
USBFS	0x4025_0000	0x4025_03FF	5	4	3 to 5	2 to 4	PCLKB	USB 2.0 FS Module
USBFS	0x4025_0400	0x4025_04FF	4	65	2 to 4	63 to 65	PCLKB	USB 2.0 FS Module
SDHIn, PDMIF, SSIEn, IICn, OSPIn, DOTFn	0x4025_2000	0x4026_89FF	4	3	2 to 4	1 to 3	PCLKB	SD Host Interface n, Pulse Density Modulation Interface, Serial Sound Interface Enhanced n, Inter-Integrated Circuit n, Octal Serial Peripheral Interface n, Decryption On-The-Fly n

Table A3.2 Access cycles (2 of 3)

Peripheral base address symbol	Address*1		Number of access cycles					
			ICLK = PCLK		ICLK > PCLK*2		Cycle unit	Related function
	From	To	Read	Write	Read	Write		
CRC, DOC	0x4031_0000	0x4031_1FFF	4	3	2 to 4	1 to 3	PCLKA	CRC Calculator, Data Operation Circuit
GPT32n, GPT_OPS	0x4032_2000	0x4032_3F0F	9	6	7 to 9	4 to 6	PCLKA	General PWM 32-Bit Timer n, Output Phase Switching Controller
GPT_GTCLK	0x4032_3F10	0x4032_3F1F	4	3	2 to 4	1 to 3	PCLKA	General PWM Timer Clock Control
PDG	0x4032_4000	0x4032_4FFF	3	2	1 to 3	0 to 2	PCLKA	PWM Delay Generation Circuit
ADC_B	0x4033_8000	0x4034_7FFF	4	3	2 to 4	1 to 3	PCLKA	A/D Converter
CEU	0x4034_8000	0x4034_FFFF	7	5	5 to 7	3 to 5	PCLKA	Capture Engine Unit
USBHS*3	0x4035_1000	0x4035_115F	BWAIT+4	BWAIT+3	(BWAIT+2) to (BWAIT+4)	(BWAIT+1) to (BWAIT+3)	PCLKA	USB 2.0 High-Speed Module
USBHS*3	0x4035_1160	0x4035_1167	BWAIT+4	130	(BWAIT+2) to (BWAIT+4)	128 to 130	PCLKA	USB 2.0 High-Speed Module
USBHS	0x4035_1168	0x4035_116F	8	130	6 to 8	128 to 130	PCLKA	USB 2.0 High-Speed Module
SCIn, SPIn, I3C	0x4035_8000	0x4035_FFFF	4	3	2 to 4	1 to 3	PCLKA	Serial Communication Interface n, Serial Peripheral Interface n, I3C Bus Interface
ECCMBn	0x4036_F200	0x4036_F3FF	5	4	3 to 5	2 to 4	PCLKA	Error correction circuit for MBRAMn
CANFDn	0x4038_0000	0x4038_3FFF	4	3	2 to 4	1 to 3	PCLKA	CANFD Module n
ESWM	0x403C_0000	0x403E_FFFF	—	—	7 to 12	2 to 4	PCLKA	Layer 3 Ethernet Switch Module
PORTn	0x4040_0000	0x4040_01FF	4	2	4	2	ICLK	Port n Control Registers
PFS	0x4040_0800	0x4040_0FFF	8	2	8	2	ICLK	Pmn Pin Function Control Register
RSIP-E50D	-	-	3 to 5	2	1 to 6	0 to 2	PCLKA	Renesas Secure IP

Table A3.2 Access cycles (3 of 3)

Peripheral base address symbol	Address*1		Number of access cycles					
			ICLK = MRPCLK		ICLK > MRPCLK*2		Cycle unit	Related function
	From	To	Read	Write	Read	Write		
MRAM	0x4013_0000	0x4013_FFFF	4	3	2 to 4	1 to 3	MRPCLK	MRAM control

Note 1. This table only shows secure address. Access cycle of the non-secure address is the same as its secure address.
Note 2. If the number of PCLK or MRPCLK cycles is non-integer (for example 1.5), the minimum value is without the decimal point, and the maximum value is rounded up to the decimal point. For example, 1.5 to 2. 5 is 1 to 3.
Note 3. BWAIT is the number of waits (not cycles) described in the USBHS.BUSWAIT register.

Appendix 4. Notes for Register R/W

- A secure bus master issues a secure access using an address marked as secure by IDAU/SAU or MSAU
- A secure bus master issues a non-secure access using an address marked as non-secure by IDAU/SAU or MSAU
- A non-secure bus master issues a non-secure access using an address marked as non-secure by IDAU/SAU or MSAU.

Table A4.1 Type of register notes (S-TYPE)

TYPE	UM description
S-TYPE-1	Only Secure access can write to this register. Read access is always allowed. Non-secure write access is ignored, but TrustZone access error is not generated.
S-TYPE-2	Read access is always allowed If the security attribution is configured as Secure: ● Secure write access is allowed ● Non-secure write access is ignored, but TrustZone access error is not generated.
	If the security attribution is configured as Non-secure: ● Secure write access is ignored, but TrustZone access error is not generated ● Non-secure access is allowed.
S-TYPE-3	If the security attribution is configured as Secure: ● Secure access is allowed ● Non-secure write access is ignored and Non-secure read access is read as 0, TrustZone access error is generated.
	If the security attribution is configured as Non-secure: ● Secure write access is ignored and Secure read access is read as 0, TrustZone access error is generated ● Non-secure access is allowed.
S-TYPE-4	If the security attribution is configured as Secure: ● Secure access is allowed ● Non-secure write access is ignored and Non-secure read access is read as 0, but TrustZone access error is not generated.
	If the security attribution is configured as Non-secure: ● Secure write access is ignored and Secure read access is read as 0, but TrustZone access error is not generated ● Non-secure access is allowed.
S-TYPE-5	Access is always allowed.
S-TYPE-6	Secure access is allowed. Non-secure write access is ignored, and Non-secure read access is read as 0, TrustZone access error is generated.
S-TYPE-7	Secure write access is ignored, and Secure read access is read as 0, TrustZone access error is generated. Non-secure access is allowed.

Note: A non-secure bus master does NOT issue any access using an address marked as secure by IDAU/SAU or MSAU.

Table A4.2 Type of register notes (P-TYPE)

TYPE	UM description
P-TYPE-1	Privileged write access is allowed. Read access is always allowed. Unprivileged write access is ignored, but TrustZone access error is not generated.
P-TYPE-2	Privileged access is allowed. Unprivileged write access is ignored, and Unprivileged read access is read as 0, TrustZone access error is generated.
P-TYPE-3	If the privilege attribution is configured as Privileged: ● Privileged access is allowed ● Unprivileged write access is ignored and Unprivileged read access is read as 0, TrustZone access error is generated.
	If the privilege attribution is configured as Unprivilege: ● Privileged access and Unprivileged access are allowed.
P-TYPE-4	If the privilege attribution is configured as Privileged: ● Privileged access is allowed ● Unprivileged write access is ignored and Unprivileged read access is read as 0, TrustZone access error is not generated.
	If the privilege attribution is configured as Unprivilege: ● Privileged access and Unprivileged access are allowed.
P-TYPE-5	Access is always allowed.

Appendix 5. Peripheral Variant

Table A5.1 shows the correspondence between the module name used in this manual and the peripheral variant.

Table A5.1 Module name vs peripheral variant

Module name	Peripheral variant
SCI	SCI_B
SPI	SPI_B
OSPI	OSPI_B
ADC16H	ADC_B
DAC12	DAC_B
DOC	DOC_B

Revision History

Revision 1.00— February 14, 2025

First edition, issued

Revision 1.10 — May 16, 2025

1. Overview:
- Updated Figure 1.2 Part numbering scheme.

2. CPU:
- Updated Table 2.6 Initial secure vector table base address in Single-chip mode.
- Updated 2.8.1 External Debugger Connectivity, Authentication by JTAG/SWD.
- Updated Table 2.10 Condition for authentication level.
- Updated 2.9.1.7 CPUnINITVTOR : CPU Initial Vector Base Address Register (n = 0, 1)

3. Inter-Processor Communication:
- Updated 3.2.10 IPC0STA0 : Inter-Processor0 Status Register0.
- Updated 3.2.12 IPC0TXD0 : Inter-Processor0 FIFO Transfer Data Register0 and 3.2.13 IPC0RXD0 : Inter-Processor0 FIFO Receive Data Register0.
- Updated 3.2.15 IPC0STA1 : Inter-Processor0 Status Register1.
- Updated 3.2.17 IPC0TXD1 : Inter-Processor0 FIFO Transfer Data Register1 and 3.2.18 IPC0RXD1 : Inter-Processor0 FIFO Receive Data Register1.
- Updated 3.2.20 IPC1STA0 : Inter-Processor1 Status Register0.
- Updated 3.2.22 IPC1TXD0 : Inter-Processor1 FIFO Transfer Data Register0 and 3.2.23 IPC1RXD0 : Inter-Processor1 FIFO Receive Data Register0.
- Updated 3.2.25 IPC1STA1 : Inter-Processor1 Request Status Register1.
- Updated 3.2.27 IPC1TXD1 : Inter-Processor1 FIFO Transfer Data Register1 and 3.2.28 IPC1RXD1 : Inter-Processor1 FIFO Receive Data Register1.
- Updated 3.3.2 Inter-Processor Non-maskable Interrupt.
- Updated Figure 3.3 Inter-Processor non-maskable interrupt.
- Updated 3.3.4 Exclusive Control Flow.
- Updated Figure 3.6 Inter-processor message FIFO.
- Updated Figure 3.7 Inter-processor message FIFO timing.

6. Resets:
- Updated Table 6.3 Module-related registers initialized by each reset source.

7. Option-Setting Memory:
- Updated 7.2.4 OFS1, OFS1_SEC : Option Function Select Register 1 for Non-Secure and Secure.
- Updated 7.2.16 SAMR : Start Address of Measurement Report.

Revision 1.10 — May 16, 2025

9. Clock Generation Circuit:
- Updated 9.2.1 CGFSAR : Clock Generation Function Security Attribute Register.
- Updated 9.2.16 HOCOCR : High-Speed On-Chip Oscillator Control Register.
- Added HOCOCR2 : High-Speed On-Chip Oscillator Control Register2.
- Updated 9.2.19 FLLCR2 : FLL Control Register2.
- Updated 9.2.33 LOCOUTCR : LOCO User Trimming Control Register and 9.2.34 MOCOUTCR : MOCO User Trimming Control Register.
- Updated 9.2.38 USBCKDIVCR : USB Clock Division Control Register, 9.2.39 OCTACKDIVCR : Octal-SPI Clock Division Control Register, 9.2.40 CANFDCKDIVCR : CANFD Core Clock Division Control Register, 9.2.41 USB60CKDIVCR : USB60 Clock Division Control Register, 9.2.42 I3CCKDIVCR : I3C Clock Division Control Register, 9.2.43 USBCKCR : USB Clock Control Register, 9.2.44 OCTACKCR : Octal-SPI Clock Control Register, 9.2.45 CANFDCKCR : CANFD Core Clock Control Register, 9.2.46 USB60CKCR : USB60 Clock Control Register, 9.2.47 I3CCKCR : I3C Clock Control Register, 9.2.48 SCICKDIVCR : SCI Clock Division Control Register, 9.2.49 SPICKDIVCR : SPI Clock Division Control Register, 9.2.50 ADCCKDIVCR : ADC Clock Division Control Register, 9.2.51 GPTCKDIVCR : GPT Clock Division Control Register, 9.2.52 SCICKCR : SCI Clock Control Register, 9.2.53 SPICKCR : SPI Clock Control Register, 9.2.54 ADCCKCR : ADC Clock Control Register, and 9.2.55 GPTCKCR : GPT Clock Control Register.
- Updated 9.2.58 ESWCKDIVCR : EtherSW Clock Division Control Register, 9.2.59 ESWPCKDIVCR : EtherSW-PHY Clock Division Control Register, and 9.2.60 ETHPCKDIVCR : Ether-PHY Clock Division Control Register.
- Updated 9.2.62 ESWCKCR : EtherSW Clock Control Register, 9.2.63 ESWPCKCR : EtherSW-PHY Clock Control Register, and 9.2.64 ETHPCKCR : Ether-PHY Clock Control Register.
- Updated 9.10.1 CPU0 Clock (CPUCLK0), 9.10.2 CPU1 Clock (CPUCLK1), 9.10.3 MRAM Code Bus Clock (MRICLK), 9.10.4 MRAM Peripheral Bus Clock (MRPCLK), and 9.10.5 System Clock (ICLK).
- Updated 9.10.7 Peripheral Module Clock (PCLKA, PCLKB, PCLKC, PCLKD, PCLKE), 9.10.8 External Bus Clock (BCLK, EBCLK), and 9.10.9 SDRAM Clock (SDCLK).
- Updated 9.10.11 SCI Clock (SCICLK), 9.10.12 SPI Clock (SPICLK), 9.10.13 Octal-SPI Clock (OCTACLK, OCTADIVCLK), 9.10.14 CANFD Core Clock (CANFDCLK), 9.10.15 USB Clock (USBCLK), and 9.10.16 USB Clock (USB60CLK).
- Updated 9.10.18 ADC Clock (ADCCLK), 9.10.19 GPT Clock (GPTCLK), 9.10.20 I3C Clock (I3CCLK), 9.10.21 Asynchronous External Bus Clock (BCLKA), 9.10.22 EtherSW Clock (ESWCLK), 9.10.23 EtherSW-PHY Clock (ESWPHYCLK), and 9.10.24 Ether-PHY Clock (ETHPHYCLK).
- Updated 9.10.33 External Pin Output Clock (CLKOUT).
- Updated Table 9.14 Example of the HOCO initial setting procedure after reset release / after Deep Software Standby cancellation (OFS1(_SEC).HOCOEN = 1, without FLL).
- Updated Table 9.15 Example of the HOCO setting procedure with FLL function after reset release / after Deep Software Standby cancellation (OFS9.HOCOEN = 1).

11. Low Power Modes:
- Updated Table 11.2 Operating state of processor low power mode.
- Updated Table 11.3 Operating state of each low power mode.
- Updated 11.2.20 DPSBYCR : Deep Software Standby Control Register.
- Updated 11.8.6 Transition to Low Power Modes.

13. Register Write Protection Function:
- Updated Table 13.1 Association between PRCR bits and use of registers to be protected.

14. Interrupt Controller Unit:
- Updated 14.2.12 IRQCRi : IRQ Control Register (i = 0 to 31).
- Updated 14.2.19 WUPEN0 : Wake Up Interrupt Enable Register 0 and 14.2.20 WUPEN1 : Wake Up interrupt enable register 1.
- Updated Table 14.5 Event List.
- Updated 14.9.2 Return from CPU Deep Sleep Mode and 14.9.3 Return from Software Standby Mode.

15. Buses:
- Modified the figure title in Figure 15.23 Example of write access operation with address/data multiplexed I/O interface.
- Modified the figure title in Figure 15.24 Example of bus timing with address/data multiplexed I/O interface.
- Added the Access Mode register to Table 15.33 Conditions for register modification.

17. DMA Controller:
- Added the new explanation to 17.3.1.4 Repeat-Block Transfer Mode.

18. Data Transfer Controller:
- Modified Figure 18.1 DTC block diagram.

20. I/O Ports:
- Modified Table 20.9 Register settings for input/output pin function (PORT2).

21. Port Output Enable for GPT:
- Modified Figure 21.1 POEG block diagram.

29. Layer 3 Ethernet Switch Module:
- Modified Table 29.1 ESWM specifications.
- Modified 29.2.2.6 TPEMIMC6t : Error and Monitoring Interrupt Mapping Configuration Register 6t.
- Modified 29.2.2.7 TPEMIMC7t : Error and Monitoring Interrupt Mapping Configuration Register 7t.

Revision 1.10 — May 16, 2025

30. Ethernet Message Forwarding Engine:
- Modified bit symbol suffix.
- Updated 30.3.1.5.2 FWLTHHC : L3 Hash Configuration Register.
- Updated 30.3.1.6.2 FWMACHC : MAC Hash Configuration Register.
- Updated 30.3.1.10.13 FWSEQNRC : Sequence Number Reset Configuration Register.

31. Ethernet Common Agent:
- Updated 31.3.1.4 RCDC : Clock Disable Configuration Register.
- Updated 31.3.2.3 CABPPFLCi : Buffer Pointer Pause Frame Level i Configuration Register, 31.3.2.4 CABPPWMLCi : Port i Buffer Pool Watermark Level Configuration Register (i = 0 to 2), 31.3.2.5 CABPPPFLCij : Port i Buffer Pointer Pause Frame Level j Configuration Register (i = 0 to 2) (j = 0, 1), and 31.3.2.6 CABPULCi : Port i Buffer Pointer Utilization Level Configuration Register i (i = 0 to 2).
- Updated 31.3.2.11 CABPMCPMi : Port i Buffer Pointer Maximum Count Monitoring Register (i = 0 to 2).
- Updated 31.5.1.1 IPV-based Watermark.
- Updated 31.5.1.3 Per-port Level-based Watermark i (i = 0 to 3).
- Updated 31.5.2.2 Per-port Pause Function i (i = 0 to 3).
- Updated 31.5.3 Per-port Memory Allocation Function.
- Updated 31.5.3.2 Reduced Memory Setting 1 (Recommended Setting) and 31.5.3.3 Reduced Memory Setting 2 (Recommended Setting).

32. Ethernet Agent:
- Modified 32.3.2.4 EATDQAC : TX Descriptor Queue Arbitration Configuration Register.
- Modified 32.3.2.9 EATDQMLMq : Transmission Descriptor Queue q Max Level Monitoring Register.
- Modified 32.3.4.2 EACC : CBS Configuration Register.
- Modified 32.3.4.5 EACOEM : CBS Oper Enable Monitoring Register.
- Modified 32.3.4.6 EACOIVMq : CBS Oper Increment Value Monitoring Register q.
- Modified 32.3.4.7 EACOULMq : CBS Oper Upper Limit Monitoring Register q.
- Modified 32.3.5.3 EATASENCi : TAS Entry Number Configuration Register i.
- Modified 32.3.5.4 EATASCTENC : TAS Cut-Through Entry Number Configuration Register.
- Modified 32.3.5.5 EATASENMi : TAS Entry Number Monitoring Register i.
- Modified 32.3.7.1.1 EAEIS0 : Error Interrupt Status Register 0.
- Modified 32.3.7.1.4 EAEIS1 : Error Interrupt Status Register 1.
- Modified 32.3.7.1.7 EAEIS2 : Error Interrupt Status Register 2.
- Modified Figure 32.9 CBS q setting flow.
- Modified Figure 32.10 CBS q disabling flow.
- Modified Table 32.5 Descriptor store ETHA/GWCA register association.
- Modified 32.5.1.3 Frame Size Check.
- Modified 32.5.1.5 CBS (Credit-based shaper) [802.1Qav].
- Modified 32.5.1.5.1 CBS Setting.
- Modified 32.5.1.6 TAS (Time Aware Shaper) [802.1Qbv].
- Modified Figure 32.19 TAS behavior when its schedule starts in the future.
- Modified Figure 32.20 TAS behavior when its schedule should have started in the past.
- Modified Figure 32.22 TAS behavior when its schedule reconfiguration should have happened in the past.
- Modified Figure 32.25 TAS recovery during gPTP offset correction.
- Modified 32.5.1.7.2 Preemption.

33. Ethernet MAC:
- Modified 33.4.1.7 MTPFC2 : Transmission Pause or PFC Frame Configuration Register 2.
- Modified Table 33.6 Reception error list.
- Modified 33.4.3.2 MEIE : Error Interrupt Enable Register.
- Modified 33.4.3.3 MEID : Error Interrupt Disable Register.
- Modified Figure 33.6 Initialization flow.
- Modified Figure 33.18 PFC frame format.
- Modified 33.6.2.2.1 Transmission of PFC frame.
- Modified 33.6.2.2.2 Reception of PFC frame.
- Modified Figure 33.22 Reception of PAUSE frame.
- Modified 33.6.3.1 Timestamp capture.
- Modified Figure 33.26 Example configuration of PTP message filter.

34. Ethernet CPU Agent:
- Modified bit symbol suffix.

35. Ethernet Generic PTP Timer:
- Modified 35.3.2.2 PTPTMDC : Timer Disable Configuration Register.
- Modified 35.6.3.6 POTPERnU : Period Setting Register nU (n = 0 to 3).

44. Octal Serial Peripheral Interface:
- Updated Table 44.8 Error list (n = 0, 1).
- Added 44.5.8 Block Protect for SiP Product.

49. Pulse Density Modulation Interface:
- Updated Table 49.2 PDM I/O pins (n = 0 to 2).
- Updated Table 49.20 PDM-IF interrupt list.

Revision 1.10 — May 16, 2025

51. Security Features:
- Updated Figure 51.14 Format and location of the measurement report.

52. Renesas Secure IP:
- Updated Table 52.1 Security engine specifications.

53. 16-bit A/D Converter:
- Updated 53.11.20 Sampling Error and Sampling Time Estimation.
- Updated 53.11.22 Notes on simultaneous operation with other modules.

54. 12-Bit D/A Converter:
- Updated base address.
- Updated 54.1 Overview.

61. Internal Voltage Regulator:
- Updated 61.2.2 External VDD Mode.

62. Electrical Characteristics:
- Updated 62.Electrical Characteristics.
- Updated 62.1 Absolute Maximum Ratings.
- Updated 62.2.5 Operating and Standby Current.
- Updated 62.3.10 SPI Timing.
- Updated 62.3.11 OSPI Timing.
- Updated 62.6 ADC Characteristics.
- Added 62.11 External VDD timing Characteristics.

Appendix:
- Updated Figure A2.3 HLQFP 176-pin in Appendix 2.
- Updated Table A5.1 Module name vs peripheral variant in Appendix 5.

Revision 1.20 — Jul 31, 2025

Specification Differences Between Products:
- Updated Table 1.1 Specification Differences Depending on Chip Versions

1. Overview:
- Updated Table 1.8 Communication interfaces
- Updated Table 1.15 Pin functions
- Updated Table 1.16 Pin list for the Standard product
- Updated Table 1.17 Pin list for the SiP product

2. CPU:
- Updated Table 2.26 DBGREG registers

9. Clock Generation Circuit:
- Updated 9.2.25 SOSTDCR : Sub-clock Oscillation Stop Detection Control Register
- Updated 9.9.3 Sub-clock Oscillation Stop Detection and Operation after Detection

11. Low Power Modes:
- Updated 11.7 Voltage Scaling Control

14. Interrupt Controller Unit:
- Modified offset address of 14.2.12 IRQCRi : IRQ Control Register (i = 0 to 31)
- Updated 14.2.13 NMISR : Non-Maskable Interrupt Status Register
- Updated 14.2.14 NMIER : Non-Maskable Interrupt Enable Register
- Updated 14.2.15 NMICLR : Non-Maskable Interrupt Status Clear Register
- Updated 14.2.16 NMICR : NMI Pin Interrupt Control Register
- Updated 14.2.17 IELSRn : Interrupt Controller Unit Event Link Setting Register n (n = 0 to 95)
- Updated 14.2.18 DELSRm : DMAC Event Link Setting Register m (m = 0 to 7)
- Updated 14.2.19 WUPEN0 : Wake Up Interrupt Enable Register 0
- Updated 14.2.20 WUPEN1 : Wake Up Interrupt Enable Register 1
- Updated 14.2.21 DSLPWUPIRQENj : Deep Sleep Wake Up IRQ Enable Register (j = 0 to 2)
- Updated Table 14.5 Event List

15. Buses:
- Updated Table 15.3 Addresses assigned for each slave

19. Event Link Controller:
- Updated Table 19.3 Association between event signal names set in ELSRn.ELS[9:0] bits and signal numbers

20. I/O Ports:
- Updated Table 20.8 Register settings for input/output pin function (PORT1)
- Updated Table 20.11 Register settings for input/output pin function (PORT4)
- Updated Table 20.14 Register settings for input/output pin function (PORT7)
- Updated Table 20.15 Register settings for input/output pin function (PORT8)

Revision 1.20 — Jul 31, 2025

22. General PWM Timer:
- Updated Table 22.2 GPT functions
- Updated Table 22.71 Interrupt sources
- Updated 22.5 A/D Conversion Start Request
- Updated Figure 22.151 Example of A/D Conversion Start Request Timing Operation
- Updated 22.6.1 Event Signal Output to ELC
- Updated Figure 22.172 Example of A/D Conversion Start Request Event Interval

29. Layer 3 Ethernet Switch Module:
- Deleted 29.2.2.1 TPEMIMC0 : Error and Monitoring Interrupt Mapping Configuration Register 0
- Updated Table 29.13 Interrupt and event output

32. Ethernet Agent:
- Updated 32.3.5.3 EATASENCi : TAS Entry Number Configuration Register i (i = 0 to 7)
- Updated 32.3.5.5 EATASENMi : TAS Entry Number Monitoring Register i (i = 0 to 7)

33. Ethernet MAC:
- Updated Figure 33.6 Initialization flow

35. Ethernet Generic PTP Timer:
- Updated Figure 35.1 GPTP block diagram
- Modified offset address of 35.6.3.4 POTSTRnM : Pulse Output Start Time Setting Register nM (n = 0 to 3)
- Modified offset address of 35.6.3.5 POTSTRnL : Pulse Output Start Time Setting Register nL (n = 0 to 3)
- Modified offset address of 35.6.3.6 POTPERnU : Period Setting Register nU (n = 0 to 3)
- Modified offset address of 35.6.3.7 POTPERnM : Period Setting Register nM (n = 0 to 3)
- Modified offset address of 35.6.3.8 POTPERnL : Period Setting Register nL (n = 0 to 3)
- Modified offset address of 35.6.3.9 POTPWRn : Pulse Width Setting Register n (n = 0 to 3)
- Modified offset address of 35.6.3.10 POTCPRnU : Time Capture Register nU (n = 0 to 3)
- Modified offset address of 35.6.3.11 POTCPRnM : Time Capture Register nM (n = 0 to 3)
- Modified offset address of 35.6.3.12 POTCPRnL : Time Capture Register nL (n = 0 to 3)

37. USB 2.0 High-Speed Module:
- Updated 37.2.34 PIPECFG : Pipe Configuration Register
- Modified offset address of 37.2.38 PIPEnCTR : Pipe n Control Register (n = 1 to 9)

40. I3C Bus Interface:
- Modified offset address of 40.2.64 SDATBASn : Slave Device Address Table Basic Register n (n = 0 to 2)

41. CAN with Flexible Data-rate:
- Modified base address of CANFDn_NS

59. MRAM:
- Updated Figure 59.5 Frequency up procedure

62. Electrical Characteristics:
- Updated Table 62.16 Current of high-speed mode, maximum data processing, CPU0 Deep Sleep, CPU1 active, peripheral clock ON (DCDC mode)
- Updated Table 62.18 Current of high-speed mode, maximum data processing (MVE operation), peripheral clock OFF (DCDC mode)
- Updated Table 62.19 Current of high-speed mode, maximum data processing (MVE operation), peripheral clock OFF (External VDD mode)
- Updated Table 62.20 Current of high-speed mode, maximum data processing (MVE operation), CPU0 active, CPU1 Deep Sleep, peripheral clock OFF (DCDC mode)
- Updated Table 62.21 Current of high-speed mode, maximum data processing (MVE operation), CPU0 active, CPU1 Deep Sleep, peripheral clock OFF (External VDD mode)
- Updated Table 62.22 Current of high-speed mode, maximum data processing, CPU0 Deep Sleep, CPU1 active, peripheral clock OFF (DCDC mode)
- Updated Table 62.23 Current of high-speed mode, maximum data processing, CPU0 Deep Sleep, CPU1 active, peripheral clock OFF (External VDD mode)
- Updated Table 62.24 Current of high-speed mode, CPU Sleep mode (DCDC mode)
- Updated Table 62.26 Current of high-speed mode, CPU0 Sleep, CPU1 Deep Sleep (DCDC mode)
- Updated Table 62.28 Current of high-speed mode, CPU0 Deep Sleep, CPU1 Sleep (DCDC mode)
- Updated Table 62.30 Current of high-speed mode, CPU Deep Sleep mode (DCDC mode)

Revision 1.30 — Feb 27, 2026

1. Overview:
- Updated Figure 1.2 Part numbering scheme
- Updated Table 1.13 Product list
- Updated Table 1.14 Function Comparison
- Updated Table 1.15 Pin functions
- Added Figure 1.5 Pin assignment for BGA 169-pin
- Updated Figure 1.6 Pin assignment for BGA 303-pin
- Updated Table 1.16 Pin list for the Standard product
- Updated Table 1.17 Pin list for the SiP product

2. CPU:
- Updated Table 2.11 Controlling of resets and interrupts generation in OCD mode
- Updated 2.17.3.5 TCMCRx : TCM Control Register (x = C or S)

5. Address Space:
- Updated Figure 5.2 1 MB MRAM product for Standard Product
- Added 5.3 Peripheral I/O Register Address Space

6. Resets:
- Updated Table 6.3 Module-related registers initialized by each reset source

7. Option-Setting Memory:
- Updated 7.1 Overview
- Updated Table 7.2 The relationship between FSBLCLK setting and system clock frequency
- Updated 7.2.18 HOEMRTRKn : Hash of OEM_ROOT_PKn (n= 0 to 3)
- Updated 7.2.20 ARCLS : Anti-Rollback Counter Lock Setting
- Updated 7.2.21 ARCCS : Anti-Rollback Counter Configuration Setting for Non-secure Application
- Updated 7.2.22 ARC_SECn : Anti-Rollback Counter for Secure Application n (n = 0, 1)
- Updated 7.2.23 ARC_NSECn : Anti-Rollback Counter for Non-secure Application n (n = 0 to 7)
- Updated 7.2.24 ARC_OEMBLn : Anti-Rollback Counter for OEMBL (n = 0, 1)
- Updated 7.2.25 GPOTPn : General Purpose OTP (n = 0 to 23)

9. Clock Generation Circuit:
- Updated 9.2.1 CGFSAR : Clock Generation Function Security Attribute Register
- Updated 9.2.25 SOSTDCR : Sub-clock Oscillation Stop Detection Control Register
- Added 9.2.66 USBCKSELR : USB Clock Selection Register
- Updated 9.9.3 Sub-clock Oscillation Stop Detection and Operation after Detection
- Updated Figure 9.11 Flow of Recovery from Detection of Sub Clock Oscillator Stop
- Updated Table 9.7 Example of the initial system clock setting procedure after reset release/after Deep Software Standby cancellation
- Updated Figure 9.14 Additional flow of initial system clock setting when PLL is used for the system clock source (assuming all module stop bits are initial value)
- Updated Table 9.8 Example of setting procedure when changing the current system clock frequency to a faster frequency
- Updated Figure 9.15 Additional flow of system clock setting when PLL is used for the system clock source and CPUCLK0 to be faster frequency
- Updated Table 9.9 Example of setting procedure when changing the current system clock frequency to a slower frequency
- Updated Figure 9.16 Additional flow of system clock setting when PLL is used for the system clock source and CPUCLK0 to be slower frequency

10. Clock Frequency Accuracy Measurement Circuit:
- Updated Figure 10.1 CAC block diagram

11. Low Power Modes:
- Updated Table 11.2 Operating state of processor low power mode
- Updated Table 11.3 Operating state of each low power mode
- Updated Figure 11.1 Low power mode transitions
- Updated Table 11.8 Setting of the bits what affect the mode after executing WFI instruction
- Updated 11.6.2.2 Canceling Software Standby Mode
- Updated 11.6.3.2 Canceling Deep Software Standby Mode
- Added 11.8.11 Notes about the WFE instruction or Sleep-on-exit function

13. Register Write Protection:
- Updated Table 13.1 Association between PRCR bits and use of registers to be protected

14. Interrupt Controller Unit:
- Updated Figure 14.2 ICU Register Address Map
- Updated Table 14.5 Event List

15. Buses:
- Updated Figure 15.1 System bus connection
- Updated Table 15.3 Addresses assigned for each slave
- Updated 15.3.1 BUSSARA : Bus Security Attribution Register A

Revision 1.30 — Feb 27, 2026

17. DMA Controller:
- Updated Table 17.1 DMAC specifications
- Updated 17.2 Register Descriptions

18. Data Transfer Controller:
- Updated Table 18.1 DTC specifications
- Updated 18.2 Register Descriptions

19. Event Link Controller:
- Updated Table 19.3 Association between event signal names set in ELSRn.ELS[9:0] bits and signal numbers

20. I/O Ports:
- Updated Table 20.1 I/O port specifications
- Updated 20.2.1 PCNTR1/PODR/PDR : Port Control Register 1
- Added Table 20.7 Handling of reserved pins for BGA169 package
- Updated Table 20.8 Register settings for input/output pin function (PORT0)
- Updated Table 20.9 Register settings for input/output pin function (PORT1)
- Updated Table 20.10 Register settings for input/output pin function (PORT2)
- Updated Table 20.11 Register settings for input/output pin function (PORT3)
- Updated Table 20.12 Register settings for input/output pin function (PORT4)
- Updated Table 20.13 Register settings for input/output pin function (PORT5)
- Updated Table 20.14 Register settings for input/output pin function (PORT6)
- Updated Table 20.15 Register settings for input/output pin function (PORT7)
- Updated Table 20.16 Register settings for input/output pin function (PORT8)
- Updated Table 20.17 Register settings for input/output pin function (PORT9)
- Updated Table 20.18 Register settings for input/output pin function (PORTA)
- Updated Table 20.19 Register settings for input/output pin function (PORTB)
- Updated Table 20.20 Register settings for input/output pin function (PORTC)
- Updated Table 20.21 Register settings for input/output pin function (PORTD)

24. Low Power Asynchronous General Purpose Timer:
- Updated Table 24.1 AGT specifications

25. Ultra-Low-Power Timer:
- Updated Table 25.1 ULPT specifications

27. Watchdog Timer:
- Updated Figure 27.1 WDT block diagram
- Updated 27.2.3 WDTSR : WDT Status Register

28. Independent Watchdog Timer:
- Updated Table 28.1 IWDT specifications
- Updated Figure 28.1 IWDT block diagram
- Updated 28.2.3 IWDTSR : IWDT Status Register
- Updated Figure 28.3 Operation example in register start mode
- Updated 28.3.3 Refresh Operation
- Added 28.5.2 Restriction on reading the counter value after returning from Deep Software Standby mode 1

29. Layer 3 Ethernet Switch Module:
- Updated 29.2.2.1 TPEMIMC1 : Error and Monitoring Interrupt Mapping Configuration Register 1
- Updated 29.2.2.2 TPEMIMC2 : Error and Monitoring Interrupt Mapping Configuration Register 2
- Updated 29.2.2.3 TPEMIMC3 : Error and Monitoring Interrupt Mapping Configuration Register 3
- Updated 29.2.2.4 TPEMIMC4 : Error and Monitoring Interrupt Mapping Configuration Register 4
- Updated 29.2.2.5 TPEMIMC6t : Error and Monitoring Interrupt Mapping Configuration Register 6t (t = 0 to 1)
- Updated 29.2.2.6 TPEMIMC7t : Error and Monitoring Interrupt Mapping Configuration Register 7t (t = 0 to 63)
- Updated 29.4.2 Precautions

30. Ethernet Message Forwarding Engine:
- Updated 30.3.1.6.30 FWVLANTEC : VLAN Table Entry Configuration Register

35. Ethernet Generic PTP Timer:
- Updated 35.3.2.3 PTPTIVCt : Timer t Increment Value Configuration Register (t = 0, 1)

37. USB 2.0 High-Speed Module:
- Updated 37.2.32 DCPCTR : DCP Control Register
- Updated 37.2.51 DPUSR2R : Deep Software Standby USB Suspend/Resume Interrupt Register

38. Serial Communications Interface:
- Updated 38.2.2 RDR/RDR_BY : Receive Data Register
- Updated 38.2.3 TDR/TDRLL/TDRLH : Transmit Data Register

Revision 1.30 — Feb 27, 2026

40. I3C Bus Interface:
- Updated 40.2.1 PRTS : Protocol Selection Register
- Updated Figure 40.140 Example of I2C initialization flowchart (single buffer transfer)
- Updated Figure 40.141 Example of each initialization flowchart for I3C master mode and slave mode
- Added 40.7.4 Notes on rewriting PRTS.PRTMD bit

41. CAN with Flexible Data-rate:
- Updated 41.2.43 CFDTMIEC : TX Message Buffer Interrupt Enable Configuration Register

44. Octal Serial Peripheral Interface:
- Updated 44.2.1.4 CMCFG0CSn : OSPI Command Map Configuration Register 0 CSn (n = 0, 1)
- Updated 44.2.1.7 LIOCFGCSn : OSPI Link I/O Configuration Register CSn (n = 0, 1)
- Added 44.2.2.22 RACTL : OSPI Register Access Control Register
- Added 44.2.2.23 MER : OSPI Map End Replace Register
- Updated 44.4 SiP Product Configuration
- Updated Figure 44.32 Connection between the MCU and the SiP Flash memory
- Updated Table 44.11 Data write possibility for each bus master
- Added 44.5.9 Constraint on OTP area for SiP Product

45. Decryption On The Fly:
- Updated 45.3.1 CONVAREAST : DOTF Conversion Area Start Address Register
- Updated 45.3.2 CONVAREAD : DOTF Conversion Area End Address Register

46. Serial Sound Interface Enhanced:
- Updated 46.2.1 SSICR : Control Register

48. Cyclic Redundancy Check:
- Updated 48.3.2 CRC Snoop Function

49. Pulse Density Modulation Interface:
- Updated Table 49.1 PDM-IF specification
- Updated 49.2.17 PDSRCHn : Status Register Channel n (n = 0 to 2)
- Updated Table 49.5 Guaranteed clock frequency and clock mode
- Added 49.3.1.1 Output Sampling Frequency
- Updated Table 49.7 Sinc filter output clip

54. 12-Bit D/A Converter:
- Updated Table 54.4 D/A conversion and analog output control
- Updated 54.4.1 DA0 Event Link Operation Setting Procedure
- Updated 54.4.2 DA1 Event Link Operation Setting Procedure

58. SRAM:
- Updated 58.2.7 SRAMCRn : SRAM Control Register n For ECC RAM (n = 0 to 3)
- Updated 58.3.7 Wait State

59. MRAM:
- Updated Table 59.1 Specifications of MRAM
- Updated 59.5.2 MRCFREQ : Code MRAM Frequency Notifications Register
- Updated 59.5.3 MREFREQ : Extra MRAM Frequency Notifications Register
- Updated 59.5.6 MRCRAES : Code MRAM Read Access Error Status Register
- Updated 59.5.9 MRERAEINT : Extra MRAM Read Access Error Interrupt Enable Register
- Updated 59.5.10 MRERAES : Extra MRAM Read Access Error Status Register
- Updated Figure 59.9 Overview of command usage in extra MRAM program mode
- Updated Table 59.16 Address used by the Configuration Set command
- Updated 59.7.4.9 Increment Counter Command
- Updated Table 59.36 Error protection type
- Updated 59.13.2 Parallel Accessibility

Revision 1.30 — Feb 27, 2026

62. Electrical Characteristics:

- Updated Table 62.2 Recommended operating conditions
- Updated Table 62.34 Coremark and normal mode current, CPU0 Deep Sleep, CPU1 active. (DCDC mode and External VDD mode)
- Added Table 62.42 VCC_USB rise gradient characteristics at power on
- Added Table 62.43 VCC_USBHS and AVCC_USBHS rise gradient characteristics at power on
- Updated Table 62.44 Thermal Resistance
- Updated Table 62.45 Power consumption of each unit (DCDC mode)
- Updated Table 62.46 Power consumption of each unit (External VDD mode)
- Updated Table 62.49 Clock timing except for sub-clock oscillator
- Updated Table 62.51 Reset timing
- Updated Table 62.52 Timing of recovery from low power modes
- Updated Table 62.63 SPI timing
- Updated Table 62.64 OSPI timing
- Updated Figure 62.85 DDR transmit/receive timing (4S-4D-4D, 8D-8D-8D)
- Updated Table 62.78 PDG Timing
- Updated Table 62.107 TSN characteristics
- Updated Table 62.109 Sub-clock oscillation stop detection circuit characteristics
- Added Figure 62.137 Sub-clock Oscillation Stop Detector start-up time
- Updated Table 62.110 Power-on reset circuit and voltage detection circuit characteristics
- Updated Figure 62.139 Power-on reset timing
- Updated Figure 62.140 Voltage detection circuit timing (Vdet0)
- Updated Figure 62.141 Voltage detection circuit timing (Vdetn) (n = 1, 2, 4, 5)
- Updated 62.15 MRAM Characteristics
- Updated Table 62.116 Code MRAM Characteristics
- Updated 62.15.2 Option Setting Memory (Configuration area) Characteristics
- Updated Table 62.120 MACI command Characteristics

Appendix:

- Updated Appendix 1. Port States in Each Processing Mode
- Added Figure A2.3 BGA 303-pin
- Updated Table A3.2 Access cycles

RA8M2 Group User’s Manual: Hardware

Publication Date: Rev.1.30 Feb 27, 2026
Rev.1.20 Jul 16, 2025

Published by: Renesas Electronics Corporation

32-Bit MCU
RA8M2 Group

Renesas Electronics Corporation

R01UH1066EJ0130